Answering visual questions through representation learning

Sumit Agarwal [1] **Suraj Tripathi** [1] **Syeda Nahida Akter** [1] **Andrew Lyubovsky** [1] **Feng Xiang** [1]

1. Introduction

In this report, we capture the different unimodal experiments we performed using the GQA dataset (Hudson & Manning, 2019) which requires answering questions using multi-hop reasoning about images. This dataset in particular tries to balance different types of questions and synthetically generate questions using scene graphs (Damodaran et al., 2021) from the Visual Genome dataset (Krishna et al., 2017) which makes it difficult to answer them. We specifically perform experiments on unimodal representations of images (including CNN features of images and also object features) and questions. We analyze different kinds of representations and how they encode information. In particular, we found ResNet models (He et al., 2015) capture image embeddings better and contextualized word embeddings like RoBERTa (Liu et al., 2019) and BERT (Devlin et al., 2019) to better represent questions. In experiments with shallow models, we didn't find any model to show good results because of the fact that the network is shallow and hence no one modality representation generalised well about the task. In conclusion, we analyzed unimodal representations and found out that models like BERT for text, and RCNN/ResNet for images are able to encode semantic information present in the modality. Also, we present visualizations and failure analyses which motivates us to look for novel solutions to exploit the structure and to resolve biases present in the GQA dataset.

2. Experiments

2.1. Object Recognition

Object recognition (Zhao et al., 2018) is a general term that refers to a group of related computer vision tasks involving the identification of objects in digital pictures. We proposed an object recognition task to assess the effectiveness of various representations to encode object information. We believe this task will give us a better understanding of how objects are represented within images, and the interactions between these object representations. To train the model, we presented images as inputs with the label of an object that was present in that image. Next, to test what objects are stored in image representations, we asked the model to identify an object that was detected within an image. To perform well on the object recognition task, we investigated the capacity of different feature extractors to encode visual information. We looked at a shallow 2-layer CNN (Krizhevsky et al., 2012), ResNet34 (He et al., 2015), and VGGNet16 (Simonyan & Zisserman, 2015) feature extractors, followed by a 1 layer of feed-forward neural network. We observed significant improvement in performance when we included ResNet/VGGNet pretrained features instead of training a shallow CNN from scratch. Table 1 presents the accuracies obtained based on different feature embeddings that were used.

The process of labeling the images from the dataset is as follows:

Data Creation for Object Recognition:

- First, image-object pairs were extracted from the GQA(Hudson & Manning, 2019) corpus, where an image-objects pair is a pair between an image and a list of objects that is present within that image.
- From all unique objects, we choose only the top 10 objects in order to exclude low-frequency objects and to aid in the analysis of failures. We calculated top-10 objects based on the frequency in the corpus.
- For each image, if top-10 objects are present we select one of the top-10 objects at random and generate an image-object instance. We selected only one object per image for these experiments so that we could analyze whether a model is able to encode a particular object well given the full image.
- The top-10 object distributions in whole corpus (train+val) is as follows: ['leg' (6.9%), 'hair' (6.7%), 'build' (6.7%), 'hand' (7.5%), 'person' (6.6%), 'wall' (16.7%), 'tree' (12.4%), 'shirt' (11.6%), 'window' (11.2%), 'man' (13.6%)]

As shown in Table 1 and Figure 1, pretrained networks like VGG/ResNet are able to encode object representation in a much better way even without finetuning on the training corpus, resulting in a significant boost in performance (9%). In the next phase of this project, we'll assess the performance improvement from finetuning.

Next, to analyze the failures of image representations, we present a confusion matrix of the the image labels and their

Approach	Val Acc
2-layer CNN + MLP	27.3
ResNet Features + MLP	**36.4**
VGGNet Features + MLP	36.1

Table 1. Object Recognition using shallow networks on GQA dataset

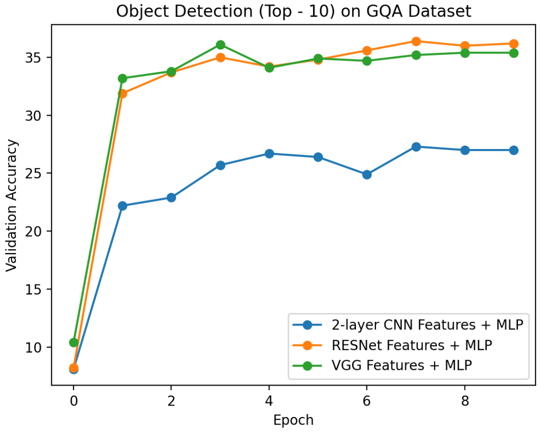
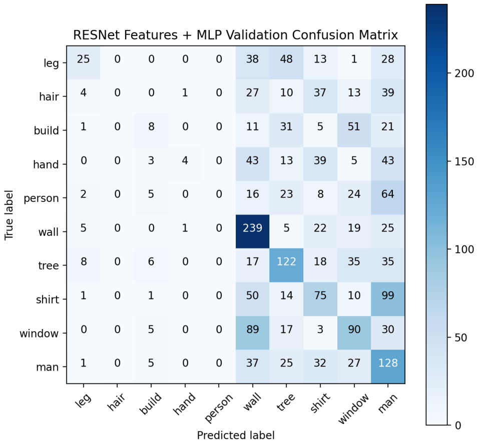


Figure 1. Evaluation of feature extractors for object detection



	leg	hair	build	hand	person	wall	tree	shirt	window	man
leg	25	0	0	0	0	38	48	13	1	28
hair	4	0	0	1	0	27	10	37	13	39
build	1	0	8	0	0	11	31	5	51	21
hand	0	0	3	4	0	43	13	39	5	43
person	2	0	5	0	0	16	23	8	24	64
wall	5	0	0	1	0	239	5	22	19	25
tree	8	0	6	0	0	17	122	18	35	35
shirt	1	0	1	0	0	50	14	75	10	99
window	0	0	5	0	0	89	17	3	90	30
man	1	0	5	0	0	37	25	32	27	128

Figure 2. Confusion Matrix for object recognition

predicted labels. Figure 2 shows a confusion matrix that presents failure cases and some findings on object recognition tasks when using ResNet(He et al., 2015) features followed by a single linear layer. One of the first observations is that the model is unable to encode the distinction between a 'person' and a 'man' representation, implying a gender bias in the dataset. Furthermore, we can observe that the model had trouble distinguishing between 'window' and 'wall' because they frequently appear combined in photos. Furthermore, 'wall' (16.7 percent) has a slightly higher percentage of total objects than 'window' (11.2 percent), resulting in a model bias in favor of 'wall'. Similar observations can be made between the 'shirt' and 'man' labels. According to previous observations, balancing across similar objects and objects which appear together in images is also significant. We plan to make use of these observations to learn better object representation method for our multimodal QA model.

2.2. Image Embeddings

For the image embeddings, 3 different variations were considered. We used pretrained CNN models - ResNET-18 and VGG-11 and also used a variation of the mean of object features extracted by a Faster-RCNN model (Ren et al., 2015). We extracted the embeddings for the images and plotted t-SNE representations of the embeddings with clus-

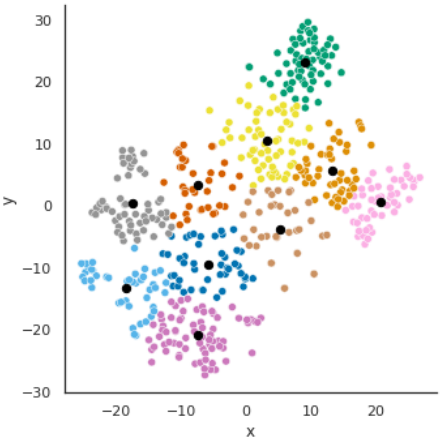


Figure 3. t-SNE plot for image representations extracted using pretrained ResNet-18

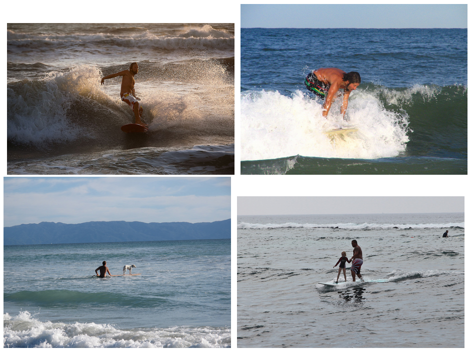


Figure 4. ResNet-18 image representations from 1 cluster show similar themes between image clusters

tering. We took the cluster centroids and extracted 3 closest neighbours of each image to see the similarities between the images. All of the 3 representations were able to group similar images into clusters. The t-SNE plot for the image representations using ResNET-18 is shown in Figure 3. Figure 4 shows how ResNet-18 grouped images that are very similar together, capturing all images where people are surfing. However, sometimes, these embedding are not able to distinguish between visually different images, although they have some similarity with respect to the color/objects in the image. Figure 5 shows how VGGNet-11 grouped images together which aren't very visually similar which shows that for some images, it might be difficult for the model to generate good embeddings.

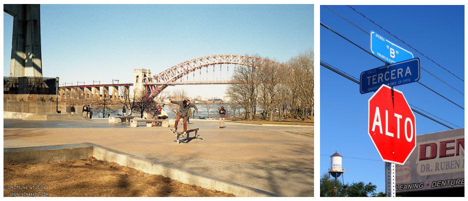


Figure 5. Different types of images which are identified close to each other by VGGNet

2.3. Textual Features Analysis

2.3.1. Sentence-Level Language Representations

First, we tested multiple sentence embeddings based on different variations of the BERT model (Devlin et al., 2019) . Four pretrained BERT models were considered during the analysis of language data for sentence-level contextual representations: BERT (Devlin et al., 2019), sentenceBERT (Reimers & Gurevych, 2019), roBERTa (Liu et al., 2019), and T5 (Raffel et al., 2019). The final hidden state was extracted from all four pretrained models, and the dimensionality of each sample was reduced to 50 using the *pca* (Jolliffe, 1986) Python library. K-Means clustering (Hartigan & Wong, 1979) was used with the number of clusters set to 10. Afterwards, the top 6 question-answer pairs were extracted that were closest to each representative cluster. Of the four models analyzed, the RoBERTa and T5 model performed qualitatively better. In multiple instances, the pretrained RoBERTa and T5 model were able to build clusters that were based on the same sentence clause, question type and of similar structure (ex. clustering questions that being with "Is the ⟨noun⟩ to the right or to the left of the ⟨noun⟩ that is ...). For the other two pretrained models, instances like with BERT (Devlin et al., 2019) and sentenceBERT (Reimers & Gurevych, 2019) were fewer and far between. Sometimes, some clusters in the other two models were able to group some similarly structured questions, but those clusters also included sentences of significantly different structure. The t-SNE (van der Maaten & Hinton, 2008) plot for the sentence representations of the RoBERTa and T5 model are shown in the Figure 6 and 7. We also analyze the clustered sentences which demonstrates T5 embeddings show more discrete clusters where RoBERTa embeddings sometimes (although very few cases compared to other models) overlap with other clusters (Table 2).

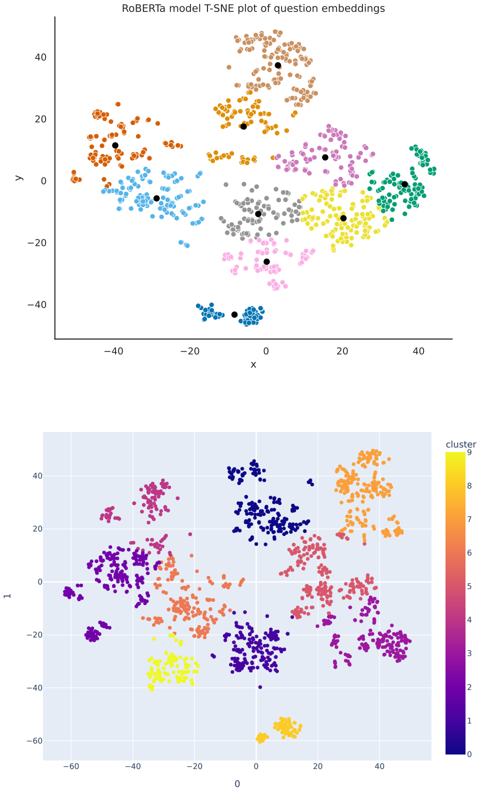


Figure 6. t-SNE plot of question embeddings from pretrained RoBERTa(top) and T5(bottom) models.

Lastly, we wanted to detect some explainable trends within question embeddings. To do so, we separated the questions into five groups based on the question type. This was done by spreading the questions based on their first word. Questions that had 'What', 'Is', 'Are', and 'Which' formed the different groups, and if a question started with a different word, it was placed in the 'Other' question type. From Fig. 8 we can observe that these questions form little clusters, where they almost fall into a line along the second axis of a t-SNE projection. Similarly, as these question types are singular and plural versions of themselves, we can observe that questions that ask about multiple objects end up having larger values along the second axis. This suggests that the ELMO (Peters et al., 2018) embeddings store useful information about what the question is asking about that is

Cluster Keywords	RoBERTa	T5
‘which side’	On which side of the image is the blue bag? On which side of the image is the young woman? On which side of the image is the bag? On which side of the image is the guy?	On which side of the picture is the man? On which side of the photo is the cellphone? On which side of the photo is the mat? On which side of the image is the bag?
‘is <noun>left or right’	Is the car to the right or to the left of the man that is on the street? Is the bottle to the right or to the left of the chair that is on the left? Is the woman to the right or to the left of the scooter that is not large? Is the player to the right or to the left of the person that is holding the cap?	Is the chair to the left or to the right of the vase on the table? Is the plate to the left or to the right of the utensil on the left? Is the backpack to the left or to the right of the people in the bottom part of the picture? Is the car to the left or to the right of the trailer on the right side?
‘color’	What color is the cat in the middle of the image? What color is the parked car that is parked on the road? Which color is the t-shirt that the man is wearing? What is the woman that is to the right of the man wearing?	Is the cat different in color than the wall? Does the post have a different color than the outfit? Is the color of the sink the same as the floor? Is the color of the shirt different than the screen?

Table 2. Example of questions clustered based on keywords using RoBERTa and T5 embedding

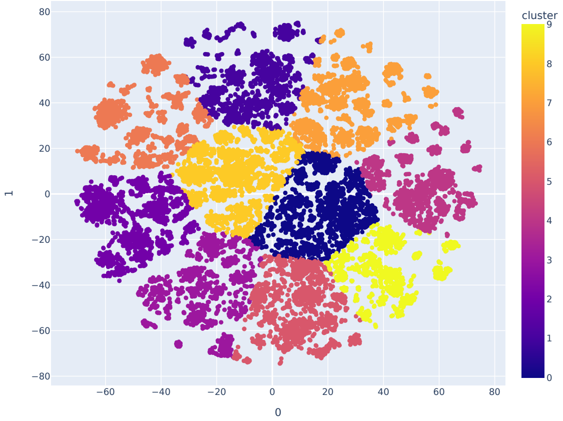


Figure 7. t-SNE plot of question embeddings from pretrained T5 model (whole 50k training sample).

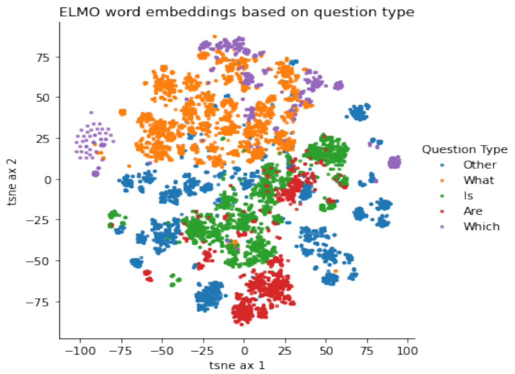


Figure 8. ELMO Embedding based on question type.

Dataset	Training Set	Validation Set	Exact Match
GQA	50000	10000	7.01%
NLVR2	86373	6982	50.89%

Table 3. Question answering without image

later necessary to formulate the answer on the VQA task. Lastly, from the ELMO embeddings, we can observe that questions form little clusters, which are likely a result of the algorithmic generation of questions based on scene graphs.

2.3.2. Text-only Question Answering

To understand if the GQA and NLVR2 datasets actually need multimodal analysis, i.e., whether a model needs both modalities to predict the correct answer or not, we take only question text with answer and run a multi-class text classification model. The motivation is to analyze whether the model can predict without any image. As we intend to model NLVR2 (Suhr et al., 2018) data alongside GQA in the future, we included NLVR2 in this experiment. For this task, we used a pretrained T5 (Raffel et al., 2019) model and finetuned it with a subset of the GQA dataset and whole NLVR2 (Suhr et al., 2018) dataset separately (Raffel et al., 2019). Then we tried to measure the model performance on the validation subset of each respective dataset. Table 3 sums up the result from this analysis. Clearly, GQA(Hudson & Manning, 2019) dataset needs both modalities to be analyzed to answer a question but NLVR2 (Suhr et al., 2018) achieves comparable accuracy to other complex models without looking into the image. That is not surprising as the NLVR2 dataset considers binary labels (True/False), so it is very possible that model is randomly guessing answers and getting it correct. So in terms of explaining reliability and utility of a multimodal model, GQA (Hudson & Manning, 2019) is better than NLVR2 (Suhr et al., 2018). But with NLVR2 (Suhr et al., 2018), we have to analyze the confidence of the predicted labels to be sure the model is considering both modalities while predicting the answer.

2.4. Question-Answering with Shallow Models and Different Modalities

In the investigation of a viable multimodal VQA model, it is best to investigate how simple QA models can perform and how different modality combinations play a role in these models. Three modalities were considered and compared: question representations from a pretrained RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2020) model, list of object labels from scene graphs, tokenized and processed through a word embedding layer, and image representations from a pretrained Resnet-18 (He et al., 2015) model, passed through a linear layer and a *tanh* nonlinear activation function. For T5 text embeddings with ResNet-18 image embeddings, we experimented using both *tanh* and *ReLU* nonlinear activation function and found that ReLU gives better validation accuracy than tanh. Modalities are jointly represented via simple vector concatenation. The joint representation is then passed through a fully connected linear layer, with hidden state dimensionality equal to the number of answer classes. The validation accuracies acquired between each modality combination were compared (see Figure 9 and 10). With RoBERTa embeddings, the modality combination that scored the highest validation accuracy was when the model used both question embeddings from RoBERTa and scene graph object label embeddings. On the other side, with T5 embeddings, using both modality achieved 0.055% gain in accuracy. Table 4 demonstrates the validation accuracy for models using T5 and RoBERTa text embeddings (all models used the ResNet-18 for image embeddings) which clearly shows that T5 embeddings improves the model performance. Another thing to note was that validation curves for the question + image modality pair tracked very closely to the question only modality model for both cases. The image representation being passed through the model was a Resnet-18 hidden state output for the entire frame of the image. One would say that looking at the entire scene would make it difficult to intelligently reason about an answer. This goes to show that some level of attention and grounding is desired in order for visual representations to contribute significantly to the performance of the model.

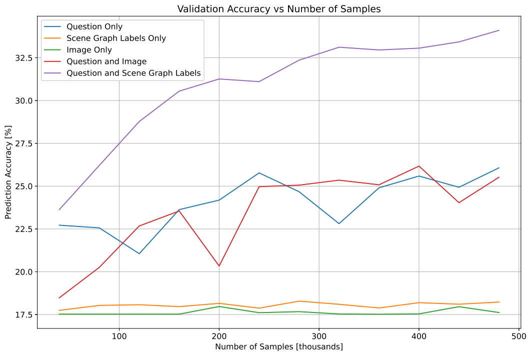


Figure 9. Validation accuracy of various combinations of simple QA models.

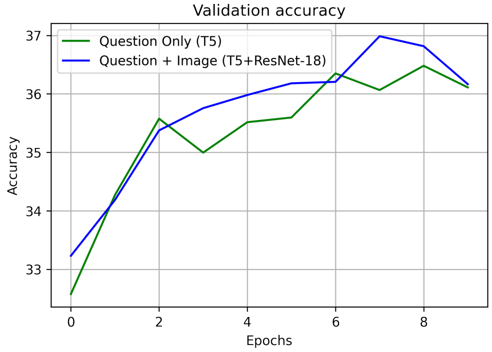


Figure 10. Validation accuracy for question only and question+image based QA models.

Training Condition	Validation Accuracy	
	RoBERTa	**T5**
Question and Image	25.52%	36.165%
Only Question	26.07%	36.11%

Table 4. Accuracy comparison between two text embeddings (all image embeddings are generated using ResNet-18) for different training condition

Lastly, we wanted to observe how different embeddings of one modality interact with the other modality. In order to do this, we concatenated image and question representations together into a single vector and passed it through a single layer preceptron. The results can be observed in Table 5. In the image modality, we tested Fast R-CNN (Ren et al., 2015) embeddings, a VGG embeddings, and a ResNet Embeddings. We can observe that while the results do not vary significanlty between different embeddings, Faster R-CNN slightly outperformed both VGG and ResNet. For the question embeddings, we tested RoBERTa, BERT, ELMO, and Sent2Vec embeddings (Pagliardini et al., 2018), and observe that BERT slightly outperformed the other models. However, similar to the image modality, the differences were relatively minor suggesting that all of the embeddings are able to encode some basic information that is necessary to answer the question.

Embedding	Single Layer Validation Accuracies			
	RoBERTa	**BERT**	**ELMO**	**Sent2Vec**
R-CNN	33%	37 %	36%	35%
VGG	30%	34%	34%	35%
ResNet	33%	35%	35%	35%

Table 5. Single layer MLP validation accuracies

References

Damodaran, V., Chakravarthy, S., Kumar, A., Umapathy, A., Mitamura, T., Nakashima, Y., Garcia, N., and Chu, C. Understanding the role of scene graphs in visual question answering, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Hartigan, J. A. and Wong, M. A. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Hudson, D. A. and Manning, C. D. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Jolliffe, I. *Principal Component Analysis*. Springer Verlag, 1986.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019.

Pagliardini, M., Gupta, P., and Jaggi, M. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 528–540, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1049. URL https://aclanthology.org/N18-1049.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations, 2018.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL http://arxiv.org/abs/1910.10683.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. URL http://arxiv.org/abs/1908.10084.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

Suhr, A., Zhou, S., Zhang, I., Bai, H., and Artzi, Y. A corpus for reasoning about natural language grounded in photographs. *CoRR*, abs/1811.00491, 2018. URL http://arxiv.org/abs/1811.00491.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Zhao, Z.-Q., Zheng, P., Xu, S.-t., and Wu, X. Object detection with deep learning: A review, 2018. URL http://arxiv.org/abs/1807.05511. cite arxiv:1807.05511.