An Approach to Generate Multi-Alternative Proposals of Architectural Plan by Semi-Supervised Learning

Chunshuo H.
Fengchia University, Taiwan
jerrycat1030@gmail.com

Abstract. We present a GAN-based model of semi-supervised machine learning that utilizes a rapid annotation algorithm called space annotation algorithm (SAA) to generate architectural plans with diverse scenarios. Through the application of SAA, we can produce space topology information with in-depth features encapsulated in learning data for machine learning. The space topology information includes spatial composition, relationships, and scale hierarchy. More specifically, our machine-learning model can provide a sequence of architectural plans with semantic information to help the architect, or the designer determine the design strategy. The paper demonstrates that our machine-learning model is flexible and editable. Without further training, we can generate diverse proposals for building plans.

1. Introduction

Machine learning (ML) has considerable development influences in all industrial fields. However, the lack of a design dataset is the main obstacle to slowing down ML development in the architectural design field. To improve the dilemma, we use semi-supervised ML to solve the problem of insufficient datasets. Moreover, the other reason we adopt semi-supervised ML is that semi-supervised ML allows the machine to save training time and enables the machine automatically annotates the labels of architectural elements and spatial features in this research. Thus, one of the primary contents of the research is the automatic annotation method.

The development of the automated labeling software "Labellmg" has allowed engineers to save a lot of labeling time on the concrete objects on learning materials. However, the object recognition work of data labeling still relies on manually selecting tangible things in the image for labeling. For this research, "Labellmg" does not meet our research needs; because, in the learning of the architectural plan, except for perceiving "what is space," the machine also needs to classify "the architectural elements" and understand "the spatial topology relationship." In other words, machine need to label the abstractive elements of learning materials. To achieve the targets, we create three algorithms assisting the machine in labeling and generating the learning data.

1. **Recognition Training Algorithm (RTA)**: Auxiliary research converts the two-dimensional drawings of learning materials into three-dimensional vector space; then, the machine can classify the columns, walls, openings, and other elements on the building plan and extract mutual vector relationship information (C. S. Hsiao: 2021).

2. **Spatial Annotation Algorithm (SAA)**: To rapidly annotate spatial topology relationships, the room, and the corridor in the learning data.

3. **Switch Route Algorithm (SRA):** To enable ML generating "diversity design proposals"

According to the above research targets and content, we build the framework of this research in Figure 1:

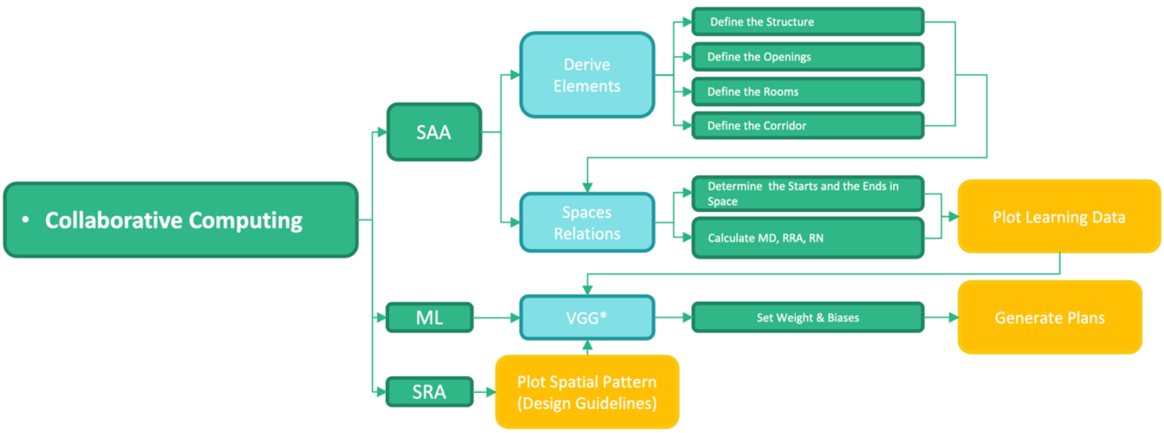


Figure 1: Research Framework and Workflow

2. Preparing Learning Data

Several related studies have been made in recent years, such as Morteza (2022), Carta (2021), Stanislas (2020) and Huang et al. (2018) applying pix2pixHD to generate architectural drawings and mark rooms with different colours to reorganize other functional divisions within the plan drawings. The related studies also inspire us to use three quantitative values (Rn: Local Integration, RRA: Real Relative Asymmetry, MD: Mean Depth) of space syntax to label the learning materials' abstractive elements and evaluate the learning result of ML. Theoretically, Rn is a relative value for the researcher to understand if the space is easier or more convenient to visit; in other words, it is a benchmark to depict the close relationship of neighbouring areas. RRA is the value of evaluating the spatial integration relationship of the building plan. MD is a benchmark to identify the feature of spatial configuration. Through the argument of Richard (2013) & Pelin (2007, 2022) that quantitative values are used to interpret spatial topology relationships, we build a "Spatial Annotation Algorithm (SAA)" through the method of the walking simulator to deduce the data analysis of MD, Rn, and RRA (see Figure 2).

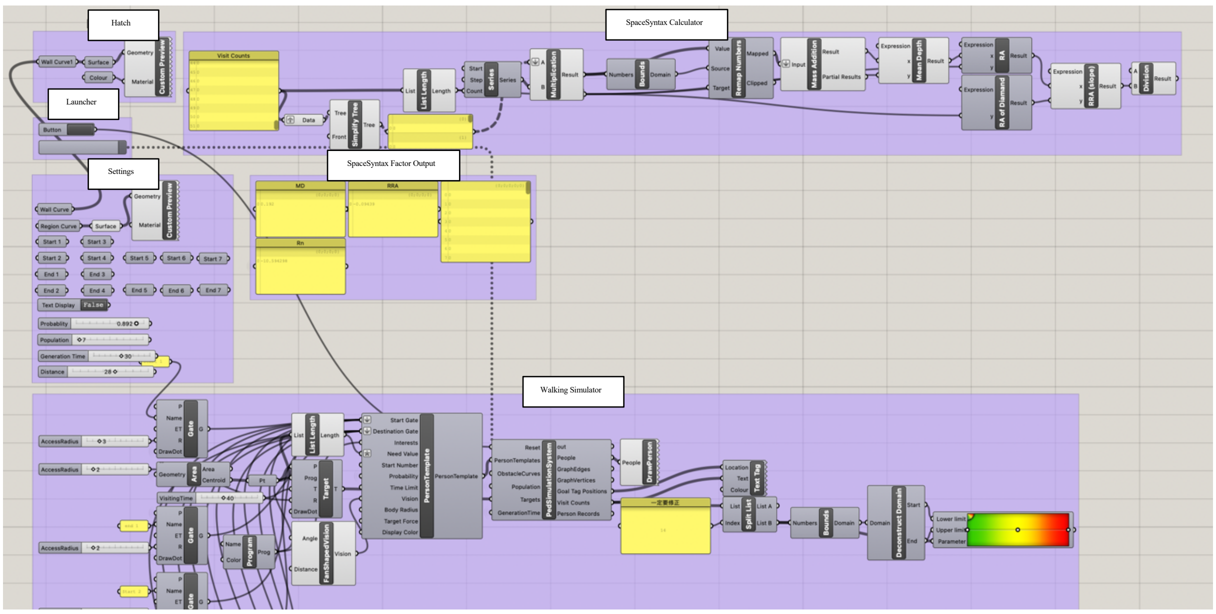


Figure 2: SAA and SpaceSyntax Calculator

2.1 Spatial Annotation Algorithm (SAA)

SAA is an automated labeling tool that involves quite a complicated variable design, including the control variable (spatial tolerance of crowd, the social distance of groups, the crowd's movement speed, and the visual distance that affects movement) and dependent variable (the location and quantity of start and end points). Moreover, according to the opinion of Bill Hillier (1996) and Akkelies & Yamu (2021), if SAA can operate adequately, no matter where the start and end points are, the quantitative values MD and RRA should be fixed, since the spatial relationships and connectivity patterns are the same. However, the layout of the start and end points would affect the spatial accessibility to make "Local Integration (Rn)" have differences. To examine the potential of SAA, we use OMA's Garage Museum of Contemporary Art (GMCA) as a case study. After converting GMCA plan to the vector model and obtaining "architectural elements" and "space," we use SAA to locate the different amounts of GMCA's entrances presented as the experimental model in Figure 3 and Figure 4.

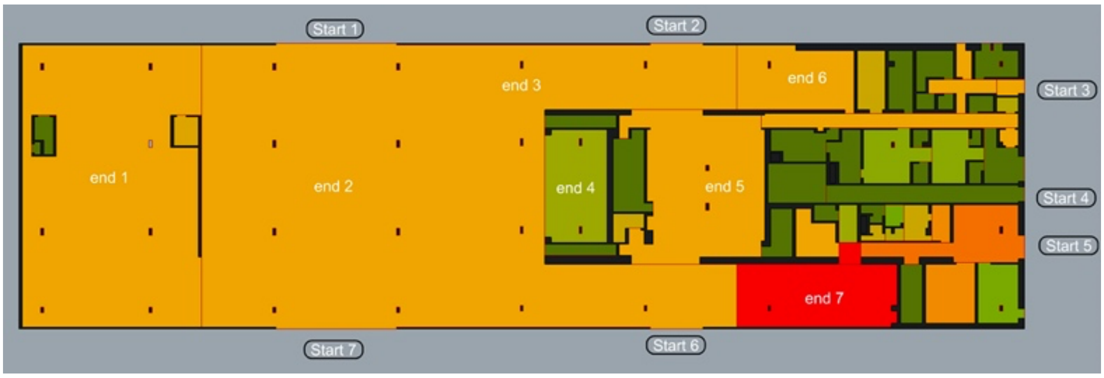


Figure 3: Quick Labelling the Relative Spatial Relationship for 1st Scenario via SAA

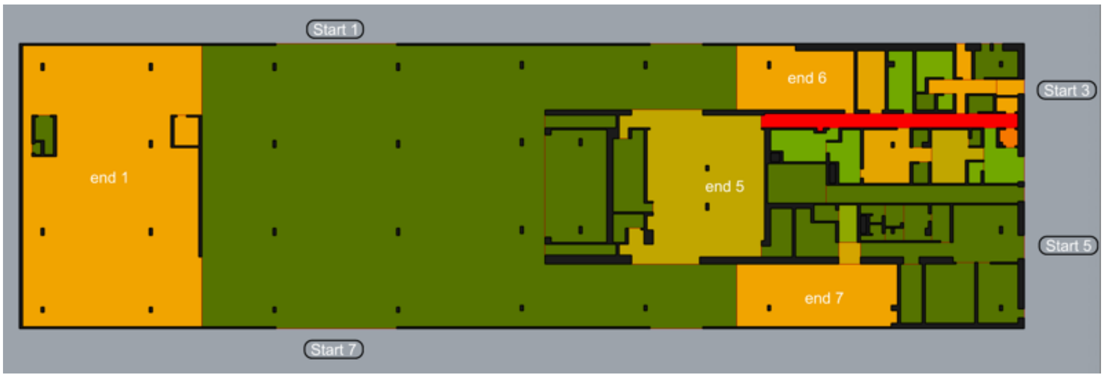


Figure 4: Quick Labelling the Relative Spatial Relationship for 2nd Scenario via SAA

In the experimental model, we have two scenarios for entering the building; in the first scenario, we open all the entrances of GMCA and locate the seven destinations based on the primary spatial functions of GMCA, such as the reception, lobby, cloakroom, bookshop, coffee bar, and laboratory (Figure 3). As for the second scenario, we close the three public entrances and relocate the destination (Figure 4). In these two scenarios, we set the control variable as follows, (1) The time of walking simulation: 330s, (2) The spatial tolerance of crowd: 40 people, (3) The social distance of groups: 0.35 units, (4) The crowd's movement speed: 20 units, (5) The visual distance that affects movement: 12 units. Through the operation of SAA, we get the quantitative values of MD, RRA, and Rn; then turn Rn to the decimal colour code to present the relative relationship of the neighbouring space. We list the quantitative values below for comparison.

- **Figure 3: MD=1.75, RRA=0.08, Rn=15.26**
- **Figure 4: MD=1.73, RRA=0.08, Rn=11.69**

This result shows that the quantitative calculation positively corresponds to the SpaceSyntax definition; consequently, we can trust that the decimal colour code plotted via SAA has certain credibility in quickly labeling the relative spatial relationships of neighbouring spaces. After we can finish the labels of learning data, to enable ML to generate "diversity design proposals," we need to create a switch route algorithm (SRA) to accomplish the goal.

2.2 Switch Route Algorithm (SRA)

ML excels in many areas with well-defined goals. However, there are areas where objective evaluations are unavailable, such as whether a configuration of the building plan reflects aesthetics. Thus, in our research framework, we need to provide the rules like "Design Guidelines," which have information on "Space Topology Relationship" and "Spatial Composition.” Also, the design guidelines have other targets of how to help ML avoid the limitation of the decision tree model and generate diverse proposals for the building plan. Thus, it is necessary to establish a mechanism like a switch route to affect the generative results of ML. The switch route can control the decision tree giving different results under the same spatial composition but other relative spatial relationships. Based on this concept, we create a switch route algorithm (SRA) and demonstrate it in Figure 5.

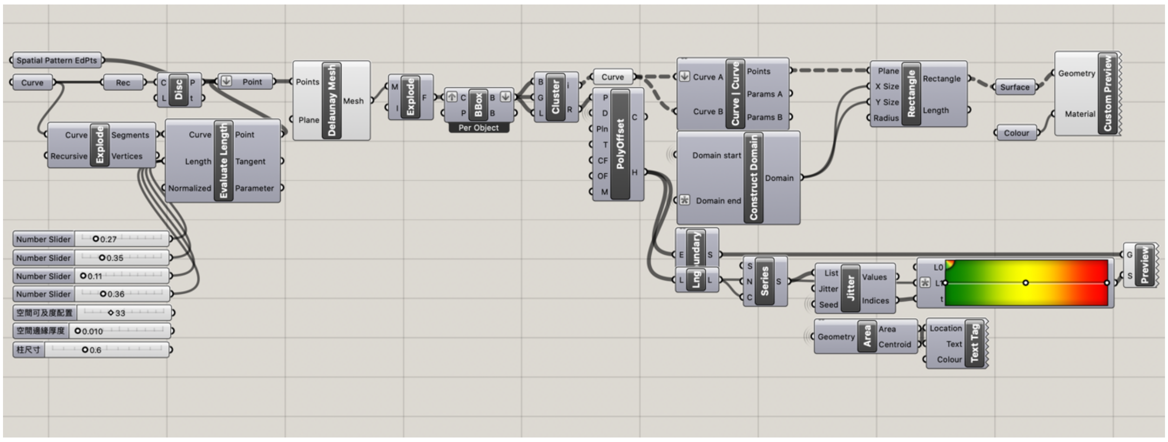


Figure 5: Switch Route Algorithm (SRA)

To implement the targets of a switch route, we need to embed two editable mechanisms associated with "Design Guidelines" in SRA, which include (1) colour configuration and (2) spatial composition; the idea of "Design Guidelines" is quite like “labeling principle” proposed by Weixin Huang (2018). In SRA, we apply the random mode to set up the coordinate of editable points; however, when we compare the difference of "Spatial Composition," respectively, in Figure 6 and Figure 7. The random mode would lead to an unreasonable spatial pattern, such as loose spatial organization and the unreasonable dimension of the room. To avoid the errors resulting from the random mode, we adopt manual control to ensure that we can keep the architect-like design quality and the architect's aesthetic value in the learning data for ML.

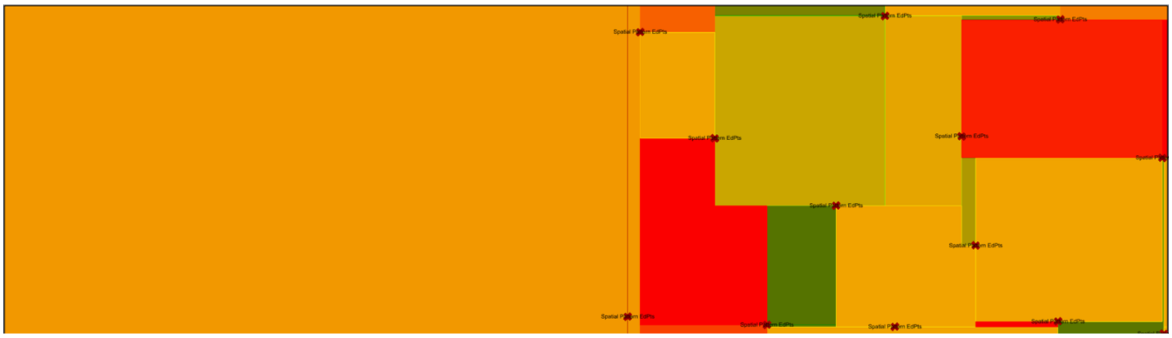


Figure 6: The Output with Random Mode in SRA

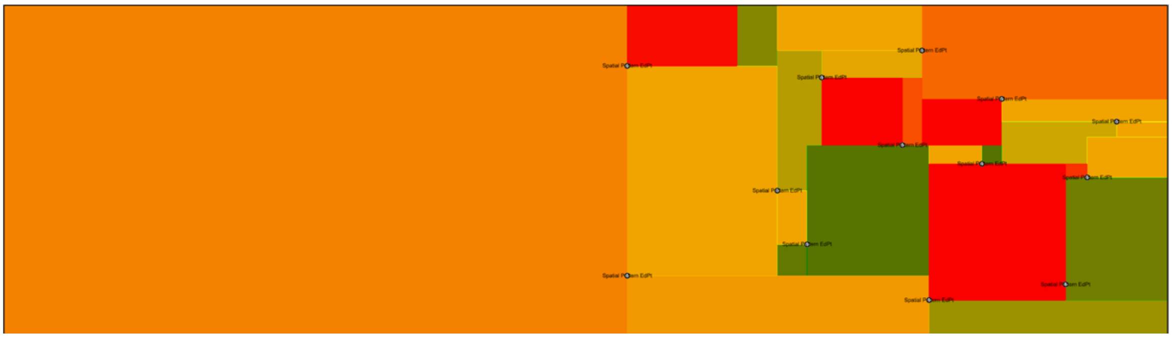


Figure 7: The Output with Manual Mode in SRA

3. The Method of ML

After deriving the learning data from SAA and SRA, we adopt "Style Transfer" as the learning method for ML. The main reasons are as follows:

- **Accurate image recognition:** Style Transfer uses the structure of a convolutional neural network (CNN). CNN is famous for applying loss functions to improve feature extraction capabilities. CNN can detect hidden units from the shallow to deeper layers of the image, meaning Style Transfer can see very subtle features of the picture to achieve a more accurate image recognition.

- **Enable the control in recognition quality:** The loss function of Style Transfer provides a flexible mechanism to control the image recognition quality on SAA and SRA's learning data.

The following paragraph will explain how we apply style transfer to correct the loss of detail and to obtain a better building plan generation.

3.1 The Method in Improving Detail Loss of Learning Data

At the beginning of the research, we used Neural Style Transfer with AdaIN (Xun Huang : 2017) to generate the building plan. Table 1 shows that the spatial pattern of the building plan (Generative Plan_B) plotted via AdaIN significantly differs from the prototype building plan labelled by SAA.

We believe that is the problem resulting from the image noise of the learning data that AdaIN cannot deal with. In the research on image noise reduction, Chang-yan (2008) confirmed that Wavelet Transformation based on the Fourier transform mechanism could effectively prevent the loss of image detail. Here, Wavelet Transformation is also a fundament of style transfer, which can deal with image repair and the details of image recognition. Furthermore, Leon (2016) proved that Style Transfer is a powerful GAN for Machine Learning in avoiding the loss of image details. According to the above study, we revise three Style Transfer GAN's components as our research GAN called Plan Style Transfer (PST) to enhance ML in generating a more accurate building plan, including (1) Modify the style transfer's structure for deeper feature extraction, (2) Define the loss function and optimizer to get a better solution for noise reduction, (3) Compile PlaidML to accelerate Machine Learning.

Regarding the style transfer's structure, it is made up of five blocks; each block must contain several "convolutional layers (Conv)," "pooling (Pool)," and "activation function (ReLU)." Compared with the learning targets of AdaIN, the generative adversarial network (GAN) used in this research needs to extract the feature of the building plan's colour, pattern, and space topology relationship. Therefore, we must amend the style transfer's structure corresponding to the research purposes. After trying different combinations of convolutional layer, pooling, and activation function, we found that the deeper structure of style transfer would cause the problem of gradient vanishing and degradation; in other words, the deeper GAN structure would decrease the accuracy of ML. To avoid this problem, we propose the structure of PST (see Table I) to make ML meet the research's targets.

Table 1: The Comparison of PST and AdaIN in Structure

Layers	**PST**			**AdaIN**		
	Conv	**Pool**	**ReLU**	**Conv**	**Pool**	**ReLU**
Block 1	2	1	2	2	1	2
Block 2	2	1	2	2	1	2
Block 3	4	1	4	4	1	4
Block 4	4	1	4	4	1	4
Block 5	4	1	4	N/A	N/A	N/A
Optimizer	ADAM+L-BFGS			ADAM		
Loss Function	Content, Style-Masked, Style			Content, Style		
SAA 	**Generative Plan_A**			**Generative Plan_B**		
SRA 						

Furthermore, unlike AdaIN, our research uses ADAM (Kingma: 2014) and L-BFGS; both are loss function optimizers for stochastic gradient descent (SGD) in ML. Although ADAM allows ML performs under low-level specification hardware, ADAM cannot provide a more accurate calculation operation of the loss function. We need L-BFGS to enable the machine to run at high-level hardware equipment to provide a more precise performance for the loss function calculation. This measure makes an attractive benefit to our style transfer structure.

Apart from the loss function optimizer, the function of mask_style_layer (Alex Wells: 2017) has a more significant influence on ML in our research. Typically, we need to add a loss layer to reduce the noise in improving the feature extraction ability. Compared to the learning targets of AdaIN, not only do we hope that ML can learn the feature of the colors and textures, even the complex topological features, including the building openings, the space shape, the spatial pattern, and the spatial scale. Table 1 shows the specific learning result between AdaIN and PST. The improvement of the learning result would be attributed to applying the function of mask_style_layer in extracting the features of learning data.

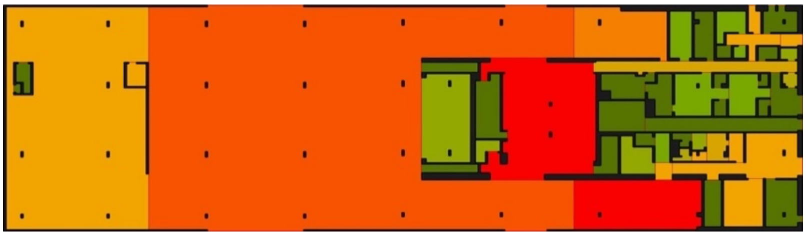
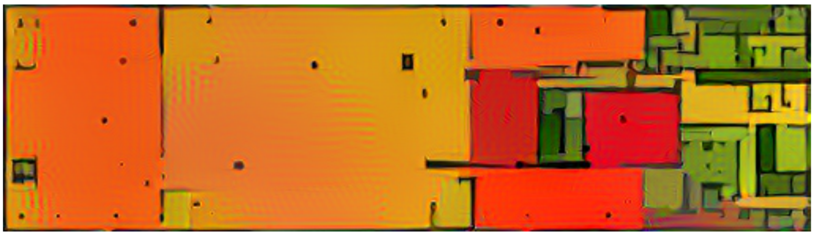
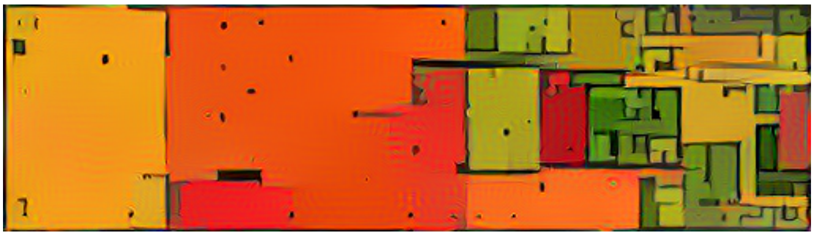
3.2 The Qualitative Evaluations

The image of "Generative Plan_A" in Table 1 shows that PST we adopt can effectively improve ML in generating the results of the building planes compared to AdaIN. Although ML can produce diversity-building plans similar to the prototype, the question remains about which one is optimal. Consequently, we need to develop an evaluative mechanism to help us to tell the optimal building plan from all solutions generated by ML. Next, we use ML to generate 40 building plans as the samples and calculate their quantitative values of MD, RRA, and Rn in Table 2. The evaluative method we adopt is "Confidence Interval (CL)." we apply the first Scenario parameters of GMCA as the benchmark (MD=1.75, RRA=0.08, Rn=15.26) to calculate the confidence interval of 40 samples; then, we start to input the confidence level from 95% to 10% to select the confident samples. A confidence level is the percentage of all possible samples expected to include the proper population parameter; if we are delighted with the investigated samples, we will use a high confidence level factor (usually 90% to 95%). However, the semi-supervised learning in this research doesn't rely on data training. Thus, we cannot use a high confidence level to determine the scope of the efficient samples. For this research, we suggest using the middle confidence level (40% to 70%) to predict the range of the accurate models. As the statistical data in Table 2, When we use the confidence level of 91%, 70%, and 10%, respectively, we can observe that the confidence level can help the research narrow down the more precise range to find the possible proposals generated by ML. In other words, we can use the method of a confidence level to estimate the likely recommendations with the standards given by the prototype building plan labeled by SAA. Besides, based on Bill Hillier's argument that quantitative values are used to interpret spatial topology relationships, MD is a benchmark value to identify the spatial configuration. RRA is the value of evaluating the spatial integration relationship in the building plan. If there have two building plans with the same topological structure, then these two building plans should have an approximate value of MD and RRA.

According to the above definitions and the confidence interval derived from the different confidence levels, we highlight three quantitative values of possible building plans on the confidence level (70%) column and the corresponding building plans in Table 2.

Table 2: The Portion SpaceSyntax Quantitative Values of 40 Building Plans Generated by PST

MD	RRA	Rn	MD	RRA	Rn	MD	RRA	Rn
1.727	0.08	11.77	1.727	0.08	11.77	1.727	0.08	11.77
1.794	0.09	10.78	1.794	0.09	10.78	1.794	0.09	10.78
1.805	0.09	10.63	1.805	0.09	10.63	1.805	0.09	10.63
1.75	0.08	11.41	1.75	0.08	11.41	1.75	0.08	11.41
1.763	0.09	11.21	1.763	0.09	11.21	1.763	0.09	11.21
1.812	0.09	10.54	1.812	0.09	10.54	1.812	0.09	10.54
1.936	0.1	9.14	1.936	0.1	9.14	1.936	0.1	9.14
1.99	0.1	8.6	1.99	0.1	8.6	1.99	0.1	8.6
1.721	0.08	11.87	1.721	0.08	11.87	1.721	0.08	11.87
Confidence Level (91%)			**Confidence Level (70%)**			**Confidence Level (10%)**		
MD	**RRA**	**Rn**	**MD**	**RRA**	**Rn**	**MD**	**RRA**	**Rn**
1.67 ~ 1.84	0.07 ~ 0.09	12.24 ~ 18.29	1.70 ~ 1.81	0.07 ~ 0.08	13.44 ~ 17.09	1.75 ~ 1.76	0.08 ~ 0.08	15.04 ~ 15.48

Garage Museum of Contemporary Art (GMCA): Prototype Building Plan			
MD	**RRA**	**RN**	
1.75	0.08	15.26	
The Selected Generative Building Plan Based on GMCA Standards			
MD	**RRA**	**RN**	
1.75	0.08	11.41	
1.727	0.08	11.77	
1.721	0.08	11.87	

4. Conclusion and Future work

The generator loss and discriminator loss play a fundamental role in training a generative adversarial network (GAN) by guiding the gradient descent process. Achieving an appropriate balance between the generator and discriminator losses is crucial for effective GAN training and improving learning results (Goodfellow: 2014). To avoid the abovementioned problem, it is necessary to collect the amount of training data and increase the pass times of the training data. Then, it increases the workforce to label the object in the training data. Consequently, the traditional GAN has a considerable obstacle in the research field that lacks examples for the training data, such as architecture design. Cause of the unique and personal properties of the architectural design, it is harder for the researcher to collect enough training examples for the specific architectural use. Even though if we can provide the machine with enough examples of building use for training, the machine only can produce the plan for this specific use; if ML wants to propose different building types, we must re-train the machine again, not to mention the same building type will have other spatial configuration habits in different countries.

To reduce the dependence of ML on training data and the workforce required for labeling data, we adopt SAA to divide the data annotations into concrete labels and abstractive ones in the learning materials, respectively. The concrete labels include the rooms and the structural elements, and the abstractive labels relate to the relative spatial relationships and spatial access frequency.

After the labeling and classification, ML can use three quantitative values (Rn, RRA, MD) as the abstractive labels to define the decision boundary and to separate the objects of concrete labels into distinctive groups. In this step, abstractive labels play an essential role in adjusting the distribution of the overall concrete labels on the decision boundary; that is to say, our research uses the spatial relationship to rearrange the configuration between rooms and structural elements on the decision boundary to achieve the targets of how ML proposes the diverse proposals via PST. Through several research experiments, we discovered that ML could produce various multi-alternative recommendations and evaluate the hierarchy of spatial relationships under the practice of our research framework.

However, as we continued to test the semi-supervised machine learning through the different types of building planes, we found that semi-supervised machine learning cannot generate the proper building plans from simple ones. We believe that the size limitation of the spatial feature map might be the main factor to trigger the problem. ML is always concerned with avoiding capture noises on feature maps, which could cause overfitting results. Thus, the engineers always use the method of down-sampling on the feature map to prevent ML from capturing the noises on the feature map. However, it also results in a state where there are not many clues on the feature map, which will cause the machine to be unable to read the features and cause ML to collapse. To improve this problem, Latent Diffusion Model (LDM), which makes DALLE 2, Stable Diffusion, and Midjourney produce highly detailed works, can provide other thinking for ML to capture the more detailed feather without less down-sampling. Thus, we will continue to explore the potential of how to apply the Latent Diffusion Model (LDM) in the research framework.

References

1. Akkelies van Nes, Claudia Yamu, “Introduction to Space Syntax in Urban Studies,” Springer, Switzerland, 2021, ISBN 978-3-030-59139-7
2. A. Z. Peixinho, B. C. Benato, L. G. Nonato and A. X. Falcão, "Delaunay Triangulation Data Augmentation Guided by Visual Analytics for Deep Learning," 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2018, pp. 384-391, doi: 10.1109/SIBGRAPI.2018.00056.
3. Bosomaier, T., & Green, D., “Patterns in the sand: Computers, complexity and life,” Australia: Allen & Unwin Pty Ltd, 1998
4. C. S. Hsiao, "Method Enhancing Machine Learning in Urban Topology Analysis,"2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE), 2021, pp. 288-293, doi: 10.1109/ECICE52819.2021.9645669.
5. Carta, S. “Self-Organizing Floor Plans,” Harvard Data Science Review, 2021, 3(3)
6. Chang-yan, Chi, Ji-xian, Zhang, Zheng-jun, Liu. “Study on Method of Noise Reduction in a Stripped Image,” The International Achieves of the Photogrammetry. Vol. XXXVII. Part B6b. 213-216, 2008
7. Hillier, B. “Space Is the Machine: A Configurational Theory of Architecture,” Cambridge University Press, Cambridge, 1996.
8. Huang, Weixin & Zheng, Hao. “Architectural Drawings Recognition and Generation through Machine Learning,” 10.52842/conf.acadia.2018.156.
9. Ian J. Goodfellow et al., "Generative Adversarial Networks," arXiv, 2014
10. Jaime de M., et al., “Deep Form Finding,” eCAADe 37, 71-80
11. Justin Johnson, et al., “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” arXiv:1603.08155v1 [cs.CV], 27 Mar 2016
12. Kingma, Diederik P. “Adam: A Method for Stochastic Optimization,” *CoRR* abs/1412.6980 (2014): n. pag.
13. Leon A. Gatys, Alexander S. Ecker, “Image Style Transfer Using Convolutional Neural Networks,” CVPR 2016, 2414-2423
14. Morteza Rahbar, “Architectural layout design through deep learning and agent-based modeling: A hybrid approach,” Journal of Building Engineering, Volume 47, 2022, 103822, ISSN 2352-7102
15. Pelin D. “Space Syntax in Architectural Design,” 6th International Space Syntax Symposium, 2007, 56:1-56:12
16. Yazar, Tuğrul & Dursun, Pelin & Kozikoglu, Nilufer & Üneşi, Oğulcan & Erden, Melike. “Space Chase: A Dynamic and Interactive Tool for Architectural Design Process,” 13th Space Syntax Symposium, 2022.
17. Richard, S. Michael, V. “Space Syntax for Generative Design,” 9th Space Syntax Symposium 2013. 50:1-50:12
18. Segu, M., Grinvald, M. “3DSNet: Unsupervised Shape-to-Shape 3D Style Transfer,” arXiv, 2020
19. Stanislas Chaillou, “Space Layouts and GANs,” Towards Data Science, 2020
20. Stiny, George & Gips, James., “Shape Grammars and the Generative Specification of Painting and Sculpture,”. IFIP Congress 1971. 71. 1460-1465.
21. Van der Laag Yamu, Claudia & Nes, Akkelies & Garau, Chiara. (2021). “Bill Hillier's Legacy: Space Syntax-A Synopsis of Basic Concepts, Measures, and Empirical Application,” Sustainability. 13. 1-25. 10.3390/su13063394.
22. Wells, Alex and Stanford. “Localized Style Transfer.” (2017).