Combinatorial 3D Shape Assembly with LEGO Bricks

Jungtaek Kim

jungtaek.kim@pitt.edu

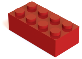
University of Pittsburgh
https://jungtaekkim.github.io

July 12, 2023

Table of Contents

Introduction

Today's Talk



- I will introduce three papers on combinatorial 3D assembly today.
- These papers solve a combinatorial construction problem that assembles unit primitives (e.g., 2×4 LEGO bricks) under a fixed rule.

Method	**Representation**	**Target Info.**	**Approach**
Kim et al. [2020]	Set	Exact target volume	Bayesian optimization
Chung et al. [2021]	Graph	Images	Reinforcement learning
Ahn et al. [2022]	Voxels	– / Incomplete target volume	Feed-forward networks

[Kim et al., 2020] **J. Kim**, H. Chung, et al. Combinatorial 3D shape generation via sequential assembly. In NeurIPS Workshop on ML4Eng, 2020.

[Chung et al., 2021] H. Chung*, **J. Kim***, et al. Brick-by-Brick: Combinatorial construction with deep reinforcement learning. In NeurIPS, 2021.

[Ahn et al., 2022] S. Ahn, **J. Kim**, et al. Budget-aware sequential brick assembly with efficient constraint satisfaction. arXiv preprint arXiv:2210.01021, 2022.

More Detailed Comparisons

Table 1: Analysis of recent studies in terms of state representation, supervision, conditioning, target objects, and action validation.

Method	**State**	**Supervision**	**Conditioning**	**Target**	**Action Validation**
Hamrick et al. [2018]	Image	Task-dependent	N/A	2D	Direct
Bapst et al. [2019]	Object/Image	Task-dependent	Object and/or image	2D	Direct
Kim et al. [2020]	Set	Overlap	Exact target volume	3D	Sampling
Thompson et al. [2020]	Graph	Step-wise CE	One-hot class info.	3D	Direct
Chung et al. [2021]	Graph	IoU	Image or set of images	3D	Pretrained
Ahn et al. [2022]	Voxels	Step-wise CE	– / Incomplete target volume	3D	Conv.

[Hamrick et al., 2018] J. B. Hamrick, K. R. Allen, et al. Relational inductive bias for physical construction in humans and machines. In CogSci, 2018.

[Bapst et al., 2019] V. Bapst, A. Sanchez-Gonzalez, et al. Structured agents for physical construction. In ICML, 2019.

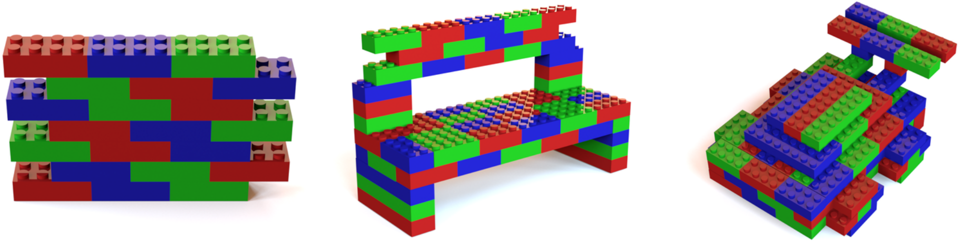
[Kim et al., 2020] **J. Kim**, H. Chung, et al. Combinatorial 3D shape generation via sequential assembly. In NeurIPS Workshop on ML4Eng, 2020.

[Thompson et al., 2020] R. Thompson, G. Elahe, et al. Building LEGO using deep generative models of graphs. In NeurIPS Workshop on ML4Eng, 2020.

[Chung et al., 2021] H. Chung*, **J. Kim***, et al. Brick-by-Brick: Combinatorial construction with deep reinforcement learning. In NeurIPS, 2021.

[Ahn et al., 2022] S. Ahn, **J. Kim**, et al. Budget-aware sequential brick assembly with efficient constraint satisfaction. arXiv preprint arXiv:2210.01021, 2022.

Introduction



- Combinatorial construction via *sequential assembly* mimics a human assembly process, by allocating a budget of primitives given.
- We introduce a sequential problem by utilizing an approach to solving *combinatorial 3D shape generation*.

Combinatorial 3D Assembly

Combinatorial 3D Assembly

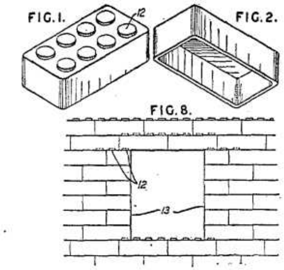


Figure 1: Taken from this link

- As introduced in [Herman, 2012], a 2×4 LEGO brick is the patent granted in 1947 of the Kiddicraft company, founded by Hilary Page.
- It opens a progressive development of building a 3D shape.

[Herman, 2012] S. Herman. Building a History: The LEGO Group. Grub Street Publishers, 2012.

Combinatorial 3D Assembly

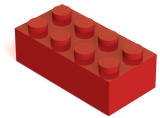


Figure 2: A 2×4 brick

- A 2×4 brick is mainly used as a unit primitive, which has eight studs and their fit cavities.
- Instead of employing other 3D representations such as point clouds, triangular meshes, and voxels, we create a sequence of unit primitives.
- Interestingly, possible positions to assemble a brick consistently grow.
- With only *six* 2×4 bricks, *915,103,765* possible combinations exist [Eilers, 2016].

[Eilers, 2016] S. Eilers. The LEGO counting problem. The American Mathematical Monthly, 123(5):415–426, 2016.

Combinatorial 3D Assembly: Assumptions

- Every connection types must follow a fixed rule.

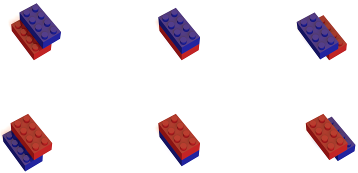


Figure 3: Example of available offsets

- No bricks mutually overlap.

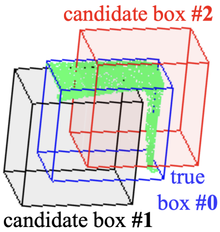


Figure 4: Taken from [Yang et al., 2019]

- If two bricks are connected, the combination of two bricks is considered that they are soldered together and cannot be broken.

[Yang et al., 2019] B. Yang, J. Wang, et al. Learning object bounding boxes for 3D instance segmentation on point clouds. In NeurIPS, 2019.

Combinatorial 3D Assembly: Action Space

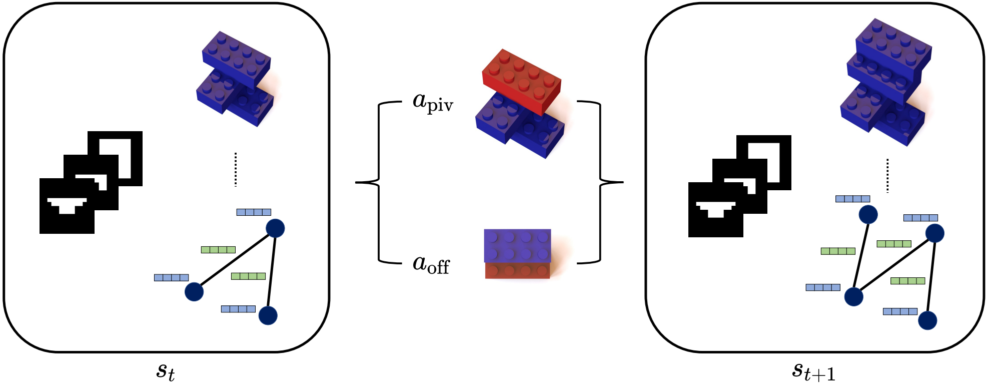


Figure 5: Successive action space with actions for selecting a pivot brick and an offset from the pivot brick

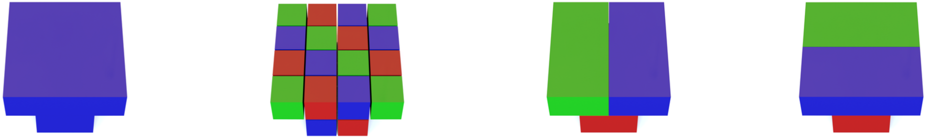
Combinatorial 3D Shape Generation via Sequential Assembly

Jungtaek Kim, Hyunsoo Chung, Jinhwi Lee, Minsu Cho, and Jaesik Park

NeurIPS Workshop on Machine Learning for Engineering Modeling, Simulation, and Design, 2020

Sequential Assembly with Unit Primitives

- Such 2×4 LEGO bricks make our problem more combinatorial and more complex, compared to other primitives.

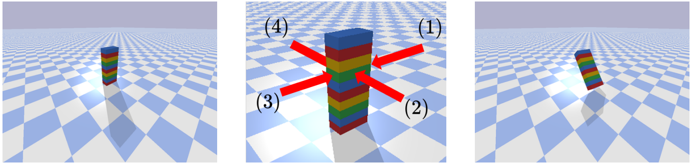


(a) Target shape (b) 1×1-sized primitives (c) 2×4-sized primitives

Figure 6: Assembly of a target shape with 1×1-sized primitives and 2×4-sized primitives

Combinatorial 3D Shape Generation

- To determine the position of the next primitive, we define two evaluation functions regarding *occupiability* and *stability*.
- Occupiability encourages us to follow a target shape and stability helps to create a physically-stable combination.
- We determine the position of the next primitive via *Bayesian optimization*.
- To avoid a suboptimal sequence, our framework includes a *rollback step*.



(a) Placing an object (b) Applying forces (c) Measuring stability

Figure 7: Stability simulation with `PyBullet`

Experimental Results

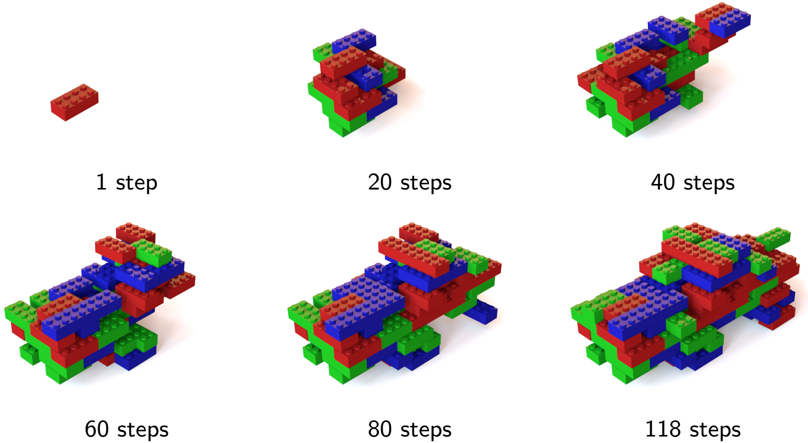
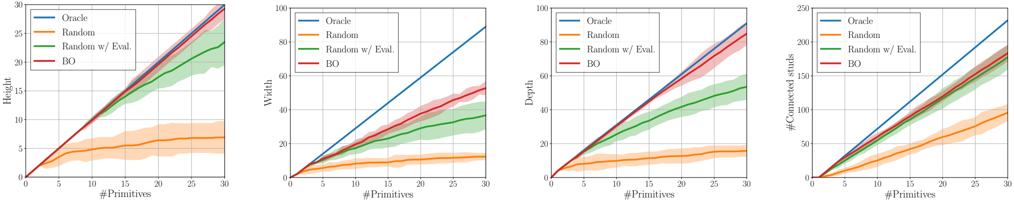


Figure 8: Generated assembling sequence that creates a *car* shape with 118 unit primitives.

Experimental Results

- We apply our framework in optimizing particular explicit functions.



(a) Height (b) Width (c) Depth (d) #Connected studs

Figure 9: Quantitative results on maximizing explicit evaluation functions

Combinatorial 3D Shape Dataset

- We also introduce a new *combinatorial 3D shape dataset* that consists of 14 classes and 406 instances.

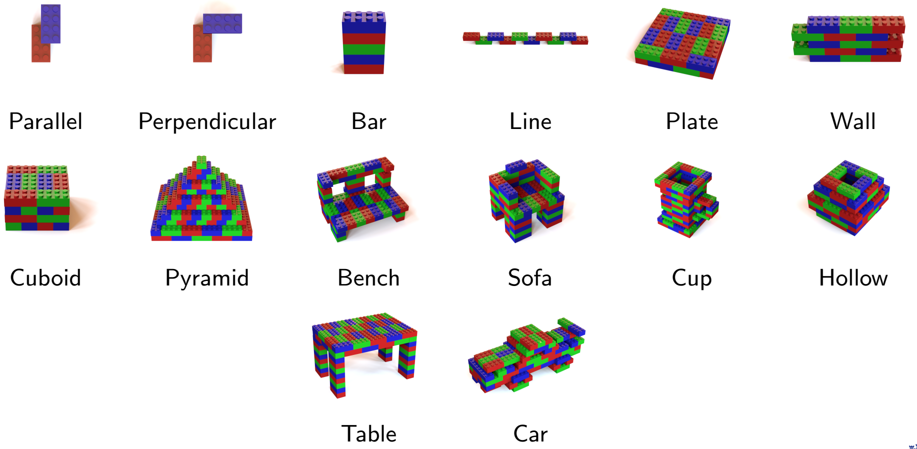


Figure 10: Selected examples from our dataset

Combinatorial 3D Shape Dataset

- The characteristics of our combinatorial 3D shape dataset are

  1. *combinatorial*: Duplicates of unit primitive is repeatedly connected;
  2. *sequential*: Allowable connections between primitives are sequentially added;
  3. *decomposable*: By the combinatorial property, parts of combination can be sampled if they are valid in terms of the contact and overlap conditions;
  4. *manipulable*: New primitive is addable or the existing primitives are removable.

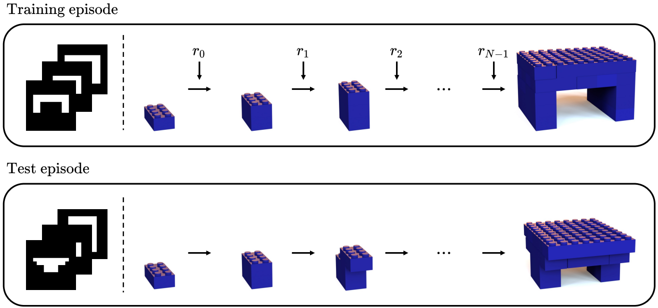
Brick-by-Brick: Combinatorial Construction with Deep Reinforcement Learning

Hyunsoo Chung*, **Jungtaek Kim***, Boris Knyazev, Jinhwi Lee, Graham W. Taylor, Jaesik Park, and Minsu Cho

NeurIPS, 2021

* Equal contribution

Overview



- It sequentially assembles unit primitives (i.e., LEGO bricks), given only incomplete target information (i.e., a 2D image or multiple views of a target object).
- It requires a comprehensive understanding of incomplete target information and long-term planning to append each brick efficiently.

Overview

- We devise a reinforcement learning (RL) approach along with the absence of sequence-level supervision.
- We express a brick combination as graph representation, where node and edge correspond to a single brick and a connection between two bricks, respectively.
- In this domain, however, we struggle to handle both an indefinite action space and the existence of many invalid actions when applying RL.
- To resolve the aforementioned issues, we adopt an action validity prediction network that filters out invalid actions to an actor-critic network.

Combinatorial Construction: Overall Scenario

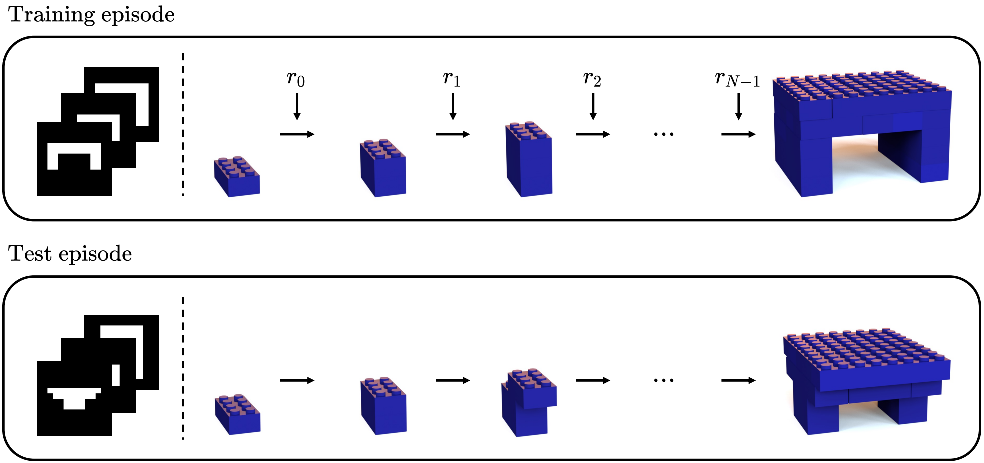


Figure 11: Training and test episodes of combinatorial construction

Brick-by-Brick

- Suppose that *target information* $\mathcal{T}$, i.e., a single binary image or a set of three binary images from different views of a target object, is given as partial information.
- Represent each t-th *state* s_t of the MDP as a tuple of a directed graph G_t composed of t bricks and target information $\mathcal{T}$, i.e., $s_t = (G_t, \mathcal{T})$.
- With t bricks assembled, define an *action* $a_t = (a_t^{\text{piv}}, a_t^{\text{off}})$ where a_t^{piv} is to select a pivot brick and a_t^{off} is to select an offset with respect to the pivot brick.
- Transform the combination of currently assembled bricks into the occupancy of the voxels and measure the overlap between them as a *reward function*:

$$\Delta\text{IoU}(\mathbf{C}_t, \mathbf{T}) = \frac{\text{vol}(\mathbf{C}_t \cap \mathbf{T})}{\text{vol}(\mathbf{C}_t \cup \mathbf{T})} - \frac{\text{vol}(\mathbf{C}_{t-1} \cap \mathbf{T})}{\text{vol}(\mathbf{C}_{t-1} \cup \mathbf{T})}, \tag{1}$$

where $\mathbf{C}_t$, $\mathbf{C}_{t-1}$, and $\mathbf{T}$ are the occupied voxels at timestep t, timestep $t-1$, and a desired target, respectively. Note that $\text{vol}(\cdot)$ is a function that measures a volume.

Brick-by-Brick

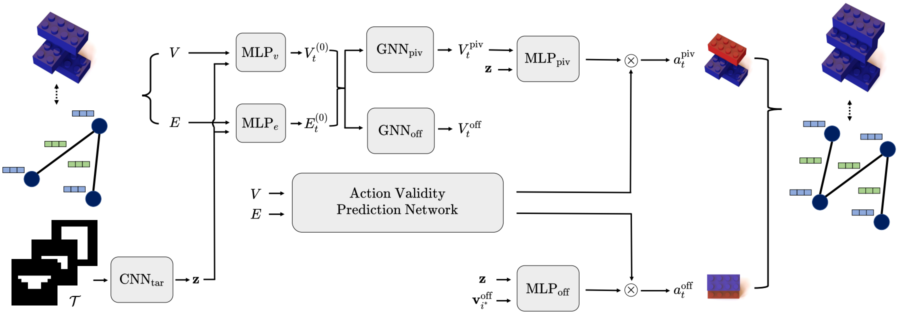


Figure 12: An overview of our proposed method, named Brick-by-Brick. The red brick in both a_t^{piv} and offset a_t^{off} indicates the chosen brick.

- We use a respective graph neural network for a_t^{piv} and a_t^{off}, inspired by Battaglia et al. [2018].

[Battaglia et al., 2018] P. W. Battaglia, J. B. Hamrick, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

Action Validity Prediction Network

- An action validity prediction network predicts an invalid action using a surrogate for confirming the validity of a given action.
- It trains a graph neural network, of which the head corresponds to the degree of validity.
- It is capable of pre-training the graph neural network with the ground-truth validity of actions, which is obtained by randomly-assembled objects.
- It applies the network in training an actor-critic network, without re-training.

Experimental Results: MNIST Construction

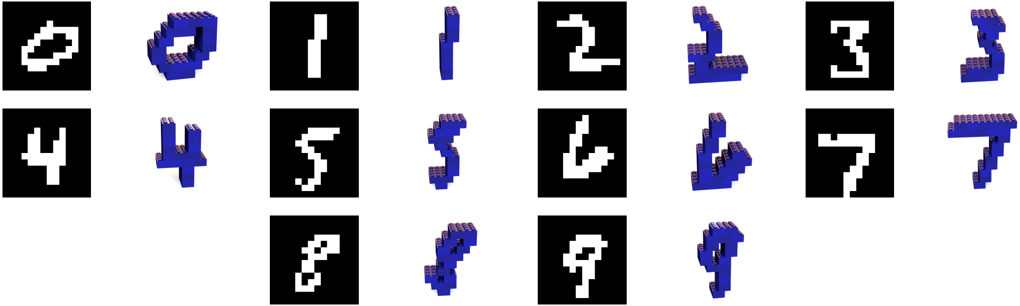
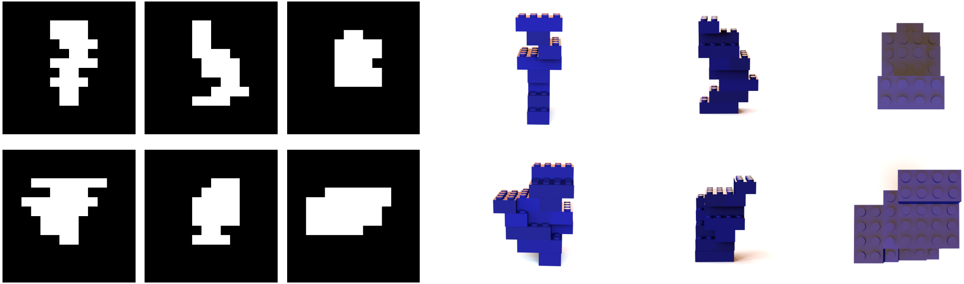


Figure 13: Qualitative results on MNIST construction

Experimental Results: Randomly-Assembled Object Construction



(a) Target images (b) Constructed object from three viewpoints

Figure 14: Qualitative results on randomly-assembled object construction

Experimental Results: ModelNet Construction

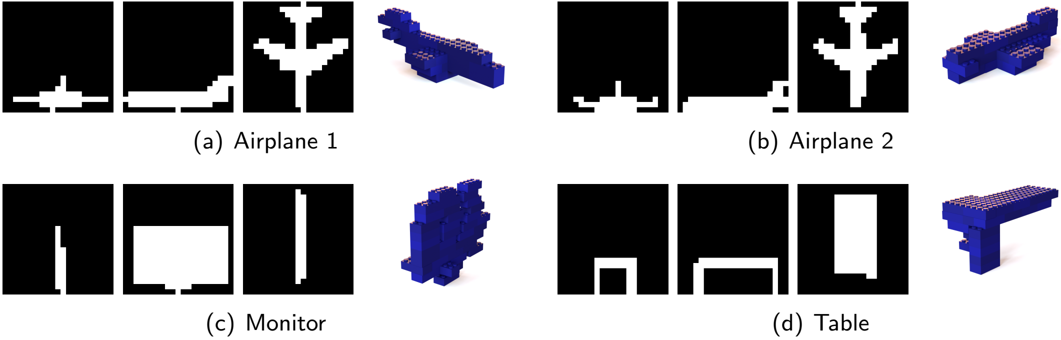
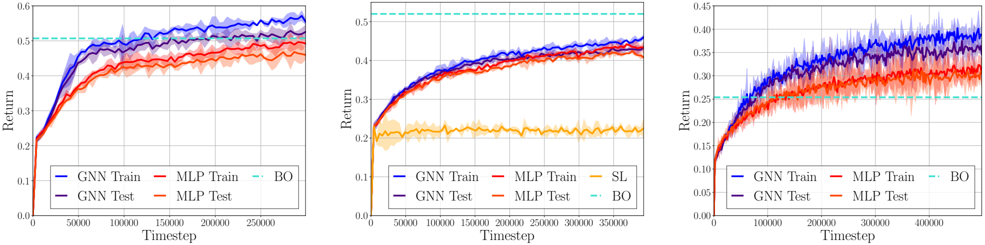


Figure 15: Qualitative results on ModelNet construction

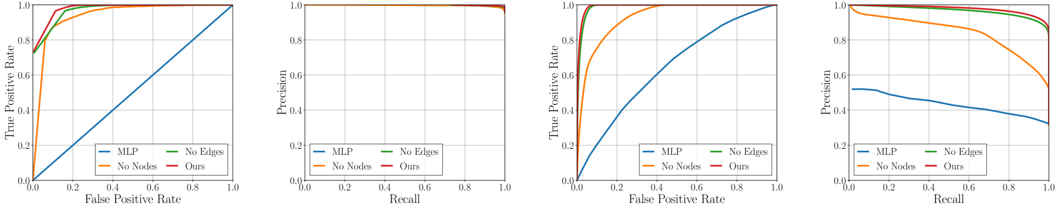
Experimental Results: Episode Return Curves



(a) MNIST (b) Randomly-Assembled (c) ModelNet

Figure 16: Episode return curves vs. timesteps in different setups. The curves measured by training and test episodes are reported by repeating 3 times with different seeds.

Experimental Results: Analysis on Action Validity Prediction Network



(a) Pivot ROC (b) Pivot PR (c) Offset ROC (d) Offset PR

Figure 17: ROC and PR curves for action validity prediction networks

Budget-Aware Sequential Brick Assembly with Efficient Constraint Satisfaction

Seokjun Ahn, **Jungtaek Kim**, Minsu Cho, and Jaesik Park

Under Review

Overview

- Our method assesses a brick structure to predict the next brick position and its confidence by employing a U-shaped sparse 3D convolutional neural network.
- A convolution filter, which is initialized by one, efficiently validates physical constraints in a parallelizable and scalable manner.
- The convolution filter effectively allows us to deal with different brick types.
- To generate a novel structure, we devise a sampling strategy to determine the next brick position.
- We consider a budget, i.e., the limited number of bricks and types, in the sampling strategy.

Sequential Brick Assembly with Efficient Constraint Satisfaction

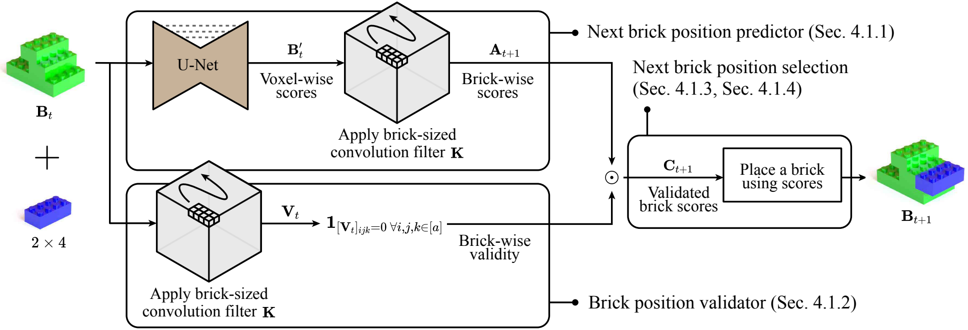
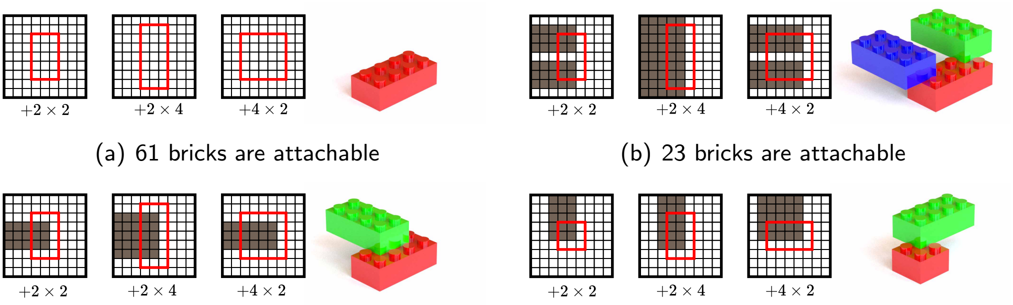


Figure 18: Our proposed efficient constraint satisfaction method with convolution filters for sequential brick assembly, named BrECS

- Completion and generation tasks can be solved with our method.

Sequential Brick Assembly with Efficient Constraint Satisfaction



(a) 61 bricks are attachable

(b) 23 bricks are attachable

(c) 33 bricks are attachable

(d) 19 bricks are attachable

Figure 19: Attachable brick positions corresponding to brick structures

Qualitative Results

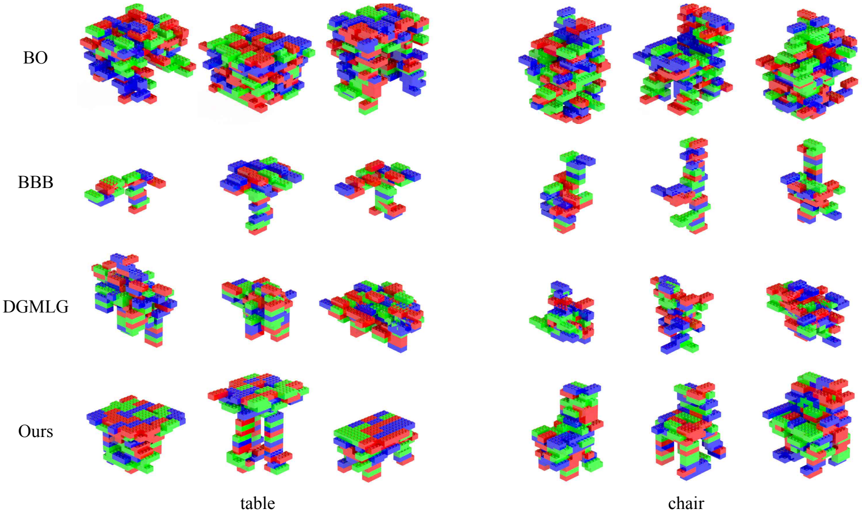


Figure 20: Qualitative results for tables and chairs

Quantitative Results

Methods	IoU (↑)				% valid (↑)				Inference Time (sec., ↓)			
	airplane	table	chair	avg.	airplane	table	chair	avg.	airplane	table	chair	avg.
BayesOpt*	0.145	0.206	0.233	0.194	**100.0**	**100.0**	**100.0**	**100.0**	1.20e6	1.11e6	1.05e6	1.12e6
Brick-By-Brick*	0.455	0.440	0.434	0.443	12.0	7.0	16.0	11.7	305.6	1502.4	2785.2	1531.1
DGMLG	0.315	0.269	0.271	0.285	0.0	1.0	0.0	0.3	237.3	340.0	473.0	350.4
BrECS (2×4)	0.571	0.586	0.534	0.564	**100.0**	**100.0**	**100.0**	**100.0**	**36.3**	**143.9**	**151.0**	**110.4**
BrECS ($2 \times 4 + 2 \times 2$)	**0.599**	**0.594**	**0.541**	**0.578**	**100.0**	**100.0**	**100.0**	**100.0**	73.8	224.1	279.0	192.3

Figure 21: Quantitative results for a completion task. Asterisk denotes that partial or full ground-truth information is given to the corresponding model.

Quantitative Results

Methods	Class prob. of target class (↑)				% valid (↑)			
	airplane	table	chair	avg.	airplane	table	chair	avg.
BayesOpt*	0.039	0.043	0.069	0.050	**100.0**	**100.0**	**100.0**	**100.0**
Brick-By-Brick*	0.430	0.042	0.032	0.168	6.0	3.0	2.0	3.7
DGMLG	0.228	0.023	0.027	0.093	0.0	0.0	0.0	0.0
BrECS (2×4)	0.415	**0.250**	0.404	0.356	**100.0**	**100.0**	**100.0**	**100.0**
BrECS ($2 \times 4 + 2 \times 2$)	**0.447**	0.229	**0.419**	**0.365**	**100.0**	**100.0**	**100.0**	**100.0**

Figure 22: Quantitative results for a generation task. Asterisk denotes that partial or full ground-truth information is given to the corresponding model.

Discussion & Conclusion

More Detailed Comparisons

Table 2: Analysis of recent studies in terms of state representation, supervision, conditioning, target objects, and action validation.

Method	State	Supervision	Conditioning	Target	Action Validation
Hamrick et al. [2018]	Image	Task-dependent	N/A	2D	Direct
Bapst et al. [2019]	Object/Image	Task-dependent	Object and/or image	2D	Direct
Kim et al. [2020]	Set	Overlap	Exact target volume	3D	Sampling
Thompson et al. [2020]	Graph	Step-wise CE	One-hot class info.	3D	Direct
Chung et al. [2021]	Graph	IoU	Image or set of images	3D	Pretrained
Ahn et al. [2022]	Voxels	Step-wise CE	– / Incomplete target volume	3D	Conv.

[Hamrick et al., 2018] J. B. Hamrick, K. R. Allen, et al. Relational inductive bias for physical construction in humans and machines. In CogSci, 2018.

[Bapst et al., 2019] V. Bapst, A. Sanchez-Gonzalez, et al. Structured agents for physical construction. In ICML, 2019.

[Kim et al., 2020] **J. Kim**, H. Chung, et al. Combinatorial 3D shape generation via sequential assembly. In NeurIPS Workshop on ML4Eng, 2020.

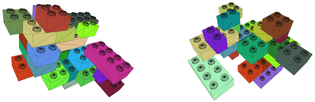
[Thompson et al., 2020] R. Thompson, G. Elahe, et al. Building LEGO using deep generative models of graphs. In NeurIPS Workshop on ML4Eng, 2020.

[Chung et al., 2021] H. Chung*, **J. Kim***, et al. Brick-by-Brick: Combinatorial construction with deep reinforcement learning. In NeurIPS, 2021.

[Ahn et al., 2022] S. Ahn, **J. Kim**, et al. Budget-aware sequential brick assembly with efficient constraint satisfaction. arXiv preprint arXiv:2210.01021, 2022.

Discussion

- We open a new line of research on a sequential and combinatorial construction problem.
- More realistic rewards such as stability and feasibility can be discussed.
- Better graph representation to express the state of brick combination can be suggested.
- Text-guided combinatorial assembly can be investigated.
- A large-scale graph dataset, which contains randomly-assembled objects with diverse types of unit primitives, e.g., 2×4, 2×2, and 1×2 bricks, can be created.



Conclusion

- We proposed the problem formulation of sequential assembly for a combinatorial construction problem.
- This line of research shows that it can successfully generate a 3D object in a combinatorial manner.
- We investigated three representations, i.e., sets, graphs, and voxels, and three approaches, i.e., Bayesian optimization, deep reinforcement learning, and feed-forward neural networks with efficient constraint satisfaction.
- Also, we created a new dataset for combinatorial 3D models, which allows us to generate 3D shapes sequentially.

University of Pittsburgh

Any Questions?

References I

S. Ahn, J. Kim, M. Cho, and J. Park. Budget-aware sequential brick assembly with efficient constraint satisfaction. *arXiv preprint arXiv:2210.01021*, 2022.

V. Bapst, A. Sanchez-Gonzalez, C. Doersch, K. Stachenfeld, P. Kohli, P. Battaglia, and J. Hamrick. Structured agents for physical construction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 464–474, Long Beach, California, USA, 2019.

P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

H. Chung, J. Kim, B. Knyazev, J. Lee, G. W. Taylor, J. Park, and M. Cho. Brick-by-Brick: Combinatorial construction with deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 5745–5757, Virtual, 2021.

S. Eilers. The LEGO counting problem. *The American Mathematical Monthly*, 123(5):415–426, 2016.

J. B. Hamrick, K. R. Allen, V. Bapst, T. Zhu, K. R. McKee, J. B. Tenenbaum, and P. W. Battaglia. Relational inductive bias for physical construction in humans and machines. In *Proceedings of the Annual Conference of the Cognitive Science Society (CogSci)*, pages 1773–1778, Madison, Wisconsin, USA, 2018.

S. Herman. *Building a History: The LEGO Group*. Grub Street Publishers, 2012.

J. Kim, H. Chung, J. Lee, M. Cho, and J. Park. Combinatorial 3D shape generation via sequential assembly. In *Neural Information Processing Systems Workshop on Machine Learning for Engineering Modeling, Simulation, and Design (ML4Eng)*, Virtual, 2020.

R. Thompson, G. Elahe, T. DeVries, and G. W. Taylor. Building LEGO using deep generative models of graphs. In *Neural Information Processing Systems Workshop on Machine Learning for Engineering Modeling, Simulation, and Design (ML4Eng)*, Virtual, 2020.

B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni. Learning object bounding boxes for 3D instance segmentation on point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 6740–6749, 2019.