**Introduction:**

House GAN++ is a machine learning framework that builds on House Gan, extends and refines it.  
House design is a time-consuming process, mostly limited to professional architects due to cost.   
The goal of House GAN++ is to generate professional floorplans automatically, based on the client’s conditions, which will save a lot of time and money.

In this work, the architecture will be based on Conv-MPN feature pooling that is reformulated to allow feature-exchange between nodes and edges. Where each node is a room with its corresponding room type, and an edge represents a door type.

The type of training done is called Component-wise Ground Truth (GT)-conditional Training (GTCT), where instead of utilizing a full GT – component-wise-conditioning (where the entire GT is provided in every iteration), only 50% of the ground truth is transmitted during each iteration.

There’s also a refinement scheme of the input constraints sent to the GAN, which will be explained later in detail.

The following articles were used:

House-Gan++: Generative Adversarial Layout Refinement Networks

House-GAN: Relational Generative Adversarial Networks for Graph-constrained House Layout Generation

Conv-MPN: Convolutional Message Passing Neural Network for Structured Outdoor Architecture Reconstruction

**Theories and Mathematical Background:**

**Graph**:

A graph is a data structure that represents a collection of elements (referred to as nodes), where a connection between 2 nodes is called an edge. There are different shapes of graphs, each used in different scenarios, according to the needs of the system.  
Each graph can also be mapped to an adjacency matrix.

For example, in House GAN++, a node represents a room, and an edge can represent a door type that connects that room to another room.

**Neural Network:**

Neural networks are a subset of machine learning and are at the heart of deep learning algorithms. Both the name and structure are inspired by the human brain, and they mimic the way biological neurons signal to each other.

Artificial Neural Networks (ANNs) consist of node layers, which contain an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has a weight and a threshold value. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed to the next layer of the network.

This allows House GAN to study from different house layouts, the differences between room types and properties.

**Convolutional Neural Network(CNN):**

DON’T FORGET

**Generative Adversarial Network (GAN):**

A GAN consists of a generator and a discriminator, where each one of the mentioned is a neural network.

The generator and the discriminator have 2 different roles:

The generator’s role is to generate samples that are indistinguishable from real data, using random noise vectors as input. A generator is typically composed of several layers (such as convolutional layers, etc)

The discriminator’s role is to output a probability score, indicating the likelihood of the input being real.

The generator is trained using real data, where as the discriminator is trained using both real and fake data.

The generator tries to deceive the discriminator, meaning that the generator passes on data, some of which is generated by the generator, while the rest is real data, to the discriminator, and the discriminator tries to differentiate between real and fake data.

**Convolutional Message Passing Network (Conv-MPN):**

MPN is a type of neural network designed to propagate information through nodes and edges of a graph.  
This allows to exchange information between nodes to capture relational dependencies and interactions between them.

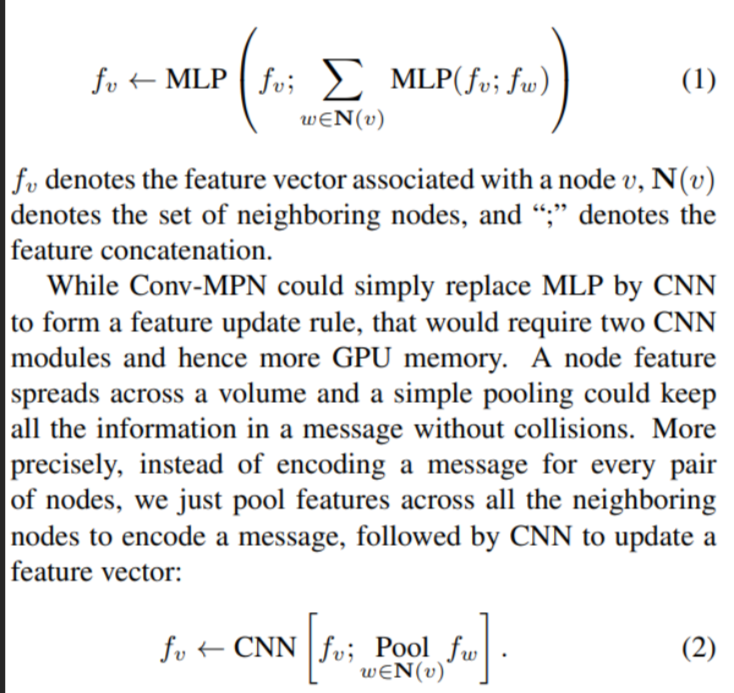
Each node receives information from its neighbors, then updates its own state by incorporation the features of the neighbors.

Conv-MPN is a variant of a graph neural network (GNN), it utilizes convolutional neural networks. It learns to infer relationships between nodes by exchanging messages. It’s specifically designed for cases where a node has a specific spatial embedding.

Two key distinctions between MPN and Conv-MPN

1)The feature of a node is represented as a 3d volume as in CNNs instead of 1-d vector.

2) Convulsions encode messages instead of fully connected layers

****

**feature pooling:**