Mallet-Based Assembly: Enabling Load-Bearing Laser-Cut Models

Shohei Katakura
Hasso Plattner Institute
Potsdam, Germany
shohei.katakura@hpi.de

Chiao Fang
Hasso Plattner Institute
Potsdam, Germany
chiaojfang@gmail.com

Mehdi Gouasmi
Hasso Plattner Institute
Potsdam, Germany
Mehdi.Gouasmi@uni-potsdam.de

Lino Hellige
Hasso Plattner Institute
Potsdam, Germany
lino.hellige@student.hpi.uni-potsdam.de

Yoan Tchorenev
Hasso Plattner Institute
Potsdam, Germany
yoan.tchorenev@student.hpi.uni-potsdam.de

David Bizer
Hasso Plattner Institute
Potsdam, Germany
hello@davidbizer.com

Conrad Lempert
Hasso Plattner Institute
Potsdam, Germany
conrad.lempert@hpi.de

Martin Taraz
Hasso Plattner Institute
Potsdam, Germany
martin.taraz@hpi.de

Muhammad Abdullah
Hasso Plattner Institute
Potsdam, Germany
muhammad.abdullah@hpi.de

Patrick Baudisch
Hasso Plattner Institute
Potsdam, Germany
patrick.baudisch@hpi.de

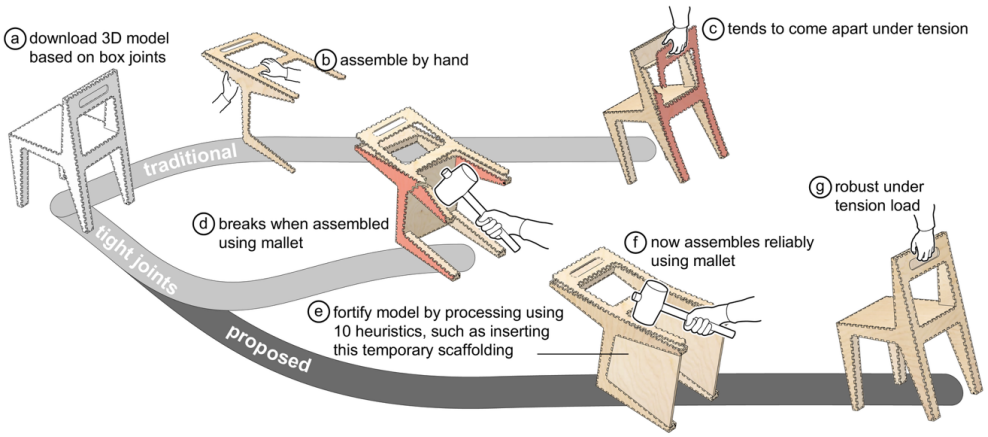


Figure 1: (a) 3D models designed for laser cutting typically use joints that are loose enough to (b) allow users to assemble the models by hand. However, (c) when tension is applied, the resulting models tend to break. (d) Presumably, one way to make the physical model robust against stress is to make joints extremely tight and assemble them using a mallet. However, our survey finds that many models break during assembly if one tightens their joints. (e) We analyze the 10 types of weaknesses identified in the survey and present an algorithm that addresses them by choosing assembly sequences that bypass error-prone states and, as shown here, introduces what we call scaffolding. (f) This allows for reliable assembly of models with tight joints, thereby enabling (g) physical models that are robust against tension

UIST '25, Busan, Republic of Korea

ACM ISBN 979-8-4007-2037-6/2025/09
https://doi.org/10.1145/3746059.3747788

Abstract

Laser cutting has a long tradition of building load-bearing 3D objects based on box joints and T-joints, as these joints are naturally robust against compression and shearing. Achieving robustness against *tension*, however, is challenging. One presumed solution is to make all joints extremely tight, to the point where they can only be assembled using a mallet. However, our survey found that making joints tight can cause models to break during assembly. In this paper, we identify the 10 underlying issues and present techniques for overcoming them: by extending parts with what we call *scaffolding* or by adjusting the models' assembly order, so as to bypass states that are subject to these issues. Based on our user study and analysis of laser-cut models, *scaffolding* speeds up assembly for an average of 14% of the assembly operations per model, which in turn gives an average of 1.3x speed-up per model, and 70% of the models would benefit from the adjusted assembly order, that in the absence of such, would require higher assembly effort.

CCS Concepts

• **Applied computing**; • **Computer-aided design; Computer-aided manufacturing;**;

Keywords

personal fabrication, laser cutting, assembly, tensile

ACM Reference Format:
Shohei Katakura, Chiao Fang, Mehdi Gouasmi, Lino Hellige, Yoan Tchorenev, David Bizer, Conrad Lempert, Martin Taraz, Muhammad Abdullah, and Patrick Baudisch. 2025. Mallet-Based Assembly: Enabling Load-Bearing Laser-Cut Models. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3746059.3747788

1 Introduction

Laser-cut structures based on box joints, such as *closed box* structures [5], offer impressive robustness against compression and shearing, making them key contenders for creating sturdy 3D objects, such as furniture and guitars [25]. The main weakness of box joints, however, is that they are susceptible to *tension*, i.e., as soon as a 3D model offers a way to apply tension, models tend to break (Figure 1a-c).

A presumed solution, as illustrated in Figure 1d, is to stick to box or T-joints but make them extremely tight, i.e., by dilating joints by a small amount (such as 0.08mm [12]) to withstand the tension (or to produce the normal force required for wood glue to set [23]). (e) Unfortunately, many of the 3D models we examined in our survey would not assemble or would break during assembly (see Section "Survey") if struck using a rubber or wooden hammer, also known as a *mallet* [21] (which allows delivering up to 4500N [24]).

At first glance, this may appear surprising. We all have struck things with a hammer or mallet many times—and successfully so. However, on closer inspection, the reason we succeeded was not because hammering is generally reliable (or easy for that matter) but because we hammered a *carefully engineered* object, such as a nail. As illustrated in Figure 2a, a nail features a head carefully crafted for impact. It transmits the impact perfectly to the tip. It is made from a robust material, etc. (b) Laser-cut 3D models, in contrast, are *not* engineered that way. Some parts cannot be successfully inserted; other models will physically break when trying to.

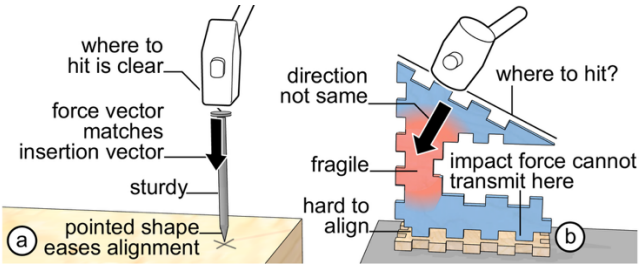


Figure 2: (a) While nails have been carefully engineered for being hammered, (b) laser-cut 3D models are not.

In this paper, we carefully analyze what exactly goes wrong when trying to assemble models containing tight joints. We classify our findings and organize them into 10 types of issues. We come up with alternative approaches that overcome the respective issues: by modifying the joints, generating what we call scaffolding, optimizing assembly order, etc. We then combine these elements into an algorithm. As illustrated by Figure 1f, we use our algorithm to update the models, which inserts scaffolding in this case, (g), and now enables assembling them *successfully* using a mallet, which gives us physical objects that we can now subject to tension. In our user study, non-expert participants assembled models with tight joints significantly faster and with less damage to the models when they were processed using our algorithms.

2 Contribution and Limitation

In this paper, we make three main contributions. First, we identify 10 systemic issues from assembling tight joints by surveying laser-cut models subject to tensile forces. Second, we propose corresponding solutions, including *scaffolding* and an *assembly order refinement algorithm* that bypasses error-prone states. In a user study, we show that pre-processing 3D models using these methods overcomes the issues.

Limitations: (1) Trapezoidal jigs simplify assembly, but at the expense of consuming additional material. (2) Our assembly order refinement algorithm builds on *assembly-by-disassembly* [14] and *directional blocking graphs* [31] and is, therefore, limited to models that can be assembled one plate at a time. (3) Our exploration is limited to models made from box joints and T-joints; we did not consider cross joints, as they tend to be less sturdy in the first place [5]. (4) For similar reasons, we only looked at models made from material suitable for being struck using a mallet, particularly plywood; we did not investigate brittle material, such as acrylic.

3 Related Work

The research builds on previous work in designing and assembling sturdy laser-cut 3D models, joint designs, and jigs.

3.1 Designing Load-Bearing Laser-Cut 3D Objects

Design tools evolved from 2D systems such as *Cutcad* [15] and *Cuttle.xyz* [9] into 3D systems such as *SketchChair* [26]*, Joinery* [32], *FlatFitFab* [22], and *Kyub* [5].

While many 3D laser-cut models are designed solely for decorative purposes, functional models like furniture must withstand significant loads. For example, chairs endure compression, string instruments and hanging lamps experience tensile stress, and shelves [25] endure bending and shear forces. We address such models, with a specific focus on tensile loads.

3.2 Laser Cutting Joints vs. Tensile Load

As the introduction mentions, tension is the biggest challenge for laser-cut 3D models. One approach to bypass the issue is to eliminate any option of applying tensile loads, e.g., fully enclosed convex volumetric models, also called closed box structures (*Kyub* [5]). However, tensile loads are back if structures feature finger-sized cutouts or embedded objects. Similarly, compressive loads applied inside the closed box, e.g., heavy items in shelves/drawers, can apply tensile loads at the joints.

To address this, researchers and makers proposed several alternative joint designs, including the ones shown in Figure 3. Unfortunately, as discussed in the introduction, each introduces a new weakness. Extending box joints into T-joints [5] fails when tension is applied from a second direction. Cross-joints [22] are susceptible to shearing. While Snap-fits [4] provide easy assembly and locking, they, together with T-joints and cross-joints, protrude out of the design, often violating aesthetic requirements. Wood screws or nails, while quite sturdy (Umetani et al. [29]), split thin materials commonly used in laser cutting, e.g., 6mm plywood. Captive nut joints [8] add complexity and assembly time.

As a result, objects requiring both *structure* and *aesthetics*, such as furniture, chairs, guitars, and shelves [25], cannot be handled using any of the above tension-aware joints. This limits the adoption of these joints, thus limiting laser cutting.

To overcome these issues, *FastForce* proposed adding reinforcement structures against tensile loads on the (assumed to be unused) *inside* of closed boxes [3]. However, large internal plates obstruct functionality, e.g., drawers and shelves.

As a result, the only universally applicable solution is to make joints tight. Given proper kerf calibration (e.g., *Kerfmeter* [16]), box joints can handle tensile loads of up to approximately 200N per pin (6mm birch plywood) based on our measurements. However, *assembling* tight joints is surprisingly challenging, as we address in this paper.

3.3 Assembling Laser-Cut Objects

Structural stability is important for the fully assembled object and plays a key role during the process, e.g., at intermediate stages of large-scale frame structures [30].

In laser cutting, assembly is optimized for different reasons, such as speed (*Roadkill* [1]), assembly by the visually impaired (Daedalus [7]), and folding (*HingeCore* [2], Lamifold [19], Flaticulation [11]).

While previous work in laser cutting considers assemblability in terms of collision and interlock between parts, we provide a new perspective: the mechanical capability of parts and joints, and human errors that make assembly impossible or difficult. Our algorithm tweaks assembly order to simplify the assembly of laser-cut objects with tight joints.

3.4 Alternative Joints Designed for a Tensile Load

Some alternative joint designs are less susceptible to tension (Figure 3). However, they tend to handle tension in only one of two directions, make models more susceptible to shearing, create undesirable protrusions, or run the risk of splitting the material, and they all add complexity and assembly effort.

3.5 Assembling Tight Joints Based on Interference/Press Fit

Most laser-cutting joints are based on *interference fits,* such as *press fits*. Here, a part (pin) is forced into a smaller opening (slot), creating static friction that secures the parts in place (*Tsugite* [18]).

Assembling *tight* press fits is particularly challenging since large forces are required that, in turn, require hammering tools, e.g., a mallet. To simplify and speed

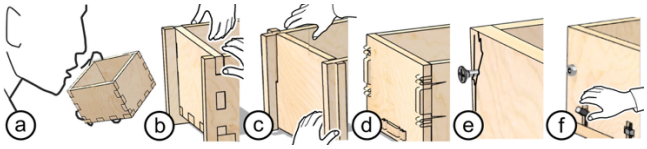


Figure 3: (a) One example of a Laser-cut object with tight box joints. Alternative joint designs have been proposed to make laser-cut 3D models more robust against tension, including (b) T-joints [3], (c) cross joints [8], (d) snap-fits [4], (e) wood screws, and (f) captive nut joints [8].

up assembly, mechanical engineers provided parts to be inserted with pointy tips, known as *Pilot pins* [6]. Gershenfeld et al. applied this concept to laser-cut cross joints to prevent jamming [13]. In this paper, we apply this concept to box and T-joints as one of many facets towards a solution.

3.6 Jigs

Jigs or custom tools are essential in woodworking (e.g., drilling) and milling 3D objects (e.g., fixtures). The primary engineering purpose of the auto-generated jigs is to overcome the imprecision of the user's ability.

With advancements in digital fabrication, jigs have evolved from in-situ physical tools to software-driven or algorithmically generated forms. Examples range from cutting wood larger than a machine (*JigFab* [20]) and manually welding metal pipes (*Trusscillator* [17]) to human-machine collaborative robotic fabrication (*Adroid* [27]).

The *trapezoidal scaffolding* introduced in this paper resembles jigs and sacrificial materials in that they are special-purpose and made to support the assembly process, but differ in that they modify the workpiece per se for easier assembly.

4 Survey: Making joints tight causes 10 issues

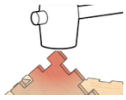
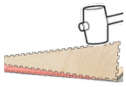
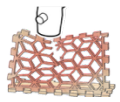
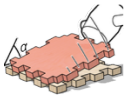
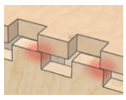
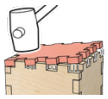
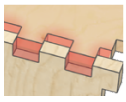
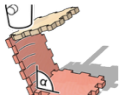
In order to better understand potential issues involved in the assembly of laser-cut 3D models with tight joints, we surveyed what happens to models when we tighten their joints. We obtained a copy of the 402 laser-cuttable 3D models at risk of falling apart upon tensile forces (from *FastForce* [3]).

Figure 4: 33 of the 402 models sensitive to tensile forces [3] (used with permission).

4.1 Results: Ten Issues

Table 1 summarizes the 10 issues we identified from looking at the models from the survey. We will discuss each point in additional detail in the following section. All issues refer to a single assembly step in which users try to add a given plate *P* to the set of all parts already assembled, which we refer to as assembly *A*. We refer

Table 1: Ten issues we identified.

1. Small Target: when the impact area of a target is not large enough the impact from the mallet will **damage the plate**.	
2. Target Orientation: if the target is angled w.r.t the *joint*, little of the impact transmits to the *joint*. Users thus hit harder, the **plate breaks**.	
3. Transmission: unless the target is roughly centered over *joint*, the impact fails to reach *all* pins. Users hit harder, the **plate breaks.**	
4. Multiple Targets: if a long joint offers too few targets to strike, some pins will not receive sufficient impact, the **plate never assembles**.	
5. Sturdy: if the path between impact area and *joint* is weakened by cutouts, the mallet's impact will **break the plate**.	
6. Angle: achieving a specific angle when assembling *two plates* is hard, especially non-90° joints: ⚠**Blocks subsequent assembly**	
7. Alignment: if pins on plate *P* are not aligned with the corresponding slots on assembly *A* **pins will break.**	
8. Hold: unless something holds *P* in place at the moment of impact, the mallet's impact may undo the alignment, on impact **pins break**.	
9. Intrusion: pins not fully inserted into slots are fine. However, if this happened for concave joints: ⚠**Blocks subsequent assembly**.	
10. Stability: striking *P* subjects all joints in *A* to the impact. Unless all plates in *A* already form a rigid structure, this **disassembles A**.	

to the joint between *P* and *A* as the *P-A joint* or just the *joint*.

Summarizing the above table, there were two main types of failures, with the first half commonly leading to plates *breaking* and the latter half leading to *blocking subsequent assembly*. We will thus address the ten issues in two separate clusters of five issues each. As we will see, each category will lead to its solution strategy, i.e., *trapezoidal scaffolding* and *constrain-before-insertion.*

To validate the relevance of the proposed strategies, we analyzed a subset of the 402 models (i.e., the 33 models in Figure 4) manually, 70% of the models would benefit from *constrain-before-insertion*, that in the absence of such, would require higher assembly effort, while 67% had at least one assembly operation that in the absence of *trapezoidal scaffolding* would run the risk of breaking the part.

5 Addressing issues 1-5: Trapezoidal Scaffolding

Looking at the first five issues from the list above, one might see a callback to the introductory discussion about hammering nails: The first five issues are issues because laser-cut objects are not like nails: Nails have right-sized impact areas that are properly oriented, they are sturdy, and transmit impact properly. Laser-cut parts do not—thus, they break.

In this section, we will try to make laser-cut parts *more like nails*. Figure 5 shows how we reinforce laser-cut parts with additional material, which we will refer to as *trapezoidal scaffolding*.

(a) If this shape (the plate from Figure 2b) is hard to hammer, (b) what shape should the plate ideally be? In analogy to a nail, such a plate might feature an exposed, easy-to-hit *target* area at the very top, and we would make that target area large enough not to break on impact. Similar to a nail, we might texture this target area to prevent the mallet from slipping and for improved affordance. The target area would be parallel to and centered over the P-A joint. We would give the plate an overall shape that transmits the mallet's impact well and transmits it equally to all pins of the P-A joint. The result of these design requirements is the shown *trapezoidal* shape.

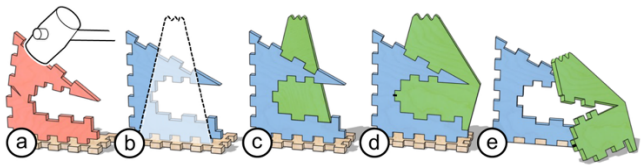


Figure 5: The trapezoidal scaffolding: (a) the plate from Figure 2b is hard to assemble. (b) A plate suitable for hammering might look like this. (c) We unite the plate and trapezoid, then subtract the plate (d) and merge the resulting parts. The resulting combination of plate and scaffolding affords easy and precise assembly. (e) When done, users discard the scaffolding.

(c) We now apply this insight to the plate *P*, shown in *blue*: to create an initial version of the scaffolding, we unite the trapezoidal with the actual plate. When subtracting the plate, we obtain what we call *scaffolding*, shown in *green*. (d) The scaffolding is a *sacrificial piece*, i.e., we plan on removing it later. To simplify the removal, we combine both parts into one. Finally, we connect the scaffolding with plate *P* using a single break-away tab, shown as a black dot, (e) making it easy to remove the scaffolding once the joint has been inserted.

While the scaffolding shown above focused on inserting a single plate into another single plate, the same concept applies equally well further along the assembly process, as illustrated by Figure 6.

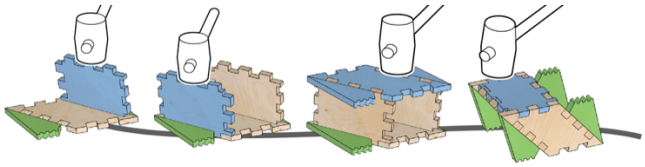


Figure 6: Trapezoidal scaffolding applies equally well further along the assembly process.

We now drill down into the top five issues from the table above and discuss how *trapezoidal scaffolding* supports them.

5.1 1. Target Size

Trapezoidal scaffolding provides plates with a target of a sufficiently large area. As illustrated by Figure 7a and b, a mallet's impact can dent or delaminate small targets. (c) The *trapezoidal scaffolding* enlarges the target, which protects it. We tend to generate target sizes of 3cm.

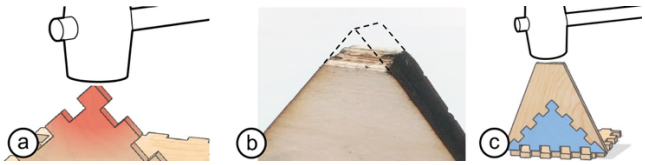


Figure 7: (a) Small targets (b) can get dented or delaminate on impact. (c) Trapezoidal scaffolding gives the plate a larger target.

As shown in Figure 8, when a target is located on the *surface* of plate *P,* we may still witness the same type of issue as plate *P* may still bend and even break

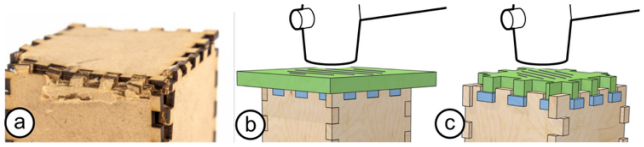


Figure 8: (a) Surface damaged by a mallet. (c) Sacrificial plate parallel to *P*. (c) In case assembly *A* uses extended pins (to allow for sanding), we may make the sacrificial piece custom.

as plate stiffness decreases squared with plate size [12]. (b) Since there is no vertical plate here to extend, we overlay a plate parallel to P instead. (c) In case assembly *A* uses extended pins (to allow for sanding [5]), we may generate a sacrificial piece in the shape of the clearance.

5.2 2/3. Can we use Existing Features instead of Scaffolding?

As we will see in our user study (See Section "User Study"), trapezoidal scaffolding works well. However, it comes at a cost because it requires additional material and cutting time. It is, therefore, worth checking whether scaffolding is necessary.

As illustrated by Figure 9, (a) Tabling the question of the target area for a moment, any protruding feature on plate *P* could serve as a target. (b) Finite element analysis verifies that this pin is in a good position to transmit an impact to all pins on the P-A joint. This makes this pin a good candidate for replacing the scaffolding. (c) In contrast, none of the pins are suitable for this plate, as each of them transmits too little of its impact to at least one of the pins on the P-A joint.

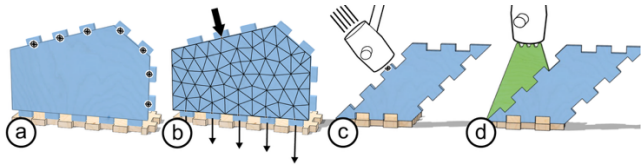


Figure 9: (a) Plates tend to offer multiple potential targets, each of which may be struck from various angles. (b) Finite element analysis confirms that this pin transfers impact to all five pins. (c) This part, in contrast, does not work; thus, (d) requires scaffolding.

Our algorithm proceeds as follows:

Step 1: Place one *potential target t* on every line segment of the outline of plate *P*, excluding those on the P-A joint. Place multiple *potential targets* along longer edges and onto surfaces. Give each *potential target* the normal *d* of the line segment as orientation.

Step 2: For each (*t, d*), compute the efficiency *e* (i.e., a value [0, 1]) of impact transmission to all pins of the P-A joint using finite element analysis (we wrote a custom FEA solver in *python* on top of *numpy*).

Step 3: Pick the target t_{max} with direction d_{max} that maximizes the minimum efficiency of force transmission across all pins of the *P-A joint*. Call that efficiency e_{max}.

Step 4: Test whether the structure is capable of withstanding an impact of the default force for a single pin divided by e_{max} (see subsection "7. Transmission").

Step 5: If not, extend plate P with trapezoidal scaffolding.

(c) To prevent assembly A from sliding, we place it on a thin rubber sheet or against a stop.

5.3 4. Multi-Trapezoidal Scaffolding for Long Joints

As illustrated by Figure 10a, the longer the *P-A joint*, the more pins need to be inserted and the higher the overall resistance. Furthermore, the lever formed by the width of plate *P* will, at some point, exceed the material stiffness, causing plate *P* to curve when struck until none of the impact reaches the outer pins anymore. Consequently, hitting a single target on a large joint may not produce sufficient impact.

(b) Our algorithm addresses this by producing *multiple* targets, spread out along plate P, so that each pin is close enough to one of the pins, thus receiving sufficient impact from at least that target.

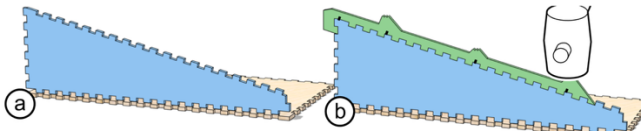


Figure 10: (a) Long joints made from many pins (b) can require multiple pieces of scaffolding, here united into one.

We may again try to save material and cut time by getting by without the scaffolding. As illustrated by Figure 11a, we may assemble *upside-down*, i.e., we insert a horizontal plate *P* into a vertical assembly *A*. In this orientation, *P* is much less stiff, allowing it to deform during insertion so that only a reasonably small neighborhood of pins can be inserted at a time, reducing the resistance and thus simplifying assembly.

(b) However, the deformation causes pins on plate *P* to come down at an angle, causing the effective width of the pin to be defined by the diagonal between its two contact points (sqrt(pinWidth^2 + intrusion^2)), which is substantially wider than the slot it is supposed to fit into, thus causing the pin to collide with the adjacent pins on assembly *A*. Users may avoid this by alternating strikes between targets, inserting each one by just a fraction of the pin length. (c) To also allow the linear approach, we provide pins with a tapered tip (also

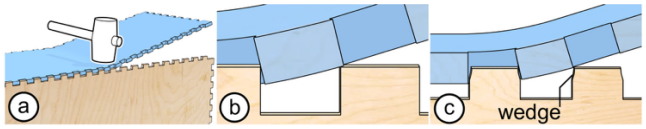


Figure 11: (a) Inserting a long plate P, as shown, (b) causes its pins to come down at an angle, causing its corners to collide. (c) We address the issue by adding a slant to the pins on A.

known as *pilot pins* [6][13]), albeit this reduces the resulting tensile strength of this joint.

5.4 5. Transmitting impact across cavities

As illustrated by Figure 12a, cavities and cutouts may weaken the path from the target to the pins on the P-A joint, causing plate *P* to collapse on impact. (b) Again, our algorithm addresses the issue by adding trapezoidal scaffolding. (c) For many small cutouts, (d) extending scaffolding to fill all the cutouts simplifies removal.

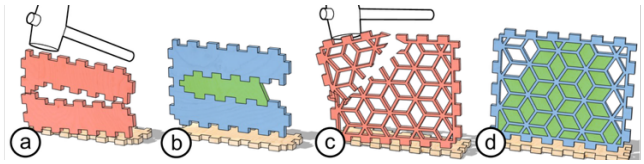


Figure 12: (a) The path between all potential targets and the *P-A joint* is weakened by a cavity. (b) Our algorithm fills in trapezoidal scaffolding. (c) For many small cutouts, (d) our algorithm simplifies removal by extending scaffolding to fill cutouts.

Again, it is worth checking whether we can get by without the scaffolding, especially for large cavities, as they consume more material. As illustrated by Figure 13a, if a cutout (here on the inside of the chair) is high and wide enough for a mallet to reach *past* the cavity,

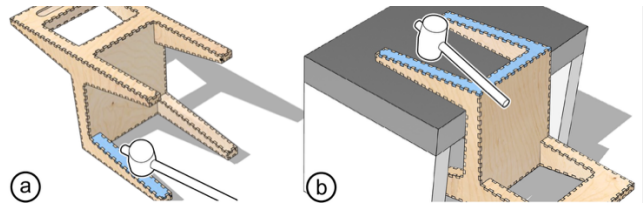


Figure 13: If the cavity or cutout is getting very large, (a) reaching the impact area becomes possible again, or (b) it can be supported with the help of an anvil.

users may get by without scaffolding. (b) When inserting a plate *P* into the *outside* of a cut-out Assembly *A*, a sufficiently large cavity allows inserting the protruding part of an *anvil* (here in the form of a table) to support the struck geometry. Again, users can save the scaffolding. This also applies to the chair in Figure 1f, which we showed for the purpose of easy explanation and which, in reality, is better addressed with an anvil.

As illustrated by Figure 14a, *vertical* cavities are challenging in a slightly different way. Even though there are adequate targets for each pin (one on each side of the cavity), striking one target at a time leaves the other half of the plate *P* behind. This results in shearing forces that tear plate *P* apart. (b) Once again, trapezoidal scaffolding resolves the issue.

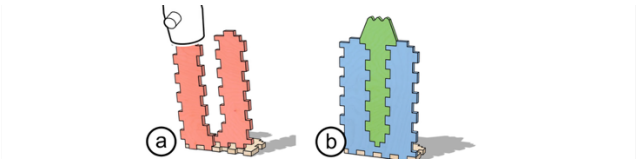


Figure 14: (a) Assembling plates with vertical cavities is subject to shearing. (b) Again, we use trapezoidal scaffolding.

This situation is particularly common with living hinges. As illustrated by Figure 15a, plate *P* consists of multiple regions separated by the living hinges. However, since the incisions forming the living hinges are thin and we need the result to be curved, we cannot simply apply trapezoidal scaffolding. (b) Our algorithm solves the issue in two steps. First, it generates a jig to bring plate *P* into the final shape, absorbing its spring forces. (c) Second, in analogy to Figure 8, it places a piece of sacrificial material across the top. This distributes the impact across all targets, eliminating the mentioned shearing forces, thus keeping the living hinge intact, until (d) the living hinges are inserted.

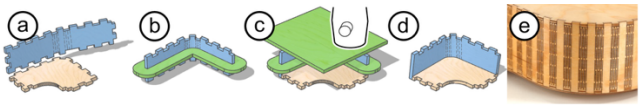


Figure 15: (a) Living hinges face similar issues. We propose (b) a custom jig to control springiness and (c) a sacrificial piece to distribute the impact. (d) Finish. (e) To allow living hinges to handle tensile loads, we alternate living hinges with pins.

Note how the strip of living hinge we inserted has a small rigid region at its center. We propose this pattern, i.e., alternating between living hinges and solid regions, as the latter allows adding pins to rounded geometry;

by making these tight, we obtain (e) rounded geometry capable of taking a tensile load.

This completes our discussion about the first five issues. Trapezoidal scaffolding addresses them by making laser-cut objects a bit more like nails.

6 Issues 6-10: Constrain-before-insertion

The remaining five issues have a different root cause: unlike the joints of traditional "loose" laser-cut 3D models, tight joints cannot be adjusted later without losing tightness or even visibly damaging the model. As basic as this observation sounds, it leads to the two hardest issues from the table of issues, i.e., the ones we marked with a ⚠. Here is how.

In the first step of any assembly process, two plates are combined into a two-part assembly, as illustrated by Figure 16a. This step, however, tends to be "noisy" in that the angle may be off a bit or the joints may not be fully inserted, which we will refer to as incomplete *intrusion*.

This lack of quality control, however, can have significant consequences further down the road, when the resulting two-part assembly will become part of a larger assembly that is intolerant of error.

The most basic of these later assemblies is a simple corner, i.e., the user tries to place a third plate onto a two-part assembly. (b) While a slightly oblique angle is relatively harmless in that it can be corrected after assembly completes, (c) if the angle of the two-part assembly is too acute, then the third plate will not assemble. As alluded to earlier, opening up the two-part assembly by a few degrees is mechanically challenging and may break the two-part assembly.

(d) In this model, the assembly process continues until, eventually, the two-part assembly is later joined with a plate that frames the two-part assembly from the outside to form a *concave joint*. In this case, errors in angle and incomplete insertion become a show-stopper, as either prevents the top plate from being mounted.

Given these dire consequences, users would be well-advised to ensure the step shown in (a) is perfect. This, however, turns out to be harder than it may appear at first glance, as there are no other plates to act as constraints yet. Neither is there much visual indication. And, since at least some of the negative consequences do not manifest themselves until much later, users may not see the incentive to put in the extra effort and double-check.

As a result of these observations, this section is mainly about combining two plates into a two-plate assembly. We will present specialized solutions for several sub-cases and will then focus on the most generalizable solution, which is to search the space of possible assembly sequences and to identify a path that circumvents the issues described above: to *delay* the assembly of all mission-critical two-plate assemblies until other plates have established the necessary constraints (see Figure 17 for a preview). We call this approach *constrain-before-insertion*.

6.1 6/7/8. Assembling Two Plates with the Correct Angle

We now tackle the challenge of assembling two plates, starting with the case from Figure 16a.

As mentioned earlier, several special cases have easier solutions and are thus worth considering upfront. Inserting plates at a 90°-angle, for example, generally tends to work out: Even though kerf is almost always slightly slanted, it is still close enough to 90° to make the back of the matching slot of a box- or T-joint act as a useful stop. Furthermore, humans are very familiar with 90°-angles and can thus judge quickly whether a joint looks correct or requires tweaking.

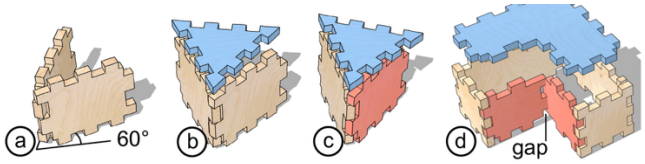


Figure 16: (a) A Sloppy two-part assembly can have consequences: (b) While a slightly too oblique angle can be hammered in later, (c) an overly acute-angle two-part assembly prevents the blue plate from assembling. (d) This two-part assembly is supposed to serve as the core of a concave joint. Here, angle errors and incomplete insertion bring the assembly to a halt, as the blue plate *P* does not fit the pins on the spaced-out red plates.

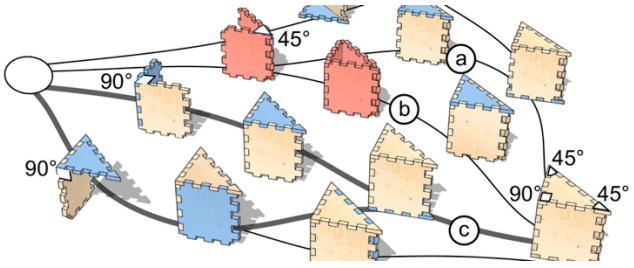


Figure 17: Our algorithm searches the space of possible assembly sequences (here, shown only partially) and avoids paths (a) and (b). Path (c) comes out ahead, delaying the insertion of the non-90° angle until after the assembly has become sufficiently constrained.

Obtaining a non-90° angle, in contrast, is generally more challenging. As illustrated by Figure 17, we therefore pursue the *constrain-before-insertion* approach, i.e., our algorithm searches the space of possible assembly sequences and assesses them. As we describe in additional detail in the Section "Algorithm" below, the algorithm incurs a penalty every time the user assembles an unconstrained non-90°-degree angle. This way, the algorithm gravitates towards path (c), which bypasses the issue.

There are rare "pathological" cases that it cannot resolve. In the example shown in Figure 18a, for example, there is no way of bypassing the issue: since *all* joints are non-90° angles, *one* of them has to go first.

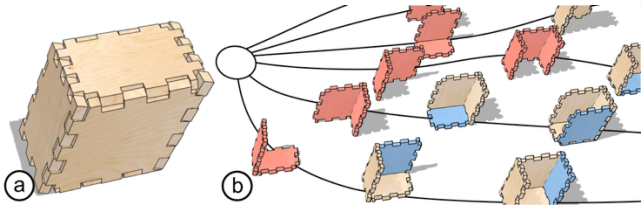


Figure 18: (a) The assembly of a cube *slanted on all sides* (b) makes it impossible to bypass the assembly of two plates at a non-90° angle.

As a fallback, we revert to the tapered pins introduced earlier in Section "3. Multi-Trapezoidal Scaffolding" to help assemble very long joints. Here, tapered pins allow users to position three (or even all four) plates loosely by hand, as aligned pins now "click" into position. This constrains the geometry, without committing to any specific angles yet. As users hold the model in this constellation, all angles are properly constrained and thus ready for being fully inserted (Figure 19).

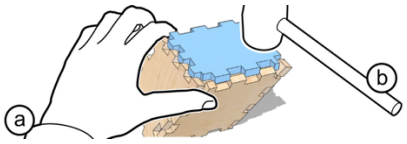


Figure 19: (a) tapered pins allow holding the slanted cube in place long enough until 3+ parts constrain each other, and (b) can be hammered into position.

Tapered pins thereby also address issue number 7 from our list, in that they help users align pins with slots (Figure 20a). (b) Once snapped into place, tapered tips help hold pins in this aligned position, which addresses issue number 8 from that same list. (c) We can amplify this effect by complementing the taper with a *straight* segment.

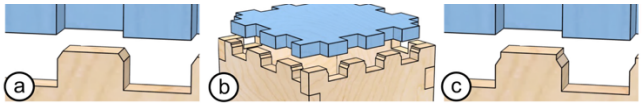


Figure 20: (a) Tapered pins help align joints before insertion, and (b) hold plate P in place, even more so when (c) complemented with an additional straight segment.

Tapered pins, however, come at a price in that they reduce the contact area between joined plates and, thus, the ability to take a tensile load (Figure 21). To minimize the negative effect, we tend to keep tapers small (for 6mm birch, we make them 0.75mm deep and 0.4mm wide), and use tapered pins only on assembly *A*, but not on plate *P*.

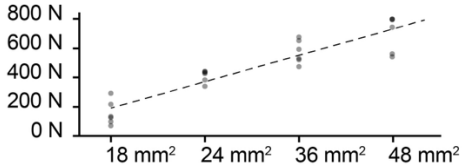


Figure 21: The result of the tensile strength test of the joint with tapers for 6mm birch plywood on a set of three pins and slots. Linear regression gives $y = 30x - 350$, where the y-axis is force (N) and the x-axis is contact area (mm^2). This indicates ~730N for 36 mm^2 (no taper) and ~590N for 31.5 mm^2 (taper we use, 0.75mm deep and 0.4mm wide).

6.2 9. Incomplete Intrusion for "Concave" Joints

As mentioned above, the even more complex issue is *concave* joints, as the resulting problem manifests itself late. This is compounded by the fact that concave joints are susceptible to angle errors and incomplete intrusion.

Again, we begin with a more specific case that allows for a more straightforward solution. The most basic case is a subset of models where no force can be applied to the concave joint: for the model shown in Figure 22, for example, we can imagine attaching

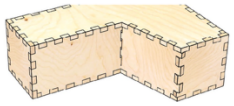


Figure 22: No force applied to this model can ever propagate to the concave joint in its center. We may, therefore, replace the tight concave joint with a loose one.

handles to any of the plates and pulling on them—yet none of them will ever apply tension to the concave joint in its center. We handle this subclass of models simply by replacing the tight joint with a *loose* box joint, which still resists torsion but will fully insert without problems, and the angle of which can be adjusted throughout.

If a concave joint *is* load-bearing but at a 90° angle, we extend it into a T-joint, as illustrated by Figure 23a ([3]). While it takes practice to make the back of a box joint act as a jig, T-joints achieve such alignment reliably. Users then achieve intrusion simply by hammering *P* all the way in. (b-c) The rest of the model is convex, making the remaining assembly easy: users quickly assemble the model into a rough shape (d) *and* hammer joints all the way in.

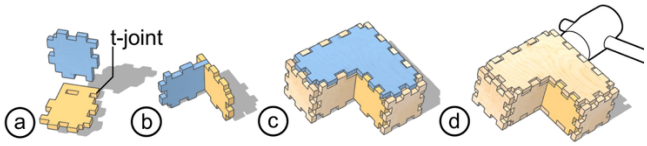


Figure 23: (a) Extending the box joints of the golden plate into a T-joint makes complete intrusion and a 90° angle easy. (b) In this orientation, (c) users complete the rest of the assembly and insert the top and bottom plates. (d) All other plates are convex, which allows fixing intrusion late.

When angles are non-90° (or for models in which the interior must stay clear, such as containers), T-joints are not suitable. As illustrated by Figure 24a, our fallback is to provide the concave joint with some slack. We achieve this by deepening the slots on assembly *A* by a small amount (we obtained good results with values as small as 0.5mm). (b) The resulting concave joint performs successfully when fully inserted and anywhere within that extra range. This allows users to complete the assembly and (c) fix potential misalignment in the concave joints at the very end by

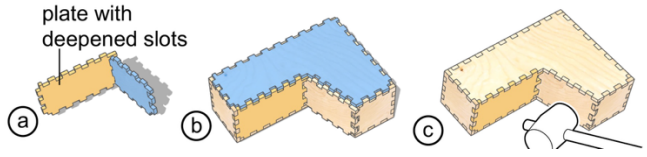


Figure 24: (a) Deepening the slot on the golden plate makes it easy for users to reach or exceed the necessary intrusion by the golden plate, (b) allowing top and bottom plates to assemble. (c) Once complete, users may hammer the golden plate *again* to fully insert it when all other assembly has been completed.

hammering the plate into its final position (similar to how convex objects allowed for this in Figure 23b-c).

However, once again, there is a small set of pathological cases where neither approach is applicable. Figure 25a shows such a model: a combination of four plates (one of them not visible as it is facing away from the camera) that are mutually interlocking in two concave joints. No sequence brings all four plates into position by means of *translating* them. Instead, (b) at least one of the four plates has to be *rotated* into position.

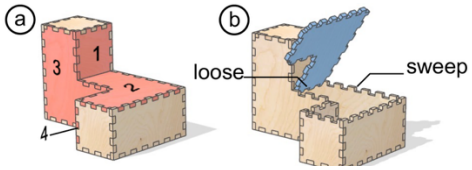


Figure 25: (a) The four red plates are interlocking around two concave joints so that (b) at least one of them has to be *rotated* into position.

While this style of inserting a plate arguably is somewhat non-traditional, it does delay the assembly of the concave joint(s) until the rest of the model is in place, thereby eliminating the risk of angle errors and the propagation thereof: constrain-before-insertion.

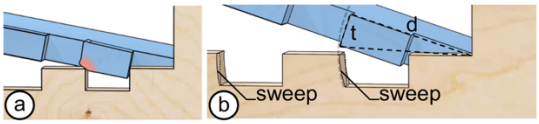


Figure 26: (a) Rotary insertion causes the pins of plate P to collide with the associated pins on assembly A. (b) We address this by carving the sweep of the pins on P out of A.

To allow for such *rotary insertion*, the box joints need some extra preparation—an adapted version of tapered pins. As illustrated by Figure 26a, the rotary movement causes the corners of the pins on plate *P* to collide with the pins on assembly *A*. (b) To address this, we compute the circular sweep of the pins on P (with radius sqrt(t^2 + d^2)) and carve this sweep out of the pins on assembly *A*. The adjustment is largest for the pin closest to the fulcrum, which then diminishes.

6.3 10. Stability

Looking at the last issue on our list, we see that striking a plate not only subjects plate *P* to this impact but also assembly *A*. Unless all plates in *A* already form a rigid structure, this may cause assembly A to deform or even disassemble, as illustrated by Figure 27a.

(b) One approach to handling unstable assemblies is to hold the assembly *A* by the plate that *P* will be inserted into, e.g., here using a vice. (c) However,

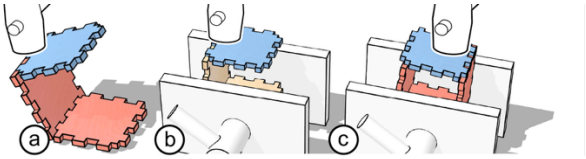


Figure 27: (a) An unstable assembly can be (b) fixated using a vise, (c) but when plate P needs to be inserted into *multiple* plates, acute angles in assembly *A* can prevent assembly.

again, there is a pathological case, i.e., when *P* needs to be inserted into *multiple* joints at once, as acute angles in assembly *A* can prevent assembly.

We again go for constrain-before-insertion. As shown in Figure 28, our algorithm searches the space of assembly sequences. It recognizes the configurations shown in red as unstable and therefore avoids paths a and b, and instead chooses path c, which makes it assemble the cube by assembling three plates into a corner first, making it sturdy early on.

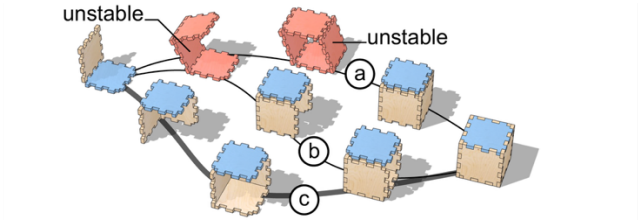


Figure 28: Our algorithm avoids paths (a) and (b) as they pass through unstable assemblies and picks path (c) instead.

6.4 Algorithm: Searching the Space of Assembly Sequences

In the previous sections, we repeatedly mentioned searching the space of assembly orders. Here is the algorithm:

Step 1: Create a lattice structure containing all sensible assembly sequences. We start with the traditional *directional blocking graphs* [31] and the *assembly-by-disassembly* algorithm [14][28]. These have traditionally been used to pick a *single* next step, which allowed these algorithms to run in $O(n^2)$ [28]. Picking all possible sequences runs the risk of running into factorial complexity. We address this using randomized *branch-and-bound*, i.e., we perform a single pass of a greedy algorithm to determine an upper bound of our cost function. Then, we repeat random walks that we terminate if that upper bound is met before assembling the model.

Step 2: For each step in every assembly sequence, consider all possible orientations of the plate and assembly (Figure 29).

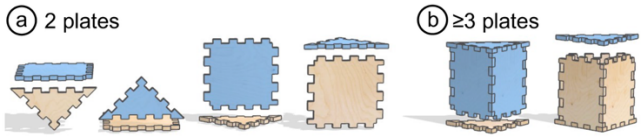


Figure 29: (a) There are generally four possible orientations for inserting a plate *P* into another single plate assembly, and (b) two, when the assembly already consists of two or more plates.

Step 3: For each step of every assembly sequence, (a) Identify which of the 10 issues this step is subject to. (b) Consider all ways of overcoming the issue discussed above. Add the additional user effort resulting from the re-engineering to the difficulty of this path.

Step 4: Compute the *difficulty* for each assembly sequence as the sum of the difficulties of each step. Difficulty is a negative number, i.e., a *penalty* function (a special *utility function*). The penalty should be in the following order: interlock > the situation that violates.

Step 5: Pick the assembly sequence that minimizes difficulty using Dijkstra's algorithm, which is inherently $O((v+e)\ log(v))$, with e being the number of edges in the assembly sequence lattice.

Step 6: Modify specific box joints with tapers or t-joints according to the assembly sequence.

7 User Study

To validate our claim that *trapezoidal scaffolding* and *constrain-before-insertion* help assemble objects with tight joints, we conducted a user study in two parts. We hypothesized that participants would assemble models faster, more easily, and with less damage when those models were processed using our algorithm.

7.1 Part 1: Trapezoidal Scaffolding

Participants assembled three simplified models with tight joints, and a second set of these models was processed by our algorithm, in counter-balanced order.

Tasks: In each of the three tasks, participants assembled one of the models shown in Figure 30. They were designed to test the usefulness of trapezoidal scaffolding for *Transmission* (Figure 9), *Small Target* (Figure 7), and *Sturdy* (Figure 12). We used 6mm birch plywood, and all joints were tuned for maximum tightness (200N / pin). Participants were provided with a 400g mallet.

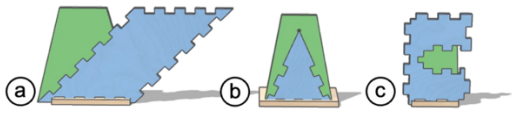


Figure 30: In each task, participants assembled one of the three simplified models shown here.

Interface conditions: There were two interface conditions. In the *trapezoidal scaffolding* condition, participants assembled the models *with* the green scaffolding shown in Figure 30. In the *control* condition, they assembled the model without.

Each participant assembled each model in each interface condition (within-subject design) in counterbalanced order.

Participants: We recruited 12 participants (8 male, 4 female, average age = 23 years, SD = 5.2) from our institution. 6 participants had no prior experience assembling objects, e.g., furniture, using a mallet, and 2 participants had prior experience assembling laser-cut objects using a mallet.

Training: Before the first trial, participants viewed a 2-minute training video that showed how to assemble parts using a mallet. The video also explained the expected precision, i.e., an intrusion error of no more than 0.2mm. Finally, participants assembled three pairs of 6cm x 6cm plates with tight joints using a mallet.

Procedure: For the trial, the experimenter measured the time between the first and last hits. If the expected precision was not met, the experimenter had participants refine the result and resume time measurement. Participants then filled in a questionnaire.

7.2 Part 2: Constrain-Before-Insertion

In analogy to Part 1, participants assembled the model shown in Figure 31. Unlike in Part 1, we combined two issues into a single 5-part model resembling one leg of the chair in Figure 1. This had *Angle* (Figure 17) and *Intrusion* (Figure 16) issues.

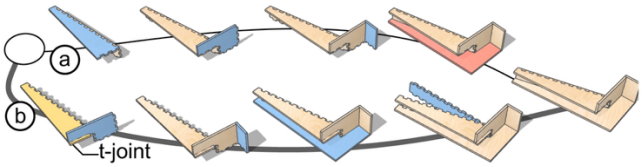


Figure 31: In part 2, participants assembled a chair leg, either (a) using an assembly order handling the concave joint early (control) or (b) late, using rotary insertion (experimental).

Task, training, interface condition, and procedure: as in Part 1.

Participants: We recruited a fresh set of 12 participants (8 male, 3 female, 1 preferred not to say, average age = 22.5 years, SD = 3.4) from our institution. 6 participants had no prior experience assembling objects, e.g., furniture, using a mallet.

7.3 Results

As expected, the participants were faster in assembling the three models in the experimental condition (Figure 32a).

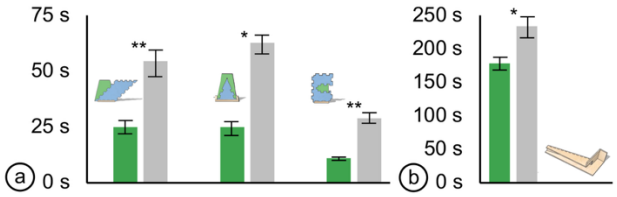


Figure 32: (a) Participants assembled models on average 2.2-2.7 times faster when provided with trapezoidal scaffolding and (b) 1.4 times faster when assembling the chair leg in constrain-before-insertion order. In the control condition, two participants could not complete assembling. (Error bars indicate standard error.)

Part 1: Participants spent more time assembling the model without trapezoidal scaffolding—54 seconds, 63 seconds, and 29 seconds, on average—compared to an average of 25 seconds, 25 seconds, and 11 seconds for the models with scaffolding. The differences were significant ($t(11) = 2.835$, $p < 0.01$, $d = 2.48$), ($t(11) = 2.711$, $p < 0.05$, $d = 4.2$), and ($t(11) = 3.356$, $p < 0.01$, $d = 4.3$). The effect was substantial: participants were, on average, 2.4 times faster. In the control conditions, 10 out of 12 participants broke the model shown in Figure 30c, and one other broke the model shown in Figure 30b. In the *scaffolding* condition, there was no damage to the models.

Part 2: Participants spent more time assembling the model in the control condition—237 seconds on average—compared to an average of 174 seconds in the experimental condition. The differences were significant ($t(11) = 2.515$, $p < 0.05$, $d = 0.9$). Participants

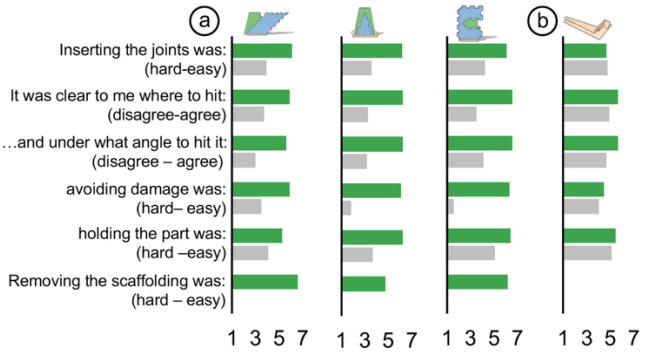


Figure 33: Participants rated the experimental conditions in the questionnaire higher along all dimensions.

were, on average, 1.4 times faster. In the control condition, 2 participants could not complete due to incomplete insertion, bringing the assembly to a halt.

As shown in Figure 33, participants rated the experimental conditions in the questionnaire higher along all dimensions.

Several participants mentioned that the trapezoidal scaffolding was clearly helpful. P10 said, "It was easier because it (trapezoidal scaffolding) provides a flat surface, not like the angled one (control condition)." P8 said, "Analyzing where to hit was mentally hard (in the control condition) but not in (the experimental condition)." A few participants also noted that scaffolding allowed them to hold plates easily, with P10 saying, "Because the attachment (trapezoidal scaffolding) surrounded the entire object, it was clear to hold and where to hit—there was simply no other choice."

For the constrain-before-insertion task, P13 experienced exactly what we highlighted in this paper, saying, "I made a mistake, and they weren't fitting together anymore (in the control condition). The one (in the experimental condition) is much easier."

7.4 Discussion

The results of our study support our hypothesis that our algorithms and techniques allow for faster and easier assembly of laser-cut models featuring tight joints.

8 Conclusion and Future Work

In this paper, we explored problems resulting from making laser-cut 3D models robust against tension by making their joints tight. We identified the 10 underlying issues causing models either to break or assembly to fail. We presented two main strategies for overcoming them, i.e., either by extending parts with what we call *scaffolding* or by adjusting the models' assembly order so as to bypass states that are subject to these issues. We thereby contribute to helping users of laser cutters transition from decorative objects to *load-bearing* objects.

As future work, we are planning to extend the concepts developed in this paper to other forces applying to load-bearing objects.

Acknowledgments

We thank Zouela Nemitz for her help with fabricating laser-cut models.

References

[1] Muhammad Abdullah, Romeo Sommerfeld, Laurenz Seidel, Jonas Noack, Ran Zhang, Thijs Roumen, and Patrick Baudisch. 2021. Roadkill: Nesting Laser-Cut Objects for Fast Assembly. In *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 972–984. DOI: https://doi.org/10.1145/3472749.3474799

[2] 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 (UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 10, 1–13. DOI: https://doi.org/10.1145/3526113.3545618

[3] Muhammad Abdullah, Martin Taraz, Yannis Kommana, Shohei Katakura, Robert Kovacs, Jotaro Shigeyama, Thijs Roumen, and Patrick Baudisch. 2021. FastForce: Real-Time Reinforcement of Laser-Cut Structures. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 673, 1–12. DOI: https://doi.org/10.1145/3411764.3445466

[4] BASF. Snap-Fit Design manual, BASFCorporation Engineering Plastics, USA, 2007 http://www8.basf.us//PLASTICSWEB/displayanyfile?i d=0901a5e1801499d5, last accessed April 2025.

[5] 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 (CHI '19)*. ACM, New York, NY, USA, Paper 566, 12 pages. DOI: https://doi.org/10.1145/3290605.3300796

[6] Geoffrey Boothroyd. Assembly automation and product design. CRC press, 2005.

[7] Ruei-Che Chang, Chih-An Tsao, Fang-Ying Liao, Seraphina Yong, Tom Yeh, and Bing-Yu Chen. 2021. Daedalus in the Dark: Designing for Non-Visual Accessible Construction of Laser-Cut Architecture. In *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 344–358. DOI: https://doi.org/10.1145/3472749.3474754

[8] CNC Panel Joinery Notebook. https://makezine.com/article/digital-fabrication/3d-printing-workshop/cnc-panel-joinery-notebook/, last accessed April 2025.

[9] Cuttle. https://cuttle.xyz, last accessed April 2025.

[10] Dead Blow Hammer. https://ncwoodworker.net/forums/index.php?threads/dead-blow-hammer.79108/, last accessed April 2025.em

[11] Chiao Fang, Vivian Hsinyueh Chan, and Lung-Pan Cheng. 2022. Flaticulation: Laser Cutting Joints with Articulated Angles. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29-November 2, 2022, Bend, OR, USA*. ACM, New York, NY, USA 16 Pages. https://doi.org/10.1145/3526113.354569

[12] James M. Gere and Stephen P. Timoshenko. Chapter 4, *Mechanics of materials*. CL Engineering, 1996.

[13] Neil Gershenfeld, Patrik Kunzler, and George Alex Popescu. Digital assembler for digital materials. U.S. Patent 7,848,838, issued December 7, 2010.

[14] Dan Halperin, Jean-Claude Latombe, and Randall H. Wilson. 1998. A general framework for assembly planning: the motion space approach. In *Proceedings of the fourteenth annual symposium on Computational geometry (SCG '98)*. Association for Computing Machinery, New York, NY, USA, 9–18. DOI: https://doi.org/10.1145/276884.276886

[15] Florian Heller, Jan Thar, Dennis Lewandowski, Mirko Hartmann, Pierre Schoonbrood, Sophy Stönner, Simon Voelker, and Jan Borchers. Cutcad-an open-source tool to design 3d objects in 2d. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 1135-1139. 2018. DOI: https://doi.org/10.1145/3196709.3196800

[16] Shohei Katakura, Martin Taraz, Muhammad Abdullah, Paul Methfessel, Lukas Rambold, Robert Kovacs, and Patrick Baudisch. 2023. Kerfmeter: Automatic Kerf Calibration for Laser Cutting. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 691, 1–13. DOI: https://doi.org/10.1145/3544548.3580914

[17] Robert Kovacs, Lukas Rambold, Lukas Fritzsche, Dominik Meier, Jotaro Shigeyama, Shohei Katakura, Ran Zhang, and Patrick Baudisch. 2021. Trusscillator: a System for Fabricating Human-Scale Human-Powered Oscillating Devices. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 1074–1088. DOI: https://doi.org/10.1145/3472749.3474807

[18] Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. 2020. Tsugite: Interactive Design and Fabrication of Wood Joints. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 317–327. DOI: https://doi.org/10.1145/3379337.3415899

[19] Danny Leen, Nadya Peek, and Raf Ramakers. 2020. LamiFold: Fabricating Objects with Integrated Mechanisms Using a Laser cutter Lamination Workflow. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 304–316. DOI: https://doi.org/10.1145/3379337.3415885

[20] Danny Leen, Tom Veuskens, Kris Luyten, and Raf Ramakers. 2019. JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Paper 156, 1–12. DOI: https://doi.org/10.1145/3290605.3300386

[21] Mallet. https://en.wikipedia.org/wiki/Mallet, last accessed April 2025.

[22] James McCrae, Nobuyuki Umetani, and Karan Singh. 2014. FlatFitFab: interactive modeling with planar sections. *In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14)*. ACM, New York, NY, USA, 13-22. DOI: https://doi.org/10.1145/2642918.2647388

[23] Ponal Joint Glue https://www.ponal.de/downloads/technische-datenblaetter Accessed April 2025

[24] Ramsthaler F, Kettner M, Potente S, Verhoff MA, Seibert H, Reis M, Diebels S, Roland M. Hammer blows to the head. *Forensic science international*. 2019 Aug 1;301:358-70.

[25] Thijs Roumen, Ingo Apel, Thomas Kern, Martin Taraz, Ritesh Sharma, Ole Schlueter, Jeffrey Johnson, Dominik Meier, Conrad Lempert, and Patrick Baudisch. 2022. Structure-Preserving Editing of Plates and Volumes for Laser Cutting. In *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication (SCF '22)*. Association for Computing Machinery, New York, NY, USA, Article 10, 1–12. DOI: https://doi.org/10.1145/3559400.3561996

[26] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2010. SketchChair: an all-in-one chair design system for end users. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (TEI '11)*. Association for Computing Machinery, New York, NY, USA, 73–80. DOI: https://doi.org/10.1145/1935701.1935717sch

[27] Rundong Tian and Eric Paulos. 2021. Adroid: Augmenting Hands-on Making with a Collaborative Robot. In *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 270–281. DOI: https://doi.org/10.1145/3472749.3474749tiany

[28] Yunsheng Tian, Jie Xu, Yichen Li, Jieliang Luo, Shinjiro Sueda, Hui Li, Karl D. D. Willis, and Wojciech Matusik. 2022. Assemble Them All: Physics-Based Planning for Generalizable Assembly by Disassembly. *ACM Trans. Graph*. 41, 6, Article 278, 11 pages. DOI: https://doi.org/10.1145/3550454.3555525

[29] Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. 2012. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.* 31, 4, Article 86 (July 2012), 11 pages. DOI: https://doi.org/10.1145/2185520.2185582

[30] Ziqi Wang, Florian Kennel-Maushart, Yijiang Huang, Bernhard Thomaszewski, and Stelian Coros. 2023. A Temporal Coherent Topology Optimization Approach for Assembly Planning of Bespoke Frame Structures. *ACM Trans. Graph.* 42, 4, Article 144 (August 2023), 13 pages. DOI: https://doi.org/10.1145/3592102

[31] Randall H. Wilson. On Geometric Assembly Planning. Stanford University Dept of Computer Science, 1992.

[32] Clement Zheng, Ellen Yi-Luen Do, and Jim Budd. 2017. Joinery: Parametric Joint Generation for Laser Cut Assemblies. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition (C&C '17)*. Association for Computing Machinery, New York, NY, USA, 63–74. DOI: https://doi.org/10.1145/3059454.3059459