Text to Robotic Assembly of Multi Component Objects using 3D Generative AI and Vision Language Models

Alexander Htet Kyaw
Massachusetts Institute of Technology (MIT)
alexkyaw@mit.edu

Richa Gupta
MIT
richag@mit.edu

Dhruv Shah
Google DeepMind
dhruvshah@google.com

Anoop Sinha
Google, Paradigms of Intelligence
anoopsinha@google.com

Kory Mathewson
Google DeepMind
korymath@google.com

Stefanie Pender
Autodesk Research
stefanie.pender@autodesk.com

Sachin Chitta
Autodesk Research
sachin.chitta@autodesk.com

Yotto Koga
Autodesk Research
yotto.koga@autodesk.com

Faez Ahmed
MIT Mechanical Engineering
faez@mit.edu

Lawrence Sass
MIT Architecture
lsass@mit.edu

Randall Davis
MIT CSAIL
davis@csail.mit.edu

Abstract

Advances in 3D generative AI have enabled the creation of physical objects from text prompts, but challenges remain in creating objects involving multiple component types. We present a pipeline that integrates 3D generative AI with vision-language models (VLMs) to enable the robotic assembly of multi-component objects from natural language. Our method leverages VLMs for zero-shot, multi-modal reasoning about geometry and functionality to decompose AI-generated meshes into multi-component 3D models using predefined structural and panel components. We demonstrate that a VLM is capable of determining which mesh regions need panel components in addition to structural components, based on object functionality. Evaluation across test objects shows that users preferred the VLM-generated assignments 90.6% of the time, compared to 59.4% for rule-based and 2.5% for random assignment. Lastly, the system allows users to refine component assignments through conversational feedback, enabling greater human control and agency in making physical objects with generative AI and robotics.

1 Introduction

Recent developments in 3D Generative AI have enabled users to create a wide variety of geometries from natural language input [12, 7, 24]. Extending this capability to make physical objects from a text prompt could empower users who don't have technical expertise in complex 3D design software or manufacturing processes. While previous research has explored physical making with 3D generative AI, challenges remain in creating objects composed of multiple component types with different functionalities [5, 6, 4]. Robotic assembly offers a potential approach to create physical objects from multiple component types, while supporting modularity, reuse and part-level editing [8, 11, 10]. However, most 3D generative AI models produces monolithic meshes that lack the component-level representation required for robotic assembly path planning and assembly sequencing generation.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Creative AI Track.

Decomposing AI-generated meshes into predefined components is challenging becuase assignments can depend on the geometry and functionality of both the object and its parts. In this study, we use two component types: structural and panel components (Figure 1). While structural components are assigned to the object's base geometry to ensure stability, the main focus is on determining the placement of panel components based on the object's geometry and intended function. For example, a stool may require horizontal panels on the seat to create a flat surface for sitting, while a lamp may need panels on its lampshade frame to diffuse light. In addition to geometry and functionality, panel assignments can also vary according to user preferences, highlighting the need for user feedback to support human-AI co-creation for physical making. To address these challenges and enable text-driven, multi-component robotic assembly with generative AI, we present:

- A function and geometry-aware approach that uses VLM-based multimodal reasoning to assign panel components and decompose AI-generated meshes into multi-component 3D models.

- A conversational feedback workflow using VLMs to enable users to adjust component assignments and provide human control in the AI-driven robotic assembly process without task-specific training.

- An end-to-end framework connecting natural language input, 3D generative AI, VLM, and robotic assembly with predefined components, to create multi-component physical objects from text prompts.

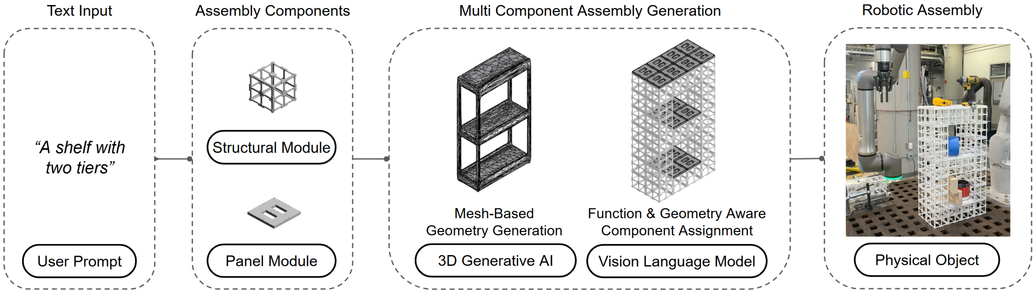


Figure 1: From text input to multi-component robotic assembly using predetermined components

2 Related Work

3D Generative AI for Prompt-Based Design and Fabrication. Recent advances in text-to-3D models such as DreamFusion [19], Get3D [7], and Latte3D [23] have enabled users to generate a wide array of geometries from natural language prompts. To physically realize these models, prior work has mainly explored 3D printing pipelines [4, 22]. Systems such as Sketch2Prototype transform sketches into printable designs for additive manufacturing [5]. Style2Fab extends this by allowing users to modify and customize the stylistic attributes of 3D models while preserving printability [6]. However, these approaches are for 3D printing with generative AI output that lacks the component-level representation necessary for robotic assembly.

Part Aware Generation of 3D Objects. Part-aware generative models have enabled structured 3D shape synthesis through component-level reasoning. For example, PartGen [3] develops a diffusion-based pipeline that reconstructs semantically meaningful parts from text inputs. StructureNet [16] introduces graph-based part hierarchies geometry synthesis. ShapeAssembly [9] generates objects using a programmatic part layout optimized for visual coherence. While these contributions advance part-level generative modeling, they mostly aim to reconstruct coherent shapes or to support part-level geometry editing. In contrast, our goal is to leverage part-level reasoning to inform the assembly geometry from a predefined set of component type.

Component Segmentation of 3D Objects. Previous approaches to decomposing 3D shapes into components have relied on supervised methods such as hierarchical recursive networks, point-based segmentation, and semantic graphs [27, 2, 26]. More recent unsupervised and generative methods, such as Neural Parts [18], auto-decoder frameworks for 3D diffusion models [17], and joint-aware techniques [13], address earlier limitations by introducing flexibility and connection constraints for multi-part reconstruction. However, while these methods improve geometric decomposition, they do not consider robotic assembly or the functional roles of components.

Vision-Language Models for Prompt Based Robotic Assembly. VLMs have previously been used to ground natural language instructions for assembly tasks. For example, CLIPort [20] couples CLIP-based perception for pick-and-place tasks. SayCan [1] integrates language models with affordance scoring to execute multi-step instructions. StructDiffusion [14] use diffusion and transformer architectures with multimodal input for compositional tasks. VLMs have also been explored for assembly sequence generation. Zhu et al. [28] introduces a seq2seq transformer that infers part assembly sequences. Neural Assembler [25] convert multi-view imagery of block-based models into step-wise assembly plans. These systems demonstrate the potential of VLMs from robotic manipulation to assembly sequence generation. In this paper, we extend the use of VLMs for generating assembly geometry by assigning component types based on object functionality.

3 Methods

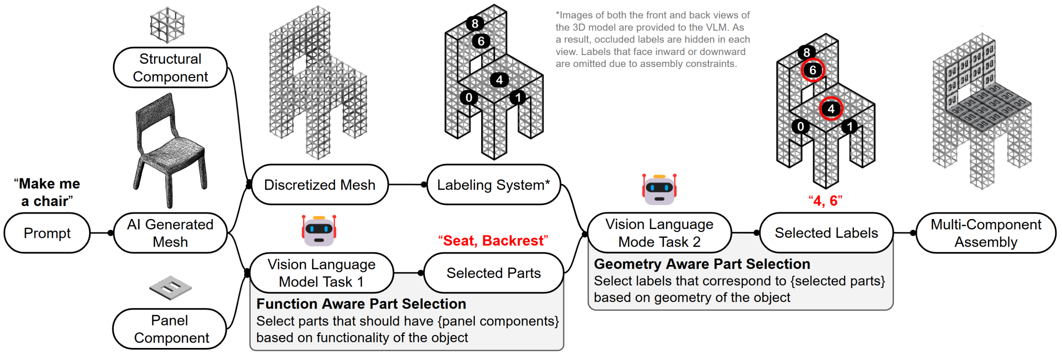
Text-to-Mesh Generation. We present a pipeline using VLM and generative AI to translate natural language inputs into multi-component 3D models for robotic assembly. The system begins with a user prompt, which is used to generate a mesh using Autodesk's 3D generative AI model [15].

Mesh Discretization. We define two classes of assembly components: (1) structural components, which form the primary load-bearing frame, and (2) panel components, which attach to the structural frame to provide functional surfaces. To create primary load-bearing frame, the AI-generated mesh is discretized into structural components. This is done by voxelizing the mesh using a fixed grid resolution based on the dimensions of the structural components. The placement of panel components depend on the functionality and geometry of the object. Attaching panels indiscriminately can add unnecessary weight and waste components. To address this, we developed a VLM task to selectively assign faces that should have panel components based on the functionality of the object.

VLM for Function Aware Part Selection. The VLM is tasked to understand the object's functionality and geometry to identify the parts requiring a predefined component. In this study we use Google's Gemini 2.5 pro model. The VLM gives a response based on the three inputs: the description of the object from the user (to understand the object's functionality), an axonometric image of the AI generated mesh (to understand the object's geometry), and the component type - in this case the type is panel component (to understand the component's functionality). Conditioned on these inputs, the VLM is tasked to reason over both functionality and geometry to determine which parts require the component. For example, given the prompt "I want a chair" and the image of the AI-generated mesh, the VLM returns for panel component "Parts = seat, backrest" as shown in Figure 2.

System Prompt: You are an assistant that selects the functional parts of an object that require a specified component type. Use: (1) the description of the object, (2) an axonometric image of the AI-generated mesh, and (3) the component type. Select the minimal set of parts that fulfill the object's functionality. Output only the part names as specified, no explanations.

Query: Given an image of {user text prompt}, identify which parts of the object should have panel component based on the object's intended functionality. Select only the minimal number of distinct parts required. Output format: Parts = []


Structural Component
"Make me a chair"
Prompt
AI Generated Mesh
Panel Component
Discretized Mesh
Labeling System*
8
6
4
0
1
Vision Language Model Task 1
"Seat, Backrest"
Selected Parts
Function Aware Part Selection
Select parts that should have {panel components} based on functionality of the object
Vision Language Mode Task 2
Geometry Aware Part Selection
Select labels that correspond to {selected parts} based on geometry of the object
"4, 6"
Selected Labels
Multi-Component Assembly

*Images of both the front and back views of the 3D model are provided to the VLM. As a result, occluded labels are hidden in each view. Labels that face inward or downward are omitted due to assembly constraints.

Figure 2: System Pipeline: Vision Language Model for Function and Geometry Aware Part Selection

VLM for Geometry Aware Part Selection. The previous VLM task provides a list of parts within the object that should have a panel component. However, the robot must also know where those parts are physically located within the 3S assembly. To map these parts back to the AI-generated geometry, we merge coplanar faces in the discretized mesh and assign each mesh face a unique integer label for the VLM to reference. Labels for inward-facing vertical and downward-facing horizontal faces are omitted, as they might not be reachable by the robot arm. This prevents the model from assigning panels to places the robot can't access. For the second VLM task, the model receives three inputs: the description of the object from the user, an axonometric rendering of the labeled mesh faces showing both sides of the geometry, and the part list from the first VLM task. Conditioned on these inputs, the VLM matches each part in the list to the corresponding face label. For the example, given the list of parts from the first VLM task "Parts = seat, backrest" and the image of the labeled mesh, the VLM returns "Labels = 4, 6" (Figure 3). The panels are mapped to the labels in the 3D model to create the multi component assembly.

System Prompt: You are an assistant that maps functional parts of an object to their face labels in a labeled axonometric mesh. Use: (1) the description of the object, (2) an image of a labeled mesh, and (3) a list of parts. Select the minimal set of labels that correspond to the listed parts. Output only the label numbers, no explanations.

Query: Given a labeled image of {prompt}, select the label numbers that exactly correspond to the following parts: {parts}. Select only the minimal set of labels needed. Output format: Labels = []

VLM for Human in the Loop Conversational User Alignment. After the initial component assignment, users can provide feedback to refine or override the VLM generated results. Because human preferences can vary, the system leverages user input instead of relying solely on the VLM. In this task, the user feedback serves as additional context to match the user's intent. The VLM processes three inputs: the description of the object, an axonometric rendering of the labeled mesh faces showing both sides of the geometry, and the user feedback (Figure 3). Conditioned on these inputs, VLM outputs the labels corresponding to the user's feedback. The updated labels are mapped to the mesh to regenerate the multi-component assembly. This human-in-the-loop approach leverages conversational feedback to adjust and control assembly outcomes.

System Prompt: You are an assistant that updates component assignments based on user request. Use: (1) the description of the object, (2) the labeled mesh image, (3) the user request. Select the minimal set of labels that fulfill the user request. Output only the label numbers, no explanations.

Query: Given a labeled image of {user prompt}, select the label numbers that match the following request: {user feedback}. Select only the minimal set of labels needed. Output format: Labels = []

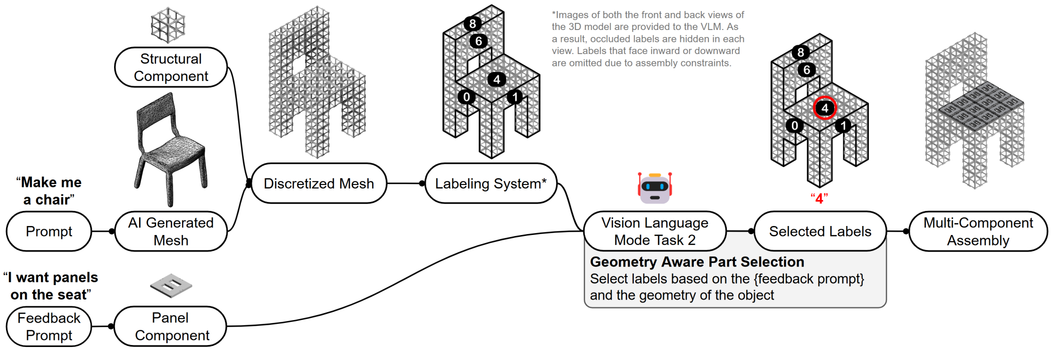

Structural
Component
"Make me
a chair"
Prompt
AI Generated
Mesh
Discretized Mesh
Labeling System*
8
6
4
0
1
*Images of both the front and back views of the 3D model are provided to the VLM. As a result, occluded labels are hidden in each view. Labels that face inward or downward are omitted due to assembly constraints.
"4"
Vision Language
Mode Task 2
Selected Labels
Multi-Component
Assembly
Geometry Aware Part Selection
Select labels based on the {feedback prompt}
and the geometry of the object
"I want panels
on the seat"
Feedback
Prompt
Panel
Component

Figure 3: User Alignment: Integrating human feedback with geometry-aware VLM part assignment

Robotic Assembly Once the multi-component assembly is generated by the VLM, a UR20 robotic arm equipped with Robotiq grippers assemblies the physical geometry. The multi-component assembly from the VLM is exported as two lists: a coordinate list $C = \{(x_i, y_i, z_i, r_{x_i}, r_{y_i}, r_{z_i})\}$, and a component type list $T = \{t_0, t_1\}$. In our implementation, the type $t_i = 0$ denotes a structural component, which is picked from source $s_i = 0$ (a conveyor belt with structural components), while $t_i = 1$ denotes a panel component, which is picked from source $s_i = 1$ (a stack of panel components). The coordinate list is sorted in a bottom-to-top sequence while preserving connectivity to ensure that physically connected components are placed consecutively.

For each component i in C, the robot moves to the source coordinate s_i based on the component type t_i, picks the component, moves to the component coordinate c_i, and place the component (Figure. 4. See algorithm 1 in appendix 4

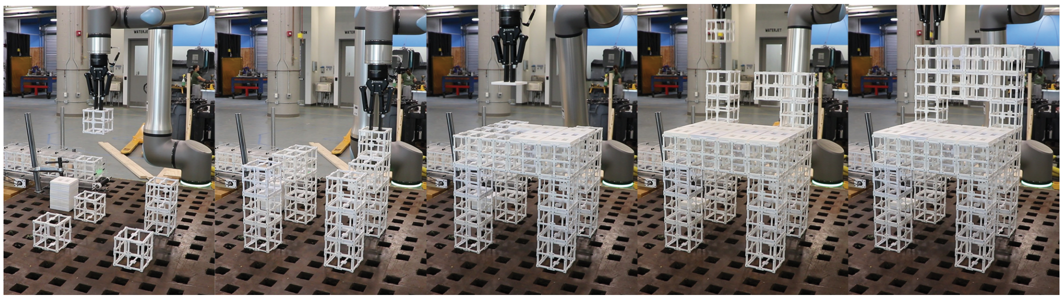


Figure 4: Text to multi-component robotic assembly of the user prompt: "Make me a chair", with the user feedback: "I want panels on the seat".

4 Experiments and Results

Experiment Setup. To assess the effectiveness of the Vision-Language Model (VLM) approach, we conducted a comparative analysis against a *rule-based* approach and a *random* approach. In the rule-based approach, panels were assigned to all upward-facing surfaces, under the assumption that horizontal surfaces are the most likely functional user case for panel components. For the random approach, panels were assigned by randomly selecting labels from the set of mesh faces (Figure 5).

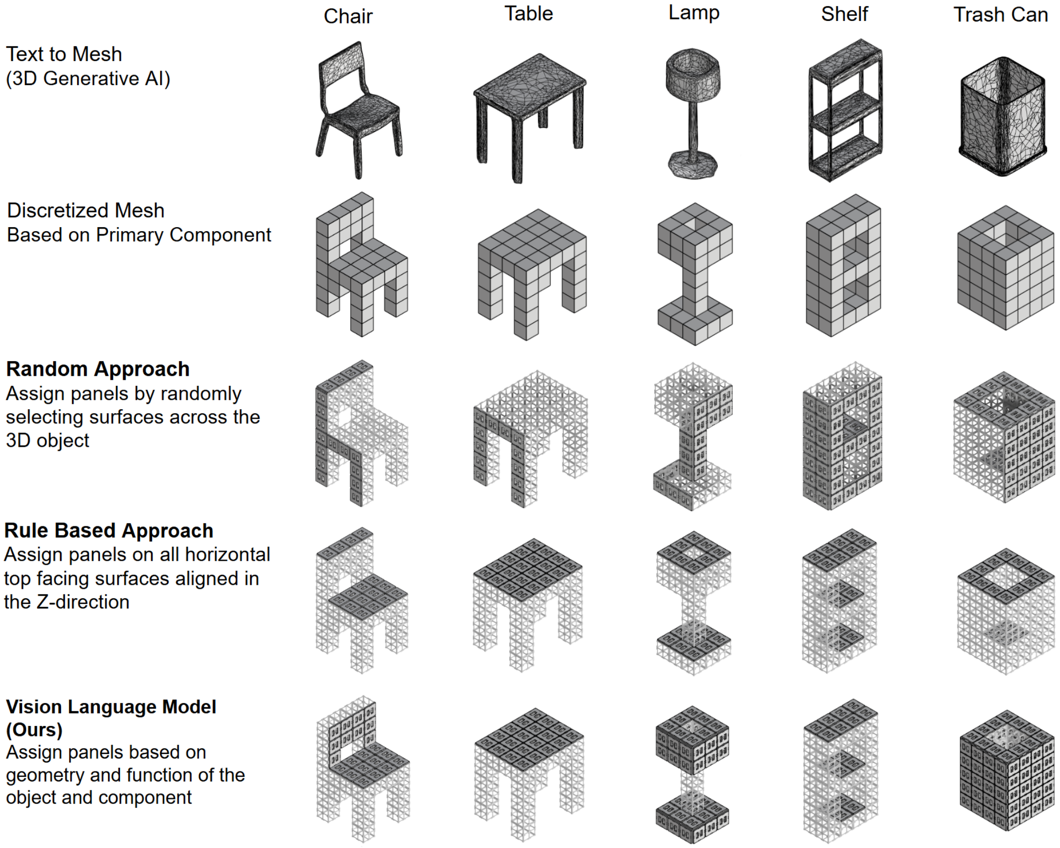

Chair
Table
Lamp
Shelf
Trash Can
Text to Mesh
(3D Generative AI)
Discretized Mesh
Based on Primary Component
Random Approach
Assign panels by randomly selecting surfaces across the 3D object
Rule Based Approach
Assign panels on all horizontal top facing surfaces aligned in the Z-direction
Vision Language Model (Ours)
Assign panels based on geometry and function of the object and component

Figure 5: Multi-component assemblies of five different objects created using three different approaches: random, rule-based, and vision-language models

We recruited thirty two participants to evaluate component assignments on five objects across three approaches, resulting in 480 judgments. Participants were asked to select all alternatives they considered appropriate or acceptable for panel placement based on the object's function. We assume that multiple valid assignments may exist based on user preference rather than a single ground truth.

To avoid forced-choice bias, participants were allowed to select multiple options or none for an object. For each object–method pair, we compute the selection rate as:

$$\text{Rate}_{o,m} = \frac{\text{\# participants selecting method } m \text{ on object } o}{32} \times 100\%.$$

Results. 90.6 % of users agreed with the VLM generated panel assignments. However, only 59.4 % of the user agreed with the rule-based panel assignments. The rule-based approach performs as well as the VLM approach on objects with predominantly horizontal surfaces, such as the table and shelf, but failing on more complex objects like the chair, lamp, and trash can. Unsurprisingly, the random assignment approach performed worst, with a mean selection rate of just 2.5 %.

Additionally, to compare conditions under non-exclusive selection, we applied pairwise McNemar tests, which evaluate responses based on discordant pairs. In this test a higher χ^2 indicates a larger imbalance, and thus a stronger preference for one method. The VLM-based approach was chosen significantly more often than both the rule-based approach ($\chi^2 = 38.11$) and the random assignment ($\chi^2 = 137.11$). All pairwise differences remain significant after Bonferroni correction ($^*p < 0.017$). These findings demonstrate that, even without forced-choice constraints, participants preferred VLM-generated placements. (See Appendix C for details.)

Table 1: Percentage and number of times a user selected a method. Evaluated by 32 participants

Method	**Chair**	**Table**	**Lamp**	**Shelf**	**Trash Can**	**Mean**
VLM (ours)	96.9 % (31)	100.0 % (32)	81.3 % (26)	100.0 % (32)	75.0 % (24)	90.6 %
Rule–based	18.8 % (6)	100.0 % (32)	34.4 % (11)	100.0 % (32)	43.8 % (14)	59.4 %
Random	0.0 % (0)	0.0 % (0)	0.0 % (0)	6.3 % (2)	6.3 % (2)	2.5 %

Table 2: Pairwise McNemar tests with Bonferroni correction for the three comparisons ($\alpha = 0.05/3 \approx 0.0167$). Continuity-corrected results are given in Appendix C

Comparison	χ^2	p**-value**	**Conclusion**
VLM Assignment (ours) *vs.* Rule-Based	38.11	< 0.001	VLM $\gg$ Rule
VLM Assignment (ours) *vs.* Random	137.11	< 0.001	VLM $\gg$ Random
Rule-Based *vs.* Random	88.17	< 0.001	Rule $\gg$ Random

Human in the Loop Feedback for VLM Outputs. Participants also suggested alternative ways of assigning the panel components outside of the three provided methods in the survey. These suggested feedback edits include: applying panels only to the seat of the chair but not the backrest, applying panels to the lampshade but not the base of the lamp, and applying panels only to the bottom two tiers of the shelf. This implies that while the VLM can generate valid outputs, there can be more than one solution.

As with many design problems, there can be multiple solutions to the same issue, and these can be subject to the user's preferences. In this case, the AI-generated output serves as a starting point for some users to diverge from. Beyond automation, natural language feedback enables human control in the AI-driven pipeline, accommodating varied user preferences and creative liberties.

Robotic Assembly of VLM Generated Designs The end-to-end robotic assembly pipeline was demonstrated using various user prompts. The robot was able to execute feasible grasp configurations and assembly sequences based on VLM-generated outputs and user-refined outputs. During assembly, the robot did not place any inward- or downward-facing panels, confirming that VLM outputs can comply with fabrication constraints when potential violations are preprocessed out of the input image (Figure. 6).

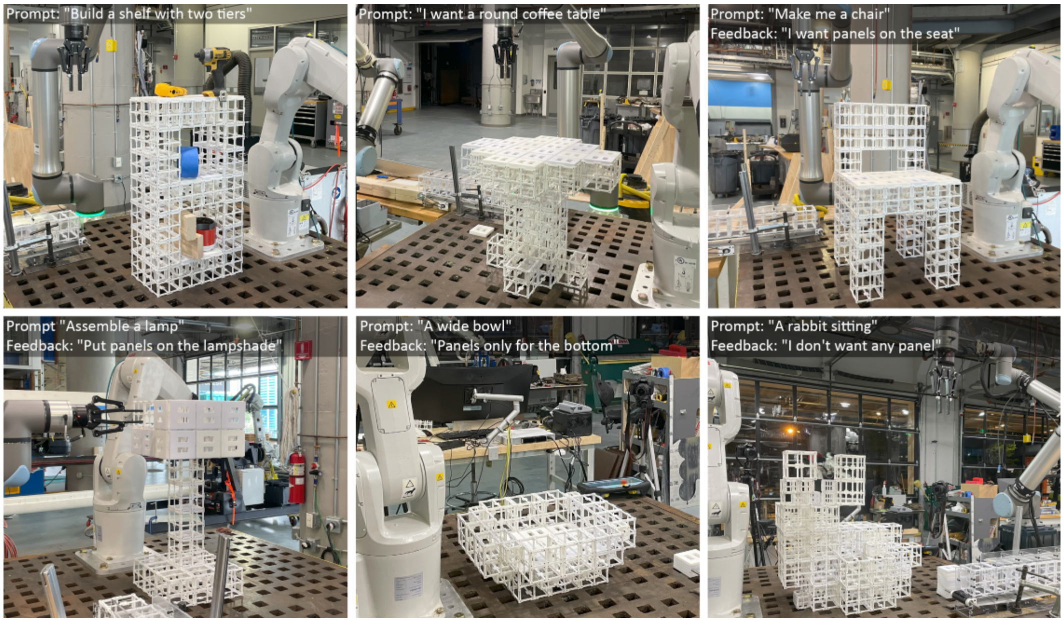

Prompt: "Build a shelf with two tiers"
Prompt: "I want a round coffee table"
Prompt: "Make me a chair"
Feedback: "I want panels on the seat"
Prompt "Assemble a lamp"
Feedback: "Put panels on the lampshade"
Prompt: "A wide bowl"
Feedback: "Panels only for the bottom"
Prompt: "A rabbit sitting"
Feedback: "I don't want any panel"

Figure 6: Robotic assembly of select user prompts and feedback, https://youtu.be/ZJrsWG7Mw5M

5 Discussion and Limitations

Our work demonstrates that VLMs can decompose AI-generated meshes into predefined components based on object functionality and geometry, using zero-shot multi-modal reasoning without task-specific training. We attribute the relative success of the VLM-based approach to its broad prior knowledge of object functions, geometric reasoning, and worldview acquired through large-scale multi-modal pretraining. Additionally, unlike the rule-based approach, VLMs can ground their reasoning in both the user prompt and the geometry of the object provided in the input image.

One of the constraints of this study is the use of predefined assembly components. The current implementation is restricted to two component types. While the narrower scope enabled controlled evaluation and physical assembly with the robotic arm, future work can expand the fixed component library to diversify the types of assembly elements [21]. This includes extending the VLM pipeline to additional functional components, such as hinges and handles, as well as material-specific components, such as wood, plastic, or metal. Currently, the evaluation is limited to common objects and simple prompts. Future studies should explore the framework with complex user prompts or unconventional objects, which may require multi-turn human-AI interaction with larger edit distances.

This work introduces a pipeline that combines 3D generative AI, VLM-based reasoning, and robotic assembly to transform natural language prompts into functional multi-component objects. In addition, the system supports user edits through natural language feedback, bridging automation with human agency. Together, these capabilities move toward a vision-language-driven workflow for human-AI co-creation and robotic assembly.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances, August 2022. URL `http://arxiv.org/abs/2204.01691`. arXiv:2204.01691 [cs].

[2] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.16. URL `http://ieeexplore.ieee.org/document/8099499/`.

[3] Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. PartGen: Part-level 3D Generation and Reconstruction with Multi-View Diffusion Models, December 2024. URL `http://arxiv.org/abs/2412.18608`. arXiv:2412.18608 [cs].

[4] Valdemar Danry and Cenk Güzelis. Organs without Bodies. Generating physical objects with AI, July 2023. URL `https://organs.media.mit.edu/`.

[5] Kristen M. Edwards, Brandon Man, and Faez Ahmed. Sketch2Prototype: rapid conceptual design exploration and prototyping with generative AI. *Proceedings of the Design Society*, 4: 1989–1998, May 2024. ISSN 2732-527X. doi: 10.1017/pds.2024.201.

[6] Faraz Faruqi, Ahmed Katary, Tarik Hasic, Amira Abdel-Rahman, Nayeemur Rahman, Leandra Tejedor, Mackenzie Leake, Megan Hofmann, and Stefanie Mueller. Style2Fab: Functionality-Aware Segmentation for Fabricating Personalized 3D Models with Generative AI. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, pages 1–13, New York, NY, USA, October 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606723. URL `https://dl.acm.org/doi/10.1145/3586183.3606723`.

[7] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, October 2022. URL `https://openreview.net/forum?id=GAUwreODU5L`.

[8] Benjamin Jenett, Amira Abdel-Rahman, Kenneth Cheung, and Neil Gershenfeld. Material–Robot System for Assembly of Discrete Cellular Structures. *IEEE Robotics and Automation Letters*, 4(4):4019–4026, October 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2930486. URL `https://ieeexplore.ieee.org/document/8769886`. Conference Name: IEEE Robotics and Automation Letters.

[9] R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J. Mitra, and Daniel Ritchie. ShapeAssembly: Learning to Generate Programs for 3D Shape Structure Synthesis. *ACM Transactions on Graphics*, 39(6):1–20, December 2020. ISSN 0730-0301, 1557-7368. doi: 10.1145/3414685.3417812. URL `http://arxiv.org/abs/2009.08026`. arXiv:2009.08026 [cs].

[10] Alexander Htet Kyaw, Se Hwan Jeon, Miana Smith, and Neil Gershenfeld. Making Physical Objects with Generative AI and Robotic Assembly: Considering Fabrication Constraints, Sustainability, Time, Functionality, and Accessibility, August 2025. URL `http://arxiv.org/abs/2504.19131`. arXiv:2504.19131 [cs].

[11] Alexander Htet Kyaw, Se Hwan Jeon, Miana Smith, and Neil Gershenfeld. Speech to Reality: On-Demand Production using Natural Language, 3D Generative AI, and Discrete Robotic Assembly, June 2025. URL http://arxiv.org/abs/2409.18390. arXiv:2409.18390 [cs].

[12] Chenghao Li, Chaoning Zhang, Joseph Cho, Atish Waghwase, Lik-Hang Lee, Francois Rameau, Yang Yang, Sung-Ho Bae, and Choong Seon Hong. Generative AI meets 3D: A Survey on Text-to-3D in AIGC Era, October 2024. URL http://arxiv.org/abs/2305.06131. arXiv:2305.06131 [cs].

[13] Yichen Li, Kaichun Mo, Yueqi Duan, He Wang, Jiequan Zhang, Lin Shao, Wojciech Matusik, and Leonidas Guibas. Category-Level Multi-Part Multi-Joint 3D Shape Assembly. pages 3281–3291. IEEE Computer Society, June 2024. ISBN 9798350353006. doi: 10.1109/CVPR52733.2024.00316. URL https://www.computer.org/csdl/proceedings-article/cvpr/2024/530000d281/20hTkUxZ5Go.

[14] Weiyu Liu, Yilun Du, Tucker Hermans, Sonia Chernova, and Chris Paxton. StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects, April 2023. URL http://arxiv.org/abs/2211.04604. arXiv:2211.04604 [cs].

[15] Kamal Rahimi Malekshan. Project Bernini. URL https://www.research.autodesk.com/projects/project-bernini/.

[16] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J. Guibas. StructureNet: Hierarchical Graph Networks for 3D Shape Generation, August 2019. URL http://arxiv.org/abs/1908.00575. arXiv:1908.00575 [cs].

[17] Evangelos Ntavelis, Aliaksandr Siarohin, Kyle Olszewski, Chaoyang Wang, Luc Van Gool, and Sergey Tulyakov. Autodecoding latent 3D diffusion models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 67021–67047, Red Hook, NY, USA, December 2023. Curran Associates Inc.

[18] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural Parts: Learning Expressive 3D Shape Abstractions with Invertible Neural Networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3203–3214, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-6654-4509-2. doi: 10.1109/CVPR46437.2021.00322. URL https://ieeexplore.ieee.org/document/9577821/.

[19] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion, September 2022. URL http://arxiv.org/abs/2209.14988. arXiv:2209.14988 [cs].

[20] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and Where Pathways for Robotic Manipulation, September 2021. URL http://arxiv.org/abs/2109.12098. arXiv:2109.12098 [cs].

[21] Miana Smith, Paul Arthur Richard, Alexander Htet Kyaw, and Neil Gershenfeld. Hierarchical Discrete Lattice Assembly: An Approach for the Digital Fabrication of Scalable Macroscale Structures, October 2025. URL http://arxiv.org/abs/2510.13686. arXiv:2510.13686 [cs].

[22] Erik Westphal and Hermann Seitz. Generative Artificial Intelligence: Analyzing Its Future Applications in Additive Manufacturing. *Big Data and Cognitive Computing*, 8:74, July 2024. doi: 10.3390/bdcc8070074.

[23] Kevin Xie, Jonathan Lorraine, Tianshi Cao, Jun Gao, James Lucas, Antonio Torralba, Sanja Fidler, and Xiaohui Zeng. LATTE3D: Large-scale Amortized Text-To-Enhanced3D Synthesis. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXVII*, pages 305–322, Berlin, Heidelberg, October 2024. Springer-Verlag. ISBN 978-3-031-72979-9. doi: 10.1007/978-3-031-72980-5_18. URL https://doi.org/10.1007/978-3-031-72980-5_18.

[24] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D Mesh Generation from a Single Image with Sparse-view Large Reconstruction Models, April 2024. URL http://arxiv.org/abs/2404.07191. arXiv:2404.07191 [cs].

[25] Hongyu Yan and Yadong Mu. Neural Assembler: Learning to Generate Fine-Grained Robotic Assembly Instructions from Multi-View Images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(14):14717–14725, April 2025. ISSN 2374-3468. doi: 10.1609/aaai.v39i14.33613. URL https://ojs.aaai.org/index.php/AAAI/article/view/33613. Number: 14.

[26] Li Yi, Hao Su, Xingwen Guo, and Leonidas Guibas. SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6584–6592, July 2017. doi: 10.1109/CVPR.2017.697. URL https://ieeexplore.ieee.org/document/8100180. ISSN: 1063-6919.

[27] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. PartNet: A Recursive Part Decomposition Network for Fine-Grained and Hierarchical Shape Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9483–9492, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-7281-3293-8. doi: 10.1109/CVPR.2019.00972. URL https://ieeexplore.ieee.org/document/8953812/.

[28] Xinghao Zhu, Devesh K. Jha, Diego Romeres, Lingfeng Sun, Masayoshi Tomizuka, and Anoop Cherian. Multi-level Reasoning for Robotic Assembly: From Sequence Inference to Contact Selection. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 816–823, May 2024. doi: 10.1109/ICRA57147.2024.10611259. URL https://ieeexplore.ieee.org/document/10611259.

A Technical Appendices and Supplementary Material

The following sections include additional details such as supplementary figures, full system prompts, robotic assembly implementation, user survey setup, and full calculation of the statistical analysis results on the experiment data

A.1 Predefined Component types: Structural and Panel

The structural component is a volumetric cube composed of six faces. It is used for the object's base geometry to ensure stable robotic assembly. Each side of the structural component has 16 small magnets arranged in a polarity pattern (positive and negative) that ensures proper alignment when the adjacent component connects. The cube's internal grid structure forms a lattice, making it lightweight while maintaining strength. The top face of the structural component includes a unique feature that allows the robotic arm's gripper to easily grasp it. The panel component is a flat plane designed to attach to the structural component. To ensure proper connection, the magnet arrangement follows the same polarity pattern as the structural component. Additionally, the panel component includes openings that allow the robotic gripper to securely grasp it during assembly. See figure 7.

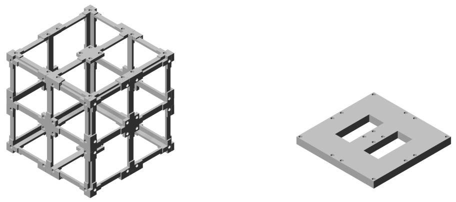


Figure 7: The two predefined component types used in our system.

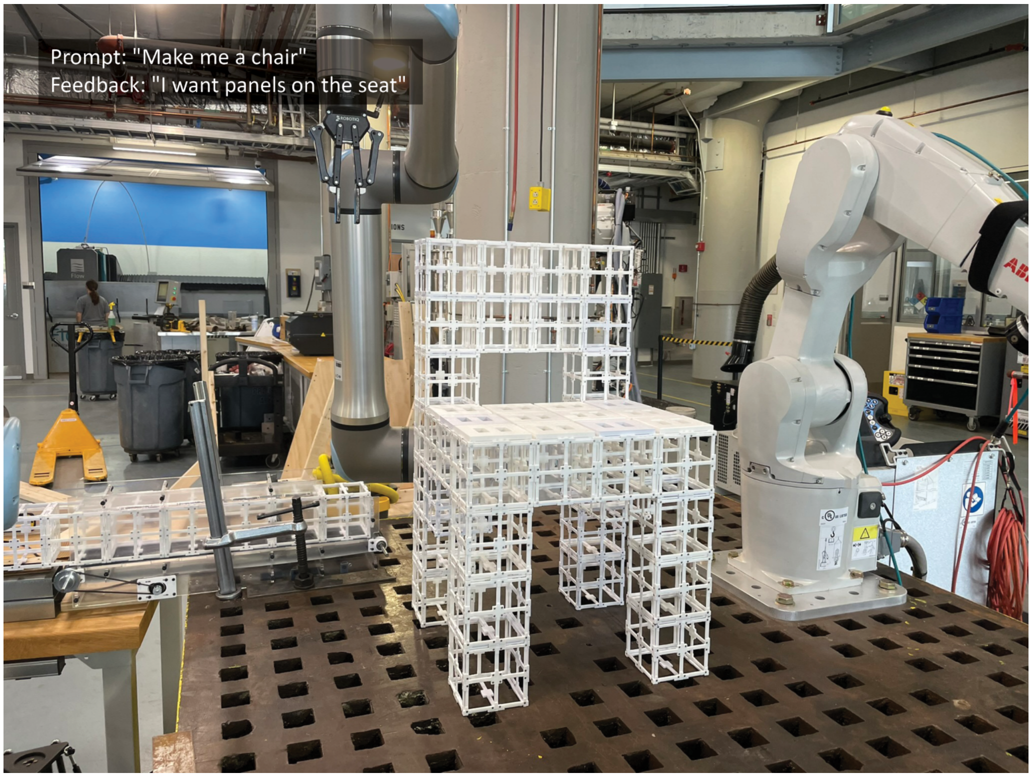

Prompt: "Make me a chair"
Feedback: "I want panels on the seat"

Figure 8: Text to multi-component robotic assembly of the user prompt: "Make me a chair", "I want panels on the seat"

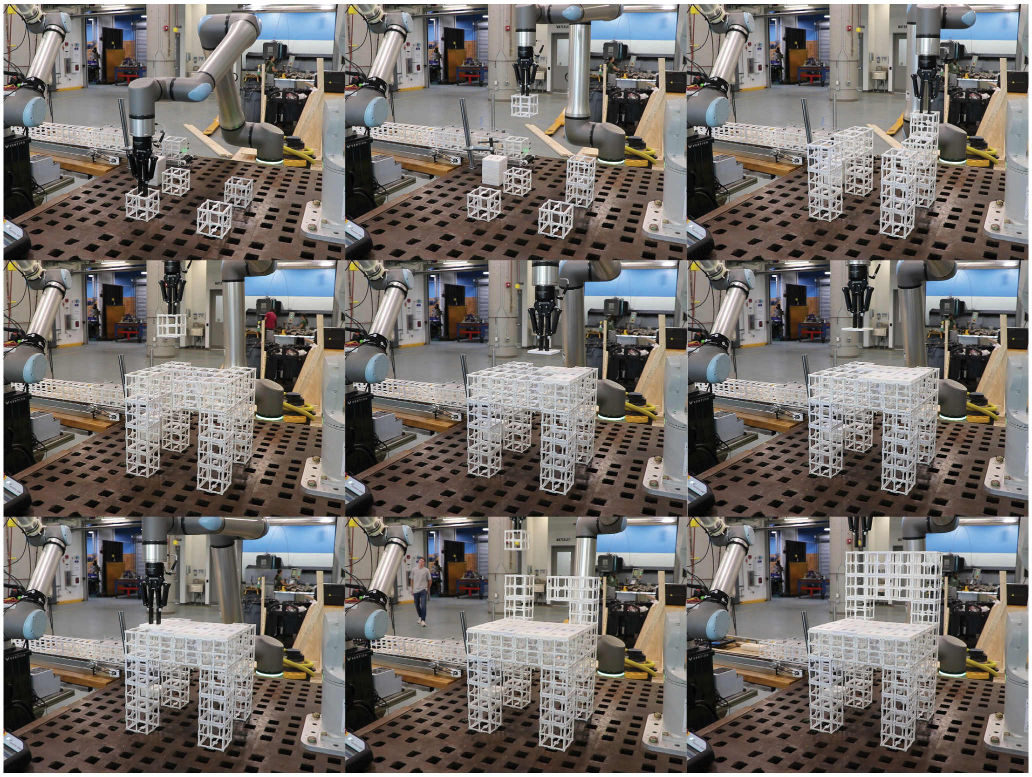


Figure 9: The robotic assembly process of the prompt: "Make me a chair", "I want panels on the seat"

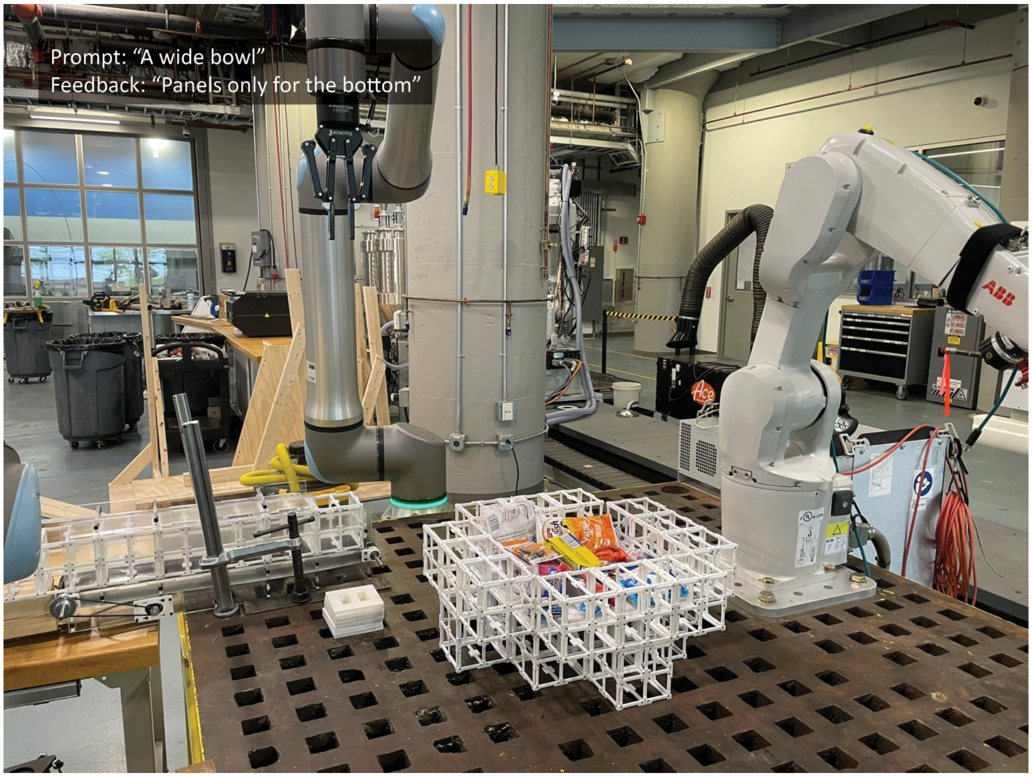


Figure 10: Text to multi-component robotic assembly of the user prompt: "A wide bowl", Panels only for the bottom

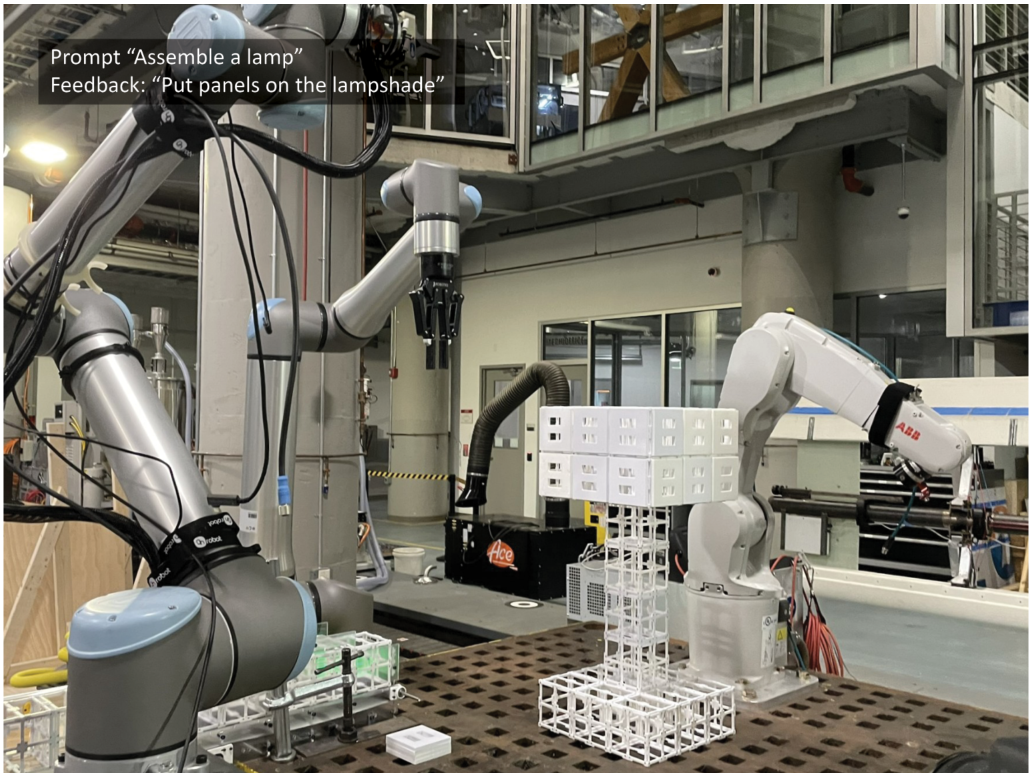


Figure 11: Text to multi-component robotic assembly of the user prompt: "Assemble a lamp", "Put panels on the lampshade"

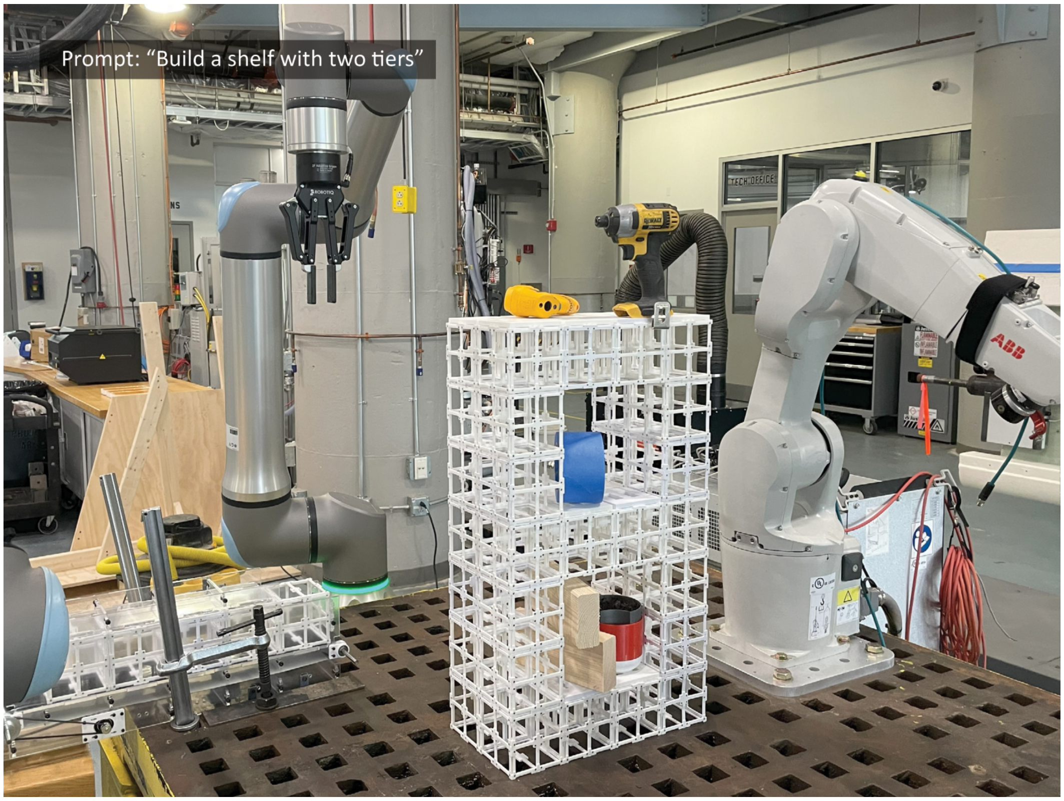


Figure 12: Text to multi-component robotic assembly of the prompt: "Build a shelf with two tiers"

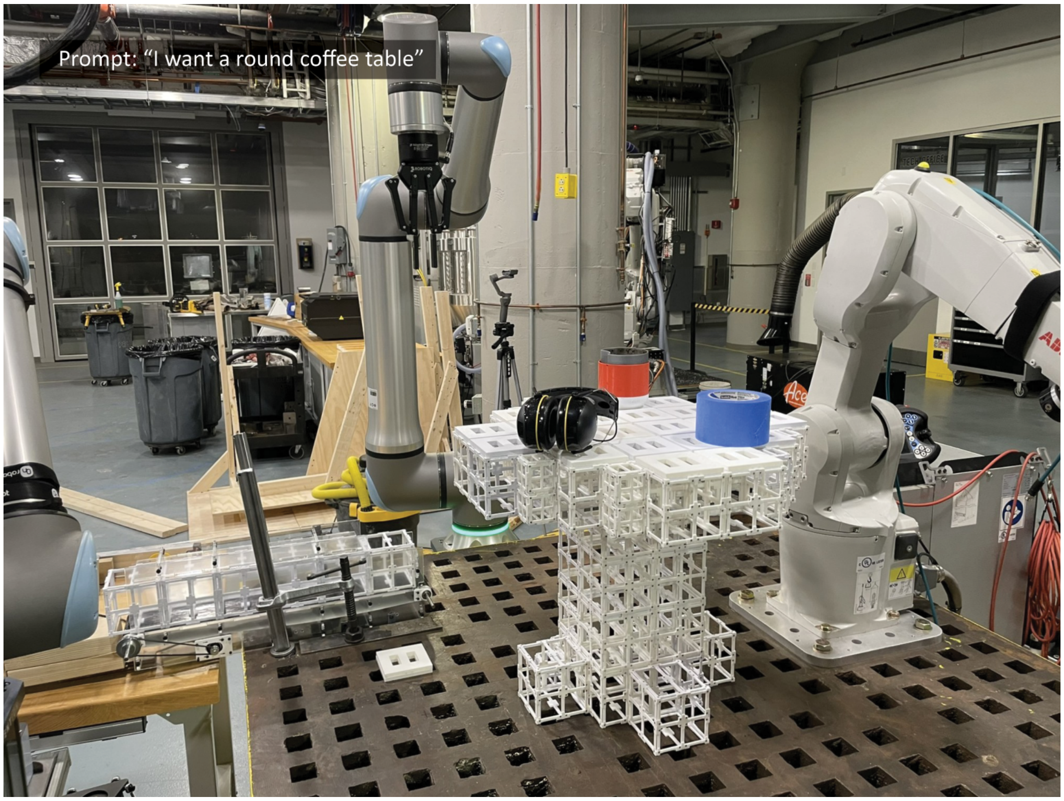


Figure 13: Text to multi-component robotic assembly of the prompt: "I want a round coffee table"

B Robotic Assembly Implementation Details

The robotic assembly demonstrations are developed using Autodesk's internal robotics research platform integrated with Fusion 360. All experiments were conducted on a standard desktop that can run Python code.

Algorithm 1 Robotic Assembly Algorithm for Panel Component and Structural Component

```
Require:  • Component coordinates $C = \{(x_i, y_i, z_i, r_{x_i}, r_{y_i}, r_{z_i})\}_{i=1}^{n}$
          • Component types $T = \{t_0, t_1\}$ where $t_i \in \{0, 1\}$
          • Source locations $S_0, S_1$ for each component type
Ensure: All component are placed at target positions using pick-and-place operations

Initialization
Initialize robot to home pose $p_{\text{root}}$
Open gripper with width $w_{\text{open}}$
Iterate through all components in component cooridnates sequence
for $i = 1$ to $n$ do
    Determine source location based on component type
    if $t_i = 0$ then Structural Component
        $source \leftarrow S_0$
    end if
    if $t_i = 1$ then Panel Component
        $source \leftarrow S_1$
    end if
    Compute pickup and placement poses
    $pickup \leftarrow (source.x, source.y, source.z)$
    $place \leftarrow (x_i, y_i, z_i, r_{x_i}, r_{y_i}, r_{z_i})$
    Pick-up sequence
    Move to $(pickup.x, pickup.y, pickup.z + h_{\text{safe}})$
    Move to $pickup$
    Close gripper with force $f_{\text{grab}}$
    Move to $(pickup.x, pickup.y, pickup.z + h_{\text{safe}})$
    Placement sequence
    Move to $(place.x, place.y, place.z + h_{\text{safe}})$
    Move to $place$
    Open gripper with width $w_{\text{release}}$
    Move to $(place.x, place.y, place.z + h_{\text{safe}})$
    Output: Component $i$ successfully placed
end for
```

C Additional Details on Experiment and Results

IRB Approval Statement This study involving human subjects was reviewed by the Massachusetts Institute of Technology Committee on the Use of Humans as Experimental Subjects (MIT COUHES), which serves as the Institutional Review Board (IRB) for MIT.

The study received an IRB exemption determination (Exemption Category 3) as it involved minimal risk and no collection of sensitive personal information. All participants were adults who provided informed consent prior to participating. No personally identifying information was collected.

Before beginning the survey, participants were informed that:

- They would be shown five objects generated by our AI-based design system.
- Their task would be to evaluate panel placements for each object.
- They could skip any question they did not wish to answer.
- Their responses would remain anonymous and would be used for academic research.
- They would not receive monetary or other forms of compensation for participating.

Instructions for Selection

For each object, you will see three possible configurations for where panels could be placed. The configurations were generated using three different methods. Select all configurations that you believe have appropriate or acceptable panel placements for the given object and its function. You may select one, more than one, or none at all. Please select according to what you think is functionally appropriate. Please use the check mark on Adobe PDF to select a checkbox.

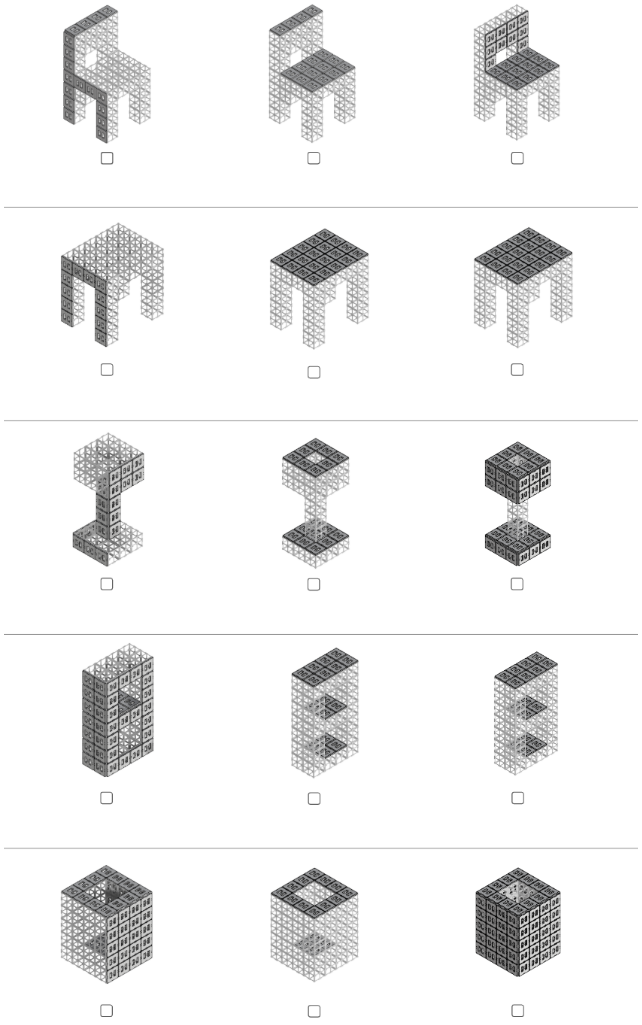


Figure 14: Layout of the survey document with three options of each of the five objects. No labels were provided to avoid bias. A checkbox allowed participants to select each option, and a line was provided after each object for text input.

Instructions for Alternative Assignments

For each object, indicate whether you think there is an alternative way to assign panels that is different from the options shown above. If yes, describe how you would assign the panels differently. If no, just write "No.". Please use Adobe PDF text box for the text entry.

C.1 Derivation of Test Statistics: Pairwise McNemar Tests

For every *participant–object–method* triple we log a binary outcome (1 = "panel placement judged appropriate", 0 = "not selected"). When comparing two methods, A and B, we form a 2 × 2 contingency table and consider only the *discordant* cells:

Table 3: Variables for McNemar Calculation.

Participant's judgement on that object	**Contributes to**
A = 1, B = 0 (A chosen, B not)	b
A = 0, B = 1 (B chosen, A not)	c
A = 1, B = 1 *or* A = 0, B = 0	(concordant; ignored)

Test statistic. McNemar's null hypothesis is $b = c$. We report the *uncorrected* statistic and, in parentheses, the continuity-corrected version:

$$\chi^2_{\text{uncorr}} = \frac{(b-c)^2}{b+c}, \qquad \chi^2_{\text{corr}} = \frac{(|b-c|-1)^2}{b+c}.$$

Table 4: Discordant counts and McNemar results

Comparison	b	c	χ^2_{uncorr}	p	χ^2_{corr}	p
VLM *vs.* Rule	56	7	38.11	< 0.001	36.57	< 0.001
VLM *vs.* Random	143	2	137.11	< 0.001	135.17	< 0.001
Rule *vs.* Random	94	2	88.17	< 0.001	86.26	< 0.001

All p-values remain well below the Bonferroni-adjusted threshold $\alpha = 0.05/3 \approx 0.0167$; applying the continuity correction does not change any conclusion.

Interpretation. A larger χ^2 indicates a stronger imbalance between the two discordant cells. For example, more participants selected one method but not the other. The results confirm that VLM-generated placements were overwhelmingly preferred over both baselines, even though participants were free to select multiple methods for the same object.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The introduction lists three concrete contributions, and these same contributions are implemented in the Methods section and validated in Experiments.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses several limitations of the proposed work, including the use of a fixed library and the evaluation being limited to simple prompts. The authors also note that future extensions should support a broader range of functional components and multi-turn user interaction.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA] .

   Justification: The paper does not present any formal theoretical results or mathematical theorems. It focuses on a novel system design and empirical evaluation of VLM-based reasoning for robotic assembly.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the experimental setup, including the user study, VLM prompting tasks, mesh discretization, component assignment logic, and user study protocol. It also includes algorithmic pseudocode for robotic assembly and statistical analysis procedures.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No] .

   Justification: The code for Autodesk's internal robotics platform and Project Bernini is currently under active research and remains closed-source / proprietary. However, the paper offers detailed descriptions of the key methodology and how alternative tools can be used as substitutes for the framework in the Appendix. The paper also provides open access to the VLM prompts and the robotic assembly algorithm in the appendix.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper clearly specifies all relevant experimental details, including the user study, the design of the VLM prompting tasks, the inputs provided (text prompts and images), the mesh discretization process, the predefined component types, and robotic assembly for the readers to understand and interpret the results. See Appendix B. The paper does not involve training new models, as it uses a pretrained VLM (Gemini 2.5 Pro) in a zero-shot setting. Therefore, there are no training data splits, hyperparameters, or optimizers to report.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The result section and Table 2 reports chi-square statistics and Bonferroni-corrected p-values for all pairwise comparisons. The statistical test used (McNemar test) is for the non-exclusive selection setup, and the calculation is explained in detail in Appendix C.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: : The paper specifies that the VLM experiments were conducted using Google's Gemini 2.5 Pro in a zero-shot setting, which does not require local training or significant compute resources. The robotic assembly experiments were conducted using a Universal Robots UR20 industrial robotic arm, with the process described in detail in Appendix B. All experiments were conducted on a standard desktop that can run Python code, which interfaced with Project Bernini for 3D generation, the VLM for reasoning, and Autodesk's internal robotics platform for robotic programming.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that the research conducted in this paper complies with all outlined principles.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses broader impacts in the Introduction and the Discussion section, highlighting societal implications of the proposed system. The Introduction section notes that the system enables accessible physical making for non-experts through natural language. We also mentioned how robotic assembly could support modular construction and part-level editing for component reuse. The discussion section calls for further study and impact, including handling complex prompts and ways to add human oversight and control in creative workflows with AI and robotic systems.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA] .

    Justification: The paper does not release any new pretrained models, large-scale datasets, or generative tools that pose a high risk of misuse. It uses existing closed-source tools (e.g., Gemini 2.5 Pro, Project Bernini) and focuses on system integration and framework design.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: All third-party assets used in the paper, including Google's Gemini 2.5 Pro vision-language model and Autodesk's Project Bernini, are properly credited in the text and citations. These tools are accessed in accordance with their respective terms of use. The paper does not redistribute these assets but uses them as services or platforms within the scope allowed for academic research. No external code or datasets with restrictive licenses are included or released.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: While the paper does not release new datasets, models, or code, it introduces new assets in the form of figures, videos, and physically assembled objects generated through the proposed pipeline. These assets are documented throughout the paper and the appendix.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The paper includes a user study involving 25 participants who evaluated component assignments across five objects. The study design is described in the main paper, including the number of participants, object conditions, and the non-exclusive selection task. Participants were recruited and completed the survey voluntarily, providing informed consent. More details on Appendix C.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The user study posed minimal risk to participants. We obtained IRB exemption through our institution's ethics review process, which determined that the study qualifies as exempt human subjects research (Appendix C) Participants were informed of the study's purpose, participated voluntarily, and provided informed consent prior to completing the survey. No sensitive personal data was collected.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper uses a language model specifically, Google's Gemini 2.5 Pro as a core component of the methodology. The VLM performs zero-shot, multimodal reasoning to assign functional components to AI-generated 3D meshes based on input prompts and object geometry.