CO-INTELLIGENT DESIGN TOWARD MODULAR STRUCTURES

Imitation learning approach for spatial scenarios creation

MINGLI SUN[1], PETER BUŠ [2], JIANHAO CHEN[3], KENAN SUN[4]
[1]Shanghai International College of Design & Innovation, Tongji University.
[2]Institute of Future Human Habitats, Tsinghua Shenzhen International Graduate School.
[3]Bartlett School of Architecture, University College London (UCL).
[4]School of Architecture, Harbin Institute of Technology Shenzhen.
[1]2233747@tongji.edu.cn, 0009-0001-8998-3159
[2]peter_bus@sz.tsinghua.edu.cn, 0000-0002-1730-4559
[3]zczlj95@ucl.ac.uk, 0009-0007-7968-9857
[4]sknju0518@gmail.com, 0009-0003-0679-5887

Abstract. Modular structures with repeating configurations are a common type of building that consists of identical unit components arranged in a specific way. Design reasoning for such structures is an unstructured and unique process that relies on the designer's intuition and rational reasoning. However, repetitive design movements often become time-consuming and energy-draining. In the era of artificial intelligence, a crucial question is whether machines can replicate the design reasoning behaviours of human designers. This research aims to integrate the strength of human unique capabilities, like creativity, intuition, and design skills, with machine-emulating creativity, applied in the modular structure design while addressing production efficiency. It has shown that the Agent can mimic the designer's stacking approach to modular structure design by utilising Generative Adversarial Imitation Learning (GAIL) and Proximal Policy Optimization (PPO). Such a co-intelligent design method facilitates the creation of diverse modular structures.

Keywords. Design Reasoning, Co-intelligent Design, Modular Structures, Generative Adversarial Imitation Learning.

1. Introduction

1.1. MODULAR STRUCTURES WITH REPEATING CONFIGURATIONS

Modular structures, such as temporary buildings, art exhibits, and urban furniture, are widely used in the architectural field (Arup Associates, 2016; Ruiz et al., 2018; Taylor, 2015). It consists of unit components that possess identical or similar forms. A certain

ACCELERATED DESIGN, Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2024, Volume 3, 171-180.

number of unit components can generate various patterns through differentiated arrangements. Such constructions enable flexible adjusting to fit multiple site constraints by increasing or decreasing the number of unit components or changing their placement.

In various case studies investigating modular architectural design at different scales, we have observed that the heart of this series of studies lies in arranging modular components to create a solution that aligns with the designer's intention.

Throughout such processes, designers focus on the form of the components and their placements to achieve their design goals and accommodate specific spatial constraints(Düzenli et al., 2017; Kim et al., 2019), which is a typical process of reasoning about the structure itself (Akin, 1993). The designer's thinking and behaviour continuously influence the reasoning process until a modular construction is created.

1.2. DESIGN REASONING FOR MODULAR FORM

Like all architectural design processes, freehand sketches (Suwa & Tversky, 1997) and manipulation of objects in 3D modelling software are the primary ways of reasoning about modular structures. Such a visual reasoning process continues translating the designer's internal design cognition and intentions into external representations, such as sketches and 3D models (Oxman, 2001; Park et al., 2006).

However, it has been observed that designers often spend a lot of time and effort on repetitive movements during the design reasoning process (Chiu, 2003). For example, they may have to move unit components around repeatedly to reorganise their placement relationships. Although the development of digital drawing and modelling tools has significantly enhanced design productivity and creativity over the past century, designers still cannot avoid implementing such inefficient design movements.

Can machines collaborate with designers in the artificial intelligence (AI) era to enable design reasoning? Like any other art and design activity, modular structure design is a unique, non-linear process closely related to the designer's logical reasoning methods and intuition (Goldschmidt & Weil, 1998). The emphasis is that the intuitive process, stemming from individual experiences and ways of thinking that are difficult to capture and verbalise, is considered a unique capability of human designers (Raami, 2015). Therefore, a crucial question is whether machines can imitate the design reasoning behaviours of human designers.

Machine Learning is a subfield of Artificial Intelligence that has emerged in the domains of art and design, serving as a versatile tool, medium, and platform for designers (Ye, 2021). Our research utilises generative adversarial imitation learning (GAIL) (Ho & Ermon, 2016) to develop cognitive computing processes, Agent, that imitate the design reasoning behaviours of human designers. This process enables the generation of modular structures through a in co-intelligent design method that combines the unique capabilities of human designers with the computational strengths of machines. The efficient collaboration enables the creation of diverse modular structures.

2. Methodology

2.1. CO-INTELLIGENT DESIGN REASONING

To start, we introduce two concepts - Agent and Masterbuilder. Agent refers to computer programs, while Masterbuilder refers to human designers. Utilising the GAIL, the Agent can undergo self-training to acquire a policy model learned from the Masterbuilder's design reasoning trajectory while generating the modular spatial scenario based on an incremental growth sequence as a design outcome of imitation learning.

As a model-free imitation learning algorithm, the GAIL, first proposed by Jonathan Ho and Stefano Ermon, consists of Inverse Reinforcement Learning (IRL) and Generative Adversarial Network (GAN). Such a system involves a policy network and a discriminator network. IRL resolves the limitations of Reinforcement Learning, thus improving learning efficiency and generalization.

The Masterbuilder's reasoning trajectory describes the correspondence between the environment state (S) and the design action (A). The design policy π denotes the set of all stationary stochastic policies that take actions in A given states in S. The occupancy measure $\rho_{\pi H}$ (s, a) describes the data distribution of state-action pairs in the design reasoning trajectory. Subsequently, utilising the ML-Agents (Unity Technologies, 2020) connected to the PyTorch (Meta AI, 2016) environment (neural network training), The designer sets up the initial environment in Unity (Unity Technologies, 2005), and the Agent randomly initialises the policy network to interact with the environment and generate an inference trajectory under the current policy.

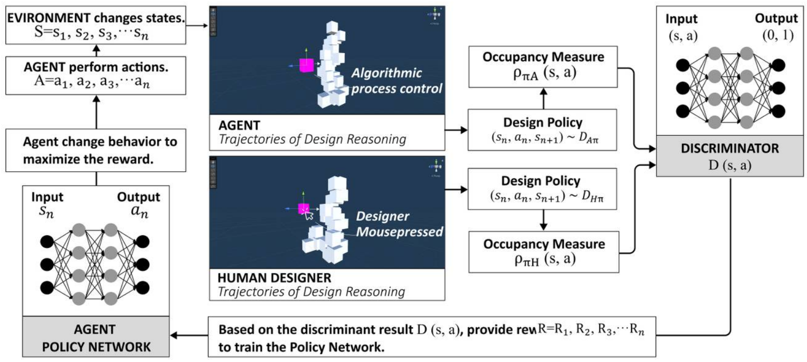

Figure 1. Create the Agent's cognitive process utilising the GAIL and PPO

Afterwards, as shown in Figure 1 above, the Masterbuilder's reasoning trajectories dataset trains the discriminator to distinguish the Agent design policy $(s_n, a_n, s_{n+1}) \sim D_{A\pi}$ and the human designer's $(s_n, a_n, s_{n+1}) \sim D_{H\pi}$. D (s, a) denotes the probability that the discriminator categorises the state-action pairs into expert-like regions. After defining a reward function based on the discriminator's prediction D (s, a), the Proximal Policy Optimization Algorithms (PPO) (Schulman, 2017) trains the policy network to maximise the reward by deceiving the discriminator. To maximise the reward Rn, the occupancy measure of the Agent's trajectory generated by the policy network tend to be similar to the Masterbuilder's trajectory - increasing the probability of classification as a Masterbuilder. Ultimately, when the discriminator cannot distinguish between the Agent's trajectory and the sample of the Masterbuilder's reasoning trajectory, it indicates that the Agent has learnt a similar reasoning strategy from the Masterbuilder's reasoning demonstration. As a result, the Agent can generate a modular spatial scenario based on an incremental growth sequence that meets the designer's intention while learning the policy.

2.2. CO-INTELLIGENT DESIGN METHOD

The proposed co-intelligent design methodology for modular structures consists of design, digital fabrication, and assembly processes, as shown in Figure 2.

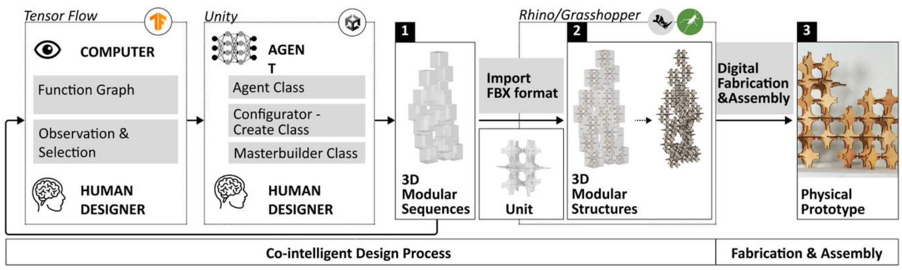

Figure 2. Co-intelligent design method towards modular structures

(1) Generation of modular spatial model based on incremental growth

The computational framework was established based on the Unity platform and ML-Agents tool, enabling designers to train AI utilising the Pytorch in the virtual environment controlled by the Anaconda tool. In Unity, interactive scripts are written to create a co-intelligent design system, including the Agent, Masterbuilder, and Configurator classes. In Unity, the Agent learns and imitates the human designer's design strategy while generating a modular spatial scenario in a sequence. After pre-training the policy model using the behavioural cloning approach before the GAIL implementation itself (the agent imitates precisely the provided demonstration by the Masterbuilder first and trains it during the first 1 million iterations out of 5 million), we observe and evaluate the GAIL process and training results through the visual charts on the TensorBoard (TensorFlow, 2015) to select the most satisfactory learning scene setup for the modular scenario generation sequentially.

(2) Form-finding of modular structures

The modular sequence spatial design generated in Unity is imported into the parametric modelling software such as Rhino and Grasshopper through FBX format. In parallel, Designers can customise the unit components based on unique design concepts in 3D modelling software. These units will then be inserted into each specific-sized cube of the modular template in Grasshopper, resulting in a 3D model of modular structures with repeating configurations.

(3) Digital fabrication and assembly

To provide a complete picture of the morphogenetic potential of the co-intelligent design method, we introduce the fabrication and assembly of physical models. Designers can choose from various digital fabrication methods, such as laser cutting and 3D printing, to produce and assemble modules that meet their specific requirements. Ultimately, modular structures are designed and produced efficiently.

3. Experiment

3.1. DESIGN SCENARIO BASED ON INCREMENTAL GROWTH

Based on the Unity platform and the ML-Agents tool, a computational system was built to enable designers to train the Agent using Pytorch in a virtual environment controlled by the Anaconda tool. Such a system consists of three classes of scripts, including the Masterbuilder, Agent, and Configuration classes. These create the logic and principles of the interaction process between the designer and the environment to generate a modular design scenario.

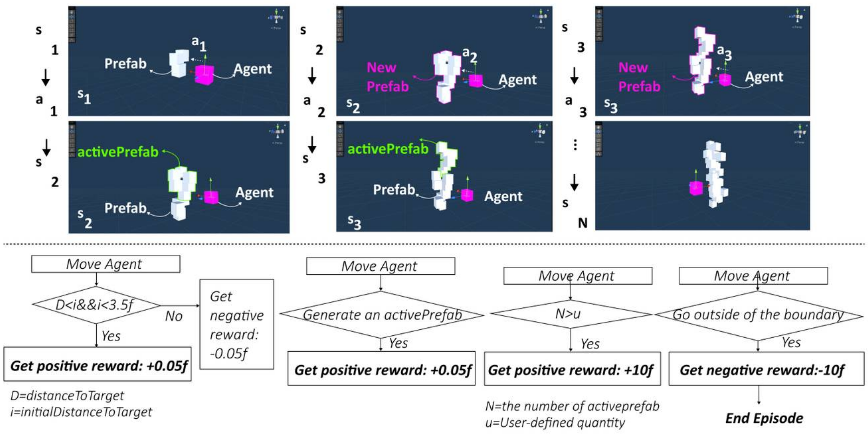

Figure 3. The policy to generate the modular spatial design scenario

First, two types of concepts during the interactive process are defined to articulate the interactive norms and underlying generation logic of the modular design scenario, as shown in Figure 3. Prefab refers to the entire cluster of building blocks, while activePrefab refers to the newly generated block in a growth sequence. In the Unity interface, we set two 1*1.5*1m blocks labelled Prefab as the initial building blocks. Meanwhile, a 1*1.5*1m block is randomly generated and tagged as the Agent at an arbitrary position away from Prefab—the script manipulates such a process. When the Agent collides with the initial building block (Prefab), a new set of building blocks will be added at that location according to the vector direction and the number of added blocks the designer defined in the script. The newly generated cluster is labelled as activePrefab. The entire set of building components is now referred to as Prefab. The Prefab and activePrefab clusters' state continuously changes as the Agent interacts with the environment. Eventually, the Prefab cluster generated is the modular spatial design scenario obtained.

The designer allows the use of a stacking method to design. The Environment Configuration Class and the Master Builder Class scripts enable the designer to move the Agent using the QWEASD keyboard keys and make it collide with the Prefab during the demonstration definition by the Masterbuilder before the training starts. It helps the designer control the growth location and level according to their design intent. Once the designer completes modular design through this interactive process, the Agent's trajectory contains a dataset of its movements and the state of the environment. This data will be used as a demonstration for the Agent default self-training.

The algorithmic system uses the Environment Configuration Class and Agent scripts to identify the values of the environment state. After undergoing the GAIL process, the policy network receives the state values and outputs action values, including direction and distance, as shown in Figure 3. These action values guide the Agent to move, and the collision Prefab generates new building blocks. In parallel, the designer creates a reward function representing the norms for different actions and states. The reward function's conditions are evaluated at three levels:

(a) Verify that activePrefab is generated in numbers that exceed the designer's specified threshold.

(b) Evaluating the distance between the Agent and the Prefabs.

(c) Verifying that the Prefabs do not cross the boundaries the designer defines.

Based on these criteria, the Agent mimics the design reasoning trajectories of the human designer and optimises the strategies in these actions to obtain the highest possible reward. After analysing the function graphs on TensorBoard, as illustrated in Figure 4, we discovered that the Agent learned strategies similar to the designer's. Meanwhile, a co-intelligent design process creates a modular spatial design scenario in an incremental growth sequence.

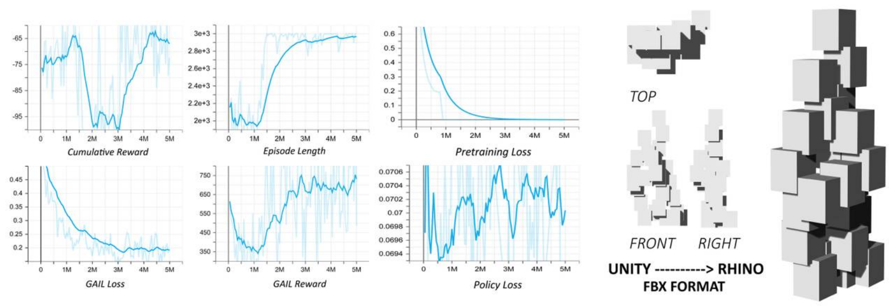

Figure 4. The evaluation of the training process on TensorBoard

3.2. FORM-FINDING OF MODULAR STRUCTURES

Firstly, we started with the module design in the 3D modelling software Rhino. After that, through the physical prototype testing, we composed the modules into modular units, which can be repeated in the modular design scenario in a growing sequence to enrich the internal structure. Then, in Grasshopper, the unit is populated in the Unity-generated modular scenarios through a parameterised approach and generates a 3D model of the modular structure.

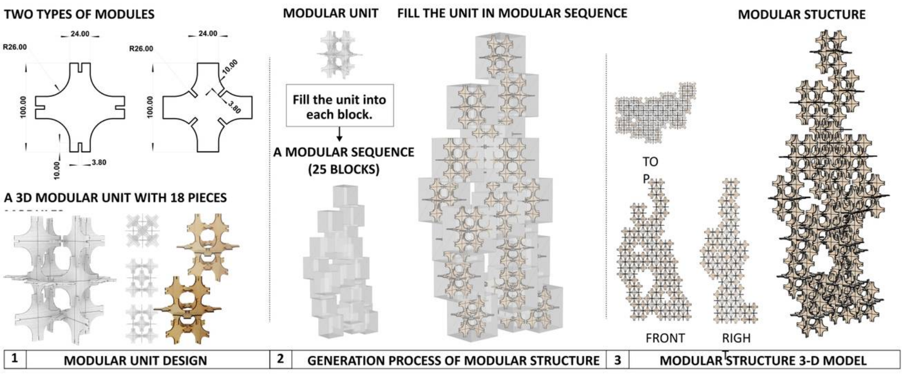

Figure 5. Form-finding of modular structures

We use a crisscross form with slots in two positions as the primary graphic of the module in the 3D modelling software, as shown in Figure 5. The slots are 10*3.6mm in size and have a width equal to the sheet thickness, ensuring a sturdy snap joint. We used laser-cut wooden boards to create these modules during the physical prototyping phase. We assembled them into model units that could be flexibly replicated in modular design in growing sequences. Considering the relationships between slots when designing modules and modular units is essential to avoid loose connections or collisions when populating the module sequence in the next stage. Eventually, we selected a modular unit that consisted of 18 module pieces and created a 3D model for the unit.

In the next step, we import a modular sequence model (the spatial template generated in Unity) in FBX format from Unity into Grasshopper. Then, we match the modular units into the sequence framework. Each block in the modular sequence is used as a bounding box, set to populate the correctly oriented modular unit. Therefore, we must ensure that the modular unit's boundary corresponds to each block boundary in scale and size. To achieve this, we extract the geometric centres of each block in all modular sequences along with the modular unit's geometric centres and orientation vectors. We can position and rotate the unit to ensure its centres align with the blocks. It allows us to generate the modular structure in an integrated way.

3.3. DIGITAL FABRICATION AND ASSEMBLY

Physical models were assembled to showcase the form-creating potential of co-intelligent design. Using the OpenNest plugin in Grasshopper (Vestartas, 2018), we flattened the modules of the entire structure onto a single board to optimise cutting materials. The designer utilised the laser cutting technique to fabricate modules using 4mm wooden panels. Then, based on the created 3D modular structure, we quickly assembled a physical prototype, as shown in Figure 6.

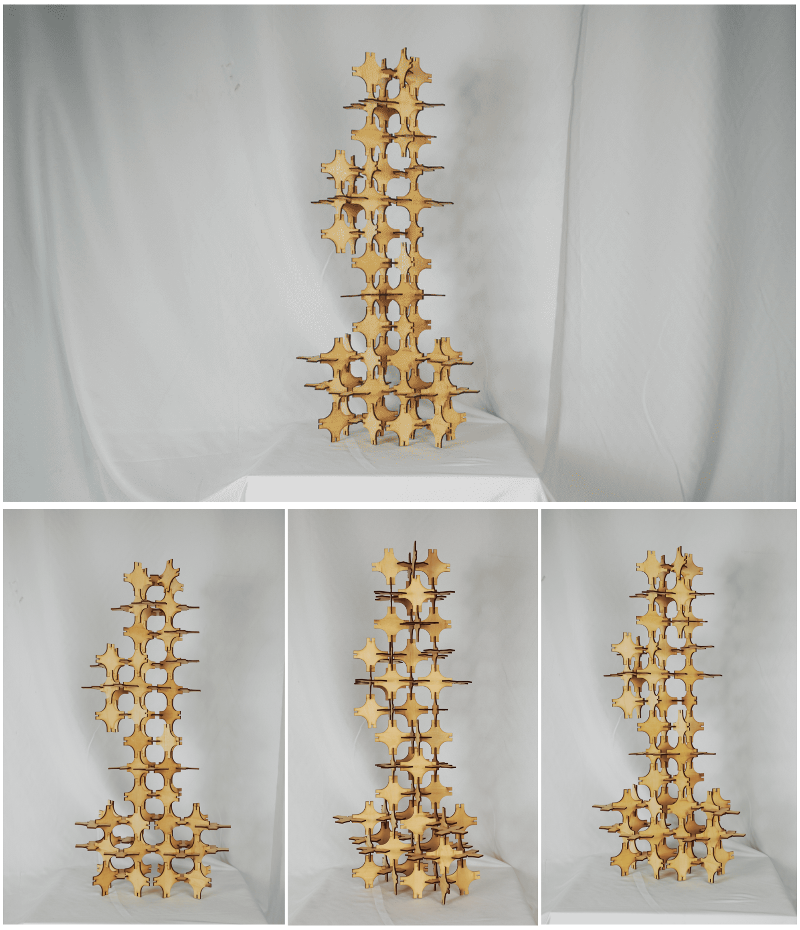

Figure 6. Modular structures

4. Discussion

4.1. ADVANTAGES

This study explores the potential benefits of combining human and machine participation in design reasoning, replacing focusing solely on competition. With the development and application of algorithms, the traditional design approach, which human designers have dominated, will undergo new impacts and changes. Our study proposes a methodology that leverages the ML-agent tool on Unity and PyTorch to develop an Agent's cognition, enabling it to imitate the designer's reasoning process. This methodology allows the Agent to access the human designer's reasoning strategy, even surpassing based on parameter settings.

Co-intelligent design reasoning combines human thought processes with algorithms to enhance the design process. The design process is often unstructured, which makes it challenging to identify a clear logic or strategy. We utilised the model-free imitation learning method GAIL employing the PPO algorithm to enable the agent to learn and adopt expert strategies. In this collaborative process, designers can express their design intentions in creative design tasks. In parallel, the agent can perform repetitive and time-consuming design movements instead of human designers.

The co-intelligent design process also enables the creation of forms that closely align with the designer's intentions. No matter the modular sequences generated by the Agent itself or the Agent interacting with the human designer, both demonstrate the method's potential for autonomous co-intelligent creation.

4.2. LIMITATIONS

We confirmed that the design methodology allows the Agent to access the human designer's reasoning strategies for arranging modular structures. However, further empirical research is required to understand the human thought process in modular design in depth. With this knowledge, cognitive computing techniques can facilitate a more comprehensive and cooperative design process between humans and machines.

We must continually enhance the scientific and quantitative evaluation of the co-intelligent design process and its outcomes to develop this approach in further research for more complex spatial design scenarios. This will enable the collaborative creative process by human designers and machines through computational methods to deliver a novel design to production space.

5. Conclusion

Our research focuses on the design reasoning process of modular structures. Based on general observations and the test experiment, designers' design reasoning is a non-structured and unique process with no apparent and intuitive patterns. Therefore, this study draws on the algorithmic process of model-free imitation learning, GAIL, which enables the Agent to summarise and learn the underlying design strategies from a human designer reasoning process and simultaneously generates the modular spatial design scenario. During the Agent's self-training process, the designer can observe the function graphs generated in TensorBoard to determine whether the Agent has learned the design strategies. It helps to ensure that the design reasoning process is well-collaborated. We use an integrated software platform to populate modular units into the spatial design scenario, creating modular structures with intricate internal design qualities. These modules are then fabricated using laser cutting and manually assembled efficiently. Our research enables the efficient collaborative design process between humans and machines to generate pattern-rich modular structures.

Our research examines collaborative design reasoning in the age of AI rather than a man/machine dichotomy. This co-intelligence design approach allows human designers' unique capabilities and machines' computational strengths to be more naturally combined to solve design problems jointly. Conducting detailed scientific and empirical research on human designers' cognitive and reasoning processes is essential for further research. It will help us achieve a more profound co-design process and enhance this approach's scientific validity and applicability.

References

Akin, Ö. (1993). Architects' reasoning with structures and functions. Environment and Planning B: Planning and Design, 20(3), 273–294. https://doi.org/10.1068/b200273

Arup Associates. (2016). Temporary Permanence Installation. Temporary Permanence Installation / Arup Associates. https://www.archdaily.com/786564/urban-public-spatial-installation-arup-associates?ad_medium=gallery

Chiu, M.-L. (2003). Design moves in situated design with case-based reasoning. Design Studies, 24(1), 1–25. https://doi.org/10.1016/S0142-694X(02)00007-8

Düzenli, T., Alpak, E. M., & Özkan, D. G. (2017). Transition from intangible to tangible in landscape architecture basic design course. European Online Journal of Natural and Social Sciences, 6(4), pp-516.

Goldschmidt, G., & Weil, M. (1998). Contents and Structure in Design Reasoning. Design Issues, 14(3), 85. https://doi.org/10.2307/1511899

Ho, J., & Ermon, S. (2016). Generative Adversarial Imitation Learning (arXiv:1606.03476). arXiv. https://doi.org/10.48550/arXiv.1606.03476

Kim, J., Ryu, R., & Kim, Y. (2019). A Basic Study on Unit Module Planning for the Design Diversity of Modular Architecture. Asia-Pacific Journal of Convergent Research Interchange, 5(1), 31–45. https://doi.org/10.21742/apjcri.2019.03.04

Meta AI. (2016). Get started with PyTorch. https://pytorch.org/

Oxman, R. (2001). Digital Media and Design Didactics in Visual Cognition. eCAADe 2001: Architectural information management. https://doi.org/10.52842/conf.ecaade.2001.186

Park, J. A., Kim, Y. S., & Cho, J. Y. (2006). Visual Reasoning as a critical attribute in design creativity. Proceedings of International Design Research Symposium, 2006.

Raami, A. (2015). Intuition unleashed: On the application and development of intuition in the creative process. School of Art and Design. https://urn.fi/URN:ISBN:978-952-60-6108-5

Ruiz, J. V., Toshiki, H., & Hirayama, T. (2018). Kodama. Kengo Kuma and Associates. https://kkaa.co.jp/en/project/kodama/

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms (arXiv:1707.06347). arXiv. http://arxiv.org/abs/1707.06347

Suwa, M., & Tversky, B. (1997). What do architects and students perceive in their design sketches? A protocol analysis. Design Studies, 18(4), 385–403. https://doi.org/10.1016/S0142-694X(97)00008-2

Taylor, J. (2015). Populous Creates Eames-Inspired Installation for World Architecture Festival London. ArchDaily. https://www.archdaily.com/769325/populous-creates-eames-inspired-installation-for-world-architecture-festival-london

TensorFlow. (2015). Get started with TensorBoard. A suite of visualization tools to understand, debug, and optimize TensorFlow programs for ML experimentation. https://www.tensorflow.org/tensorboard/get_started

Unity Technologies. (2005). Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine. Unity. https://unity.com

Unity Technologies. (2020). Unity Machine Learning Agents | Train and embed intelligent agents by leveraging state-of-the-art deep learning technology. https://unity.com/products/machine-learning-agents

Vestartas, P. (2018). OpenNest—2D Polyline Packing for fabrication such as laser or CNC cutting. Food4Rhino. https://www.food4rhino.com/en/app/opennest

Ye, C. (2021). Evolution and Application of Artificial Intelligence Art Design Based on Machine Learning Algorithm. 2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE), 293–297. https://doi.org/10.1109/ICISCAE52414.2021.9590775