DEEP REINFORCEMENT LEARNING FOR FURNITURE LAYOUT SIMULATION IN INDOOR GRAPHICS SCENES

Xinhan Di*
Digital Technology Department
Longfor Group Holdings Limited
dixinhan@longfor.com

Pengqian Yu*
IBM Research
Singapore
peng.qian.yu@ibm.com

ABSTRACT

In the industrial interior design process, professional designers plan the size and position of furniture in a room to achieve a satisfactory design for selling. In this paper, we explore the interior graphics scenes design task as a Markov decision process (MDP), which is solved by deep reinforcement learning. The goal is to produce an accurate layout for the furniture in the indoor graphics scenes simulation. In particular, we first formulate the furniture layout task as a MDP problem by defining the state, action, and reward function. We then design the simulated environment and deploy a reinforcement learning agent that interacts with the environment to learn the optimal layout for the MDP. We conduct our experiments on a large-scale real-world interior layout dataset that contains industrial designs from professional designers. Our numerical results demonstrate that the proposed model yields higher-quality layouts as compared with the state-of-art model.

1 INTRODUCTION

People spend plenty of time indoors such as the bedroom, living room, office, and gym. Function, beauty, cost, and comfort are the keys to the redecoration of indoor scenes. Many online virtual interior tools are developed to help people design indoor spaces in the graphics simulation. Machine learning researchers make use of the virtual tools to train data-hungry models for the auto layout (Dai et al., 2018; Gordon et al., 2018), including a variety of generative models (Fisher et al., 2012; Li et al., 2019b).

Reinforcement learning (RL; Sutton & Barto (2018)) consists of an agent interacting with the environment, in order to learn an optimal policy by trial and error for the Markov decision process (MDP; Puterman (2014)). The past decade has witnessed the tremendous success of deep reinforcement learning in the fields of gaming, robotics and recommendation systems (Gibney, 2016; Schrittwieser et al., 2020; Silver et al., 2017). Researchers have proposed many useful and practical algorithms such as DQN (Mnih et al., 2013) that learns an optimal policy for discrete action space, DDPG (Lillicrap et al., 2015) and PPO (Schulman et al., 2017) that train an agent for continuous action space, and A3C (Mnih et al., 2016) designed for a large-scale computer cluster. These proposed algorithms solve stumbling blocks in the application of deep RL in the real world.

The models proposed in previous work (e.g., Fisher et al. (2012); Li et al. (2019b); Qi et al. (2018); Wang et al. (2018)) for furniture layout only produce approximate size of the furniture, which is not practical for industry use as illustrated in Figure 1. Moreover, these prior work neglect the fact that the industrial interior design process in the simulated graphics scenes is indeed a sequential decision-making process, where professional designers need to make multiple decisions for furniture layout by trial and error.

Motivated by the above challenges, we formulate the furniture layout task as a MDP problem by defining the state, action, and reward function. We then design a simulated environment and deploy a RL agent to learn the optimal layout for the MDP. Our work is related to data-hungry methods for synthesizing indoor graphics scenes simulations through the layout of furniture. Early work in the scene modeling implement kernels and graph walks to retrieve objects from a database (Choi

*Denotes equal contribution.

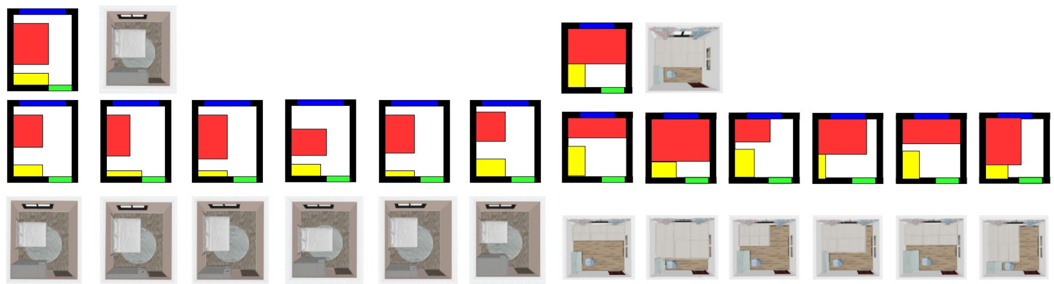

Figure 1: Examples of layouts produced by the state-of-the-art models (Wang et al., 2019). These layouts are for bedroom and tatami room. The ground truth layout in the simulator and the real-time renders can be found in the first row. The layouts produced by the state-of-art models are shown in the second and third rows. It can be observed that the state-of-art model produces inaccurate position and size of the furniture.

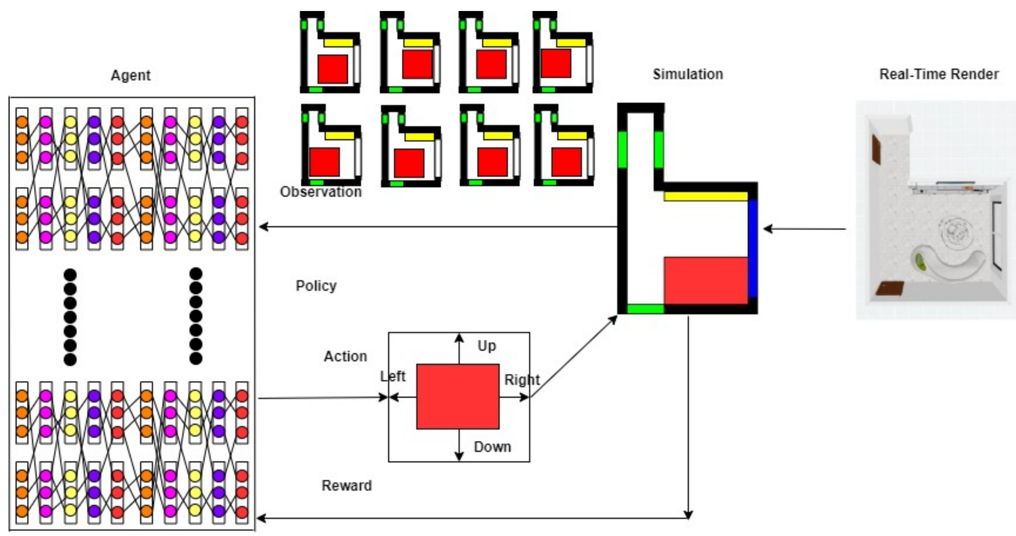

Figure 2: Simulation environment for the layout of furniture in the indoor scenes.

et al., 2013; Dasgupta et al., 2016). The graphical models are employed to model the compatibility between furniture and input sketches of scenes (Xu et al., 2013). Besides, an image-based CNN network is proposed to encoded top-down views of input scenes (Wang et al., 2018). A variational auto-encoder is applied to learn similar simulations (Zhang et al., 2020; Li et al., 2019a; Jyothi et al., 2019b). Furthermore, the family for the simulation of indoor graphic scenes in the form of tree-structured scene graphs is studied (Li et al., 2019b; Wang et al., 2018; 2019). However, this family of models produces inaccurate size and position for the furniture layout.

We highlight our two main contributions. First, we formulate the interior graphics scenes design task as a Markov decision process problem. To the best of our knowledge, this is the first time that the task is studied from a sequential decision-making perspective. Secondly, we develop an indoor graphics scenes simulator and use deep reinforcement learning technique to solve the MDP in the learning of the simulated graphic scenes. The developed simulator and codes are available at `https://github.com/CODE-SUBMIT/simulator1`.

2 Problem Formulation

We formulate the planning of furniture layout in the simulation of graphics indoor scenes as a Markov decision process (MDP) augmented with a goal state G that we would like an agent to learn. We define this MDP as a tuple (S, G, A, T, γ), in which S is the set of states, G is the goal,

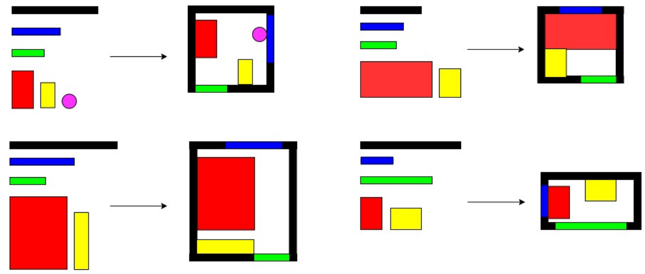

Figure 3: Given the sizes and positions of the walls, windows, doors and furniture in a real room, the developed simulator transfers the real indoor scenes to simulated graphics indoor scenes. Different components are in different colors in the simulation.

A is the set of actions, T is the transition probability function in which $T(s, a, s^{'})$ is the probability of transitioning to state $s^{'} \in S$ when action $a \in A$ is taken in state $s \in S$, R_t is the reward function at timestamp t, and γ is the discount rate $\gamma \in [0, 1)$. At the beginning of each episode in a MDP, the solution to a MDP is a control policy $\pi : S, G \rightarrow A$ that maximizes the value function $V_\pi(s, g) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s, g = G]$ for given initial state s_0 and goal g.

In this paper, we define the states $S = (S_{\text{w}}, S_{\text{win}}, S_{\text{d}}, S_{\text{f}})$ where S_{w} consists of the geometrical position p and the size l of the wall. Here p is the location for the center of the wall, and the size l consists of the width and height of the wall. S_{win}, S_{d} and S_{f} are defined similarly for the window, door and furniture. The goal state G describes the correct position p and direction d of the furniture. The action set A is discrete (left, right, up or down) and it specifies the direction of the furniture should move towards in the indoor scenes. The reward function is designed to encourage the furniture to move towards the correct position. It is defined as the following:

$$R := \theta \, \text{IoU}(f_{\text{target}}, f_{\text{state}})$$

where θ is a positive constant, f_{target} and f_{state} represent the ground truth and the current layouts for the furniture in the indoor scenes. IoU computes the intersection over union between f_{state} and f_{target}. The state, action, and reward are illustrated in Figure 2.

3 Simulation environment

Given the sizes and positions of the walls, windows, doors and furniture in a real room, we develop a simulator to transfer the real indoor scenes to simulated graphics indoor scenes as illustrated in Figure 3. The RL environment is a simulator $\mathcal{E}$ where $(s', R) = \mathcal{E}(s, a)$, $a \in A$ is the action from the agent in the current state $s \in S$, $s' \in S$ is the next state and R is the reward associated with the action a. In the next state s', the geometrical position and size of walls, doors, and windows are not changed, and only the geometrical position of the furniture is updated according to the action. Recall that the action space is discrete, and the center of the furniture can move left, right, up and down in each timestamp. The simulator $\mathcal{E}$ drops the action if furniture moves out of the room, and the furniture will stay in the same place. As the action space is discrete, we adapt the DQN algorithm (Mnih et al., 2013) for the RL agent in the simulation. The agent network contains three convolution layers with $8, 16$, and 32 hidden features and 2 fully connected layers with 512 and 4 hidden features. We train the agent using 9 episodes where each episode the agent is initialized at random states in the indoor simulation environment.

4 Experiments

We discuss the qualitative and quantitative results in this section. In our numerical experiments, four main types of indoor rooms including the bedroom, bathroom, study room and kitchen are evaluated.

We compare our proposed model with the state-of-art models including the PlanIT (Wang et al., 2019) and the LayouGAN (Li et al., 2019a). Note that we do not compare with layoutVAE (Jyothi et al., 2019a) and NDN (Lee et al., 2020) since they generates outputs in a conditional manner. For

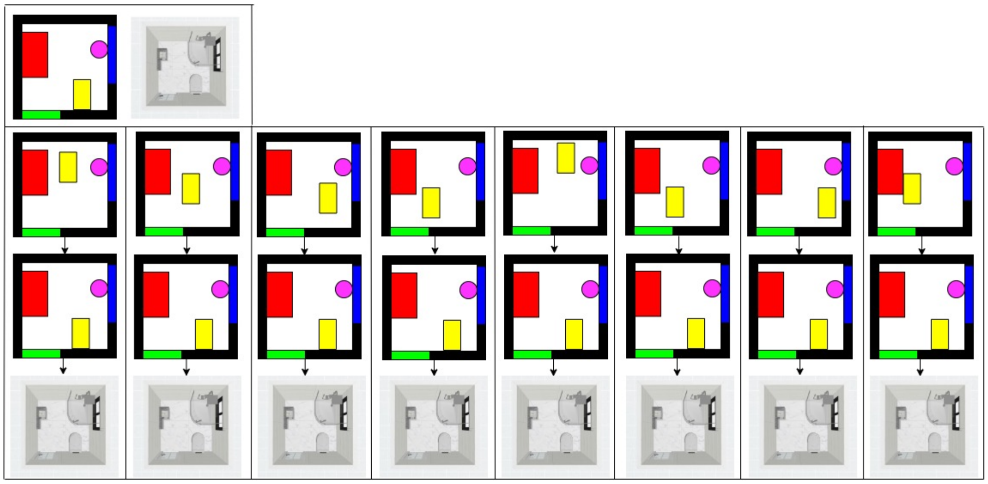

Figure 4: Given a bathroom with random furniture positions, the trained RL agent is able to produce a good layout for the bathroom graphics scenes. The first row represents the the ground truth layout for a bathroom in the simulation and its corresponding render. The second row represents the bathroom with random furniture positions. The third row represents the final layouts produced by the proposed method. The fourth row represents the corresponding layout renders.

Table 1: IoU scores for various models.

Room	PlanIT	LayoutGAN	Ours
Bathroom	0.623 ± 0.008	0.651 ± 0.010	0.961 ± 0.014
Bedroom	0.647 ± 0.009	0.648 ± 0.017	0.952 ± 0.026
Study	0.619 ± 0.006	0.662 ± 0.014	0.957 ± 0.018
Kitchen	0.637 ± 0.006	0.639 ± 0.012	0.948 ± 0.049

each room, we test the performance of the proposed model in the environment with $2,000$ random starting points. We train on $5,000$ samples for each type of rooms and test on $1,000$ samples for the corresponding type of rooms. We use the intersection over union (IoU) to measure of the intersection between the predicted layout and the ground truth layout. The comparison for different models is shown in Table 4.

We also conduct a two-alternative forced-choice (2AFC) perceptual study to compare the images from generated scenes with the corresponding scenes from the sold industrial solutions. The generated scenes are generated from our models, PlanIT (Wang et al., 2019) and LayouGAN (Li et al., 2019a), respectively. Ten professional interior designers were recruited as the participants. The scores are shown in Table 2, and the examples generated by our proposed method are shown in Figure 4. It can be concluded that our proposed approach outperforms the state-of-art models.

Table 2: Percentage ($\pm$ standard error) of 2AFC perceptual study for various models where the real sold solutions are judged more plausible than the generated scenes.

Room	PlanIT	LayoutGAN	Ours
Bathroom	79.61 ± 4.12	78.23 ± 6.17	61.08 ± 2.71
Bedroom	75.29 ± 3.89	79.15 ± 4.93	69.35 ± 2.83
Study	77.42 ± 5.92	82.03 ± 4.52	62.74 ± 5.72
Kitchen	78.13 ± 6.92	83.17 ± 5.98	64.51 ± 2.79

5 Discussion

In this paper, we tackle the interior graphics scenes design problem by formulating it as a sequential decision-making problem. We further solve the problem using a deep reinforcement learning agent that learns to produce high quality layout. It is worthwhile to extend our work and consider planning for multiple furniture layouts or learning in 3D space.

References

Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3d geometric phrases. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. Example-based synthesis of 3d object arrangements. *ACM Trans. Graph.*, 31(6), November 2012. ISSN 0730-0301. doi: 10.1145/2366145.2366154. URL `https://doi.org/10.1145/2366145.2366154`.

Elizabeth Gibney. Google ai algorithm masters ancient game of go. *Nature News*, 529(7587):445, 2016.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *International Conference on Computer Vision (ICCV)*, 2019a.

Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. Layoutvae: Stochastic scene layout generation from a label set. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019b.

Hsin-Ying Lee, Weilong Yang, Lu Jiang, Madison Le, Irfan Essa, Haifeng Gong, and Ming-Hsuan Yang. Neural design network: Graphic layout generation with constraints. In *Proceedings of European Conference on Computer Vision (ECCV)*, August 2020. doi: 10.1007/978-3-030-58580-8_29.

Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. *CoRR*, abs/1901.06767, 2019a. URL `http://arxiv.org/abs/1901.06767`.

Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM Trans. Graph.*, 38(2), February 2019b. ISSN 0730-0301. doi: 10.1145/3303766. URL `https://doi.org/10.1145/3303766`.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Kai Wang, Manolis Savva, Angel X. Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201362. URL `https://doi.org/10.1145/3197517.3201362`.

Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X. Chang, and Daniel Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Trans. Graph.*, 38(4), July 2019. ISSN 0730-0301. doi: 10.1145/3306346.3322941. URL `https://doi.org/10.1145/3306346.3322941`.

Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Trans. Graph.*, 32(4), July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461968. URL `https://doi.org/10.1145/2461912.2461968`.

Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. Deep generative modeling for scene synthesis via hybrid representations. *ACM Trans. Graph.*, 39(2), April 2020. ISSN 0730-0301. doi: 10.1145/3381866. URL `https://doi.org/10.1145/3381866`.