Hierarchical Discrete Lattice Assembly: An Approach for the Digital Fabrication of Scalable Macroscale Structures

Miana Smith*
Center for Bits and Atoms
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
miana@mit.edu

Paul Arthur Richard*
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
paul.richard@epfl.ch

Alexander Htet Kyaw
School of Architecture and Planning
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
alexkyaw@mit.edu

Neil Gershenfeld
Center for Bits and Atoms
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
gersh@cba.mit.edu

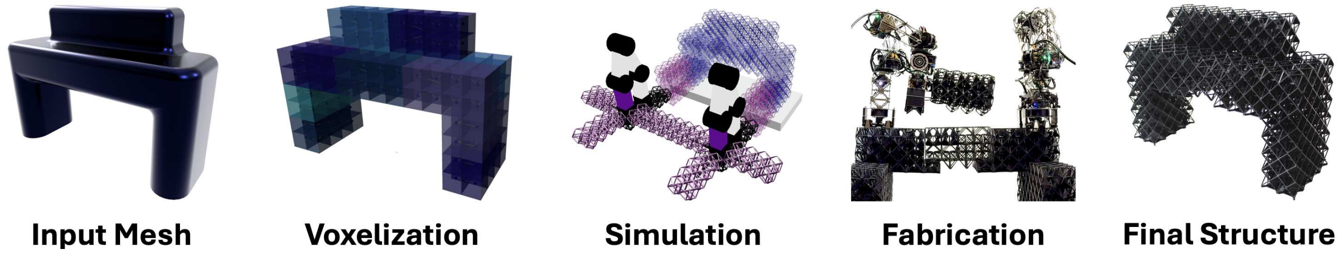

Figure 1: Pipeline from STL Mesh to voxelization, to simulation, to robotic assembly, to the final structure for a small bench.

Abstract

Although digital fabrication processes at the desktop scale have become proficient and prolific, systems aimed at producing larger-scale structures are still typically complex, expensive, and unreliable. In this work, we present an approach for the fabrication of scalable macroscale structures using simple robots and interlocking lattice building blocks. A target structure is first voxelized so that it can be populated with an architected lattice. These voxels are then grouped into larger interconnected blocks, which are produced using standard digital fabrication processes, leveraging their capability to produce highly complex geometries at a small scale. These blocks, on the size scale of tens of centimeters, are then fed to mobile relative robots that are able to traverse over the structure and place new blocks to form structures on the meter scale. To facilitate the assembly of large structures, we introduce a live digital twin simulation tool for controlling and coordinating assembly robots that enables both global planning for a target structure and live user design, interaction, or intervention. To improve assembly throughput, we introduce a new modular assembly robot, designed for hierarchical voxel handling. We validate this system by demonstrating the voxelization, hierarchical blocking, path planning, and robotic fabrication of a set of meter-scale objects.

CCS Concepts

• **Computer systems organization** → **External interfaces for robotics**; **Robotic components**; • **Human-centered computing** → *User interface toolkits.*

Keywords

Large-Scale Fabrication, Digital Fabrication, Modular Assembly, Robotics, Cellular Materials

ACM Reference Format:
Miana Smith, Paul Arthur Richard, Alexander Htet Kyaw, and Neil Gershenfeld. 2025. Hierarchical Discrete Lattice Assembly: An Approach for the Digital Fabrication of Scalable Macroscale Structures. In *ACM Symposium on Computational Fabrication (SCF '25), November 20–21, 2025, Cambridge, MA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3745778.3766665

1 Introduction

Machines for digital fabrication typically cannot make objects or structures larger than themselves. Although this is not necessarily limiting on the desktop scale, as target structures scale toward human or architectural scales, the prospect of ever-growing machine footprints eventually becomes untenable [Meisel et al. 2022]. Current digital fabrication approaches for larger-scale structures often section the target structure into parts that can be produced

*Both authors contributed equally to this research.

SCF '25, Cambridge, MA, USA

ACM ISBN 979-8-4007-2034-5/2025/11
https://doi.org/10.1145/3745778.3766665

separately and then assembled, usually manually [Formlabs 2023] [Kovacs et al. 2017], [Baudisch et al. 2019]. Approaches that consider assembly automation often rely on large and expensive industrial robotic arms with limited overall mobility [Menges et al. 2017]. In either case, there is no clear strategy toward scaling to unlimited build footprints or autonomous environments.

In contrast, collective robotic construction, in which a group of robots collaborate to assemble a structure larger than any individual robot, offers significant promise towards scale-agnostic, efficient, and autonomous fabrication [Petersen et al. 2019]. However, hardware demonstration of these systems has typically been limited in scale [Jenett et al. 2019], load capacity [Petersen et al. 2011], or geometric freedom [Melenbrink et al. 2021]. This is because automated large-scale fabrication with mobile robots is inherently challenging, with many competing priorities, such as structural stability versus mass efficiency, geometric complexity versus ease of robotic fabrication, or robotic functionality versus swarm simplicity.

By designing a material system together with its robotic assembly system, we can address some of these issues. The mechanical interplay between the assembly robots and the underlying material system can enable local error correction, allowing relatively simple robots to build large and precise structures [Jenett et al. 2019]. In this work, we present an approach to the assembly of large structures using small relative robots that manipulate hierarchical blocks of lattice material. Our approach uses a new type of discrete lattice block and robotic system aimed at improving assembly throughput and mechanical stability. We use standard digital fabrication processes to pre-produce compounded blocks of architected lattices– a high efficiency material system– at the size scale of tens of centimeters, and then use mobile robots to take over beyond that, to the meter scale. To ensure structural stability, we developed a tool similar to a 3D printing slicer that imports a 3D mesh, voxelizes it, and finds interleaved patterns for robots to assemble. It then simulates the assembly live, streaming instructions to the hardware system. Our contributions are:

- An interactive tool for voxelizing 3D shapes and finding connected patterns for different shapes of blocks.
- A digital twin path planning and simulation environment integrated with the hardware system, for both assembly control and live design and re-design of structures.
- The development of a compounded set of self-aligning, interlocking, and load bearing octet lattice blocks designed for robotic assembly.
- The development of a new modular inchworm style assembly robot, designed for hierarchical assembly and low system cost and complexity.

2 Related Work

Our approach builds on existing research in scalable building systems, robotic construction and assembly systems, and material-robot and modular robotic systems. In this section, we overview these fields and identify trends and limitations that guided our approach in this work.

2.1 Scalable Building Systems

Although this work is focused on the robotic assembly of structures, scalable prototyping systems designed for human use offer useful insights into reliable mechanical design for building at larger scales. Specifically, we draw from systems that are not significantly limited by the build volume of the fabrication systems used. Broadly, these systems tend to fall into three categories: continuous, sectioned, and discrete fabrication processes.

2.1.1 Continuously Fabricated Systems. An example of a continuous fabrication system capable of producing meter-scale structures is Protopiper [Agrawal et al. 2015], which extrudes tape-based pipes to draw out human- to architecture- scale structures. However, this system has limited load bearing capacity. Wire benders, such as [PensaLabs 2025], are similarly able to extrude much larger structures than the machine itself, and with potentially good load-bearing capacity, such as in [Bhundiya and Cordero 2023]. However, these continuous extrusion based processes have no mechanisms for preventing error accumulation in the built structure, and instead rely on the precision of the machine, or human intervention, to correct problems as they arise— a potentially significant issue, given that a small error in bend angle over a long distance can result in substantial deflection from the target shape.

2.1.2 Sectioned Fabrication and Assembly. Splitting up a target structure into subcomponents can help ameliorate some of the issues with error accumulation, while additionally opening up a wider array of fabrication systems. These approaches are typically geared toward breaking a larger model down into either 2D components that can be cut and assembled together, or smaller 3D models that can be separately fabricated and assembled. LuBan3D [Luban3D 2025] is a software package aimed at doing exactly this: facilitating users to fabricate structures much bigger than the machines they have, such as the house in [Sass and Botha 2006]. More specialized examples aimed at non-expert users include Kyub [Baudisch et al. 2019], which enables the design and automatic creation of load-bearing structures made from laser cut plywood, or HingCore [Abdullah et al. 2022] and PopCore [Abdullah et al. 2024], which offer joinery-free methods for processing foamcore or equivalent sandwich panel material. The primary issue with these methods is that they produce highly specific parts for specific designs, often requiring dexterous methods for assembly. This makes automation much harder: handling extremely diverse geometries in many degrees of freedom is a hard problem in robotics. Additionally, this typically prevents re-use of parts, as components are likely not portable to new designs, requiring full re-fabrication and re-assembly to make design iterations.

2.1.3 Discrete Assembly. Continuing the trend toward breaking up a big structure into smaller parts is the class of systems that use discrete assembly, where a single part type (or a small library of component types) is used to iteratively build many different target structures. A classic example of this is LEGO®, which enables an imprecise assembler (e.g. a child) to build precise and relatively robust structures. This is because the material system is designed to self-align and error correct during the assembly process, so that the accuracy of the final structure is primarily determined by the tolerances in the part manufacturing, and not by the tolerances of the assembler, in contrast to continuous extrusion fabrication processes. LEGO® uses a volumetric brick-based decomposition, but other forms exist, including strut-and-node systems [Kovacs et al. 2017], facet-based assemblies [Jenett et al. 2018], and voxel-based assemblies [Gregg et al. 2018]. This style of system lends itself well to robotic fabrication, as it features a limited and similar set of parts, often designed to accommodate imprecise assemblers.

Among discrete assembly approaches for large-scale structures, we focus on using discrete architected lattices, as these systems offer competitive mechanical properties at a lightweight [Schaedler and Carter 2016]. This is desirable both to improve the performance of the resultant structure and to lower the payload demands on an assembly robot, which reduces some engineering challenges associated with heavy building materials [Goessens et al. 2018]. Discrete versions of architected lattices have demonstrated extreme performance results [Cheung and Gershenfeld 2013], designed material anisotropies [Jenett et al. 2020], and integrated electronic functionality [Smith et al. 2025]. Manually, these have been assembled into a wide array of meter-scale structures, such as morphing aero- and hydro- structures [Jenett et al. 2017] [Parra Rubio et al. 2023], vehicles [Jenett 2020], or load-bearing static structures [Smith et al. 2025], demonstrating the broad applicability that is desirable for a generalist building system.

2.2 Robotic Fabrication Systems

We now consider the question of how to automate the building of large structures. Approaches for this, again, roughly fall into three categories: large machines to build large structures, large robots to assemble large structures, or small mobile robots to assemble large structures.

2.2.1 Static Gantries. In the first approach, a typically gantry-based machine is used to assemble a structure, as in [Apolinarska et al. 2016], or, more commonly, to 3D print a structure, as shown in [Batikha et al. 2022] often using systems in the style of [COBOD 2025], for example. Though these approaches have been demonstrated at commercial architectural scales, they are still fundamentally limited by the need for very large machines, which introduces both an upper limit on the size of the structure that can be made, as well as significant logistics challenges, either in terms of transporting parts, in the case of pre-fabrication, or building the machine in place, in the case of in-situ fabrication. With the extrusion based approaches, there are additional open challenges in material formulation and performance [Marchment and Sanjayan 2020] [Roux et al. 2023]. For these reasons, we are interested in further exploring assembly based approaches.

2.2.2 Industrial Arms. At the large, complex, and expensive side of the assembly based approaches are projects that have focused on using industrial robotic arms for assembly. These have demonstrated a range of material types, such as steel rebar [Ma et al. 2020], timber [Apolinarska et al. 2021], or bricks [Gharbia et al. 2020], including commercial systems such as [HadrianX 2025] or [Construction-Robotics 2025]. However, these systems have limited accessible footprints, based on the reach of the robotic arm (or the linear axis it is mounted on), or feature significant localization challenges, if on a mobile base [Bodea et al. 2022]. These projects often rely on the high performance of their robotic and sensing systems to compensate for the simplicity in their building materials— by shifting some of the complexity onto the material system, we can reduce some of the demands (and size and cost) on the robotic system.

2.2.3 Material-Robot Systems. Building systems that rely on the co-design and interaction of the robots and material systems are often referred to as material-robot systems [Jenett et al. 2019]. The degree of material-robot integration varies across the literature. At one extreme end, self-reconfiguring modular robots, as in [Neubert and Lipson 2016] or [Hauser et al. 2020], could be considered a material-robot system where there is no distinction between the building system and the robot, resulting in a very complex material system. At the other end, systems such as [Goessens et al. 2018] or [Leder et al. 2022] only shift a minimal amount of complexity onto the material system.

For discrete lattices, systems for voxel-based assembly, such as in [Jenett et al. 2019], [Smith et al. 2024], or [Gregg et al. 2024], as well as strut-based assembly, such as in [Hsu et al. 2016] or [Yoon and Rus 2007] have been explored. [Jenett et al. 2019] demonstrated that an inchworm style robot could accurately traverse and assemble a voxel structure with no global feedback, based on the error correction between the voxels and the robot. However, this system relies on magnets and an underlying steel table to form connections and provide mechanical stability, and has low load bearing capacity on its own. This system is evolved in [Park et al. 2023] to demonstrate the assembly of 100s of voxels [Gregg et al. 2024]; however, the complexity of the robotic system is substantially increased, requiring three separate high degree of freedom robotic systems to perform assembly (a voxel carrying robot, a voxel installing robot, and a voxel fastening robot internal to the lattice), while the size scale of the lattice limits shape fidelity (300mm pitch).

In both of these voxel assembly systems, as the structure grows, the assembly throughput decreases, as the robot must traverse the built structure back and forth to pick up and install new material. [Abdel-Rahman et al. 2022] proposes that recursive hierarchical systems can help improve this issue— if assembly robots can assemble more assembly robots, then the swarm can increase its own parallelization, and if assembly robots can manipulate larger quantities of material at once, then they can build more efficiently. [Smith et al. 2024] demonstrates a first version of the hardware for a load-bearing modular voxel assembler, but is limited to desktop-scale objects.

2.2.4 Synthesis. Based on the prior art, we can distill some guidelines for an ideal robotic assembly-based digital fabrication system. Collective robotic approaches that have demonstrated the most promise towards scaling to practical applications typically strike a careful balance between the complexity of the material system and robotic system. The material system needs to be kept simple enough to be manufactured, while sophisticated enough to account for limited robotic functionality. The robot, in turn, needs enough features and degrees of freedom to reliably traverse and manipulate the lattice, but not so many that it becomes untenable to increase the amount of robots used. An ideal connection system only requires access from a single direction, without adding significant installation time. To maintain effective assembly times for larger structures, as well as to improve reliability, the system, as much as possible, should be able to parallelize its function and offer hierarchical assembly.

3 System Overview

We present a system for robotically assembling hierarchical discrete lattices. While prior voxel assembly systems have focused on a cuboctahedron lattice (equivalent to a node-connected octet lattice), we instead use an edge connected octet lattice because it inherently creates stable alignment features within each unit cell of the lattice (see Fig.2 for reference). By pre-connecting these unit cells laterally, we create compounded lattice/voxel blocks that can then tile to form stable 1D, 2D, or 3D structures (see Fig.3 for example tilings). These compounded blocks are thus all built from the same 1×1×1 unit voxel, but may vary in size and orientation (e.g., 2×2, 2×3, or 2×4), allowing a structure to mix different voxel types to optimize for geometry, performance, or assembly speed. The basic geometry of the lattice block is designed to self-align and constrain all but one degree of freedom when placed. In this work, the final degree of freedom is constrained with a releasable snap-fit (see Fig.6 for further detail).

The self-aligning compounded voxel block offers a few key advantages for robotic fabrication: 1) the alignment features permit placement error on the order of 1/2 the lattice pitch; 2) the installation of new voxels is purely vertical; and 3) the snap fit connection does not require physical access to the connection plane to engage, but is still reversible. Together, these features substantially reduce the requirements on the robotic system. To successfully build up a structure, the robots only need to traverse over the existing lattice, carry new voxels, and coarsely place them. To this end, we developed inchworm-style robotic arms that can crawl over the structure, extending on the paradigm used in e.g. [Jenett and Cheung 2017], [Park et al. 2023], or [Abdel-Rahman et al. 2022].

However, the use of compounded blocks introduces new challenges to the design and path planning of these structures. To this end, we developed a new pipeline for intaking standard 3D meshes, voxelizing them, grouping the voxels into compounded blocks, and then simulating their robotic assembly. The simulation space is integrated with the hardware, so that it additionally acts as the central control/coordination for assembly. This workflow is shown in Fig.1. The software system is further discussed in Section 4, while the hardware system is discussed in Section 5, with assembly examples and system evaluation in Section 6.

4 Software Implementation

A custom Web-based software environment was developed in JavaScript, enabling full control and simulation of the robotic assembly pipeline on any internet-connected device, without requiring local installation. Leveraging the Three.js visualization library, the system offers real-time 3D interaction through a synchronized digital twin of the robot. The pipeline begins with voxelization of a 3D mesh and proceeds through build sequencing, path planning, and real-time visualization with robot feedback.

4.1 Voxelization

The first step in the pipeline is the voxelization of the input geometry, any 3D mesh in STL format. The mesh is discretized using the size of the unit voxel, resulting in a resolution of 65mm. Once voxelized, we analyze the grid to identify repeating patterns composed of multiple adjacent voxels. These patterns can vary in size and orientation and may be composed of simple or stacked configurations. Some example patterns are detailed in Figure 2. To ensure connectivity and constructibility, the search prioritizes larger patterns first (e.g. 4x2x2 blocks) and proceeds hierarchically down to smaller units, filling in gaps as needed. Each detected pattern must be locally connected to at least one neighbor, either directly or via a base layer.

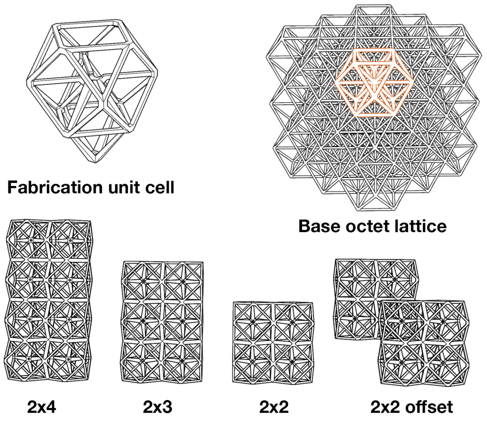

Figure 2: An overview of the lattice type and building blocks used in this project. The basic lattice type is an edge-connected octet lattice, which is decomposed into an extended cuboctahedron-octet, which is then compounded into different arrangements for robotic assembly.

4.2 Building Sequence

Given the location and type of each voxel in the structure, the next step is to define a build order that ensures feasibility while minimizing future path planning complexity. The building sequence determines the order in which each voxel of the structure is assembled and is influenced primarily by two parameters: the number of robots and the location of their associated voxel Feed. We assume each robot is assigned a unique voxel Feeder from which it collects and places voxels. The structure is first partitioned by assigning each voxel to the closest source based on the Manhattan distance. In case of equidistant voxels (a tie), a simple alternating policy ensures balanced distribution among Feeds.

Once assigned, the construction process is parallelized between robots. For each robot, a building sequence is computed independently. The structure is decomposed into horizontal layers, starting from the base and progressing upward. Within each layer, voxels are placed outward from the feed point, ensuring that the built region grows in a connected and stable manner.

This strategy guarantees that every newly placed voxel rests on an existing one from the layer below, thereby creating a new accessible surface on the current layer. As a result, the robot can step onto this newly created platform to continue placing subsequent

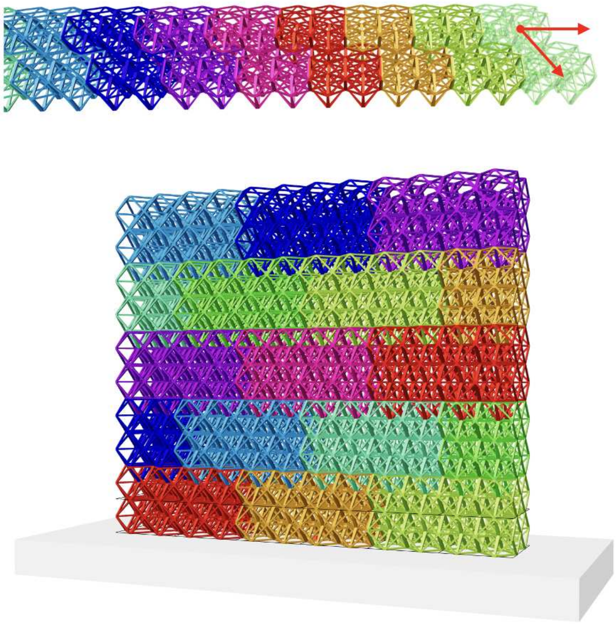

Figure 3: (Top) An example of tiling 2x2 offset voxel blocks to establish a first layer or overhang, with red arrows indicating the axes of potential extension. (Bottom) Beyond the first layer, layers can achieve interconnection through staggering layers.

voxels. The building sequence algorithm enforces this outward growth, ensuring that at every stage, the next voxel to be placed is both structurally supported and physically reachable by the robot.

4.3 Path Planning

Once the build sequence is established, each robot must autonomously navigate from its associated Feed location to its assigned placement position, taking into account the current state of the partially build structure. To generate feasible and collision-free paths, we employ an A* algorithm adapted to the discrete 3D voxel space, which minimizes the function:

$$h(n) = f(n) + g(n) \tag{1}$$

Where g(n) is the cost of moving from the start to voxel n and h(n) is a heuristic estimating the cost to reach the goal from n. We use the Manhattan distance as the heuristic, which performs efficiently in the voxel grid while preserving path optimality in our constrained setup.

The A* algorithm is particularly powerful in a voxelized environment, as the search space is inherently discrete, the connectivity is regular, and the heuristic directly reflects the grid geometry. By expanding nodes along the most promising directions first, it efficiently balances exploration and optimality. As a result, the algorithm consistently returns near-optimal paths with limited computation, even as the structure grows. Its effectiveness has already been validated in earlier iterations of this project [Abdel-Rahman et al. 2022; Smith et al. 2024].

4.4 Robotic Assembly Simulation

4.4.1 Digital Twin for Live Simulation: To monitor and validate the robotic assembly process in real time, we developed a digital twin environment that mirrors the simulated robot's actions on the physical hardware. Path planning outputs are executed in the simulation and simultaneously relayed to the robot via a Python middleware layer, using WebSocket communication over Wi-Fi. This architecture offers several key benefits:

(1) Allows real-time updates during construction, enabling small-scale corrections in position or trajectory.
(2) Provides live mapping of the robot's state and the evolving structure.
(3) Offloads computation entirely to the simulation side, reducing the computational resources needed on the robot side.

This tool is used within the pipeline to align the simulation with the hardware using the data flow presented in 4.4.2, with all elements generated directly from the target structure. Nevertheless, it can also be used as a standalone application, allowing the user to manually place every element of the simulation (voxelized environment, robots, feeds, voxels to build) and subsequently link the configuration to the hardware. An example is presented in Figure 5.

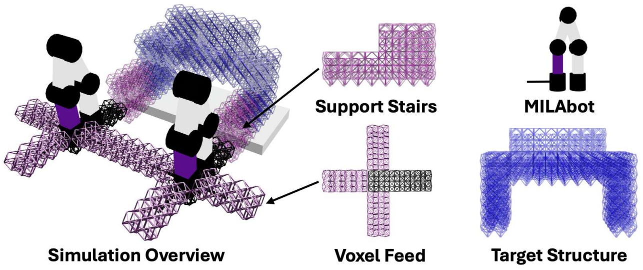

Figure 4: Simulation overview illustrating the four main elements: MILAbot, support stairs, voxel feed, and target structure.

4.4.2 Data Flow and Feedback: The system operates using a feedforward architecture: voxelization, build sequencing, and path planning are executed sequentially within the simulation environment. The resulting movement instructions are transmitted from the digital twin to the physcial robot— the Modular Inchworm Lattice Assembler robot (MILAbot)— via a Python bridge, which converts high-level actions into low-level joint target values using WebSocket communication. Simultaneously, the robot streams back its current joint states (see Figure 5). If a deviation from the expected pose is detected, a realignment sequence is triggered. This involves returning to the last valid position and re-approaching the target slowly to ensure accurate alignment without requiring full re-planning.

5 Hardware Implementation

In this section, we discuss the hardware implementation of the voxel assembly system, covering the voxel design, manufacturing, and performance, as well as the robotic system design and operation.

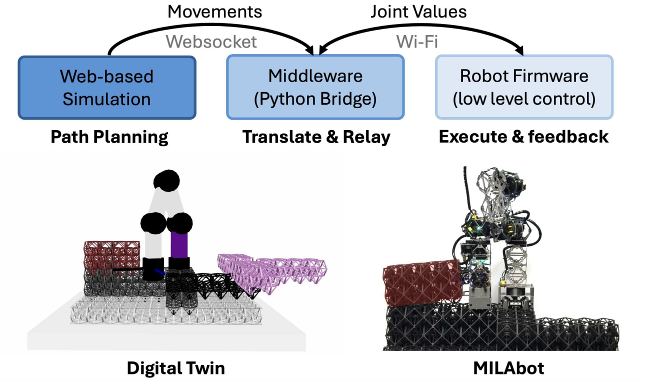

Figure 5: Top: Data flow from web-based simulation to MILAbot through middleware. Bottom: Digital twin synchronized with physical execution.

5.1 Material System

Our material system uses self-aligning compounded octet lattice blocks to create interlocking structures. In this work, we produce these lattice blocks in polylactic acid (PLA) through FFF 3D printing. We use FFF 3D printing because of its ability to easily handle complex geometries at a low cost and a reasonable speed. However, we envision that for future versions of this system, the compounded blocks might be further broken down into separate voxels, such as in [Gregg et al. 2018], or individual faces, such as in [Jenett et al. 2020], [Smith et al. 2024], or [Jenett et al. 2018], so that they can be mass-manufactured from higher performance materials, such as GFRP in [Jenett et al. 2020], CFRP in [Gregg et al. 2024] or aluminum in [Smith et al. 2025].

To improve the printability of the voxels, we add flanges to support the large overhanging regions during printing, as done in [Leamon 2025]. The flanges enable clean print quality without the use of supports, enabling more efficient material usage as well as improved absolute mechanical properties of the voxel block. However, the flanges are not intrinsically necessary, and it possible to print the lattice in its unmodified form. We printed the compounded voxels in different overall sizes depending on printer availabilty and print bed-size, and if necessary, later laterally pre-assemble them with printed snap clips. A representative 4x2x2 compounded block is shown in Fig.6.

Voxel blocks are vertically connected using a screw-released snap fit connector. The snap fit consists of three parts: a set of pincers that by default sit too wide to engage with a voxel below it, a screw (in this case an M4 8 mm long socket head screw) to fasten it directly into the base of a voxel, and a screw hat, which acts as a spacer to push the legs of snap pincers out, so that they can engage with the voxels below (these are shown in Fig.6). The snap fit connectors are installed into the lowest octets of the printed voxels. The act of fastening the M4 screw then forces the pincers into the correct state, such that the voxel block can snap into one below it. To uninstall the connector, the M4 screw can be removed, which released the snap fit, allowing the block to removed. Though this is currently done manually, future versions of the robotic system could include an end effector for unscrewing the connectors.

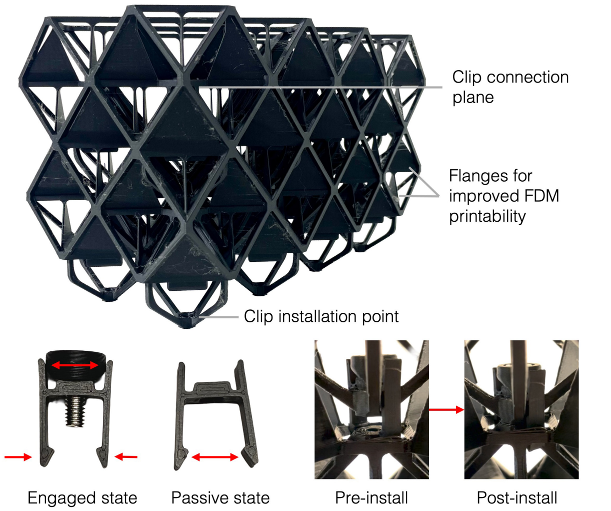

Figure 6: (Top) A 4x2x2 block of FFF printed PLA octet lattice with printability features added. (Bottom) Screw-release snap fit used for vertical connections.

So that the voxels sit on a flat surface, we additionally produced base-layer voxel blocks that consist of the upper half of the standard block. These are added onto the protruding octets to create a flat base layer without adding extra height to the system.

5.2 Assembly Robots

We introduce the Modular Inchworm Lattice Assembler robot, or MILAbot. As in prior voxel assembly robot systems, such as [Smith et al. 2024], [Park et al. 2023], [Jenett et al. 2019], the primary objective of the MILAbot design is to enable scalable voxel assembly with a system that can locomote over and assemble load-bearing discrete lattice structures. Looking toward scaling to the assembly of larger structures, the MILAbot is designed to increase assembly throughput while minimizing overall system cost and complexity. Additionally, the MILAbot is designed with eventual self-assembly in mind, that is, a starter MILAbot should be able to assemble more MILAbots, to autonomously improve efficiency as described in [Abdel-Rahman et al. 2022] and [Smith 2023].

The MILAbot is made from four primary module types: actuated joints, passive spacers, gripper feet, and voxel carriers as shown in Fig.7. These modules are arranged in a configuration similar to a five degree of freedom robot arm, with grippers on both ends, and three voxel carriers on one side. Toward eventual robotic-self assembly, the modules are capped by incoming- and outgoing- PCBs on either side that use a heating circuit to reflow low-melt solder between modules, forming a simultaneous eletrical and mechanical connection, such as in [Smith et al. 2024], though this is not a focus of this work. The robot modules each have their own microcontroller, which are all networked over an I2C bus [NXP 2021], with a

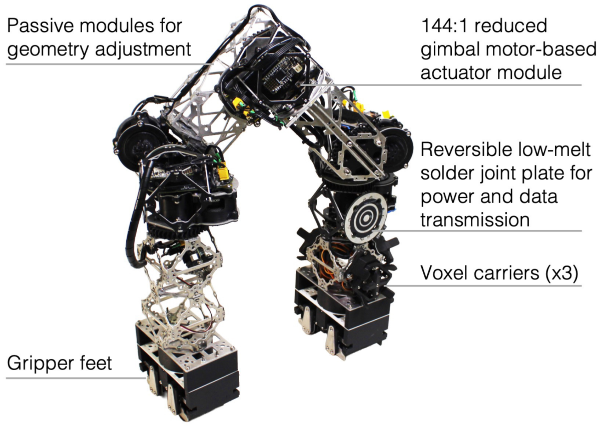
Passive modules for geometry adjustment
144:1 reduced gimbal motor-based actuator module
Reversible low-melt solder joint plate for power and data transmission
Voxel carriers (x3)
Gripper feet

Figure 7: The Modular Inchworm Lattice Assembler robot (MILAbot) consists of four primary module types: actuated joints, passive spacers, gripper feet, and voxel carriers.

WiFi-enabled microcontroller acting as the primary microcontroller on the bus. We typically run the robot at 14V, but the electronics are designed to handle up to 24V.

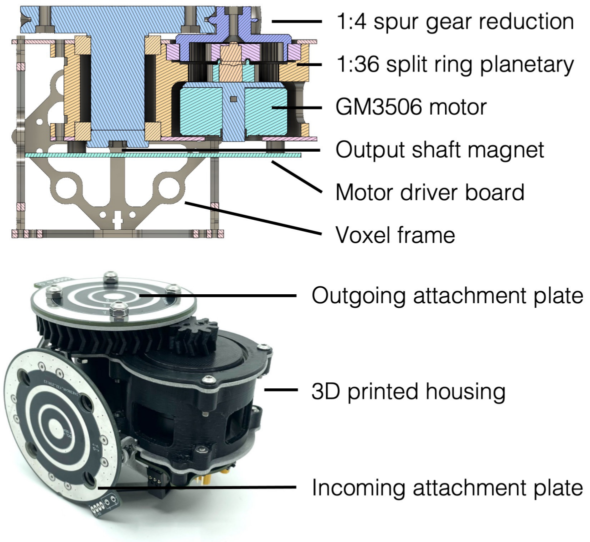

Figure 8: (Top) A cutaway view of the motor module design, which is based on a BLDC motor with two stages of reduction, feedback on the output shaft, and a voxel frame to enable module manipulation by another robot. (Bottom) A complete motor module, with incoming and outgoing PCB attachment plates.

The actuator modules are built around a GM3506 gimbal motor, a small brushless DC motor. We use a two stage 3D printed reduction consisting of a 1:36 split ring planetary followed by a 1:4 spur gear reduction, which has the added benefit of moving the output shaft off from the motor shaft, allowing us to more easily track the position of the output shaft. The total reduction is then 1:144. The module is fabricated from a mix of 3D printed polycarbonate, for its temperature resistance, and aluminum plates. The module controller board uses an Adafruit M4 Express (SAMD51) as the microncontroller, a DRV8316 for motor control, and two AS5407 magnetic encoders for motor and output shaft feedback. Motor control is done using SimpleFOC [Skuric et al. 2022]. Each module weighs approximately 400g and can output approximately 10-15 Nm before failure.

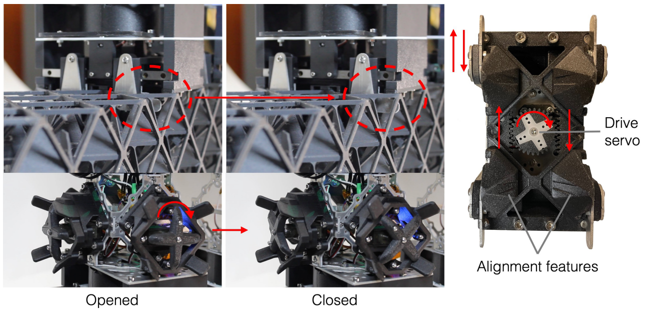

Figure 9: The operation of both gripper types in the MILAbot. (Top) The structure gripper, (bottom) the voxel payload gripper(s).

The MILAbot uses two voxel gripper types. The first type adheres the robot to the structure, and the second type carries a voxel payload. The operation of both types is shown in Fig.9. The gripper feet are designed to step in the middle of a 2x2x1 grid of voxels, which introduces some constraints on where the robot can locomote. Its relatively larger size affords the robot greater stability and helps to distribute the load of the robot's locomotion over a larger area on the voxels. Each gripper foot is actuated using a FeeTech FS117 hobby servo, which retracts and extends a set of aluminum gripping features underneath the upper corner nodes of some voxels.

Because the voxel carrier grippers do not need to support a significant load, they are designed to be smaller and lighter, using Miuzei MG90s micro servos to open and close a petal feature, modeled after the grippers used in [Jenett et al. 2019] and [Abdel-Rahman et al. 2022]. To maximize the carrying capacity of the robot without significantly impacting its maneuverability, we use three per robot installed on one leg (though the robot could physically support installing an additional three voxel carriers on the other leg, this would cause significant collision issues for any non-flat structure).

Both grippers are designed with large alignment features. In the case of the gripper feet, these match the alignment features of the voxel blocks themselves, while the voxel carriers use a smaller petal shape. The combination of these mechanical alignment features in the robot and in the underlying structure means that we are able to achieve relatively precise robot and voxel placement without

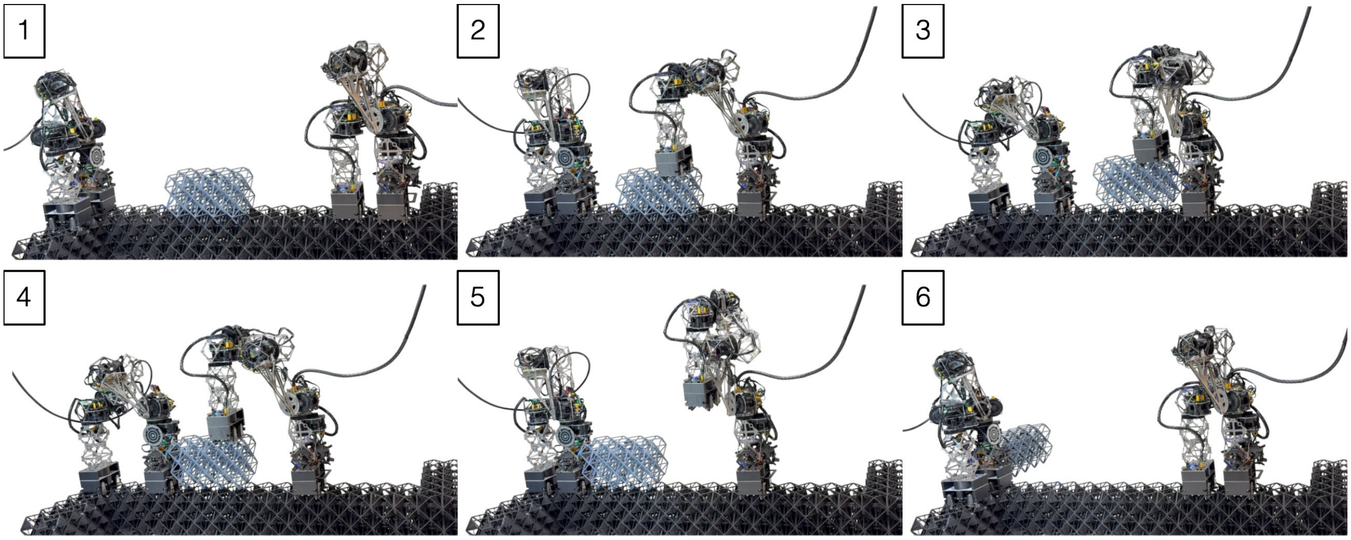

Figure 10: Picking up and passing a voxel block from one robot to another.

high-performance robots (i.e. the robots have minimal feedback and a lot of compliance).

An example of this is shown in Fig.10, where two robots pass a voxel block to each other. The two robots approach the target voxel (gray), and one steps onto it and lifts to remove it (the voxel is not constrained with snaps in this case). The other robot then steps forward, re-indexing its position on the lattice. Because the voxel carriers are placed on the robot at a multiple of the lattice pitch, this places the second robot at a "known" location for the robot carrying the other voxel, which only needs to then move into that position, with the alignment petals proving fine alignment. The second robot then grips the voxel while the first lets go and retracts.

The robot relies on the ability of the voxels to correct large placement errors to install new blocks simply by dropping them into place and then stepping on them to engage the snap fits. This process is shown in Fig. 11, where a robot carrying two voxels steps into position, swings the back leg forward and rotates it to place the correct voxel in the appropriate pre-placement position, drops the voxel, returns the back leg backwards, steps forward once, and then stomps on the voxel to engage it, forming a sturdy connection.

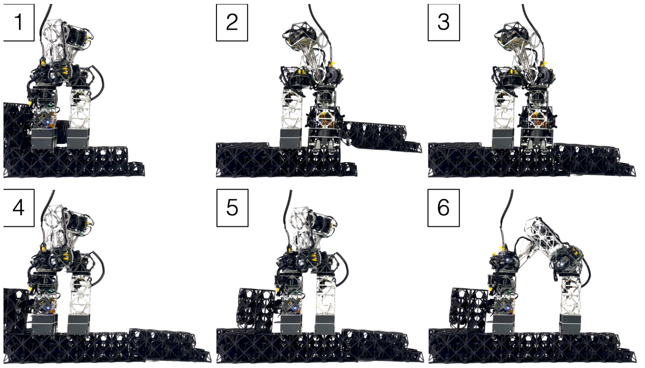

Figure 11: An example voxel installation sequence. 1) The robot is in pre-placement position. 2) The robot steps into the placement position and 3) drops the voxel roughly into place, before 4) retracting, and then 5) taking a step forward. 6) The robot stomps the new voxel to engage the snap fit connection.

6 Evaluation

In this section, we evaluate the performance of the presented system through a set of assembly demonstrations, the throughput and fidelity of the mechanical system, the mechanical efficiency of the lattice type, and general robotic performance metrics.

6.1 Assembly Demonstrations

We use the complete workflow to demonstrate robotic assembly of basic structures using one and two robots.

6.1.1 Single Robot Assembly. First, we use a single robot to assemble a 4x4x4 voxel block. We voxelize the structure into four 4x2x2 blocks, the largest compounded size our system currently accommodates. This way, the robot only needs make four placements, in contrast to prior approaches to voxel assembly, which would have required 64 voxel placements for the equivalent structure. A condensed version of the assembly sequence is shown in Fig. 12. The robot carries two blocks at once, as the minimum amount of trips it can make for this structure is two (either two voxels twice, or three voxels and one voxel once). The first two voxels are placed oriented toward the camera plane, while the second two are placed parallel to it, resulting in a fully connected structure.

Next, each robot assembles its support staircase for the bench example, as previously shown in Fig.1 and Fig.4. Freeze frames from the build sequence are shown in Fig.13. The build process for the stair case is similar to that of the cube, but at the end, we

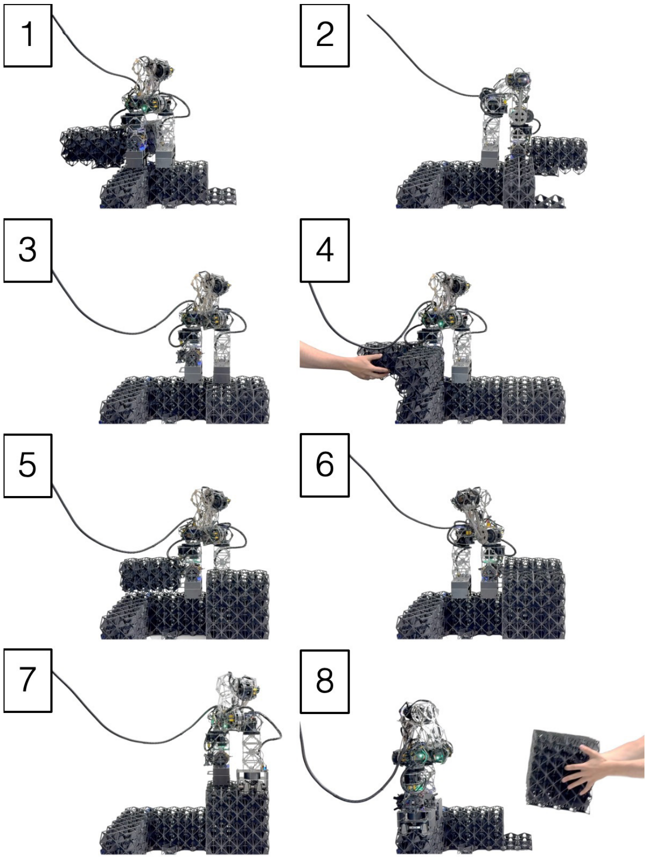

Figure 12: Assembly of a 4x4x4 voxel block using a set of four compounded blocks, carried two at a time.

demonstrate a small overhang with this structure that is able to support the weight of the assembler robot on it. Overhangs, or first layers, both require staggered voxel blocks to achieve. This can be done either by layer shifting a 2-voxel tall stack, or by attaching base-plate voxels at a stagger, as done here. The offset provides alignment features for the next block placement and braces the newly installed voxel against the existing structure, especially during the drop-to-place procedure.

6.1.2 Double Robot Assembly. Because each robot has its own voxel feed in our workflow, most multi-robot structure assembly is identical to single robot assembly, up until the point at which the structures meet. The joining of separate build fronts, or unlinked structures, is not something considered by any of the prior voxel assembly systems, but is likely a critical feature for scaling these systems up.

In this example, the two robots finish the assembly of the bench by filling in the base of the seat (see Fig.14 for reference). One robot drops a voxel block into the empty space, and then steps on it to align it and the structure. The second robot then installs its voxel into the remaining space. This is possible because the alignment features on the voxels are permissive of a fairly large amount of error, the error is larger than the voxels, and the structure is able to slightly slide and deform. We cannot always guarantee that this will be the case, so for joining build fronts, it is likely necessary to consider alternative voxel types, such as ones that are slightly undersized or compliant, to account for the potential misfit. However, even though this is not a reliable method for joining a voxel seam, this block placement is taken to be infeasible in the prior planning systems for voxel assembly [Gregg et al. 2024], [Smith et al. 2024], [Jenett et al. 2019].

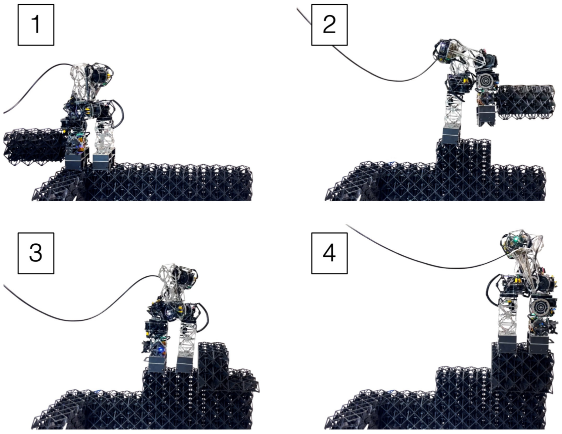

Figure 13: A single robot assembling a support staircase for the bench build.

After installing the base of the bench, the two robots install the back of the seat. These two voxels are necessarily installed simultaneously, as to avoid collisions in the placement. Though for a small structure with only two robots, keeping synchronicity (or restoring it if lost) is not difficult, as the system scales to larger robot counts, it may be necessary to consider additional methods for synchronizing the movements of all robots at critical junctures, without needing to run the entire system at the pace of the slowest robot. Or, the methodology for voxel placement could be revisited, to consider more sophisticated obstacle avoidance, depending on the build state of the structure.

Once completed, the bench is removed from the support staircase, where the path planning has left a seam. The completed bench is then directly usable, as the voxels have decent mechanical performance, which is further discussed in Section 6.3. We qualitatively evaluated the bench by having two people sit on it, as shown in Fig.15. The bench supports this weight, though the overall design is a bit small for comfort— luckily, the voxels support disassembly and reassembly, so we can change the design.

6.2 Throughput and Fidelity

As shown in Figure 16, we assess the performance of different voxelization strategies by analyzing the trade-off between the number of voxels that make up the structure and geometric precision across various mesh types. The geometric precision can be defined as:

$$precision = n_{voxel} * V_{voxel}/V_{mesh} \tag{2}$$

Where n_{voxel} is the number of unit voxel with V_{voxel} being the volume of one unit voxel and V_{mesh} the volume of the input mesh.

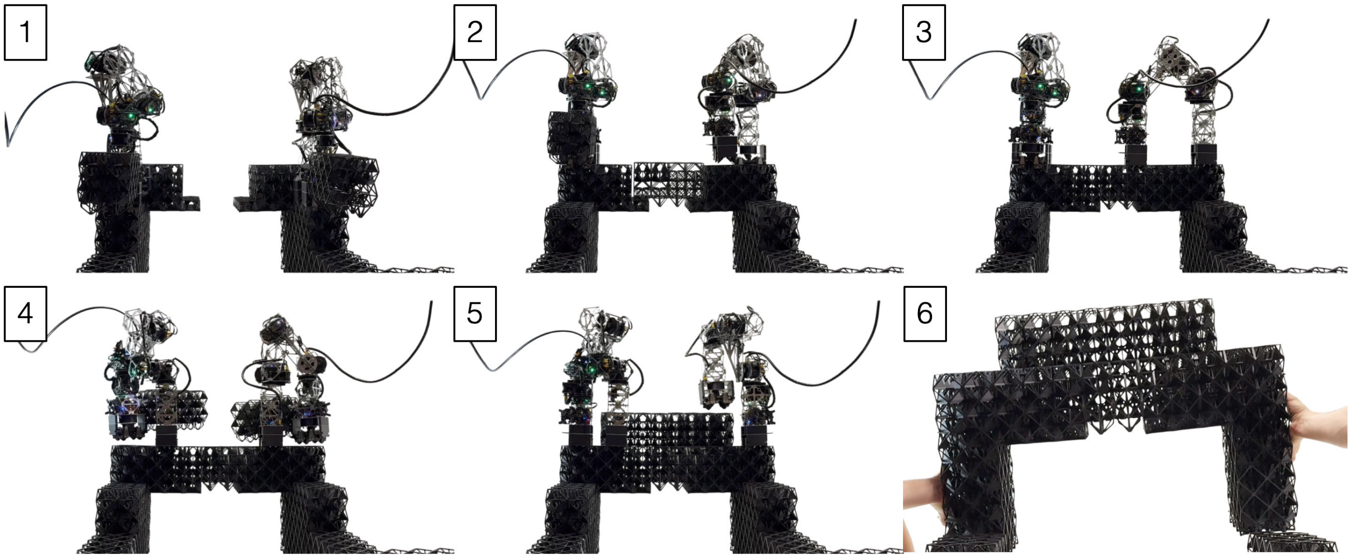

Figure 14: Two robots assembling a bench together.

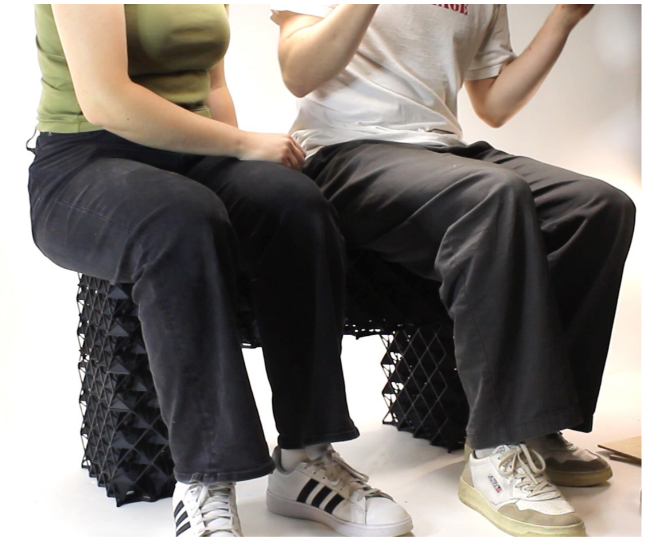

Figure 15: The assembled bench supports two people.

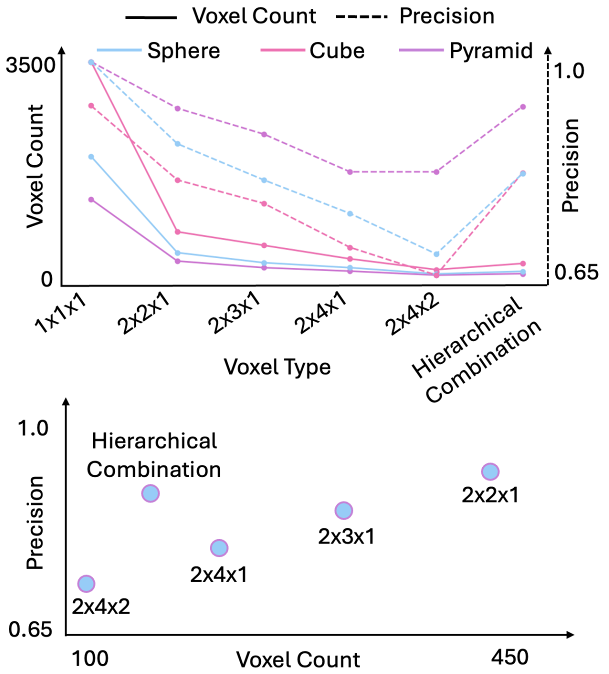

Figure 16: Evaluating voxel pattern efficiency— hierarchical combination outperforms any single voxel type by breaking the linear trade-off between voxel size and precision. (Top: Voxel count and precision decrease linearly with increasing voxel size when voxelizing meter-scale meshes. Bottom: Precision as a function of voxel count, highlighting the benefits of the hierarchical combination.)

The left panel shows how larger voxel patterns drastically reduce the number of elements required but come with a loss in precision. The right panel highlights this trade-off by plotting precision as a function of voxel count. Patterns like 2x3x1 and 2x2x1 achieve a good balance, while the hierarchical combination offers a compelling middle ground with high precision and low voxel count.

Figure 17 highlights two key components to accelerate construction: increasing the number of robots and improving the voxel carrying strategy. On the top row, simulation snapshots show how multiple robots parallelize the construction of a 16x16x16 cube, reducing total build time. While the benefit is clear, the speed-up

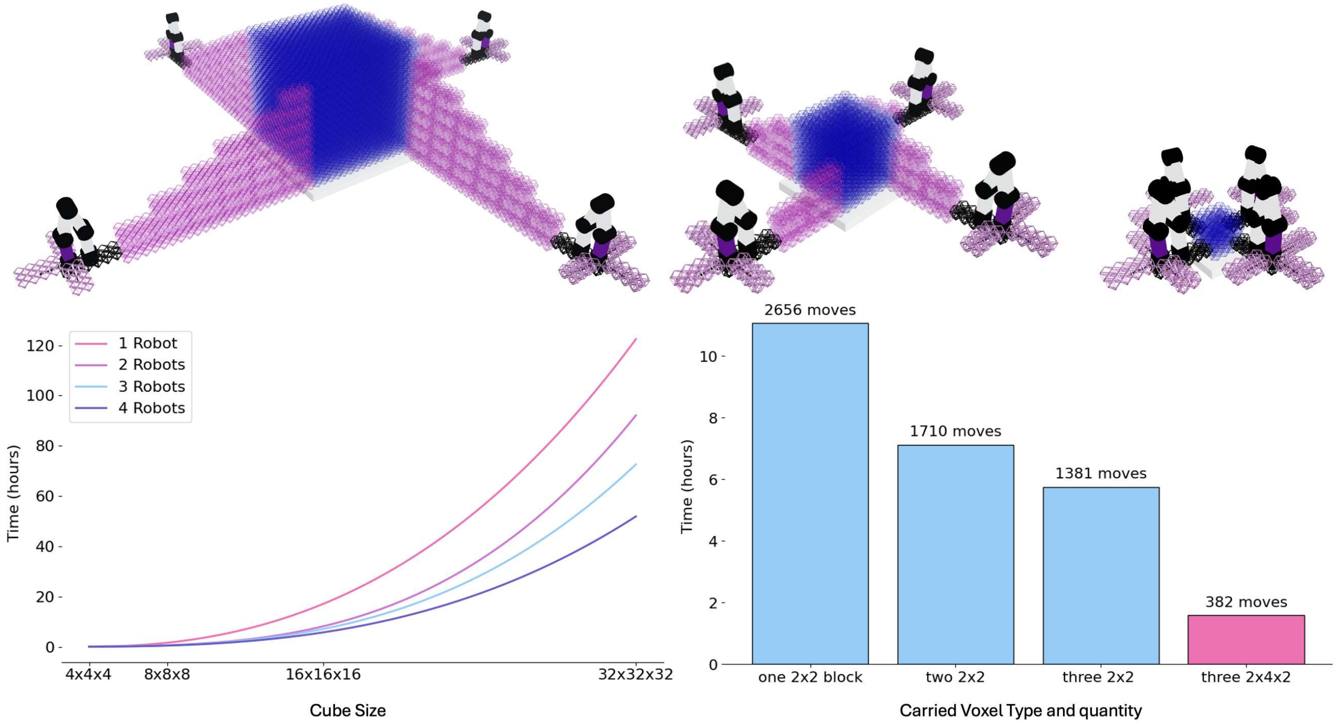

Figure 17: Impact of robot quantity and voxel carrying strategy on build time. (Top) Simulation snapshots of cubes of decreasing size (16x16x16, 8x8x8, and 4x4x4 voxels). (Bottom Left) Estimated build time as a function of cube size and number of robots. (Bottom Right) Time required to build the same structure depending on the number or size of voxels carried per trip. These graphs motivate the use of a larger robot fleet for assembling large structures, as they show a substantial decrease in assembly time. They also highlight the importance of maximizing robot payload, with each robot ideally carrying three 4×4×2 voxels rather than operating below full capacity.

is not exactly proportional to the robot quantity due to each robot building its own support stairs independently, without sharing intermediate infrastructure.

The graph on the bottom left shows that construction time increases exponentially with structure size, but adding robots significantly flattens the curve. On the bottom right, we analyze the impact of voxel carrying capacity. Carrying multiple small voxels per trip already provides notable gains, but the greatest efficiency comes from transporting larger compound blocks (e.g., 4x2x2). This confirms that both scaling the number of robots and optimizing carried voxel configuration are critical for improving overall build throughput.

6.3 Hardware Performance

Because other robotically assembled discrete lattices have been demonstrated, we can compare the performance of our system against those in literature. First, we consider the mechanical performance of the modified octet lattice we use. In evaluating this performance, we consider the version of the lattice presented here to be a proof-of-concept demonstration, and the comparisons below to indicate, generally if the design decisions we have made are in desirable directions. Because of the anisotropy in FFF printed parts, the robustness of our voxel system can be low, and we do not perform a full mechanical characterization of the voxels under tensile and bending loads. Further, the overall interplay between vertically interconnected blocks under various loads is also still unknown, and beyond the scope of the current work.

To determine the behavior of the lattice under compressive loads, we tested a single unit cell, a 2x2x2 lattice block, and a 3x3x3 lattice block on an Instron 4411 with a 5kN load cell for the single unit at the 2x2x2, and on an Instron 5985 with a 250kN load cell for the 3x3x3 at a compression rate of 10 mm/min. The results of this testing are summarized in Table 1. The lattices demonstrate good strength and stiffness at a reasonable weight (the average density of the lattice is 81.85 kg/m^3). Under compression, the lattices can support very high loads— the 2x2x2 block reached a maximum load of 3445N, which is about 2,220x its own weight, while the 3x3x3 block reached a maximum load of 8712N, which is about the load exerted by the weight of an average cow.

We can determine the mechanical efficiency of the lattice by looking at how its compressive modulus and density scale relative to the bulk material it is made from. The relative compressive modulus is given by E/E_s where E is the compressive modulus of the lattice and E_s is the compressive modulus of the bulk material. Similarly, relative density is given as ρ/ρ_s, where ρ is the density of the lattice and ρ_s is the density of the bulk material. Ideal stretch dominated behavior, that is, the most efficient scaling of modulus with density, is given by linear scaling E/ρ. However, this is generally considered inaccessible [Schaedler and Carter 2016]. Instead, ideal bending dominated behavior is given by quadratic scaling $E^{1/2}/\rho$, while below $E^{1/3}/\rho$ is accessible via foams. So, the target for the structural efficiency of an architected lattice is to surpass the quadratic scaling for its bulk material.

We plot a comparison of the relative compressive modulus and relative density for this work, as well as the other discrete lattice assembly systems: [Gregg et al. 2024], [Smith et al. 2024], and [Jenett et al. 2019]. Both this work and [Gregg et al. 2024] achieve stretch dominated behavior, indicating efficient and desirable material usage. Note that [Jenett et al. 2019] does not provide mechanical testing data; this is instead estimated using an ideal beam simulation of the presented lattice using the reported parameters and materials. Additionally [Jenett et al. 2019] uses a non-structural magnetic joint, but for this comparison we disregard that.

We can additionally compare the assembly throughput and system cost for these systems. We use the reported volumetric assembly throughput for each system, with the exception of [Gregg et al. 2024], in which their reported throughput is for a hundred voxel structure, and so is an underestimate of the speed of the system as compared to the systems in [Jenett et al. 2019] and [Smith et al. 2024] which only demonstrate assembly of tens of voxels. Instead, we draw an average from the installation time per voxel for the first 15 voxels in their structure, to better align with the data we have here, resulting in a much higher throughput estimate than what they report.

We find that our volumetric assembly throughput is 4,394,000 mm^3/min for a single robot carrying two blocks of 4x2x2 voxels, as compared to 2,700,000 mm^3/min for [Gregg et al. 2024], 751,879 mm^3/min for [Jenett et al. 2019], and 343,281 mm^3/min for [Smith et al. 2024]. Our higher volumetric throughput is owed to two main factors: we can carry multiple blocks of compounded voxels while moving at a comparable speed to [Jenett et al. 2019], and unlike [Gregg et al. 2024] the current implementation requires minimal installation/fastening time. This result indicates that we likely have budget to increase the time spent per voxel (such as by implementing a different connection system) while still staying competitive in volumetric throughput.

We additionally compare these results against the cost of each robotic system. For [Smith et al. 2024] and [Jenett et al. 2019] these values are taken from their reported bills of materials, while for [Gregg et al. 2024] we estimate the cost based on the components reported in [Park et al. 2023], where the current figure is based on doubling the cost of the actuation system for one inchworm robot, to account for the need for both a carrying robot and an installing robot, with an additional $300 added as a low-end estimate for the cost of the third robot, as well as all of non-actuation components in the inchworm robots. Interestingly, there is a relatively linear relationship between system cost and assembly throughput for the prior single-voxel based systems, which this work sidesteps via implementing hierarchical material handling in an otherwise very simple robot.

Table 1: Summary of mechanical properties for different voxel block sizes

	1x1x1	2x2x2	3x3x3
Stiffness [N/mm]	949	2278	4868
Maximum load [N]	602	3445	8712
Compressive Modulus [MPa]	14.6	17.5	24.9

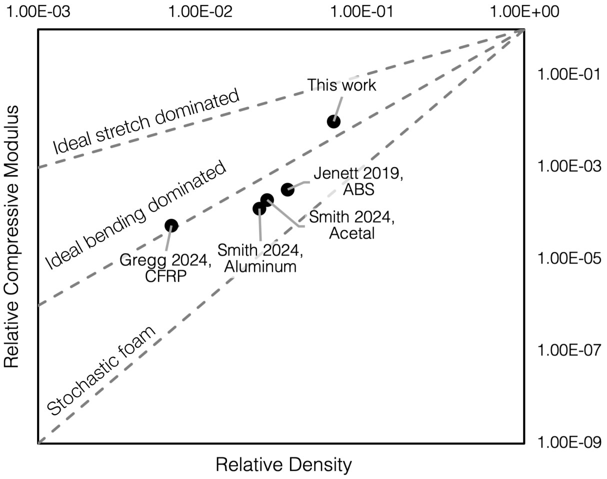

Figure 18: A comparison of the relative compressive modulus (voxel modulus divided by bulk material modulus) and the relative density (voxel density divided by bulk density) for robotically assembled discrete lattices. This work achieves stretch dominated behavior, demonstrating material efficiency relative prior robotically assembled voxel systems.

7 Future Work and Limitations

7.1 Voxelization

While our voxelization algorithm proved sufficient to demonstrate the feasibility of a complete robotic assembly pipeline across a defined set of structures, it remains limited in its generalizability and is sensitive to variations in input geometry. In future work, we plan to explore more robust and generalizable approaches. This includes developing enhanced rule-based algorithms inspired by an octree subdivision approach as developed in [Abdel-Rahman et al. 2022], which remain fully discrete and deterministic, as well as investigating learning-based methods, such as those proposed in [Pun et al. 2025], that leverage the power of large language model to Generate physically stable and buildable LEGO® structures. Incorporating a structure-aware strategy that accounts for physical constraints and evolving geometry could significantly improve the reliability and feasibility of voxel placement over time. And would extend the buildable structure to the one presented in Figure 20. Future work will also explore generating input geometries based on desired performance, such by integrating the system in [Kyaw

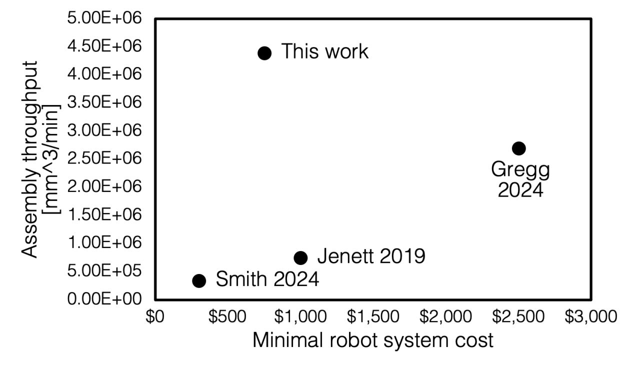

Figure 19: A comparison of the volumetric throughput for different voxel assembly systems for low voxel count structures, against the cost of the minimal robotic assembly unit. This work achieves the highest throughput at the second lowest cost.

et al. 2025], or verifying structural performance, such as through built in FEA, as in [Smith et al. 2025].

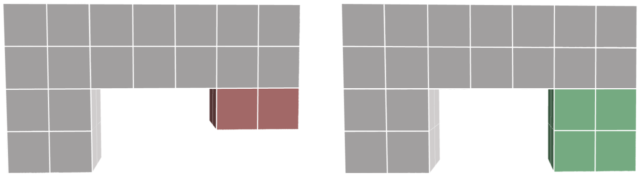

Figure 20: Comparison of infeasible (left) and feasible (right) structures due to local connectivity constraints.

7.2 Feeder Management

As discussed in Section 4.2, in the current setup, each robot is paired with a unique Feed location, allowing independent operation. This constraint simplifies coordination, but limits scalability and task flexibility. Future work will explore decoupling the number of robots from the number of sources, enabling more dynamic task allocation. Related work [Abdel-Rahman et al. 2022] has demonstrated that shared and flexible sourcing strategies can significantly improve resource utilization and system throughput.

Additionally, in the current system, we manually feed voxels to the robot. In a fully automated set up, this would be accomplished using e.g. conveyor belt to deliver voxels from their manufactured or stored location to the robots.

7.3 Robot Control

As the system scales to multiple robots, hardware variability is expected to introduce differences in kinematics and precision. To ensure consistent behavior across platforms, we plan to develop a more adaptable control system with integrated per-robot calibration. This includes implementing a tunable inverse kinematics (IK) model, allowing each robot to compensate for mechanical deviations and maintain precise control during assembly.

Additionally, to fully realize voxel reconfiguration, the robots will need to eventually be able to disassemble voxel structures, which will require both additions to the robotic hardware as well as the path-planning tool. Eventually, we hope to be able to fully support the arbitrary disassembly and reconfiguration of a voxel structure using this system.

7.4 Voxel Material

We use FFF printed voxel blocks in this work primarily because they are readily accessible and offer high performance along some loading directions. However, the anisotropies inherent in FFF plastic printing likely preclude its use for many practical applications, so future work will focus on developing alternate methods for prebuilding the voxel blocks, such as through injection molded faces assembled together [Jenett et al. 2020] or formed metallic facets [Rubio et al. 2023]. Additionally, [Smith et al. 2025], [Cameron et al. 2022], and [Wang et al. 2023] present electrically active voxel systems, which could be integrated with this system in the future. Additionally, the system is likely mechanically anisotropic, and in future work we plan to develop additional tools simulated structure behavior, as well as more fully characterize the material system under bending and tensile loads.

8 Conclusion

This work demonstrates a novel approach to scalable digital fabrication through the integration of architected lattice blocks, modular mobile robots, and a live digital twin system for coordinated assembly. By leveraging the strengths of small-scale digital fabrication to produce complex interlocking components, and combining them with simple, mobile robots capable of relative traversal and block placement, we enable the construction of meter-scale structures with reduced system complexity and cost. The hierarchical voxelization strategy, modular assembly robot design, and real-time simulation interface collectively contribute to a flexible and scalable assembly pipeline. Our validation of the system through the successful fabrication of meter-scale structures highlights its potential for broader application, such as for architecture, infrastructure, or in-space assembly and manufacturing. This work establishes groundwork for further exploring multi-robot collaboration in hardware, the interplay between structural robotic systems, and, eventually, hierarchical and recursive robotic assembly.

Acknowledgments

This work was supported by MIT Center for Bits and Atoms Consortia funding.

References

Amira Abdel-Rahman, Christopher Cameron, Benjamin Jenett, Miana Smith, and Neil Gershenfeld. 2022. Self-replicating hierarchical modular robotic swarms. *Communications Engineering* 1, 1 (November 2022), 35. https://doi.org/10.1038/s44172-022-00034-3

Muhammad Abdullah, Laurenz Seidel, Ben Wernicke, Mehdi Gouasmi, Anton Hackl, Thomas Kern, Conrad Lempert, Clara Lempert, David Bizer, Wieland Storch, Chiao Fang, and Patrick Baudisch. 2024. PopCore: Personal Fabrication of 3D Foamcore Models for Professional High-Quality Applications in Design and Architecture. In *Proceedings of the 9th ACM Symposium on Computational Fabrication* (Aarhus, Denmark) *(SCF '24)*. Association for Computing Machinery, New York, NY, USA, Article 6, 14 pages. https://doi.org/10.1145/3639473.3665787

Muhammad Abdullah, Romeo Sommerfeld, Bjarne Sievers, Leonard Geier, Jonas Noack, Marcus Ding, Christoph Thieme, Laurenz Seidel, Lukas Fritzsche, Erik Langenhan, Oliver Adameck, Moritz Dzingel, Thomas Kern, Martin Taraz, Conrad Lempert, Shohei Katakura, Hany Mohsen Elhassany, Thijs Roumen, and Patrick Baudisch. 2022. HingeCore: Laser-Cut Foamcore for Fast Assembly. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 10, 13 pages. https://doi.org/10.1145/3526113.3545618

Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofen, Hsiang-Ting Chen, Stefanie Mueller, and Patrick Baudisch. 2015. Protopiper: Physically Sketching Room-Sized Objects at Actual Scale. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 427–436. https://doi.org/10.1145/2807442.2807505

Aleksandra Anna Apolinarska, Michael Knauss, Fabio Gramazio, and Matthias Kohler. 2016. The Sequential Roof. In *Advancing Wood Architecture* (1st edition ed.). Routledge, 14. eBook.

Aleksandra Anna Apolinarska, Matteo Pacher, Hui Li, Nicholas Cote, Rafael Pastrana, Fabio Gramazio, and Matthias Kohler. 2021. Robotic assembly of timber joints using reinforcement learning. *Automation in Construction* 125 (2021), 103569. https://doi.org/10.1016/j.autcon.2021.103569

Mustafa Batikha, Rahul Jotangia, Mohamad Yasser Baaj, and Ibrahim Mousleh. 2022. 3D concrete printing for sustainable and economical construction: A comparative study. *Automation in Construction* 134 (February 2022), 104087. https://doi.org/10.1016/j.autcon.2021.104087

Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen. 2019. Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300796

Harsh G. Bhundiya and Zachary C. Cordero. 2023. Bend-Forming: A CNC deformation process for fabricating 3D wireframe structures. *Additive Manufacturing Letters* 6 (2023), 100146. https://doi.org/10.1016/j.addlet.2023.100146

Serban Bodea, Pascal Mindermann, Götz T. Gresser, and Achim Menges. 2022. Additive Manufacturing of Large Coreless Filament Wound Composite Elements for Building Construction. *3D Printing and Additive Manufacturing* 9, 3 (June 2022), 145–160. https://doi.org/10.1089/3dp.2020.0346 Publisher: Mary Ann Liebert, Inc., publishers.

Christopher G. Cameron, Zach Fredin, and Neil Gershenfeld. 2022. Discrete Assembly of Unmanned Aerial Systems. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, Dubrovnik, Croatia, 339–344. https://doi.org/10.1109/ICUAS54217.2022.9836082

Kenneth C. Cheung and Neil Gershenfeld. 2013. Reversibly Assembled Cellular Composite Materials. *Science* 341, 6151 (September 2013), 1219–1221. https://doi.org/10.1126/science.1240889

COBOD. 2025. EU COBOD BOD2 Tech Specs. https://cobod.com/wp-content/uploads/2022/06/eu-cobod-bod2-techspecs.pdf

Construction-Robotics. 2025. SAM. https://www.construction-robotics.com/sam-2/

Formlabs. 2023. *How to Create Models Larger than Your 3D Printer's Build Volume*. https://formlabs.com/blog/how-to-create-models-larger-than-your-3d-printers-build-volume/?srsltid=AfmBOorRcjw2tRh1lKB0ZLuxpdlQneRp-MpEOIe1SXpBOsrqNYOpiKLh Accessed: 2025-07-17.

Marwan Gharbia, Alice Chang-Richards, Yuqian Lu, Ray Y. Zhong, and Heng Li. 2020. Robotic technologies for on-site building construction: A systematic review. *Journal of Building Engineering* 32 (2020), 101584. https://doi.org/10.1016/j.jobe.2020.101584

Sébastien Goessens, Caitlin Mueller, and Pierre Latteur. 2018. Feasibility study for drone-based masonry construction of real-scale structures. *Automation in Construction* 94 (October 2018), 458–480. https://doi.org/10.1016/j.autcon.2018.06.015

Christine E. Gregg, Damiana Catanoso, Olivia Irene B. Formoso, Irina Kostitsyna, Megan E. Ochalek, Taiwo J. Olatunde, In Won Park, Frank M. Sebastianelli, Elizabeth M. Taylor, Greenfield T. Trinh, and Kenneth C. Cheung. 2024. Ultralight, strong, and self-reprogrammable mechanical metamaterials. *Science Robotics* 9, 86 (January 2024), eadi2746. https://doi.org/10.1126/scirobotics.adi2746 Publisher: American Association for the Advancement of Science.

Christine E. Gregg, Joseph H. Kim, and Kenneth C. Cheung. 2018. Ultra-Light and Scalable Composite Lattice Materials. *Advanced Engineering Materials* 20, 9 (2018), 1800213. https://doi.org/10.1002/adem.201800213 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adem.201800213.

HadrianX. 2025. Hadrian X® | Outdoor Construction & Bricklaying Robot from FBR. https://www.fbr.com.au/view/hadrian-x

S. Hauser, M. Mutlu, P.-A. Léziart, H. Khodr, A. Bernardino, and A.J. Ijspeert. 2020. Roombots extended: Challenges in the next generation of self-reconfigurable modular robots and their application in adaptive and assistive furniture. *Robotics and Autonomous Systems* 127 (2020), 103467. https://doi.org/10.1016/j.robot.2020.103467

Allen Hsu, Annjoe Wong-Foy, Brian McCoy, Cregg Cowan, John Marlow, Bryan Chavez, Takao Kobayashi, Don Shockey, and Ron Pelrine. 2016. Application of micro-robots for building carbon fiber trusses. In *2016 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*. 1–6. https://doi.org/10.1109/MARSS.2016.7561729

Benjamin Jenett, Amira Abdel-Rahman, Kenneth Cheung, and Neil Gershenfeld. 2019. Material–Robot System for Assembly of Discrete Cellular Structures. *IEEE Robotics and Automation Letters* 4, 4 (October 2019), 4019–4026. https://doi.org/10.1109/LRA.2019.2930486

Benjamin Jenett, Sam Calisch, Daniel Cellucci, Nick Cramer, Neil Gershenfeld, Sean Swei, and Kenneth C. Cheung. 2017. Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures. *Soft Robotics* 4, 1 (March 2017), 33–48. https://doi.org/10.1089/soro.2016.0032 Publisher: Mary Ann Liebert, Inc., publishers.

Benjamin Jenett, Christopher Cameron, Filippos Tourlomousis, Alfonso Parra Rubio, Megan Ochalek, and Neil Gershenfeld. 2020. Discretely assembled mechanical metamaterials. *Science Advances* 6, 47 (November 2020), eabc9943. https://doi.org/10.1126/sciadv.abc9943

Ben Jenett and Kenneth Cheung. 2017. BILL-E: Robotic Platform for Locomotion and Manipulation of Lightweight Space Structures. In *25th AIAA/AHS Adaptive Structures Conference*. American Institute of Aeronautics and Astronautics, Grapevine, Texas. https://doi.org/10.2514/6.2017-1876

Benjamin Jenett, Neil Gershenfeld, and Paul Guerrier. 2018. Building Block-Based Assembly of Scalable Metallic Lattices. In *Volume 4: Processes*. American Society of Mechanical Engineers, College Station, Texas, USA, V004T03A053. https://doi.org/10.1115/MSEC2018-6442

Benjamin(Benjamin Eric) Jenett. 2020. *Discrete mechanical metamaterials.* Thesis. Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/130610 Accepted: 2021-05-14T16:29:23Z.

Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. 2017. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 2606–2616. https://doi.org/10.1145/3025453.3026016

Alexander Htet Kyaw, Se Hwan Jeon, Miana Smith, and Neil Gershenfeld. 2025. Making Physical Objects with Generative AI and Robotic Assembly: Considering Fabrication Constraints, Sustainability, Time, Functionality, and Accessibility. arXiv:2504.19131 [cs.RO] https://arxiv.org/abs/2504.19131 Preprint, presented at the CHI 2025 Workshop on Generative AI and Human–Computer Interaction.

Sophie Leamon. 2025. *Computational Design of Architected Lattices for Construction Applications*. Master's thesis. MASSACHUSETTS INSTITUTE OF TECHNOLOGY, Cambridge, MA, USA. Any additional info (optional).

Samuel Leder, HyunGyu Kim, Ozgur Salih Oguz, Nicolas Kubail Kalousdian, Valentin Noah Hartmann, Achim Menges, Marc Toussaint, and Metin Sitti. 2022. Leveraging Building Material as Part of the In-Plane Robotic Kinematic System for Collective Construction. *Advanced Science* 9, 24 (2022), 2201524. https://doi.org/10.1002/advs.202201524 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.202201524.

Luban3D. 2025. *Luban 3D*. https://www.luban3d.com/ Accessed: 2025-07-17.

Z. Ma, A. Walzer, C. Schumacher, R. Rust, F. Gramazio, M. Kohler, and M. Bächer. 2020. Designing Robotically-Constructed Metal Frame Structures. *Computer Graphics Forum* 39, 2 (2020), 411–422. https://doi.org/10.1111/cgf.13940 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13940

Taylor Marchment and Jay Sanjayan. 2020. Mesh reinforcing method for 3D Concrete Printing. *Automation in Construction* 109 (January 2020), 102992. https://doi.org/10.1016/j.autcon.2019.102992

Nicholas A. Meisel, Nathan Watson, Sven G. Bilén, José Pinto Duarte, and Shadi Nazarian. 2022. Design and System Considerations for Construction-Scale Concrete Additive Manufacturing in Remote Environments via Robotic Arm Deposition. *3D Printing and Additive Manufacturing* 9, 1 (February 2022), 35–45. https://doi.org/10.1089/3dp.2020.0335 Publisher: Mary Ann Liebert, Inc., publishers.

Nathan Melenbrink, Ariel Wang, and Justin Werfel. 2021. An Autonomous Vault-Building Robot System for Creating Spanning Structures. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 7066–7072. https://doi.org/10.1109/ICRA48506.2021.9561004 ISSN: 2577-087X.

Achim Menges, Bob Sheil, Ruairi Glynn, and Marilena Skavara. 2017. *Fabricate 2017*. UCL Press. https://doi.org/10.2307/j.ctt1n7qkg7

Jonas Neubert and Hod Lipson. 2016. Soldercubes: a self-soldering self-reconfiguring modular robot system. *Autonomous Robots* 40, 1 (January 2016), 139–158. https://doi.org/10.1007/s10514-015-9441-4

NXP. 2021. I2C-bus specification and user manual. 2021 (2021).

In-Won Park, Damiana Catanoso, Olivia Formoso, Christine Gregg, Megan Ochalek, Taiwo Olatunde, Frank Sebastianelli, Pascal Spino, Elizabeth Taylor, Greenfield Trinh, and Kenneth Cheung. 2023. SOLL-E: A Module Transport and Placement Robot for Autonomous Assembly of Discrete Lattice Structures. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Detroit, MI, USA, 10736–10741. https://doi.org/10.1109/IROS55552.2023.10341479

Alfonso Parra Rubio, Dixia Fan, Benjamin Jenett, José Del Águila Ferrandis, Filippos Tourlomousis, Amira Abdel-Rahman, David Preiss, Jiri Zemánek, Michael

Triantafyllou, and Neil Gershenfeld. 2023. Modular Morphing Lattices for Large-Scale Underwater Continuum Robotic Structures. *Soft Robotics* 10, 4 (August 2023), 724–736. https://doi.org/10.1089/soro.2022.0117

PensaLabs. 2025. *Pensa Labs*. https://www.pensalabs.com/ Accessed: 2025-07-17.

Kirstin Petersen, Radhika Nagpal, and Justin K. Werfel. 2011. TERMES: An Autonomous Robotic System for Three-Dimensional Collective Construction. In *Robotics: Science and Systems VII*, Hugh Durrant-Whyte, Nicholas Roy, and Pieter Abbeel (Eds.). MIT Press, Cambridge, MA. https://doi.org/10.15607/RSS.2011.VII.035

Kirstin H. Petersen, Nils Napp, Robert Stuart-Smith, Daniela Rus, and Mirko Kovac. 2019. A review of collective robotic construction. *Science Robotics* 4, 28 (March 2019), eaau8479. https://doi.org/10.1126/scirobotics.aau8479 Publisher: American Association for the Advancement of Science.

Ava Pun, Kangle Deng, Ruixuan Liu, Deva Ramanan, Changliu Liu, and Jun-Yan Zhu. 2025. Generating Physically Stable and Buildable Brick Structures from Text. arXiv:2505.05469 [cs.CV] https://arxiv.org/abs/2505.05469

Charlotte Roux, Kateryna Kuzmenko, Nicolas Roussel, Romain Mesnil, and Adélaïde Feraille. 2023. Life cycle assessment of a concrete 3D printing process. *The International Journal of Life Cycle Assessment* 28, 1 (January 2023), 1–15. https://doi.org/10.1007/s11367-022-02111-3

Alfonso Parra Rubio, Klara Mundilova, David Preiss, Erik Demaine, and Neil Gershenfeld. 2023. Kirigami Corrugations: Strong, Modular and Programmable Plate Lattices. In *Proceedings of the ASME 2023 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE2023)*. ASME. https://doi.org/10.1115/DETC2023-116481

Lawrence Sass and Marcel Botha. 2006. The Instant House: A Model of Design Production with Digital Fabrication. *International Journal of Architectural Computing* 4, 4 (2006), 109–123. https://doi.org/10.1260/147807706779399015

Tobias A. Schaedler and William B. Carter. 2016. Architected Cellular Materials. *Annual Review of Materials Research* 46, 1 (2016), 187–210. https://doi.org/10.1146/annurev-matsci-070115-031624 _eprint: https://doi.org/10.1146/annurev-matsci-070115-031624.

Antun Skuric, Hasan Sinan Bank, Richard Unger, Owen Williams, and David González-Reyes. 2022. SimpleFOC: A Field Oriented Control (FOC) Library for Controlling Brushless Direct Current (BLDC) and Stepper Motors. *Journal of Open Source Software* 7, 74 (2022), 4232. https://doi.org/10.21105/joss.04232

Miana Smith, Amira Abdel-Rahman, and Neil Gershenfeld. 2024. Self-Reconfigurable Robots for Collaborative Discrete Lattice Assembly. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 3624–3631. https://doi.org/10.1109/ICRA57147.2024.10609866

Miana Smith, Jack Forman, Amira Abdel-Rahman, Sophia Wang, and Neil Gershenfeld. 2025. Voxel Invention Kit: Reconfigurable Building Blocks for Prototyping Interactive Electronic Structures. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3706598.3713948

Miana M. Smith. 2023. *Recursive Robotic Assemblers*. Thesis. Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/152015 Accepted: 2023-08-30T16:00:19Z.

Sophia Wang, Miana Smith, and Neil Gershenfeld. 2023. Voxelcopter: Modular Autonomous Aerial Systems. In *Proceedings of the 8th ACM Symposium on Computational Fabrication*. ACM, New York City NY USA, 1–2. https://doi.org/10.1145/3623263.3629155

Yeoreum Yoon and Daniela Rus. 2007. Shady3D: A Robot that Climbs 3D Trusses. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, Rome, Italy, 4071–4076. https://doi.org/10.1109/ROBOT.2007.364104 ISSN: 1050-4729.