Spatial Commonsense Graph for Object Localisation in Partial Scenes

Francesco Giuliari[1,2] Geri Skenderi[3] Marco Cristani[1,3] Yiming Wang[1,4] Alessio Del Bue[1]
[1]Istituto Italiano di Tecnologia (IIT) [2]University of Genoa [3]University of Verona
[4]Fondazione Bruno Kessler (FBK)

Abstract

We solve object localisation in partial scenes, a new problem of estimating the unknown position of an object (e.g. where is the bag?) given a partial 3D scan of a scene. The proposed solution is based on a novel scene graph model, the Spatial Commonsense Graph (SCG), where objects are the nodes and edges define pairwise distances between them, enriched by concept nodes and relationships from a commonsense knowledge base. This allows SCG to better generalise its spatial inference to unknown 3D scenes. The SCG is used to estimate the unknown position of the target object in two steps: first, we feed the SCG into a novel Proximity Prediction Network, a graph neural network that uses attention to perform distance prediction between the node representing the target object and the nodes representing the observed objects in the SCG; second, we propose a Localisation Module based on circular intersection to estimate the object position using all the predicted pairwise distances in order to be independent of any reference system. We create a new dataset of partially reconstructed scenes to benchmark our method and baselines for object localisation in partial scenes, where our proposed method achieves the best localisation performance. Code and Dataset are available here: https://github.com/IIT-PAVIS/SpatialCommonsenseGraph

1. Introduction

The localisation of unobserved objects given a partial observation of a scene is a fundamental task that humans solve often in their everyday life as shown in Fig. 1. Such a task is useful for many automation applications, including domotics for assisting visually impaired humans to find everyday items [10], visual search for embodied agents [3], and layout proposal for interior design [23]. Yet, object localisation in partial scenes has never been formally studied in the literature. We formalise the problem as the inference of the position of an arbitrary object in an unknown area of a scene based only on a partial observation of the scene.

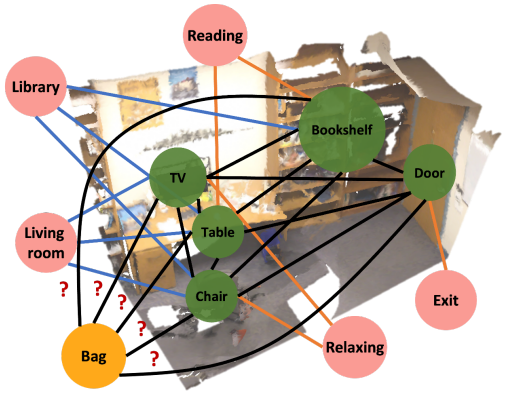


Figure 1: Given a set of objects (indicated in the **green** circles) in a partially known scene, we aim at estimating the position of a target object (indicated in the **orange** circle). We treat this localisation problem as an edge prediction problem by constructing a novel scene graph representation, the Spatial Commonsense Graph (SCG), that contains both the spatial knowledge extracted from the reconstructed scene, i.e. the proximity (**black** edges) and the commonsense knowledge represented by a set of relevant concepts (indicated in the pink circles) connected by relationships, e.g. *UsedFor* (**orange** edges) and *AtLocation* (**blue** edges).

Humans perform this object localisation task not only by using the partially observed environment, but also by relying on the *commonsense* knowledge that is acquired during our lifetime experience. For example, by knowing that pillows are often close to beds (the *spatial* relationship), and that chairs and beds are often used for resting (the *affordance* relationship), one could infer the whereabouts of pillows even if only a bed and a chair were observed. In this paper, we question whether it is possible to computationally solve this task by injecting the commonsense knowledge within a scene graph representation [19, 12, 32], so that a machine can also reasonably localise an object in the unseen part of the scene, without the use of any visual/depth information.

In this work, we propose a new scene graph representa-

This project has received funding from the European Union's Horizon 2020 research and innovation programme "MEMEX" under grant agreement No 870743, and the Italian Ministry of Education, Universities and Research (MIUR) through PRIN 2017 - Project Grant 20172BH297: I-MALL and "Dipartimenti di Eccellenza 2018-2022".

tion, the **Spatial Commonsense Graph (SCG)**, having heterogeneous nodes and edges that embed the commonsense knowledge together with the spatial proximity of objects as measured in the partial 3D scan of the scene. The underlying intuition is that commonsense knowledge extracted from an external knowledge base is not specific to any observed visual scene, and thus allows for a better generalisation, but at the cost of a coarser localisation. At the same time, the objects' arrangement in the known portion of the scene is useful in providing better pairwise object distances, strengthening the estimate of the target object position. The main challenge here is devising a model that promotes the generalisation of commonsense while increasing the accuracy of the scene-specific metrics.

The proposed scene graph, as shown in Fig. 2, is first defined by nodes representing the known objects in the scene that are fully connected through edges representing the *proximity*, *i.e.* the relative distance between a pair of objects. We call this spatial representation the Spatial Graph (SG) of the known partial 3D scan. Then, the SG is further expanded into the SCG by adding and connecting nodes that represent concepts through relevant commonsense relationships extracted from ConceptNet [29].

The SCG is instrumental to address the localisation problem. In this work, we propose a two-stage solution, dubbed **SCG Object Localiser** (**SCG-OL**). First we predict the pairwise proximity between the target object node, having an unknown position, and each of the known object nodes through our graph-based *Proximity Prediction Network* (PPN), formulating the task as an edge regression problem. We then use our *Localisation Module* to compute the position of the target based on the pairwise distances. The localisation module estimates the most probable position as the intersection of the circular areas defined by all pairwise object distances. Note that by only using distances between pairs of objects, our model does not depend on the scene's reference frame, thus being considered agnostic to the coordinate system.

We also introduce a new dataset built from partial reconstructions of real-world indoor scenes using RGB-D sequences from ScanNet [7], which we will use as a benchmark for this novel problem. We construct the dataset to reflect different completeness levels of the reconstructed scenes. We define the evaluation protocol via a set of performance measures to quantify the localisation success and accuracy.

To summarise, our core contributions are the following:

- We identify a novel task of object localisation in partial scenes and propose a graph-based solution. We make available a new dataset and evaluation protocol, and show that our method achieves the best performance w.r.t. other comparing methods.
- We propose a new heterogeneous scene graph, the **Spatial Commonsense Graph**, for an effective integration between the commonsense knowledge and the spatial scene, using attention-based message passing for the graph updates to prioritise the assimilation of knowledge relevant to the task.
- We propose **SCG Object Localiser**, a two-staged localisation solution that is agnostic to scene coordinates. The distances between the unseen object and all known objects are first estimated and then used for the localisation based on circular intersections.

2. Related work

We will cover prior work related to the inference of scene graphs, the current dataset used for experimental validation and the use of commonsense for spatial reasoning.

Scene graph modelling and inference. Scene graphs were initially used to describe images of scenes based on the elements they contained and how they were connected. The work of [18] showed that for certain applications, e.g. Image Retrieval, the abstraction of higher-level image concepts was improving the results compared to using the standard pixel space. Since then, scene graphs have been successfully used in many other tasks such as image captioning [39, 40, 14] and visual question answering [27, 20].

Recently, the use of scene graphs has also been extended to the 3D domain, providing an efficient solution for 3D scene description. The 3D scene graph can vary from a simple representation of a scene and its content, in which the objects are nodes, and the spatial relationships between objects are the graph's edges [12, 32, 38]; to a more complex hierarchical structure that describes the scene at different levels: from the image level with description about the scene from only a certain point of view, moving up to a higher level description of objects, rooms and finally buildings [1]. The work of [42] uses a scene graph to augment 3D indoor scenes with new objects matching their surroundings using a message passing approach. A relatively similar task is indoor scene synthesis [33], in which the goal is to generate a new scene layout using a relation graph encoding objects as nodes and spatial/semantic relationships between objects as edges. A graph convolutional generative model synthesises novel relation graphs and thus new layouts. In [9] and [23] the authors use a 3D scene graph to describe the object arrangement, they then modify the scene graph and generate a new scene. Like these works, we use an underlying scene representation, but unlike them we embed commonsense knowledge into the scene graph. This way, our approach can better generalise to unseen rooms with unseen object arrangements by leveraging prior semantic knowledge.

Dataset for Object Localisation. Datasets existing in the literature are not suited for this type of object localisation task. For instance, Scene Synthesis datasets [34] do not have enough variability in the scene structure, as all environments represented are of identical shape and of similar size. Moreover, the scenes mostly contain the same set of objects.

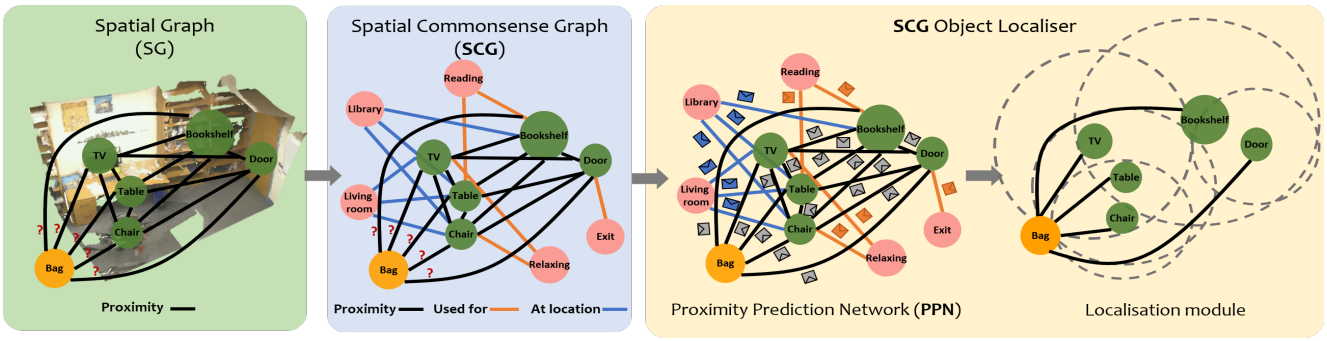


Figure 2: Overall architecture of our proposed approach. First, we construct a spatial commonsense graph (SCG) from the known scene by enriching the scene graph with concept relationships, resulting in edges of three types: *UsedFor* (**orange** edges), *AtLocation* (**blue** edges) and Proximity (**black** edges). The SCG is then fed into the Proximity Prediction Network (PPN) that performs message passing with attention to update the node features taking into consideration the heterogeneous edges. PPN then concatenates the node features of the target node and one of the scene object nodes and passes it through an MLP to predict the pairwise distance. The localisation module then uses the predicted pairwise distances to estimate the position of the target object within the area where most distances overlap.

These characteristics lead to datasets that do not reflect the real world and cannot be used to train models to be deployed in real indoor environments. Another major limitation of existing datasets is their assumption that the entire layout of the room is known and that the objects lie within the boundaries of the observed part of the scene [33, 22], which is atypical. In robotic applications like Visual Search [37, 13, 5], the robot only has partial information about the environment, that gets updated during navigation. In general, the searched object has to be found in the unexplored part of the scene, yet to be discovered. Our work is based on partially observed scenes and performs localisation without navigation.

Commonsense Knowledge in Neural Networks. Commonsense reasoning focuses on imitating the high level reasoning employed by humans when solving tasks. Typically, we do not only use the information directly related to the task, but also rely on knowledge gained through prior experience. The field of Natural Language Processing, [11] makes use of ConceptNet [29] to create richer, contextualised sentence embeddings with the BERT architecture [8]. In [2] , the authors utilise the knowledge graph Freebase (now Google Knowledge Graph) to enrich textual representations in a knowledge-based question answering system. In computer vision, [21] exploits commonsense knowledge using Dynamic Memory Networks for Visual Question Answering (VQA), stating it helps the network to reason beyond the image contents. In the scene graph generation task, [15] exploits the ConceptNet [29] knowledge graph to refine object and phrase features to improve the generalisation of the model. The authors state that the knowledge surrounding the subject of interest also benefits the inference of objects related to it, helping the model to generalise better and generate meaningful scene graphs. In this work, we exploit the commonsense knowledge to enrich a spatial scene representation used for predicting proximity among pairs of objects in a scene context.

3. Spatial Commonsense Graph (SCG)

Our model of the scene has the objective to embed commonsense knowledge into a geometric scene graph extracted from a partial scan of an area.

As illustrated in Fig. 2, we construct the SCG with nodes that are *i)* object nodes including all the observed objects in the partially known environment and any target unseen object to be localised, or *ii)* concept nodes that are retrieved from ConceptNet [29]. Each SCG is constructed on top of a Spatial Graph (SG) composed of object nodes that are fully connected. Each object node is further connected to concept nodes via the semantic relationships. The edges of SCG are of three heterogeneous types:

- *Proximity* relates the pairwise distances between *all* the object nodes given the partial 3D scan;
- *AtLocation* is retrieved from ConceptNet, indicating which environment the objects are often located in;
- *UsedFor* is retrieved from ConceptNet, describing the common use of the objects.

The proximity edges connect all the objects nodes of the SCG in a fully connected manner, while the semantic *AtLocation* and *UsedFor* edges connect each object node with its related concept nodes that are queried from ConceptNet (e.g. *bed AtLocation apartment* or *bed UsedFor resting*). The two semantic edge types provide useful hints on how objects can be clustered in the physical space, thus benefitting the position inference of indoor objects.

We formulate SCG as an undirected graph that is composed by a set of nodes $\mathcal{H} = \{\mathbf{h}_i | \ i \in (0, N]\}$, where $N = N_o + N_c$ is the total number of nodes in SCG with N_o the number of the object nodes and N_c the number of the

concept nodes. The D-vector $\mathbf{h}_i$ is the node's corresponding word embedding in NumberBatch [30] (i.e. $D = 300$). The edges are defined by the set $\mathcal{E} = \{\mathbf{e}_{i,j} | \, i, j \in (0, N], i \neq j\}$, where $e_{i,j}$ is the edge between node i and node j. Let $\mathcal{N}_i$ be the neighbouring nodes of node i connected by any edge. We use a 4-dimensional feature vector, i.e. $\mathbf{e}_{i,j} \in \mathbb{R}^4$, whose first three elements indicate the previously defined edge type in a one-hot manner while the last element is a scalar indicating the pairwise distance between two scene objects. Note that the distance is only measurable on the observed part of the 3D scan (i.e. between known object nodes). Otherwise, we initialise the distance value to -1 when the edges are *AtLocation*, *UsedFor*, or *proximity* edges involving the unknown target object node.

4. SCG Object Localiser (SCG-OL)

We define a two-stage solution to address the task of localising the arbitrary unobserved target object using the SCG. In the first stage, we propose a Proximity Prediction Network (PPN) on top of the SCG. PPN aims to predict the pairwise distances between the unseen target object and the objects in the partially known scene. In the second stage, our localisation module takes as input the set of pairwise distances and it outputs the position of the target object based on a probabilistic circular intersection. The following sections provide more details regarding the Proximity Prediction Network and the Localisation module.

4.1. Proximity Prediction Network

The goal of the PPN is to predict all the pairwise distances between the unseen object and the observed scene objects. We utilise a variant of the Graph Transformer [28] and update the nodes iteratively over the heterogeneous edges, to allow effective fusion between the commonsense knowledge and the metric measurements.

The input to the network is the set of node features $\mathcal{H}$ and the output is a new set of node features $\mathcal{H}^{'} = \{\mathbf{h}'_i | \, i \in (0, N]\}$, with $\mathbf{h}'_i \in \mathbb{R}^D$. Each node i in the graph is updated by aggregating the features of its neighbouring nodes $\mathcal{N}^i$ via two rounds of message passing. The resulting $\mathbf{h}'_i$ forms a *contextual* representation of its neighbourhood.

At each round of message passing, we first learn the attention coefficient $\alpha_{i,j}$ using a graph based version of the scaled dot-product attention mechanism [28], conditioned on each edge feature $\mathbf{e}_{i,j}$ from node j to node i, and on both nodes' features, $\mathbf{h}_i$ and $\mathbf{h}_j$. This allows the network to understand how important each neighbour is for the node representation's update, which is:

$$\mathbf{v}_j = W_v \mathbf{h}_j + b_v, \tag{1}$$

$$\hat{\mathbf{h}}_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij} (\mathbf{v}_j + e_{i,j}), \tag{2}$$

where W_v, b_v represent respectively the weight matrix and bias used to calculate the value vector $\mathbf{v}$ for the scaled dot-product attention mechanism. The updated state $\mathbf{h}'_i$ is then defined as:

$$\mathbf{h}'_i = ReLU(LNorm((1 - \beta_i)\hat{\mathbf{h}}_i + \beta_i W_r \mathbf{h}_i + b_r)), \tag{3}$$

where β_i is the output of a gated residual connection [28], which prevents all the nodes from converging into indistinguishable features. W_r, b_r represent the weight matrix and bias respectively used in the linear transformation of $\mathbf{h}_i$.

After message passing, we obtain the set of final node embeddings $\mathcal{H}^* = \{\mathbf{h}^*_i | \, i \in (0, N]\}$, with $\mathbf{h}^*_i \in \mathbb{R}^{2D} = Concat(\mathbf{h}_i, \mathbf{h'}_i)$, where $Concat(\cdot)$ represents a concatenation operation. This way, the final representation of each node contains both the original object embedding and the aggregated embedding of its context in the scene. Finally, we combine the features of the two nodes $\mathbf{h}^*_{i,t} = Concat(\mathbf{h}^*_i, \mathbf{h}^*_t)$ by concatenation, and predict the pairwise distances $\hat{d}_{i,t}$ between the target object node t and the observed object node i via fully connected layers.

SCG-OL loss. To train our PPN, we compute the Mean Square Error (MSE) between the predicted pairwise distances $\hat{d}_{i,t}$ of the object node i and the target node t and the set of ground-truth pairwise distances $d_{i,t}$. The loss is expressed as:

$$\mathcal{L}_{\text{MSE}}(\hat{d}, d) = \frac{1}{N_o - 1} \sum_{i=1}^{N_o - 1} (\hat{d}_{i,t} - d_{i,t})^2. \tag{4}$$

Note that the class of the target object can have multiple instances in the unknown part of the scene, i.e. multiple ground-truth positions. Our method, as a localiser, uses the GT position of the instance that is closest to the predicted position for the computation of the MSE loss.

4.2. Distances to position: Localisation module

In the localisation module, we solve the problem of converting the set of predicted object-to-object distances to a single position $\hat{\mathbf{p}}_t$ in the space that defines the position of the searched object in a bird's eye view. The distances $\hat{d}_{i,t}$ predicted by the PPN, and the known objects positions $\mathbf{p}_i$, can be used to define a set of circles of radius $\hat{d}_{i,t}$, centred in the positions $\mathbf{p}_i$. With perfect predictions, $\hat{\mathbf{p}}_t$ would be obtained as the point of intersection of all the circles. In this case we would need at least three known object nodes to unambiguously define $\hat{\mathbf{p}}_t$. For this reason, in this study we only consider instances with three or more known objects. Let us define $\hat{\mathbf{p}}_t$ as the point in the space that minimises the squared distance from all the circles:

$$\hat{\mathbf{p}}_t = \arg\min_{\mathbf{p}_t} \sum_{i}^{N_o - 1} (\|\mathbf{p}_t - \mathbf{p}_i\|_2 - \hat{d}_i)^2. \tag{5}$$

While it is possible to obtain a closed form solution of Eq. 5 via Linear Least Squares [36], this is not robust to noise in the measured distances, noise which is likely present in the PPN predictions. An alternative is to minimise this problem by brute force: we first subdivide the space into a grid and compute the sum of the residuals at each position. We then take the position with the lowest value and use it as an initial guess for the Nelder-Mead's simplex algorithm [24] to obtain the final estimate.

5. Experiments

We evaluate our proposed method on a new dataset of partially reconstructed indoor scenes. First, we provide the implementation details of our method followed by the metrics used for evaluation.

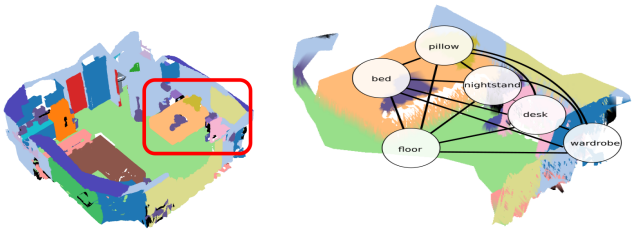
Implementation Details. We train our network using the Adafactor optimiser [26]. The network is trained for 100 epochs. The dimension of the first message passing projection is set to $D = 256$ and $2D$ for the second round. Both use 4 attention heads. For localisation, we ignore edges with a predicted distance of more than 5m, as such high distance values are not trustworthy for the localisation.

Evaluation Measures. We evaluate the performance in terms of both the proximity prediction and target object localisation. For the edge proximity prediction, we report the *mean Predicted Proximity Error (mPPE)*, which is the mean absolute error between the predicted distances and the ground-truth pairwise distances between the target object and the objects in the partially known scene. We quantify the localisation performance by the *Localisation Success Rate (LSR)*, which is defined as the ratio of the number of successful localisations over the number of tests. A localisation is considered successful if the predicted position of the target object is close to a target instance within a predefined distance. Unless stated differently, the distance threshold for a success is set to 1m. We consider LSR as the *main* evaluation measure for our task. Finally, to quantify the localisation accuracy among successful cases, we report the *mean Successful Localisation Error (mSLE)*, which is the mean absolute error between the predicted target position and the ground-truth position among all successful tests.

5.1. Dataset

We built a new dataset of partial 3D scenes using sequences available in ScanNet [7]. ScanNet contains RGB-D sequences taken at a regular frequency with a RGB-D camera. It provides the camera pose corresponding to each captured image, as well as the point-level annotations, i.e. class and instance id, for the complete Point Cloud Data (PCD) of each reconstructed scene.

The original acquisition frequency in ScanNet is very high (30Hz), meaning that most images are similar with redundant information for the scene reconstruction. We therefore use ScanNet_frames_25k, a subset provided in the ScanNet benchmark[1] with a frequency of about $1/100^{th}$ of the initial one. We further divide the full RGB-D sequences of each scene into smaller sub-sequences to reconstruct the partial scenes. We vary the length of the sub-sequences to reflect different levels of completeness of the reconstructed scenes. For each sub-sequence, we integrate the RGB-D information with the camera intrinsic and extrinsic parameters to reconstruct the PCD at the resolution of 5cm using Open3D [41]. The annotation for each point in the partial PCD is obtained by looking for the corresponding closest point in the complete PCD scene provided by ScanNet.



(a) Complete scene (b) Partial scene

Figure 3: The proposed dataset with (a) the complete scene from the ScanNet dataset, and (b) our reconstructed partial scene overlaid with the Spatial Graph.

From each partially reconstructed scene, we extract the corresponding Spatial Graph with its object nodes, i.e. the graph with only proximity edges (see Fig. 3 for an example). The nodes of the graph contain the object information: e.g. the *position*, defined as the centre of the bounding box containing the object, and the *object class*. We consider the position of each scene object as a 2D point (x, y) on the ground plane as most objects in the indoor scenes of ScanNet are located at a similar elevation. Each node is marked as *observed* if it represents an object in the partially known scene; or as *unseen* if it represents the object in the unknown part of the scene, i.e. the target object to localise.

Moreover, we construct our SCGs by adding two semantic relationships *AtLocation* and *UsedFor*, as well as the concepts that are linked by the relationships. We extract the concepts from ConceptNet by querying each scene object using the two semantic relationships. The query returns a set of related concepts together with their corresponding weight w indicating how "safe and credible" each related concept is to the query. We include a concept to the SCG only when it has a weight $w > 1$. Fig. 4 shows the average number of nodes linked by different types in the SCGs. On average, each SCG contains about 5 times more the concept nodes than the object nodes in the SG, demonstrating that rich commonsense knowledge is introduced within the SCG. The outliers in the boxplot visualisation are introduced by uncommon room types with a large amount of objects, e.g.

[1] http://kaldir.vc.in.tum.de/scannet_benchmark

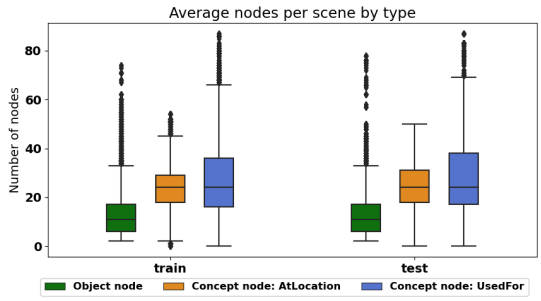


Figure 4: Average number of different types of nodes among the SCGs in the train and test split of the dataset.

libraries with several books. More statistics regarding our dataset can be found in the Supplementary Material.

Finally, we divide the dataset into training, validation and test sets. While we have access to the ScanNet training and validation data (1201 and 312 scenes respectively), we do not have access to their test data. To address this, we use ScanNet's validation sequences as our testing set, while we randomly sample a subset of scenes from the training set as the validation. By splitting ScanNet's sequences into partial reconstruction, we have 24896 partial scenes with 19461 partial scenes to be used for training and validation, and 5435 partial scenes for testing; where each partial scene has its corresponding SCG.

5.2. Experimental Comparisons

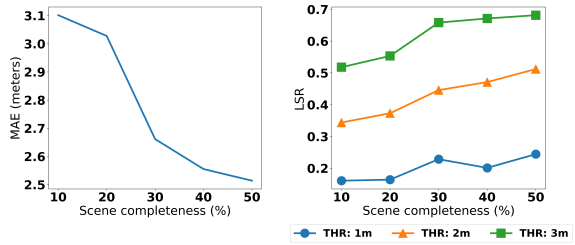
We validate **SCG-OL** by comparing its performance on our new dataset against a set on baselines and state-of-the-art methods for layout prediction. All the baselines follow the two-staged pipeline by first predicting the pairwise distances and then estimating the position with the localisation module. We summarise below all the evaluated approaches.

- **Statistics-based baselines** uses the statistics of the training set, i.e. the *mean*, *mode*, and *median* values of the pairwise distances between the target object and the scene objects, as the predicted distance.
- **MLP** learns to predict pairwise distances between the target object and every other observed object in the scene without considering the spatial nor the semantic context. The input to this model is a pair of the target object and the observed object with each object represented by a one-hot vector indicating the class, which is passed to a MLP that predicts pairwise distances.
- **MLP w Commonsense** learns to predict the pairwise distance between the target object and every other observed object in the scene without considering the spatial context. We first use GCN to propagate the conceptnet information to object nodes, then the features are passed to a MLP that predicts pairwise distances.
- **LayoutTransformer** [16] uses the transformer's self-attention to generate the 2D/3D layout in an auto-regressive manner. We describe the observed objects as a sequence of elements as in [16], where each element contains the object class and the position (x, y). We then feed the class of the target object to generate its corresponding position (x, y). For a fair comparison, we retrain the model with our training set.
- **GNN w\o Commonsense** is a variant of our approach that we have implemented to test the capability of our method when used without commonsense knowledge. The input is the Spatial Graph, which is composed only by the object nodes and proximity edges. The initial node features are not word embeddings, but are learned during training via an embedding layer.
- **SCG-OL(Ours)** is our method with two variants that are trained with learnable node embeddings and with pretrained node embeddings from ConceptNet, respectively.

Table 1: Methods comparison for object localisation in partial scenes. mPPE: mean Predicted Proximity Error. mSLE: mean Successful Localisation Error. LSR: Localisation Success Rate (the *main* measure). SG: Spatial Graph. SCG: Spatial Commonsense Graph.

Method	Data type	mPPE(m)↓	mSLE(m)↓	**LSR** ↑
Statistics-Mean	Pairwise	1.167	0.63	0.140
Statistics-Mode	Pairwise	1.471	0.63	0.149
Statistics-Median	Pairwise	1.205	0.64	0.164
MLP	Pairwise	1.165	0.62	0.143
MLP w Commonsense	Pairwise	1.090	0.64	0.163
LayoutTransformer [16]	List	-	**0.59**	0.176
GNN w\o Commonsense	SG	0.998	0.61	0.212
SCG-OL(Ours) - Learned Emb	SCG	0.974	0.61	0.234
SCG-OL(Ours) - Concept. Emb	SCG	**0.965**	0.61	**0.238**

Discussion. Table 1 reports the localisation performance measures in terms of mPPE, LSR, and mSLE, of all compared methods evaluated on our dataset comprised of partially reconstructed scenes. We can observe that methods with only pairwise inputs, e.g. statistics-based approaches or MLP, lead to worse performance compared to methods that account for other objects present in the observed scene. Nevertheless, introducing some semantic reasoning on top of these methods seems to improve the performances, as shown by MLP w Commonsense with an improvement of 2% on LSR compared to the standard MLP. LayoutTransformer directly predicts the 2D position of the target object by taking as input the list of all the observed scene objects and using the target class as the last input token. LayoutTransformer can better encode the spatial context and outperforms the statistic-based and MLP baselines. The graph-based methods achieve the highest performances, suggesting that for this problem a graph-based representation of the scene is more effective than a list-based one. Our **SCG-OL** that use the full SCG is able to improve on all metrics w.r.t. the **GNN** without Commonsense knowledge, when using either embeddings learned during training and pretrained ConceptNet embeddings. This shows how the SCG can effectively be



(a) Localisation error (b) LSR

Figure 5: Localisation performance over different levels of scene completeness. (a) The localisation error in terms of MAE between the estimated target position and the ground-truth position. (b) The LSR at different threshold levels.

used to improve the localisation problem. The better performances with the pretrained embeddings are likely due to the fact that these embeddings are learned on a broader set of tasks, thus including additional information that cannot be learned directly from the localisation task.

Fig. 5 shows how the completeness level of the known scene impacts the localisation performance of **SCG-OL**. Fig. 5a reports the mean absolute error (MAE) between the estimated position and the ground-truth position in function of the scene completeness. Note that the MAE is calculated on all the test cases including both the successful and the failed ones. In general, with an increasing scene completeness, **SCG-OL** can predict more accurately the position of the target object. Fig. 5b presents how the LSR varies as the scene gets more complete. In general, the LSR increases when the localisation error decreases. We report the LSR at three different threshold values, i.e. 1m, 2m, and 3m, where a larger threshold leads to a larger LSR value.

Qualitative results. Fig. 6 shows the qualitative results obtained using our method **SCG-OL**. Fig. 6a shows that the "bag" object class was successfully located near the area where the bag instances are. Similarly in Fig. 6b, the position of the second sofa in the room (target object) is correctly estimated at a position opposite to the first sofa in the SCG. Interestingly, Fig. 6c presents a failure case in which the method locates a television at the opposite side of the ground-truth television instance. Despite the estimated position being far from the real instance, the prediction is plausible due to the symmetry of the scene. We present more qualitative results in the Supplementary Material.

5.3. Ablation study

We further analyse **SCG-OL** to justify the usefulness of the commonsense relationships and the types of attention graph networks. We also investigated the impact of increasing the number of message passing layers, as well as using only the updated features when predicting the distances.

Which commonsense relationship is more important? In order to better understand the effects of using different commonsense relationships, we compare **SCG-OL** against its variants in which the SCG contains: i) only *Proximity* edges without commonsense relationships, ii) *Proximity* edges with *AtLocation* edges, iii) *Proximity* edges with *UsedFor* edges, and vi) *Proximity* edges with *AtLocation* and *UsedFor* edges. We report the main Localisation Success Rate (LSR) measure for all variants, as well as the scene average percentage of object nodes which are linked by 0, 1, or 2 types of semantic edges, i.e. *AtLocation* and *UsedFor* edges.

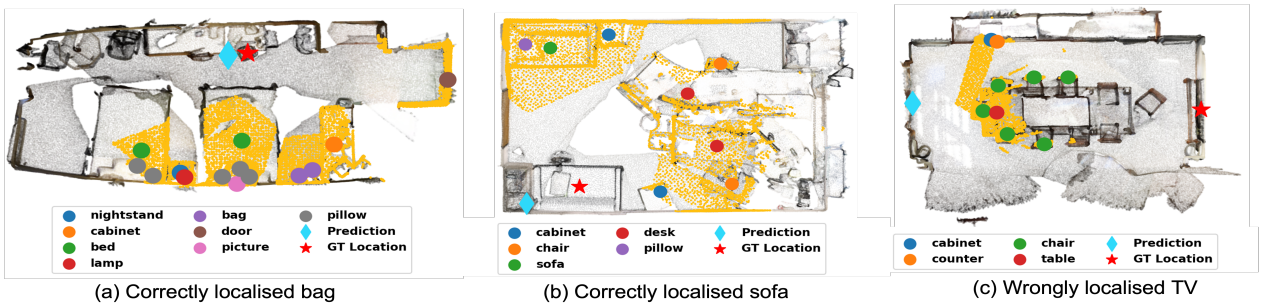
Table 2: Impacts of different ConceptNet relationships with the proposed **SCG-OL**. LSR: Localisation Success Rate.

Edge Types	Obj. linked by *n* semantic edges (%)			LSR ↑
	0	1	2	
Proximity	100	0	0	0.226
AtLocation, Proximity	8	92	0	0.233
UsedFor, Proximity	19	81	0	0.227
AtLocation, UsedFor, Proximity	8	12	80	**0.238**

Discussion. Table 2 shows that *AtLocation* is more effective than *UsedFor* for localising objects. A possible reason is that using the *AtLocation* edge leads to message passing among objects that are connected in the very same location, thus prioritising information more relevant to the localisation task. However, the best performance is obtained when the SCG can rely on all types of edges. Moreover, most of the object nodes (80%) are linked to concept nodes by both *AtLocation* and *UsedFor* edges. This boosts the knowledge fusion much more effectively than when only one type of semantic edges are used in the SCG.

Which attention network is more effective? We examine the usefulness of the attentional network of **SCG-OL** compared to other attention modules for the localisation task.

- **No attention:** We use GINEConv [17] during message passing without any attention module.
- **Sequential GAT:** We use GAT [31] as our attentional message passing layer. As GAT cannot distinguish heterogeneous edges and cannot be used with edge features, we use it sequentially for each semantic edge: first on the *AtLocation* edges, and then on the *UsedFor* edges. We then use GraphTransformer for the message passing on the proximity edges encoding the pairwise distances on the edge feature.
- **Sequential GATv2:** This method operates similarly to Sequential GAT, but employs GATv2 [4] for the attention layer instead of GAT.
- **HAN** [35]**:** This method defines multiple meta-path that connect neighbouring nodes either by specific node or edge types. It employs attentional message passing sequentially by first calculating the semantic-specific node embedding and then updating them by another round of attentional message passing. With SCG we define three sets of meta neighbours, i.e. the proximity neighbours, the *AtLocation* neighbours, and the *Used-*



(a) Correctly localised bag (b) Correctly localised sofa (c) Wrongly localised TV

Figure 6: Qualitative results obtained with **SCG-OL**. The partial known scene is coloured with a yellow background, while the unknown scene is indicated with grey. The coloured circles indicate the object nodes present in the SCG. The red star indicates the GT position of the target object, while the cyan diamond indicates the predicted positions. The network is able to correctly predict the position of a bag in (a) and a sofa in (b). In the failure case of (c), the network positioned the television at the wrong side of the table. Best viewed in colour.

Table 3: Impacts of different attentional networks for the object localisation task on our SCG. LSR: Localisation Success Rate.

Attentional Network	Propagation mode	LSR ↑
No attention	-	0.207
GAT [31]	Sequential	0.212
GATv2 [4]	Sequential	0.206
HAN [35]	Sequential	0.205
SCG-OL	Simultaneous	**0.238**

For neighbours connected by the specific edges.

Discussion. As shown in Table 3, different attention modules can produce results that vary greatly in terms of LSR. Among all, HAN achieves the worst performance. Sequential GAT and Sequential GATv2 were also not as effective as **SCG-OL**. This could be explained by a failure to integrate semantic and spatial information into the object node representation, as the semantic edges and the spatial context are aggregated separately, in a sequential manner. In contrast, **SCG-OL** performs simultaneous message passing on all the edge types, leading to the best localisation accuracy.

Do the number of message passing layers and the final node concatenation of SCG-OL make a difference? We examine a set of variants of our **SCG-OL** with between 1 to 4 message passing layers. Table 4 shows how using two message passing layers leads to the best performances: using a single layer leads to the worst results, and using more than two fails to further improve the performances. This happens because of the over-smoothing problem [6, 25], where after multiple message passing rounds, the embeddings for different nodes are indistinguishable from one another.

Given the best layer number, we also validate the choice of concatenating the original embedding to the aggregated *contextual* ones, instead of using only the aggregated features. Concatenation is more advantageous with a LSR score of 0.238 while directly using the aggregated node representation obtains a LSR of 0.224. Concatenation allows the network to develop a better understanding of the context after message passing while still remembering the initial representation.

Table 4: Impact of different numbers of message passing layers in our **SCG-OL**. LSR: Localisation Success Rate.

# Layers	1	2	3	4
LSR ↑	0.190	**0.238**	**0.238**	0.234

6. Discussion

Conclusions. We addressed the new problem of object localisation given a partial 3D scan of a scene. We proposed a novel scene graph model, the commonsense spatial graph, by augmenting a spatial graph with rich commonsense knowledge to improve the spatial inference. With such a graph formulation, we proposed a two-stage solution for unseen object localisation. We first predict the pairwise distances between the target node and the other object nodes using the graph-based Proximity Prediction Network, and then estimate the target object's position via circular intersection. We tested our proposed method and baselines on a new dataset composed of partially reconstructed indoor scenes, and showed how our solution achieved the best localisation performance w.r.t. the other compared approaches. As future work, we will investigate the applicability of our approach to large-scale scenarios in wider geographical areas, e.g. a city.

Limitations. The proposed localisation pipeline is not trainable end-to-end, as we enforce supervision on the intermediate information of the pairwise object distances rather than on the target object position. This choice allows the model to be reference-free, resulting in a better generalisation. Applying end-to-end supervision on the target position might lead to a more accurate localisation, but it is challenging to achieve without damaging the generalisation capabilities.

Broader impacts. Our dataset is built on top of ScanNet, featuring static indoor scenes without the involvement of human subjects. The dataset and the proposed scene graph formulation can facilitate and motivate further research towards scene understanding.

References

[1] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[2] Junwei Bao, Nan Duan, Ming Zhou, and Tiejun Zhao. Knowledge-based question answering as machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014. 3

[3] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 1

[4] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 7, 8

[5] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2020. 3

[6] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 8

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 3

[9] Helisa Dhamo, Fabian Manhardt, Nassir Navab, and Federico Tombari. Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[10] Wafa Elmannai and Khaled Elleithy. Sensor-based assistive devices for visually-impaired people: current status, challenges, and future directions. *Sensors*, 17(3):565, 2017. 1

[11] Keyur Faldu, Amit Sheth, Prashant Kikani, and Hemang Akabari. Ki-bert: Infusing knowledge context for better language and domain understanding. *arXiv preprint arXiv:2104.08145*, 2021. 3

[12] Paul Gay, James Stuart, and Alessio Del Bue. Visual graphs from motion (vgfm): Scene understanding with object geometry reasoning. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018. 1, 2

[13] Francesco Giuliari, Alberto Castellini, Riccardo Berra, Alessio Del Bue, Alessandro Farinelli, Marco Cristani, Francesco Setti, and Yiming Wang. Pomp++: Pomcp-based active visual search in unknown indoor environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. 3

[14] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[15] Jiuxiang Gu, Handong Zhao, Zhe L. Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[16] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S. Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 6

[17] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 7

[18] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016. 1

[20] Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. Visual question answering over scene graph. In *Proceedings of the First International Conference on Graph Computing (GC)*, 2019. 2

[21] Guohao Li, Hang Su, and Wenwu Zhu. Incorporating external knowledge to answer open-domain visual questions with dynamic memory networks. *arXiv preprint arXiv:1712.00733*, 2017. 3

[22] 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 Transactions on Graphics (TOG)*, 38(2):1–16, 2019. 3

[23] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B Tenenbaum. End-to-end optimization of scene layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[24] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 01 1965. 5

[25] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 8

[26] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of*

the International Conference on Machine Learning (ICML), 2018. 5

[27] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[28] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 4

[29] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2, 3

[30] Robyn Speer and Joanna Lowry-Duda. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proceedings of the International Workshop on Semantic Evaluation(SemEval)*, 2017. 4

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 7, 8

[32] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[33] 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 Transactions on Graphics (TOG)*, 38(4):132–147, 2019. 2, 3

[34] Kai Wang, Manolis Savva, Angel X. Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4):1 – 14, 2018. 2

[35] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *Proceedings of The World Wide Web Conference (WWW)*, 2019. 7, 8

[36] Yue Wang. Linear least squares localization in sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):1–7, 2015. 5

[37] Yiming Wang, Francesco Giuliari, Riccardo Berra, Alberto Castellini, Alessio Del Bue, Alessandro Farinelli, Marco Cristani, and Francesco Setti. Pomp: Pomcp-based online motion planning for active visual search in indoor environments. In *Proceedings of the British Machine Vision Virtual Conference (BMVC)*, 2020. 3

[38] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7515–7525, June 2021. 2

[39] Ning Xu, An-An Liu, Jing Liu, Weizhi Nie, and Yuting Su. Scene graph captioner: Image captioning based on structural visual representation. *Journal of Visual Communication and Image Representation*, 58:477–485, 2019. 2

[40] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[41] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 5

[42] Y. Zhou, Zachary While, and E. Kalogerakis. Scenegraphnet: Neural message passing for 3d indoor scene augmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2