CLIPGraphs: Multimodal Graph Networks to Infer Object-Room Affinities

Ayush Agrawal*[1], Raghav Arora*[1], Ahana Datta[1], Snehasis Banerjee[1,2], Brojeshwar Bhowmick[2], Krishna Murthy Jatavallabhula[3], Mohan Sridharan[4], Madhava Krishna[1]

[1]Robotics Research Center, IIIT Hyderabad, India
[2]TCS Research, Tata Consultancy Services, India
[3]CSAIL, Massachusetts Institute of Technology, USA
[4]Intelligent Robotics Lab, University of Birmingham, UK

***Abstract*—This paper introduces a novel method for determining the best room to place an object in, for embodied scene rearrangement. While state-of-the-art approaches rely on large language models (LLMs) or reinforcement learned (RL) policies for this task, our approach, CLIPGraphs, efficiently combines commonsense domain knowledge, data-driven methods, and recent advances in multimodal learning. Specifically, it (a) encodes a knowledge graph of prior human preferences about the room location of different objects in home environments, (b) incorporates vision-language features to support multimodal queries based on images or text, and (c) uses a graph network to learn object-room affinities based on embeddings of the prior knowledge and the vision-language features. We demonstrate that our approach provides better estimates of the most appropriate location of objects from a benchmark set of object categories in comparison with state-of-the-art baselines[1].**

***Index Terms*—Commonsense knowledge, graph convolutional network, knowledge graph, large language models, scene rearrangement.**

I. Introduction

Imagine a robot being tasked with tidying up an unfamiliar house. This task is a variant of the *scene rearrangement* challenge for embodied AI [1]. To perform this task, the robot must first determine what tidying up means in this specific house, which requires constructing a representation of the current state of the house and inferring a possible goal state (i.e., a configuration in which the house is deemed *tidy*). Any errors in this step can influence downstream planning and control, resulting in irrecoverable failure. Computing the most appropriate room location for specific object categories is thus critical to the successful completion of such tasks.

Human-inhabited environments such as homes and offices are designed to be functional and aesthetically pleasing. A key characteristic of such environments is the semantic organization, i.e., objects are placed in locations based on their purpose. This enables humans to adapt efficiently to new environments designed to serve the same purpose. For example, when a person enters a new home and wants to find sugar to make a cup of coffee, they instinctively look in the kitchen or pantry. We leverage this semantic organization to enable robots to predict the likely locations of any given object. Specifically, we leverage recent developments in multimodal (vision-language) representation learning to propose a flexible approach for learning *object-room affinities*, i.e., the relative likelihood of any given object belonging to a particular room in a house, based on image and text input.

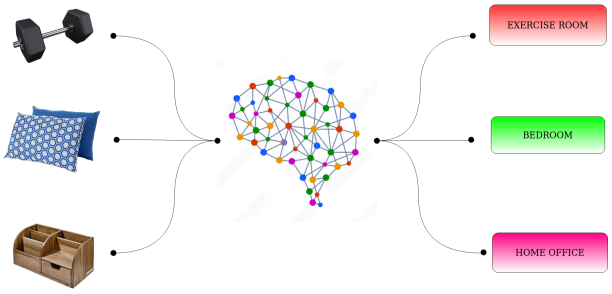

Fig. 1: Our method leverages semantic organization (e.g., "dumbbells are usually in the exercise room") to better compute the most suitable location for any given object.

State-of-the-art methods have used Large Language Models (LLMs) as *commonsense* reasoning machinery for this *tidy up* task [2]. These methods are limited to textual descriptors, which can be challenging to ground to a specific scene. Moreover, they use ground truth object labels for generating object-room affinities, which limits their operation outside of the training data distribution. Others have used reinforcement learning (RL) to compute policies for related tasks such as visual semantic navigation [3]–[6], and Multi-Object Navigation [7]–[9], but do not fully leverage knowledge from different sources in the learning process.

Our framework, *CLIPGraphs*, seeks to leverage the complementary strengths of commonsense knowledge, data-driven methods, and multimodal embeddings to estimate object-room affinities accurately. It does so by incorporating:

1) A *knowledge graph* that encodes human preferences of the room location of objects in home environments;
2) Joint embeddings of image and text features [10] to

*Denotes equal contribution
[1]Supplementary material and code: https://clipgraphs.github.io

support multimodal learning and queries in the form of images or text; and

3) A graph network that learns object-room affinities over a dataset of common household objects based on latent embeddings of the knowledge graph that includes the image and text feature embeddings.

The novelty is in the combination of these components to achieve the desired objective. We evaluate our framework's ability to correctly estimate the best room location for any given object, the key step in scene rearrangement. We do so using a dataset of 8000 image-text pairs that we created by extracting images from the Web for 268 benchmark object categories [2]. We show experimentally that our framework substantially improves performance compared with state of the art baselines comprising LLMs and language embeddings encoding commonsense knowledge of the location of objects.

II. Related Work

We motivate our novel framework by reviewing the limitations of related work.

Embodied AI: To train embodied agents to perform human-like activities, many common tasks have being explored recently like goal navigation [11]–[14], object navigation [3], [5], [7], [15]–[17], scene exploration [18], [19], embodied QA [20]–[22], and rearrangement [1], [23], [24]. ALFRED [25], TEACh [26], and [27] study the ability of agents to perform actions based on natural language instructions, and [28]–[30] use knowledge graphs for visual classification and detection. While these works include explicit specification of the goal state by a human agent, recent works have started the inclusion of reasoning with commonsense knowledge to enable agents to perform these tasks intelligently.

Commonsense Reasoning In the context of rearrangement, Housekeep [2], and TIDEE [31] work on tidying a house using commonsense reasoning based on the training of Large Language Models (LLMs); and CSR [32] generates reasoning from a scene graph to detect objects and changes in room states. Other works like JARVIS [33], DANLI [34], and LLM-Planner [35] show the effectiveness of prompting LLMs for language understanding and sub-goal planning using natural language instructions. [36] evaluates the performance of different language models and studies their limitations concerning commonsense in the physical world.

CLIP for Embodied AI CLIP (Contrastive Language-Image Pre-training) [37] uses large-scale text-image pairs for training image and text encoders simultaneously and has shown remarkable performance for object recognition. The effectiveness of CLIP image and text embeddings for Embodied AI tasks has been evidenced by recent studies [23], [38]–[40] over traditional ResNet-based architectures [41]. [42] demonstrated the use of CLIP to match objects in a cross-instance setting with visual features as a measure of similarity to complete tabletop object rearrangement tasks. A recent work, ZSON [4], proposes a zero-shot object navigation agent that uses CLIP embeddings to localize objects in the environment and navigate towards them without any additional training. The agent leverages the semantic similarity between the object category name and the visual features of the object to guide its exploration. Similarly, CLIP was used by [43], [44] for zero-shot vision and language navigation by using natural language expressions for descriptions of target objects. Recent works [6], [45]–[48] use pixel-level CLIP features for robotic navigation using language commands. [49], [50] have demonstrated the use of CLIP visual and language embeddings for learning robotic scenes, and [51], [52] use CLIP for generating 3D scene memories from 2D images and natural language.

III. Problem Formulation and Framework

To perform tidying up or other scene rearrangement tasks, a robot needs the key ability to accurately compute the appropriate location for any given object. To explore this ability, we created the *Images for Room-Object Nexus through Annotations* (IRONA) dataset of 30 RGB images from the Web for each of the 268 categories of household objects used by Housekeep [2][2]. For any such image, the robot had to compute the likelihood that the object in the image belongs to each of 17 room categories.

Our framework, called CLIPGraphs, trains a Graph Convolutional Network (GCN) [53] to compute embeddings that are used to estimate these object-room affinities. Figure 2 shows the training pipeline. It uses a knowledge graph to encode existing information of human preferences (of room location of objects) for the object categories [2], and incorporates a modified contrastive loss function to compute better latent embeddings of the image and language encoder features provided by CLIP [37] for the nodes of the knowledge graph. The resultant node embeddings model the information about the room location of various objects in the latent space. During inference, the CLIP features generated for any (test) RGB images are processed by the GCN, with the cosine similarity between the embeddings of the rooms and the image providing the desired estimate of object-room affinities. We describe individual components of our framework below.

A. Knowledge Graph

Our framework's first step uses the human-annotated preferences included in the Housekeep data [2]. For every object-room pair, 10 human annotators ranked the receptacles in that room based on the likelihood of the object being placed there correctly or incorrectly. For each object-room-receptacle tuple, there are thus 10 opinions that could be positive, negative, or zero. We filter the dataset to ensure good agreement amongst annotators. We calculate the positive (negative) soft scores as the mean of the positive (negative) reciprocal preference of all the receptacles for a given object-room pair. To establish ground truth object-room mappings, we use the object-room-receptacle scores, i.e., we select the room with the highest positive-scored receptacle. Every other

[2]Supplementary material at: https://clipgraphs.github.io

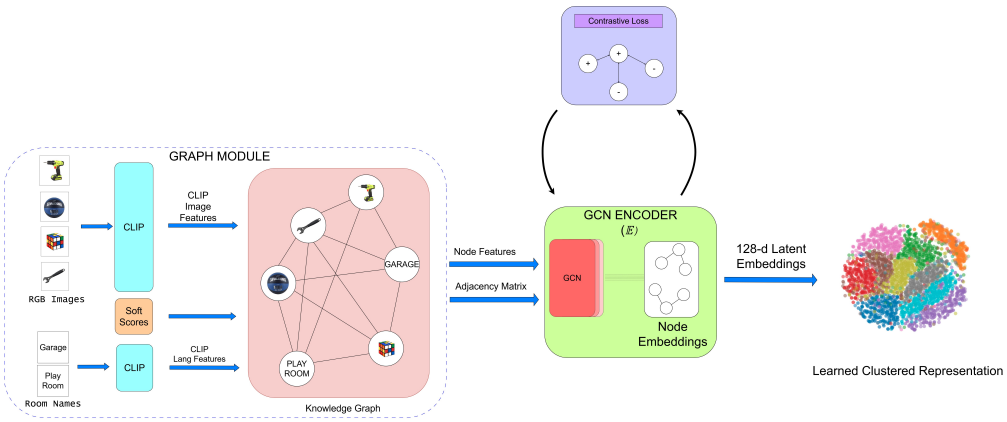

Fig. 2: *CLIPGraphs* constructs a graph module (bottom-left) using CLIP encoders and passes that to a GCN Encoder($\mathbb{E}$) module. The encoder is trained using contrastive loss to create better node embeddings that bring similar embeddings closer. Visualization of final layer activations confirms the formation of well-defined node clusters.

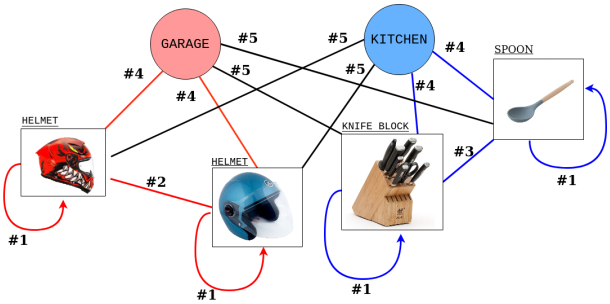

Fig. 3: An illustration of the five types of edges in our knowledge graph. The colored edges denote positive edge weights whereas black ones denote negative weights. The number on the edge denotes the type of edge.

room in the domain is assigned the mean negative soft score of receptacles in that room[2].

To use the available annotated information to populate a knowledge graph, we partitioned the IRONA web-scraped dataset into training, validation, and test sets in a ratio of 15:5:10 images per object category. The knowledge graph is instantiated with each image of the training set as a node, along with room names, i.e., there are 268*15 + 17 = 4037 nodes. We then considered five types of edges connecting nodes (see Figure 3): (1) image self edge (edge weight=1); (2) edge between images of same object (edge weight=1); (3) edge between two objects in the same ground truth room; (4) edge between object and its correct room node; and (5) edge between object and its incorrect room nodes. Next, we assigned weights for each type of edge. Weights for edges of type 4 and 5 were based on the object-room soft scores. Edges of type 3 were given a randomly chosen weight between 0.5 to 0.7, and edges of type 1 and 2 were assigned a weight of 1.

Once the basic knowledge graph is created, we initialize the graph's nodes using the pretrained CLIP model's high-dimensional embeddings. Specifically, each object node is initialized with the corresponding CLIP image encoder embedding, and each room node is initialized with the corresponding CLIP language encoder embedding. This is because we want to capture the appearance of the objects and the known association between objects and rooms (based on the large dataset used to train CLIP embeddings). In particular, we considered three pretrained architectures of CLIP in our experiments: Vision Transformer (ViT), ResNet-50, and ConvNeXt. ViT-H/14 [54] is trained on LAION-2B, which is a 2.3 billion subset of the LAION-5B [55] dataset with English captions. ResNet-50 [56] uses OpenAI's pre-trained weights [37], and ConvNeXt base [57] is pre-trained on LAION-400m [58], which contains 400 million image-text pairs[3]. For a discussion about how we experimentally chose the embedding for different nodes, please refer to our supplementary material. Once we have associated CLIP embeddings with our knowledge graph's nodes, we move to the next steps of our training pipeline.

B. GCN Training

The next step of training feeds these node embeddings, each of 512 or 1024 dimensions based on the CLIP architecture chosen, and the adjacency matrix (of knowledge graph structure) to a Graph Convolutional Network (GCN) [53] to learn better latent space embeddings of our knowledge graph. GCNs are able to capture non-linear relationships between nodes, and learn from both local and global structures in a graph. As a result, nodes that are more similar are mapped to points that are closer in the latent embeddings space, whereas nodes that are dissimilar are mapped to points further away in the latent space. For example, the output 128-dimensional GCN (object) embedding for a microwave will have a higher cosine similarity with the output 128-dimensional GCN (language) embedding for the kitchen.

An important design decision during training is the choice of the loss function. Prior work has devoted much attention to functions such as contrastive loss [60], triplet loss [61], and multi-class N-pair loss [62]. Recent work has demonstrated

[3]Implementation used existing code [59].

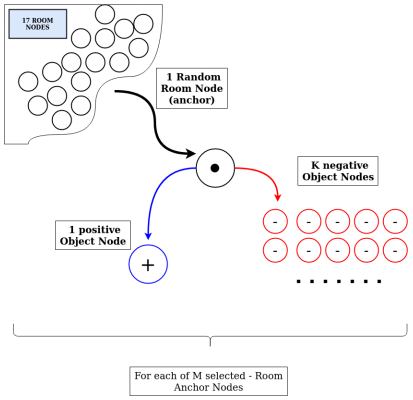

Fig. 4: Sampling method used in the loss function; shown for K = 10 and M = 1; we average the loss over M batches.

the benefits of using the loss function introduced in the CLIP-Fields method [52]. We modify this loss function to further leverage the knowledge graph created using the IRONA dataset and human preference annotations.

Loss Function. We train our GCN using a contrastive loss function similar to that described in the CLIP-Fields method [52] with the objective of clustering similar embeddings closer in the latent space and mapping dissimilar embeddings to points that are further away in the latent space. We adapt the basic loss function to our problem formulation and use the additional information of edge weights.

$$L = -e^{-weight_{+\bullet}} \log \left(\frac{e^{(sim_{+\bullet}/T)}}{\sum_{i=1}^{K} e^{(sim_{-\bullet,i}/T)}} \right) \quad (1)$$

where $weight_{+.}$ is the edge weight between the positive node and the anchor node, $sim_{+\bullet}$ is the cosine similarity between the anchor and a positive node embedding, and $sim_{-\bullet,i}$ is the cosine similarity between anchor node embedding and i^{th} negative node embedding. T is a temperature term that is tuned over a validation set. We randomly select one of the 17 rooms as our anchor node, then choose a positive node (for numerator in Equation 1) by picking an object within that room at random, and finally sample k negative nodes for the denominator of the loss function from objects located outside the room; Figure 4 illustrates this process, which is repeated for a batch of samples and the mean loss is calculated. This formulation of the loss function minimizes the distance between the anchor node and the positive node while maximizing the distance with each of the negative nodes, leading to distinct clusters in the graph embeddings. As stated before, the training pipeline is outlined in Figure 2.

C. Testing

Once the GCN has been trained, the pipeline used for testing (i.e., inference) is shown in Figure 5. Similar to the process of training, we compute the CLIP image encoder embedding for the test image, and the CLIP language encoder embedding for the possible rooms. These embeddings are passed to the GCN with only self-edges (in the absence of a knowledge graph) to obtain the output (latent space) embedding for the test image and the possible rooms. Next, similarity scores are calculated between each image node $\vec{x}$ and each of the room(s) $\vec{y}$ using the cosine similarity function: $\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \cdot ||\vec{y}||}$. We then average the similarity scores over different images of each object category to get the affinity score between that object category and each of the candidate rooms.

IV. Experimental Setup and Results

This section describes the experiments we conducted and discusses the corresponding results.

A. Experimental Setup

Object-room affinities have predominantly been determined by language-based embeddings or human input in prior work. Since our work combines prior knowledge and multimodal (vision, language) inputs, our chosen baselines were off-shelf language encoders and the GPT-3 LLM. We experimentally evaluated the following hypothesis:

- **H1:** CLIP language embeddings result in better performance than other language encoder embeddings;
- **H2:** Multimodal CLIP embeddings, by themselves, do not perform better than language-based embeddings;
- **H3:** Our framework leads to better performance than (i) the underlying CLIP embedding, (ii) just the language-based encodings, and (iii) the GPT-3 LLM;
- **H4:** Our framework provides robustness to previously unseen noisy backgrounds.

We evaluated **H1-H3** quantitatively and evaluated **H4** qualitatively. The performance task was to compute estimates of object-room affinities for all 268 object categories and 17 rooms in the test split of the IRONA dataset. We considered two performance measures:

1) **mAP:** The mean average precision (mAP) is the average of precision scores at different recall values for each instance of an object category, and the mean over all the object categories.
2) **Top k Hit Ratio:** The average fraction of object categories for which the ground truth correct room was among the Top k estimates from our framework.

All claims are statistically significant unless stated otherwise.

B. Quantitative Results

To evaluate **H1**, we first compared two existing language encoder embeddings (RoBerta [63], GloVE [64]) with just the CLIP-based language embeddings with each of the three CLIP architectures. As shown in Table I, the CLIP-based language embeddings (particularly the ViT architecture) resulted in better performance, supporting **H1**.

Next, we compared the performance of the multimodal (vision, language) CLIP embeddings for each of the three CLIP architectures. As shown in Table II, performance is comparable but slightly worse than that in Table I. These results support **H2** and motivate the use of GCNs.

Next, we computed the performance of our architecture, i.e., with GCNs trained using the contrastive loss function and the underlying multimodal CLIP embeddings, with the

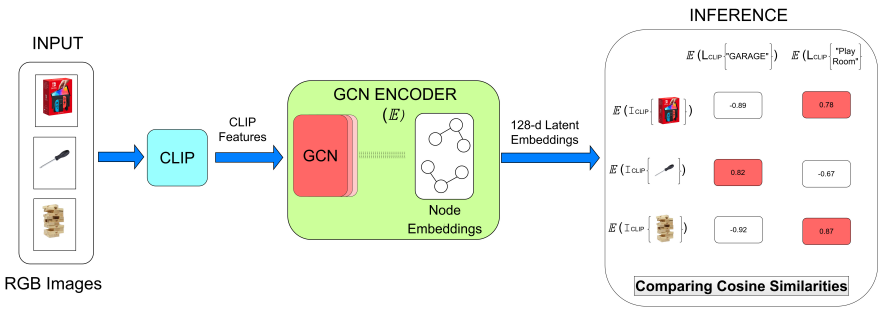

Fig. 5: Our inference pipeline processes input RGB images to generate CLIP image embeddings. These embeddings are processed by the GCN Encoder to produce latent image embeddings. Cosine similarity between these latent embeddings and previously learned room embeddings determines object-room affinities.

Lang Model	Test mAP ⇑	Hit-Ratio ⇑		
		Top-1	Top-3	Top-5
ConvNeXt	0.405	0.223	0.472	0.632
ViT	**0.456**	0.256	**0.576**	**0.710**
RN50	0.453	**0.275**	0.546	0.643
RoBerta	0.417	0.238	0.491	0.636
GloVE	0.148	0.123	0.208	0.278

TABLE I: CLIP-based language embeddings perform better than other popular language encoders; results support **H1**.

UnTuned-CLIP	Test mAP ⇑	Hit-Ratio ⇑		
		Top-1	Top-3	Top-5
ConvNeXt	0.41	0.24	0.46	0.62
ViT	**0.42**	**0.25**	**0.49**	**0.65**
RN50	0.39	0.19	0.45	0.67

TABLE II: Multimodal CLIP embeddings, by themselves, do not improve performance compared with just the CLIP-based language embeddings (see Table I). Results support **H2**.

corresponding results shown in Table III. The best performance was (once again) with the ViT version of the CLIP architecture. Also, performance was substantially better than with the multimodal CLIP embeddings (Table II) or CLIP's language encoder embeddings (Table I). For example, there is an $\approx 40\%$ increase in mAP score compared with not using the GCNs. These results partially support **H3**.

To further explore the benefits of a multimodal CLIP representation, we conducted experiments with our framework, but with GCN embeddings of only the language-based encoding of CLIP. The results reported in Table IV show the benefits of using the multimodal CLIP embeddings.

The next experiment compared our framework's performance with the GPT-3 LLM and a state of the art language encoder that provided the best performance among language-based encoders. The results summarized in Table V show that our framework provides substantially better performance by fully leveraging prior commonsense knowledge and multimodal CLIP embeddings. These results strongly support **H3**.

GCN-CLIP	Test mAP ⇑	Hit-Ratio ⇑		
		Top-1	Top-3	Top-5
ConvNeXt	0.73	0.62	0.81	0.88
ViT	**0.85**	**0.76**	**0.93**	**0.97**
RN50	0.66	0.53	0.75	0.81

TABLE III: CLIPGraphs use of GCN embeddings of multimodal CLIP features and commonsense knowledge results in substantially better performance compared with just the CLIP embeddings in Tables I and II. Results support **H3**.

GCN-CLIP[Lang]	Test mAP ⇑	Hit-Ratio ⇑		
		Top-1	Top-3	Top-5
ConvNeXt	0.64	0.53	0.69	0.76
ViT	**0.77**	**0.68**	**0.77**	**0.83**
RN50	0.59	0.46	0.63	0.74

TABLE IV: Using our GCN-based embedding with just the underlying language-based CLIP encoding results in better performance than in the absence of the GCN embedding, but performance is not as good as when GCNs are used with the multimodal CLIP embeddings (in Table III).

	Test mAP ⇑	Hit-Ratio ⇑		
		Top-1	Top-3	Top-5
Our[GCN-CLIP] [III]	**0.85**	**0.76**	**0.93**	**0.97**
GPT-3	0.66	0.52	0.76	0.81
Best Lang Encoder[I]	0.456	0.275	0.576	0.71

TABLE V: Our framework, with GCN and underlying multimodal CLIP embeddings, substantially improves performance compared with standalone GPT-3 LLM and language-based encoders; hence, the results strongly support **H3**.

C. *Qualitative Results*

Figure 6 shows the result of using our framework with images of previously seen objects but in noisy, previously unseen backgrounds. In each case, the object's room association was estimated correctly. Next, Figure 7 shows the success cases when our trained framework was used with objects from previously unseen object categories. Success (i.e., estimating the correct room association for the objects) can be attributed to leveraging commonsense knowledge extracted from similar images.

Figures 8 and 9 show some examples of our framework's limitations. In Figure 8, an input image of earpods (not present in the training set) was mapped to the *utility room* because it was similar in appearance to hair dryers that were

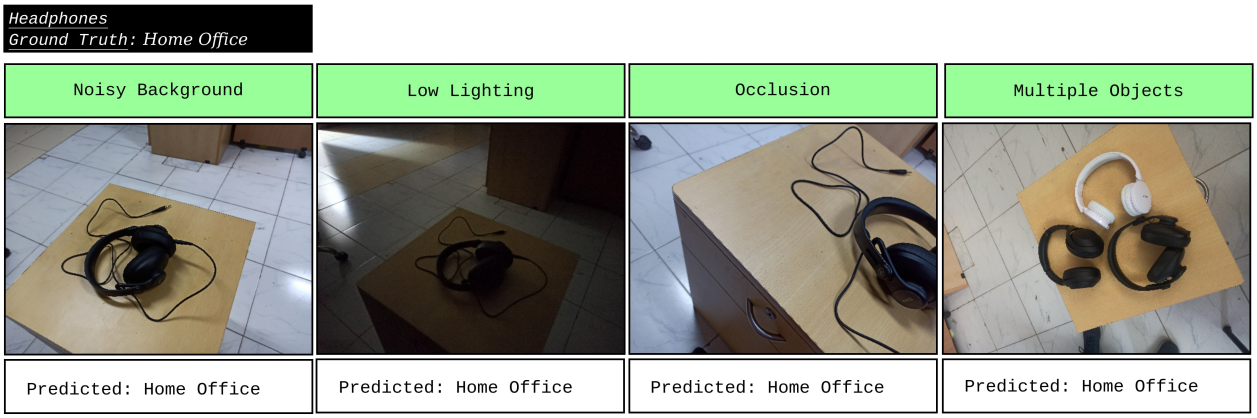

Fig. 6: Qualitative results for previously seen objects in new backgrounds; supports **supports H4**.

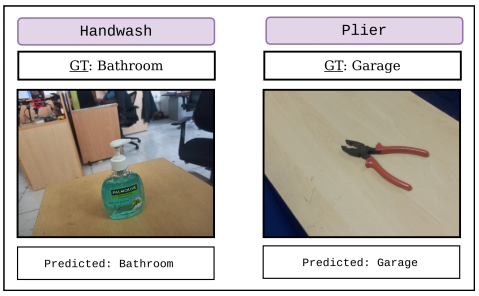

Fig. 7: Successful placement of previously unseen object categories (Handwash, plier) in the correct room by leveraging commonsense domain knowledge.

Fig. 8: Failure to determine correct room for object category *earpods* (not in our train set) because it was structurally similar to *hair dryer* category that was in our training set.

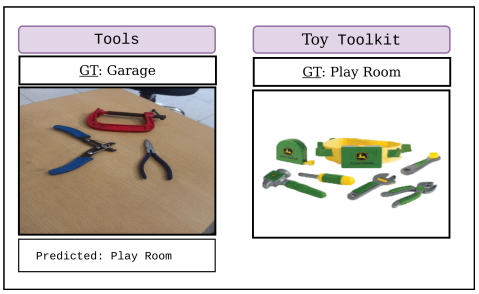

Fig. 9: Failure with composite object categories; *tools* was not a category in our training set, but they were incorrectly associated with the *play room* because they were structurally similar to the *toy toolkit* that was in the training set.

known to our framework. Figure 9 shows another failure case in which our framework estimated the room association for actual tools (which it has not seen before) as *playroom* because the training set contained an image of a toy tool kit in a playroom. However, each tool, when considered individually, is associated with the correct room location. These results support hypothesis **H4**.

V. Conclusion and Future Work

Accurately estimating object-room affinities is an important step in performing scene rearrangement tasks. We presented a framework called CLIPGraphs, which estimates these affinities by leveraging the complementary strengths of commonsense knowledge, data-driven methods, and multimodal (vision, language) embeddings. Specifically, our framework encodes prior human preferences in a knowledge graph and considers CLIP-based image and language embeddings of nodes in this graph. It then uses Graph Convolutional Network (GCN)-based embeddings of these CLIP embeddings to learn and estimate the object-room affinities. We experimentally evaluated our framework's performance in estimating object-room affinities using our IRONA dataset of 8040 images of 268 benchmark object categories. We experimentally demonstrated a substantial improvement in the ability to estimate object-room affinities compared with language encoder embeddings and the GPT-3 LLM. We also showed qualitatively that our framework provides robustness to previously unseen noisy backgrounds.

Our framework opens up directions for further research. For example, we plan to train our model with top-k correct rooms to generate object-room affinities that would be useful in downstream tasks such as multi-object navigation.We also plan to develop personalized or task-specific embeddings that allow our framework to calculate object-room affinities tailored to individual users, homes, or tasks. This will enable physical robots to assist humans in complex scene rearrangement tasks, and other embodied AI tasks characterized by semantic organization.

REFERENCES

[1] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, M. Savva, and H. Su, "Rearrangement: A challenge for embodied ai," 2020.

[2] Y. Kant, A. Ramachandran, S. Yenamandra, I. Gilitschenski, D. Batra, A. Szot, and H. Agrawal, "Housekeep: Tidying virtual households using commonsense reasoning," in *European Conference on Computer Vision*, 2022.

[3] D. S. Chaplot, D. Gandhi, A. K. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *ArXiv*, vol. abs/2007.00643, 2020.

[4] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *ArXiv*, vol. abs/2206.12403, 2022.

[5] N. Gireesh, D. A. S. Kiran, S. Banerjee, M. Sridharan, B. Bhowmick, and M. Krishna, "Object goal navigation using data regularized q-learning," *International Conference on Automation Science and Engineering*, pp. 1092–1097, 2022.

[6] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Clip on wheels: Zero-shot object navigation as object localization and exploration," *ArXiv*, vol. abs/2203.10421, 2022.

[7] N. Gireesh, A. Agrawal, A. Datta, S. Banerjee, M. Sridharan, B. Bhowmick, and M. Krishna, "Sequence-agnostic multi-object navigation," in *IEEE International Conference on Robotics and Automation*, 2023, (to be published).

[8] K. Ellis, D. Hadjivelichkov, V. Modugno, D. Stoyanov, and D. Kanoulas, "Navigation among movable obstacles via multi-object pushing into storage zones," *IEEE Access*, vol. 11, pp. 3174–3183, 2023.

[9] P. Marza, L. Matignon, O. Simonin, and C. Wolf, "Teaching agents how to map: Spatial reasoning for multi-object navigation," *International Conference on Intelligent Robots and Systems*, pp. 1725–1732, 2021.

[10] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," *ArXiv*, vol. abs/2212.07143, 2022.

[11] E. Wijmans, A. Kadian, A. S. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," in *International Conference on Learning Representations*, 2019.

[12] P. Anderson, A. X. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On evaluation of embodied navigation agents," *ArXiv*, vol. abs/1807.06757, 2018.

[13] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. H. Oh, "Topological semantic graph memory for image-goal navigation," in *Conference on Robot Learning*, 2022.

[14] O. Kwon and S. Oh, "Image-goal navigation algorithm using viewpoint estimation," *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pp. 689–692, 2021.

[15] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *ArXiv*, vol. abs/2006.13171, 2020.

[16] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6743–6752, 2018.

[17] W. Yang, X. Wang, A. Farhadi, A. K. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," *ArXiv*, vol. abs/1810.06543, 2018.

[18] D. S. Chaplot, D. Gandhi, S. Gupta, A. K. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *ArXiv*, vol. abs/2004.05155, 2020.

[19] D. S. Chaplot, H. Jiang, S. Gupta, and A. K. Gupta, "Semantic curiosity for active visual learning," in *European Conference on Computer Vision*, 2020.

[20] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2135–213 509, 2017.

[21] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4089–4098, 2017.

[22] C. Cangea, E. Belilovsky, P. Lio', and A. C. Courville, "Videonavqa: Bridging the gap between visual and embodied question answering," in *British Machine Vision Conference*, 2019.

[23] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, "Visual room rearrangement," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5918–5927, 2021.

[24] B. Trabucco, G. Sigurdsson, R. Piramuthu, G. S. Sukhatme, and R. Salakhutdinov, "A simple approach for visual rearrangement: 3d mapping and semantic search," 2022.

[25] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 737–10 746, 2019.

[26] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramithu, G. Tur, and D. Z. Hakkani-Tür, "Teach: Task-driven embodied agents that chat," in *AAAI Conference on Artificial Intelligence*, 2021.

[27] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2017.

[28] X. Chen, L.-J. Li, L. Fei-Fei, and A. K. Gupta, "Iterative visual reasoning beyond convolutions," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7239–7248, 2018.

[29] K. Marino, R. Salakhutdinov, and A. K. Gupta, "The more you know: Using knowledge graphs for image classification," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20–28, 2016.

[30] X. Wang, Y. Ye, and A. K. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6857–6866, 2018.

[31] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki, "Tidee: Tidying up novel rooms using visuo-semantic commonsense priors," in *European Conference on Computer Vision*, 2022.

[32] S. Y. Gadre, K. Ehsani, S. Song, and R. Mottaghi, "Continuous scene representations for embodied ai," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 829–14 839, 2022.

[33] K. Zheng, K.-Q. Zhou, J. Gu, Y. Fan, J. Wang, Z. xiao Li, X. He, and X. E. Wang, "Jarvis: A neuro-symbolic commonsense reasoning framework for conversational embodied agents," *ArXiv*, vol. abs/2208.13266, 2022.

[34] Y. Zhang, J. Yang, J. Pan, S. Storks, N. Devraj, Z. Ma, K. Yu, Y. Bao, and J. Y. Chai, "Danli: Deliberative agent for following natural language instructions," in *Conference on Empirical Methods in Natural Language Processing*, 2022.

[35] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," 2023.

[36] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi, "Piqa: Reasoning about physical commonsense in natural language," *ArXiv*, vol. abs/1911.11641, 2019.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021.

[38] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," 2022.

[39] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, and K. Keutzer, "How much can clip benefit vision-and-language tasks?" *ArXiv*, vol. abs/2107.06383, 2021.

[40] K. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. Tenenbaum, C. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "Conceptfusion: Open-set multimodal 3d mapping," *arXiv*, 2023.

[41] E. Wijmans, I. Essa, and D. Batra, "How to train pointgoal navigation agents on a (sample and compute) budget," 2020.

[42] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner, "Semantically grounded object matching for robust robotic scene rearrangement," 2021.

[43] V. S. Dorbala, G. A. Sigurdsson, R. Piramuthu, J. Thomason, and G. S. Sukhatme, "Clip-nav: Using clip for zero-shot vision-and-language navigation," *ArXiv*, vol. abs/2211.16649, 2022.

[44] V. S. Dorbala, J. F. Mullen, and D. Manocha, "Can an embodied agent find your "cat-shaped mug"? llm-based zero-shot object navigation," *ArXiv*, vol. abs/2303.03480, 2023.

[45] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," *ArXiv*, vol. abs/2210.05714, 2022.

[46] D. Shah, B. Osinski, B. Ichter, and S. Levine, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*, 2022.

[47] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," *ArXiv*, vol. abs/2209.09874, 2022.

[48] K. Jain, V. Chhangani, A. Tiwari, K. M. Krishna, and V. Gandhi, "Ground then navigate: Language-guided navigation in dynamic scenes," 2022.

[49] J. Thomason, M. Shridhar, Y. Bisk, C. Paxton, and L. Zettlemoyer, "Language grounding with 3d objects," *ArXiv*, vol. abs/2107.12514, 2021.

[50] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," *ArXiv*, vol. abs/2109.12098, 2021.

[51] H. Ha and S. Song, "Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models," in *Conference on Robot Learning*, 2022.

[52] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. D. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," *ArXiv*, vol. abs/2210.05663, 2022.

[53] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[54] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[55] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "Laion-5b: An open large-scale dataset for training next generation image-text models," 2022.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[57] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *CVPR*, 2022.

[58] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs," 2021.

[59] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, "Openclip," July 2021, if you use this software, please cite it as below. [Online]. Available: https://doi.org/10.5281/zenodo.5143773

[60] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine, "Time-contrastive networks: Self-supervised learning from video," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, 2017.

[61] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.

[62] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[63] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[64] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing*, 2014.