# How Important are Visual Features for Visual Grounding? It Depends.

**Fan Yang** and **Prashan Wanigasekara** and **Mingda Li** and **Chengwei Su** and **Emre Barut**
Amazon Alexa, Cambridge, MA, USA
{fyaamz,wprasha,mingda,chengwes,ebarut}@amazon.com

## Abstract

Multi-modal transformer solutions have become the mainstay of visual grounding, where the task is to select a specific object in an image based on a query. In this work, we explore and quantify the importance of CNN derived visual features in these transformers, and test whether these features can be replaced by a semantically driven approach using a scene graph. We propose a new approach for visual grounding based on BERT (Devlin et al., 2019), named metaBERT, that enables reasoning over scene graphs. In order to quantify the importance of visual features, we inject both the scene graph information and the visual features to metaBERT. We find that the additional performance due to the visual features vary among datasets, but is mainly limited to a 10-15% accuracy improvement. Through detailed experiments, we explore the effect of the scene graph quality on the performance, and observe that utilizing scene graphs is notably beneficial for selecting non-human objects.

## 1 Introduction

Visual grounding (VG) refers to the multi-modal problem in which an image and a query about the image is input to a machine learning model, and the model is tasked to find the matching object in the image. VG is critical in embodied AI: accurate object recognition is a crucial component in selection or manipulation of objects – it is not reasonable to expect a robot to perform an action on a specific object, e.g. 'the red square block on the table', if it cannot recognize the object in the first place.

The majority of the literature so far has focused on two main scenarios: (i) phrase localization that relates phrases of a full sentence describing an image to its corresponding objects (Plummer et al., 2015; Rohrbach et al., 2016; Plummer et al., 2017; Wang et al., 2019), and (ii) referring expression comprehension that detects a particular region through object categories, attributes, and relationships with other objects (Yu et al., 2016; Hu et al., 2016; Yu et al., 2018; Liu et al., 2019a; Yang et al., 2019c). Both scenarios require multi-modal models to process natural language and visual features, which is often achieved by using an R-CNN for the visual features (Girshick, 2015; Ren et al., 2015), a language model (e.g. an LSTM or a transformer) for the query, and finally a fusion model that combines both features (Su et al., 2019; Lu et al., 2019).

A core component for the success of VG is to identify which objects are in the image, and how these objects relate to each other. This sub-task is ripe for application of scene graphs (SG), which view objects as nodes and relationships of objects as edges. Scene graphs have been used as supplementary features for question answering (Li et al., 2019; Lee et al., 2019), image generation (Johnson et al., 2018), captioning (Yao et al., 2018; Zhong et al., 2020), video understanding (Ma et al., 2018), and image retrieval (Wang et al., 2020a; Mafla et al., 2021). However, few works (Cirik et al., 2018; Yang et al., 2020) explore visual grounding with scene graphs and the limitations of SG in this framework are not clear, especially in the light of recent developments such as multi-modal transformers.

We note that a successful scene graph based approach for VG would have numerous implications: (i) A VG solution that utilizes scene graphs would be far more interpretable. Scene graphs translate image regions to natural language, and thus provide an interpretable input. Further, they make it possible to compute the importance of any of the components in the scene graph via perturbation driven interpretability techniques (Ribeiro et al., 2016; Smilkov et al., 2017) or by simply changing (or removing) the edges (or nodes) in the scene graph. (ii) The method would allow easy data augmentation. The components of the SG can be altered to create new training examples, for instance by replacing the instances of the word "red" to "blue" both in the scene graph and the query. This

allows easy adaptation of the model to zero-shot scenarios — similar augmentation techniques are not possible via methods that utilize visual features except through image generation methods such as DALL-E (Ramesh et al., 2021) or LX-MERT (Tan and Bansal, 2019; Cho et al., 2020). (iii) In various applications, the scene graph is easier to obtain than the visual features. For instance, in voice-based navigation applications on the web, the scene graph can be derived immediately from the markup language (e.g. HTML) whereas utilizing visual language transformers would add significant computational cost.

To that end, in this paper we investigate whether high quality scene graphs can replace visual features for visual grounding. Our contributions are as follows:

- We propose a new visual language architecture, named "metaBERT", that utilizes masked attention heads to reason over scene graphs instead of visual features as in other visual language transformers.
- We present studies that quantitatively assess the importance of visual features in visual language transformers by comparing metaBERT against state-of-the-art visual transformers. We find that, unsurprisingly, the difference depends on various factors, such as scene graph accuracy.
- We observe that in some datasets scene-graph based approaches can perform as well as state-of-the-art visual language transformers and that the additional benefit of the visual features are limited.

In the following sections, we provide the details of metaBERT in Section 2. The numerical results and analysis are given in Section 3. The literature review and the ablation studies are relegated to the Appendix.

## 2 Enabling VG Through Scene Graphs via metaBERT

In this section we describe the architecture of the proposed model. We utilize BERT (Devlin et al., 2019) as the starting point: the philosophy of our design is to allow BERT to encode the scene graph using natural language, while also considering the structured relationship between object pairs. In order to achieve the said scheme, we utilize token type embedding based on the scene graph connections to handle the relationship triplets of (subject, relation, object).

The visual grounding task involves an image $m$ and a natural language query $q$. For a given $(m, q)$ pair, the model outputs a bounding box $b = [x, y, w, h]$, where $(x, y)$ denote the bottom-left coordinate and $(w, h)$ are the width and height, respectively. Ideally, the box covers the target referred to by the query.

In our work, we investigate the case where the input image $m$ is replaced with a directed scene graph $g = (V, E)$. The vertex set $V$ contains $n$ objects detected in the image, and each object is represented by its category $c$ and attribute $a$. The edge set $E$ contains the relationship between objects. The $k$-th relationship $(o_i, r_k, o_j)$ is represented by its name $r_k$ and a pair of objects $(o_i, o_j)$. The scene graph grounding task can be formulated as follows: $\hat{o} = \text{argmax}_{o \in V} f_\theta(q, g)$, where $f_\theta$ is parameterized by learnable parameters $\theta$, and it assigns a score to each object in $V$ that indicates the object's relevance to the query.

### 2.1 MetaBERT

In order to feed BERT with the given inputs, we treat each object as a single training instance and concatenate (i) the natural language query, (ii) the category and the attribute of the target object, and (iii) its relationship with other objects, into a single flat sequence. We illustrate how we construct the input sequence from a scene graph in Figure 1. We train the model to predict if the target object is referred to by the query or not, and, thus, an image could have training instances as same as the number of objects. During inference, we rank the prediction score of all targets of an image and take the highest one as the answer.

We name the proposed model metaBERT due to its capability of encoding structured meta information, the main architecture of the model is given in Figure 2. We input the summation of four embeddings, namely, token embedding, segment embedding, sequence position embedding, and token type embedding to indicate the connections in the scene graph. In addition, MetaBERT can take visual feature embeddings for each object. Following BERT and other visual-language transformers, the input sequence starts with a special token "[CLS]", the query and an another special token "[SEP]". We append the target object and its relationships, and


Figure 1: Constructing the input for metaBERT from the scene graph of the target object “Printer". We omit the query segment to focus on the scene graph. The sequence starts with visual features of the whole image and the target object, followed by the attribute (purple) and the category (red) of the target, the relationship (yellow) where the target is the object (subject is marked by green), and the relationship (yellow) where the target is the subject (object is marked by blue). The model encodes one object and its relationship per instance. The relationship of other object pairs, e.g., “box" and “tray", will be processed when either one is the target.

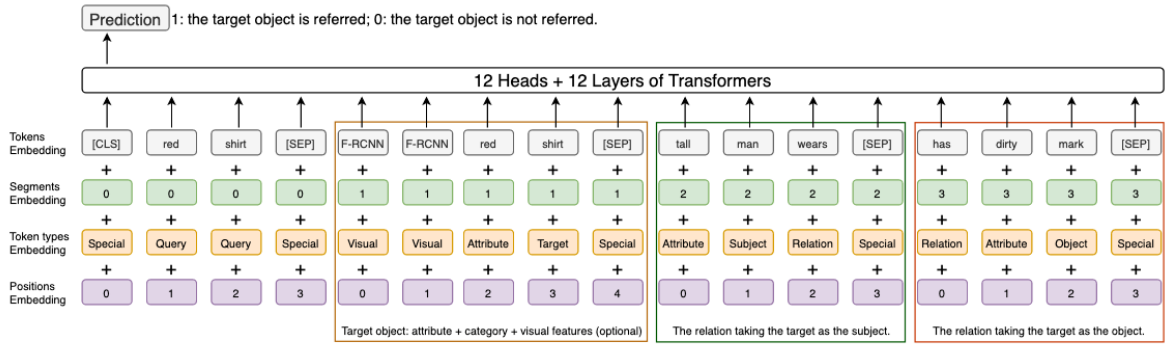



Figure 2: The architecture of metaBERT. The model treats each object as a single instance and takes the concatenation of the query, the category and the attribute of the object, and its relationship with other objects, as the input. It outputs a binary value to predict if the target object is referred to.

use “[SEP]" to separate them. We only keep the relation name and the name of the other objects in the relationship to avoid duplicating the name of the target object multiple times. All tokens are tokenized with a 30,000 vocabulary and embedded using the Word Piece Embedding (Wu et al., 2016). A learnable position embedding is leveraged to indicate the order of the input sequence. The index of position is reset to zero after each “[SEP]". Different from BERT that has two segments (two sentences), we label the segment based on number of “[SEP]" tokens, which helps to distinguish relationships. We further introduce token type embeddings to differentiate components of relationships.

MetaBERT can also leverage visual feature embeddings. As in other work (Su et al., 2019; Lu et al., 2019, 2020), we obtain the visual features of each object $v_i^{RoI}$ by applying Faster R-CNN (Ren et al., 2015) and extract the second to last layer's output. We also provide the average feature of all objects to represent the full context. Visual features are input to the model in the target segment, at the same layer as the other token embeddings.

**Graph Mask:** When processing the target object, we do not consider the relations among other objects. For instance, in Figure 1, the relation between “Tray" and “Desk" will not be input to the model when “Printer" is the target. Note that the relationship will be considered once either “Tray" or “Desk" is included as the target.

To make better sense of such scene graph structures, we regularize the transformer's attention scheme. We rely on attention masking to ensure that the attention weights are related to the underlying structure of the inputs, and that tokens representing different identities (e.g. relations vs. object attributes) attend each other relatively less. More specifically, given the input sequence $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$, the self-attention outputs $\mathbf{Z} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$, where $\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{X}\mathbf{W}^Q, \mathbf{X}\mathbf{W}^K, \mathbf{X}\mathbf{W}^V$, re-

| | V7w-pointing | | RefCOCO+ Detected | | | RefCOCO+ Ground Truth | | |
|---|---|---|---|---|---|---|---|---|
| | Val | Test | Val | TestA | TestB | Val | TestA | TestB |
| MAttNet | - | 86.42 | 65.33 | 71.62 | 56.02 | 71.01 | 75.13 | 66.17 |
| VL-BERT w/o pre-training | - | - | 66.03 | 71.87 | 56.13 | 74.41 | 77.28 | 67.52 |
| ViL-BERT w/o pre-training | - | - | 68.61 | 75.97 | 58.44 | - | - | - |
| VL-BERT with pre-training | - | - | 71.60 | 77.72 | 60.99 | 79.88 | 82.40 | 75.01 |
| ViL-BERT with pre-training | - | 80.51 | 72.34 | 78.52 | 62.61 | - | - | - |
| metaBERT w/o visual features | 81.07 | 80.39 | 56.77 | 61.36 | 50.75 | 60.87 | 63.09 | 54.86 |
| metaBERT with visual features | 80.25 | 80.06 | 69.04 | 75.15 | 60.03 | 73.30 | 77.09 | 66.72 |

Table 1: Main results (accuracy) on various visual grounding tasks. RefCOCO+ Ground Truth and Detected refer to the cases where the object bounding boxes are provided or are estimated via an object detector, respectively.

spectively, and $\mathbf{W}^Q$, $\mathbf{W}^K$ and $\mathbf{W}^V$ are learnable weights, and they can be initialized from a pre-trained BERT. The graph mask zeros out attention scores for particular tokens by introducing a special Boolean matrix $\mathbf{M}$ and a large negative scalar $C$. Then, we set $\mathbf{Z} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{M}C\right)\mathbf{V}$.

**Scene Graph Generation:** We follow Neural Motifs (Zellers et al., 2018) with unbiased generation (Tang et al., 2020) to obtain machine generated scene graphs. The generator is given the image along with the region proposals; the latter can either be the ground truth or bounding boxes obtained via object detectors. We train the model from scratch using the Visual Genome dataset (Krishna et al., 2017).

## 3 Experiments

We evaluate metaBERT on two public visual grounding datasets, Visual7w Pointing questions (Zhu et al., 2016) and RefCOCO+ (Yu et al., 2016). The Visual7w Pointing dataset provides 93K training examples, 37K examples for validation, and 57K examples for testing. Each example contains *four* candidate regions. RefCOCO+ takes place in a more practical setting, where all possible objects in the image can be referred to. It has 120K and 10K examples for training and validation, respectively. The authors divide the test set into two splits: near 6K examples in TestA that covers human objects and 5K examples in TestB that involves non-human objects.

We compare against three methods: ViL-BERT, VL-BERT and MAttNet (Yu et al., 2018). The first two models are state-of-the-art techniques that extend the transformer structure to leverage multiple modalities, and their architectures are close to metaBERT. MAttNet uses a modular attention structure and is specialized to attend to certain attribute words.

### 3.1 Main results

We report the accuracy of various models in Table 1. The last two lines compare the results of metaBERT with and without the visual features. We see that the visual features provide no additional benefits for the Visual 7w-pointing dataset, but it results in a 10% to 14% absolute accuracy improvement in RefCOCO+.

On the Visual7w-pointing dataset, MAttNet outperforms ViL-BERT and metaBERT by a large margin, possibly because it explicitly matches the modular components of the query to the relevant visual features. MetaBERT achieves comparable performance to ViL-BERT, whereas adding visual features slightly reduces the accuracy. The result of metaBERT is much different on RefCOCO+: when combined with visual features, it yields a comparable result to non-pretrained visual language transformers, but the scene graphs by themselves do not contain enough information to address the query.

Since we use annotated scene graphs on Visual7w-pointing and generated scene graphs on RefCOCO+, we hypothesize that the quality of the scene graph could significantly affect the performance of metaBERT without visual features. We explore the idea in Appendix A.3.

MetaBERT is trained on a single object at a time, whereas VL-BERT and ViL-BERT are both trained on multiple objects in a batch, and thus can learn better contextualization. We also note that metaBERT with visual features has a higher accuracy than non-pretrained ViL-BERT and VL-BERT on RefCOCO+ TestB. TestB set contains non-human objects, and its expressions are easier to reason on via scene graphs. Finally, we note that the visual features require a large amount of computing time. For instance, on the Visual7w-pointing dataset, metaBERT without visual features requires less than three hours to train one epoch, and takes around 23ms per instance on inference, whereas metaBERT with visual features takes 18

hours per epoch and 95ms per instance for training and inference, respectively.

We include the implementation details in Appendix A.2. We further present the ablation study and qualitative results in Appendix A.4 and A.6.

## 4 Conclusion and Discussion

In this work, we examine visual grounding and investigate the feasibility of using scene graph to replace visual features. We propose metaBERT to encode and reason over scene graphs, and compare metaBERT against state-of-the-art visual language transformers to assess the importance of visual features. The results suggest that scene graph quality is an important factor that determine the performance. We find that metaBERT works well with annotated scene graph. This allows potential applications where scene graph can be derived from the markup language (e.g. HTML) rather than visual features from an object detector.

In our future works, we will explore ideas to automatically generate <text, scene graph, image> triplets, and pre-train metaBERT on them. We hope to see additional improvements as pre-training VLBERT and ViLBERT. MetaBERT requires as many inferences as objects in the scene, which might not be practical for complex scenes. We will explore graph-based methods that directly process the full scene graph and retrieve the relevant object to match the query.

**Ethical consideration** MetaBERT is for visual grounding applications where scene graphs can be leveraged, and does not involve identity characteristics. We conduct experiments on benchmark datasets that have been well examined in the literature. Besides, we observe that processing scene graphs for grounding requires less computing time than calculating visual features from scratch, and it could potentially reduce energy and carbon costs. However, we must point out that some scene graph generation methods (like the one we used in the paper) still rely on visual features. Thus, exploring alternative approaches that are eco-friendly to obtain scene graph could be a possible direction.

## References


Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Remi Cadene, Hedi Ben-younes, Matthieu Cord, and Nicolas Thome. 2019. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng Cai and Wai Lam. 2020. Graph transformer for graph-to-sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7464–7471.

Benson Chen, Regina Barzilay, and Tommi Jaakkola. 2019a. Path-augmented graph transformer network. *arXiv preprint arXiv:1905.12712*.

Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. 2019b. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4613–4623.

Jaemin Cho, Jiasen Lu, Dustin Schwenk, Hannaneh Hajishirzi, and Aniruddha Kembhavi. 2020. X-lxmert: Paint, caption and answer questions with multi-modal transformers. *arXiv preprint arXiv:2009.11278*.

Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. 2018. Using syntax to ground referring expressions in natural images. In *AAAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. 2019a. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10323–10332.

Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. 2019b. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1315–1325.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. 2017. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Zecheng He, Srinivas Sunkara, Xiaoxue Zang, Ying Xu, Li juan Liu, Nevan Wichers, Gabriel Schubiner, R. Lee, and Jindong Chen. 2020. Actionbert: Leveraging user actions for semantic understanding of user interfaces. *ArXiv*, abs/2012.12350.

Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. 2020. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*.

Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2017. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33.

Chenchen Jing, Yuwei Wu, Mingtao Pei, Yao Hu, Yunde Jia, and Qi Wu. 2020. Visual-semantic graph matching for visual grounding. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4041–4050.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Rajat Koner, Poulami Sinhamahapatra, and Volker Tresp. 2020. Relation transformer network. *arXiv preprint arXiv:2004.06193*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. 2019. Visual question answering over scene graph. In *2019 First International Conference on Graph Computing (GC)*, pages 45–50. IEEE.

Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2020a. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. *arXiv preprint arXiv:2012.15409*.

Yiming Li, Xiaoshan Yang, and Changsheng Xu. 2020b. Structured neural motifs: Scene graph parsing via enhanced context. In *International Conference on Multimedia Modeling*, pages 175–188. Springer.

Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. 2019a. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Li Su, and Qingming Huang. 2019b. Knowledge-guided pairwise reconstruction network for weakly supervised referring expression grounding. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pages 539–547. ACM.

Yongfei Liu, Bo Wan, Xiaodan Zhu, and Xuming He. 2020. Learning cross-modal context graph for visual grounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11645–11652.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In

*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446.

Chih-Yao Ma, Asim Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. Graf. 2018. Attend and interact: Higher-order object interactions for video understanding. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6790–6800.

Andres Mafla, Sounak Dey, Ali Furkan Biten, Lluis Gomez, and Dimosthenis Karatzas. 2021. Multimodal reasoning graph for scene-text based fine-grained image classification and retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4023–4033.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. 2016. Modeling context between objects for referring expression understanding. In *Computer Vision – ECCV 2016*, pages 792–807, Cham. Springer International Publishing.

Bryan A. Plummer, Arun Mallya, Christopher M. Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. 2017. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Aditya Ramesh, M. Pavlov, G. Goh, Scott Gray, Chelsea Voss, A. Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. 2016. Grounding of textual phrases in images by reconstruction. In *Computer Vision – ECCV 2016*, pages 817–834, Cham. Springer International Publishing.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

L. Wang, Y. Li, J. Huang, and S. Lazebnik. 2019. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407.

Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. 2019. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. 2020a. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1508–1517.

Tianming Wang, Xiaojun Wan, and Hanqi Jin. 2020b. Amr-to-text generation with graph transformer. *Transactions of the Association for Computational Linguistics*, 8:19–33.

Weitao Wang, Ruyang Liu, Meng Wang, Sen Wang, Xiaojun Chang, and Yang Chen. 2020c. Memory-based network for scene graph with unbalanced relations. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2400–2408.

Y. Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, G. Kurian, Nishant Patil, W. Wang, C. Young, J. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, G. S. Corrado, Macduff Hughes, and J. Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Pengfei Xu, Xiaojun Chang, Ling Guo, Po-Yao Huang, Xiaojiang Chen, and Alexander G Hauptmann. 2020. A survey of scene graph: Generation and application. *EasyChair Preprint*.

Sibei Yang, Guanbin Li, and Yizhou Yu. 2019a. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4644–4653.

Sibei Yang, Guanbin Li, and Yizhou Yu. 2020. Graph-structured referring expression reasoning in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9952–9961.

Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. 2019b. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10685–10694.

Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. 2019c. A fast and accurate one-stage approach to visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4683–4693.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7145–7154, Online. Association for Computational Linguistics.

Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Sanghyun Yoo, Young-Seok Kim, Kang Hyun Lee, Kuhwan Jeong, Junhwi Choi, Hoshik Lee, and Young Sang Choi. 2020. Graph-aware transformer: Is attention all graphs need? *arXiv preprint arXiv:2006.05213*.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840.

Cheng Zhang, Wei-Lun Chao, and Dong Xuan. 2019. An empirical study on leveraging scene graphs for visual question answering. *arXiv preprint arXiv:1907.12133*.

Hanwang Zhang, Yulei Niu, and S. Chang. 2018. Grounding referring expressions in images by variational context. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4158–4166.

Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. 2020. Comprehensive image captioning via scene graph decomposition. In *European Conference on Computer Vision*, pages 211–229. Springer.

Yiyi Zhou, Rongrong Ji, Gen Luo, Xiaoshuai Sun, Jinsong Su, Xinghao Ding, Chia wen Lin, and Qi Tian. 2019. A real-time global inference network for one-stage referring expression comprehension.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

# A Appendix

## A.1 Related Work

Visual grounding has been widely studied in the literature. We consider a broad definition and use the term to describe both phrase localization (Plummer et al., 2015; Rohrbach et al., 2016; Plummer et al., 2017; Wang et al., 2019) and referring expression comprehension (Yu et al., 2016; Hu et al., 2016; Yu et al., 2018). A common practice is to first generate candidate regions of an image, usually achieved by Fast R-CNN (Girshick, 2015; Ren et al., 2015), and then rank the regions based on a natural language query using CNN/LSTM (Yu et al., 2016; Mao et al., 2016; Nagaraja et al., 2016; Hu et al., 2016), attention mechanism (Liu et al., 2019b,a), modular network (Hu et al., 2017; Yu et al., 2018), graph model (Wang et al., 2019; Liu et al., 2020; Yang et al., 2019a; Jing et al., 2020), variational context (Zhang et al., 2018), or, similar as this work, transformer (Su et al., 2019; Lu et al., 2019; Li et al., 2020a). Recently, some works relax region proposals and propose one-stage approach that output the referred region directly from image pixels (Yang et al., 2019c; Zhou et al., 2019).

Since the release of the visual genome dataset (Krishna et al., 2017), scene graph generation has achieved remarkable progress (Zellers et al., 2018; Chen et al., 2019b; Gu et al., 2019b; Tang et al., 2020; Li et al., 2020b; Koner et al., 2020; Xu et al., 2020; Wang et al., 2020c) and, due to its abstraction of images, motivates various applications including image retrieval (Johnson et al., 2015; Wang et al., 2020a; Mafla et al., 2021), image generation (Johnson et al., 2018), caption generation (Yao et al., 2018; Yang et al., 2019b; Gu et al., 2019a; Zhong et al., 2020), and visual question answering (Li et al., 2019; Lee et al., 2019; Zhang et al., 2019; Cadene et al., 2019; Hildebrandt et al., 2020). This work differs from the literature in the sense that we utilize the transformer (Vaswani et al., 2017; Devlin et al., 2019) to encode scene graph for the visual grounding task. Compared to VL-BERT (Su et al., 2019) and ViLBERT (Lu et al., 2019), metaBERT processes a single object at a time and utilize its scene graph with graph mask. Some works modify the transformer to attend on partial inputs (Jiang et al., 2020; Beltagy et al., 2020; Guo et al., 2019; Zaheer et al., 2020; Kitaev et al., 2020) and encode structured graph (Wang et al., 2020b; Yao et al., 2020; Chen et al., 2019a). In addition, ActionBERT leverages the view hierarchies of mobile applications for UI component retrieval and icon prediction (He et al., 2020). ETC splits hierarchical input into global tokens and local tokens (Ainslie et al., 2020). (Cai and Lam, 2020) update the attention score by adding the relation embedding to the node embedding. (Yoo et al., 2020) scale the product of the attention key and query through the relation embedding.

### A.2 Implementation Details

We keep queries as given in both original datasets. Since the Visual7w Pointing dataset is a subset of Visual Genome, we map each candidate region of Visual7w Pointing to the object of Visual Genome with the highest IoU, and retrieve the *human annotated scene graph*, including attributes, categories, and relationships, for those regions. We discard samples where the IoU is less than a threshold (70%) — these constitute less than 0.1% of the dataset. For the RefCOCO+ dataset, we directly apply the bounding box proposals provided in (Yu et al., 2018), which use a Mask R-CNN (He et al., 2017) pretrained on the COCO dataset (Lin et al., 2014). Notably, MAttNet, VL-BERT, and ViL-BERT utilize the same proposals to report the result.

We initialize metaBERT with pre-trained BERT model. The size of metaBERT aligns with BERT base, which has a hidden size of 762, 12 heads per layer, and a total of 12 layers. Following (Jiang et al., 2020; Ainslie et al., 2020), we apply graph mask on six heads instead of all of them to mix the local attention and the global attention. We also explored adding graph mask on all heads but observed a worse result. Different from VL-BERT and ViL-BERT, we omit pre-training on the conceptual caption dataset. For generated scene graphs, we include top-3 predictions for the name and attribute of the object and the relationship between object pairs. we use Faster R-CNN pretrained on the Visual Genome dataset for visual features. The visual embedding has 2048 dimensions, and we apply a linear layer to reduce the dimension to 762.

We conduct experiments on 8 Tesla V100 GPUs with a total batch size of 64. We train 4 epochs for the visual7w-pointing dataset and 6 epochs for the RefCOCO+ dataset, because the latter contains more object candidates per image. We use the Adam optimizer with a initial learning rate of 1e-4. We also apply a linear learning rate scheduler with warmups.

### A.3 Sensitivity on the Quality of Scene Graph

We investigate the influence of the quality of scene graph on the visual7w-pointing dataset. Specially, we consider two scenarios: 1. the scene graph shows no relationship (only attributes and categories remain); 2. the scene graph includes a mixture of gold and false positive relationships. The second case is achieved by replacing annotated scene graph with generated ones. We do not manipulate the target object because doing so should guarantee a wrong prediction.

Table 2 presents the results. According to the table, the performance drops 2% when gold relationships are not contained and 5% when false positive predictions are included. The observation also explains the low accuracy of metaBERT w/o visual features on RefCOCO+ in Table 1: the quality and difficulty of generated scene graph holds back the performance. Remarkably, visual7w-pointing contains annotated scene graph, and each visual object has on average $1.5 \pm 1.5$ relationships, whereas in RefCOCO+ our generated scene graph has $16.3 \pm 12.1$ relationships per object. However,

| | V7w-pointing | | | |
|---|---|---|---|---|
| | Val | | Test | |
| metaBERT | 81.07 | | 80.39 | |
| metaBERT w/o any relationships | 79.14 | -2.3% | 78.73 | -2.0% |
| metaBERT w/ false relationships | 76.85 | -5.2% | 76.20 | -5.2% |

Table 2: The sensitivity on the quality of the scene graph. MetaBERT shows worse performance when no relationship is used or there are false-positive relationships.

| | V7w-pointing | | | |
|---|---|---|---|---|
| | Val | | Test | |
| metaBERT | 81.07 | | 80.39 | |
| metaBERT w/o segment emb. | 80.11 | -1.2% | 79.80 | -0.7% |
| metaBERT w/o position emb. | 78.98 | -2.6% | 78.54 | -2.3% |
| metaBERT w/o token type emb. | 80.16 | -1.1% | 79.87 | -0.7% |
| metaBERT w/o graph mask | 80.77 | -0.4% | 80.44 | +0.1% |

Table 3: The ablation study. Segment embedding, position embedding, and token type embedding contribute to a better performance, whereas the graph mask does not change the result significantly.

we can improve the performance on RefCOCO+ by incorporating visual features.

### A.4 Ablation Study

To show the efficacy of the proposed metaBERT, we conduct ablation experiments by excluding each component. We show the results in Table 3. On the visual7w-pointing dataset, metaBERT without segment embedding, position embedding, or token type embedding hurt the performance, while removing position embedding leads to the most degradation. Moreover, the usage of the graph mask does not change the result significantly. There might be two possible reasons: 1. other embeddings might already address the structure of scene graph; 2. we instantiate each training example with only one target object, and it partially relaxes the assumption of encoding the structure. Is there another way to encode the structured scene graph via metaBERT? This remains an open question, and we will explore it in the short future.

### A.5 Attention Statistics

We present the distribution of the normalized attention score at the last layer that the “[CLS]" token has received. The purpose of the visualization is to show how the graph mask change the attention distribution rather than to express the “goodness" of it. We conduct the Wilcoxon signed-rank test to check whether the graph mask yields a different attention distribution compared to not using it. Since we applied the graph mask to heads 7-12, those heads present a different pattern of attention scores. Although the difference of heads 1-6 are not as visible as heads 7-12, the Wilcoxon signed-rank test suggests that the two attention distributions are significantly different across all heads.

### A.6 Qualitative Results

We present examples of Visual7w Pointing to analyze metaBERT without visual features in Figure 4. The top four examples are correctly answered by metaBERT, and they indicate that the scene graph could be valuable for grounding because it contains the object or the relationship that the query is referring to. The bottom four examples demonstrate where metaBERT makes mistakes. We observe that: (i). in the 5th example metaBERT misclassifies toppings on pizza as the mentioned food and ignores the jar filled with peppers; (ii). in the 6th example metaBERT might capture the attribute “orange" but fail to understand facing front; (iii). in the 7th example metaBERT is confused by the object “black shorts" that the player rather than the ball boy wearing; (iv). in the last example metaBERT might fail on encoding the location of the object, which is also missed from the scene graph.

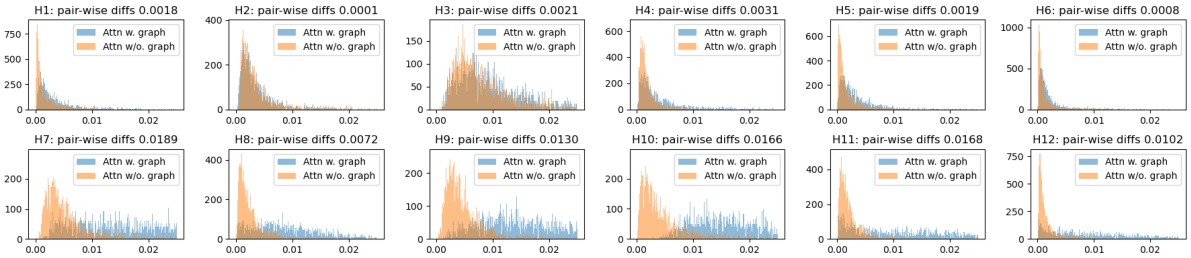
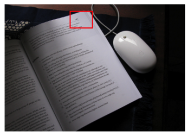



Figure 3: The distribution of the normalized attention score that the "[CLS]" token has received at the last layer. Attn w. Mask and Attn w/o. Mask mean that the attention scores are from models with the graph mask and without graph mask, respectively. The pair-wise difference indicates the attention score with graph mask is higher than the score without graph mask.

Q: which box frames the black pillow?
A & P: blue gold trim [SEP] long black pillow with [SEP] on long black pillow [SEP]

Q: which object is on the keyboard ?
A & P: grass [SEP] on black green keyboard [SEP]

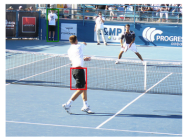
Q: which person with short hair sits and is smiling?
A & P: short hair [SEP] sitting smiling woman girl wears [SEP] on sitting smiling woman girl [SEP]



Q: which part of the book shows the page number ?
A & P: corner [SEP] number in [SEP]

Q: which food item makes other foods spicier?
A: glass clear filled jar [SEP] flakes crushed pepper in [SEP] metal screw on top [SEP] of flakes crushed pepper [SEP] filled with flakes crushed pepper [SEP]
P: toppings [SEP] on slice large pizza [SEP]

Q: which is boy in orange facing front?
A: teen [SEP] coach man stand with [SEP]
P: orange jersey [SEP] walking teen wear [SEP]



Q: which ball boy has on black shorts?
A: person [SEP] standing off blue tennis court [SEP]
P: black shorts [SEP] playing player wearing [SEP]

Q: which pair of skis are on the upper left edge?
A: skier [SEP] with poles [SEP] with snow covered grey different color boots [SEP]
P: skis [SEP] close person with [SEP] have back part [SEP]

Figure 4: We show qualitative examples that metaBERT predicts correctly (top four) and wrongly (bottom four). A means the correct answer. P means the prediction of metaBERT. The red box in the image is the prediction of metaBERT, and the green box is the ground truth.