Notes of “SceneGraphNet: Neural Message Passing for 3D Indoor Scene Augmentation”

Jihang Li

January 16, 2020

Contents

Abstract

Proposing a neural message passing approach to augment an input 3D indoor scene with new objects matching their surroundings. Distribution is predicted through passing learned messages

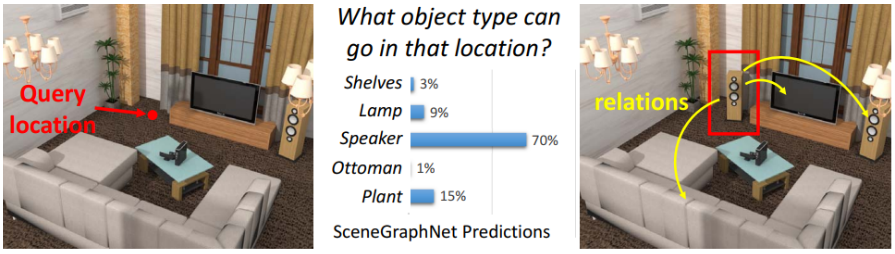

Figure 1: SceneGraphNet captures relationships between objects in an input 3D scene through iterative message passing in a dense graph to make object type predictions at query locations.

in a dense graph whose **nodes** represent **objects** in the input scene and **edges** represent spatial and structural **relationships**.

1 Introduction

The predicted distribution can be used in:

1. Enhancing 3D object recognition in scenes by taking into account the scene context (Figure 2).
2. Automatically populating 3D scenes with more objects by evaluating the probability distribution at different locations in the scene (Figure 3).
3. Providing object type recommendations to designers while modeling a 3D scene.

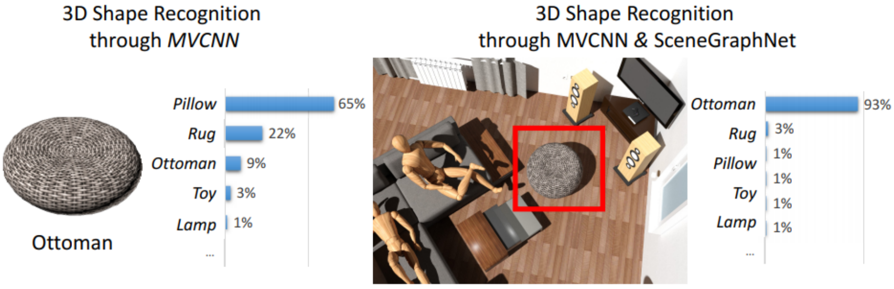

Figure 2: Context-based object recognition. Left: Object recognition using a multi-view CNN without considering the scene context. Right: Improved recognition by fusing the multi-view CNN and SceneGraphNet predictions based on scene context.

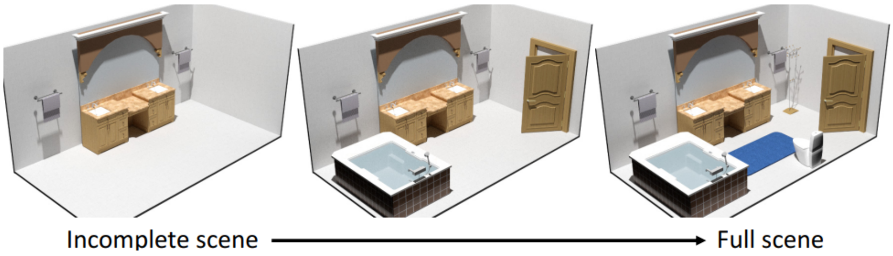

Figure 3: Iterative scene synthesis. Given an incomplete scene, our method is used to populate it progressively with more objects at their most likely locations predicted from SceneGraphNet.

The method models the scene as a graph, where edges represent:

- supporting
- surrounding
- adjacency

- co-occurrence
- long-range dependencies, *e.g.*, the choice of a sofa in one side of a room can influence the selection of other sofas, chairs, or tables in the opposite side of the room to maintain a plausible object set and arrangement.

It is found that predictions are more plausible when not only local or strictly hierarchical object relationships, but also long-range relationships are modeled.

An attention mechanism is designed to weight different messages for predicting objects at query locations.

Contributions:

- a new graph neural network architecture to model short- and long-range relationships between objects in 3D indoor scene.
- an iterative message passing scheme, reinforced with an attention mechanism, to perform object-related prediction tasks in scenes, including spatial query answering, context-based object recognition, and iterative scene synthesis.

2 Method

In a scene s, given a query location p, the output is a probability distribution over different object types/categories, $P(C|p, s)$ expressing how likely is for objects from each of these categories to fit well in this location and match the scene context.

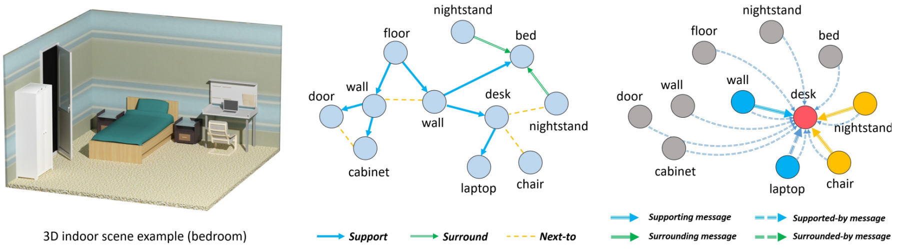

Figure 4: An example of the graph structure used for neural message passing in SceneGraphNet for a bedroom scene. Left: an input 3D scene. Middle: graph structure and object relationships we modeled (some relationships, *e.g.* dense “co-occurrence” and “next-to” ones, are skipped for clarity). Right: messages received by the object “desk” from other nodes in the graph for all different types of relationships.

2.1 Message Passing

Each node carries a vectorial representation h_i encoding information about the shape and its scene context based on the message it receives from other nodes. New messages are emitted from nodes; the message passing runs iteratively.

The passing procedure:

Initialization: based on the shape representation x_i at the node:

$$h_i^{(0)} = f_{init}(x_i; w_{init}) \tag{1}$$

where f_{init} is a two-layer MLP with learnable parameters w_{init} outputting a 100-dimensional node representation. $x_i = [c_i, p_i, d_i]$ where c_i is a one-hot vector representing category, $p_i \in \mathcal{R}^{\ni}$ representing position, $d_i \in \mathcal{R}^{\ni}$ representing oriented bounding box lengths.

Messages: a message from node k to i carries information based on $h_k^{(t)}$ and $h_i^{(t)}$ and the relationship r between them.

$$m_{k \to i}^{(r,t)} = f_{msg}^{r}(h_k^{(t)}, h_i^{(t)}; w_{msg}^{(r)}) \tag{2}$$

where f_{msg}^{r} is a two-layer MLP with learnable parameters $w_{msg}^{(r)}$ (weights are different per relationship r) outputting a 100-dimensional representation

Weights on messages: based on attention mechanism:

$$a_{k,i} = f_{att}(x_k, x_i; w_{att}) \tag{3}$$

where f_{att} is a two-layer MLP followed by a sigmoid layer with learnable parameters w_{att}.

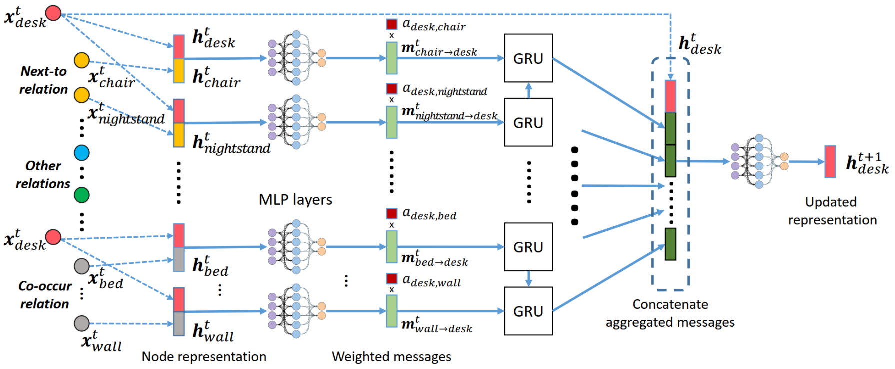

Figure 5: Overview of our message passing and underlying neural network architecture. We take the example in Figure 4 to illustrate a single message passing iteration.

3 Experiments

(Omitted.)

4 Discussion