Appendix of Multi-Modal Dynamic Graph Transformer for Visual Grounding

1. One-step Evaluate-and-rank Matching Architecture

In the main content of our paper, we summarize the proposed state-of-the-art visual grounding (VG) methods [4, 6, 7, 18, 21, 25, 32, 33, 44, 49] into one-step evaluate-and-rank matching architecture. It is desired to emphasize that this is a rough classification as these works, such as [7], utilize box regression and box refinement, to predict the bounding box for the text query. Besides, works such as [6, 21] potentially leveraged the idea of progressive learning to adjust and refine the attention scores to predict tight bounding boxes for the text query. However, we argue that mechanisms, such as box regression behave as a tool to obtain matched region-query pairs. Compared with our work that gradually searches the target bounding boxes from scratch, i.e., some randomly initialized boxes, these methods inherently belong to a matching architecture instead of a searching architecture. More importantly, they generally perform the one-step evaluate-and-rank process to predict the target box once without continuously fine-tuning the bounding boxes in the learning process. Particularly in works [6, 21], the progressive learning idea was adopted to obtain an accurate attention assignment [6] or to discover the full latent alignments [21]. Nevertheless, the box prediction is performed once based on the learned attention scores to obtain unadjustable region-text matching, making them follow the one-step evaluate-and-rank matching architecture.

2. Effectiveness of Progressive Learning in Visual Grounding

Progressive learning (PL) is the iterative-based method in which the main idea is to complete one task by approaching the objective gradually. The primary characteristic of PL is to shrink the solution search field in each step by sufficiently modeling the detailed information, thereby achieving more accurate results than the one-step manner. Thus, many computer vision works introduced the PL to complete image recognition [12], visual attribute prediction [37], and visual reasoning [20]. Then, the benefits of PL used in visual grounding [5, 6, 9, 11, 31, 38, 43, 48] are attributed to three aspects. Our M-DGT integrates these three advantages of PL into an end-to-end trainable framework through transformers and multi-modal graph neural networks.

Firstly, visual grounding is a complex task in which the model needs to obtain specific bounding boxes that match the text input from an image containing infinite regions. The PL can achieve the idea of divide and conquer, thereby converting this task into many easy-to-solve subproblems. Then, the learning of each sub-problem can benefit from sufficient information and more straightforward objectives. Many works have proven that this can lead to more accurate results with a lightweight model. For example, compared with conventional methods [32, 34, 39] that require an external region proposal model to obtain region candidates, the PL-based method [9] can achieve better performance with lower time-consumption.

Secondly, PL can effectively capture the required target information from the original input with redundant, irrelevant, and interference information. More specifically, unlike the idea that processes the image as a whole, PL can gradually detect and filter out irrelated regions to shrink the attention to target ones. Each stage of PL is able to utilize relatively adequate information to model a simpler objective, leading to a stable, efficient, and accurate learning process. For example, in the work [6], starting from a rough semantic that is easy to be located, the proposed model can refine the semantic expression in each step to reduce the scope of localization information. The idea proposed in the work [20] presents an iterative inference pipeline to continuously adjust the attention between the subject and the object to localize two entities.

Thirdly, in the architecture with a one-step prediction manner, many regions are estimated simultaneously without further adjustments. This always requires a well-prepared input such that there are small deviations between candidates and targets. Otherwise, this leads to suboptimal and low-quality results. However, by working directly on the whole information that contains target regions, PL-based methods can continuously adjust the learning process to reach these objections without missing useful parts. Specifically, due to the sustainable optimization property of PL, the model can produce tighter bounding boxes through fastidious learning. For example, as discussed in the work [12], through recursive learning to discriminate regional attention and region-based feature representation, fine-grained image

recognition can be achieved.

3. Graph Backbone Network

Graph neural networks (GNNs) [2, 3, 15, 17, 30, 36, 42] have been extensively utilized in many domains. GNNs are connectionist models that capture the dependence of knowledge via message passing between nodes of the graph. Unlike standard neural networks, GNNs retain a state that can represent information from their neighborhood with arbitrary depth. Therefore, graph structure can organize unstructured information directly and then learn it efficiently.

As shown by existing works [18, 25, 40, 41] in visual grounding, the graph is a natural way to model multi-modal semantic relations. Working on the multi-modal information built with the graph makes the learning process effectively model non-linear relations to improve the performance. For instance, as pointed by the work [25], there are inherent correspondences between the language graph and the visual graph. Thus, converting the visual grounding into the multi-modal graph matching can directly exploit the phrase and visual object's inter-relations to boost performance. In addition, visual reasoning behaves as a major requirement in visual grounding. The effectiveness of graph structure in visual reasoning is proven by the work [18].

Our work is motivated by the achievements of applying the graph structure in visual grounding. However, instead of consistently trapping in the one-step matching architecture, our work pioneered to propose the search-based visual grounding achieved by a dynamic graph method. Thus, visual grounding is reformulated into a progressively optimized visual semantic alignment process with the graph as the backbone network. Besides, we embed the text query and visual information to each node to instantiate the multi-modal graph. Then, in each search step, the spatial information and inter-modal relations are modeled by this multi-modal graph to gradually locate target regions, thereby achieving progressive learning in visual grounding.

Fig. 1 presents an instance of how to construct the graph structure in our work. Firstly, similar to the method such as faster-RCNN that generates anchors based on the feature map grid from CNN network, our work generates anchors with a single scale 128×128, the stride 128, and constant aspect ratios $(1, 1)$. These anchors are used as the initialization bounding boxes in our M-DGT. Then, these boxes are converted to nodes of the graph by utilizing the box center as the spatial position of the node. Finally, the graph constructs edges by connecting each node with its directly connected neighbors.

There are several advantages of using the graph structure as the backbone network in our method. The major challenge in search-based visual grounding with progressive learning is to pass the information from local to global efficiently. The graph provides an effective way to achieve

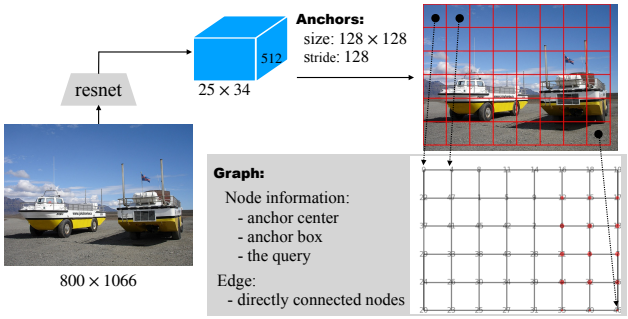

Figure 1. An instance of how to generate the initialization boxes and the corresponding graph.

this by forwarding the information through the multi-hop path. Then, the learning of each node depends on its connected neighbors and other related nodes. As the node can selectively learn from nodes with approximate features, this learning process can be stable and convergent. Otherwise, transforming a node to target ones can be damaged if there are large deviations. In addition, the graph is an effective way to present spatial information, especially the relative positions of nodes. This also ensures that the node can only be transformed based on the information passed from its similar neighbors.

4. Dynamic Graph in M-DGT

Our search-based visual grounding builds upon the idea of combing progressive learning with graph transformation. As each node in the graph corresponds to one box in the image, adjusting the boxes to the ground truth is equivalent to the transformation of the graph. Thus, the spatial and multi-modal information of nodes in the graph will change over iteration. Besides, the graph layout, including nodes and edges, is also refined to facilitate visual-phrase reasoning during the transformation. Therefore, the graph is gradually shrunk to the target layout in which the nodes correspond to regions matched with the query.

Therefore, the whole learning process can be presented as a series of dynamic graphs. As shown in Fig. 2, starting from the initialization graph, our proposed multi-modal dynamic graph transformer (M-DGT) transforms the graph to approach the ground truth regions progressively. During this learning process, M-DGT constructs the dynamic graph.

Another critical point of M-DGT is that edges and nodes of the graph are pruned in each iteration step. As shown by the second row of the right sub-figure in Fig. 2, compared with the initialization graph when $o = 1$, there are fewer nodes and edges with the increase of the iteration number o. Correspondingly, in the first row of the right sub-figure, we present the graph transformation without pruning nodes

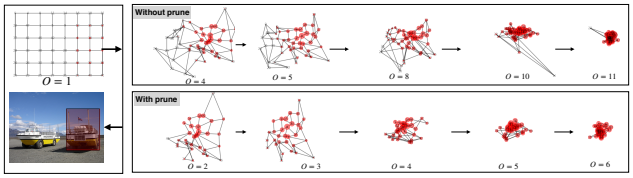

Figure 2. An instance of the dynamic graph that gradually shrink to the target layout corresponding to the ground truth regions. In the left sub-figure, the first row shows the initialization graph, and the second two presents the final predicted bounding boxes drawn by the red rectangle. The right sub-figure presents the graph transformation in each iteration. The first row presents the dynamic graph without using *graph transformer*. In contrast, the second two presents the results obtained by utilizing the *graph transformer*.

and edges. In this case, more iterations and computations are required to reach the ground-truth regions. Besides, as can be seen in our experiment shown in subsection 5.3 of main content, without applying the *graph transformer*, our M-DGT spends a long inference time to obtain a relatively low accuracy in both Flickr30k Entities [34] and RefCOCO datasets [47].

5. Limitations of M-DGT

The limitations of M-DGT mainly exist in three aspects. Firstly, M-DGT is significantly difficult to be trained. Moreover, it is susceptible to parameters initialization. We particularly rely on Optuna [1] to search for effective hyperparameters. Meanwhile, sometimes the model learning does not converge during the training process, so we need to restart the training. Secondly, the non-backtracking search method of M-DGT makes it difficult to relocate accurately once the output bounding boxes leave the target regions during iterations. Besides, the situation of leaving the target regions damages the training and causes the accumulation of errors. This limitation can specifically be observed in the Fig. 11 and Fig. 12. Thirdly, as shown by Fig. 9, M-DGT focuses on locating large objects and tends to ignore small ones. The imbalanced positioning problem caused by the algorithm design has not yet been solved with a good solution.

6. Implementation Details

This section describes the implementation details for our proposed multi-modal dynamic graph transformer (M-DGT) and the conducted experiments.

6.1. Node Transformation as 2D Transformation

As described in the main content, each node in the graph corresponds to one box in the image. The *node transformer*

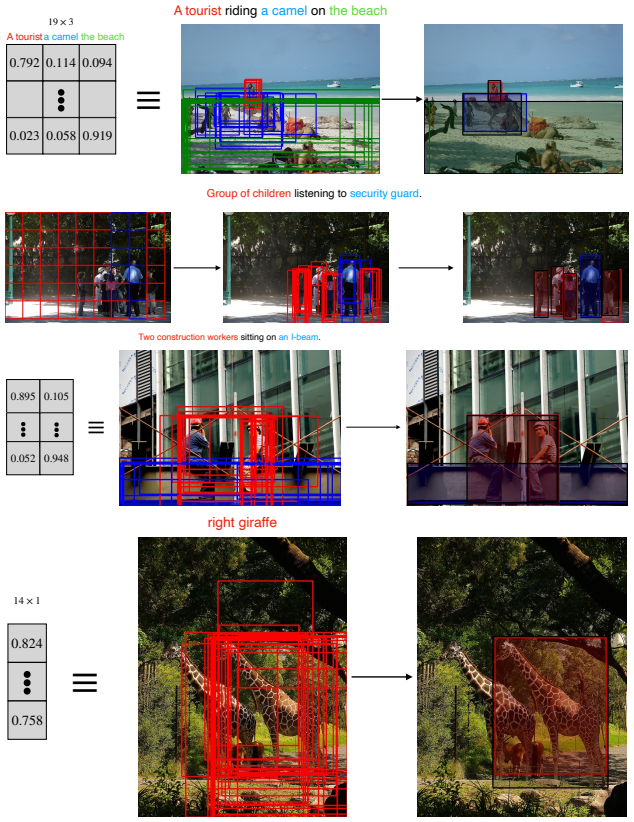

Figure 3. Four instances about how to generate final bounding boxes from the predicted boxes.

of the M-DGT essentially operates the 2D transformation on bounding boxes to approach ground-truth regions. To achieve this, we regard the box as the rectangle in 2D space and define the box transformation as the affine transforms with translation and scaling transformation in 2D space. Thus, this transformation operation can be formulated by matrix multiplication in homogeneous coordinates. For one coordinate (x, y), the corresponding transformation matrix is as follows:

$$\begin{bmatrix} s_1 & 0 & r_1 \\ 0 & s_2 & r_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{1}$$

where the r_1, r_2, s_1, s_2 are the output of the *node transformer*.

6.2. Obtain Final Bounding Boxes

Our M-DGT outputs a matrix with the shape $N \times P$ where N and P are the numbers of nodes in the graph and phrases in the query, respectively. The subfigure in the first row of Fig. 3 presents an example of the produced matrix.

In summary, we operate two easy-to-compute steps to decide the bounding boxes.

Firstly, each row of the matrix demonstrates the matching score between one node and the query phrases. Then, we select the node-phrase pair with the maximum matching score in each matrix row. By doing this, we know which query phrase corresponds to the bounding box of the node. This leads to the results shown in the second column of Fig. 3, where each bounding box is colored the same as the corresponding phrase. Secondly, applying DIOU [51] to remove the abundant bounding boxes achieves the best performance for M-DGT. Thus, we obtain the final bounding boxes as shown in the third column of Fig. 3.

Based on this method, we present three phrase localization instances from the Flickr30k Entities dataset and one referring expressions instance from the RefCOCO dataset in Fig. 3. These examples also demonstrate that our M-DGT can handle the one-to-one matching and one-to-many matching challenges. In the one-to-one example shown in the first row, one phrase corresponds to one bounding box. In the one-to-many example shown in the second and third row, one phrase corresponds to multiple bounding boxes. Especially in the third row, one final bounding box is covered by another one.

6.3. Training Settings

This subsection mainly describes the detailed information of datasets and architecture settings utilized in our experiments.

Flickr30k Entities. Flickr30k Entities [34] is phrase localization dataset that augments the original Flickr30K [45] with region-phrase correspondence annotations. It links 31,783 images in Flickr30K [45] with 427K referred entities. We use the common splits used in previous works [7, 32, 34] to obtain 29783, 1000, and 1000 images for train, validation, and test, respectively. There are mainly two challenges in this dataset. Firstly, each image contains dense bounding boxes that can overlap with each other. Secondly, one phrase may correspond to multiple bounding boxes, i.e., one-to-many. Our M-DGT can effectively handle these two challenges.

RefCOCO. Three referring expression datasets, including RefCOCO [47], RefCOCO+ [47], and RefCOCOg [27], are based on the images of the COCO dataset [23]. Then, a piece of text is used as a referring expression to describe a unique object in an image. Thus natural language referring expressions are used to describe objects in images. Specifically, no restrictions are required on the type of language text used in the referring expressions of the RefCOCO [47] dataset. Expressions of the RefCOCO+ [47] are constrained on the purely appearance-based description following a computer vision-based perspective. Thus, no location-based description is allowed in RefCOCO+ [47] dataset. As the annotations of RefCOCOg [27] are collected in a non-interactive setting, it has a more detailed description of objects compared to RefCOCO [47]. Thus, RefCOCOg [27] has 8.4 words per expression compared with a short 3.5 words in the RefCOCO [47]. There are $19,994$ and $19,992$ images in RefCOCO [47] and RefCOCO+ [47] datasets, respectively. Specifically, RefCOCO [47] consists of $142,209$ refer expressions for $50,000$ objects while RefCOCO+ [47] has $141,564$ expressions for $49,856$ objects. Also, following the official splits of RefCOCO and RefCOCO+ [47], the samplers are split into train, val, testA, and testB. The testA and testB have different focuses in evaluation. The testA set has multiple persons, while testB has multiple objects from other categories. RefCOCOg [27] has $25,799$ images with $49,856$ referred objects and expressions. We utilized the commonly used split protocol RefCOCOg-google [28] in our experiment.

ReferItGame. ReferItGame [19] dataset contains 130,525 expressions for referring to 96,654 objects in 19,894 images of natural scenes collected from the SAIAPR-12 dataset [10]. Following the settings in the work [7], we split the samples into three subsets. For the train set, we have $54,127$ referring expressions. We have $5,842$ and $60,103$ referring expressions for the test set and validation set, respectively.

Architecture. The ResNet18, ResNet50, and ResNet-101 [14] are utilized to extract the visual features. For the image with shape $H \times W \times 3$, we utilize the image preprocessing method in the work [35] to prepare the input for our M-DGT. Then, the backbone feature map is the output $H' \times W' \times C$ of the final convolution layer. BERT [8] is used for the language embedding, while The official tools provided in [26] are used to extract inter-phrase dependencies. For the initialization of boxes, the boxes with size 128×128 and the stride 128 are generated to cover the image. Then, for each box (i.e., the anchor) in the image, the corresponding feature is cropped from the backbone feature map and then is processed by the RoiAlign [13] to generate a $5 \times 5 \times C$ feature. The common space dimension is 512. Fig. 4 presents the detailed information of the architecture and the hyper-parameters of M-DGT in our experiment.

Additional details. (1) As we illustrated in Section 3 of the appendix, we followed the faster-RCNN to rescale the image with a minimum size of 800. The CNN backbones are initialized with the weights pre-trained on the MSCOCO dataset following previous works to make a fair comparison. (2) The ground-truth annotations are only utilized in the training stage as our M-DGT is a supervised model. (3) As we mentioned in Section 6 of the appendix, M-DGT is hard to train potentially because of its complexity. It costs approximately 21 hours to train based on one Tesla P100 GPU. The inference time listed in Table 1 is computed by performing tests on NVIDIA 1080TI as previous one-stage works [44] for a fair comparison. (4) The initialization

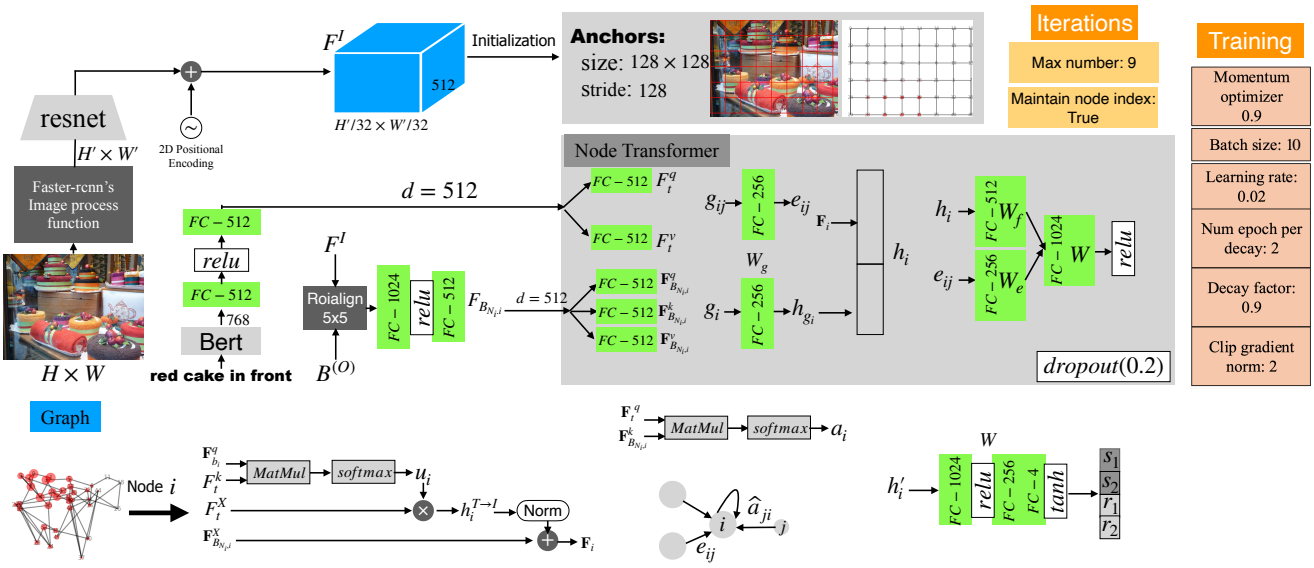

Figure 4. The detailed configurations of our experiments.

boxes with size 128×128 and the stride 128 are generated to cover the image. The 48 is the average number of initial boxes generated for test images.

Language embedding. In the experiments, we compare the M-DGT performance of using three types of language embedding methods, including Word2vec [29], LSTM, and Bert [8]. Specifically, for the Word2vec, the phrase feature is obtained by averaging the features of words. For the LSTM, each word is presented as a one-hot embedding, and then a pre-trained bi-directional LSTM is applied to encode the expression. For the Bert, we obtain the phrase feature vector by weighted sum word features embedded by the Bert [8]. For the language embedding part of our M-DGT, we directly utilize the pre-trained models without training them during the learning process.

7. Supplement Quantitative Results

We first present the experimental results on the ReferItGame [19] dataset. Then, under the IoU threshold ranges from 0.3 to 1, the accuracy of our M-DGT is compared with alternative leading visual grounding methods to show that M-DGT can get tighter bounding boxes. Finally, we provide detailed experimental results of M-DGT on Flickr30k Entities to demonstrate that our M-DGT can effectively address the one-to-many issues in the phrase grounding.

7.1. Supplementary on ReferItGame

Table 1 presents the accuracy of leading methods on the ReferItGame dataset with the IoU threshold 0.5. Our M-DGT achieves the best performance as compared with all other methods. With the ResNet-50 backbone, M-DGT obtains 72.41% accuracy that outperforms the current best method TransVG but utilizes 5.77ms less inference time than it. After using a stronger ResNet-101 backbone, the performance of M-DGT boosts to 73.63%, which is 2.98% higher than the TransVG. Nevertheless, our method still maintains a competitive inference time that is 31 ms higher than the TransVG. Compared with the fastest method RCCF, M-DGT spends 68ms more inference time but achieves 9.84% higher accuracy.

Table 1. Comparisons with state-of-the-art methods on the test set of ReferItGame [19] in terms of top-1 accuracy (%) with IOU threshold 0.5.

Method	Visual Backbone	Region Proposals	Language Embedding	Acc@0.5	Time (ms)
CMN[16]	VGG16	Faster-RCNN N=300	LSTM	28.33	-
VC[50]	VGG16	Faster-RCNN N=200	Word2vec, FV	31.13	-
DDPN[49]	ResNet101	Faster R-CNN N=100	LSTM	63.00	196
CITE[32]	ResNet101	Faster R-CNN N=200	Word2vec, FV	35.07	184
Two-branch[39]	ResNet101	Edgebox N=200	Word2vec, FV	34.54	-
MAttNet [46]	ResNet101	Faster R-CNN N=200	LSTM	29.04	314
OneStageVG [44]	DarkNet53	None	Bert	60.67	38
RCCF [22]	DLA-34	None	LSTM	63.79	**25**
ReSCLarge [43]	DarkNet53	None	Bert	64.60	36
TransVG [7]	ResNet-50	None	Bert	69.76	61.77
TransVG [7]	ResNet-101	None	Bert	70.73	61.77
M-DGT_FV	ResNet50	None	Word2vec, FV	72.96	61
M-DGT_LSTM	ResNet50	None	LSTM	71.3	68
M-DGT_Bert	ResNet18	None	Bert	71.04	56
M-DGT_Bert	ResNet50	None	Bert	72.41	80
M-DGT_Bert	ResNet101	None	Bert	**73.63** (↑ 2.9%)	93

7.2. Bounding Boxes Tightness Comparison

With the idea of progressive semantic search, our framework has the ability of meticulous regional adjustment that leads to the tighter predicted regions for the query, which is reflected in maintaining high accuracy under high IoU

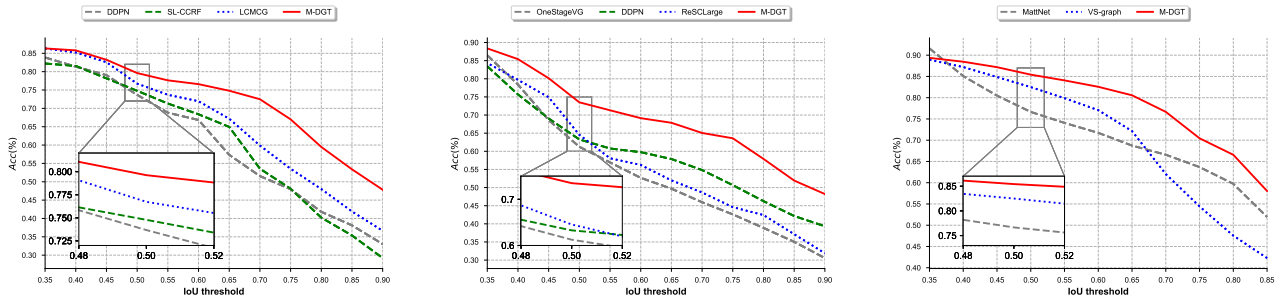

Figure 5. The accuracy $Acc(\%)$ of methods under IoU threshold range from 0.35 to 0.9. (a) presents the comparison results of DDPN [49], SL-CCRF [24], LCMCG [25], and our M-DGT. (b) presents the comparison results of OneStageVG [44], DDPN [49], and ReSCLarge [43]. (c) presents the comparison results of MAttNet [46] and VS-graph [18].

thresholds. As shown in Fig. 5, when the IoU threshold is greater than 0.5, our proposed M-DGT can still maintain high accuracy without a significant drop in the performance until the IoU threshold reaches 0.7, 0.75, and 0.65 on Flickr30K Entities [34], ReferItGame [19], and RefCOCO [47] datasets, respectively. Besides, compared with other alternative state-of-the-art methods, M-DGT obtains the highest grounding boxes under IoU threshold 0.9. And, the corresponding accuracy is often 9% higher than the second-best method. These experimental results solid prove that the utilization of the progressive search based on the graph leads to tighter bounding boxes for the query. The insight is that M-DGT can continuously optimize the bounding boxes to approach the ground truth regions. Also, each box can receive sufficient information from other boxes through the graph structure to make the visual reasoning. This is also proven by the qualitative results in Fig. 10. M-DGT builds the first graph based on the initialization boxes that cover the image. Then, in the connected path, each node in the graph can exploit sufficient spatial and multi-modal information from all other nodes to adjust its bounding boxes, thereby gradually approaching the target regions. During this process, as shown in Fig. 10, we can obtain dynamic graphs that progressively shrink to the target layout in which the nodes correspond to the tight bounding boxes.

7.3. One-to-many Challenge

In the phrase grounding task of Flickr30k Entities dataset [34], the one-to-many challenge is that one phrase corresponds to larger than two bounding boxes. In existing works, the authors count one correct prediction once the precited bounding box matches one of the ground truth boxes. In this way, these methods ignore the one-to-many challenge. However, as our M-DGT models search-based progressive learning with multi-modal graph transformer, it naturally has a strong ability to address the one-to-many challenge. As shown by Fig. 6, M-DGT obtains correct predicted bounding boxes in most cases. Mainly when each

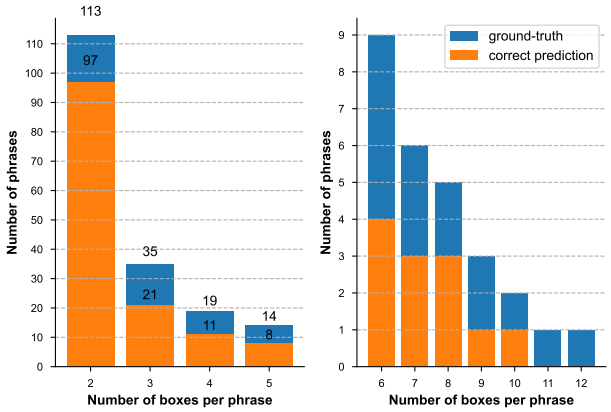

Figure 6. We obtained the detailed performance of M-DGT for the one-to-many challenge on the Flickr30k Entities test set that contains 1000 images. The ground truth is the given boxes for each phrase. Then, for each phrase, we count the correct prediction once predicted boxes correctly match the ground truth boxes. The IoU score of each predicted box and the corresponding target box exceeds 0.5.

phrase corresponds to two bounding boxes, the accuracy is 85.84%. We present two instances in Fig. 7. Besides, M-DGT obtains around 50% accuracy when the number of bounding boxes per phrase ranges from 6 to 10. Two corresponding instances are presented in Fig. 8.

8. Supplement Qualitative Results

We present more qualitative results from the Flickr30k Entities [34], RefCOCO [47], RefCOCO+ [47], and RefCOCOg [27] datasets.

Fig. 10 shows some success instances obtained by M-DGT in the Flickr30k Entities dataset. Fig. 9 shows some instances that have errors obtained by M-DGT in the Flickr30k Entities dataset. Fig. 11 and Fig. 12 show some typical mistakes made by M-DGT in the RefCOCO [47] and

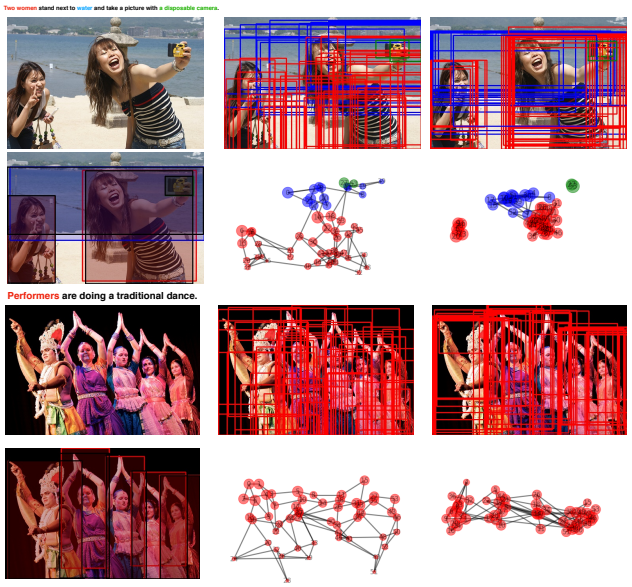

Figure 7. The results obtained by M-DGT on two instances of the Flickr30k Entities dataset. For each subfigure, the first row in the first column presents the original image and the predicted bounding boxes. The color for ground truth boxes is black, while the colors of predicted boxes are consistent with the corresponding query phrases. The first row in the second and third columns presents the output bounding boxes in the learning process. The second row of these two columns presents the corresponding graphs.

RefCOCO+ [47], respectively. Then, Fig. 13 presents two inaccurate instances of M-DGT on RefCOCOg [27] dataset.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] James Atwood and Don Towsley. Diffusion-convolutional neural networks. *arXiv preprint arXiv:1511.02136*, 2015.

[3] Jianfei Chen, Jun Zhu, and Le Song. Stochastic training of graph convolutional networks with variance reduction. *arXiv preprint arXiv:1710.10568*, 2017.

[4] Kan Chen, Rama Kovvuri, and Ram Nevatia. Query-guided regression network with context policy for phrase grounding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 824–832, 2017.

[5] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7239–7248, 2018.

[6] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7746–7755, 2018.

[7] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1769–1779, 2021.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[9] Pelin Dogan, Leonid Sigal, and Markus Gross. Neural sequential phrase grounding (seqground). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4175–4184, 2019.

[10] Hugo Jair Escalante, Carlos A Hernández, Jesus A Gonzalez, Aurelio López-López, Manuel Montes, Eduardo F Morales, L Enrique Sucar, Luis Villasenor, and Michael Grubinger. The segmented and annotated iapr tc-12 benchmark. *Computer vision and image understanding*, 114(4):419–428, 2010.

[11] Zicong Fan. *Visual grounding through iterative refinement*. PhD thesis, University of British Columbia, 2020.

[12] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[16] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1115–1124, 2017.

[17] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 5308–5317, 2016.

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

[19] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[20] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. Referring relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6867–6876, 2018.

[21] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching.

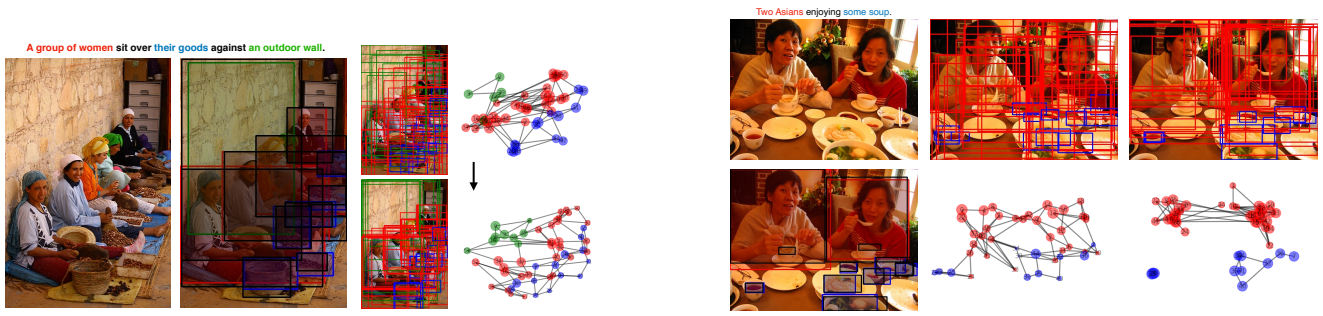

Figure 8. The results obtained by M-DGT on two more-challenging instances of the Flickr30k Entities dataset. For each subfigure, the first row in the first column presents the original image and the predicted bounding boxes. The color for ground truth boxes is black, while the colors of predicted boxes are consistent with the corresponding query phrases. The first row in the second and third columns presents the output bounding boxes in the learning process. The second row of these two columns presents the corresponding graphs.

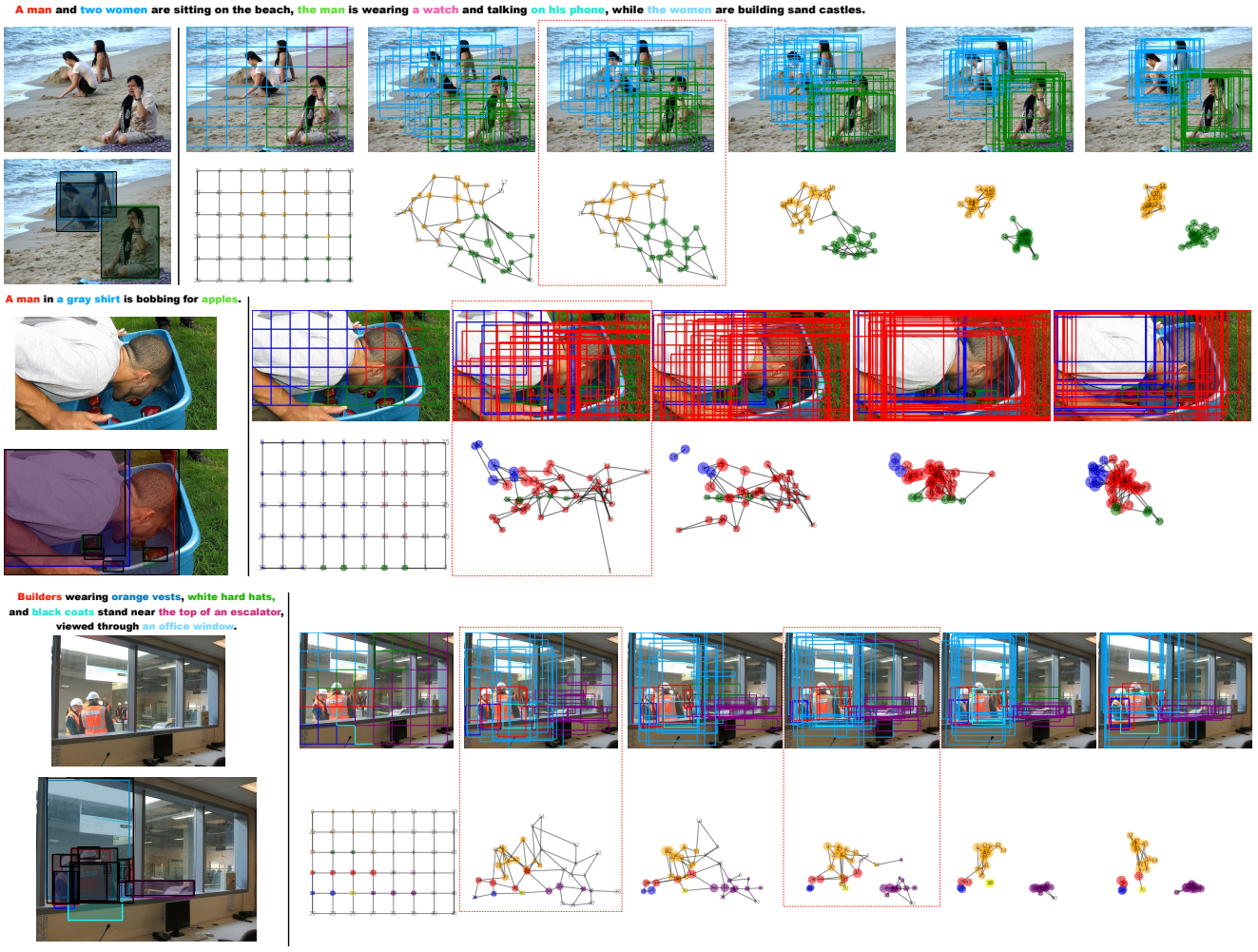

Figure 9. The results with mistakes obtained by applying the M-DGT to the challenge instances of the Flickr30k Entities dataset. The red dashed box indicates where the error occurred in the iterations.

Figure 10. The success results obtained by applying the M-DGT to the challenge instances of the Flickr30k Entities dataset.

In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.

[22] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vi-*

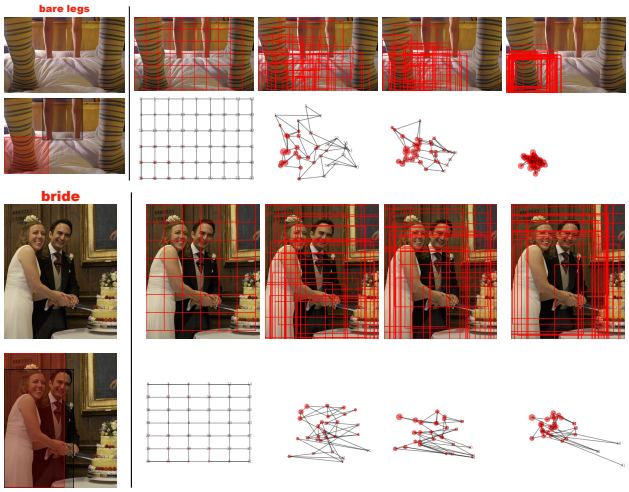

Figure 11. The results with mistakes obtained by applying the M-DGT to the challenge instances of the RefCOCO.

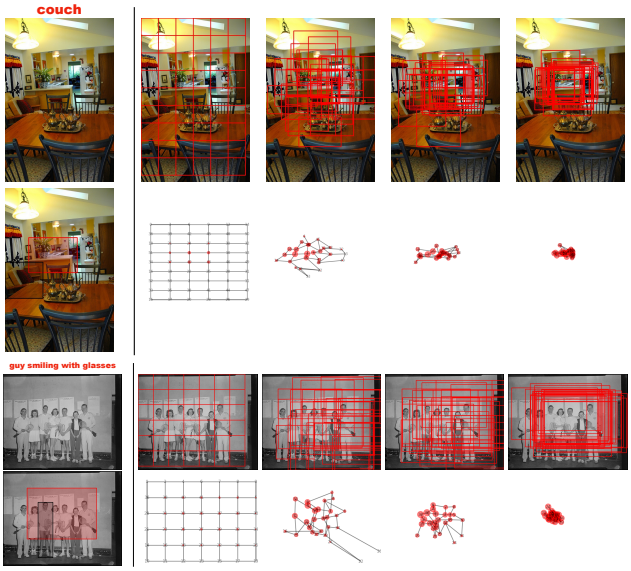

Figure 12. The results with mistakes obtained by applying the M-DGT to the challenge instances of the RefCOCO+.

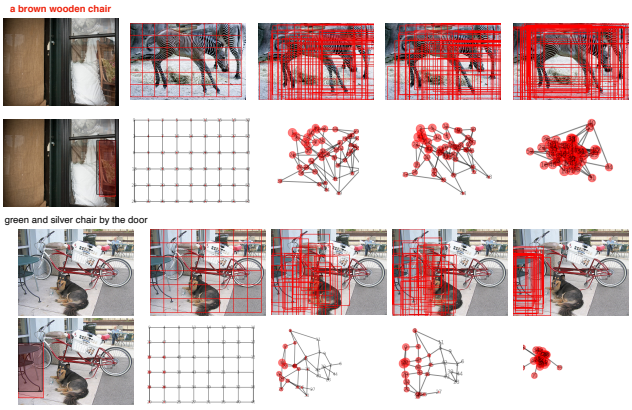

Figure 13. Inaccurate results obtained by applying M-DGT to the challenge instance of RefCOCOg.

sion and Pattern Recognition, pages 10880–10889, 2020.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[24] Jiacheng Liu and Julia Hockenmaier. Phrase grounding by soft-label chain conditional random field. In *EMNLP/IJCNLP*, 2019.

[25] Yongfei Liu, Bo Wan, Xiaodan Zhu, and Xuming He. Learning cross-modal context graph for visual grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11645–11652, 2020.

[26] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[27] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[28] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[30] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

[31] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1881–1889, 2017.

[32] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–264, 2018.

[33] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *Proceedings of the IEEE Interna-*

tional Conference on Computer Vision, pages 1928–1937, 2017.

[34] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[36] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[37] Paul Hongsuck Seo, Zhe Lin, Scott Cohen, Xiaohui Shen, and Bohyung Han. Progressive attention networks for visual attribute prediction. *arXiv preprint arXiv:1606.02393*, 2016.

[38] Mingjie Sun, Jimin Xiao, and Eng Gee Lim. Iterative shrinking for referring expression grounding using deep reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14060–14069, 2021.

[39] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2018.

[40] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1960–1968, 2019.

[41] Peixi Xiong, Huayi Zhan, Xin Wang, Baivab Sinha, and Ying Wu. Visual query answering by entity-attribute graph matching and reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8357–8366, 2019.

[42] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[43] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive sub-query construction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 387–404. Springer, 2020.

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

[45] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014.

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

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

[48] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7282–7290, 2017.

[49] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. In *IJCAI*, 2018.

[50] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4158–4166, 2018.

[51] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12993–13000, 2020.