# A Simple Baseline for Weakly-Supervised Scene Graph Generation

Jing Shi[1] Yiwu Zhong[2] Ning Xu[3] Yin Li[2] Chenliang Xu[1]
[1]University of Rochester [2]University of Wisconsin-Madison [3]Adobe Research
[1]{j.shi,chenliang.xu}@rochester.edu [2]{yzhong52,yin.li}@wisc.edu [3]nxu@adobe.com

## Abstract

*We investigate the weakly-supervised scene graph generation, which is a challenging task since no correspondence of label and object is provided. The previous work regards such correspondence as a latent variable which is iteratively updated via nested optimization of the scene graph generation objective. However, we further reduce the complexity by decoupling it into an efficient first-order graph matching module optimized via contrastive learning to obtain such correspondence, which is used to train a standard scene graph generation model. The extensive experiments show that such a simple pipeline can significantly surpass the previous state-of-the-art by more than 30% on the Visual Genome dataset, both in terms of graph matching accuracy and scene graph quality. We believe this work serves as a strong baseline for future research. Code is available at https://github.com/jshi31/WS-SGG.*

## 1. Introduction

Given an image, scene graph generation (SGG) is to generate a scene graph [17, 44], consisting of the detected objects and the possible relationships between the objects. Such abstraction mimics the structured representations of language [33, 42], facilitating various downstream visual reasoning tasks, *e.g.* VQA [38, 12, 13], image caption [46, 59], image generation [15]. However, most of the current SGG models are supervised trained with scene graph annotations, which suffered from two limitations. First, they rely on the expensive annotations of object locations and relations. Second, they are hard to generalize to out-of-domain objects or relations that are required by downstream tasks, *e.g.*, the VQA dataset, where the questions query novel objects and relations that are out of the scene graph dataset domain.

To overcome the above limitations, we investigate the weakly-supervised scene graph generation (WS-SGG) problem, demonstrated in Fig. 1. We relax the annotation requirements of SGG only to consider the *ungrounded scene graph*, composed of solely image-level object and relation labels without knowing the exact object locations,

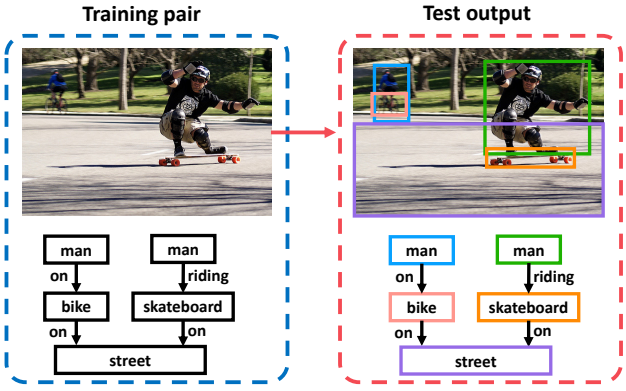


Figure 1. Demonstration the task of WS-SGG. During training, only the pair of image and the ungrounded scene graph are provided. For test stage, given an image, the model should output the full scene graph.

i.e., bounding boxes. Such a weakly-supervised learning setting effectively assuages the difficulties associated with data annotation. For the generalization issue, one can obtain the ungrounded scene graph label from the caption of the image through a language parser [42], and the ungrounded scene graph can be paired with the image that is described by the caption. As the community has collected extremely large image caption datasets [34], it provides WS-SGG task with a considerable amount of training data and alleviates the generalization problem. Hence, WS-SGG is of great significance.

We are not the first one to deal with this task. Typically one can deem the region proposals extracted from an image as the nodes of a *visual graph*, and the ungrounded scene graph as the *label graph*. The node alignment between these two graphs has to be discerned during the optimization process of SGG. Therefore, one major challenge for WS-SGG is graph matching, as we should consider the similarity of both the nodes and their relationships. One previous work VSPNet [52] converts the standard scene graph, where objects are nodes and relations are edges, to a bipartite graph with one part object nodes and another part relations nodes, where the role (subject, object) becomes the edge. It iteratively match each part of nodes, which fails to apply to

standard scene graph structure, which has already been extensively researched [21, 53, 14, 45, 11, 29, 56, 3, 22, 37, 41, 50, 36] Therefore directly addressing WS-SGG on the standard scene graph structure can easily leverage the previous research outcomes.

Therefore, we tackle the WS-SGG task on the standard scene graph structure and propose a simple baseline that decouples the problem into two parts: a weakly-supervised graph matching (WSGM) module that learns to align the visual and label graph; and a standard supervised SGG model to generate the scene graph. We select the efficient first-order graph matching (FOGM) algorithm (only match by node similarity) and employ the Multi-Instance Learning (MIL) mechanism with the contrastive learning objective to train the graph matching module, and use the matched scene graph as pseudo ground truth to train the standard SGG model. The decoupling allows the baseline to adapt to any standard SGG model, resulting in a versatile model. To our surprise, we find the simple baseline can already achieve much higher performance than VSPNet, the current state-of-the-art, in terms of both graph matching and SGG.

Furthermore, we try to answer the following core questions related to our model. (1) Is the graph matching idea better than a grounding model [16]? (2) What is the good practice of selecting the negative sample and loss to enable such contrastive learning for WSGM? (3) Is higher-order graph matching (considering relation similarity) better?

In a word, our contribution is that we propose a very simple baseline that versatilely works for standard scene graphs but can significantly outperform the complex state-of-the-art method.

The rest of this paper is organized as follows. We review the related works about SGG, WS-SGG, and graph matching in Sec. 2. We formulate the problem and introduce our model pipeline in Sec. 3. Experimental settings and result analysis are presented in Sec. 4. Finally, we conclude the paper in Sec. 5.

## 2. Related Work

**Scene Graph Generation.** A scene graph is a structured abstraction of the scene [17, 44] that can serve various downstream visual reasoning tasks, *e.g.* VQA [13], image caption [59], image generation [15], action recognition [30]. One major stream is Recurrent Neural Network (RNN) based SGG [53], including the SGG in tree structure [37, 41]. Another stream is Graph Convolutional Network (GCN) based SGG [21, 14, 45, 11, 29, 56, 3, 22]. The above methods only consider the image as a fully connected visual graph with objects as nodes. However, our graph encoder requires the ability to encode label graph, where the relation types are specified, making the label graph partially connected. [7] regards the relation as a function that transforms the node feature across the graph, allowing the explicit encoding of relation feature. But it is computationally expensive since it needs to keep all the relation functions and feed the node feature to all of the relation functions for each message passing, while our proposed edge attention message passing is able to encode the relation feature both for visual and label graph in a cheaper manner.

**Weakly-supervised Scene Graph Generation.** Therefore, the WS-SGG task only needs the image-level label that describes the object and relation type without box location assigned. The difficulty lies in the assignment of relation triplet to the image. [55] addresses the task using the weakly-supervised object detection framework WSDDN [2], but it is trained with each relation triplet individually. [28] considers the triplet supervision globally with only a simple linear regression model. [1] tackles SGG with only relation label without subject and object label, but relies on a pretrained object detector. VSPNet [52] uses a complex iterative graph matching process to match the label graph to the node, which can handle higher-order relations. However, our paper proposes a much simpler graph matching process and decouples the graph matching with the scene graph generation process. Our method also achieves much better results than all previous methods. Recently, [47, 58] try to tackle SGG via language supervision, our method can serve as an important block for them.

**Graph Matching.** Graph matching builds the correspondence between two graphs in terms of unary node structure (first-order), pair-wise relationships (second-order), or even high-order relationships. The first-order graph matching can be efficiently solved via Hungarian algorithm [18] in cubic time complexity. More works focus on second-order graph matching, which is formulated as a quadratic integer program [23]. This is known as NP-hard so the approximated solution is derived [9, 20, 6, 60, 49]. Recently researchers also integrate deep learning with graph matching algorithms through differentiable solvers, leading to an end-to-end training fashion [51, 39, 48, 8, 57]. However, the above deep graph matching methods only work in supervised graph matching, while in this paper, we are faced with the unsupervised situation. [40] presents the unsupervised graph matching based on graduated assignment [9]. Alternatively, we try to reduce the second-order or high-order relation into first-order representation and address the unsupervised problem in the efficient Hungarian algorithm.

## 3. Method

### 3.1. Problem Formulation

Given an image $I$, the goal is to generate a visual graph $G = (\mathcal{N}, \mathcal{E})$, where each node is a bounding box $b_i$ paired with an entity class $c_i \in \mathcal{C}_e$ and each edge is a predicate class $p_{ij} \in \mathcal{C}_p$ connecting subject node $i$ and object node $j$, *i.e.*, $\mathcal{N}_v = \{(b_i, c_i)\}_{i=1}^{n_e}$, $\mathcal{E}_v = \{p_{ij}\}_{i,j=1}^{n_e}$. In the train-

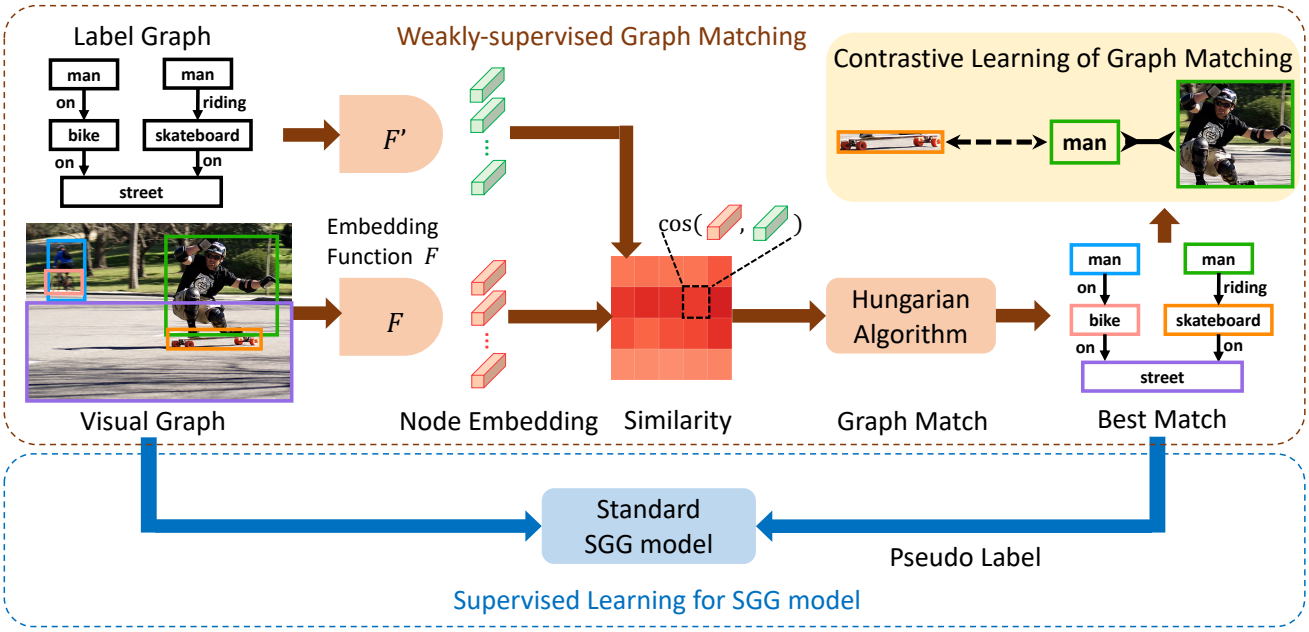


Figure 2. Demonstration of our pipeline. The whole model is composed of a weakly-supervised graph matching module and a supervised SGG model. In the graph matching module, first-order graph matching is applied and the parameter in $F$ is learned via the contrastive learning.

ing stage of WS-SGG, the visual graph has unknown entity and predicate classes. And we define a label graph $G' = (\mathcal{N}', \mathcal{E}')$ to denote the ungrounded scene graph label, where $\mathcal{N}' = \{c_i\}_{i=1}^{n'_e}$ and $\mathcal{E}' = \{p_{ij}\}_{i,j=1}^{n'_e}$. The label graph does not contain the location information for each entity node. Therefore, one has to align the visual graph and label graph to enable the training of SGG. To tackle this challenging task, we propose a simple baseline that decouples the problem into two steps. The first step is formulated as a weakly-supervised graph matching (WSGM) between the visual and label graphs to obtain the class labels for the nodes and edges in the visual graph. The second step is to learn a standard SGG model from the pseudo scene graph label obtained from the first step. The advantage of such a decoupling design is that it can work for any standard SGG model with just a simple plugin of the graph matching module. The overall pipeline is shown in Fig. 2.

### 3.2. Weakly-Supervised Graph Matching

The graph matching process is weakly-supervised because we only know that $G$ and $G'$ is a matched pair of graphs but do not know the exact node correspondence. To begin with, the input visual node feature $\mathbf{e}$ is an RoIPooling [31] feature concatenating an Multiple Layer perceptron (MLP) processed spatial feature $\mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^9$ consists of normalized coordinate $(x_1, y_1, x_2, y_2)$, center $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$, size $(x_2 - x_1, y_2 - y_1)$, and area $(x_2 - x_1)(y_2 - y_1)$. The input label node feature $\mathbf{e}'$ is the GloVe [27] embedding of the object class.

We use two embedding functions $F$, $F'$ to respectively encode the node feature $\mathbf{e}$, $\mathbf{e}'$, leading to the embedding of node $i$ in $G$ as $\mathbf{h}_i$ and node $j$ in $G'$ as $\mathbf{h}'_j$. Our solution is to do graph matching firstly, and then follow the idea of MIL and learn parameters in $F$ and $F'$ through contrastive learning, which is illustrated in the brown box in Fig 2. For graph matching, the second-order graph matching (match both nodes and edges) algorithm usually resorts to Gradually Assignment [9, 40] technique, which is time-consuming. Hence, we alternatively choose first-order graph matching (FOGM) that only match the nodes. Precisely, we first compute the cosine similarity between the two nodes as

$$s_{ij} = \cos(\mathbf{h}_i, \mathbf{h}'_j); \tag{1}$$

then we expect to find a one-to-one alignment $\mathcal{I}$ between the two graphs

$$\mathcal{I} = \{(i,j) | i \in \{1 \dots n_e\}, j \in \{1 \dots n'_e\}\}, \tag{2}$$

such that

$$\mathcal{I}^* = \arg\max_{\mathcal{I}} \sum_{(i,j)\in\mathcal{I}} s_{ij}. \tag{3}$$

The optimal alignment $\mathcal{I}^*$ is solved using Hungarian algorithm [18], whose complexity is only $\mathcal{O}(n^3)$. Note that although the FOGM algorithm focuses only on the similarities of nodes instead of edges, our intuition is that if we can let the node embedding function $F$ encode the edge context, the FOGM algorithm can still work for higher-order graph matching. The selection of $F$ is later discussed in Sec. 3.4.

Next, to learn the embedding function $F$, following the idea of MIL [16], we adopt the triplet loss to enhance such alignment by forcing the similarity of the matched nodes and distancing the unmatched nodes. Therefore the graph matching loss $L_{gm}$ is represented as

$$L_{gm} = \sum_{(i,j)\in\mathcal{I}^*, i\neq i'} \max(0, s_{i'j} - s_{ij} + \Delta), \quad (4)$$

where $\Delta$ is a positive margin. The negative objects are drawn from the current label's unmatched objects, both from the current frame and other frames. In Sec. 4.4 we observe the negative samples from other frames can significantly boost the performance.

### 3.3. SGG Generation

Having obtained the pseudo scene graph label via the graph matching, we further train a standard fully-supervised SGG model to output the scene graph from the input image (blue dashed box in Fig. 2). Any standard SGG model can be applied into this pipeline. The final object and predicate category are trained with cross-entropy loss $L_{sgg}$, following [53]. Hence, the final loss $L$ is the combination of graph matching loss and SGG loss $L = L_{gm} + L_{sgg}$.

### 3.4. Selection of Node Embedding $F$

The most straightforward choice of $F$ is an MLP, which is used as our default setting. In this case, no edge information is encoded to the node representation. Therefore, it is natural to ask whether we can use graph neural network (GNN) to encode the edge context into the node. To answer this question, we need a GNN that can encode the label graph's categorical edge information into the node representation. However, standard SGG model [44, 53] fails to do so because they only take in the object feature without knowing their relation type. Therefore we propose a message passing scheme for GNN named Edge Attention Massage Passing (EAMP), allowing edge type feature to be explicitly encoded into node representation. For the EAMP, the initial node state is the input node embedding $\mathbf{h}_i^{(0)} = \mathbf{e}_i$. At the $k^{\text{th}}$ iteration, we define an score to measure the confidence if there is an edge pointing from node $i$ to $j$:

$$\beta_{ij}^{(k)} = \text{Sigmoid}(f_\beta([\mathbf{h}_i^{(k)}; \mathbf{h}_j^{(k)}])), \quad (5)$$

where the $[\cdot;\cdot]$ denotes concatenation of two vectors, and $f$ denotes MLP. Then $D_p \in \mathbb{R}^{|\mathcal{C}_p|\times d}$ is an embedding dictionary of all predicates in $\mathcal{C}_p$ where the first predicate is background class. And a valid predicate dictionary $\hat{D}_p \in \mathbb{R}^{(|\mathcal{C}_p|-1)\times d}$ is $D_p$ without the background embedding. To enforce the message-passing process to be aware of the edge type, we compute the attention score from the pair-wise node feature to the valid dictionary as

$$\hat{\boldsymbol{\alpha}}_{ij}^{(k)} = \text{Softmax}(f_\alpha([\mathbf{h}_i^{(k)}; \mathbf{h}_j^{(k)}])\hat{D}_p^\intercal/\sqrt{d}). \quad (6)$$

However, there may be no valid relation type between two nodes, so we augmented the predicate attention with edge confidence such that it can attend to the background class, and the attended predicate representation is obtained from the augmented attention as

$$\boldsymbol{\alpha}_{ij}^{(k)} = [1-\beta_{ij}^{(k)}; \beta_{ij}^{(k)}\hat{\boldsymbol{\alpha}}_{ij}^{(k)}], \quad \mathbf{p}_{ij}^{(k)} = \boldsymbol{\alpha}_{ij}^{(k)} D_p. \quad (7)$$

Then each node aggregate neighbors' information through both subject and object FC layers as

$$\mathbf{m}_i^{(k)} = \sum_{j,j\neq i} \left(\bar{\beta}_{ij}^{(k)} f_s([\mathbf{h}_j^{(k)}; \mathbf{p}_{ij}^{(k)}]) + \bar{\beta}_{ji}^{(k)} f_o([\mathbf{h}_j^{(k)}; \mathbf{p}_{ji}^{(k)}])\right), \quad (8)$$

where $\bar{\beta}_{ij}^{(k)} = \beta_{ij}^{(k)} / \sum_{j,j\neq i} \beta_{ij}^{(k)}$. Note the above aggregation also considered the predicate type, enabling the message passing process aware of the relation categories. Finally, GRU [5] is adopted to update the node feature as

$$\mathbf{h}_i^{(k+1)} = \text{GRU}(\mathbf{h}_i^{(k)}, \mathbf{m}_i^{(k)}). \quad (9)$$

After $K$ iterations, the refined node feature with edge type context is obtained as $\mathbf{h}_i^K$.

The aforementioned message passing is formulated with soft attention to the predicate type, suitable for the visual graph, as its predicate category is not determined. Since the label graph has determined relation type, we adopt hard attention instead of the soft one. Therefore, $\beta_{ij}$ in Eq. (5) will be changed as

$$\beta_{ij}^{(k)} = \begin{cases} 0 & \text{if } p_{ij} = \text{background}, \\ 1 & \text{otherwise}, \end{cases} \quad (10)$$

and $\boldsymbol{\alpha}_{ij}^{(k)} \in \mathbb{R}^{|\mathcal{C}_p|}$ in Eq. (7) changed as a one-hot vector indicating the predicate category of $p_{ij}$. So far, we can increase the number of message passing to encode the edge context into the node and are ready for higher-order graph matching using Hungarian algorithm.

## 4. Experiment

### 4.1. Dataset and Metric

**Dataset.** We evaluate our method on Visual Genome (VG) dataset [17], consisting of 108,077 images with scene graph annotations. Following VSPNet [52], we evaluate our method on two common splits [44, 55] with different label preprocessing strategies. [44] keeps most-frequent 150 object categories and 50 predicate types, with train/test set 75,651/32,422 images. While [55] select 200 object categories and 100 predicate types, with train/test set 73,801/25,857 images.

**Metric.** We firstly introduce the metric to measure the graph matching performance. Note that in the label graph, even if two nodes belong to the same object category, they

| Method | SGGen | | | SGCls | | | PredCls | |
|---|---|---|---|---|---|---|---|---|
| | $R_{inst}$ | $R_{obj}$ | $R_{prd}$ | $R_{inst}$ | $R_{obj}$ | $R_{prd}$ | $R_{inst}$ | $R_{prd}$ |
| Upper bound | 39.65 | 39.65 | 27.32 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| VSPNet | 2.75 | 4.50 | 0.78 | 59.89 | 71.86 | 50.31 | 70.94 | 63.54 |
| WSGM (Ours) | **9.07** | **13.15** | **1.87** | **67.61** | **77.93** | **58.99** | **74.22** | **68.83** |

Table 1. Comparison of graph matching accuracy on the training set of the VG split [44].

are two distinctive instances due to different neighbor contexts, as the "man" in Fig. 1. Hence we devise the following metrics: (1) *Instance-level recall* ($R_{inst}$): A box is correctly matched if the box is matched with the correct node in the label graph and is correctly located (has more than 50% intersection-over-union (IoU) with the ground truth (GT) box). The recall is computed as the ratio of the correctly matched boxes to all the GT boxes for each image, followed by an averaged across all images. (2) *Object-level recall* ($R_{obj}$): A box is correctly matched if the box is assigned with the correct object category and is correctly located. Then the recall is computed the same way as $R_{inst}$. Note that $R_{obj}$ is looser than $R_{inst}$, because although a box is assigned with the correct category, it can be matched to a wrong node instance with the correct object category in the label graph. (3) *Predicate-level recall* ($R_{prd}$): A predicate is correctly matched if its subject and object boxes are correctly matched in the instance-level.

Next we describe the common evaluation metric scene graph generation. (1) Predicate classification (PredCls): given GT boxes and object labels, predict relationship types of object pairs. (2) Scene graph classification (SGCls): given GT object boxes, predict object categories and relationship types. (3) Scene graph detection (SGGen): given an image, predict boxes, categories of region proposals and relation types of object pairs. Only when the labels of the `subject-relation-object` triplet are correctly classified, and the boxes of subject and object have more than 50% IoU with the GT, it is counted as a correctly detected entity. (4) Phrase detection (PhrDet) [55]: given an image, predict the relationship triplet with a union bounding box enclosing both the object and subject. It is correct if the labels of the triplet are correct and the union box match with GT union box with IoU greater than 0.5. Recall of the above metric is computed for each image and then averaged over the dataset, leading to Recall@K metrics (K = [20, 50, 100]). Moreover, in the triplet ranking process, we have the *graph constraint* that the same object pair cannot predict multiple predicates in our default setting. If such constraint is disabled, No Graph Constraint Recall@K will be indicated, following [53].

### 4.2. Implementation Details

We follow the same way of VSPNet [52] to extract visual features. 20 top region proposals for each image are extracted from the RoIPooling [31] feature pretrained on Open Image dataset [19]. We use 200-dimensional GloVe embeddings [27] to represent the object and predicate features in the label graph. The EAMP's message passing iteration is set as 1 when used as a SGG model. The WSGM and SGG model are trained together, while the SGG learning loss is subject to a linear warmup for 12k iterations to ensure the FOGM has been well trained. We set $\Delta = 0.1$ according to grid search and assign a discount weight 0.1 for background object and 0.01 for background relation for the $L_{sgg}$ due to data balance. We optimize the model via SGD with learning rate 0.002 and momentum 0.9. The batch size is 32.

### 4.3. Main Results

**Comparison Methods.** We compare our system with the following methods for WS-SGG:

- *PPR-FCN* [55]: it extends the structure of WSDDN [2] to detect relation triplets.
- *VTransE-MIL* [55]: it follows the same pipeline as VTransE [54] but using the NoisyOR MIL [24] as the loss function for object and relation detections.
- *VSPNet* [52]: it converts the scene graph to a bipartite graph of entity nodes and predicate nodes, where each part of nodes iteratively conduct first-order match to approximate the second-order graph matching.

Due to the space limitation, we move the result on VG split [55] to Appx. A, where PPR-FCN and VTransE-MIL are compared.

**Graph Matching Performance.** We firstly compare our method with VSPNet, and the result is shown in Tab. 1, indicating our simple FOGM framework out-perform VSPNet, which does the second-order graph matching (SOGM). Although intuitively SOGM is better than FOGM, it may not be true considering the WSGM, where the inaccurate matching will exaggerate the noisy for the node embedding learning. Multiple factors might contribute the matching accuracy, including the loss function, the network message passing complexity, etc. We suspect that our simple FOGM has less error propagation than the nested optimization used in VSPNet. And our contrastive loss could be an more ef-

| Method | Supervision | SGGen | | | SGCls | | | PredCls | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | R@20 | R@50 | R@100 | R@20 | R@50 | R@100 | R@20 | R@50 | R@100 |
| MOTIFS [53] | Full | 25.48 | 32.78 | 37.16 | 35.63 | 38.92 | 39.77 | 58.46 | 65.18 | 67.01 |
| VSPNet* [52] | Weak | - | 4.01 | 4.17 | - | 23.43 | 23.50 | - | 44.59 | 44.77 |
| WSGM+IMP | | 3.87 | 5.06 | 5.73 | 25.09 | 30.04 | 31.85 | **48.22** | **61.37** | **65.83** |
| WSGM+Motif | | 4.12 | **5.59** | **6.45** | 23.54 | 29.16 | 31.39 | 44.10 | 59.07 | 64.60 |
| WSGM+EAMP | | **4.19** | 5.43 | 6.02 | **25.32** | **30.38** | **32.10** | 46.57 | 59.19 | 64.22 |
| Best Improve | | - | 39% | 54% | - | 30% | 37% | - | 38% | 47% |

Table 2. Comparison with other methods on VG split [44]. Best improve is the best relative improvement over VSPNet among the WSGM based methods. * denotes the re-evaluated number.

| Rate | R@20 | R@50 | R@100 | $R_{inst}$ |
|---|---|---|---|---|
| $L_{gm}^{vis}$ | **4.19** | **5.43** | **6.02** | **10.12** |
| $L_{gm}^{lbl}$ | 3.61 | 4.79 | 5.43 | 9.71 |
| $L_{gm}^{comb}$ | 3.77 | 4.84 | 5.39 | 9.33 |

Table 3. Ablation study of different construction of the contrast in the setting of SGGen.

| Loss | R@20 | R@50 | R@100 | $R_{inst}$ |
|---|---|---|---|---|
| Logistic | 1.23 | 1.73 | 2.01 | 2.33 |
| NCE | 1.21 | 1.80 | 2.17 | 2.46 |
| Triplet | **2.84** | **3.95** | **4.56** | **8.77** |

Table 4. Ablation study of different graph matching loss in the setting of SGGen.

fective loss function then the loss of VSPNet, as the way to formulate the contrastive loss and negative sample selection will boost the performance significantly (Sec. 4.4). Finally, the upper bound is computed by assuming all the matching are correct, unless the region proposals cannot cover all the GT boxes. We can see the upper bound graph matching recall for SGGen setting is quite low (39.65% for the instance recall and 27.32% for predicate recall), severely limiting the matching performance, indicating that the OpenImage [19] pretrained RPN still has a large semantic gap to the VG dataset. And the SGCls and PredCls upper bound are 100% as they use the ground truth box as region proposals.

**SGG Performance.** To show the effect of the WSGM on the SGG task, we use the pseudo label computed from the WSGM to train the existing standard SGG models, namely Iterative Message Passing (IMP) [44], Neural Motif (MOTIF) [53], and the proposed EAMP as our SGG models. As the IMP and MOTIF both require the input feature of the union of box pair, which is not available in our proposal setting, we replace it with the concatenation of the subject and object feature followed by an FC layer. Also, we disable the object label feature in the input for MOTIF as no object detector is available now. The frequency prior is applied by default for all the SGG models, following MOTIF [53]. The performance of SGG with and without scene graph constraint is shown in Tab. 2 and Tab. 5, respectively. Note that the number of original VSPNet [52] paper does not strictly satisfy the scene graph constraint, so for a fair comparison, we re-evaluate the VSPNet by keeping the top predicate from a unique object-subject pair. Moreover, we compare the original number of VSPNet in non-constraint graph (Tab. 5). We observe that the FOGM based SGG models all outperform the VSPNet by a large margin (SGGen R@100: 54% relative improvement, SGCls R@100 37%), demonstrating the effectiveness of our algorithm. We can see that different SGG models share comparable performance. The MOTIF is not always better than IMP because the object label feature is disabled, which is different from a supervised setting. The qualitative visualization is shown in Appx. C. Also, the weakly-supervised performance is still far behind the fully-supervised setting for SGGen, while for SGCls and PredCls the gap is smaller, indicating good object proposal is critical for WS-SGG.

### 4.4. Ablation Study

Due to the space limit, we move the study of the iteration number of EAMP as a SGG model in Appx. B.1, and the study on the margin of the triplet loss to Appx. B.2. All the ablation study are done in VG split [44].

**Different ways to learn the contrast.** We observe that the way to construct the positive and negative contrast is important. We studied three different construction of the contrast. The first case is Eq. (4), where the anchor is the label node and the positive and negative samples are the visual nodes, so we let $L_{gm}^{vis}$ to denote Eq. (4). Reversely, if the visual node is the anchor and the label nodes are used to construct the contrast, the loss becomes

$$L_{gm}^{lbl} = \sum_{(i,j)\in\mathcal{I}^*, j\neq j'} \max(0, s_{ij'} - s_{ij} + \Delta); \qquad (11)$$

and the combination of both direction is $L_{gm}^{comb} = L_{gm}^{vis} +$

| Method | Supervision | SGGen | | | SGCls | | | PredCls | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | R@20 | R@50 | R@100 | R@20 | R@50 | R@100 | R@20 | R@50 | R@100 |
| MOTIFS [53] | Full | 27.04 | 36.58 | 43.43 | 40.58 | 48.48 | 51.98 | 66.39 | 81.02 | 88.24 |
| VSPNet [52] | Weak | - | 4.70 | 5.40 | - | 30.50 | 32.70 | - | 57.70 | 62.40 |
| WSGM+IMP | | 3.91 | 5.26 | 6.31 | 27.03 | 34.57 | 38.89 | **51.13** | **69.57** | **80.19** |
| WSGM+Motif | | 4.16 | **5.74** | **6.94** | 25.17 | 33.12 | 37.85 | 46.76 | 66.86 | 78.46 |
| WSGM+EAMP | | **4.25** | 5.70 | 6.70 | **27.40** | **35.09** | **39.43** | 49.60 | 67.12 | 77.74 |
| Best Improve | | - | 22% | 28% | - | 15% | 21% | - | 21% | 29% |

Table 5. Comparison with other methods in non-constraint graph.

| | R@20 | R@50 | R@100 | $R_{inst}$ |
|---|---|---|---|---|
| grounding | 3.26 | 4.07 | 4.48 | 6.57 |
| FOMP | 4.19 | 5.43 | 6.02 | 10.12 |
| sub-graph 10% | 3.51 | 4.40 | 4.79 | 9.13 |
| sub-graph 50% | 4.20 | 5.29 | 5.81 | 9.18 |

Table 6. Compare with grounding and different sub-graph rate in the setting of SGDet.

$L_{gm}^{lbl}$. The comparison of the three cases is shown in Tab. 3. We observed that the contrast in visual nodes is more favorable than contrast for the label nodes.

**Comparison with alternative contrastive loss.** We compare the triplet loss against the other two commonly used contrastive losses: logistic loss [26] and NCE loss [35, 43], which have been widely applied in unsupervised learning literature [4, 25]. To make the comparison fair, for all losses, the anchor is the label node, and negative samples are drawn from the unmatched visual nodes in the same image with the positive samples. In the triplet loss and logistic loss, the same number of negative proposals are randomly sampled as the matched proposals. With the same notation as Eq. (4), the logistic loss is written as:

$$-\sum_{(i,j)\in\mathcal{I}^*, i\neq i', j\neq j'} \big(\log(\sigma(s_{ij})) + \log(\sigma(-s_{i'j}))\big) \quad (12)$$

where $\sigma(\cdot)$ is the sigmoid function. The contrastive loss will take all the negative proposals in the images on the denominator, which is given as

$$\sum_{(i,j)\in\mathcal{I}^*} -\log \frac{\exp(s_{ij})}{\sum_{i'}\exp(s_{i'j})} \quad (13)$$

The performance of them are presented in Tab. 4. We find that the triplet loss in the graph matching problem is significantly better than the other two alternatives. Not that such finding is different from the common observation that the NCE loss outperforms the triplet loss [4]. The different feature extraction setting might cause this discrepancy. In our case, the proposal feature is extracted by the off-the-shelf RPN [32], and the learnable embedding on top of it is shallower compared with the learnable embedding as a ResNet50 [10] used in SimCRL [4]. Therefore, our embedding has to maintain the original semantics of RPN feature and may not be able to shape a large similarity gain of the positive pair over the negative pair. From this view, the triplet loss is the most suitable one as the margin of the similarity can be tuned to be smaller, while NCE loss and logistic loss will steadily enlarge the similarity gain, leading to the difficulty of reconciling the original RPN feature.

**Is higher-order graph matching better?** In our standard-setting, the graph matching process adopts the MLP to encode only the object feature without considering the predicate feature. Therefore, we pay a tentative effort to study if a message passing model, *e.g.* EAMP, can encode the predicate information into the node feature such that a simple first-order graph matching can still be applied to achieve higher-order matching. Since EAMP can encode the relation feature into node feature for both visual graph and label graph, we replace the MLP to EAMP and adjust the number of message passing iterations. The result is shown in Tab. 7, and 0 iteration makes the EAMP reduced to MLP. We can see that for SGGen, MLP yields the best result, and increasing the iteration number will reversely deteriorate the matching accuracy. We guess that as the instance recall is low (only around 10%), the model will force the similarity of the rest 90% of the unmatched nodes, already introducing much noise in the matching process. Further involving the embedding of relation context into the node might trigger more noise since the relation is also error-prone. Another reason is that we do not have the topology structure of the visual graph, and thus it is assumed initially as a fully connected graph, leading to higher graph matching difficulties than those graph match where all graphs have their topologies. However, as we can see that for SGCls and PredCls where the instance noise is smaller than SGGen, more iterations of message passing will bring certain benefits.

**Is the graph matching better than a grounding model?** We firstly highlight the difference of our WSGM with

| Method | SGGen | | | SGCls | | | PredCls | |
|---|---|---|---|---|---|---|---|---|
| | $R_{inst}$ | $R_{obj}$ | $R_{prd}$ | $R_{inst}$ | $R_{obj}$ | $R_{prd}$ | $R_{inst}$ | $R_{prd}$ |
| EAMP iter 0 | **9.07** | **13.15** | **1.87** | **67.61** | 77.93 | **58.99** | 74.22 | 68.83 |
| EAMP iter 1 | 8.39 | 12.18 | 1.63 | 64.39 | **80.00** | 53.64 | 74.36 | 69.05 |
| EAMP iter 2 | 7.19 | 10.75 | 1.00 | 63.31 | 78.97 | 53.00 | **74.51** | **69.50** |

Table 7. Ablation study of the iteration number of EAMP as the embedding function for graph matching on the training set.

| Loss | R@20 | R@50 | R@100 | $R_{inst}$ |
|---|---|---|---|---|
| Full | **4.19** | **5.43** | **6.02** | **10.12** |
| w/o CrossMatch | 2.84 | 3.95 | 4.56 | 8.14 |
| w/o HardNeg | 3.63 | 4.74 | 5.31 | 8.48 |

Table 8. Ablation study of different negative example mining strategies in the setting of SGDet.

| Curriculum | R@20 | R@50 | R@100 | $R_{inst}$ |
|---|---|---|---|---|
| simple→hard | 4.21 | 5.44 | 6.08 | 9.52 |
| hard→simple | 4.30 | 5.50 | 6.14 | 9.88 |
| No curriculum | 4.19 | 5.43 | 6.02 | 10.12 |

Table 9. Train with curriculum learning in the setting of SGDet.

grounding model, *e.g.*, DVSA [16]. Each noun will query its most similar object independently in standard grounding model. Nevertheless, we use graph supervision where the label graph is a holistic structure, and we do not regard the label nodes as independent queries but apply the one-to-one mapping constraint such that two queries will not match the same object. Tab. 6 shows that the WSGM outperforms the grounding setting, indicating that the one-to-one mapping constraint is essential to our model and the purely grounding model is inadequate.

**The importance of graph supervision.** The one-to-one mapping might also cause the error since one mismatch will trigger other mismatched nodes. Thus we further cut the original scene graph into a sub-graph which might reduce such mismatch propagation. Note that such a random cut happens at each training iteration to ensure all the label nodes have the chance to be trained. Tab. 6 presents the sub-graph with 50% and 10% of nodes remained, indicating that fewer nodes will decrease the performance and further advocating that one-to-one mapping constraint is more important than the concern of its mismatch propagation.

**Different negative sample mining.** Here we study the importance of selecting negative samples. By default, for an anchor label node, its negative samples come from the matched visual node of other label nodes in both the current image and the other image. CrossMatch: indicating the negative objects can come from other images. HardNeg: indicating the negative objects must be matched by other nodes. Without HardNeg, the negative sample can be any unmatched objects of current label nodes. The Tab. 8 shows that the CrossMatch can bring significant improvement because the model can see more negative samples and thus lead to better contrast for learning. Also, the HardNeg is essential as the visual nodes matched by other label nodes have better semantic meaning than the background object; therefore, such semantic difference of other objects from current label nodes will assist contrastive learning.

**What may not help.**

*Handcrafted curriculum learning.* Motivated by the intuition that the label graph with fewer nodes may be easier to learn than the ones with more nodes, we split the training set into four splits according to the number of nodes in the label graphs from easy to hard. From Tab. 9, simple→hard means training from easy split to hard, and hard→simple means the reverse. We observe that no matter which direction we use, the performance has slight change; therefore, ranking the complexity of the label graph will not affect the model performance.

## 5. Conclusion

In summary, we decouple the WS-SGG task into a WSGM module and a standard SGG model, where a contrastive learning framework based on efficient first-order graph matching is introduced. Our method is much simpler than the previous method while achieves significant improvement on both graph matching accuracy and SGG performance. We further empirically illustrate the graph matching is better than a grounding model, provide good practice of selecting negative samples and the loss. We believe this work serves as a simple yet strong baseline for the future development of WS-SGG problem.

**Acknowledgments.** This work has been partially supported by the National Science Foundation (NSF) under Grants 1741472, 1813709, and 1909912 and a research gift from Adobe. YZ and YL acknowledge the support provided by the UW-Madison Office of the Vice Chancellor for Research and Graduate Education with funding from WARF. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

## References


[1] Federico Baldassarre, Kevin Smith, Josephine Sullivan, and Hossein Azizpour. Explanation-based weakly-supervised learning of visual relations with graph networks. In *ECCV*, pages 612–630. Springer, 2020. 2

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016. 2, 5

[3] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*, pages 4613–4623, 2019. 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 7

[5] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 4

[6] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *ECCV*, pages 492–505. Springer, 2010. 2

[7] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and Fei-Fei Li. Visual relationships as functions: Enabling few-shot scene graph prediction. In *ICCV Workshops*, pages 0–0, 2019. 2

[8] Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*, 2020. 2

[9] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *PAMI*, 18(4):377–388, 1996. 2, 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7

[11] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. *NeurIPS*, 31:7211–7221, 2018. 2

[12] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 1

[13] Drew A Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *arXiv preprint arXiv:1907.03950*, 2019. 1, 2

[14] Seong Jae Hwang, Sathya N Ravi, Zirui Tao, Hyunwoo J Kim, Maxwell D Collins, and Vikas Singh. Tensorize, factorize and regularize: Robust visual relationship learning. In *CVPR*, pages 1014–1023, 2018. 2

[15] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pages 1219–1228, 2018. 1, 2

[16] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015. 2, 4, 8

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 1, 2, 4

[18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2, 3

[19] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, pages 1–26, 2020. 5, 6

[20] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. 2005. 2

[21] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *ECCV*, pages 335–351, 2018. 2

[22] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, pages 3746–3753, 2020. 2

[23] Joao Maciel and Joao Paulo Costeira. A global solution to sparse correspondence problems. *PAMI*, 25(2):187–199, 2003. 2

[24] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *NeurIPS*, pages 570–576, 1998. 5

[25] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, pages 9879–9889, 2020. 7

[26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 7

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 3, 5

[28] Julia Peyre, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Weakly-supervised learning of visual relations. In *ICCV*, pages 5179–5188, 2017. 2

[29] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *CVPR*, pages 3957–3966, 2019. 2

[30] Mengshi Qi, Jie Qin, Annan Li, Yunhong Wang, Jiebo Luo, and Luc Van Gool. stagnet: An attentive semantic rnn for group activity recognition. In *ECCV*, pages 101–117, 2018. 2

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 3, 5

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *PAMI*, 39(6):1137–1149, 2016. 7

[33] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically

precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 1

[34] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 1

[35] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, pages 1857–1865, 2016. 7

[36] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pages 3716–3725, 2020. 2

[37] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pages 6619–6628, 2019. 2

[38] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *CVPR*, pages 1–9, 2017. 1

[39] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *ICCV*, pages 3056–3065, 2019. 2

[40] Runzhong Wang, Junchi Yan, and Xiaokang Yang. Graduated assignment for joint multi-graph matching and clustering with application to unsupervised graph matching network learning. *NeurIPS*, 33, 2020. 2, 3

[41] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Sketching image gist: Human-mimetic hierarchical scene graph generation. 2020. 2

[42] Yu-Siang Wang, Chenxi Liu, Xiaohui Zeng, and Alan Yuille. Scene graph parsing as dependency parsing. *arXiv preprint arXiv:1803.09189*, 2018. 1

[43] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 7

[44] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 5410–5419, 2017. 1, 2, 4, 5, 6

[45] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pages 670–685, 2018. 2

[46] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019. 1

[47] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *CVPR*, pages 8289–8299, 2021. 2

[48] Tianshu Yu, Runzhong Wang, Junchi Yan, and Baoxin Li. Learning deep graph matching with channel-independent embedding and hungarian attention. In *ICLR*, 2019. 2

[49] Tianshu Yu, Junchi Yan, Yilin Wang, Wei Liu, and Baoxin Li. Generalizing graph matching beyond quadratic assignment model. In *NeurIPS*, pages 861–871, 2018. 2

[50] Cong Yuren, Hanno Ackermann, Wentong Liao, Michael Ying Yang, and Bodo Rosenhahn. Nodis: Neural ordinary differential scene understanding. 2020. 2

[51] Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *CVPR*, pages 2684–2693, 2018. 2

[52] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *CVPR*, pages 3736–3745, 2020. 1, 2, 4, 5, 6, 7

[53] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018. 2, 4, 5, 6, 7

[54] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, pages 5532–5540, 2017. 5

[55] Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *ICCV*, pages 4233–4241, 2017. 2, 4, 5

[56] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph generation. 2019. 2

[57] Zhen Zhang and Wee Sun Lee. Deep graphical feature learning for the feature matching problem. In *ICCV*, pages 5087–5096, 2019. 2

[58] Yiwu Zhong, Jing Shi, Jianwei Yang, Chenliang Xu, and Yin Li. Learning to generate scene graph from natural language supervision. In *ICCV*, 2021. 2

[59] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. Comprehensive image captioning via scene graph decomposition. In *ECCV*, pages 211–229. Springer, 2020. 1, 2

[60] Feng Zhou and Fernando De la Torre. Factorized graph matching. In *CVPR*, pages 127–134. IEEE, 2012. 2