CS 231N Project Report: 2D Relational Search Task with Image and Graph Feature Learning with Graph Attention

Minjune Hwang
Department of Computer Science
Stanford University
mjhwang@stanford.edu
mjhwang@stanford.edu

Abstract

Visual search task is one of the fundamental tasks in Embodied AI. In the problem, an agent is asked to navigate to a target instance, using visual inputs of the current scene. As such, previous methods often focus in leverage different modalities with vision approaches, such as raw image input, object detection, or depth estimation. However, an agent can learn relations between objects in the scene and actively update and use this relational information when navigating to the target object, using a structured representation like graphs with different types of nodes and edges. This is especially true when there exists similar objects with different relational state (e.g. above or below a table). In this work, we created a 2D relational object choice environment in which the agent has to choose a relevant object given by a relational goal specification. In the environment, this paper shows that if an agent can update and learn from this additional modality of relational graphs, when combined with other visual modalities, can help itself to navigate to target objects more effectively. Also, experimental results demonstrated that leveraging graph attention is beneficial when multiple occlusion and distractor objects are present in the observation.

Disclaimer: Previously, my proposed topic in the project proposal was ”Communication via Visual Activation Map in Cooperative Multi-Agent RL”. However, I decided to work on the project over the summer, and changed my project topic to the current topic of relational visual search with object detection and graphical neural network, as a part of an ongoing research project at Stanford Vision Lab (SVL). I worked on this project under supervision of the Ph.D. student Michael Lingelbach at SVL.

1. Introduction

Embodied AI refers to an intelligent machine (often a robot) that is capable of learning and interacting within a physical or virtual environment, with its physical or virtual embodiment. Among various fundamental skills required for such robots, the ability to navigate and search an object is a crucial skill to accomplish high-level tasks in a physical or virtual environment. The agent has to leverage visual inputs from the environment to efficiently navigate and identify a target object, potentially with different set of features or information from those visual inputs. These include depth information, bounding boxes or point clouds of objects, or other visual features. Thanks to the advance in computer vision and pattern recognition, these methods often leverage convolutional neural networks to extract complex features from high-dimensional input images, and leverage such information to represent the current state in a feature space. These features are then used to learn an efficient policy to execute the given task in the form of sequential decision making problems.

However, there often exists another rich feature in the real-world problems: relations between objects. For instance, an apple can be on top of a table or under a table. A bottle of water can be inside a cabinet. Many objects can be in a given room, and rooms can be either connected or not connected. These kinds of relational information can be represented an edge in a scene graph. This scene graph can be leveraged as an additional feature to represent the current state, and can be dynamically updated as the agent navigates the scene.

In this project, we aim to suggest and compare various models to efficiently process the visual features as well as relational features (scene graphs) for relational visual navigation tasks, where the agent is asked to either choose or navigate to a target object with a target relation. Additionally, we experiment to show that graph attention mechanism can positively affect the model performance in presence of occlusion and distractor objects.

2. Related Works

Li et al. adds a graph as an additional modality in instance-level object navigation tasks, but only uses CNNs on the image of the scene graph, rather than using GCNNs.

Ravichandran et al. uses GNNs to learn hierachical representations in navigation policies. Their work seems to provide an effective navigation policies on 3D scenes. We aim to extend this method by using heterogenous GCNN with attention to efficiently extract and learn goal information is relational visual navigation tasks.

The importance of the attention mechanism in human visual search has been long recognized in vision science [1, 5, 7]. Developing an attention mechanism for an embodied AI agent is an active research topic. Such attention can be in the form of saliency map on egocentric RGB images [3], or weights on 2D maps [**?**, 4]. Attention is a useful tool that can potentially help leverage large scene graphs in visual search, but an appropriate attention mechanism still needs to be explored since there are many forms of attention in graph neural networks developed for different purposes [**?**, 6, 8]. Our work builds upon existing research and proposes a framework for incorporating task-driven attention into a scene graph representation for the challenging hierarchical relational object navigation problem.

3. Experimental Design

We design our experiments to answer the following questions.

- **1**: Does the scene graph help the agent to learn faster to choose correct objects in relational choice tasks?

- **2**: Does the additional attention mechanism help GNN models to learn faster? If so, should we leverage learned attention model or fixed / heuristic-based attention model?

3.1. Model Design

In our tasks, the robot is mainly given two observations: RGB data and depth data. These observations represent the local, high resolution features. I aim to use convolutional neural networks to extract image features from these observations. On top of this, I also have a graph that represents the relations between objects. I aim to use graph convolutional networks to extract global features from this data. The graph is dynamically updated as the robot navigates in the scene, and thus it contains episodic memory that is required for high level planning. Figure 1 shows the three main components of the model: a convolution encoder for RGB input, a convolution encoder for depth input, and a graph neural network for scene graphs. In our 2D tasks, we do not use the convolution encoder for depth input.

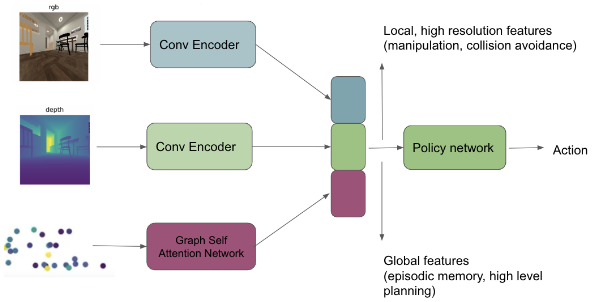

rgb
depth
Conv Encoder
Conv Encoder
Graph Self Attention Network
Local, high resolution features (manipulation, collision avoidance)
Policy network
Action
Global features (episodic memory, high level planning)

Figure 1. A model diagram

For the convolution encoder, we use ResNet structure with the environment RGB observation size (input size: 128x128). For the graphical neural network encoder for scene graphs, I used Heterogeneous graph transformer (HGT) and graph attention, as shown below.

3.1.1 Heterogeneous graph transformer (HGT)

To compute a per node embedding, the graph is passed through three heterogeneous graph transformer layers [2] with ReLU activations. The HGT convolutions use distinct edge-based matrices for each edge type when computing attention, allowing the model to learn representations conditioned on different edge types, rather than connectivity alone.

3.1.2 Graph attention pooling

We additionally introduce an attention mechanism at the final pooling layer in order to effectively aggregate task-relevant nodes. The task-driven attention mask is constructed by assigning a value of 1 to all nodes for which the semantic category matches any semantic category found in the current episodic goal description and 0 otherwise. The mask is used as the weights for a weighted mean pooling of the node embeddings to extract a graph embedding. For this attention, we have two possible designs. First is the task-driven (TD) attention, which fixes the weight of 1 to objects of interest and 0 to other distractor objects. Second is the learned (LN) attention, which has the separate GNN model to learn the weights for graph attention.

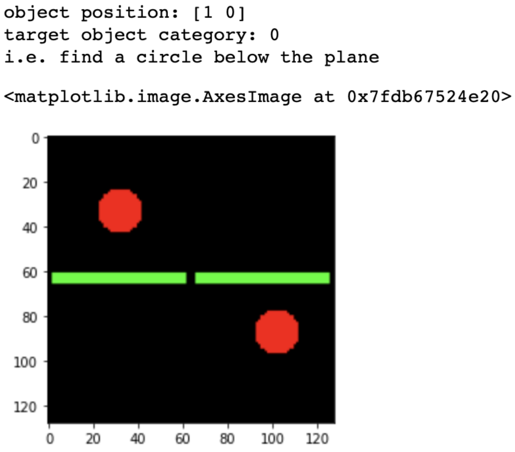
3.2. Model Lists

As described above, we aim to investigate the effect of GNNs and graph attention mechanism in 2D relational choice tasks. Thus, we test the following combinations to attest the effect. SG refers to scene graph of the environment.

- CNNs (RGB)
- CNNs (RGB) + GNNs (SG)
- CNNs (RGB) + GNNs (SG) + Task-Driven (TD) Attention
- CNNs (RGB) + GNNs (SG) + Learned (LN) Attention

3.3. Data & Environment Design

For the experiment, I have designed a 2D environment for relational choice tasks. The agent is presented with a symmetric 2D environment with two circles and two rectangles: one circle above and the other below their respective rectangle. The agent is given a goal specification of "circle above rectangle" or "circle below rectangle" and must choose one of two possible actions (`left`/`right`) to select the side of the environment that satisfies the goal definition. The agent is given a reward of 1 if it selects the action that matches the goal descriptor, and a reward of 0 otherwise. The episode terminates after one step (bandit problem).



```
object position: [1 0]
target object category: 0
i.e. find a circle below the plane

<matplotlib.image.AxesImage at 0x7fdb67524e20>
```

Figure 2. An example of a randomly generated RGB of the default environment

Here, Label 0 represents the room node, Label 1 represents the plane node, and Label 2 represents the object node. To make the above problem more challenging to solve, I also added configurations for dummies in the environments. The number of desired dummies (min/max) and the number of desired dummy types are set as hyperparameters.

To study the effect of high scene complexity, we simulate a heavily populated scene by adding a random number (up to 75) of triangles as distractors. This distractor objects have two detrimental effects to the CNN and the GNN model. First, the distractor objects occlude other relevant objects in the scene, and the detection model is harder to identify locations. Also, as we add more distractors, the graph size becomes greater, and it is harder for the GNN model to identify objects of interest in the scene graph. When we add these dummies, we pass in the minimum and maximum number of dummies, and the environment generates a uniformly random number of dummies within the range. By doing so, we also aim to see if the graph model can learn from dynamically sized graph size.

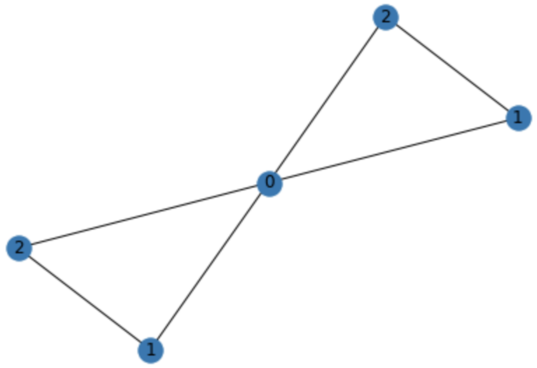
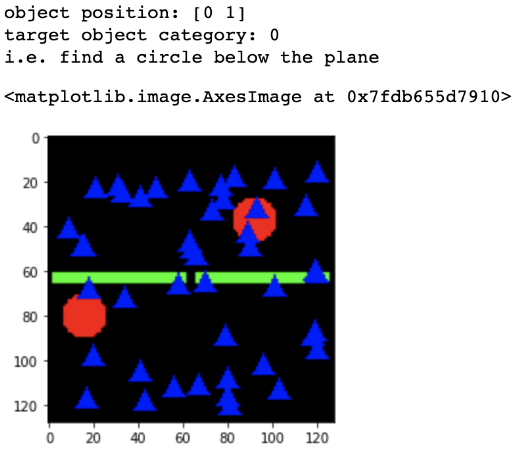


Figure 3. An example of a scene graph of the default environment



```
object position: [0 1]
target object category: 0
i.e. find a circle below the plane

<matplotlib.image.AxesImage at 0x7fdb655d7910>
```

Figure 4. An example of a randomly generated RGB of the the environment with dummies

This 2D environment is an abstracted version of relational choice tasks in the real-world simulation. Below shows the sample task in real-world-like simulation environments, iGibson. In the task below, the agent is given two objects with different relations to the secondary object. One object is located inside the cabinet, whereas the other object is located on top of the cabinet. The agent is given a target relation (”inside” or ”on top”), and it has to navigate to choose the correct object.

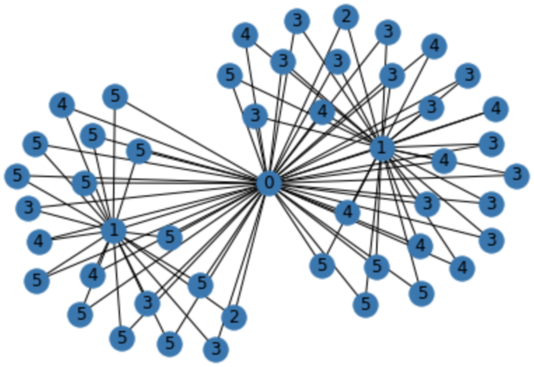


Figure 5. An example of a scene graph of the environment with dummies

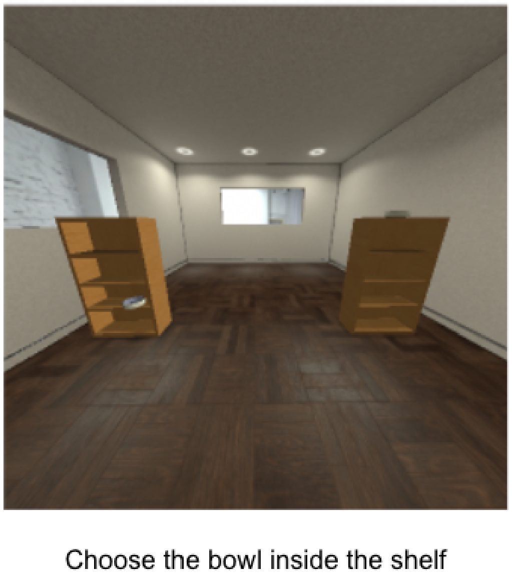


Figure 6. A sample task in iGibson simulation environment.

Model (No Distractors)	SR↑
RGB	0.992±0.001
RGB + SG	0.997±0.003
RGB + SG + TD ATTN	0.991±0.009

Table 1. Success Rate (SR) for Task without Distractors

4. Experiment Result

From Figures and Tables, we observed that when there are no distractors, all our models quickly learn to output the correct action that matches the goal description. However, when there are a large number of distractors, our model without attention **SG (NO ATTN)** has a significant drop in performance. The task-driven attention **SG (TD ATTN)** can salvage most of the performance loss and achieve near-perfect success.

Model (Distractors)	SR↑
RGB	0.458±0.004
RGB + SG	0.659±0.002
RGB + SG + TD ATTN	**0.993±0.007**

Table 2. Success Rate (SR) for Task with Distractors

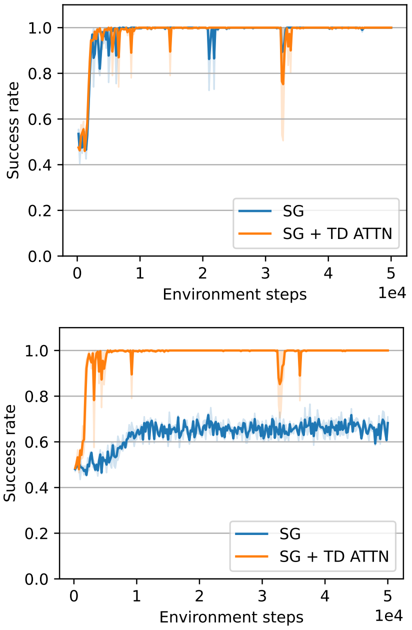


Figure 7. Success Rate (SR) versus environment steps over training for the 2D relational choice tasks. These plots depict the task with and without distractors, respectively. We observed that without attention, the performance of our model suffers significantly when there are many distractors. We observed that attention model performs much better than **RGB only**, suggesting that scene graphs are effective representation for tasks that require relational reasoning.

Another interesting observation was the evolution of learned attention weights in the learned attention model, as shown below. In the below example, the goal is to choose the right side as defined by the location of the circle with respect of the plane (squares in this figure). Highest attention is indicated with dark red and lowest attention is dark blue, with lighter red and blue indicating middle-high and middle-low attention. The attention shifts gradually towards task-relevant objects (circles and rectangles) and away from

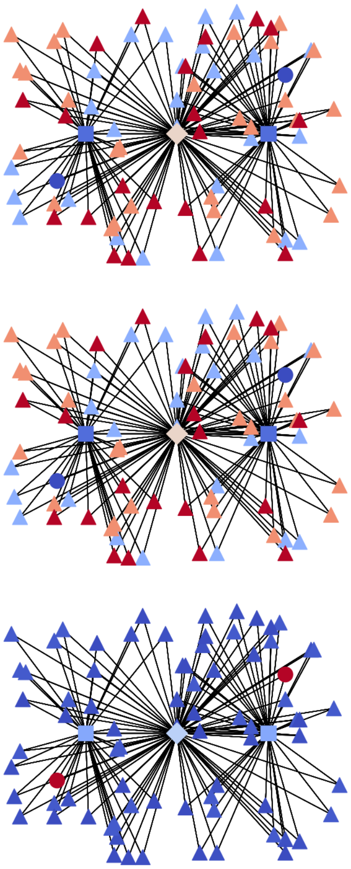


Figure 8. Evolution of learned attention over training in the experiment with distractors.

distractors (triangles). In comparison, our task-driven attention would focus on the two planes (squares) and the circles.

5. Conclusion

In this work, we design a novel environment for 2D relational choice task, and show the benefits of the scene graph as a representation for relational choice tasks that require reasoning about object relations. We show that leveraging graph represention is benefical compared to solely relying on RGB observations in relational choice task. Moreover, in large, populated scenes, having a task-driven attention mechanism is essential in aggregating task-relevant information and achieving a high success rate.

Ideally, the model might learn to attend to not only target objects in the goal description, but also the semantically related ones. We experimented with a scene graph model with learned attention in the above task, and achieved comparable, near-perfect results with task-driven attention. When we visualize the learned attention, we can see that the attention weights gradually shift towards task-relevant objects and away from distractors. However, the convergence was faster with task-driven, fixed attention. We conclude that

In future works, one can leverage the scene graph and attention approaches in real-world robotic tasks. This can be experiment with embodied AIs with real-world scenarios and tasks. Alternatively, more tasks can be designed in simulated 3D environment to test the proposed model's performance. Another extension of this work would be graph attention in robotic tasks with graph inputs other than scene graphs. Such example would be navigation of autonomous vehicles with graphical representation of roads and objects in the environment.

References

[1] Mary J Bravo and Ken Nakayama. The role of attention in different visual-search tasks. *Perception & Psychophysics*, 51(5):465–472, 1992. 2

[2] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference*, 2020. 2

[3] Bar Mayo, Tamir Hazan, and Ayellet Tal. Visual navigation with spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16898–16907, 2021. 2

[4] Zachary Seymour, Kowshik Thopalli, Niluthpol Mithun, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Maast: Map attention with semantic transformers for efficient visual navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 13223–13230. IEEE, 2021. 2

[5] George Sperling and Melvin J Melchner. The attention operating characteristic: Examples from visual search. *Science*, 202(4365):315–318, 1978. 2

[6] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2

[7] Jeremy M Wolfe and Todd S Horowitz. Five factors that guide attention in visual search. *Nature Human Behaviour*, 1(3):1–8, 2017. 2

[8] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In *Advances in Neural Information Processing Systems*, 2019. 2