# ArchiGAN: Artificial Intelligence x Architecture

**Stanislas Chaillou**

AI will soon massively empower architects in their day-to-day practice. This article provides a proof of concept. The framework used here offers a springboard for discussion, inviting architects to start engaging with AI, and data scientists to consider Architecture as a field of investigation. In this article, we summarize a part of our thesis, submitted at Harvard in May 2019, where Generative Adversarial Neural Networks (or *GANs*) get leveraged to design floor plans and entire buildings (Fig. 1).

We believe that a statistical approach to design conception will shape AI's potential for Architecture. This approach is less deterministic and more holistic in character. Rather than using machines to optimize a set of variables, relying on them to extract significant qualities and mimicking them all along the design process represents a paradigm shift.

We can unpack floor plan design into three distinct steps:

(I) **building footprint massing**
(II) **program repartition**
(III) **furniture layout**

Each step corresponds to a Pix2Pix GAN-model trained to perform one of the three tasks above. By nesting these models one after the other, we create an entire apartment building "*generation stack*" while allowing for user input at each step. Additionally, by tackling multi-apartment processing, this project scales beyond the simplicity of single-family houses.

Beyond the mere development of a generation pipeline, this attempt aims at demonstrating the potential of GANs for any design process, whereby nesting GAN models, and allowing user input between them, we try to achieve a back and forth between humans and machines, between disciplinarian intuition and technical innovation.

S. Chaillou (✉)
Spacemaker, Oslo, Norway
e-mail: stan@spacemaker.ai



P. F. Yuan et al. (eds.), *Architectural Intelligence*,
https://doi.org/10.1007/978-981-15-6568-7_8

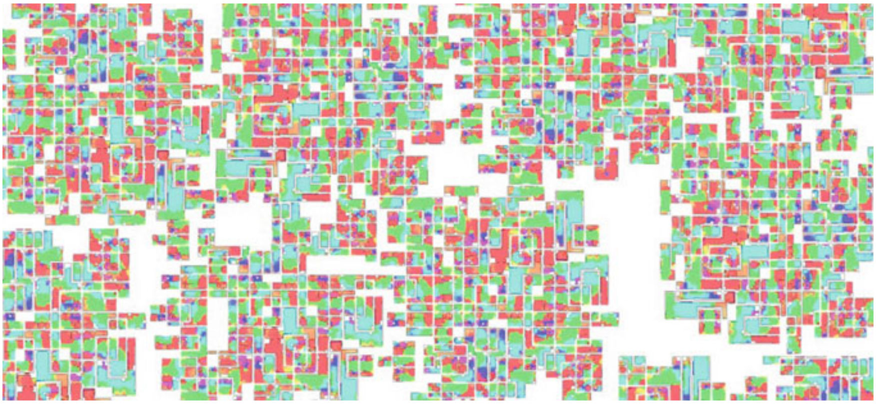

**Fig. 1** GAN-Generated masterplan. *Source* Author

## 1 Representation, Learning, and Framework

Pix2Pix uses a conditional Generative Adversarial Network (*cGAN*) to learn a mapping from an input image to an output image. The network consists of two main pieces, the Generator and the Discriminator. The Generator transforms the input image to an output image; the Discriminator tries to guess if the image was produced by the generator or if it is the original image. The two parts of the network challenge each other resulting in higher quality outputs which are difficult to differentiate from the original images.

We use this ability to learn image mappings which lets our models learn topological features and space organization directly from floor plan images. We control the type of information that the model learns by formatting images. As an example, just showing our model the shape of a parcel and its associated building footprint yields a model able to create typical building footprints given a parcel's shape.

We used Christopher Hesse's implementation of Pix2Pix. Figure 2 displays the results of a typical training. We show there how one of my GAN-models progressively learns how to layout rooms and the position of doors and windows in space–also called fenestration–for a given apartment unit in the sequence in Fig. 2. Although the initial attempts proved imprecise, the machine builds some form of intuition after 250 iterations.

## 2 Precedents

The early work of Isola et al. in November 2018 enabling image-to-image translation with their model Pix2Pix has paved the way for our research.

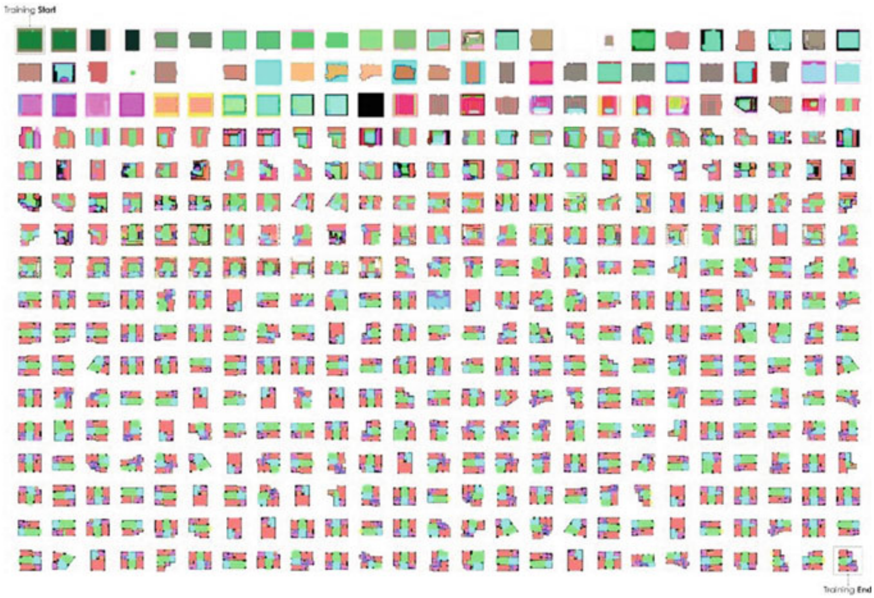


**Fig. 2** Apartment architectural sequence. *Source* Author

Zheng and Huang in 2018 first studied floor plan analysis using GAN. The authors proposed to use GANs for floor plan recognition and generation using Pix2PixHD. [1] Floor plan images processed by their GAN architecture get translated into programmatic patches of colors. Inversely, patches of colors in their work turn into drawn rooms. If the user specifies the position of openings and rooms, the network elements laid out become furniture.

Nathan Peters' thesis [2] at the Harvard Graduate School of Design in the same year tackled the possibility of laying out rooms across a single-family home footprint. Peters' work turns an empty footprint into programmatic patches of color without specified fenestration.

Regarding GANs as design assistants, Nono Martinez' thesis [3] at the Harvard GSD in 2017 investigated the idea of a loop between the machine and the designer to refine the very notion of "*design process*".

## 3 Stack and Models

We build upon the previously described precedents to create a three-step generation stack. As described in Fig. 3, each model of the stack handles a specific task of the workflow: **(I) footprint massing, (II) program repartition, (III) furniture layout**.

An architect is able to modify or fine-tune the model's output between each step, thereby achieving the expected machine-human interaction.

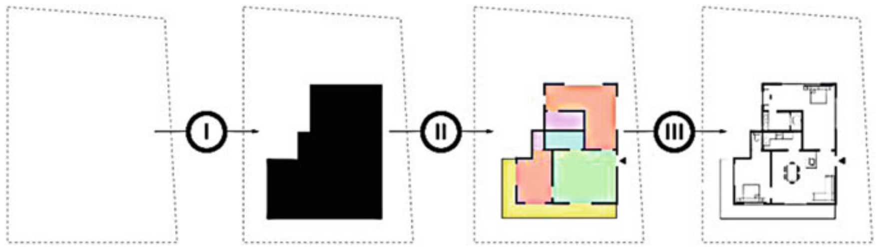


Fig. 3 Generation stack in three models. *Source* Author

## 3.1 Model I: Footprint

Building footprints significantly define the internal organization of floor plans. Their shape is heavily conditioned by their surroundings and, more specifically, the shape of their parcel. Since the design of a housing building footprint can be inferred from the shape of the piece of land it stands on, we have trained a model to generate typical footprints, using GIS-data (Geographic Information System) from the city of Boston. We feed pairs of images to the network during training in a format suitable for Pix2Pix, displaying the raw parcel (left image) and the same parcel with a given building drawn over it (right image). We show some typical results in Fig. 4.

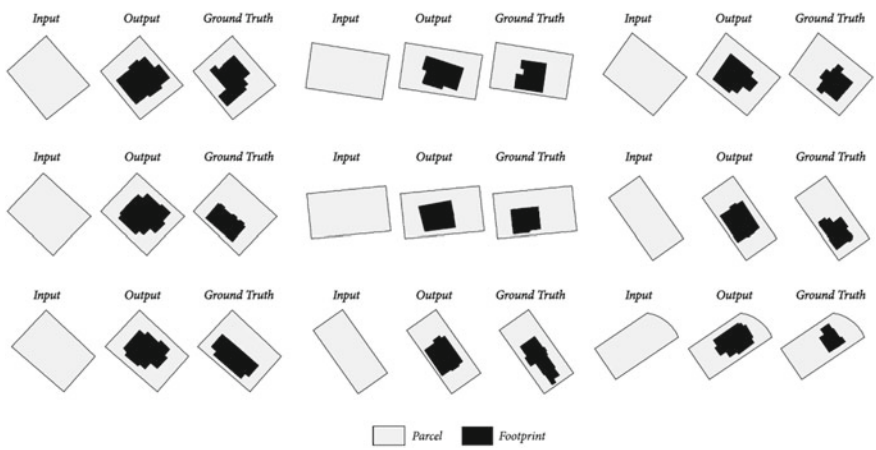


Fig. 4 Results of Model I. *Source* Author

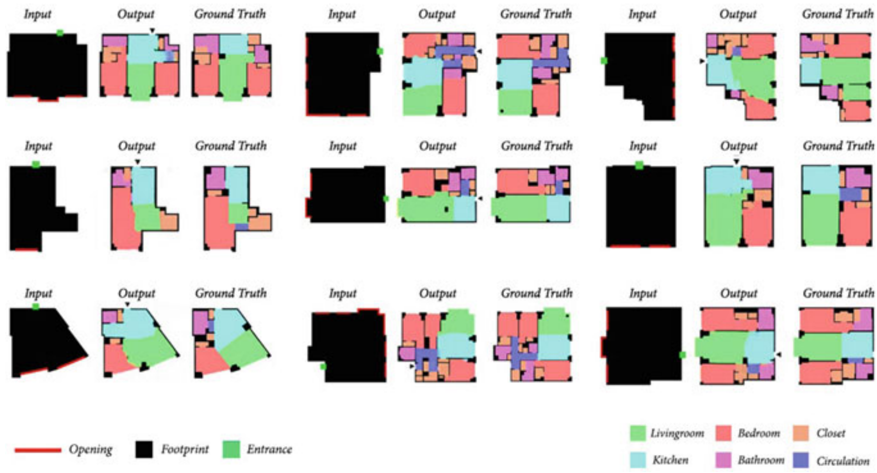


**Fig. 5** Results of model II. *Source* Author

### 3.2 Model II: Program

Model II handles repartition and fenestration. The network takes as input the footprint of a given housing unit produced by Model I, the position of its entrance door (green square), and the position of the main windows specified by the user (red lines). The plans used to train the network to derive from a database of 800 + plans of apartments, properly annotated and given in pairs to the model during training. In the output, the program encodes rooms using colors while representing the wall structure and its fenestration using a black patch. Some typical results are displayed in Fig. 5.

### 3.3 Model III: Furnishing

Finally, Model III tackles the challenge of furniture layout using the output of model II. This model trains on pairs of images, mapping room programs in color to adequate furniture layouts. The program retains wall structure and fenestration during image translation while filling the rooms with relevant furniture, specified by each room's program. Figure 6 displays some typical results.

## 4 UI and Experience

We provide the user with a simple interface for each step throughout our pipeline. On the left, users can input a set of constraints and boundaries to generate the resulting

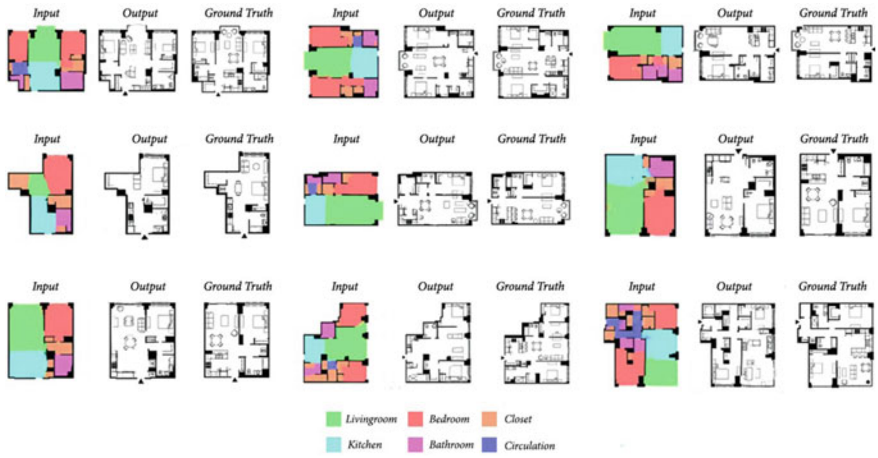


**Fig. 6** Results of model III. *Source* Author

plan on the right. The designer can then iteratively modify the input on the left to refine the result on the right. Figure 7 showcases this type of interface & process, set up for Model II.

## 5 Model Chaining and Apartment Building Generation

We scale the utilization of GANs in this part to the entire apartment building design. The project uses an algorithm to chain Models I, II, and III, one after the other, processing multiple units as single images at each step. Figure 8 shows this pipeline.

The challenge of drawing floor plates hosting multiple units marks the difference between single-family houses and apartment buildings. Strategically, the ability to control the position of windows and units' entrances is key to enable unit placement while ensuring each apartment's quality. Since Model II takes doors and windows position as input, the generation stack described above can scale to entire floor plates generation.

The user is invited to specify the unit split between Model I and Model II, in other words, specifying how each floor plate divides into apartments and to position each unit entrance door and windows, as well as potential vertical circulations (staircases, cores, etc.). The proposed algorithm then feeds each resulting unit to Model II (results shown in Fig. 9), and then III (result in Fig. 10), to finally reassemble each floor plate of the initial building. The algorithm finally outputs as individual images, all floor plates of the generated building.

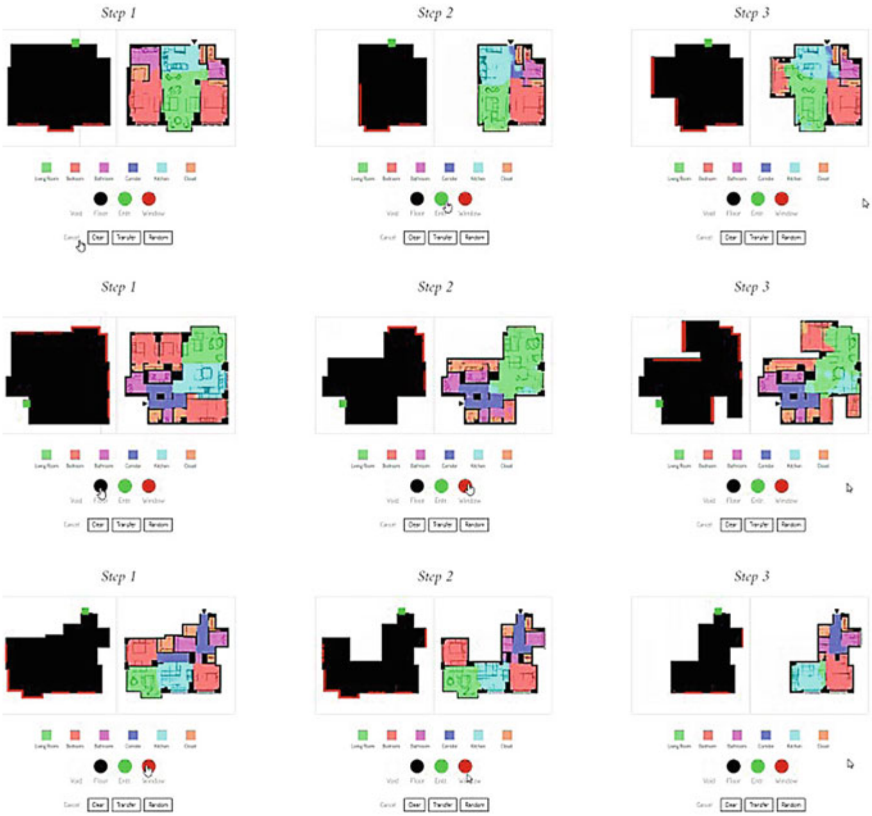


**Fig. 7** Design iterations using model II interface. *Source* Author

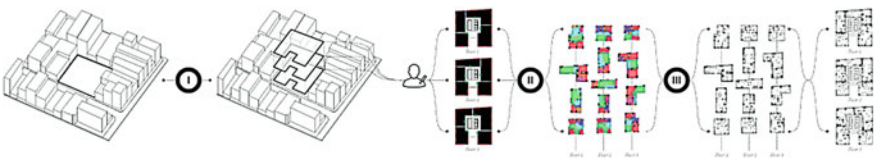

**Fig. 8** Apartment building generation pipeline. *Source* Author

## 6 Going Further

If generating standard apartments can be achieved using this technique, pushing the boundaries of our models is the natural next step. GANs offer remarkable flexibility to solve seemingly highly constrained problems. In the case of floor plan layout, partitioning and furnishing the space by hand can be a challenging process as the


**Fig. 9** Model II output, program for each floor plate. *Source* Author


**Fig. 10** Model III output, furnishing of each individual unit. *Source* Author

footprint changes in dimension and shape. Our models prove to be quite "smart" in their ability to adapt to changing constraints, as evidenced in Fig. 11.

The ability to control the units' entrance door and windows position, coupled with the flexibility of my models, allows us to tackle space planning at a larger scale, beyond the logic of a single unit. In Fig. 12, we scale our pipeline to entire buildings generation while investigating our model's reaction to odd apartment shapes and contextual constraints.

## 7 Limitations and Future Improvements

If the above results lay down the premise of GANs' potential for Architecture, some clear limitations will drive further investigations in the future.

**First, as apartment units stack up in a multi-story building, we cannot guarantee for now the continuity of load-bearing walls from one floor to the next.** Since all the internal structure is laid out differently for each unit, load-bearing walls

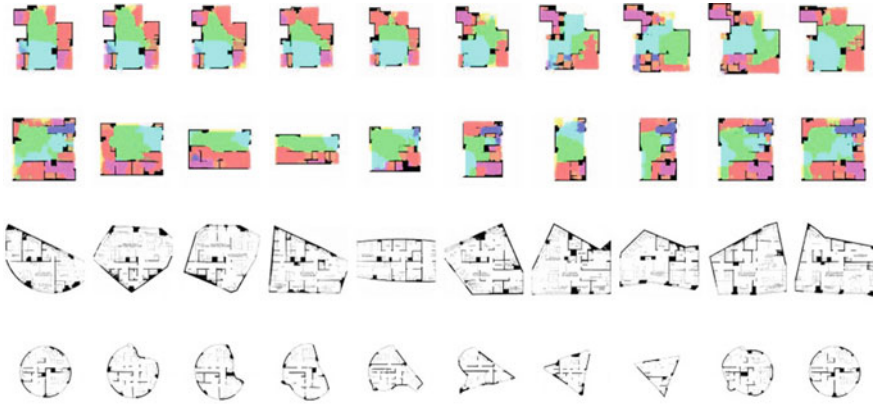

**Fig. 11** Flexible building layouts. *Source* Author

might not be aligned. For now, we consider the façades to be load bearing. However, the ability to specify load-bearing elements' position in the input of Model II could potentially help address this issue.

**Additionally, increasing the size of the output layer by obtaining larger images which offer better definition is a natural next step.** We want to deploy the Pix2Pix HD project developed by NVIDIA in August 2018 to achieve this.

Finally, a major challenge comes from the data format of our outputs. GANs like Pix2Pix handle only pixel information. The resulting images produced in our pipeline cannot, for now, be used directly by architects & designers. **Transforming this output from a raster image to a vector format is a crucial step for allowing the above pipeline to integrate with common tools & practices.**

## 8 Conclusion

AI will soon massively empower architects in their day-to-day practice. As such potential is about to be demonstrated, our work participates to the proof of concept while our framework offers a springboard for discussion, inviting architects to start engaging with AI, and data scientists to consider Architecture as a field of investigation. However, today, our manifesto could be summarized in four major points.

Conceptually first, our belief is that a statistical approach to design conception shapes AI's potential for Architecture. Its less-deterministic and more-holistic character is undoubtedly a chance for our field. Rather than using machines to optimize a set of variables, relying on them to extract significant qualities and mimicking them all along the design process is a paradigm shift.

Second, we are directionally convinced that our ability to design the right pipeline will condition AI's success as a new architectural toolset. Our choice for the "*Grey*

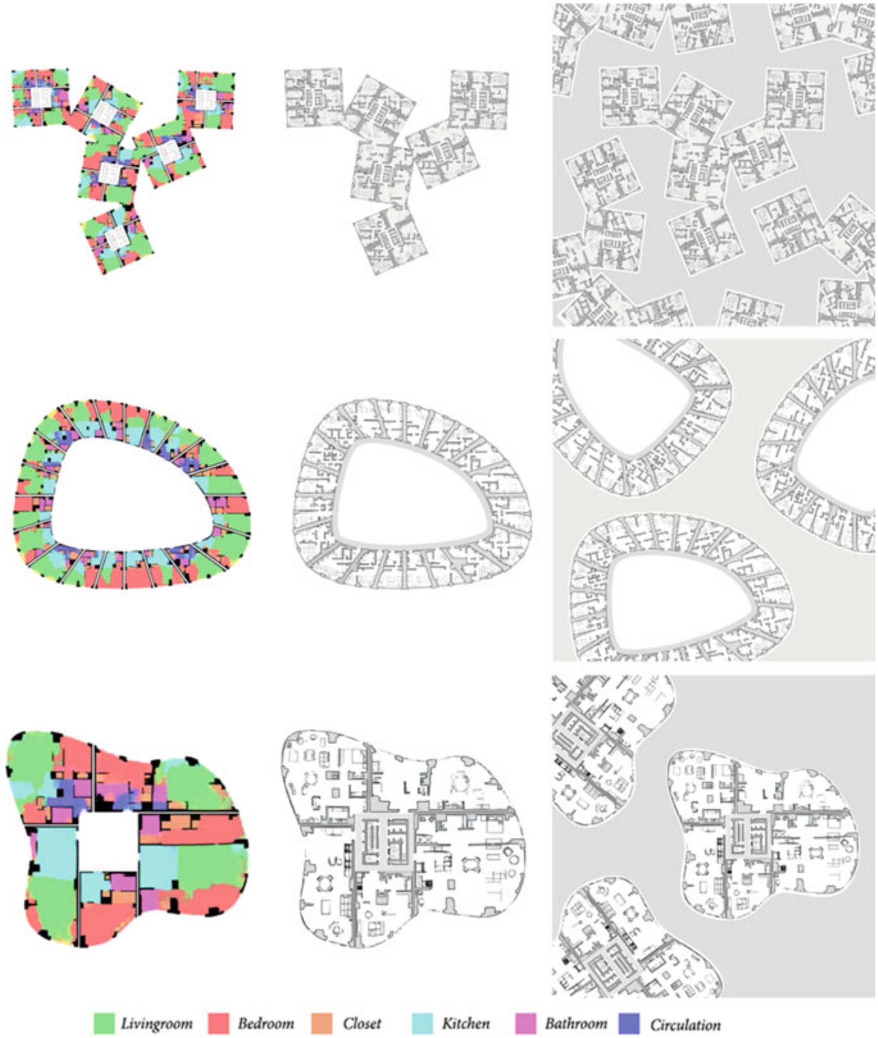


**Fig. 12** GAN-enabled building layouts. *Source* Author

*Boxing*" approach, as introduced by Prof. Andrew Witt in Log, will likely secure the best potential results. This method contrasts with the "*black box*" approach, that only allows users to input information upfront, and to get finished design options at the end of the process, without any control over the successive generation steps. To the contrary, by breaking out our pipeline into discrete steps, "*Grey Boxing*" permits the user to intervene all along the way. His tight control over the machine is his ultimate guarantee of the design process quality.

Third, we technically believe that the sequential nature of the application will facilitate its manageability and foster its development. The ability to intervene throughout the generating process is a fundamental dimension: as each step of the pipeline represents a distinct portion of architectural expertise, each model can be trained independently, opening the way to significant improvements and experimentation in the near future. Indeed, improving this entire pipeline end-to-end could be a long and cumbersome task, while amending it step by step remains a manageable process, within the reach of most architects and engineers in the industry.

Finally, we hope our framework will help address the endless breadth and complexity of the models to be trained and those used in any generation pipeline. Tackling *parcels-footprint-room split*-etc., as we do is one possible approach among, we believe, a large set of options. To encapsulate the necessary steps of space planning, the key is more the principle than the method. And with the growing availability of architectural data, we encourage further work and open-minded experimentation.

Far from thinking about AI as the new dogma in Architecture, we conceive this field as a new challenge, full of potential, and promises. We see here the possibility for rich results, that will complement our practice and address some blind spots of our discipline.

## References

1. Zheng, H., & Huang, W. (2018). *Architectural drawings recognition and generation through machine learning*. Cambridge: MA, ACADIA.
2. Peters, N. (2017). *Master thesis: "Enabling alternative architectures: Collaborative frameworks for participatory design."* Cambridge, MA: Harvard Graduate School of Design.
3. Martinez, N. (2016). *Suggestive drawing among human and artificial intelligences*. Cambridge, MA: Harvard Graduate School of Design.