Informing 3D Scene Graph Generation with Common-Sense Spatial Knowledge

Jared Strader, William Chen, Nathan Hughes, Alberto Speranzon, and Luca Carlone

Abstract—**In recent years, 3D scene graphs have emerged as a powerful representation for 3D environments, capturing information from low-level geometry to high-level semantics. A 3D scene graph represents an environment as a hierarchical graph grounded in the physical world, where nodes represent spatial concepts and edges represent relationships between concepts. While significant progress has been made for 3D scene graph generation, extensions to large-scale or novel environments remains an open problem. A key challenge in doing this is the lack of fully annotated datasets that include high-level concepts. Thus, there is a need for techniques that can leverage common-sense knowledge and symbolic reasoning to infer semantic labels that have not been seen at training time. In this work, we inform 3D scene graph construction with a spatial ontology using logic tensor networks (LTNs), largely reducing the need for supervised training data. We show that using a spatial ontology during training allows the models to predict previously unseen labels as well as enhance performance from 5.0% to 24.6% when training the proposed model using only 0.1% of the training data.**

I. Introduction

Motivation. A fundamental requirement for robots to interpret and execute high-level instructions from humans is a semantic understanding of the environment. In recent years, 3D scene graphs have emerged as an expressive hierarchical representation of complex scenes integrating *geometry* and *semantics* at multiple levels of abstraction. A 3D scene graph is a hierarchical graph [1] grounded in the physical world where nodes represent spatial concepts (*e.g.,* from low-level geometry to high-level semantics) and edges represent relations between concepts. Significant progress has been made in 3D scene graph generation [1–5], but, existing techniques are restricted to only a few types of abstractions (*e.g.,* objects, rooms, and buildings) and use hard coded algorithms to construct these layers. A key challenge in expanding 3D scene graphs across different environments is the lack of training datasets capturing high-level semantics. Thus, there is a need to develop techniques for constructing 3D scene graphs that can leverage common-sense spatial knowledge to infer semantic labels that have not been seen at training time.

Contribution. In this work, the main objective is to enable 3D scene graph construction in scenarios where large annotated datasets are not available. Specifically, we present a neuro-symbolic framework to inform 3D scene graph construction with common-sense spatial knowledge, largely reducing the need for supervised training data. For example, if annotations are not available in the training data, common-sense facts relating high-level concepts (*e.g.,* "kitchen") to low-level semantics detectable by the robot (*e.g.,* "sink", "counter") can be leveraged as a supervisory signal. We design a model composed of a graph neural network (GNN) and a multi-layer perceptron (MLP) and wrap the model in a logic tensor network (LTN) [6] to incorporate common-sense facts from a spatial ontology. While common-sense facts may be used directly to make predictions (*i.e., generalized truth queries* [6]), a learning-based framework is required to learn a model leveraging both the training data and common-sense facts. This allows for self-supervised learning of labels not present in the training data and prevents the model from making mistakes that would typically not be made by humans (*e.g.,* a room containing a "bed" is typically a "bedroom"). In the experiments, we show the proposed model is able to predict previously unseen labels and enhance performance with limited data.

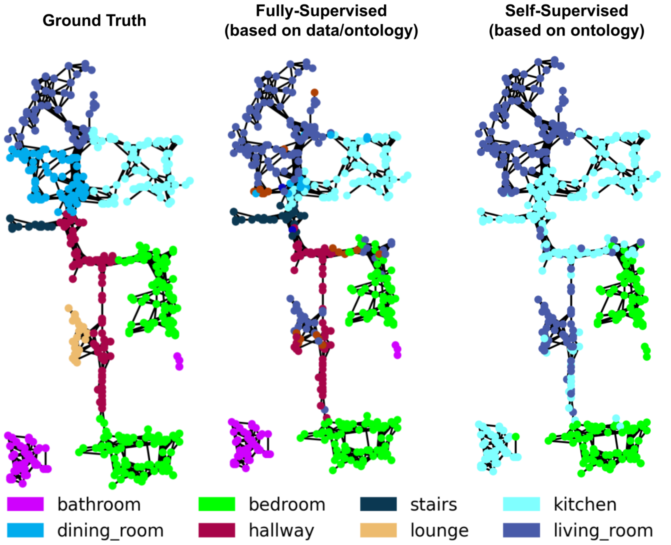
Ground Truth
Fully-Supervised (based on data/ontology)
Self-Supervised (based on ontology)
bathroom
dining_room
bedroom
hallway
stairs
lounge
kitchen
living_room

Fig. 1: Comparsion of grounded concepts in the places layer of a 3D scene graph using our fully-supervised approach (middle) against the ground-truth labels (left) and our self-supervised approach (right). A top-down view of the labeled 3D scene graph for each approach is shown, with each place node colored by either the ground-truth or predicted label. Our self-supervised approach achieves qualitatively similar accuracy to the fully-supervised approach despite not having trained with labels.

II. Related Work

3D Scene Graphs. The pioneering work of Armeni *et al.* [2] introduced the notion of 3D scene graphs as a hierarchical model of 3D environments. In [3, 7], the model was extended to construct 3D scene graphs from sensor data while adding

J. Strader, W. Chen, N. Hughes, L. Carlone are with the Laboratory for Information & Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA, USA, {jstrader,verityw,na26933,lcarlone}@mit.edu

A. Speranzon is with Lockheed Martin, Advanced Technology Labs, Eagan, MN, USA, alberto.speanzon@lmco.com

This work is partially founded by the ARL DCIST program, by the ONR program "Low Cost Autonomous Navigation & Semantic Mapping in the Littorals", and by Lockheed Martin Corporation.

richer hierarchy of layers. Several works explore constructing 3D scene graphs from geometric data such as Wald *et al.* [8] based on GNNs and Gothoskar *et al.* [9] based on MCMC. More recently, several approaches explore the creation of 3D scene graphs in real time [4, 5]. Hughes *et al.* [1] further discuss constructing higher-level concepts (*e.g.,* rooms) using a fully-supervised GNN approach. Finally, several works have successfully used 3D scene graphs for planning in robotics [10–12].

Ontologies. A fundamental requirement for robots to communicate a meaningful representation of a 3D environment is a common vocabulary describing the concepts and relations necessary to complete a given task. Such a vocabulary may be represented as an *ontology*. A significant effort has been made in the creation of common-sense ontologies and knowledge graphs [13–21]. In recent years, there has been a surge in applying these to problems such as 2D scene graph generation [22–25], image classification [26, 27], visual question answering [28–32], task planning [33–35], and representation learning [36, 37], to name a few.

III. Preliminaries

3D Scene Graphs. A 3D scene graph is a compact graphical representation of a 3D environment where nodes represent entities in the scene, and edges represent spatial or logical relationships between nodes. Formally, a 3D scene graph is a hierarchical graph [1] grounded in the physical world. A 3D scene graph is composed of groupings of nodes and edges called *layers*, which are subgraphs corresponding to different levels of abstraction. For instance, places in an indoor environment can be grouped in rooms, and rooms into buildings. Inter-layer edges denote groundings from higher-level concepts to lower-level concepts. While existing work on 3D scene graphs focus on grounding concepts in indoor environments, we extend the 3D scene graph model in [3, 5] to support constructing 3D scene graphs based on common-sense knowledge. Specifically, we retain the *mesh* and *places* layers as described in [1] as the low-level layers and construct the high-level layers based on the concepts desired to be grounded in the 3D scene graph. We refer to the high-level layers as *abstract layers* and construct them based on the task and/or capabilities of the robot encoded in a spatial ontology.

Logic Tensor Networks. A logic tensor network (LTN) is a neural-symbolic framework that uses the satisfaction of a knowledge base $\mathcal{K}$ as an objective for training neural networks. To allow for back-propagation of model parameters, LTNs use *real logic*, a differentiable first-order language, to define an interpretation[1] $\mathcal{I}$ to associate a tensor of real numbers to any term of the language and a real number in the interval $[0, 1]$ to any formula $\phi \in \mathcal{K}$. The interpretation defines the mapping of logical connectives interpreted using fuzzy logic semantics and quantifiers interpreted using fuzzy aggregators to real numbers. In practice, an LTN converts a knowledge base (i.e., set of logical formulas) into a computational graph that enables gradient-based optimization. In this work, we use LTNs to train a model using common-sense knowledge (*e.g.,* from a spatial ontolgy) to ground concepts in 3D scene graphs.

Spatial Ontology. We represent a spatial ontology as a graph where nodes represent spatial concepts and edges represent spatial relations between concepts. We consider nodes representing low-level concepts (*e.g.,* objects) and high-level concepts (*e.g.,* locations), and to simplify the problem, we only consider edges representing inclusion relation meaning an edge between a low-level concept A (*e.g.,* “sink”) and a high-level concept B (*e.g.,* “kitchen”) means that A is located in B (*e.g.,* a “sink” is located in a “kitchen”). In contrast to a 3D scene graph, which is a grounded representation, a spatial ontology is independent of the actual existence of such concepts. For example, the concept of “chair” in an ontology is not associated with a particular instance of a “chair”, but in a 3D scene graph, the concept of “chair” is grounded in the geometry of the scene creating an instance of a “chair”. In other words, an ontology is a graph representing a database of all possible concepts and their relations, while a 3D scene graph is a model of a specific environment.

IV. Leveraging Commonsense Spatial Knowledge for 3D Scene Graph Generation

In this section, we present a framework to inform 3D scene graph construction with a spatial ontology using GNNs and LTNs. We formulate this as a node classification problem using GNNs and incorporation the spatial ontology in the loss using LTNs to constrain the model to make predictions consistent with the spatial ontology. The proposed architecture for learning and inference is illustrated in Fig. 2, which we discuss in detail in the remainder of this section.

A. Architecture

Let the graph representing the places layer of a 3D scene graph be denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the set of labels we seek to infer denoted by $\mathcal{Y}$. Here, the labels $\mathcal{Y}$ correspond to the concepts to be grounded in the 3D scene graph, which represent labels of the next level up in the hierarchy (*e.g.,* rooms). We are given a set of graphs with partially labeled nodes $v \in \mathcal{V}$ each associated with a label $y_v \in \mathcal{Y}$. The features associated to the nodes $v \in \mathcal{V}$ are denoted by

$$\mathcal{X} = \{x_v\}_{v \in \mathcal{V}}, \tag{1}$$

where x_v is the feature vector for node $v \in \mathcal{V}$. Additionally, we encode the semantics of the children $C(v)$ (*i.e.,* nodes in the mesh layer) of each node $v \in \mathcal{V}$ as

$$\mathcal{Q} = \{q_v\}_{v \in \mathcal{V}}, \tag{2}$$

where $q_v = \sum_{u \in C(v)} \tilde{y}_u / ||\sum_{u \in C(v)} \tilde{y}_u||_1$ with $\tilde{y}_u$ representing the one-hot encoding for the label y_u of node $u \in \mathcal{V}$. The objective is to design a model to predict the labels of the nodes in the places layer for previously unseen graphs. In contrast to standard techniques, we wrap the model in an LTN, and we use the satisfaction of a knowledge base $\mathcal{K}$ (discussed in detail in the following section) as a loss during training. We

[1] The interpretation is the assignment of constants, variable and logical symbols to tensors or operations on tensors in the field of real numbers. Here, we prefer using “interpretation”, standard in the logics literature, compared to “grounding” used in [6].

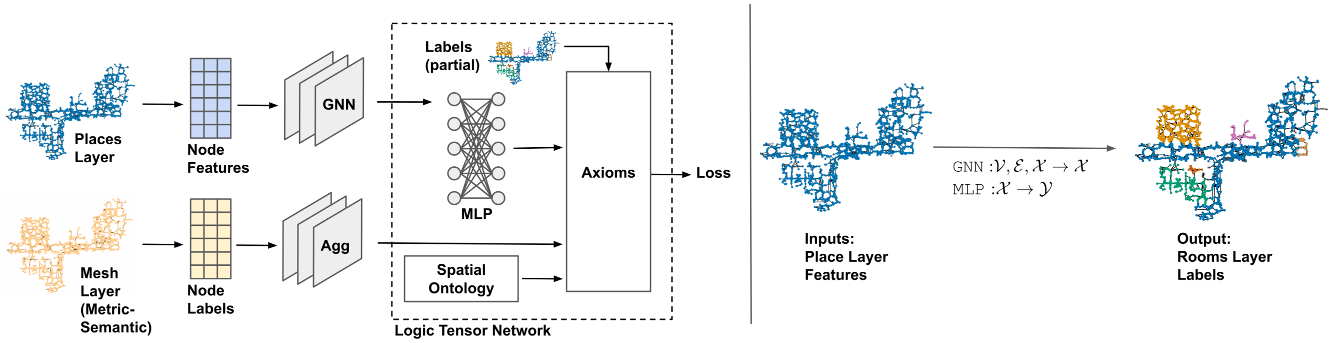
Places Layer
Node Features
GNN
Labels (partial)
MLP
Axioms
Loss
Mesh Layer (Metric-Semantic)
Node Labels
Agg
Spatial Ontology
Logic Tensor Network
GNN :$\mathcal{V}, \mathcal{E}, \mathcal{X} \to \mathcal{X}$
MLP :$\mathcal{X} \to \mathcal{Y}$
Inputs: Place Layer Features
Output: Rooms Layer Labels

Fig. 2: (Left) Architecture for training phase. The place features are processed by the GNN to compute node embeddings, and MLP takes the embeddings as input and outputs the logits. The MLP is wrapped in an LTN to compute the satisfaction of the knowledge base using the data-driven predicates based on ground truth labels and ontology-driven predicates based on the aggregated semantics of the mesh layer. The parameters of both the GNN and MLP are updated based on the loss. (Right) Architecture for inference phase. The LTN is dropped during the inference phase, and the trained GNN and MLP predict the node labels of the room layer given the place features.

decompose the model into a GNN for computing embeddings and an MLP for predicting the node labels:

$$\texttt{GNN} : \mathcal{V}, \mathcal{E}, \mathcal{X} \to \mathcal{X} \tag{3}$$

$$\texttt{MLP} : \mathcal{X} \to \mathcal{Y}. \tag{4}$$

where the parameters for the entire model include both the GNN and MLP parameters $\theta = \{\theta_{\texttt{GNN}}, \theta_{\texttt{MLP}}\}$. We aim to learn the parameters θ^* that maximize the satisfiability of the knowledge base $\mathcal{K}$:

$$\theta^* = \underset{\theta \in \Theta}{\arg\min} \left(1 - \underset{\phi \in \mathcal{K}}{\texttt{SatAgg}} \, \mathcal{I}_\theta(\phi) - \lambda R(\theta) \right) \tag{5}$$

where R is a regularization term weighted by λ and $\texttt{SatAgg} : [0,1]^* \to [0,1]$ is an aggregating operator for the formulas in the knowledge base.

B. Knowledge Base

Axioms. The axioms in the knowledge base are logical formulas composed of constants, variables, and predicates. We distinguish the axioms by whether they are data-driven ϕ_D (*i.e.,* interpreted from labeled data) or ontology-driven ϕ_Ω (*i.e.,* interpreted from the ontology and unlabeled data). We define the knowledge base $\mathcal{K}$ that our model seeks to satisfy by the following axioms:

$$\phi_D : \forall \texttt{Diag}\,(x, y)\ \texttt{IsClass}(x, y) \tag{6}$$

$$\phi_\Omega : \forall \texttt{Diag}\,(x, q)\ \texttt{Exp}(\Omega, q) \to \ \texttt{Sat}(x, \Omega, q). \tag{7}$$

where $\texttt{Diag}(\cdot)$ denotes diagonal quantification as described in [6] and Ω is an $n \times m$ matrix with entries ω_{ij} encoding the ontology where n is the number of high-level semantics (*i.e.,* labels of the room layer) and m is the number of low-level semantics (*i.e.,* labels of the mesh layer). If an edge exists between high-level label i and low-level label j in the ontology, $\omega_{ij} > 0$; otherwise, $\omega_{ij} = 0$. This implies that the high-level label i is a valid label according to the spatial ontology for a node with low-level label j. The data-driven axioms ϕ_D promote the model to make predictions that agree with the ground truth labels, and the ontology-driven axioms ϕ_Ω promote the model to make predictions that agree with the spatial ontology despite the ground truth labels.

Predicates. Next, we define the interpretations of the predicates, which are functions projecting constants and variables onto values in the interval $[0, 1]$. The interpretation of the $\texttt{IsClass}$ predicate is given by

$$\mathcal{I}(\texttt{IsClass}|\theta) : x, \tilde{y} \mapsto \tilde{y}^T \cdot \texttt{softmax}(\texttt{MLP}(x)), \tag{8}$$

and the interpretations of the $\texttt{Exp}$ and $\texttt{Sat}$ predicates are given by

$$\mathcal{I}(\texttt{Exp}) : \Omega, q \mapsto \texttt{sum}(\Omega \cdot q) \tag{9}$$

$$\mathcal{I}(\texttt{Sat}|\theta) : x, \Omega, q \mapsto \texttt{softmax}(\texttt{MLP}(x))^T \cdot \Omega \cdot q. \tag{10}$$

where the notation $(\cdot|\theta)$ denotes the predicate is parameterized by θ; thus, the model parameters θ for the GNN and MLP are updated together during training based on the satisfaction. In words, the $\texttt{IsClass}$ predicate provides a similar function to training a model using a cross-entropy in typical supervised scenarios with the exception that the $\texttt{IsClass}$ predicates return a value between $[0, 1]$. The $\texttt{Exp}$ predicate returns values close to 1 if the spatial ontology includes relations between the low-level labels of mesh nodes in the data (*e.g.,* encoded by q) and the high-level labels in the ontology. In contrast, if the ontology does not include these relations, $\texttt{Exp}$ returns values close to 0. The $\texttt{Sat}$ predicate returns values close to 1 if high-level labels predicted by the model match the high-level labels in the ontology adjacent to the low-level labels of mesh nodes (*e.g.,* encoded by q).

V. Experiments

A. Experimental Setup

Dataset. We utilize the Matterport3D (MP3D) dataset [38], which is an RGB-D dataset consisting of 90 indoor scenes. We use the Habitat simulator [39] along with Hydra [1] to generate 3D scene graphs for 4 random trajectories, approximately 100 meters in length, in each of the 90 scenes. Each 3D scene graph is produced using the ground-truth 2D semantic segmentation provided by Habitat. The provided label space has 40 semantic labels (*i.e.,* mpcat40 [38]), of which we discard the labels *gym_equipment* and *beam*. We manually build a spatial ontology based on common-sense knowledge (shown in Table I); however, the spatial ontology only captures a few intuitive relations in the dataset (*e.g., sink* is contained

	Bedroom	Bathroom	Kitchen	Living Room
Bed	1	0	0	0
Drawers	1	0	0	0
Toilet	0	1	0	0
Shower	0	1	0	0
Sink	0	0	1	0
Appliance	0	0	1	0
Cabinet	0	0	1	0
Sofa	0	0	0	1
Fireplace	0	0	0	1

TABLE I: The spatial ontology used in the experiments where rows denote low-level labels corresponding to the mesh and the columns denote high-level labels to be grounded in the places layer.

Layers	3
Hidden Dimension	32
Learning Rate	0.001
Dropout	0.25
L_2 Regularization	1e-05
Attention Heads	4
Activation	RELU

TABLE II: Parameters used for training the proposed model where the same parameters are used for all predicate configurations.

in *bathroom* in the dataset although not included in the spatial ontology). We use the official train, test, and validation splits of the 90 MP3D scenes [38].

Training. We train the model using different axioms configurations that include (i) only the data-driven axioms ϕ_D, (ii) only the ontology-driven axioms ϕ_Ω, and (iii) both the data-driven and ontology-driven axioms. The models trained using only data-driven axioms are used as a baseline. The node features for the places are the concatenation of the position and the word2vec [40] encoding of the aggregated semantic labels of the child nodes in the mesh layer. For the GNN, we use a graph attention network (GAT) [41, 42], but any message passing architecture is applicable to our framework. We tune the hyper-parameters using a grid search over a set of predefined values, and the hyperparameters remain constant between different axiom configurations as the grid search returns the same hyperparameters for each configuration. The hyperparameters used during training are given in Table II. For the connectives and quantifiers, we use a product configuration except for the implication operator where the Goguen operator is used instead of the Reichenbach operator [43]. We using stochastic gradient descent (SGD) with the Adam optimizer, and we run the optimization until the loss is converged (*e.g.*, loss changes less than 0.0001 for 3 epochs) or 500 epochs is reached. After each epoch, the model with the best validation accuracy is retained for testing. Each model is trained 3 times using random initializations, and the statistics are reported in the results. We implement the learning architectures using PyTorch Geometric [44] and LTNTorch [6].

B. Results

The goal is to test the model in scenarios with limited data and previously unseen labels. We simulate such scenarios by training the model (i) only using a fraction of the training dataset and (ii) holding out labels for a set of concepts. In both cases, the model is tested on the entire test dataset. The results are presented in Table III for keeping all labels and in Table IV for holding out a set of labels. The labels held out capture concepts both contained and not contained in the spatial ontology.[2] As shown in Table III, the incorporation of the ontology-driven axioms provide similar or better accuracy than only the data-driven axioms. However, the improvement is significant when using the ontology-driven axioms if concepts are missing in the training data. This is due to the fact that the ontology-driven axioms incorporate a self-supervised component to the loss even if labels are unknown. This is illustrated in Fig. 1 where the predicted labels using only ontology-driven axioms are compared against using both the ontology-driven and data-driven axioms. The *stairs*, *lounge*, *hallway*, and *dining room* are misclassified since not included in the spatial ontology, but *bedroom*, *kitchen*, and *living room* are classified more accurately. Furthermore, misclassifying a *kitchen* with a *bathroom* is a reasonable approximation given the ontology does not associate *bathroom* with *sink*.

	% of Data Kept for Training (fully known labels)		
	100%	1%	0.1%
ϕ_D (baseline)	0.487 ± 0.017	$\mathbf{0.443 \pm 0.016}$	0.053 ± 0.018
ϕ_Ω	0.245 ± 0.001	0.246 ± 0.001	0.242 ± 0.004
ϕ_D, ϕ_Ω	$\mathbf{0.487 \pm 0.004}$	0.405 ± 0.016	$\mathbf{0.246 \pm 0.004}$

TABLE III: Accuracy for different holdout percentages of training data. No data held out during testing. Best in bold.

	% of Data Kept for Training (partially unknown labels)		
	100%	1%	0.1%
ϕ_D (baseline)	0.168 ± 0.010	0.157 ± 0.038	0.035 ± 0.029
ϕ_Ω	0.249 ± 0.001	0.246 ± 0.001	0.242 ± 0.004
ϕ_D, ϕ_Ω	$\mathbf{0.255 \pm 0.002}$	$\mathbf{0.251 \pm 0.004}$	$\mathbf{0.244 \pm 0.001}$

TABLE IV: Accuracy for different holdout percentages of training data with labels masked for *bathroom*, *kitchen*, *hallway*, *bedroom*, *living room*, and *family room*. No data held out during testing. Best in bold.

VI. Conclusion

While the proposed framework was demonstrated on predicting high-level semantics in the places layer, this procedure can be applied sequentially to produces additional layers in the hierarchy. Furthermore, the experiments were performed in an indoor scenario due to the abundance of indoor datasets for benchmarking; however, the proposed framework is not restricted to indoor abstractions. Instead, the framework allows for self-supervised training on unlabeled datasets to infer high-level concepts defined in a spatial ontology. We show in the experiments that the proposed framework allows the prediction of labels unseen during training and significant benefits in scenarios where data is limited. While the main limitation of the proposed framework is the reliance on a spatial ontology, existing knowledge bases may be leveraged depending on the application, or language models may be used to infer common-sense spatial knowledge [45], which show promise in filling this gap in the future.

[2] The labels held out include: *bathroom*, *kitchen*, *hallway*, *bedroom*, *living room*, and *family room*. The labels held in include: *closet*, *entryway*, *garage*, *library*, *laundry room*, *conference room*, *lounge*, *office*, *recreation room*, *stairs*, *utility room*, *theatre room*, *gym*, *balcony*, *classroom*, *dining room*, *spa*, and *unknown*.

References

[1] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *arXiv preprint: 2305.07154*, 2023. 1, 2, 3

[2] I. Armeni, Z. He, J. Gwak, A. Zamir, M. Fischer, J. Malik, and S. Savarese, "3D scene graph: A structure for unified semantics, 3D space, and camera," in *Intl. Conf. on Computer Vision (ICCV)*, 2019, pp. 5664–5673. 1

[3] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: from SLAM to spatial perception with 3D dynamic scene graphs," *Intl. J. of Robotics Research*, vol. 40, no. 12–14, pp. 1510–1546, 2021. 1, 2

[4] S. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "SceneGraphFusion: Incremental 3D scene graph prediction from RGB-D sequences," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[5] N. Hughes, Y. Chang, and L. Carlone, "Hydra: a real-time spatial perception engine for 3D scene graph construction and optimization," in *Robotics: Science and Systems (RSS)*, 2022. 1, 2

[6] S. Badreddine, A. d'Avila Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022. 1, 2, 3, 4

[7] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans," in *Robotics: Science and Systems (RSS)*, 2020. 1

[8] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3D semantic scene graphs from 3D indoor reconstructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3961–3970. 2

[9] N. Gothoskar, M. Cusumano-Towner, B. Zinberg, M. Ghavamizadeh, F. Pollok, A. Garrett, J. Tenenbaum, D. Gutfreund, and V. Mansinghka, "3DP3: 3D scene perception via probabilistic programming," in *ArXiv preprint: 2111.00312*, 2021. 2

[10] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti, "Taskography: Evaluating robot task planning over large 3D scene graphs," in *Conference on Robot Learning (CoRL)*. PMLR, Jan. 2022, pp. 46–58. 2

[11] Z. Ravichandran, L. Peng, N. Hughes, J. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3D scene graphs using graph neural networks," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022.

[12] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6541–6548. 2

[13] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, Nov. 1995. 2

[14] D. B. Lenat, "Cyc: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.

[15] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proceedings of the International Conference on Formal Ontology in Information Systems*, 2001, pp. 2–9.

[16] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a web of open data," in *Semantic Web*. Springer, 2007, pp. 722–735.

[17] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A large ontology from wikipedia and WordNet," *Journal of Web Semantics*, vol. 6, no. 3, pp. 203–217, 2008.

[18] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1250.

[19] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. Hruschka, and T. Mitchell, "Toward an architecture for never-ending language learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, no. 1, 2010, pp. 1306–1313.

[20] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, p. 78–85, sep 2014.

[21] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: an open multilingual graph of general knowledge," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17. AAAI Press, 2017, p. 4444–4451. 2

[22] A. Zareian, S. Karaman, and S.-F. Chang, "Bridging knowledge graphs to generate scene graphs," in *European Conf. on Computer Vision (ECCV)*. Springer, 2020, pp. 606–623. 2

[23] Y. Guo, L. Gao, X. Wang, Y. Hu, X. Xu, X. Lu, H. T. Shen, and J. Song, "From general to specific: Informative scene graph generation via balance adjustment," in *Intl. Conf. on Computer Vision (ICCV)*, 2021, pp. 16 383–16 392.

[24] F. Amodeo, F. Caballero, N. Díaz-Rodríguez, and L. Merino, "Og-sgg: Ontology-guided scene graph generation—a case study in transfer learning for telepresence robotics," *IEEE Access*, vol. 10, pp. 132 564–132 583, 2022.

[25] Z. Chen, S. Rezayi, and S. Li, "More knowledge, less bias: Unbiasing scene graph generation with explicit ontological adjustment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4023–4032. 2

[26] Y. Movshovitz-Attias, Q. Yu, M. C. Stumpe, V. Shet, S. Arnoud, and L. Yatziv, "Ontological supervision for fine grained classification of street view storefronts," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1693–1702. 2

[27] D. Porello, M. Cristani, and R. Ferrario, "Integrating Ontologies and Computer Vision for Classification of Objects in Images," in *Workshop on Neural Cognitive Integration*, 2015, p. 15. 2

[28] S. Aditya, Y. Yang, C. Baral, Y. Aloimonos, and C. Fermüller, "Image understanding using vision and reasoning through scene description graph," *Computer Vision and Image Understanding*, vol. 173, pp. 33–45, 2018. 2

[29] F. Gardères, M. Ziaeefard, B. Abeloos, and F. Lecue, "ConceptBert: Concept-aware representation for visual question answering," in *Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 489–498.

[30] K. Marino, X. Chen, D. Parikh, A. Gupta, and M. Rohrbach, "KRISP: Integrating implicit and symbolic knowledge for open-domain knowledge-based VQA," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 111–14 121.

[31] W. Zheng, L. Yin, X. Chen, Z. Ma, S. Liu, and B. Yang, "Knowledge base graph embedding module design for visual question answering model," *Pattern Recognition*, vol. 120, p. 108153, 2021.

[32] Y. Ding, J. Yu, B. Liu, Y. Hu, M. Cui, and Q. Wu, "MuKEA: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5089–5098. 2

[33] H. Chen, H. Tan, A. Kuntz, M. Bansal, and R. Alterovitz, "Enabling robots to understand incomplete natural language instructions using commonsense reasoning," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1963–1969. 2

[34] A. Daruna, L. Nair, W. Liu, and S. Chernova, "Towards robust one-shot task execution using knowledge graph embeddings," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 118–11 124.

[35] S. Tuli, R. Bansal, R. Paul *et al.*, "ToolTango: Common sense generalization in predicting sequential tool interactions for robot plan synthesis," *The Journal of Artificial Intelligence Research*, vol. 75, pp. 1595–1631, 2022. 2

[36] J. Hao, M. Chen, W. Yu, Y. Sun, and W. Wang, "Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 1709–1719. 2

[37] J. H. Kwak, J. Lee, J. J. Whang, and S. Jo, "Semantic grasping via a knowledge graph of robotic manipulation: A graph representation learning approach," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9397–9404, 2022. 2

[38] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017. 3, 4

[39] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3

[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprints arXiv:1301.3781*, 2013. 4

[41] D. Busbridge, D. Sherburn, P. Cavallo, and N. Y. Hammerla, "Relational graph attention networks," *arXiv preprint arXiv:1904.05811*, Apr. 2019. 4

[42] J. Lee, R. Rossi, S. Kim, N. Ahmed, and E. Koh, "Attention models in graphs: A survey," *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 6, Nov. 2019. 4

[43] E. van Krieken, E. Acar, and F. van Harmelen, "Analyzing differentiable fuzzy logic operators," *Artificial Intelligence*, vol. 302, p. 103602, 2022.

4

[44] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *Intl. Conf. on Learning Representations (ICLR) Workshop on Representation Learning on Graphs and Manifolds*, 2019. 4

[45] W. Chen, S. Hu, R. Talak, and L. Carlone, “Leveraging large language models for robot 3D scene understanding,” *arXiv preprint: 2209.05629*, 2022. 4