Shaurya Sarma

University of Pennsylvania

Immersive Experience of Lattice Structure Configurator using Haptic Gloves and MR glasses

RWTH Aachen University

Digital Additive Production DAP

Supervisors:
Gustavo Melo, Henrik Kruse, Kilyan Talhouet

UROP – Undergraduate Research Opportunities Program

Aachen Germany
July 23, 2025

Table of Contents:

Abstract

This research project focuses on how immersive technologies—specifically Virtual Reality (VR), Augmented Reality (AR), and Extended Reality (XR)—can transform Additive Manufacturing (AM) workflows through interactive, haptic simulations of compressible lattice structures. Conventional approaches to evaluating lattice mechanical behavior often rely on physically printing and testing samples, which is time-consuming and expensive. This project explores a digital alternative: a real-time XR environment that allows users to generate lattice structures procedurally, manipulate parameters dynamically, and feel their mechanical response through SenseGlove Nova 2 haptic gloves. To accomplish this, a custom pip eline was built between Unity and Blender. Material behavior is simulated using data from compression stress-strain curves to provide a realistic experience. The system functions as a digital twin—bridging simulation and reality through physical sensation—and offers a novel platform for product development, enabling designers and engineers to interactively compare structural variants and make faster, more informed decisions. This simulation exemplifies how digital twins and immersive tools can accelerate innovation across design, additive manufacturing, materials science, and engineering.

Introduction

Additive Manufacturing (AM), particularly with polymeric and metallic lattice structures, has opened new design frontiers in engineering—enabling lightweight, high-strength, and highly customized parts. These lattice structures, composed of repeating unit cells, are critical in applications ranging from **automotive components** to **custom insoles**, where their geometry determines key mechanical properties such as stiffness, energy absorption, and compression response (see Fig. 1).

However, many of these properties are not very easily **predictable** through simply looking at geometry. Instead, engineers conventionally rely on physical prototypes—printing samples and mechanically testing them, which can be costly and inefficient.

This project explores how **immersive XR technologies**, particularly VR paired with **haptic feedback gloves**, can provide a solution. XR can be used to display the results of simulations and **material behaviors**, allowing engineers and designers to evaluate compression response without ever leaving the virtual environment. By combining data from compression stress-strain curves with **force response models** and **procedural geometry generation**, users can feel how a lattice might behave in the real world, without needing to manufacture it.

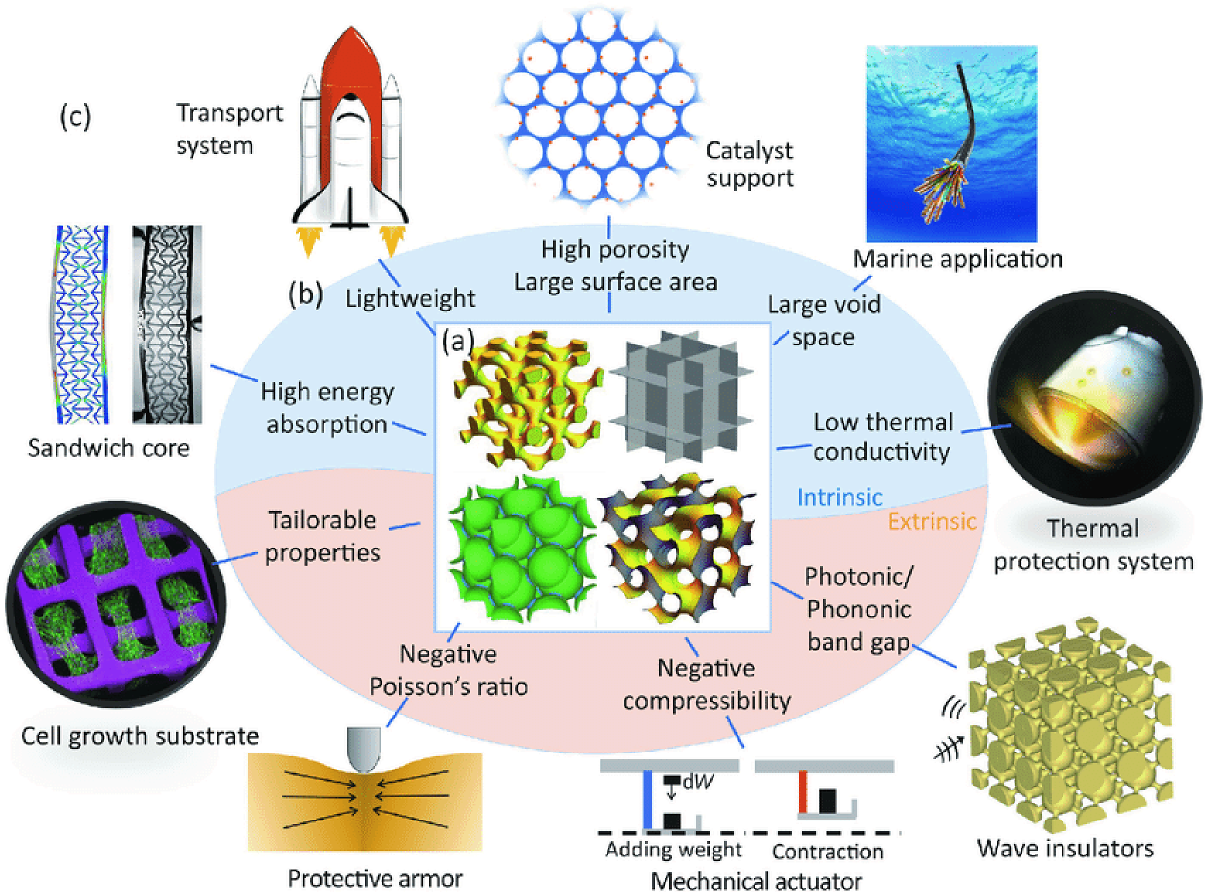

Fig. 1. Range of lattice applications based on unit cell geometry [6]

This work builds upon prior research conducted at the Digital Additive Production (DAP) Lab, specifically the paper *Force-Feedback Interaction with Lattice Structures using Haptic Gloves in Virtual Reality* [2], which examined how participants interacted with and perceived the stiffness of various lattice structures with differing elastic moduli. A primary focus of that study was optimizing the user experience in extended reality (XR). Furthermore, it was able to identify the most natural pinching gesture and determine appropriate dimensions for a lattice cube to fit comfortably within a user's hand during interaction (see Fig. 2).

This project also draws on results from other studies, such as *Compressive Behavior of Thermoplastic Polyurethane with Active Agent Foaming for 3D-Printed Customized Comfort Insoles* [1], which provided stress-strain curve data from real-world TPU lattice structures and demonstrated how structural variations impact material performance. This data was temporarily used to inform the force-compression response values in the virtual environment.

Fig. 2. Participant using Oculus Quest 3 and SenseGlove Nova 2 haptic gloves [2]

A key innovation in this project is the creation of a **real-time bidirectional pipeline** between **Unity** and **Blender**. Users adjust lattice design properties via sliders in Unity; these parameters are written to a JSON file, processed by a headless Blender instance, and used to regenerate a procedural mesh using Geometry Nodes. The updated `.glb` file is then automatically loaded into the Unity environment for users to immediately test with the haptic gloves. [3]

The resulting system provides a high-fidelity, cost-effective, and immersive alternative to physical prototyping, transforming how engineers can design, test, and iterate on AM components in the **Industrial Metaverse**.

Project Description

The goal of the project was to build an XR experience to allow engineers and designers to test the compressive behavior of lattice cube specimens seamlessly. The solution required combining **materials engineering** concepts, **procedural modeling**, and **virtual reality**.

Key topics included:

- Understanding how **lattice geometries** (unit cells) affect material properties like stiffness and energy absorption (see Fig. 4).
- Applying **stress-strain curve data** to model real-world compression behavior in VR.
- Designing a custom, file-based **Unity–Blender bridge** for live geometry and material behavior updates.
- Building realism through tweaking VR controls and the **force-feedback** settings for the haptic gloves by adding damping and controlling elastic moduli.
- Addressing technological and physical constraints, such as SenseGlove's maximum 20N force per finger limit and inaccurate finger tracking due to bulky and limited sensors.

Project Data / Conducted Research

1. Mechanical Behavior of Lattices

Research into additive manufacturing revealed that **predicting the mechanical behavior** of lattice structures is extremely difficult due to their complex geometries (see Fig. 3). As a result, physical testing remains the most reliable method for deriving accurate **stress-strain relationships**, especially under compression.

In particular, **thermoplastic polyurethane (TPU)** lattices have been widely used in applications such as footwear and protective padding due to their energy absorption and ability to recover to their original state after deformation [1]. These materials typically exhibit an **elastic phase**, followed by a **plateau region** characterized by large strains and minimal stress increase (plastic deformation), and eventually densification (see Fig. 4).

For this project, the simulation focused on accurately modeling the **elastic deformation region**, as this is:

- The **most perceptible** phase via the SenseGlove's force-feedback response
- The **easiest and simplest** to simulate using linear or pseudo-linear models,
- The phase where **design variation** (e.g., varying unit cell geometry) has the greatest impact and change.

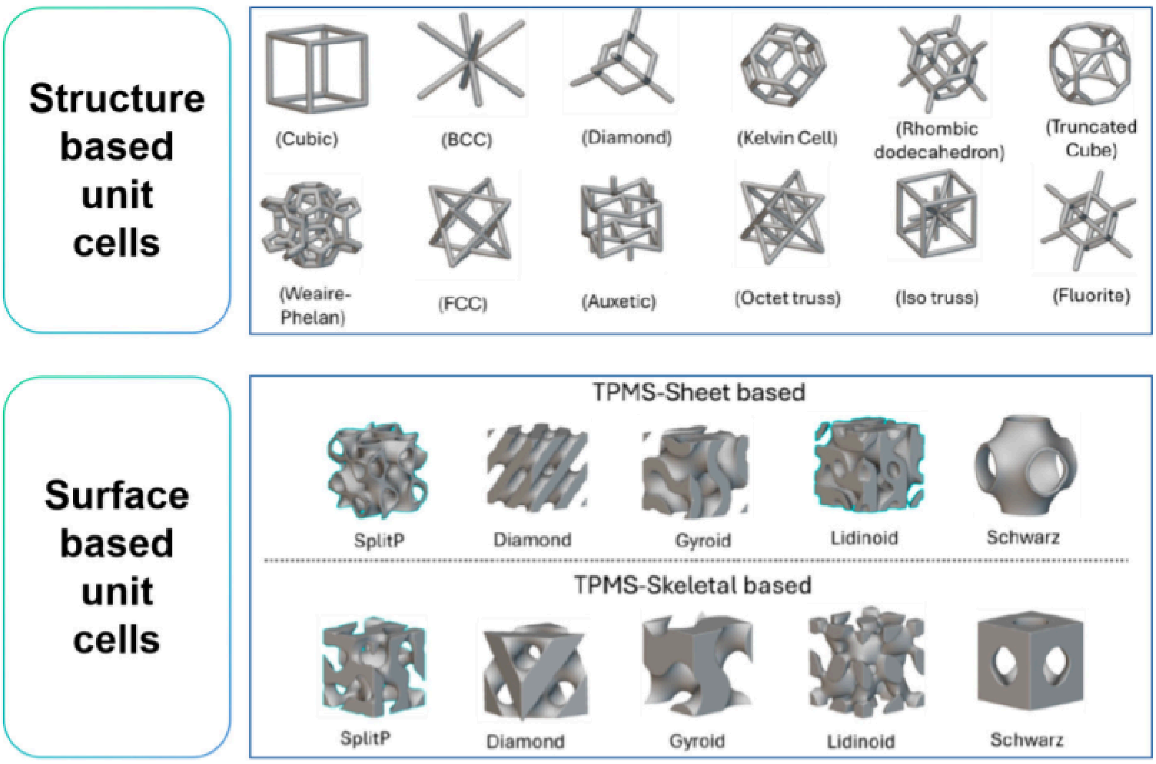

Fig. 3. Different types of lattices with varying unit cell structures [7]

To replicate a real-world feel, TPU stress-strain curves from literature [1] were referenced, and values within the **0.1–1.0 MPa modulus range** were mapped to Unity's force response curves. This mapping formed the foundation of the haptic simulation logic used throughout the XR testing environment.

Additionally, the choice to prioritize elastic behavior aligns with prior findings that **users are more sensitive to force changes in low-strain regions**. By calibrating Unity's haptic parameters to this zone, the system could deliver meaningful differences in perceived stiffness between lattice variations.

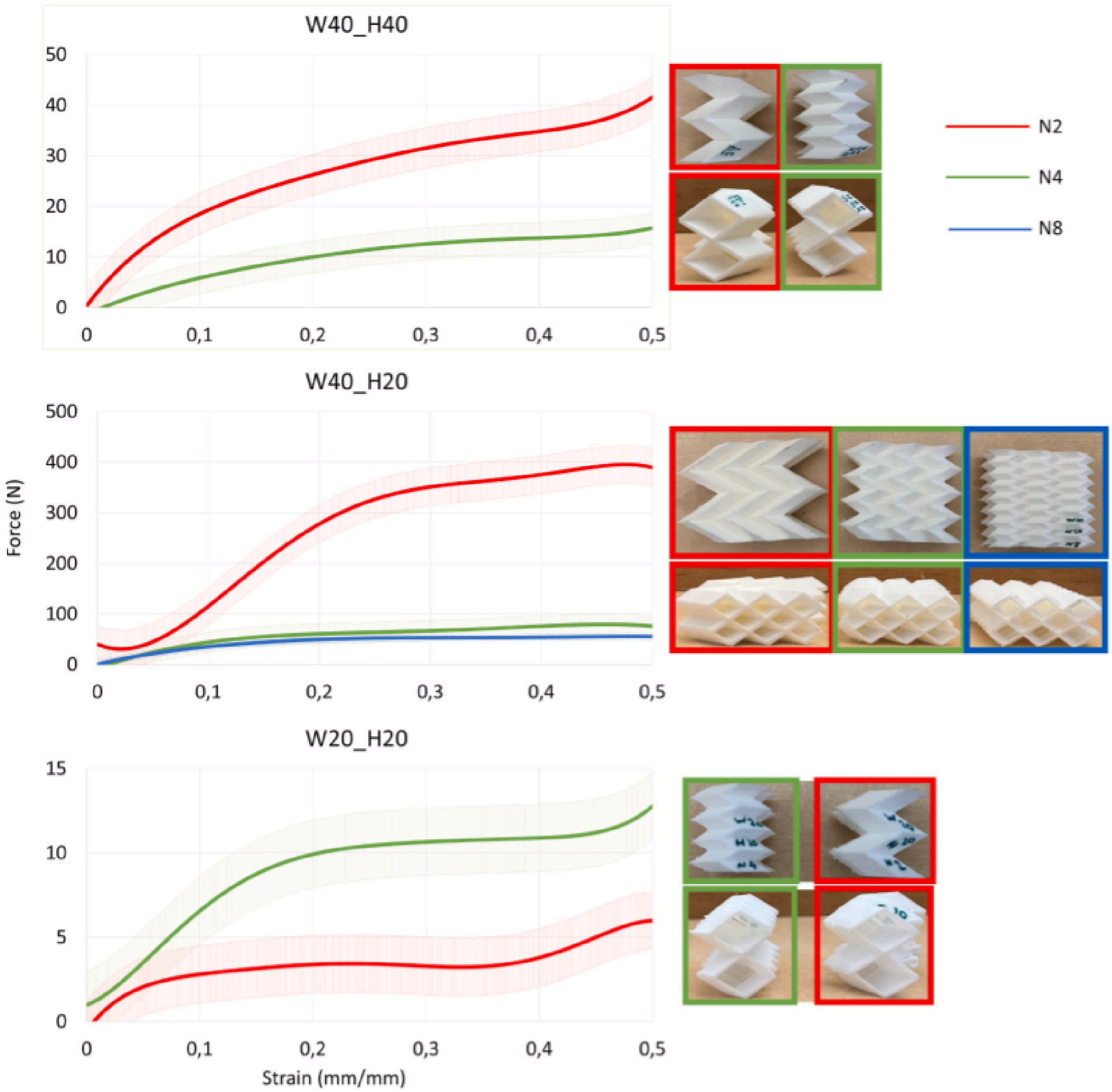

Fig. 4. Stress-strain curves for 3D-printed TPU auxetic structures[5]

2. Force Modeling in Unity and SenseGlove Integration

Realistic simulation of lattice compression in an XR environment required an in-depth understanding of the **SenseGlove SDK** and its integration into the **Unity XR Interaction Toolkit**. Following the official Unity tutorials and hardware documentation [3][4], the system was configured to support customized force-feedback responses with materials while providing locomotion and interaction simultaneously.

To facilitate user locomotion and material interaction at the same time, a dual-input setup was implemented: one hand was assigned to a **VR controller** for movement, turning, and VR interaction, while the other was equipped with a **SenseGlove Nova 2** for object manipulation. This configuration required a custom input layer to be written that overrode the default prefab structure provided by the toolkit.

SenseGlove Unity SDK manages hand detection, physics interactions, and object manipulation within an XR environment (see Fig. 5). Each box represents a key or component, with arrows indicating functional dependencies and method calls between layers.

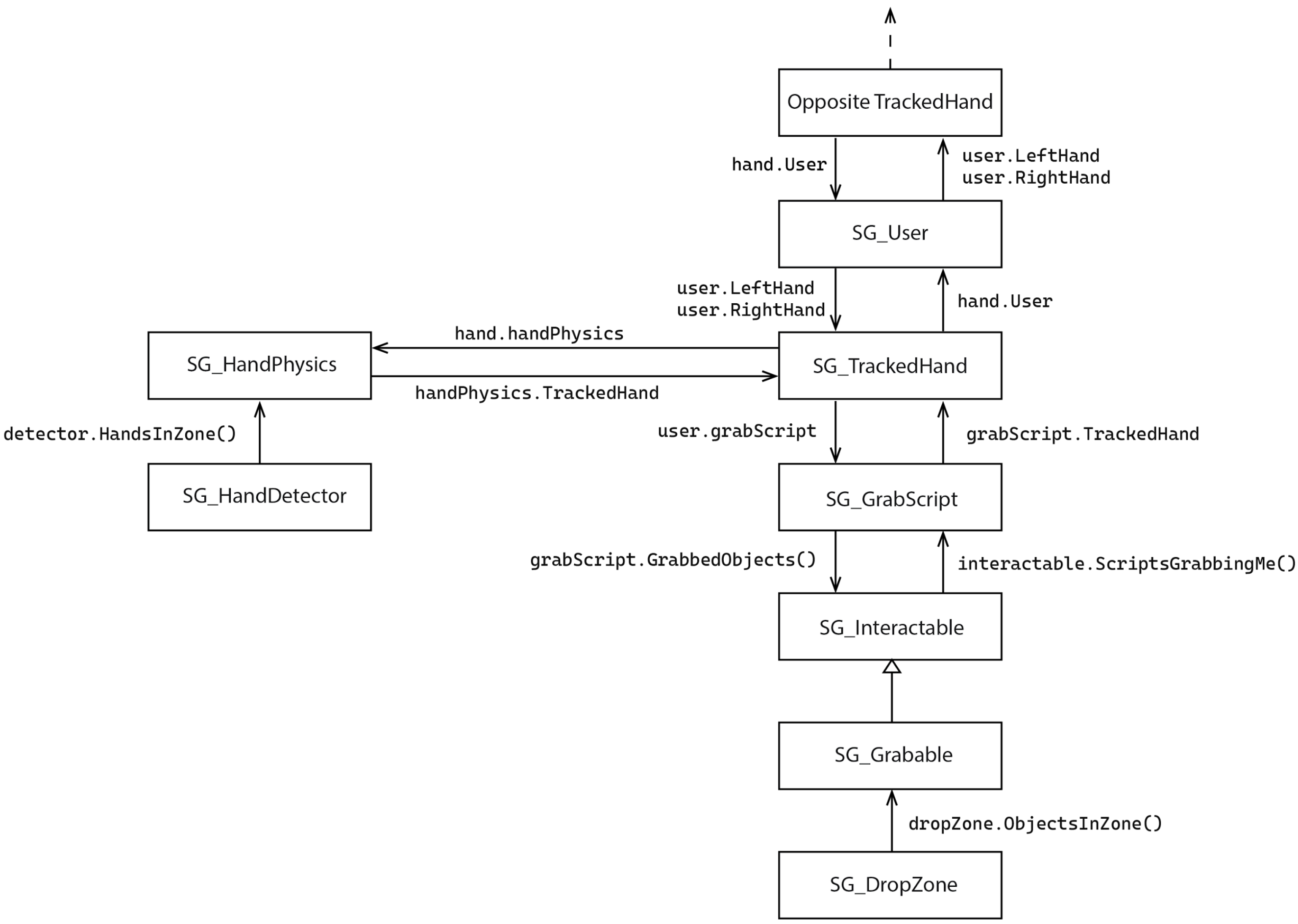

Fig. 5. Layers of SenseGlove and hierarchy of functions and behaviors [3]

Within Unity, each interactable object that supports haptic interaction is configured using three key parameters exposed by the SenseGlove SDK:

- **Force Response** – A curve defining how the applied force scales as a finger penetrates deeper into the object collider. The X-axis maps penetration depth, and the Y-axis maps force output.
- **Max Force Distance** – A scalar value that controls how much distance corresponds to full force output (max compression)
- **Max Force** – The peak force that a material can output to the glove's flexor tendons (maximum of 20N per finger)

Adjusting these parameters allowed the system to simulate realistic **force-displacement responses** derived from **stress-strain curves** of thermoplastic polyurethane (TPU) lattice samples [1].

Furthermore, visual deformation was added to enhance realism, based on evidence that **visible compression significantly improves perceived stiffness** [2]. Unity's SenseGlove toolkit contained scripts and deformation shaders, enabling objects to visibly deform when compressed (see Fig. 6).

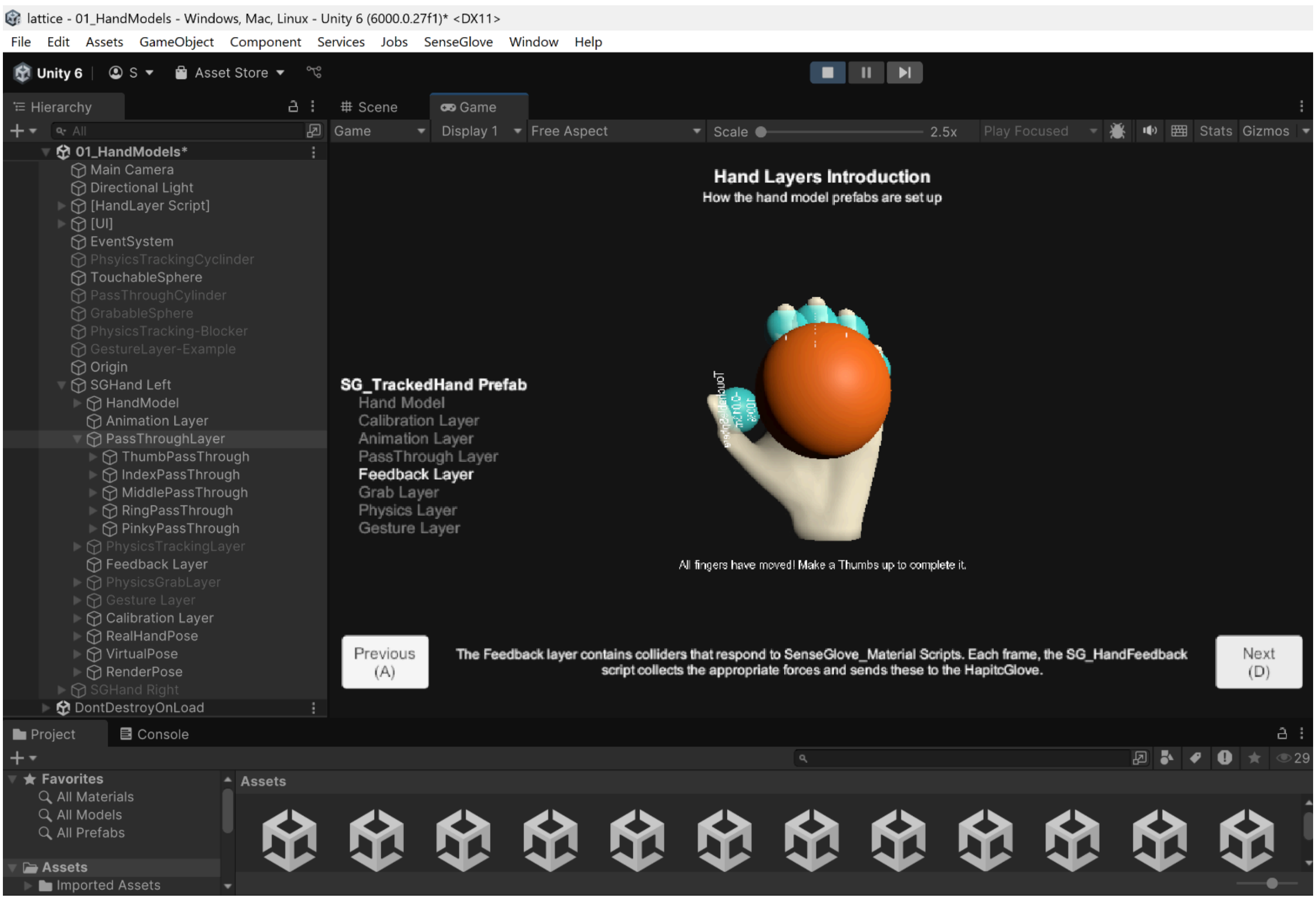

Fig. 6. The feedback layer that allows visual deformation through SenseGlove's SDK by collecting and sending force information in <20 ms to the haptic glove model

While the Nova 2 glove provides robust haptic output, it presents several **hardware limitations**:

- **Pinky tracking is not supported** due to the lack of a fifth finger sensor. Instead, its movement is mirrored from the ring finger
- **Vibrotactile motors** are bulky and mounted on the fingertips, which can occasionally interfere with accurate finger tracking or small object grasping.
- When performing **pinch-based compression** with two fingers on top and a thumb below, the thumb would be exerting a maximum of 40N to keep the object in place and therefore exceed the 20N per finger threshold. Thus, the force readings for the thumb may be inaccurate.

Despite these constraints, a functional and immersive XR scene was established, allowing users to grab, compress, and feel lattice structures with varying elastic responses using the SenseGlove system. The table below summarizes key technical specifications:

Feature	Description
Finger Tracking	3 independently tracked fingers (thumb, index, middle); ring/pinky are mirrored
Force Feedback Output	Max 20N per finger via flexor tendons
Haptic Feedback Parameters	Force Response (curve), Max Force Distance, Max Force
Grasp Detection	Built-in object grabbing through Unity toolkit; visual deformation included
Vibrotactile Feedback	Fingertip-mounted motors though bulky; provide buzzing and resistance
Other Specifications	Lightweight exoskeleton; motorized wrist front strap for squeezing sensation

3. Determining Specimen Dimensions and Compression Parameters

A critical part of the project involved determining the **appropriate size and shape of the virtual specimen**, which had to fulfill two constraints:

- Be **ergonomically sized** to fit in the user's hand and be pinchable between two fingers.
- Be **mechanically compatible** with the elastic modulus (0.1–1 MPa) defined by TPU stress-strain curves, such that deformation is easily perceptible

We used **Hooke's Law** to relate force F and displacement ΔL for linear elastic materials:

$$F = \frac{E \cdot A \cdot \Delta L}{L}$$

Where:

- F is the applied force,
- E is the elastic modulus (MPa),
- A is the cross-sectional area of the specimen (cm²),
- L is the initial length of the specimen (cm),
- ΔL is the compression (cm).

By varying the **cross-sectional area** and **initial height**, we were able to control the **maximum displacement** achievable within the 20N-40N force limitation of the SenseGlove. Several dimension sets were tested:

Dimensions (W × D × H)	Initial Height (L)	Max/Min Compression (ΔL)	Deformation Profile
2 cm × 2 cm × 4 cm	40 mm	4 cm to 0.04 cm	Deformation very noticeable; hard to grip
4 cm × 4 cm × 6 cm	60 mm	2.25 cm to 0.0225 cm	Moderate deformation; easier to hold
6 cm × 6 cm × 6 cm	60 mm	1 cm to 0.01 cm	Very stiff; near-zero feedback
4 cm × 4 cm × 10 cm	100 mm	2.5 cm to 0.25 cm	Ideal balance between grip and feedback

4. Design Configurator Evolution

The project began with a **Rhino + Grasshopper** workflow to create a **lattice design configurator**—a visual programming system that allows users to modify 3D geometry through a node-based interface (see Fig. 7). In this setup:

- **Rhino** served as the CAD environment for previewing geometry.
- **Grasshopper**, a plugin for Rhino, enabled **parametric control** of the lattice models using **nodes** and **wires**, where each node represented a geometric operation or mathematical function.

This configurator allowed designers to experiment with **lattice unit cell parameters** such as strut thickness, cell size, wall thickness, and repetition count. Modifying these parameters affected the **overall stiffness, density, and compressibility** of the printed structure.

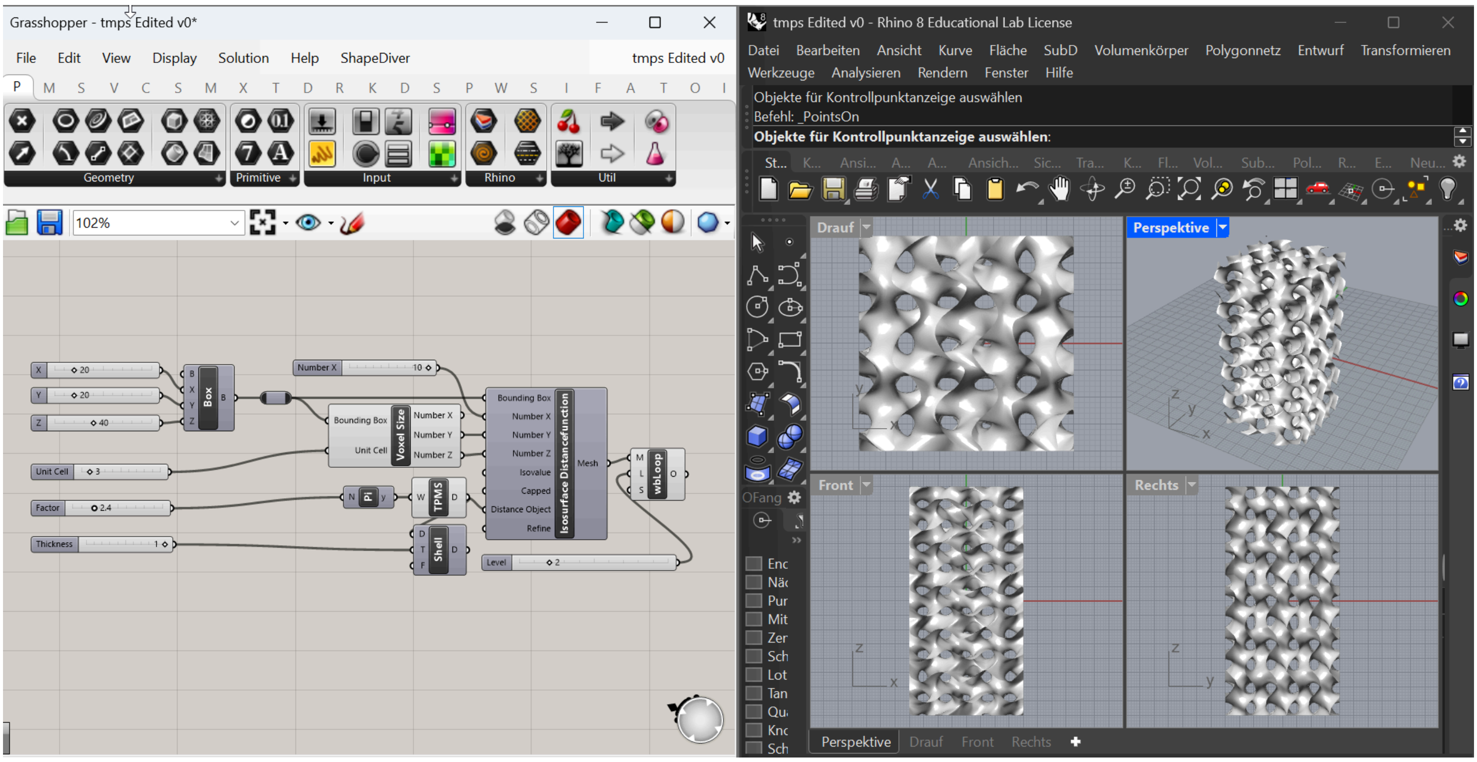

Fig. 7. Procedural 3D lattice structure modeling setup with Grasshopper's parametric node-based system (left-side) and Rhino's mesh evaluation results (right-side)

Parameter-driven (procedural) modeling is especially flexible and scalable for lattice design because these structures consist of **repetitive**, **mathematically** defined unit cells. Instead of manually modeling complex geometries, designers can quickly generate countless variations by adjusting key parameters like strut thickness or cell size. This enables rapid iteration and optimization, whereas manual modeling of such repetitive geometry would be inefficient and impractical.

However, a significant drawback of the Rhino-Grasshopper pipeline was the **mesh generation performance**. As the lattice became more refined or complex, the polygon count rose dramatically. Generating a complete mesh often took **12-16 minutes**, making it impractical for interactive or iterative design in real time—especially when integrated into an XR workflow.

5. BlenderBridge Implementation

A custom file-based system was built using **Blender Geometry Nodes** and **Python scripting** (see Fig. 8). Most of the processes occur within a shared file directory called *BlenderBridge*.

- Unity VR sliders update a `params.json` file in a shared folder.
- Unity triggers a **headless Blender** subprocess that loads a scene with a Geometry Node setup.
- A **Python script** replaces the input values in Blender using the JSON data.
- Blender evaluates the mesh and **simplifies geometry** to reduce vertex count.
- The result is exported as a `.glb` file to the same shared directory.
- A **Unity file-watcher** detects the change and loads the new model in VR without restarting play mode.

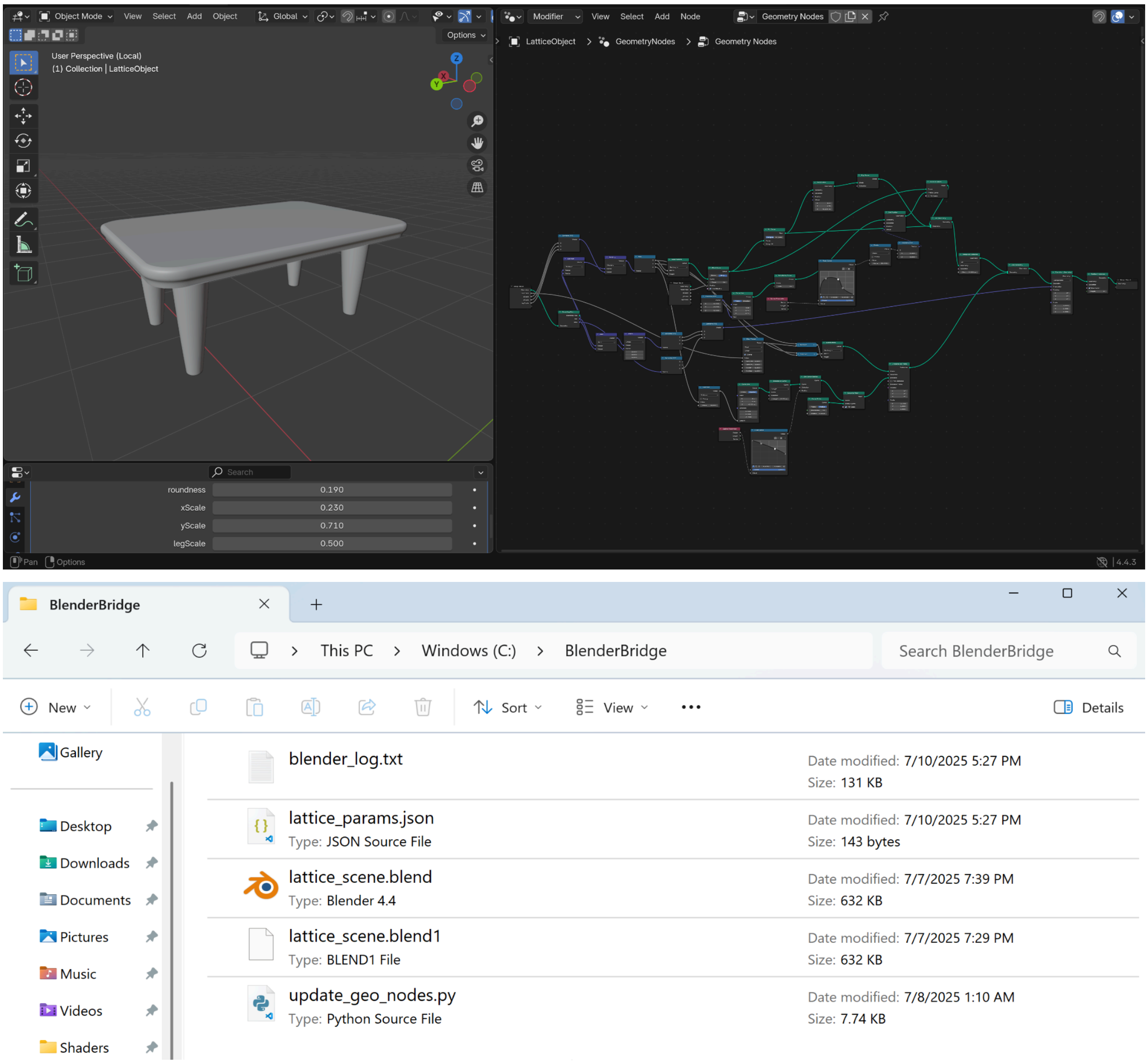

Fig. 8. A test version of an arbitrary geometry node setup in Blender (top) and the shared *BlenderBridge* directory with the corresponding files (bottom)

However, a fully procedural Geometry Node system for generating lattice meshes in Blender could not be completed in time. Instead, a **custom Python script** was created, leveraging libraries such as **NumPy, PyVista, and Trimesh** to generate and export 3D **TPMS (Triply Periodic Minimal Surface)** lattice meshes (see Fig. 9).

This Python pipeline followed a similar process to the BlenderBridge system, but invoked the script via a terminal subprocess. A huge advantage with the Python script was that it, on average**, took 10-60 second**s to generate a mesh depending on the complexity of the lattice structure (far quicker than previous methods).

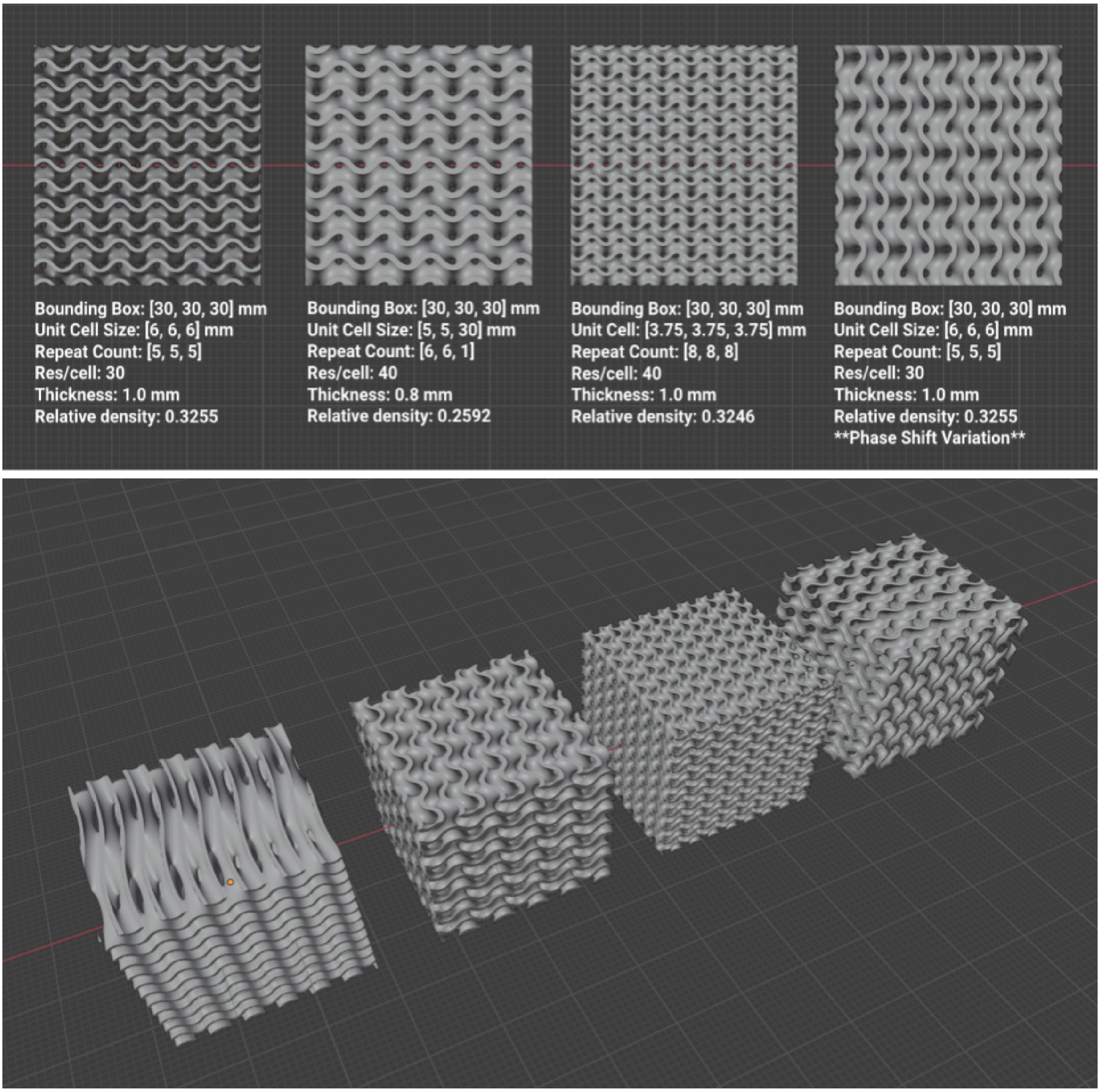

Fig. 9. Sample lattices generated with different variations of parameters via the Python script.

6. Unity Simulation Scene and Demo

The VR environment was split into **three distinct scenes**, each focusing on providing a different experience for the participant: onboarding, testing, and open-ended exploration.

Tutorial Scene

The first scene introduced users to the core VR interactions required throughout the simulation:

- Moving and turning using XR locomotion controls
- Grabbing and releasing virtual objects
- Calibrating and using the **SenseGlove** to interact with lattice samples

This scene focused on building comfort with VR and hand-tracking systems before users encountered any task-based challenges, since the simulation ideally should be usable by even users with no prior VR experience (see Fig. 10).

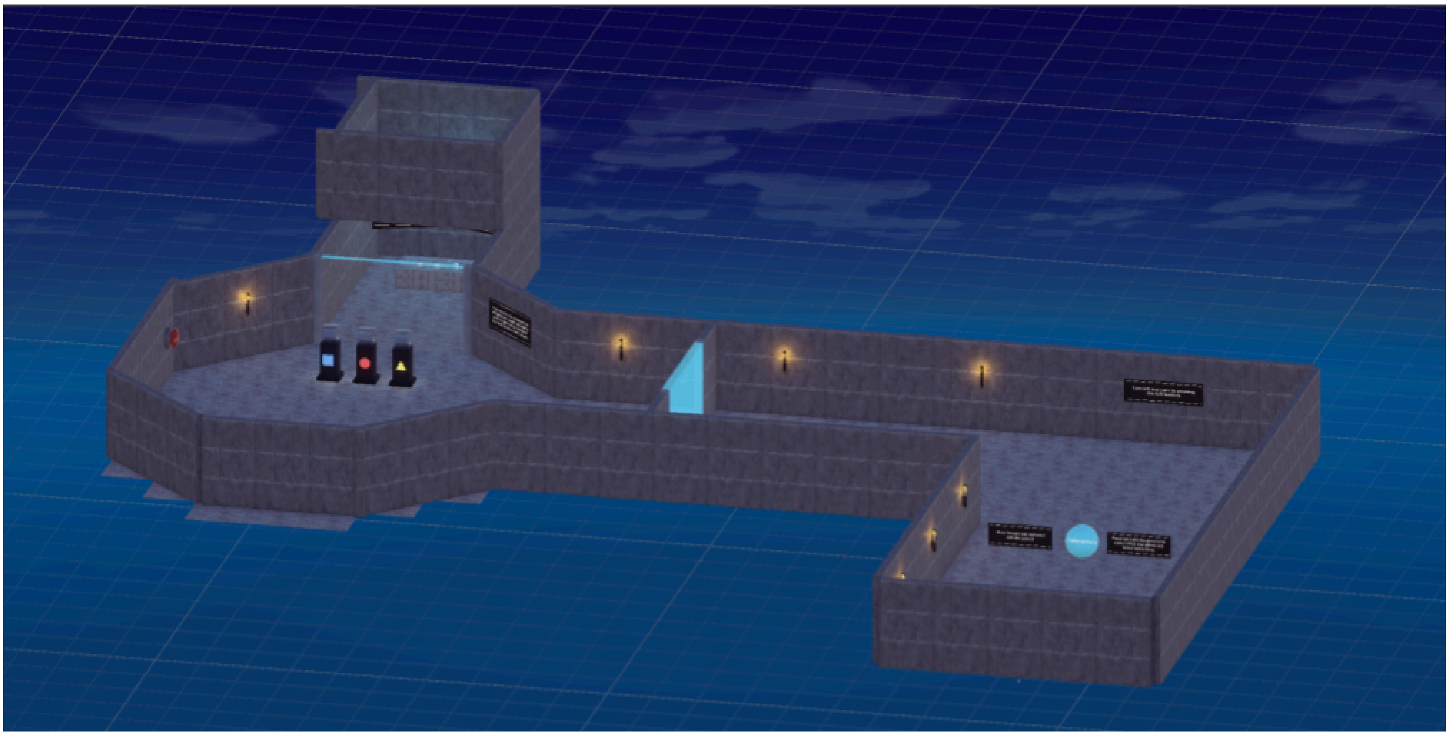

Fig. 10. Overview of the tutorial scene setup in the Unity Editor viewport

Testing Scene

The second scene was designed as a **blind stiffness sorting task** (see Fig. 11). Users were presented with three physically identical lattice cubes, each with a different simulated stiffness **(soft, medium, hard)** in the 0.1–1.0 MPa range.

Key interaction flow:

- Users were asked to sort the lattices into labeled pedestals based solely on haptic perception.
- They received no visual or numerical cues about stiffness.
- Only upon correctly sorting all three cubes could they proceed.

This setup allowed for controlled perceptual testing to verify whether users could perceive differences in the **stiffnesses** (elastic moduli) of the virtual lattice cubes.

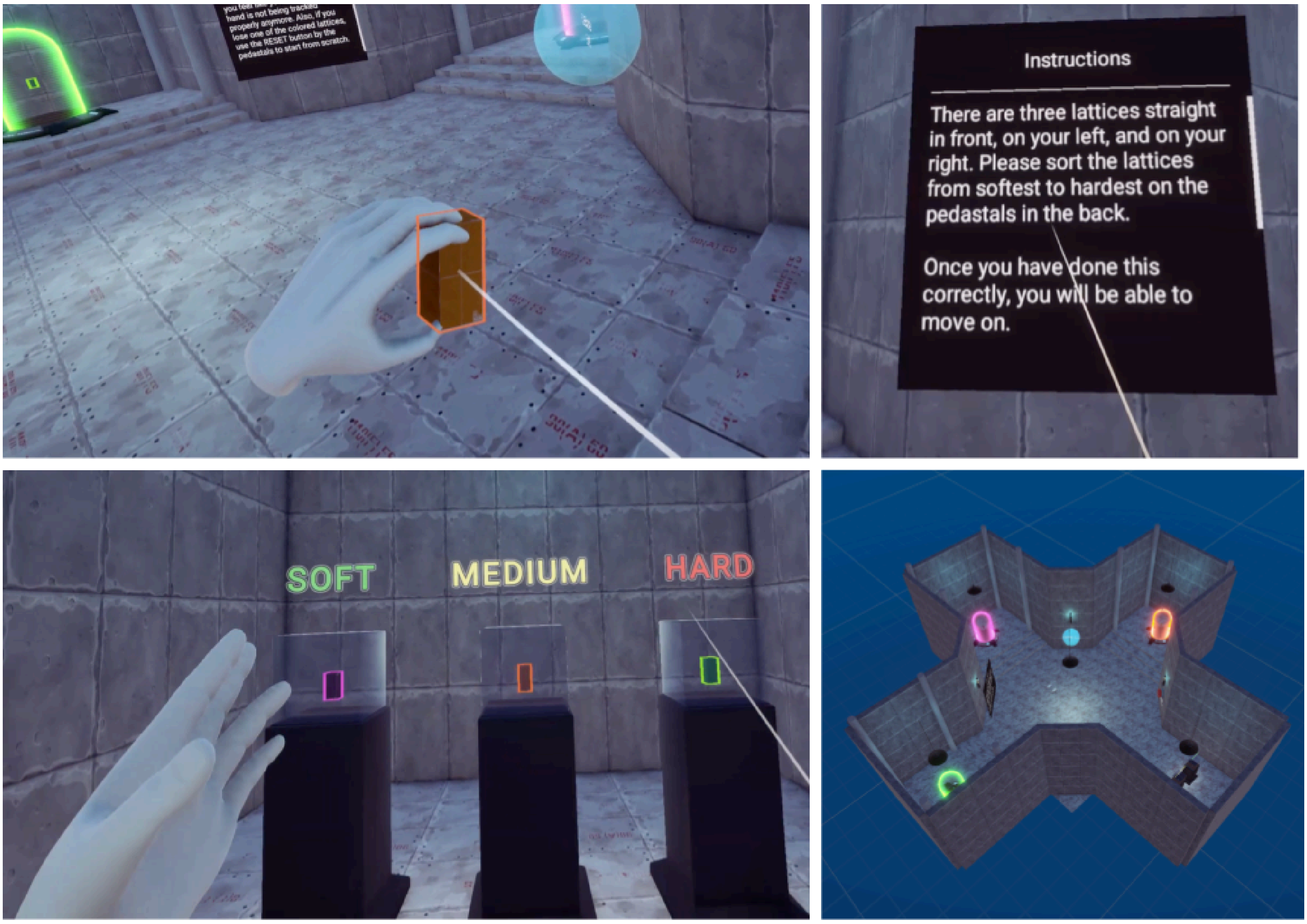

Fig. 11. The blind stiffness sorting task where users were given three cubes to sort into soft, medium, and hard.

Playground Scene

The third scene enabled users to explore **procedural lattice design** more freely. Users could manipulate several key parameters: **bounding box dimensions (xyz), unit cell size, lattice thickness, repetition count (xyz), mesh resolution per cell, and phase shift (xyz).** This was all conducted through a 3D console mesh within the environment, allowing users to interact with sliders and dials to adjust parameters. (see Fig. 12).

These changes were applied in real-time to generate **TPMS-based lattices** through the **custom Python script**. For each variation, the system calculated the **geometric density (relative density)**—defined as the ratio of the lattice's solid volume to the total bounding box volume—which directly influenced the lattice's estimated **elastic modulus**. A denser lattice (relative density closer to 1.0) resulted in a stiffer structure, whereas a sparse lattice (e.g., 0.05) was softer. To convey this stiffness through haptics, the **force response** of the SenseGlove was **linearly mapped (lerped)** to the 0–1 relative density range, ensuring that higher densities felt firmer and lower densities softer.

Users could then test the stiffness with the haptic gloves, providing a **proof of concept** of how XR technology could be used in new design and engineering workflows.

Creation of the Environment

Several newer XR-native components within Unity's XR toolkit were used to build immersion and intuitive interaction:

- **XR Sockets:** Used to snap lattice objects into pedestals during sorting, ensuring precise placement and reinforcing the task structure.
- **XR Sliders/Dials**: Granted users the ability to adjust lattice parameters via "physical" interaction inside virtual reality, as opposed to traditional UI elements, to enhance immersion
- **Raycast Interactables:** Allowed users to interact with UI elements or distant objects by pointing and clicking the trigger button
- **Poke Interactors:** Enabled push button logic, allowing users to physically push buttons to activate them and trigger functionality to enhance immersion

Additionally, the environment was built using various assets from the **Unity Asset Store** and **Sketchfab**. The rest of the models were custom-made using Blender and textured with assets found online.

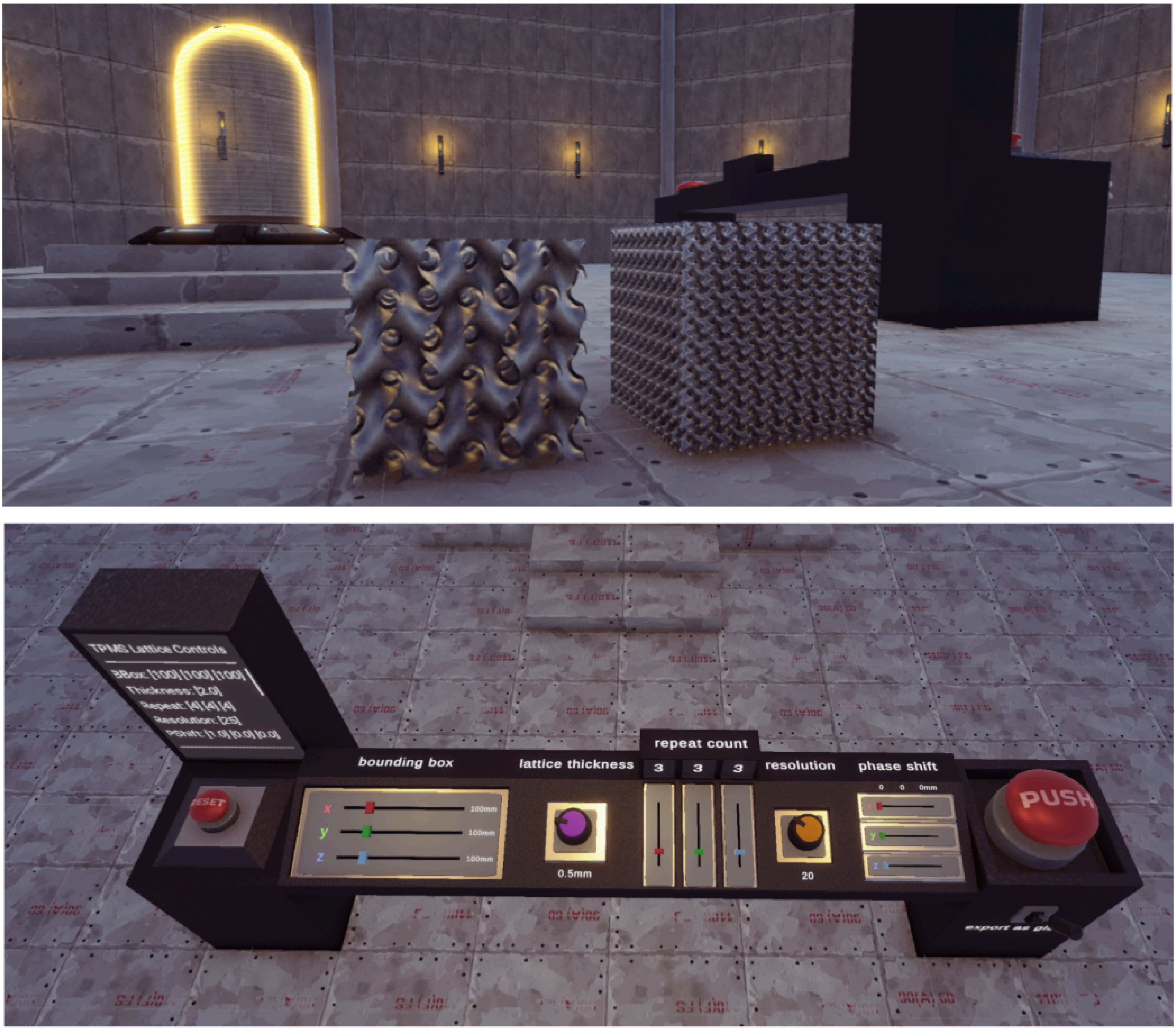

Fig. 12. Breakdown of the Playground Scene including the main console with adjustable parameters (bottom) and examples of generated meshes within the environment

Result of Research

- Developed a seamless XR workflow to configure, simulate, and compress virtual lattice structures in real time.
- Mapped real-world compression behavior to Unity's haptic model, simulating elastic deformation with force feedback
- Implemented a custom file-based bridge between Unity and Blender, enabling scalable mesh generation and parameter-driven control.
- Designed an interactive XR environment, giving users a step-by-step approach to working with VR and haptic gloves with both an experimental and open-ended simulation.
- Created a working prototype for future bidirectional XR design systems that combine procedural modeling with immersive interaction.

Evaluation / What did I learn?

This project helped me combine knowledge across multiple disciplines: materials science, XR system design, CAD modeling, and haptics. Specifically, I learned:

- Learned how to connect digital prototypes with physical behavior, making simulation-based design **more accessible** and **interactive**
- Developed a deeper understanding of how **immersive tools** can support engineering workflows, enabling more intuitive, data-driven, and collaborative prototyping
- Strengthened my skills in system integration, ensuring smooth **interoperability** between CAD tools, simulation platforms, and XR interfaces
- The **limitations** of current glove tech and how to find workarounds to simulate real-world interactions as closely as possible
- The importance of **procedural modeling tools** and their real-time performance tradeoffs.

More broadly, this work gave me a deeper appreciation of how immersive tech can transform traditional workflows by bridging the gap between the physical and digital worlds.

Bibliography

[1] M. Zhu, H. Li, S. Song, Q. Liu, Y. Zhang, and Z. Zhang, “Compressive behavior of thermoplastic polyurethane with an active agent foaming for 3D-printed customized comfort insoles,” *Polymer*, vol. 287, p. 126430, 2024. https://doi.org/10.1016/j.polymer.2024.126430

[2] Digital Additive Production (DAP) Lab, “Force-Feedback Interaction with Lattice Structures using Haptic Gloves in Virtual Reality,” unpublished manuscript, 2024.

[3] SenseGlove, *Unity SDK Documentation – Getting Started*. [Online]. Available: https://unity.docs.senseglove.com/GettingStarted/overview.html

[4] SenseGlove, *Nova 2 Technical Documentation*. [Online]. Available: https://senseglove.gitlab.io/SenseGloveDocs/nova-2.html

[5] A. R. Jabareen, D. Yahav, A. Shirizly, “A hybrid cellular design for improving the mechanical performance of 3D printed TPU lattice structures,” Polymer Testing, vol. 134, May 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142941824003465

[6] J. R. Greer and J. T. M. De Hosson, "Engineering lattice metamaterials for extreme mechanics," J. Appl. Phys., vol. 127, no. 15, p. 150901, Apr. 2020. [Online]. Available: https://doi.org/10.1063/1.5145090

[7] Y. Li, D. Jiang, R. Zhao, X. Wang, L. Wang, and L.-C. Zhang, “High mechanical performance of lattice structures fabricated by additive manufacturing,” *Metals*, vol. 14, no. 10, art. no. 1165, Oct. 2024.