NeurIPS 2021

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

Hyunsoo Chung*[1] Jungtaek Kim*[1] Boris Knyazev[2,3] Jinhwi Lee[1,4] Graham W. Taylor[2,3] Jaesik Park[1] Minsu Cho[1]

* Equal contribution [1]POSTECH [2]University of Guelph [3]Vector Institute [4]POSCO

Introduction

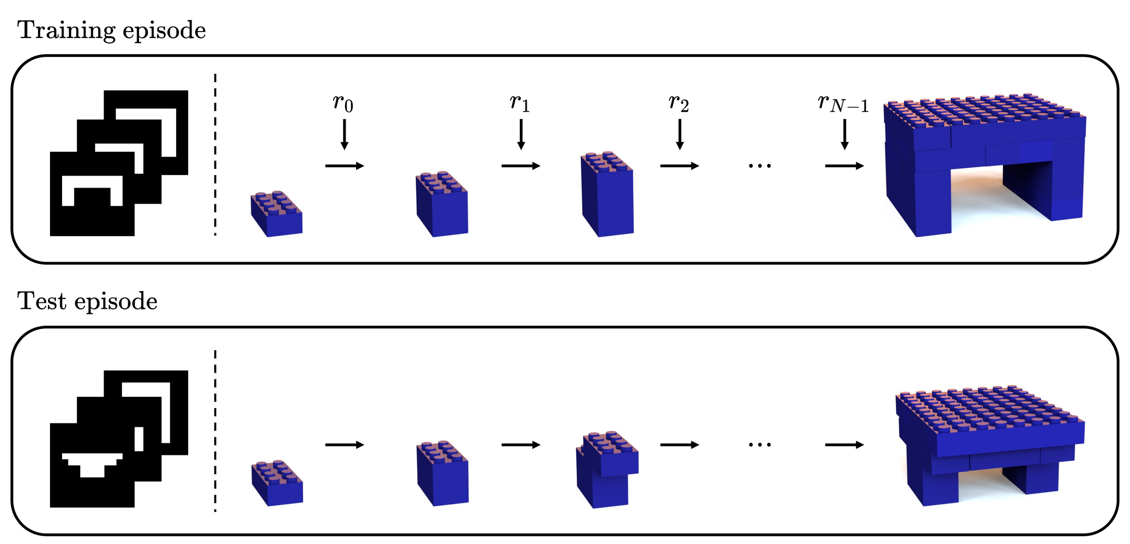
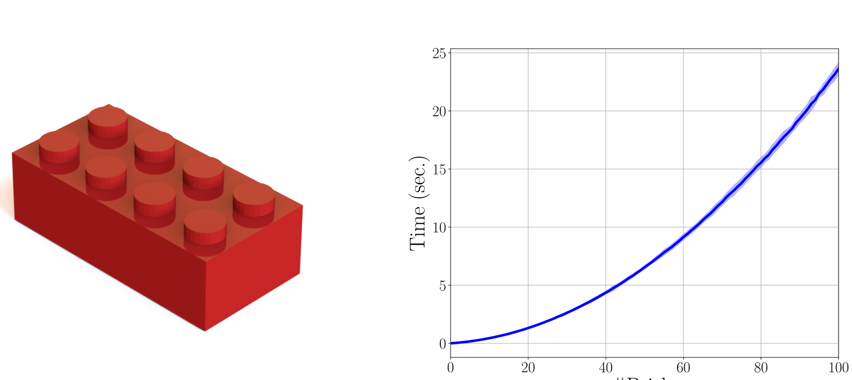


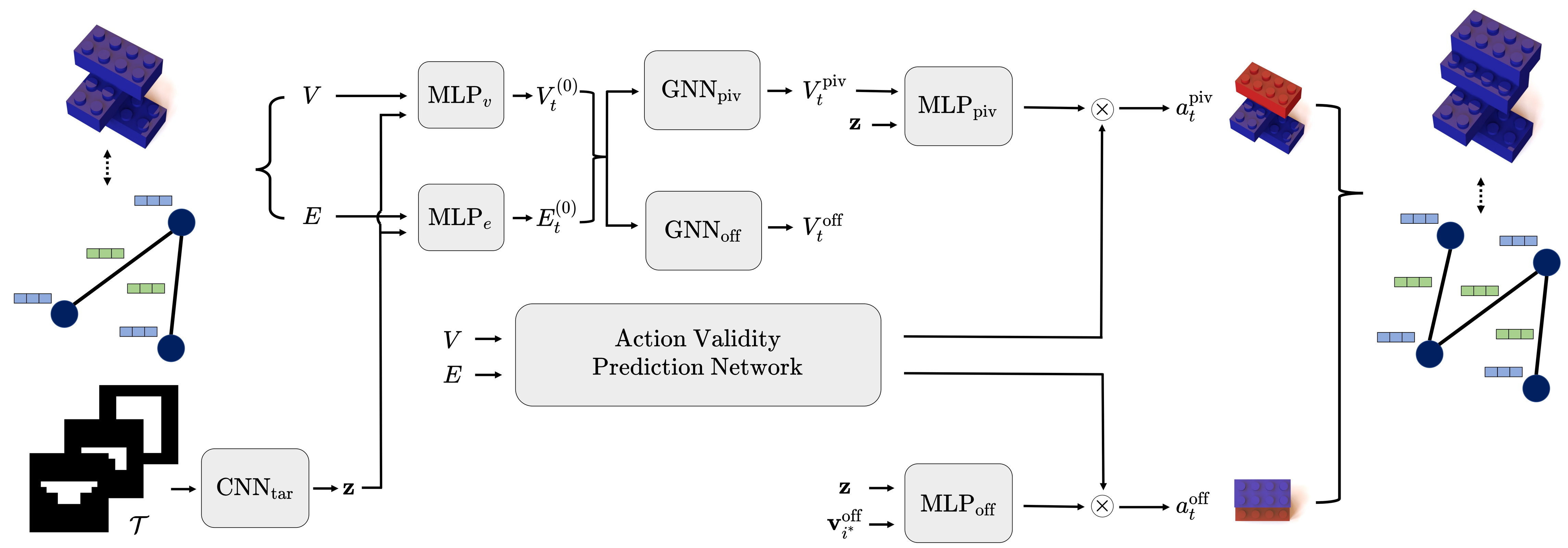
Figure 1: Training and test episodes

- Sequentially assemble unit primitives given incomplete target information, which is an interesting yet challenging problem defined on a combinatorial space.
- Propose an RL approach, which does not require sequence-level supervision.
- Adopt an action validity prediction network to handle both an indefinite action space and the existence of many invalid actions when applying RL.

Combinatorial Construction

- 2×4 brick as a unit primitive, which has eight studs and their fit cavities.
- Consistently growing possible positions to assemble a brick.
- For only six 2×4 bricks, 915,103,765 possible combinations.





Action Space

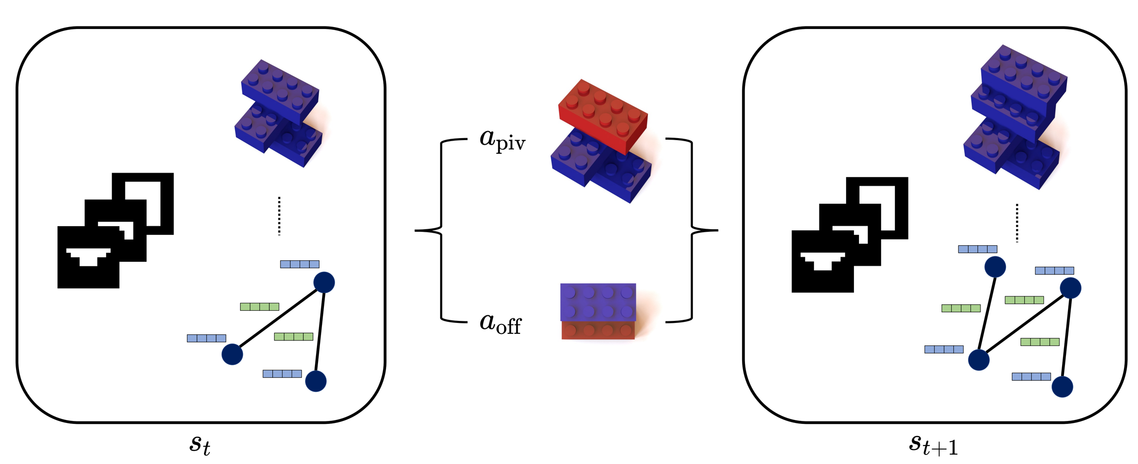
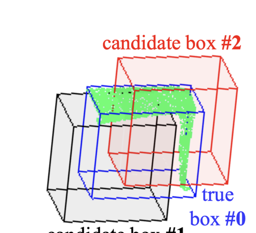


Figure 2: Successive action space

Assumptions

- Every connection type must follow a fixed rule.

- No bricks mutually overlap.



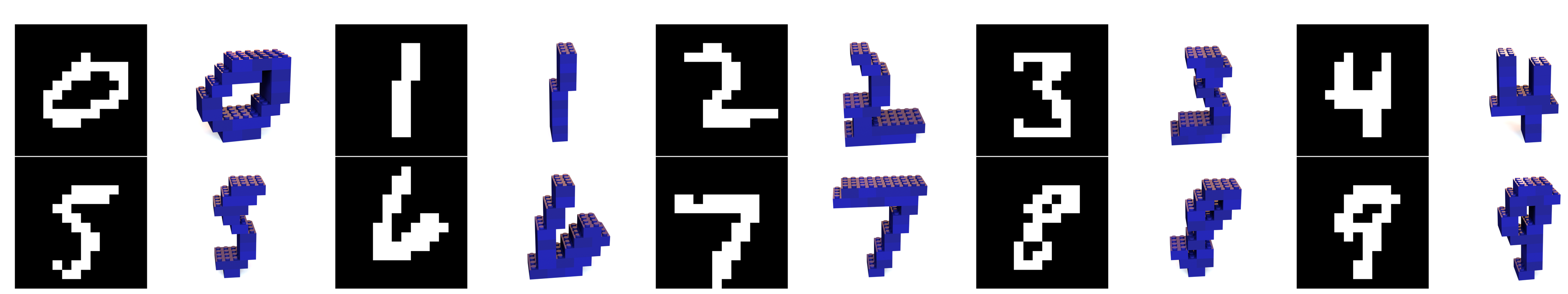


Figure 3: Qualitative results on MNIST construction

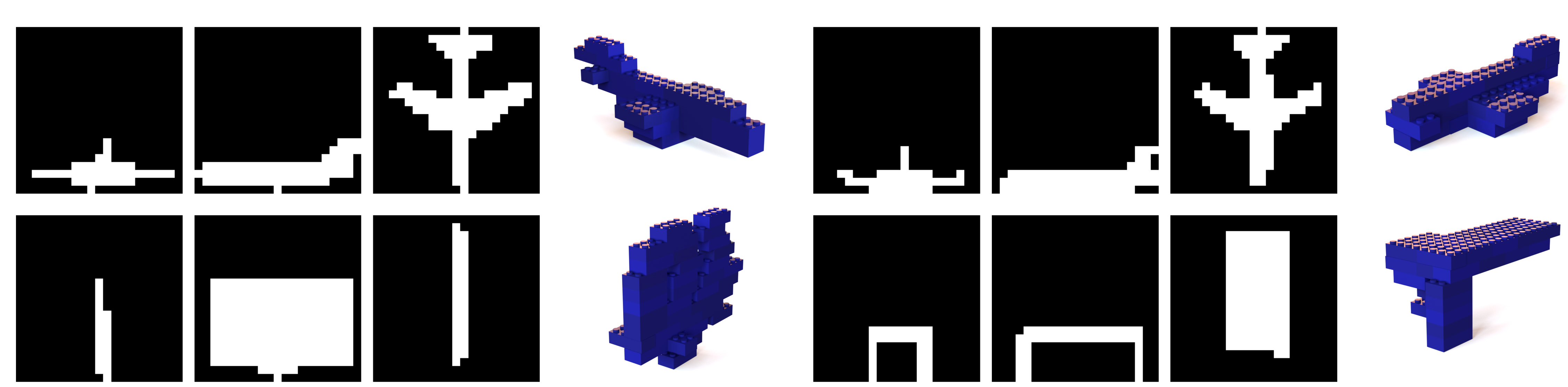


Figure 4: Qualitative results on ModelNet construction

Brick-by-Brick

- Suppose that target information $\mathcal{T}$ is given as partial information.
- Represent each t-th state s_t of the MDP as a tuple of a directed graph G_t and target information $\mathcal{T}$, i.e., $s_t = (G_t, \mathcal{T})$.
- With t bricks assembled, define an action $a_t = (a_t^{\text{piv}}, a_t^{\text{off}})$.
- Measure the overlap between the currently-assembled bricks and the target object as a reward function:

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

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.

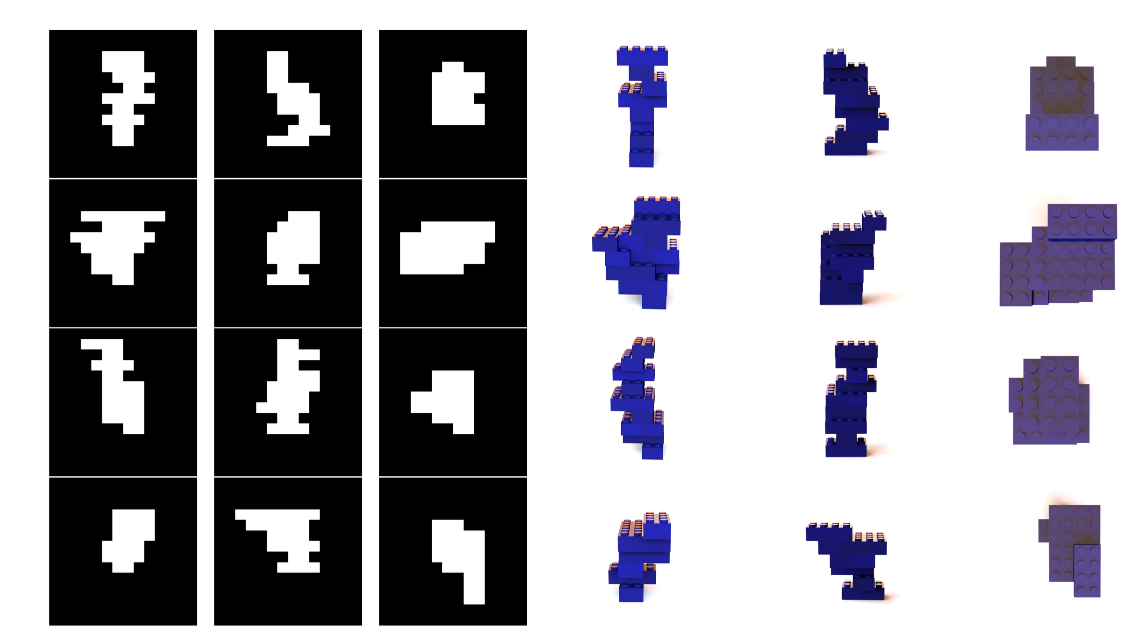


Figure 5: Results on randomly-assembled object construction

Available on arXiv/GitHub

arXiv GitHub