CLIPGraphs: Multimodal graph networks to infer object-room affinities for scene rearrangement

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

[1]Robotics Research Center, IIIT Hyderabad, India
[2]TCS Research, Tata Consultancy Services, India
[3]Intelligent Robotics Lab, University of Birmingham, UK
[4]MIT

***Abstract*—We propose a multimodal learning approach to infer human preferences of a *tidy room*. This is crucial for robotic scene rearrangement problems, where much of the prior work has centered around assuming rearrangement goals to be concretely specified. Our method, dubbed CLIPGraphs, encodes features from off-the-shelf vision-language models (specifically CLIP) using a graph neural network over the set of rooms and objects present in a scene. CLIPGraphs are multimodal – they may be queries using both vision and language queries, and map each query to a room of the house the object should belong to, in line with human preferences.This approach provides a powerful tool for inferring object-room affinities and enabling more effective scene rearrangement.**

***Index Terms*—Visual CommonSense,**

I. Appendix

[Thoughts] We also plan to discuss a bit about the High-Level category mAP of various baselines. So from **??**, we get 3 plots of High Level mAP's. Before showing that we also plan to tell them what are the various high level categories [**?**] We then show the confusion matrix of the best GCN + CLIP finetuned model

A. Loss Function

1) Structure: The spine of our training lies in the efficacy of our loss function. There have been various experiments with what works well for Representation Learning [cite some papers]. However with the recent success of CLIP-Field's Formulation of Contrastive Loss [**?**] we choose this as the skeleton for our loss function ablations. Here for each batch they take 1 anchor, 1 positive node wrt to that anchor, and k negative nodes wrt to that anchor. We run ablations on $batch\ size$ as well as k and tune the $temp$ hyperparameter on the validation set.

2) Sampling: The benefits of k negatives over the traditional triplet loss [**?**] are intuitive as well as have been sufficiently well proven in the literature. [cite plz]. Thus we directly begin with our ablation results. For our sampling and batch size ablations, we keep the $temp$ as 0.01. Once we got the best batch size and sampling parameters, we tuned our $temp$ hyperparameter to maximize mAP over the validation set. We get it as 0.01

*Denotes equal contribution

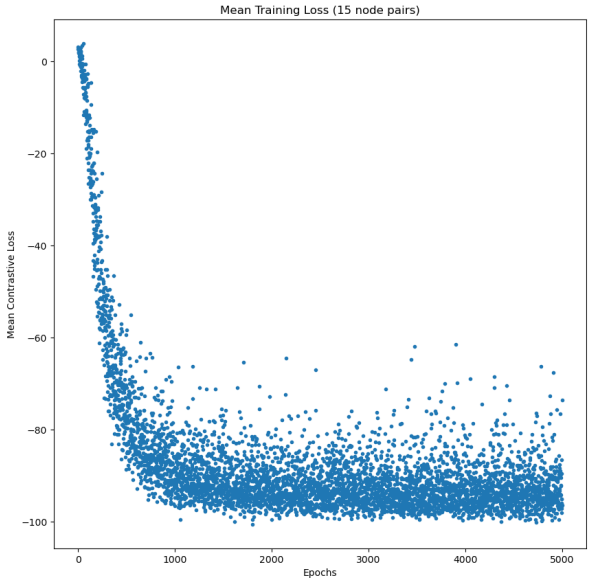


Fig. 1: Convergence of the Loss Curve

add the plots for Vit vs RN50 vs COnvNext for batch size , k, sampling technique, temp =0.01

3) Comments:

- batch size, circumventing the room bias

B. Loss Function Ablations

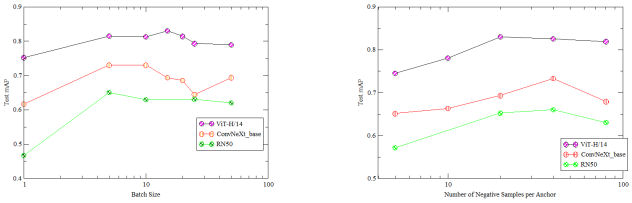
- L2 : single +ve & k negative
- k negatives , batch size, edge weight vs no edge weight, norm edge weight vs unnorm edge weight, usage of edge weights

C. Hyperparameter Tuning Plots

We tried initializing the room node features as 1-hot features, random and corresponding to the CLIP Language Encoders but everytime our training converged to the same point and gave us the same Test mAP. Thus we can safely conclude that the room node features don't affect the performance

#	Hyperparameter	Value
1	Node Feature Size	1024(ViT) / 512 (Others)
2	Output Node Embedding	128
3	GCN Layers	
4	Learning Rate	
5	Learning Rate Schedule	
6	Temperature	
7	Batch Size [Loss]	
8	Negatives Per Batch	
9	Sampling Method	

TABLE I: Hyperparameter choices for fine-tuning CLIP Visual Encoder Features on our GCN



(a) Ablation on batch-sizes with test mAP.

(b) Ablation on the number of negative samples in the contrastive loss.

Fig. 2: Ablations. Needed? Change the scale of x-axis to linear?

D. More About Graph Creation

- Getting Object-Room GT Relationships on the basis of human annotations. A journey from human ranks to soft score. And further till OR Ground truths.

E. Are these figures needed?

Figures 2 and 1 , figure comparing training and validation mAPs?

Statistics About our Knowledge Graph	Value
#Nodes	4020 Object Images Nodes & 17 Room Nodes
#Edges	7,66,649
Self Loops	Yes
Train Images	268*15 Images
Test Images	268*10 Images
Val Images	268*5 Images
Types of edges	weighted undirected

TABLE II: Statistics for the Knowledge Graph created using the web scraped dataset

F. Qualitative Results

Fig. 3: Representative Image Of Our Web Scraped Dataset ; 1 image per object category

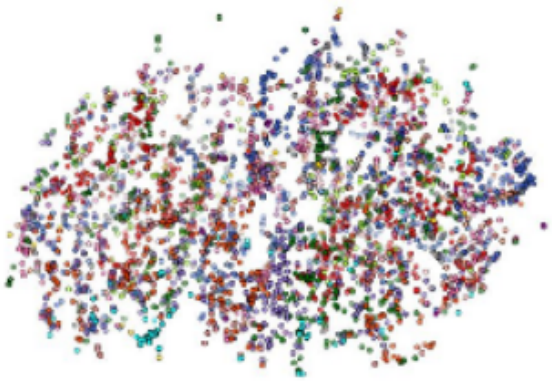


Fig. 4: Untrained TSNE

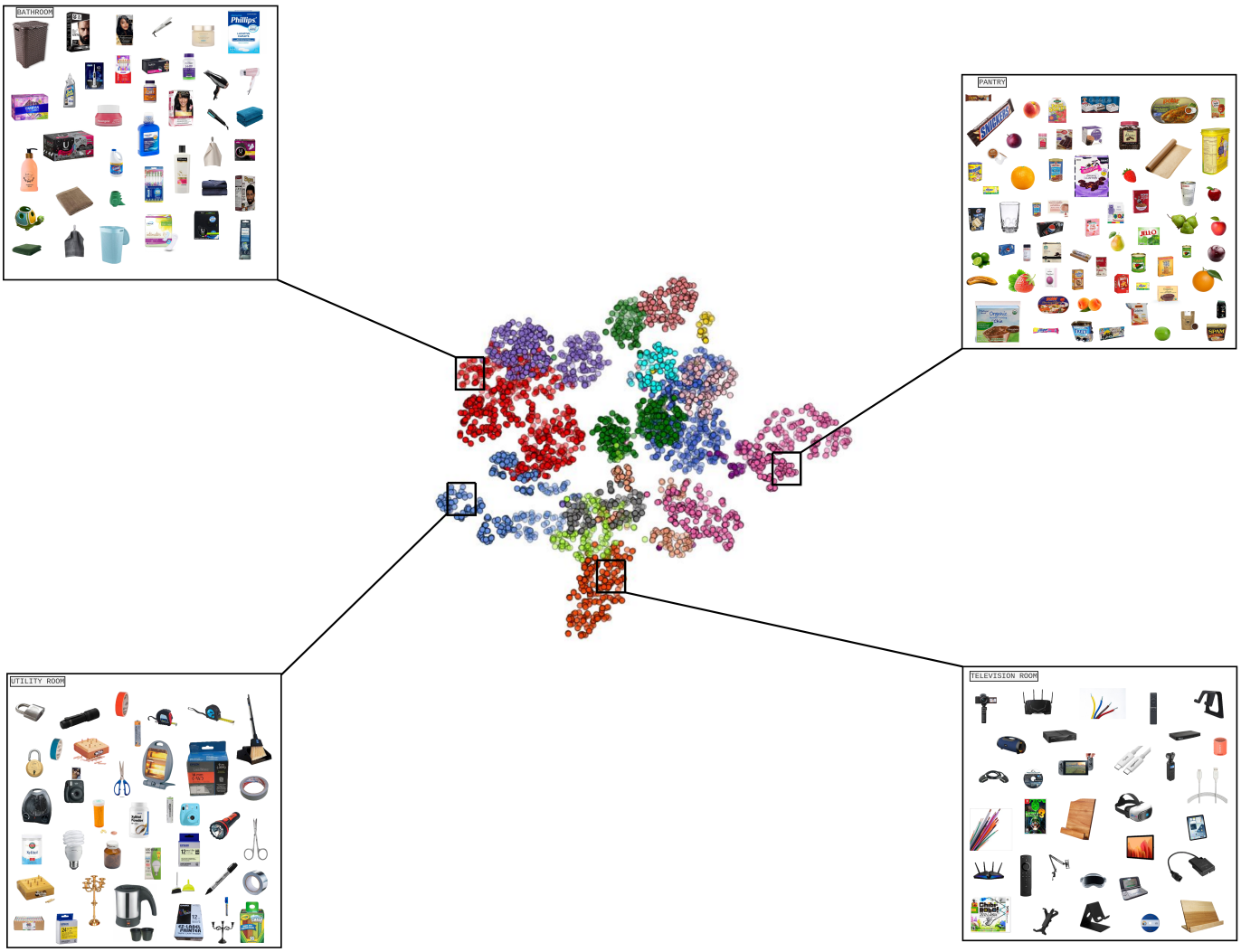


Fig. 5: t-SNE visualization of our embeddings on the test split of the Web Scraped Dataset**??**. The boxes are representative of the type of images getting clustered.

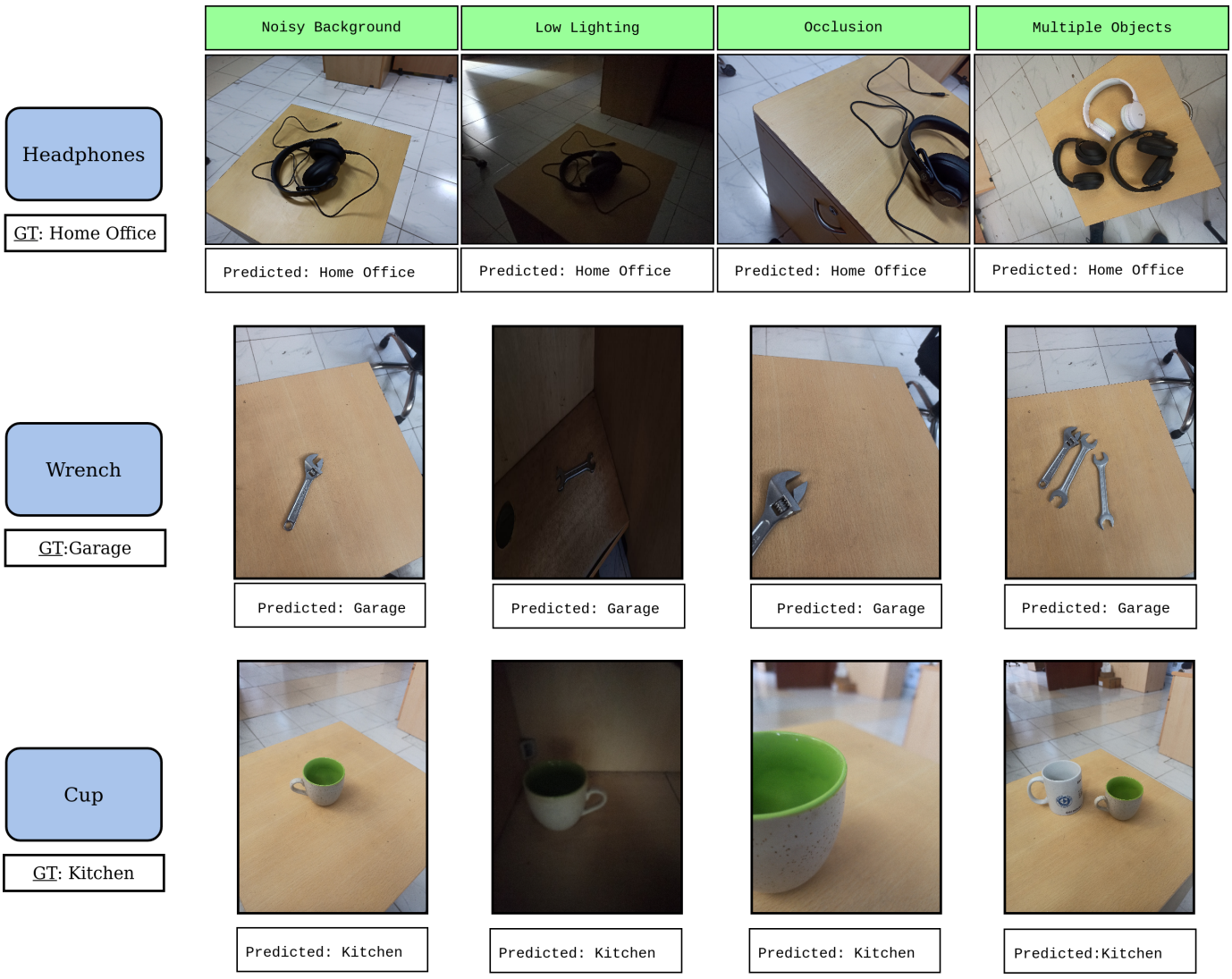


Fig. 6: Qualitative Results for Real World Images for Seen Objects

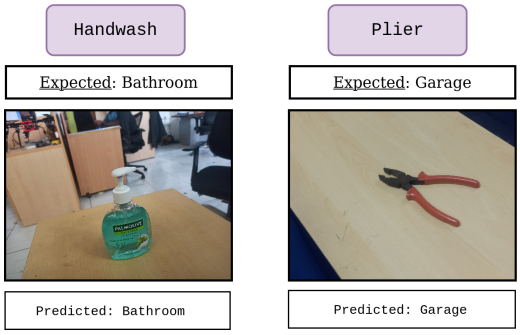


Fig. 7: Qualitative Results: Success Example for Unseen Object Category; Handwash,Plier[not in our train set] was predicted correctly

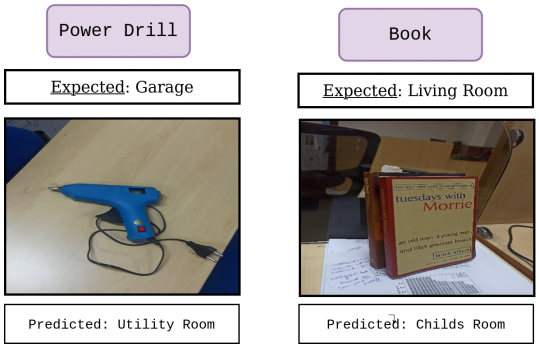


Fig. 8: Qualitative Results: Failure cases for Power Drill, Book [present in our train dataset]

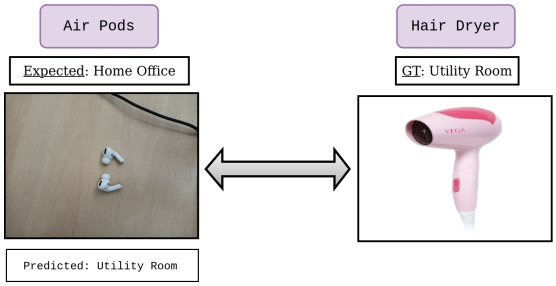


Fig. 9: Qualitative Results: Failure Case for scale ignorant model; airpods[not in our train set] are structurally similar to a hair dryer[in our train set] ; and thus predicts incorrectly

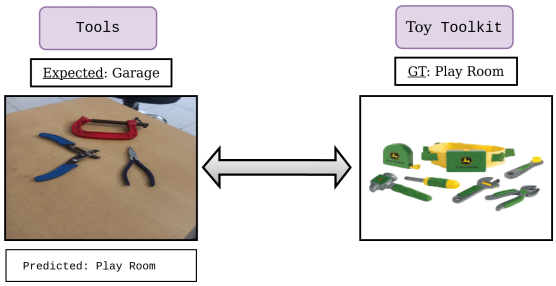


Fig. 10: Qualitative Results: Failure Case for weird object categories; tools[not in our train set] are structurally similar to toy toolkit[in our train set] ; and thus predicts incorrectly