PAPER

Fostering Augmented Intelligence in Architectural Education to Address Complexity

DAVID WILLIAM NEWTON
University of Nebraska-Lincoln

Keywords: augmented intelligence, computation, pedagogy, artificial intelligence

In the coming decades, climate change driven migration will dramatically increase urban populations and create profound design challenges. Designing for this context will be difficult and involve balancing multiple objectives. Preparing future generations of architects to tackle such complexity is crucial. Developing pedagogical approaches that can better equip students to manage complex design problems in an increasingly hot and crowded future is, therefore, a pressing problem for architectural education. The emerging field of augmented intelligence explores the combination of human and artificial intelligence to solve complex problems and offers a new lens for architectural education to engage computation. This research presents educational case studies introducing key concepts and tools for augmented workflows in project-based architectural studio settings.

INTRODUCTION

It is estimated that 55% of the world's population currently lives in cities and by 2050 models estimate this number could balloon to 70% as climate change makes farmland unusable and drives mass migration into cities from rural areas[1]. Researchers have identified several megacities around the world that will see dramatic increases in population due to climate change-driven migration – straining the infrastructural capacities of these cities and creating dense living conditions that may be unhealthy and dehumanizing[2]. Providing sustainable design solutions that adequately address the complexity of these scenarios and can balance multiple quantitative and qualitative stakeholder needs is a pressing problem for the architectural discipline. Training future architects to successfully manage such problems is, therefore, crucial. Traditionally, design education has focused on inculcating action-centric models of design that emphasize an improvised search of a space of possibility based on the designer's personal interests and the constraints given by a site and a program brief. These approaches often do not provide concepts, representational techniques, and technologies that can allow students to assess spaces of trade-offs involving multiple competing objectives more rigorously in complex design problems. How can we better equip students to manage complex design problems in an increasingly hot and crowded future? What pedagogical approaches can be used to provide students with the necessary knowledge and skills to engage complexity?

The emergence of artificial intelligence technologies in the last several decades has opened the door to the creation of new ways of defining and searching a space of possible design solutions based on quantifiable criteria (e.g., energy use; daylight exposure, cost, etc.) with unparalleled speed - allowing millions of solutions to be explored in the same time it would take a human designer to explore dozens of design alternatives. Numerous automated strategies have been developed by researchers to efficiently explore design spaces, but researchers have found that augmented intelligence-based processes[3] - in which a human and a search algorithm work together - provide the greatest capabilities to search a design space and find solutions to problems involving many objectives. In these workflows, the dynamic and non-linear styles of human thinking are layered with varying levels of artificial intelligence to create a hybrid style of thinking that can solve problems neither the human nor the algorithm could solve individually. These augmented workflows have demonstrated efficacy in finding efficient designs in complex design scenarios involving multiple objectives but their application in architectural education has been limited in its exploration of how human intelligence can play a role alongside computational processes.

In order to address this gap, this research investigates pedagogical models that integrate augmented-intelligence-driven workflows into project-based educational contexts at multiple levels within an architectural curriculum. These pedagogical models are investigated for their capacity to foster metacognitive knowledge. Metacognitive knowledge can be defined as knowledge about how knowledge is created, or strategic thinking, and its value for managing complex design problems is that it can allow students to reflect on how the design process itself can be structured in order to tackle the specific needs of a complex design problem[4,5]. The research is structured in the following manner: First, precedent research in architectural education involving computational processes are discussed; next, educational case studies are presented that introduce key concepts and tools in relation to augmented workflows; lastly, key challenges and future directions are then discussed - providing a roadmap for the discipline on the value of augmented intelligence-based processes in architectural education.

COMPUTATION IN PROBLEM-BASED ARCHITECTURAL STUDIO PEDAGOGY

The proliferation of low-cost personal computers and computer-aided design (CAD) software in the late 80's and early 90's inaugurated a new era in architectural education in which the integration of computational tools with architectural design studios became ubiquitous and routine. In the decades since, educators have explored a variety of ways that these tools might best be used in project-based studio pedagogy to improve student learning outcomes and the quality of their design proposals. These approaches can be roughly categorized based on how they define the role of the student designer and the computer in the design process in three ways: designer-driven; computationally-driven; and augmented intelligence-driven processes.

The majority of approaches place the student designer in the role of primary author of the design, while the computer plays a largely representational role as a drafting machine that can also provide some analysis capabilities. These designer-driven workflows allow the student the most agency to control how a design problem is explored and to drive the design based on factors that are hard to quantify (e.g., experiential goals, aesthetics, etc.) but are limited by the speed in which a student, or team of students, can develop and evaluate design solutions. Further, these processes often lack methods and tools that allow students to rigorously explore the trade-offs between the multiple objectives present in any studio-based design project, as well as the ramifications of prioritizing those objectives in different ways. The result is a process that often produces designs that have little to no evidence, beyond subjective speculation, that they meet the goals of the project.

The second category of approach involves placing the computer in the role of the primary author, while the student designer plays a secondary role in the design process - as an interpreter and editor of the created design. In these computationally-driven processes, generative algorithms are used to produce and evaluate designs with minimal designer input. A wide variety of generative design algorithms have been explored for automating design tasks, such as architectural plan generation[6], building massing development[7], and building envelope design[8]. The advantages of these processes reside in the ability to harness the speed and intelligence of computer-coded algorithms to explore a space of possible designs and their trade-offs in an automated fashion that can be quantitatively described. Even with this advantage, these approaches are the least used in problem-based studio settings. This is due to the technical difficulty involved, the limited control, and lack of intellectual engagement that students often have in these workflows. Further, these processes are biased heavily towards design objectives that are easily quantifiable—giving qualitative factors less priority in design development.

The third category of approach splits the role of primary author between both the student designer and the computer. In these augmented intelligence-driven approaches, the computer is used as a collaborative intelligence in the design process – aiding in design invention, while allowing the student designer to apply their creative imagination and pattern finding abilities at the same time. This human-computer collaboration has the effect of layering human and computational forms of intelligence to solve design problems that the other two categories of approach may find mathematically intractable. Research in multi-objective optimization has demonstrated this benefit - especially for design problems that have more than three objectives[3]. These processes are also able to allow both quantitative and qualitative design objectives to steer design invention, while providing rigorous methods to justify developed designs relative to the trade-offs between design objectives. One of the more common applications of these workflows in architectural education is the use of interactive multi-objective optimization processes, in which the student works interactively with an optimization algorithm to explore design variations relative to quantitative and qualitative objectives[9,10].

Despite the benefits of these augmented workflows, their use in architectural design studios has been limited by a couple key factors. First, the lack of a coherent vocabulary and theory for augmented intelligence in architectural education has limited the discipline's ability to recognize and explore the pedagogical benefits of these processes. Second, there has been a limited variety of examples of such workflows, their potential pedagogical benefits, and how they might be integrated into different levels of an architectural curriculum. Lastly, the technical skills and computational resources needed in these workflows pose considerable challenges to students and educators who may be new to this way of working.

COMPUTATIONAL THINKING IN ARCHITECTURAL EDUCATION

The proliferation of computational technologies in architectural education has posed a significant challenge for educators concerning how these tools can be most effectively integrated into the cognitive processes associated with architectural design. One popular approach to address this problem, called "computational thinking", comes from the field of computer science. Computational thinking is characterized by the following concepts: problem decomposition; data representation and pattern recognition; generalization and abstraction; and the use of procedural heuristics/algorithms[11]. Its proponents argue that computational thinking provides fundamental skills in problem solving that can be applied outside of computer science in a variety of disciplines - from science and engineering to the humanities and the arts.

In the field of architectural education, it has been used to provide a conceptual framework for how computation might be theorized in architectural pedagogy[12]. Its critics argue that its description is vague and that many of its components are not unique compared to patterns of though found in other disciplines[13]. Another limitation is that it tends to emphasize a procedural thinking style. This bias tends to discourage an approach to architectural design that values diverse styles of thinking (e.g., associative, metaphorical, inductive, etc.) in the design process. This runs counter to evidence that such diversity can improve the ability to solve problems[14]. Further, it does little to promote a higher-level metacognitive understanding of the role of computational intelligence and human intelligence in the design process. Fostering metacognitive thinking can help students address complex design problems by allowing them to better understand and strategize the role of technology as well as their own thinking patterns in the design process. Architectural education is, therefore, in need of alternative conceptual frameworks beyond computational thinking that can address these limitations.

AUGMENTED THINKING IN ARCHITECTURAL EDUCATION

Architectural education has sought concepts to help guide the integration of computational technologies into problem-based studio settings from the realm of computer science, but perhaps the wrong principles from this field have been imported, while other, more beneficial, concepts have been overlooked. Artificial intelligence is a sub-field within the larger umbrella of computer science that is devoted to building intelligent entities. It is a field that draws on a variety of disciplines, such as cognitive science, neuroscience, philosophy, psychology, economics, mathematics, linguistics, computer engineering, and control theory, as the basis for defining and understanding intelligence. Through the lens of artificial intelligence, computational processes can be understood as a form of intelligence. The software that we interact with everyday can also be understood as having different levels of intelligence and agency. When computation is viewed from this perspective, it can be understood as a mirror and an extension of human thought processes and reasoning, while also bringing to the table new and non-human styles of thinking.

Through an emphasis on artificial intelligence, a pedagogical approach around computation and design that emphasizes metacognitive knowledge can be fostered. This approach, that will be referred to as *augmented thinking*, encourages students to reflect on their own thinking patterns in the design process and to become aware of other styles of thinking - including those of computational agents. The design process is, therefore, understood as ensuing from a system of networked intelligences and thinking styles of different degrees and kinds. The notion of augmented intelligence and workflows then naturally follow, and students are encouraged to explore how combining multiple thinking styles can solve problems that may be difficult, or impossible, to solve any other way. Further, fostering this form of metacognitive thinking can help students address complex design problems by allowing them to better strategize the role of computation as well as their own thought and reasoning processes in the design process.

The field of artificial intelligence, with its emphasis on metacognition, may, therefore, provide a more fruitful source of concepts to theorize the role of computational technologies in architectural education. This naturally leads to several important questions. How can augmented thinking and workflows be brought into architectural studio curriculums at beginning, intermediate, and advanced levels? What specific computational processes can be used? The next sections introduce educational case studies that explore these questions within project-based architectural studio settings.

AUGMENTED INTELLIGENCE IN BEGINNING DESIGN

Teaching architectural students effectively requires a critical reflection on the processes that underlie learning and invention. In 1964, Benjamin Bloom proposed a general framework for describing the essential steps in learning[4]. This framework has been widely studied over the years, and also criticized as being an over-simplified view of learning that assumes a sequential process of learning and ignores the role of motivation[5]. In response to these criticisms, a revised version of Bloom's taxonomy was created in 2001 that attempts to emphasize the dynamism present in the learning process[15]. This new taxonomy attempts to emphasize the dynamism of the learning process by describing it in terms of six cognitive processes: remembering; understanding; applying; analyzing; evaluating; and creating. Knowledge is placed at the base of these six categories and is broken-down into four subcategories: factual knowledge; conceptual knowledge; procedural knowledge; metacognitive knowledge.

In beginning architectural design studios, students are new to the discipline and learning a number of design fundamentals that make teaching and learning in these levels challenging. In most curriculums, factual (i.e., learning terminology and the concrete facts within a discipline), conceptual (i.e., learning principals and theories), and procedural knowledge (i.e., specific skills, processes, methods, and algorithms) is prioritized in these early studios, while metacognitive knowledge, which deals with the assessment and strategic planning of one's own cognitive processes, is typically not emphasized as heavily, if at all. Instead, students are often asked to create and navigate their own design processes absent a metacognitive knowledge-base that might help them create design processes that are better tailored to a specific design scenario, or set of priorities.

In order to introduce metacognitive thinking in early studios, an emphasis on augmented intelligence was used in the instruction of a 3rd year architectural studio at the University of Nebraska-Lincoln. In the studio, students were specifically asked to combine styles of thinking that are computational in nature with associative and poetic modes. This was done as a way of introducing the notion of thinking about thinking and also how thinking styles have a relation to how designers might search a space of possibility.

In Figure 1, the work from one exercise is shown where the students are first asked to develop an idiosyncratic procedural algorithm and to explore, by hand through model making, how variations in its rules can produce different spatial-structural organizations. Students are then asked to choose an artist and to try to describe their thinking style (e.g., How do they define problems? How do they address them?). They are then asked to modify their original algorithm to be reflective of the thinking style of their chosen artist. Students then explore the capabilities of this hybrid algorithmic process to produce spatial-structural and programmatic organizations through model making and digital drawings.

The process described in this exercise encourages students to understand the design process itself as a design problem. The discussion of thinking styles in the exercise further encourages them to think metacognitively and to understand thinking itself as a multiplicity; with many different thinking styles; each bringing different opportunities to a design process. The emphasis on executing algorithms by hand throughout the exercise helps to make algorithms less abstract, tactile, and consequential, while avoiding the challenges of introducing computer programming languages to a curriculum which is already packed-full for students and faculty. The result is a process that merges computational styles of thinking with other non-linear modes to create an augmented analog workflow that builds metacognitive knowledge about the design process that can hopefully help students better navigate more complex design problems as they move into more advanced levels of the curriculum.

Another exercise at this level involves having students explore the concept of computational intelligence through directly engaging computational processes. In the exercise, students collect and curate a dataset of precedent design images of architectural facades. Students then use these images to train a deep learning model. The model learns the stylistic structures behind the images - forming a type of intelligence from its experience with the dataset. Students then explore this intelligence and use the deep learning model to create a series of novel hybrid design patterns based on this curated intelligence. Images of some of these deep learning generated façade images are shown in Figure 2. They are then asked to interpret and use these images as a starting point for their own façade design investigations. The exercise therefore initiates a metacognitive discussion about intelligence in the design process while demonstrating how human and computational intelligence might be combined to illuminate a space of design possibilities that might be difficult to find without a combination of different thinking styles.

AUGMENTED INTELLIGENCE IN INTERMEDIATE DESIGN

For the intermediate levels of the design curriculum, students are introduced to computational processes that allow for more sophisticated augmented intelligence-driven workflows to

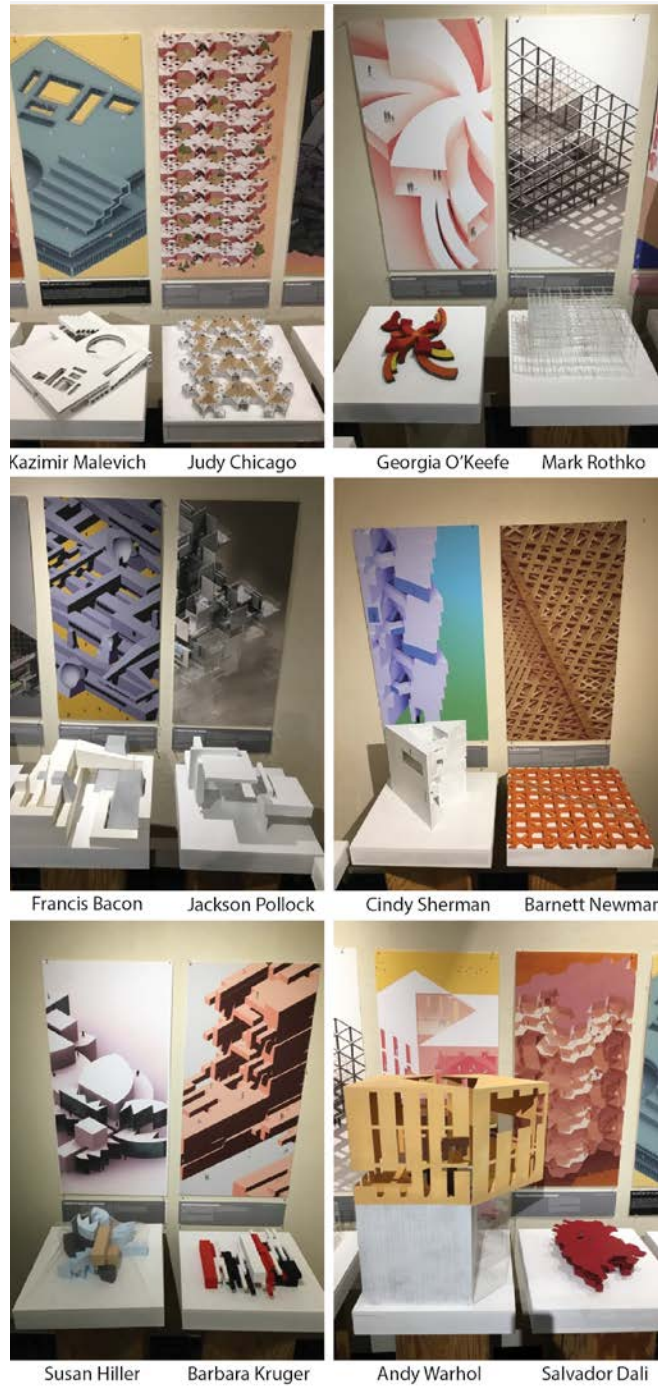

Figure 1. Samples of student work for a 3rd year architetcural studio are shown that explore augmented intelligence concepts. For the exercise, students develop an idiosyncratic algorithm by hand and are then asked to modify their original algorithm to be reflective of the thinking style of a chosen artist. They produce models and drawings exploring the created process.

Figure 2. Three student work samples are shown from an augmented intelligence workflow that involves using deep learning to create hybrid design patterns based on curated image sets of facades. These images are then creatively interpretted by students in an iterative fashion to develop the designs further.

deal with design problems that are more complex in nature that require balancing multiple quantitative and qualitative objectives and understanding the trade-offs between those objectives. The comprehensive design studio at the University of Nebraska-Lincoln provides a project-based learning experience that exemplifies this kind of complexity and challenge for students. In the studio, concepts and technologies from the field of optimization, which is a sub-field of artificial intelligence, are integrated during each of the major phases of the term project to build metacognitive knowledge.

In the early phases of the term, concepts from the field of optimization are introduced and used to give the students a vocabulary to discuss multi-objective design problems and analyze precedent projects from a metacognitive perspective. Key terms introduced that help to shift student thinking from a focus on a singular design to a space of multiple design possibilities and their trade-offs include the following: decision/design variables; decision space (e.g., space of all possible designs); quantitative design objectives (e.g., measurable building performances); qualitative design objectives (e.g., aesthetic, experiential, and conceptual goals); objective space (e.g., space of all possible trade-offs between objectives).

In the latter phases of the term, methods and digital tools for decision space definition and exploration are introduced in the form of parametric modelling. Simulation tools for the evaluation of objectives are also introduced, along with computational multi-objective optimization tools to help students search a space of possibility and comparatively evaluate design solutions. In these augmented workflows, computational tools are used for their speed in searching through design variations, while the students use their creative imagination and pattern-finding abilities in an iterative process that involves the following: creating an initial design; steering an optimization search process to refine that design; interpreting the results and the trade-offs between designs; refining the design based on the results; conducting a refined optimization search; and repeating.

Figure 3 shows an example student project from the comprehensive studio involving the design of a research institute. The top left of the figure shows a map of the objective space and the trade-offs between designs for the optimization of the primary structural system of a portion of the project. The top right of the figure shows a similar objective space map for the optimization of the glazing for a portion of the building envelope. These maps foster metacognitive thinking and are used to help the students understand the "landscape" of their design space; to identify areas that are under-explored; to foster discussions on how design spaces might be searched; and to build knowledge relating design variations to their performance trade-offs. This type of strategic thinking allows students to breakdown the complexity of a comprehensive design problem and to approach the integration of multiple architectural systems and design objectives in a more rigorous fashion.

AUGMENTED INTELLIGENCE IN ADVANCED DESIGN

In advanced studio levels, students are asked to tackle increasingly complex problems using augmented workflows that combine several different forms of computational intelligence with their own. In these studios, students start by defining a design language that is governed by a rule system that determines its growth. This rule system is executed by hand and is reflective of their own intuitive mode of thought and can be non-linear and fuzzy. They are then asked to select and study an algorithm from a list of algorithms that have diverse capabilities in terms of their degree of intelligence and style of computation. From there, they are asked to create a hybrid process - mixing the idiosyncratic analog algorithm with the chosen computational algorithm. This hybrid process is encoded in computer code and used as a kind of sketching tool that students are asked to explore intuitively. This sketching tool illuminates a space of possibility

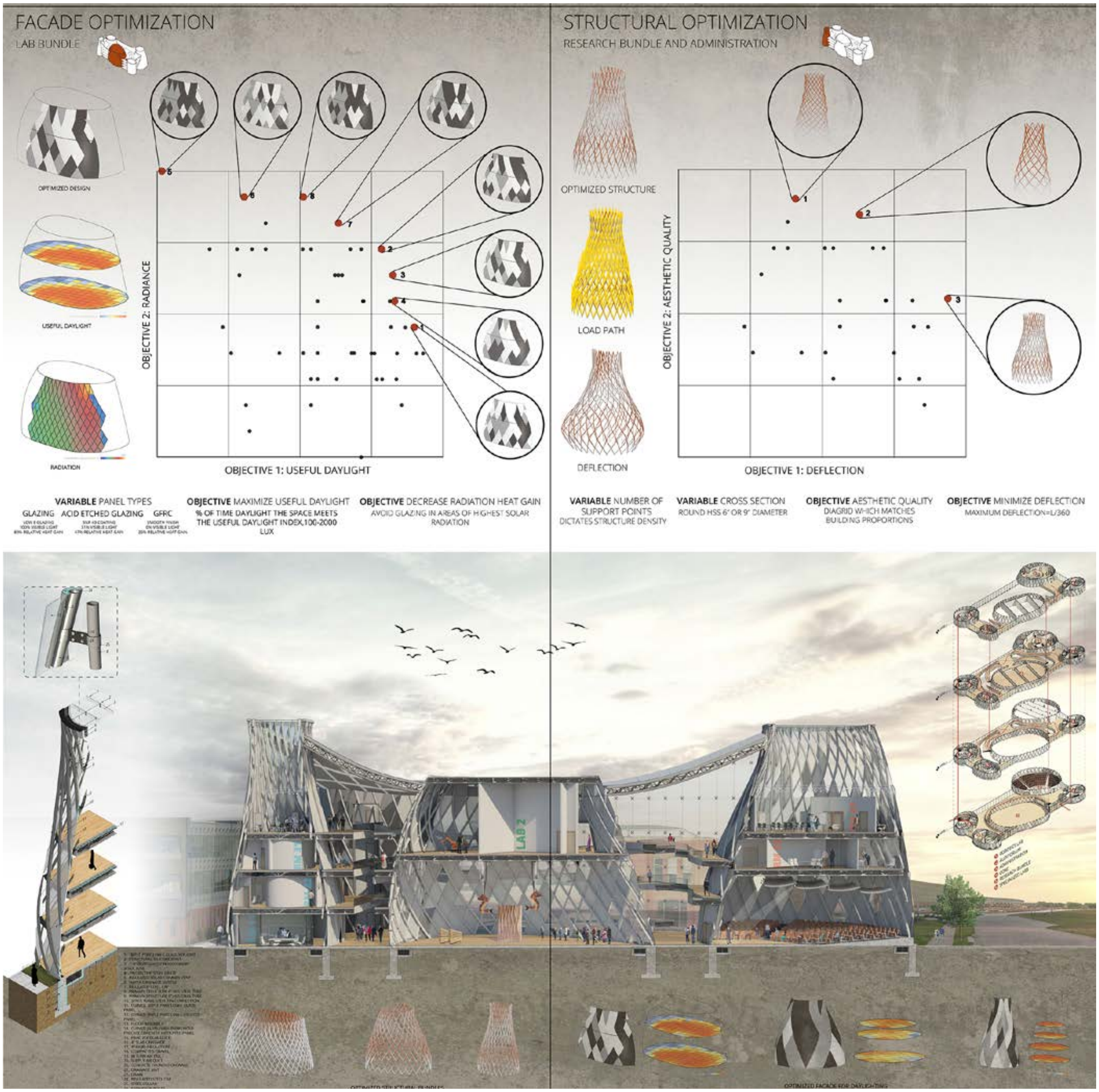

Figure 3. Pictured above is work from the latter stages of design. (Top Right) The primary structural system is optimized for deflection as well as an aesthetic objective related to the smoothness of the structural pattern. (Top Left) The glazing pattern of the facade is explored in relation to useful daylight and solar irradiance objectives. For both tasks, a graph of the objective space with the Pareto optimal solutions is shown.

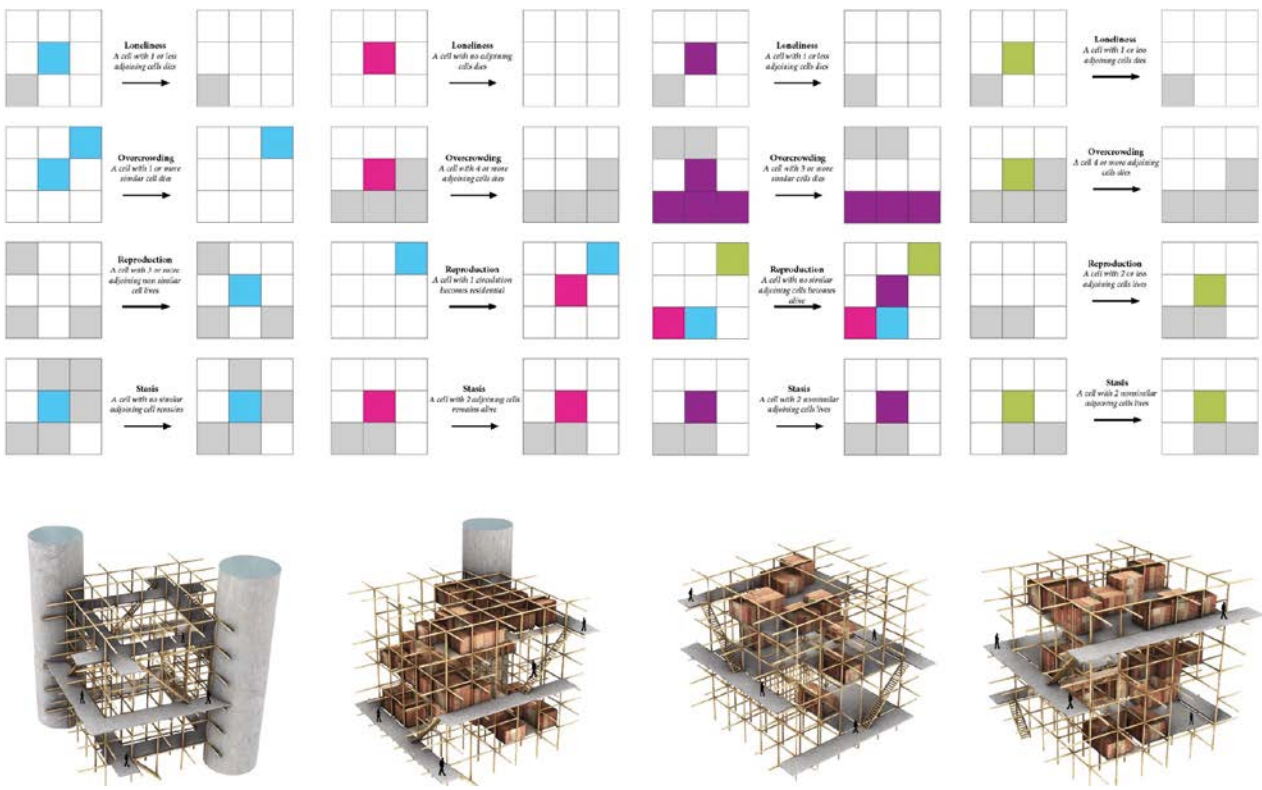

Figure 4. The image features a student work example from a graduate studio at the UNL exploring the creation of a generative sketching tool that combines two different thinking patterns: one that is student defined and one based on an established algorithm.

that was previously inaccessible for each student. Figure 4 shows an example of one student's sketching tool based on the logic of cellular automata.

After exploring this tool's capabilities, students are given a design problem with a specific site and objectives and asked to layer another type of intelligence onto the system that can help them explore their system more intentionally relative to quantitative and qualitative objectives. Specifically, they are asked to add an optimization algorithm to their system along with various performance simulation tools (e.g., daylighting, structure, energy simulation, etc.). Students are then asked to take on the role of a kind-of cartographer and are asked to use the optimization process to map a space of possibility and to note important landmarks in the design space they have uncovered. In the examples shown in Figure 5, the project in part a of the figure is exploring the design of a building system that can provide an infrastructural scaffolding for informal dwellings in Mumbai. The image in the upper left is a design space map in which objectives for sun exposure, structure, and circulation efficiency are being optimized for. Part b of the figure shows a similar mapping for a different student project exploring the development of a new high-density residential fabric that integrates traditional agricultural practices in Mexico City.

Similar to the work discussed at the intermediate level, this map is used by the students to understand the trade-offs between the objectives for the project and to stimulate discussions around integration – that is how objectives are related to one another and might be prioritized. These maps are also used as a means to steer the process based on qualitative goals – this is done by selecting designs of interest that focusses the optimization process in a particular direction. The result is a process that layers computational thinking styles with those of a human designer – allowing a space of tradeoffs between multiple objectives to be more rigorously explored and understood, while building metacognitive knowledge about the role that computational intelligence can play in the design process.

CONCLUSIONS

The pedagogical approach described in this work is the result of a dialogue between teaching and research and is centered around a few key ideas. First, that fostering metacognitive thinking can help students address complex design problems by allowing them to better understand and strategize the role of technology as well as their own thinking patterns in the design process. Second, that a way to foster metacognitive thinking involves engaging computation from the lens of artificial intelligence instead of computational thinking. Lastly, that preparing students to tackle complex design problems requires a shift to pedagogical approaches that integrate augmented intelligence

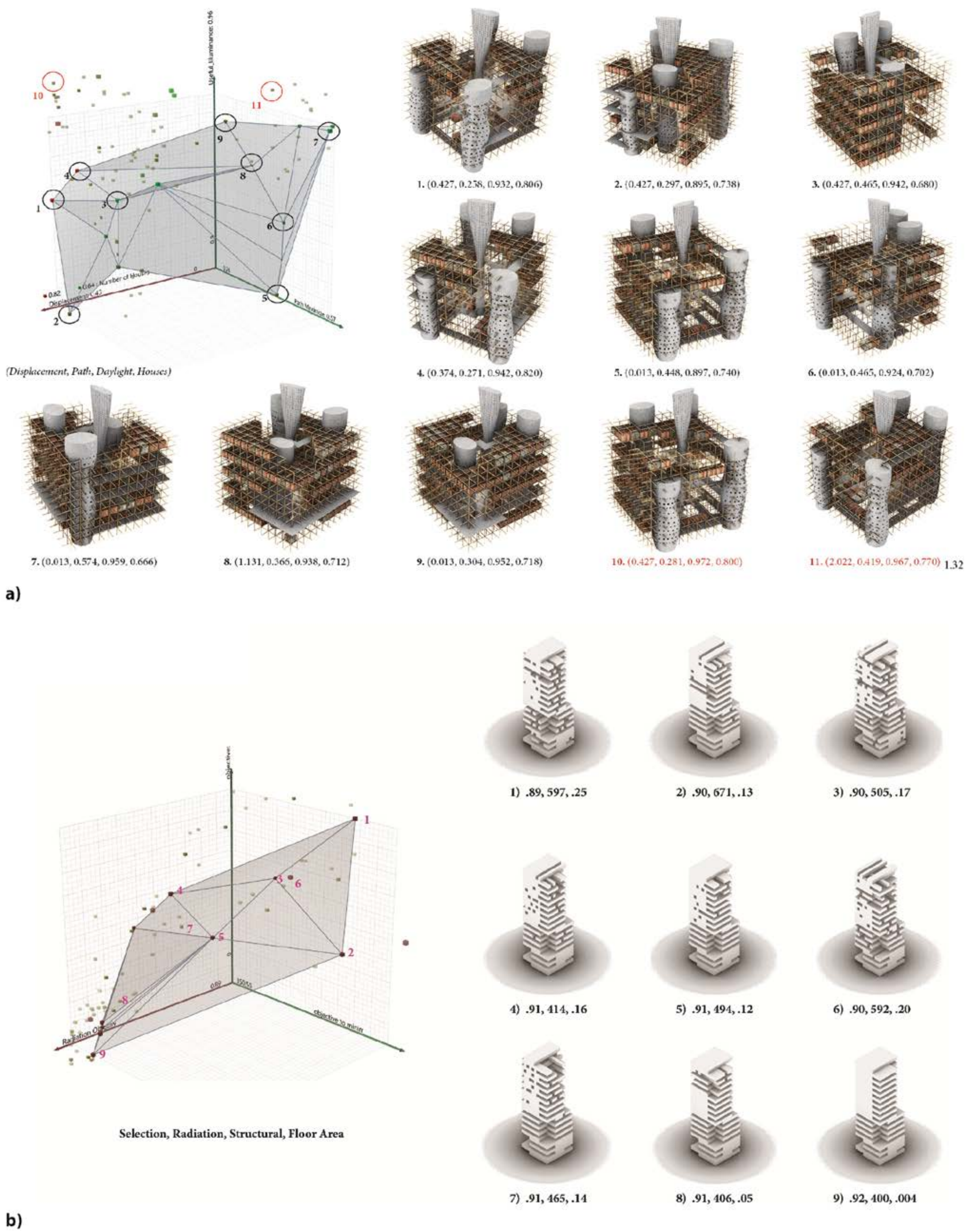

Figure 5. a) The pictured design uses multi-objective optimization to explore and visualize a space of trade-offs relating design features to performance for the design of flood resistant infrastructural system for informal dwellings in Mumbai. b) Shows a similar mapping for the design of a new residential fabric that integrates traditional agricultural practices in Mexico City.

workflows. These workflows combine human and artificial intelligence and have the potential to solve problems neither could solve by themselves.

The educational case studies presented provide examples of how these ideas might be integrated into multiple levels of an architectural curriculum, but there are significant challenges and areas for future work worthy of brief discussion. The biggest challenge in working with augmented workflows is the technical skillsets required by instructor and student to engage them. One way to address this issue, described in the case studies, was to have students work by hand to explore algorithms and augmented thinking. This helps to make these abstract processes tangible, while emphasizing the role of cognitive processes in the design process. Another possible solution could involve the creation of shared instructional resources for architectural education on augmented workflows that could help beginners more easily engage these processes. Further, the software development community could play a role by making these processes more accessible and user friendly, just as CAD technologies have become more capable and accessible over the years.

If this challenge is addressed, augmented workflows could be more widely engaged and inform a new generation of design processes and technologies that could help students and architects address design problems that have previously been difficult, or impossible, to tackle with traditional design methods. The hot, crowded, and resource depleted future that is taking shape will require methods like these that can address complexity in new ways. The allied design fields will, therefore, need to adapt and fundamentally rethink the role of the computer in the design process in order to meet the needs of this future.

ENDNOTES:

1. DESA, UN. 2018. “Revision of world urbanization prospects.” UN Department of Economic and Social Affairs 16.
2. Rigaud, Kanta Kumari, Alex De Sherbinin, Bryan Jones, Jonas Bergmann, Viviane Clement, Kayly Ober, Jacob Schewe, Susana Adamo, Brent McCusker, and Silke Heuser. 2018. “Groundswell.”
3. Bechikh, Slim, Maha Elarbi, and Lamjed Ben Said. 2017. “Many-objective Optimization Using Evolutionary Algorithms: A Survey.” In Recent Advances in Evolutionary Multi-objective Optimization, 105-137. Springer.
4. Bloom, Benjamin Samuel, Committee of College, and University Examiners. 1964. Taxonomy of educational objectives. Vol. 2: Longmans, Green New York.
5. Krathwohl, David R. 2002. “A revision of Bloom’s taxonomy: An overview.” Theory into practice 41 (4):212-218.
6. Galle, Per. 1981. “An algorithm for exhaustive generation of building floor plans.” Communications of the ACM 24 (12):813-825.
7. Aranda, Benjamin, and Chris Lasch. 2006. Pamphlet Architecture 27: Tooling: Princeton Architectural Press.
8. Turrin, Michela, Peter von Buelow, Axel Kilian, and Rudi Stouffs. 2012. “Performative skins for passive climatic comfort.” Automation in Construction 22:36-50. doi: 10.1016/j.autcon.2011.08.001.
9. Benjamin, David. 2012. “Beyond Efficiency.” Digital Workflows in Architecture:14-25.
10. Buelow, Peter Von. 2016. “Genetically enhanced parametric design in the exploration of architectural solutions: Beyond their Limits.” In Structures and Architecture, edited by Cruz. London: Taylor & Francis Group.
11. Wing, Jeannette M. 2006. “Computational thinking.” Communications of the ACM 49 (3):33-35.
12. Senske, Nick. 2017. “Evaluation and impact of a required computational thinking course for architecture students.” Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.
13. Nardelli, Enrico. 2019. “Do we really need computational thinking?” Communications of the ACM 62 (2):32-35.
14. Lamm, Alexa J, Catherine Shoulders, T Grady Roberts, Tracy A Irani, Lori J Unruh Snyder, and Joel Brendemuhl. 2012. “The Influence of Cognitive Diversity on Group Problem Solving Strategy.” Journal of Agricultural Education 53 (1):18-30.
15. Anderson, Lorin W, David R Krathwohl, P Airasian, K Cruikshank, R Mayer, P Pintrich, James Raths, and M Wittrock. 2001. “A taxonomy for learning, teaching and assessing: A revision of Bloom’s taxonomy.” New York. Longman Publishing. Artz, AF, & Armour-Thomas, E.(1992). Development of a cognitive-metacognitive framework for protocol analysis of mathematical problem solving in small groups. Cognition and Instruction 9 (2):137-175.