Learning Latent Graph Representations for Relational VQA

Liyan Chen
Department of Computer Science
University of Texas at Austin
liyanc@cs.utexas.edu

Raymond Mooney
Department of Computer Science
University of Texas at Austin
mooney@cs.utexas.edu

Abstract

Recent state-of-the-art Visual Question Answering (VQA) systems achieve impressive results by tokenizing image objects and language queries and learning a transformer-based model on top of these representations. The key mechanism of transformer-based models is cross-attentions, which implicitly form graphs over tokens and act as diffusion operators to facilitate information propagation through the graph for question-answering that requires some reasoning over the scene. We reinterpret and reformulate transformer-based model to explicitly construct latent graphs over tokens and thereby support improved performance for answering visual questions about relations between objects. We demonstrate the feasibility of our hypothesis by using ground-truth scene graphs to help answer questions in the relational GQA dataset, showing a human-like upper-bound performance. Then we propose a graph-transformer-based model to learn latent graph representations in an end-to-end fashion without explicit supervision. Finally, we demonstrate that jointly learning latent graph representations along with VQA actually out-performs VQA using the current best supervised scene-graph parser.

1 Introduction

Within the deep-learning (DL) framework, vision-and-language tasks are often treated with a multi-modal-encoder approach, i.e. a vision encoder, a language encoder, and optionally alignments and fusion across modalities. There has been substantial past progress in the search for three parts of the multi-modal-encoder approach. Specifically in the Visual Question Answering (VQA) space, people first emphasized the vision side and treated images as either a singular entity or a continuous media (Karpathy and Fei-Fei, 2015; Kulkarni et al., 2011). With the emphasis on the vision side, a naturally induced form of alignments and fusion is soft attentions on image pixels correlated with language tokens (Rennie et al., 2017; Lu et al., 2017; Vinyals et al., 2015). A hypothesis supporting such an approach is that only parts of the image content are helpful for VQA (Xu et al., 2015; Pedersoli et al., 2017). As object-detection methods matured, people started to take a more ”bottom up” approach by extracting semantic information from and tokenizing the image to answer questions (Anderson et al., 2018a) , which lead to further progress. Concurrently, the language community proposed much stronger language encoders built using attention (Vaswani et al., 2017; Devlin et al., 2018), which lead people to develop VQA models with more emphasis on the language side (Das et al., 2017; De Vries et al., 2017; Selvaraju et al., 2019; Jiang et al., 2018; Lu et al., 2019; Tan and Bansal, 2019). Coincidentally, transformer-based language encoders can not only take advantage of the tokenization trend but also are intrinsically built for information fusion and alignments due to its core self-attention mechanism. Therefore, a wide variety of VQA systems with a transformer encoder on top of image tokenization have achieved impressive state-of-the-art results on many categories of VQA tasks, including VQA 2.0 and GQA (Li et al., 2019; Jiang et al., 2018; Lu et al., 2019; Tan and Bansal, 2019).

Such success of transformer-based VQA systems indicates the effectiveness of two insights: image tokenization, and pairwise token interaction across textual tokens and image tokens. We observe that pairwise token interactions collectively form a graph, and traversing this graph forms a kind of reasoning, which might be an explanation for the claims of reasoning capabilities of these transformer-based models (Li et al., 2019; Anderson et al., 2018b).

In order to further improve performance, especially on the ”relational” questions that require reasoning about the relations between objects, we reinterpret transformer-based VQA systems as graph convolutions, and propose a model that explicitly learns a latent graph representation to answer questions in an end-to-end fashion. We show that our model benefits from its latent graph representations and is able to outperform current state-of-the-art systems on the GQA dataset (Hudson and Manning, 2019), which emphasizes compositional, relational question-answering. In addition, we show that our model can effectively use graph representations and demonstrate a human-like upper-bound performance when we input a ground-truth scene graph into the model. This result is not surprising since the GQA dataset is derived programmatically from the ground truth scene graph, but it shows the potential of the idea of leveraging graph representations for VQA. To the best of our knowledge, current transformer-based models cannot benefit from graph information, and there have not been work on taking advantage of scene graphs or graph representations in general for VQA. Finally, we show the advantage of our end-to-end learned latent graph representation over current scene-graph parsers by showing that, on GQA, it out-performs a VQA system that uses a state-of-the-art scene-graph parser.

2 Related Works

2.1 UpDn

Current state-of-the-art VQA systems combine bottom-up attention for proposing image regions containing objects and top-down attention over image region features to predict final answers (Selvaraju et al., 2019; Jiang et al., 2018; Lu et al., 2019; Tan and Bansal, 2019). A common choice for architecturing these systems is to encode the image and question as two sets of features. Specifically, the image is encoded as a set of regions with corresponding visual features; the question is encoded as a set of textual tokens or a single question vector. Then, various approaches are used to go through those features, build interactions, and aggregate them to compute final answers.

2.2 Transformer-based VQA

Recently, there have been several concurrent works achieving impressive results with a similar idea on a spectrum of VQA datasets including VQA v2 and GQA. ViLBERT, LXMERT, and Unicoder-VL (Li et al., 2019; Lu et al., 2019; Tan and Bansal, 2019) are examples of this trend. They follow the UpDn idea of tokenizing images and textual questions as individual nodes. Their core idea includes two aspects: apply a large transformer encoder over multimodal nodes and take advantage of pretraining tasks on large datasets. Therefore, these models achieve similarly impressive results on various vision-language tasks including VQA.

2.3 Graph Neural Networks

Graph Neural Networks have gained much popularity due to its effectiveness of processing data with graph structures. They are usually categorized into two types: spectral (Bruna et al., 2013; Defferrard et al., 2016; Henaff et al., 2015; Kipf and Welling, 2016; Xu et al., 2019) and non-spectral methods (Chen et al., 2018; Gilmer et al., 2017; Monti et al., 2017). Graph Convolution Networks (GCN) is a line of works (Bruna et al., 2013; Kipf and Welling, 2016) that use spectral graph theory to develop convolution operations on graphs. A typical task for a GCN is node classification, as GCN is capable of learning node representations from a given static homogeneous graph. Graph Transformer Networks (GTN) (Yun et al., 2019) are a model for handling heterogeneous graphs, graphs with various types of edges, as well as generating new graphs. In our model, the goal is to learn to generate a latent graph representation and then perform node classification on the resulting heterogeneous graph. Therefore, GCNs and GTNs are used in our model as described below.

3 Attention as Graph

Transformer-based encoders aggregate the vision-language features. Here, we take a closer look at transformers and examine their similarity to graph-based models. Without loss of generality, transformer-based encoders can be described using the formula

$$f(H_Q^{(l)}, H_K^{(l)}, H_V^{(l)}) = \sigma(H_Q^{(l)} \dot{H}_K^{(l)^T}) H_V^{(l)} \qquad (1)$$

where σ is a normalization function and $H_Q^{(l)}, H_K^{(l)}, H_V^{(l)}$ represent queries, keys, and values for the l-th attention head, respectively. Such (query key value) triplets come from either a layer below or encoded vectors of the forementioned two sets of features with each token corresponding to a triplet. If we only look at one token in the output

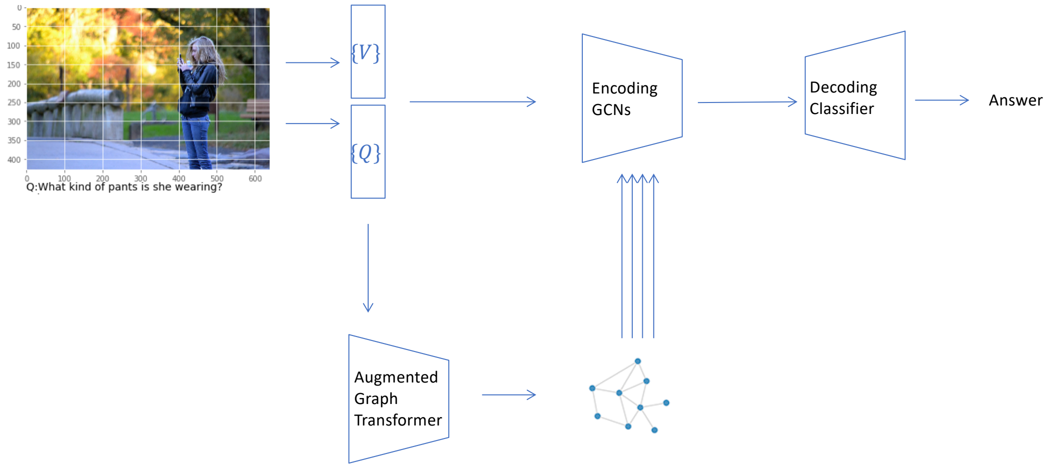

Figure 1: The schematics of our proposed architecture: we use Augmented Graph Transformer layers to generate the latent graph, a heterogeneous graph over input nodes. Then encoding GCN layers progressively encode input nodes according to the latent graph. The final answer is decoded from a decoding classifier.

i with K_i, Q_i, V_i representing the triplet, it can be seen as

$$f(H_Q^{(l)}, H_K^{(l)}, H_V^{(l)})_i = \sum_{j \in \Omega} \tilde{\sigma}(Q_i^{(l)}, K_j^{(l)}) V_j^{(l)} \tag{2}$$

where $\tilde{\sigma}$ represents an interaction function related to σ and dot product and Ω represents the scope of tokens attended over. LXMERT and VilBert both manipulate Ω in different stages. They first let Ω be the same set that i belongs to and then set Ω to the opposite set to allow information exchange. Restricting Ω on two complementary subsets can be seen as a special case of not restricting Ω with the same number of passes. This generalization allows us to reformulate the equation as

$$f(H_Q^{(l)}, H_K^{(l)}, H_V^{(l)}) = [A_{\tilde{\sigma}}^{(l)}]_{|\Omega| \times |\Omega|} V_{|\Omega|}^{(l)} \tag{3}$$

where $[A_{\tilde{\sigma}}^{(l)}]$ is a $|\Omega| \times |\Omega|$ matrix that simply collects all results of $\tilde{\sigma}(Q_i^{(l)}, K_j^{(l)})$ by having $[A_{\tilde{\sigma}}^{(l)}]_{i,j} = \tilde{\sigma}(Q_i^{(l)}, K_j^{(l)})$. This is essentially an operation of propagation through a graph with adjacency matrix $[A_{\tilde{\sigma}}^{(l)}]$, an input-dependant matrix. This interpretation gives rise to our model of GNN with learned graphs.

In addition, transformer-based encoders typically have multiple attention heads for each pass, which implies that there are multiple independent graphs corresponding to the same set of nodes for each operation. Conceptually, this view is equivalent to a heterogeneous graph (Shi et al., 2016), where the graph can have k distinct types of edges and can be represented by k homogeneous subgraphs, each corresponding to a particular edge type. This interpretation also coincides with the scene graph idea from the vision community (Yang et al., 2018; Krishna et al., 2016), a heterogeneous graph over objects with edges corresponding to semantic relations (such as look-at, sitting-on, and next-to), which describes a scene at a relational level and can support answering questions requiring reasoning about such relations. Thus, such a graphical interpretation of transformer encoders and motivations from the scene-graph idea give rise to our architecture of learning a latent graph representation of a scene using Graph Transformers to answer relational VQA questions.

4 Method

We establish our model as a three-stage pipeline: candidate-graph generation, latent graph generation, and answer classification. Like LXMERT and VilBert, our model begins with embedding the textual question into a list of vectors and encoding the image as a list of image regions each represented by a vector. The union of these two sets of vectors is considered as the nodes of our graph model, $\mathcal{V} = \{v_1, ..., v_{\|\mathcal{V}\|}\}$, and we propose a Graph-Transformer-based model to learn a latent graph representation on top of these nodes, and use this graph to answer questions using a standard Graph Convolution Network (GCN).

4.1 Candidate-graph Generation

Since we are interested in learning graph representations from scratch without any supervised input graph, there is a bootstrapping stage to generate the initial set of candidate graphs. A straightforward way to bootstrap is to take one layer of multi-head cross-attentions as described in §Attention as Graph. Specifically,

$$[A^{(l)}_{\tilde{\sigma}}]_{i,j} = \tilde{\sigma^{(l)}}(v_i, v_j) \tag{4}$$

where v_i, v_j are node vectors from the node set $\mathcal{V}$ and $A^{(l)}_{\tilde{\sigma}}$, $\tilde{\sigma^{(l)}}$ correspond to l-th attention head.

4.2 Latent Graph

We propose the Augmented Graph Transformer (AGT), a GT-based architecture, to learn the latent graph representation. Like a GT layer, an AGT layer selects softly from input graphs and takes compositions of them to connect meta-paths. It consists of two stages: the first stage learns convex combinations of input graphs as intermediate graphs, and the second stage generates output graphs by the compositions of intermediate graphs from the previous stage. However, the original GT layer is limited to a set of input-independent convex weights and composing two adjacency matrices per layer. AGT, on the other hand, learns input-dependent convex weights, composes multiple adjacency matrices per layer, and generates a new set of adjacency matrices from the last layer's outputs.

Convex Combination Let $\mathbb{A} \in \mathbb{R}^{N \times N \times C}$ be the adjacency matrix with C channels. This stage computes intermediate adjacency matrices $\mathbb{Q}$ as $\mathbb{Q}_{m,o} = \sum_{c \in C} \alpha_c^{m,o} \mathbb{A}_c$, where o is the output channel and m represents the m-th hop on the meta-path to be composed. The convex weights are given by $Softmax_{c \in C}(\lambda^{m,o})$ and λ is computed as a Rayleigh quotient of the graph Laplacian:

$$\lambda_c^{m,o} = \frac{g^T \mathcal{L}_c g}{g^T g} \tag{5}$$

where $g \in \mathbb{R}^{C \times M \times O}$ is a trainable parameter. The Rayleigh quotient has the nice property of having critical points at eigenvalues of the graph Laplacian and therefore reflecting the spectral structures of the graph. We know that the spectrum of graph Laplacians is closely related to graph structures, including connectedness and regularity, so that this formulation is not only input-dependent but can also reflect the structural properties of the input graph (Chung and Graham, 1997). Like GT layer, we also assign $\mathbb{A}_0$ as an identity matrix to allow both short meta-paths and long meta-paths.

Meta-path Composition Like the GT layer, AGT connects meta-paths by composing the intermediate adjacency matrices with multiplication, i.e. $\prod_{m \in M} D^{-1} \mathbb{Q}_{m,o}$ and output adjacency matrices with O channels for the next layer. The D^{-1} normalization maintains the numerical stability for the product.

4.3 Answer Classifier

As the latent graph representation $\mathcal{G}$ is given as C channels of adjacency matrices, we combine each one of them with a layer of GCN to learn node representations progressively as

$$H^{(t+1)} = \sigma(D_t^{-\frac{1}{2}} \mathcal{G}_t D_t^{-\frac{1}{2}} H^{(t)} W^{(t)}) \tag{6}$$

Finally, we follow the standard practice to decode the answer from the node vector of CLS by stacking two layers of MLP on top of it. As standard, we treat final answer generation as a multi-label classification task.

5 Implementation Details

Like VilBERT, we initialize our embedding layer with $BERT_{LARGE}$ weights and the image region detection features are from Faster R-CNN (Ren et al., 2015) pretrained on VisualGnome (Krishna et al., 2016). We only keep the image region detections with top-36 confidence scores and the feature vectors are mapped to the hidden dimension with a fully-connected layer.

In terms of hyperparameters, the hidden dimension is 1024, meta-path hops per AGT layer $M = 6$, adjacency matrix channels $C = 64$, latent graph channels $O = 8$, and 8 layers of AGT. We use the Adam optimizer with cross-entropy loss to train our model with initial loss at 3e-4.

6 Experimental Evaluation

Dataset We present our experiment evaluations on the GQA dataset, a relational VQA dataset including 22M reasoning questions about object relationships. We use this dataset to specifically evaluate the reasoning capability of our model on relational VQA. In addition, the GQA dataset includes a ground truth scene graph for each image,

	Models				
	ViLBERT	LXMERT	AGT	Ground truth scene graph	Graph R-CNN + AGT
GQA Acc	71.4	62.7	77.9	87.6	70.6

Table 1: Experimental Results on GQA

from which the question-answer pairs are generated. More than 92% of the questions in GQA have 4 compositional steps or less, so we chose the meta-path hops per AGT layer at 6 as described in the last section.

Baselines We use LXMERT and ViLBERT as our state-of-the-art baseline models. We cite the highest accuracy of LXMERT from the leaderboard and run our implementation of ViLBERT on GQA. We follow the pretraining process of ViLBERT and fine-tune the model on GQA as described in (Lu et al., 2019).

Upper bound We hypothesize that a reasonable model can easily learn to perform GQA if it has access to the ground truth scene graph from which the QA data was generated. Therefore, we set up an experiment to examine the potential upper bound of our model by inputting ground truth scene graphs. To implement it, we replace the candidate-graph generation bootstrapping with the ground truth scene graph input and strip the AGT layers down to 1 to just convert the graph format to a compatible one. We train and test the model with this setup to establish a putative upper bound.

Scene graph parser To further evaluate our latent graph representation, we also set up our model to take scene graph predictions from a state-of-the-art scene graph parser, Graph R-CNN(Yang et al., 2018). The implementation is similar to the upper bound system – we collect the predictions of the scene graph parser, input them into the model, and remove the bootstrapping as well as most of the latent graph generation layers. The training and testing is the same.

6.1 Results and Discussion

Table 1 shows the GQA performance of the baselines and our AGT-based models. We observe that our AGT-based model achieves a new state-of-the-art for this dataset and has the highest performance among the three models that do not have access to a gold-standard scene graph. Our model with ground truth scene graph inputs achieves very high performance, which indicates that a graph-based neural model that can fully utilize scene graph information has the potential to approach human-like performance. On the other hand, the performance of the AGT model with a state-of-the art scene graph parser is similar to the baseline models, which shows that our end-to-end learned latent graph representation is actually more effective than directly using current scene graph parsers, despite never using the supervised scene graphs available to this model during training. We believe this quite surprising result clearly demonstrates the capabilities of our latent graph approach (as well as the serious limitations of existing scene-graph parsers).

7 Conclusion

We take inspirations from the successful transformer-based VQA models and reinterpret them as graph convolution models to leverage the graphical nature of VQA. We proposed a new approach that learns a latent graph representation for answering relational VQA questions using an Augmented Graph Transformer. The proposed AGT model can process heterogeneous graph data and generate new graphs, which leads to the desired latent graph representation. For the GQA dataset with a specific emphasis on relational, compositional visual question-answering, the proposed model shows state-of-the-art performance due to its effective latent graph representation. In addition, the upper-bound performance of the proposed model using ground truth scene graphs indicates the huge potential of such graph-based models in VQA tasks.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen

Gould, and Anton van den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and deep locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Jie Chen, Tengfei Ma, and Cao Xiao. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*.

Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. 92. American Mathematical Soc.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. 2017. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org.

Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6700–6709.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0. 1: the Winning Entry to the VQA Challenge 2018. *arXiv preprint arXiv:1807.09956*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep Visual-semantic Alignments for Generating Image Descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. 2011. Baby talk: Understanding and generating image descriptions.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2019. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124.

Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. 2017. Areas of Attention for Image Captioning. In *ICCV-International Conference on Computer Vision*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *NIPS*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical Sequence Training for Image Captioning. In *CVPR*, volume 1, page 3.

Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Dhruv Batra, and Devi Parikh. 2019. Taking a HINT: Leveraging Explanations to Make Vision and Language Models More Grounded. In *ICCV*.

Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2016. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE.

Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. 2019. Graph wavelet neural network. *arXiv preprint arXiv:1904.07785*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International Conference on Machine Learning*, pages 2048–2057.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. In *Advances in Neural Information Processing Systems*, pages 11960–11970.