

# Counterfactual Critic Multi-Agent Training for Scene Graph Generation

Long Chen<sup>1</sup> Hanwang Zhang<sup>2</sup> Jun Xiao<sup>1</sup> Xiangnan He<sup>3</sup> Shiliang Pu<sup>4</sup> Shih-Fu Chang<sup>5</sup>

<sup>1</sup>Zhejiang University <sup>2</sup>Nanyang Technological University

<sup>3</sup>National University of Singapore <sup>4</sup>Hikvision Research Institute <sup>5</sup>Columbia University

## Abstract

*Scene graphs — objects as nodes and visual relationships as edges — describe the whereabouts and interactions of objects in an image for comprehensive scene understanding. To generate coherent scene graphs, almost all existing methods exploit the fruitful visual context by modeling message passing among objects. For example, “person” on “bike” can help determine the relationship “ride”, which in turn contributes to the confidence of the two objects. However, we argue that the visual context is not properly learned by using the prevailing cross-entropy based supervised learning paradigm, which is not sensitive to graph inconsistency: errors at the hub or non-hub nodes should not be penalized equally. To this end, we propose a Counterfactual critic Multi-Agent Training (CMAT) approach. CMAT is a multi-agent policy gradient method that frames objects into cooperative agents, and then directly maximizes a graph-level metric as the reward. In particular, to assign the reward properly to each agent, CMAT uses a counterfactual baseline that disentangles the agent-specific reward by fixing the predictions of other agents. Extensive validations on the challenging Visual Genome benchmark show that CMAT achieves a state-of-the-art performance by significant gains under various settings and metrics.*

## 1. Introduction

Visual scene understanding, *e.g.*, what and where the things and stuff are, and how they relate with each other, is one of the core tasks in computer vision. With the maturity of modern object detection [37, 27] and segmentation [11], computers can recognize object categories, locations, and visual attributes well. However, scene understanding goes beyond the whereabouts of objects. A more crucial step is to infer their visual relationships — together with the objects, they offer a comprehensive and coherent visually-grounded knowledge, called *scene graphs* [17]. As shown in Figure 1 (a), the nodes and edges in scene graphs are objects and visual relationships, respectively. Moreover, scene graph is an indispensable knowledge representa-

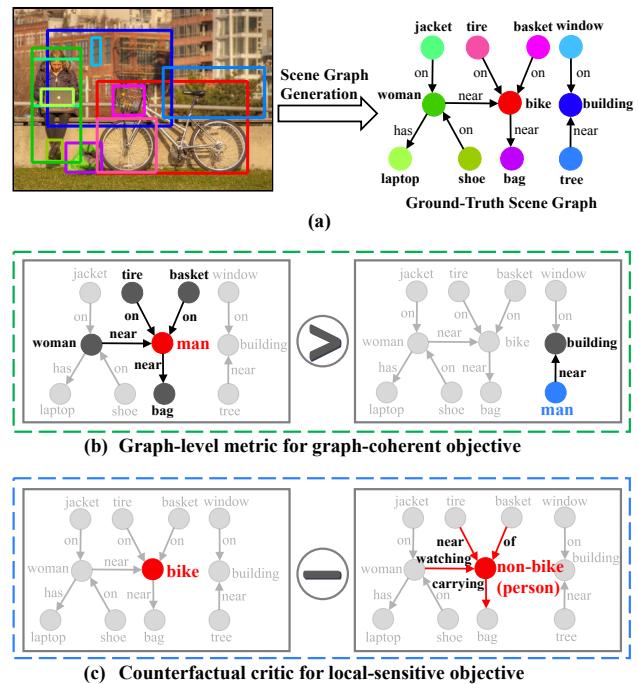


Figure 1: (a) An input image and its ground-truth scene graph. (b) For graph-coherent objective, a graph-level metric will penalize the red node **more than** (>) the blue one, even though both are misclassified as man. (c) For local-sensitive objective, the individual reward for predicting the red node as bike can be identified by **excluding** (−) the reward from non-bike predictions.

tion for many high-level vision tasks such as complex-query image retrieval [17], image captioning [51], visual reasoning [40], image generation [15], and VQA [45, 32].

A straightforward solution for Scene Graph Generation (SGG) is in an independent fashion: detecting object bounding boxes by an existing object detector, and then predicting the object classes and their pairwise relationships separately [29, 55, 50]. However, these methods overlook the fruitful visual context, which offers a powerful inductive bias [6] that helps object and relationship detection. For example in Figure 1, window and building usually co-occur within an image, and near is the most common rela-

tionship between `tree` and `building`; it is easier to infer that `? is building from window-on-?` or `tree-near-?`. Such intuition has been empirically shown benefits in boosting SGG [47, 4, 21, 23, 22, 52, 14, 54]. More specifically, these methods use a conditional random field [59] to model the joint distribution of nodes and edges, where the context is incorporated by message passing among the nodes through edges via a multi-step mean-field approximation [19]; then, the model is optimized by the sum of cross-entropy (XE) losses of nodes (*e.g.*, objects) and edges (*e.g.*, relationships).

Nevertheless, the coherency of the visual context is not captured effectively by existing SGG methods due to the main reason: the XE based training objective is NOT **graph-coherent**. By “graph-coherent”, we mean that the quality of the scene graph should be graph-level: the detected objects and relationships should be contextually consistent; however, the sum of XE losses of objects and relationships is essentially independent. To see the negative impact of this inconsistency, suppose that the red and the blue nodes are both misclassified in Figure 1 (b), based on the XE loss, the errors are penalized equally. However, the error of misclassifying the red node should be more severe than the blue one, as the red error will influence more nodes and edges than the blue one. Therefore, we need to use a graph-level metric such as Recall@K [29] and SPICE [1] to match the graph-coherent objective, which penalizes more for misclassifying important hub nodes than others. Meanwhile, the training objective of SGG should be **local-sensitive**. By “local-sensitive”, we mean that the training objective is sensitive to the change of a single node. However, since the graph-coherent objective is a global pooling quantity, the individual contribution of the prediction of a node is lost. Thus, we need to design a disentangle mechanism to identify the individual contribution and provide an effective training signal for each local prediction.

In this paper, we propose a novel training paradigm: Counterfactual critic Multi-Agent Training (CMAT), to simultaneously meet the graph-coherent and local-sensitive requirements. Specifically, we design a novel communicative **multi-agent** model, where the objects are viewed as cooperative agents to maximize the quality of the generated scene graph. The action of each agent is to predict its object class labels, and each agent can communicate with others using pairwise visual features. The communication retains the rich visual context in SGG. After several rounds of agent communication, a visual relationship model triggers the overall graph-level reward by comparing the generated scene graph with the ground-truth.

**For the graph-coherent objective**, we directly define the objective as a graph-level reward (*e.g.*, Recall@K or SPICE), and use policy gradient [43] to optimize the non-differentiable objective. In the view of Multi-Agent Rein-

forcement Learning (MARL) [44, 28], especially the actor-critic methods [28], the relationship model can be framed as a **critic** and the object classification model serves as a policy network. **For the local-sensitive objective**, we subtract a counterfactual baseline [8] from the graph-level reward by varying the target agent and fixing the others before feeding into the critic. As shown in Figure 1 (c), to approximate the true influence of the red node acting as `bike`, we fix the predictions of the other nodes and replace the `bike` by non-`bike` (*e.g.*, `person`, `boy`, and `car`), and see how such **counterfactual** replacement affects the reward (*e.g.*, the edges connecting their neighborhood are all wrong).

To better encode the visual context for more effective CMAT training, we design an efficient agent communication model, which discards the widely-used relationship nodes in existing message passing works [47, 23, 14, 21, 52, 22]. Thanks to this design, we disentangle the agent communication (*i.e.*, message passing) from the visual relationship detection, allowing the former to focus on modeling the visual context, and the latter, which is essentially a communication consequence, to serve as the critic that guides the graph-coherent objective.

We demonstrate the effectiveness of CMAT on the challenging Visual Genome [20] benchmark. We observe consistent improvements across extensive ablations and achieve state-of-the-art performances on three standard tasks: scene graph detection, scene graph classification, and predicate classification, respectively.

In summary, we make three technical contributions in this paper:

1. We propose a novel training paradigm: Counterfactual critic Multi-Agent Training (CMAT) for SGG. To the best of our knowledge, we are the first to formulate SGG as a cooperative multi-agent problem, which conforms to the graph-coherent nature of scene graphs.
2. We design a counterfactual critic that is effective for training because it makes the graph-level reward local-sensitive by identifying individual agent contributions.
3. We design an efficient agent communication method that disentangles the relationship prediction from the visual context modeling, where the former is essentially a consequence of the latter.

## 2. Related Work

**Scene Graph Generation.** Detecting visual relationships regains the community attention after the pioneering work by Lu *et al.* [29] and the advent of the first large-scale scene graph dataset by Krishna *et al.* [20]. In the early stage, many SGG works focus on detecting objects and visual relationships independently [29, 55, 61, 60, 56], but these independent inference models overlook the fruitful visual context. To benefit both object and relationship detection from visual

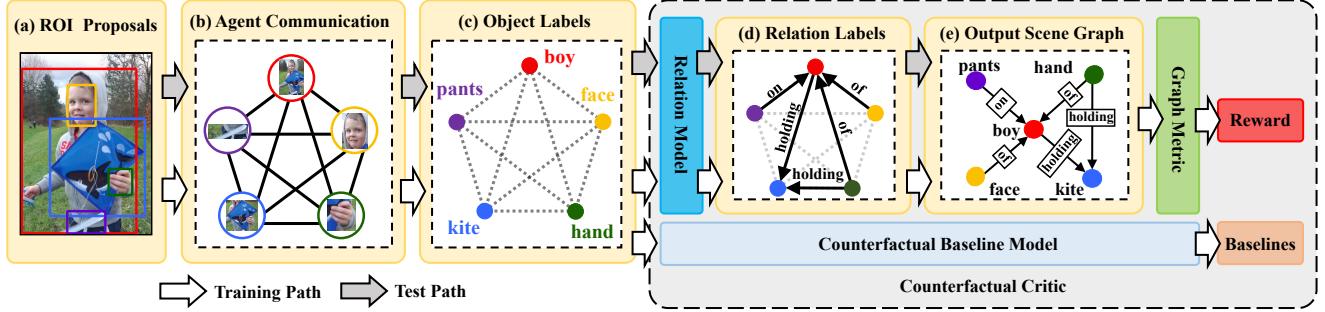


Figure 2: The pipeline of CMAT framework. Given an image, the model uses RPN to propose object regions (a). Then, each object (agent) communicates with others to encode visual context (b). After agent communication, the model predicts class confidence for all objects. Based on the confidence, it selects (random or greedily sampling) object labels (c) and infers visual relationship of object pairs (d). Finally, it generates the scene graph (e). In the training stage, a counterfactual critic is used to calculate the individual contribution.

context, recent SGG methods resort to the message passing mechanism [47, 4, 23, 22, 52, 14, 49]. However, these methods fail to learn the visual context due to the conventional XE loss, which is not graph-level contextually consistent. Unlike previous methods, in this paper, we propose a CMAT model to simultaneously meet the graph-coherent and local-sensitive requirements.

**Multi-Agent Policy Gradient.** Policy gradient is a type of reinforcement learning method which can optimize non-differentiable objective. It had been well-studied in many scene understanding tasks like image captioning [34, 38, 26, 39, 57, 25], VQA [13, 16], visual grounding [3, 53], and visual dialog [5]. Liang *et al.* [24] used a DQN to formulate SGG as a single agent decision-making process. Different from these single agent policy gradient settings, we formulate SGG as a cooperative multi-agent decision problem, where the training objective is graph-level contextually consistent and conforms to the graph-coherent nature of scene graphs. Meanwhile, compared with many well-studied multi-agent game tasks [7, 33, 9, 44], the agent number (64 objects) and action sample space (151 object categories) in CMAT are much larger.

### 3. Approach

Given a set of predefined object classes  $\mathcal{C}$  (including background) and visual relationship classes  $\mathcal{R}$  (including non-relationship), we formally represent a scene graph  $\mathcal{G} = \{\mathcal{V} = \{(v_i, l_i)\}, \mathcal{E} = \{r_{ij}\} | i, j = 1...n\}$ , where  $\mathcal{V}$  and  $\mathcal{E}$  denote the set of nodes and edges, respectively.  $v_i \in \mathcal{C}$  is the object class of  $i^{th}$  node,  $l_i \in \mathbb{R}^4$  is the location of  $i^{th}$  node, and  $r_{ij} \in \mathcal{R}$  is the visual relationship between  $i^{th}$  and  $j^{th}$  node. SGG is to detect the coherent configuration for nodes and edges.

In this section, we first introduce the components of our CMAT (Section 3.1). Then, we demonstrate the details about the training objective of CMAT: graph-coherent and local-sensitive (Section 3.2).

### 3.1. SGG using Multi-Agent Communication

We sequentially introduce the components of CMAT following the inference path ( $\Rightarrow$  path in Figure 2), including object proposals detection, agent communication, and visual relationship detection.

#### 3.1.1 Object Proposals Detection

$$\text{Input : IMAGE} \longmapsto \text{Output : } \{(l_i, x_i^0, s_i^0)\}$$

We use Faster-RCNN [37] as the object detector to extract a set of object proposals. Each proposal is associated with a location  $l_i$ , a feature vector  $x_i^0$ , and a class confidence  $s_i^0$ . The superscript 0 denotes the initial input for the following  $T$ -round agent communication. We follow previous works [47, 54] to fix all locations  $\{l_i\}$  as the final predictions. For simplicity, we will omit  $l_i$  in following sections.

#### 3.1.2 Agent Communication

$$\text{Input : } \{(x_i^0, s_i^0)\} \longmapsto \text{Output : } \{(x_i^T, s_i^T, h_i^T)\}$$

Given the  $n$  detected objects from the previous step, we regard each object as an agent and each agent will communicate with the others for  $T$ -rounds to encode the visual context. In each round of communication, as illustrated in Figure 3, there are three modules: **extract**, **update**, and **message**. These modules share parameters among all agents and time steps to reduce the model complexity. In the following, we introduce the details of these three modules.

**Extract Module.** The incarnation of the extract module is an LSTM, which encodes the agent interaction history and extracts the internal state of each agent. Specifically, for agent  $i$  ( $i^{th}$  object) at  $t$ -round ( $0 < t \leq T$ ) communication:

$$\begin{aligned} h_i^t &= \text{LSTM}(h_i^{t-1}, [x_i^t, e_i^{t-1}]), \\ s_i^t &= F_s(s_i^{t-1}, h_i^t), \quad e_i^t = F_e(s_i^t), \end{aligned} \quad (1)$$

where  $h_i^t$  is the hidden state of LSTM (*i.e.*, the internal state of agent).  $x_i^t$  is the time-step input feature and  $s_i^t$  is the

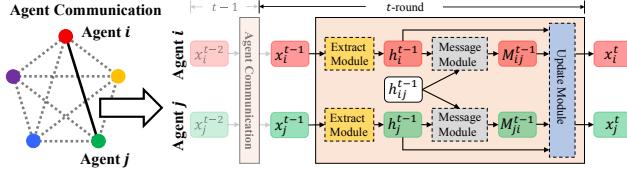


Figure 3: The illustration of agent communication (at time step  $t$ ) between agent  $i$  and agent  $j$  (the red and green node).

object class confidence. The initialization of  $\mathbf{x}_i^t$  (*i.e.*  $\mathbf{x}_i^0$ ) and  $s_i^t$  (*i.e.*  $s_i^0$ ) come from the proposal detection step.  $e_i^t$  is soft-weighted embedding of class label and  $[,]$  is a concatenate operation.  $F_s$  and  $F_e$  are learnable functions<sup>1</sup>. All internal states  $\{\mathbf{h}_i^t\}$  are fed into the following message module to compose communication messages among agents.

**Message Module.** Considering the communication between agent  $i$  and  $j$ , the message module will compose message  $M_{ij}^t$  and  $M_{ji}^t$  for each agent. Specifically, the message  $M_{ij}^t$  for agent  $i$  is a tuple  $M_{ij}^t = (\mathbf{m}_j^t, \mathbf{m}_{ij}^t)$  including:

$$\mathbf{m}_j^t = F_{m1}(\mathbf{h}_j^t), \quad \mathbf{m}_{ij}^t = F_{m2}(\mathbf{h}_{ij}^t), \quad (2)$$

where  $\mathbf{m}_j^t$  is a unary message which captures the identity of agent  $j$  (*e.g.*, the local object content), and  $\mathbf{m}_{ij}^t$  is a pairwise message which models the interaction between two agents (*e.g.*, the relative spatial layout).  $\mathbf{h}_{ij}^t$  is the pairwise feature between agent  $i$  and  $j$ , and its initialization is the union box feature extracted by object detector.  $F_{m*}$  are message composition functions<sup>1</sup>. All communication message between agent  $i$  and the others (*i.e.*,  $\{M_{i*}^t\}$ ) and its internal state  $\mathbf{h}_i^t$  are fed into the following update module to update the time-step feature for next round agent communication.

**Update Module.** At each round communication, we use a soft-attention to fuse message from the other agents:

$$\begin{aligned} \alpha_j^t &= F_{att1}(\mathbf{h}_i^t, \mathbf{h}_j^t), \quad \alpha_{ij}^t = F_{att2}(\mathbf{h}_i^t, \mathbf{h}_{ij}^t), \\ \mathbf{x}_i^{t+1} &= F_{u1}(\mathbf{h}_i^t, \{\alpha_j^t \mathbf{m}_j^t\}, \{\alpha_{ij}^t \mathbf{m}_{ij}^t\}), \\ \mathbf{h}_{ij}^{t+1} &= F_{u2}(\mathbf{h}_{ij}^t, \mathbf{h}_i^t, \mathbf{h}_j^t), \end{aligned} \quad (3)$$

where  $\alpha_j^t$  and  $\alpha_{ij}^t$  are attention weights to fuse different message,  $F_{att*}$  and  $F_{u*}$  are attention and update functions<sup>1</sup>.

**Discussion.** We dubbed the communication step as agent communication instead of the conventional message passing [47, 23] for two main reasons: 1) To be consistent with the concept of the multi-agent framework, where agent communication represents passing message among agents. 2) To highlight the difference with existing message passing methods that our communication model disentangles the relationship prediction from the visual context modeling.

### 3.1.3 Visual Relationship Detection

$$\text{Input : } \{(s_i^T, h_i^T)\} \longmapsto \text{Output : } \{r_{ij}\}$$

<sup>1</sup>For conciseness, we leave the details in supplementary material.

After  $T$ -round agent communication, all agents finish their states update. In inference stage, we greedily select the object labels  $v_i^T$  based on confidence  $s_i^T$ . Then the relation model predict the relationship class for any object pairs:

$$r_{ij} = F_r(\mathbf{h}_i^T, \mathbf{h}_j^T, v_i^T, v_j^T), \quad (4)$$

where  $F_r$  is the relationship function<sup>1</sup>. After predicting relationship for all object pairs, we finally obtain the generated scene graph:  $(\{v_i^T\}, \{r_{ij}\})$ .

## 3.2 Counterfactual Critic Multi-Agent Training

We demonstrate the details of the training objective of CMAT, including: 1) multi-agent policy gradient for the graph-coherent objective, and 2) counterfactual critic for the local-sensitive objective. The dataflow of our CMAT in training stage is shown in Figure 2 ( $\Rightarrow$  path).

### 3.2.1 Graph-Coherent Training Objective

Almost all prior SGG works minimize the XE loss as the training objective. Given a generated scene graph  $(\hat{\mathcal{V}}, \hat{\mathcal{E}})$  and its ground-truth  $(\mathcal{V}^{gt}, \mathcal{E}^{gt})$ , the objective is:

$$L(\theta) = \sum_{ij} (\text{XE}(\hat{v}_i, v_i^{gt}) + \text{XE}(\hat{r}_{ij}, r_{ij}^{gt})). \quad (5)$$

As can be seen in Eq. (5), the XE based objective is essentially independent and penalizes errors at all nodes equally.

To address this problem, we propose to replace XE with the following two graph-level metrics for graph-coherent training objective of SGG: 1) **Recall@K** [29]: It computes the fraction of the correct predicted triplets in the top  $K$  confident predictions. 2) **SPICE** [1]: It is the F-score of predicted triplets precision and triplets recall. Different from the XE loss, both Recall@K and SPICE are non-differentiable. Thus, our CMAT resorts to using the multi-agent policy gradient to optimize these objectives.

### 3.2.2 Multi-Agent Policy Gradient

We first describe formally the *action*, *policy* and *state* in CMAT, then derive the expression of parameter gradients.

**Action.** The action space for each agent is the set of all possible object classes, *i.e.*, at time step  $t$ , the action of agent  $i$  is  $v_i^t$ . We denote  $V^t = \{v_i^t\}$  as the set of actions of all agents.

**State.** We follow previous work [10] to use an LSTM (extract module) to encode the interaction history of each agent. The hidden state  $\mathbf{h}_i^t$  can be regarded as an approximation of the partially-observable environment state for agent  $i$ . We denote  $H^t = \{\mathbf{h}_i^t\}$  as the set of states of all agents.

**Policy.** The stochastic policy for each agent is the object classifier. In the training stage, the action is sampled based on the object class distribution, *i.e.*,  $p_i^T = \text{softmax}(s_i^T)$ .

Because our CMAT only samples actions for each agents after  $T$ -round agent communication, based on the policy gradient theorem [43], the (stochastic) gradient for the cooperative multi-agent in CMAT is:

$$\nabla_{\theta} J \approx \sum_{i=1}^n \nabla_{\theta} \log p_i^T(v_i^T | h_i^T; \theta) Q(H^T, V^T), \quad (6)$$

where  $Q(H^T, V^T)$  is the state-action value function. Instead of learning an independent network to fit the function  $Q$  and approximate reward like actor-critic works [2, 28, 18]; in our CMAT, we follow [35] to directly use the real global reward to replace  $Q$ . The reasons are as follows: 1) The number of agents and possible actions for each agent in SGG are much larger than the previous multi-agent policy gradient settings, thus the number of training samples is insufficient to train an accurate value function. 2) This can reduce the model complexity and speed up the training procedures. Thus, the gradient for our CMAT becomes:

$$\nabla_{\theta} J \approx \sum_{i=1}^n \nabla_{\theta} \log p_i^t(v_i^T | h_i^T; \theta) R(H^T, V^T), \quad (7)$$

where  $R(H^T, V^T)$  is the real graph-level reward (*i.e.*, Recall@K or SPICE). It is worth noting that the reward  $R(H^T, V^T)$  is a learnable reward function which includes a relation detection model.

### 3.2.3 Local-Sensitive Training Objective

As can been seen in Eq. (7), the graph-level reward can be considered as a global pooling contribution from all the local predictions, *i.e.*, the reward for all the  $n$  agents are identical. We demonstrate the negative impact of this situation with a toy example as shown in Figure 4.

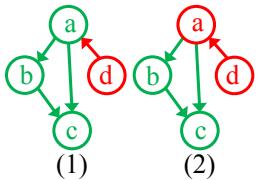


Figure 4: (1)(2) are two generated scene graph results. The color green and red represents right and wrong prediction, respectively. The graph-level reward for this toy example is the number of right predicted triplets minus the number of wrong predicted triplets.

Suppose all predictions of two generated scene graph are identical, except the prediction of node “a” is different. Based on Eq. (7), all nodes in the first graph and second graph get a positive reward (*i.e.*, 3 (right) -1(wrong) = +2) and a negative reward (*i.e.*, 1 (right)-3 (wrong) = -2), respectively. The predictions for the nodes “b”, “c”, and “d” are identical in the two graphs, but their gradient directions for optimization are totally different, which results in many inefficient optimization iteration steps. Thus, the training objective of SGG should be local-sensitive, *i.e.*, it can identify the contribution of each local prediction to provide an efficient training signal for each agent.

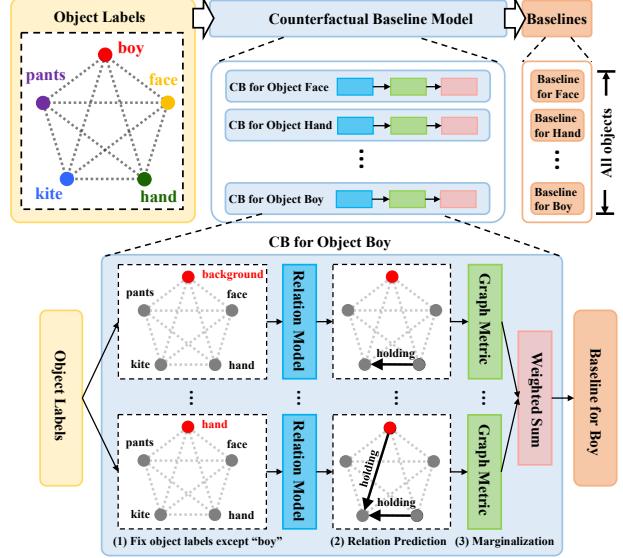


Figure 5: The illustration of the counterfactual baseline (CB) model in Figure 2. For this given image, the model calculates CB for all agents (*e.g.*, boy, face, hand, pants, and kite). As shown in the bottom, for the CB for boy, we traverse to replace class label boy to all possible classes (*e.g.*, background, ..., hand *et al.*) and marginalize these rewards.

### 3.2.4 Counterfactual Critic

An intuitive solution, for identifying the contribution of a specific agent’s action, is to replace the default action of the target agent with other actions. Formally,  $R(H^T, V^T) - R(H^T, (V_{-i}^T, \tilde{v}_i^T))$  can reflect the true influence of action  $\tilde{v}_i^T$ , where  $V_{-i}^T$  represents all agents except agent  $i$  (*i.e.*, the others  $n - 1$  agents) using the default action, and agent  $i$  takes a new action  $\tilde{v}_i^T$ .

Since the new action  $\tilde{v}_i^T$  for agent  $i$  has  $|\mathcal{C}|$  choices, and we can obtain totally different results for  $R(H^T, (V_{-i}^T, \tilde{v}_i^T))$  with different action choices. To more precisely approximate the individual reward of the default action of agent  $i$  (*i.e.*,  $v_i^T$ ), we marginalize the rewards when agent  $i$  traverse all possible actions:  $CB^i(H^T, V^T) = \sum p_i^T(\tilde{v}_i^T) R(H^T, (V_{-i}^T, \tilde{v}_i^T))$ , where  $CB^i(H^T, V^T)$  is the **counterfactual baseline** for the action of agent  $i$ . The counterfactual baseline represents the average global-level reward that the model should received when all other agents take default actions and regardless of the action of agent  $i$ . The illustration of counterfactual baseline model (Figure 2) in CMAT is shown in Figure 5.

Given the global reward  $R(H^T, V^T)$  and counterfactual baseline  $CB^i(H^T, V^T)$  for action  $v_i^T$  of agent  $i$ , the disentangled contribution of the action of agent  $i$  is:

$$A^i(H^T, V^T) = R(H^T, V^T) - CB^i(H^T, V^T). \quad (8)$$

Note that  $A^i(H^T, V^T)$  can be considered as the *advan-*

tage in actor-critic methods [42, 30],  $\text{CB}^i(H^T, V^T)$  can be regarded as a *baseline* in policy gradient methods, which reduces the variance of gradient estimation. The whole network to calculate  $A^i(H^T, V^T)$  is dubbed as the *counterfactual critic* (Figure 2). Then the gradient becomes:

$$\nabla_{\theta} J \approx \sum_{i=1}^n \nabla_{\theta} \log \mathbf{p}_i^T(v_i^T | h_i^T; \theta) A^i(H^T, V^T). \quad (9)$$

Finally, we incorporate the auxiliary XE supervised loss (weighted by a trade-off  $\alpha$ ) for an end-to-end training, and the overall gradient is:

$$\begin{aligned} \nabla_{\theta} J \approx & \underbrace{\sum_{i=1}^n \nabla_{\theta} \log \mathbf{p}_i^T(v_i^T | h_i^T; \theta) A^i(H^T, V^T)}_{\text{CMAT}} + \\ & \underbrace{\alpha \sum_{i=1}^n \sum_{j=1}^n \nabla_{\theta} \log \mathbf{p}_{ij}(r_{ij})}_{\text{XE for relationships}} + \underbrace{\alpha \sum_{i=1}^n \nabla_{\theta} \log \mathbf{p}_i^T(v_i^T)}_{\text{XE for objects}}, \end{aligned} \quad (10)$$

where CMAT encourages visual context exploration and XE stabilizes the training [35]. We also follow [48, 13] to add an entropy term to regularize  $\{\mathbf{p}_i^T\}$ .

**Discussion.** Although the critic in CMAT is not a value function to estimate the reward as in actor-critic [42], we dubbed it as critic for two main reasons: 1) The essence of a critic is calculating advantages for the actions of policy network. As in previous policy gradient work [39], the critic can be an inference algorithm without a value function. 2) The critic in CMAT includes a learnable relation model, which will also update its parameters at the training stage.

## 4. Experiments

**Dataset.** We evaluated our method for SGG on the challenging benchmark: Visual Genome (VG) [20]. For fair comparisons, we used the released data preprocessing and splits which had been widely-used in [47, 54, 31, 49, 12]. The release selects the most frequent 150 object categories and 50 predicate classes. After preprocessing, each image has 11.5 objects and 6.2 relationships on average. The released split uses 70% of images for training (including 5K images as validation set) and 30% of images for test.

**Settings.** As the conventions in [47, 54, 14], we evaluate SGG on three tasks: **Predicate Classification** (PredCls): Given the ground-truth object bounding boxes and class labels, we need to predict the visual relationship classes among all the object pairs. **Scene Graph Classification** (SGCls): Given the ground-truth object bounding boxes, we need to predict both the object and pairwise relationship classes. **Scene Graph Detection** (SGDet): Given an image, we need to detect the objects and predict their pairwise relationship classes. In particular, the object detection

needs to localize both the subject and object with at least 0.5 IoU with the ground-truth. As the conventions in [54, 14], we used Recall@20 (R@20), Recall@50 (R@50), and Recall@100 (R@100) as the evaluation metrics.

### 4.1. Implementation Details

**Object Detector.** For fair comparisons with previous works, we adopted the same object detector as [54]. Specifically, the object detector is a Faster-RCNN [37] with VGG backbone [41]. Moreover, the anchor box size and aspect ratio are adjusted similar to YOLO-9000 [36], and the RoIPooling layer is replaced with the ROIAlign layer [11].

**Training Details.** Following the previous policy gradient works that use a supervised pre-training step as model initialization (*aka*, teacher forcing), our CMAT also utilized this two-stage training strategy. In the supervised training stage, we froze the layers before the ROIAlign layer and optimized the whole framework with the sum of objects and relationships XE losses. The batch size and initial learning rate were set to 6 and  $10^{-3}$ , respectively. In the policy gradient training stage, the initial learning rate is set to  $3 \times 10^{-5}$ . For SGDet, since the number of all possible relationship pairs are huge (*e.g.*, 64 objects leads to  $\approx 4,000$  pairs), we followed [54] that only considers the relationships between two objects with overlapped bounding boxes, which reduced the number of object pairs to around 1,000.

**Speed vs. Accuracy Trade-off.** In the policy gradient training stage, the complete counterfactual critic calculation needs to sum over all possible object classes, which is significantly time-consuming (over 9,600 ( $\approx 151 \times 64$ ) times graph-level evaluation at each iteration). Fortunately, we noticed that only a few classes for each agent have large prediction confidence. To make a trade-off between training speed and accuracy, we only sum over the two highest positive classes and the background class probabilities to estimate the counterfactual baseline. In our experiments, this approximation only results in a slight performance drop but 70x faster training time.

**Post-processing for SGDet.** For SGDet, we followed the post-processing step in [54] for a fair comparison. After predicting the object class probabilities for each ROI, we used a per-class NMS to select the ROI class and its corresponding class-specific offsets from Faster-RCNN. The IoU threshold in NMS was set to 0.5 in our experiments.

### 4.2. Ablative Studies

We run a number of ablations to analyze CMAT, including the graph-level reward choice (for graph-coherent characteristic), the effectiveness of counterfactual baseline (for local-sensitive characteristic), and the early saturation problem in agent communication model. Results are shown in Table 2 and discussed in detail next.

**Graph-level Reward Choices.** To investigate the influence

	Model	SGDet			SGCls			PredCls			Mean
		R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100	
Constraint	VRD [29]	-	0.3	0.5	-	11.8	14.1	-	27.9	35.0	14.9
	IMP [47]	-	3.4	4.2	-	21.7	24.4	-	44.8	53.0	25.3
	MSDN [23, 49]	-	7.0	9.1	-	27.6	29.9	-	53.2	57.9	30.8
	AsscEmbed [31]	6.5	8.1	8.2	18.2	21.8	22.6	47.9	54.1	55.4	28.3
	FREQ+ <sup>◦</sup> [54]	20.1	26.2	30.1	29.3	32.3	32.9	53.6	60.6	62.2	40.7
	IMP+ <sup>◦</sup> [47, 54]	14.6	20.7	24.5	31.7	34.6	35.4	52.7	59.3	61.3	39.3
	TFR [14]	3.4	4.8	6.0	19.6	24.3	26.6	40.1	51.9	58.3	28.7
	MOTIFS <sup>◦</sup> [54]	21.4	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1	43.7
	Graph-RCNN [49]	-	11.4	13.7	-	29.6	31.6	-	54.2	59.1	33.2
	<b>CMAT</b>	<b>22.1</b>	<b>27.9</b>	<b>31.2</b>	<b>35.9</b>	<b>39.0</b>	<b>39.8</b>	<b>60.2</b>	<b>66.4</b>	<b>68.1</b>	<b>45.4</b>
No constraint	AsscEmbed [31]	-	9.7	11.3	-	26.5	30.0	-	68.0	75.2	36.8
	IMP+ <sup>◦</sup> [47, 54]	-	22.0	27.4	-	43.4	47.2	-	75.2	83.6	49.8
	FREQ+ <sup>◦</sup> [54]	-	28.6	34.4	-	39.0	43.4	-	75.7	82.9	50.6
	MOTIFS <sup>◦</sup> [54]	22.8	30.5	35.8	37.6	44.5	47.7	66.6	81.1	88.3	54.7
	<b>CMAT</b>	<b>23.7</b>	<b>31.6</b>	<b>36.8</b>	<b>41.0</b>	<b>48.6</b>	<b>52.0</b>	<b>68.9</b>	<b>83.2</b>	<b>90.1</b>	<b>57.0</b>

Table 1: Performance (%) compared with the state-of-the-art methods w/o graph constraint on VG [20]. Since some works doesn't evaluate on R@20, we compute the mean on all tasks over R@50 and R@100. <sup>◦</sup> denotes the methods using the same object detector as ours.

	XE	R@20	SPICE
SGCls	R@20	34.08	<b>35.93</b>
	SPICE	15.39	<b>16.01</b>
SGDet	R@20	16.23	<b>16.53</b>
	SPICE	7.48	<b>7.66</b>

(a) Results (%) of different reward choices.

	XE	MA	SC	CF
SGCls	R@20	34.08	34.76	34.68
	R@50	36.90	37.58	37.54
	R@100	37.61	38.29	38.25
SGDet	R@20	16.23	16.07	16.37
	R@50	20.62	20.41	20.82
	R@100	23.24	23.02	23.41

(b) Results (%) of different baseline types.

	2-step	3-step	4-step	5-step
SGCls	R@20	35.09	35.25	35.40
	R@50	37.95	38.19	38.37
	R@100	38.67	38.91	39.09
SGDet	R@20	16.35	16.43	16.47
	R@50	20.89	20.88	20.92
	R@100	23.49	23.50	23.54

(c) Results (%) of different #communication steps.

Table 2: **Ablations.** All results are with graph constraints. XE: The initialization performance after supervised XE pre-training. For clarity, the results of SGDet without post-processing are shown.

of choosing different graph-level metrics as the training reward, we compared two metrics: **Recall@K** and **SPICE**. In particular, we used the top-20 confident triplets as the predictions to calculate Recall and SPICE. The results are shown in Table 2 (a). We can observe that using both Recall and SPICE as the training reward can consistently improve the XE pre-trained model, because **the graph-level metrics is a graph-coherent objective**. Meanwhile, using Recall@K as training reward can always get slightly better performance than SPICE, because SPICE is not a suitable evaluation metric for the incomplete annotation nature of VG. Therefore, we used Recall@K as our training reward in the rest of the experiments.

**Policy Gradient Baselines.** To evaluate the effectiveness of our counterfactual baseline (**CF**), we compared it with other two widely-used baselines in policy gradient: Moving Average (**MA**) [46] and Self-Critical (**SC**) [39]. MA is a moving average constant over the recent rewards [48, 13]. SC is the received reward when model directly takes greedy actions as in the test. From Table 2 (b), we can observe that our CF baseline consistently improves the supervised initialization and outperforms others. Meanwhile, MA and SC can only improve the performance slightly or even worsen it. Because **the CF baseline is a local-sensitive objective**

and provides a more effective training signal for each agent, while MA and SC baselines are only globally pooling rewards which are still not local-sensitive.

**# of Communication Steps.** To investigate the early saturation issue in message passing models [47, 23], we compared the performance of CMAT with different numbers of communication steps from 2 to 5. From Table 2 (c), we can observe the trend seems contiguously better with the increase of communication step. Due to the GPU memory limit, we conducted experiments up to 5 steps. Compared with existing message passing methods, the reason why CMAT can avoid the early saturation issue is that our agent communication model discards the widely-used relationship nodes.

### 4.3. Comparisons with State-of-The-Arts

**Settings.** We compared our CMAT with the state-of-the-art SGG models. According to whether the model encodes context, we group these methods into: 1) VRD [29], AsscEmbed [31], FREQ [54] are all independent inference models, which detect object and predict visual relationship classes independently. 2) MSDN [23], IMP [47], TFR [14], MOTIFS [54], Graph-RCNN [49] are all joint inference models, which adopt message passing to encode the context. All these models are optimized by XE based training objective.

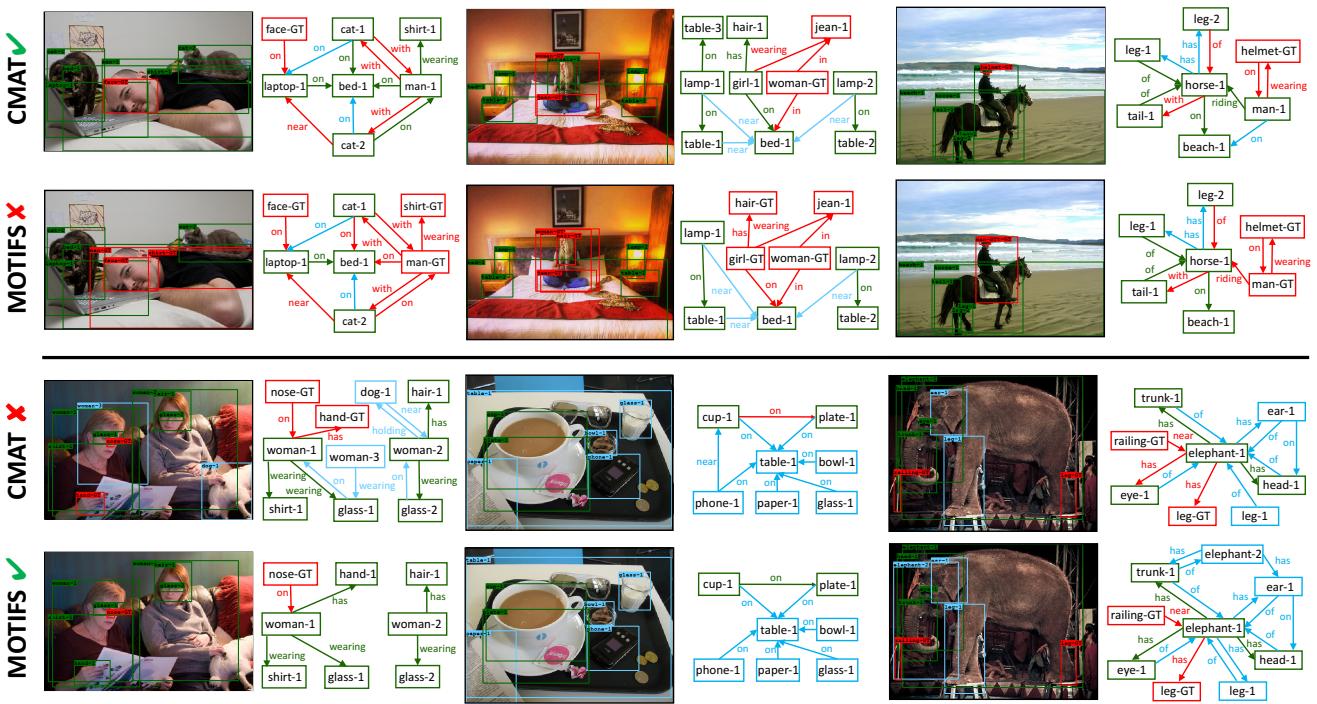


Figure 6: Qualitative results showing comparisons between higher R@20 (green tick) and lower R@20 (red cross) by CMAT and MOTIFS in SGDet. Green boxes are detected boxes with IoU large 0.5 with the ground-truth, blue boxes are detected but not labeled, red boxes are ground-truth with no match. Green edges are true positive predicted by each model at the R@20 setting, red edges are false negatives, and blue edges are false positives. Only detected boxes overlapped with ground-truth are shown.

**Quantitative Results.** The quantitative results are reported in Table 1. From Table 1, we can observe that our CMAT model achieves the state-of-the-art performance under all evaluation metrics. It is worth noting that CMAT can especially improve the performance of SGCLs significantly (*i.e.*, 3.4% and 4.3% absolute improvement in with and without graph constraint setting respectively), which means our CMAT model can substantially improve the object label predictions compared to others. The improvements in object label predictions meet our CMAT design, where the action of each agent is predicting an object label. Meanwhile, it also demonstrates the effectiveness of counterfactual critic multi-agent training for message passing models (agent communication) compared with XE based training. For PredCLs task, although we use the easiest visual relationship model and it still achieves best performances, which means the input for relationship model (*i.e.*, the state of agent) can better capture the internal state of each agent. Meanwhile, it is worth noting that any stronger relationship model can seamlessly be incorporated into our CMAT. For SGDet task, the improvements are not as significant as the SGCLs, the reason may come from the imperfect and noisy detected bounding boxes.

**Qualitative Results.** Figure 6 shows the qualitative results compared with strong baseline MOTIFS. From the results

in the top two rows, we can observe that CMAT rarely mistakes at the important hub nodes such as the man or girl. Because CMAT directly optimizes the graph-coherent objective. From the results in the bottom two rows, the mistakes of CMAT always come from the incomplete annotation of VG: CMAT can detect more false positive (the blue color) objects and relationship than MOTIFS. Since the evaluation metric (*i.e.*, Recall@K) is based on the ranking of labeled triplet confidence, thus, detecting more reasonable false positive results with high confidence can worsen the performance.

## 5. Conclusions

We proposed a novel method CMAT to address the inherent problem with XE based objective in SGG: it is not a graph-coherent objective. CMAT solves the problems by 1) formulating SGG as a multi-agent cooperative task, and using graph-level metrics as the training reward. 2) disentangling the individual contribution of each object to allow a more focused training signal. We validated the effectiveness of CMAT through extensive comparative and ablate experiments on the VG benchmark. Moving forward, we are going to 1) design a more effective graph-level metric to guide the CMAT training and 2) apply CMAT in downstream tasks such as visual Q&A, dialog, and captioning.

## References

- [1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. [2](#), [4](#)
- [2] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio. An actor-critic algorithm for sequence prediction. In *ICLR*, 2017. [5](#)
- [3] K. Chen, R. Kovvuri, and R. Nevatia. Query-guided regression network with context policy for phrase grounding. In *ICCV*, 2017. [3](#)
- [4] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017. [2](#), [3](#), [11](#)
- [5] A. Das, S. Kottur, J. M. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *ICCV*, 2017. [3](#)
- [6] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, 2009. [1](#)
- [7] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *NeurIPS*, 2016. [3](#)
- [8] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. In *AAAI*, 2018. [2](#)
- [9] J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *ICML*, 2017. [3](#)
- [10] M. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI*, 2015. [4](#)
- [11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. [1](#), [6](#)
- [12] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *NeurIPS*, 2018. [6](#)
- [13] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, 2017. [3](#), [6](#), [7](#)
- [14] S. Jae Hwang, S. N. Ravi, Z. Tao, H. J. Kim, M. D. Collins, and V. Singh. Tensorize, factorize and regularize: Robust visual relationship learning. In *CVPR*, 2018. [2](#), [3](#), [6](#), [7](#), [8](#)
- [15] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. [1](#)
- [16] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017. [3](#)
- [17] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015. [1](#)
- [18] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *NeurIPS*, 2000. [5](#), [12](#)
- [19] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, 2011. [2](#)
- [20] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowd-sourced dense image annotations. *IJCV*, 2017. [2](#), [6](#), [7](#)
- [21] Y. Li, W. Ouyang, X. Wang, and X. Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, 2017. [2](#)
- [22] Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *ECCV*, 2018. [2](#), [3](#)
- [23] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, 2017. [2](#), [3](#), [4](#), [7](#), [8](#)
- [24] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*, 2017. [3](#)
- [25] D. Liu, Z.-J. Zha, H. Zhang, Y. Zhang, and F. Wu. Context-aware visual policy network for sequence-level image captioning. In *ACM MM*, 2018. [3](#)
- [26] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy. Improved image captioning via policy gradient optimization of spider. In *ICCV*, 2017. [3](#)
- [27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. [1](#)
- [28] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, 2017. [2](#), [5](#)
- [29] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. [1](#), [2](#), [4](#), [7](#), [8](#)
- [30] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016. [6](#)
- [31] A. Newell and J. Deng. Pixels to graphs by associative embedding. In *NeurIPS*, 2017. [6](#), [7](#), [8](#)
- [32] W. Norcliffe-Brown, S. Vafeias, and S. Parisot. Learning conditioned graph structures for interpretable visual question answering. In *NeurIPS*, 2018. [1](#)
- [33] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *ICML*, 2017. [3](#)
- [34] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016. [3](#)
- [35] Y. Rao, D. Lin, J. Lu, and J. Zhou. Learning globally optimized object detector via policy gradient. In *CVPR*, 2018. [5](#), [6](#)
- [36] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. [6](#)
- [37] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. [1](#), [3](#), [6](#)
- [38] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*, 2017. [3](#)
- [39] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. [3](#), [6](#), [7](#)
- [40] J. Shi, H. Zhang, and J. Li. Explainable and explicit visual reasoning over scene graphs. In *CVPR*, 2019. [1](#)

- [41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6
- [42] R. S. Sutton, A. G. Barto, F. Bach, et al. *Reinforcement learning: An introduction*. MIT press, 1998. 6
- [43] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*, 2000. 2, 5, 12
- [44] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente. Multiagent cooperation and competition with deep reinforcement learning. In *arXiv*, 2015. 2, 3
- [45] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *CVPR*, 2017. 1
- [46] L. Weaver and N. Tao. The optimal reward baseline for gradient-based reinforcement learning. In *UAI*, 2001. 7
- [47] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017. 2, 3, 4, 6, 7, 8
- [48] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 6, 7
- [49] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018. 3, 6, 7, 8
- [50] X. Yang, H. Zhang, and J. Cai. Shuffle-then-assemble: learning object-agnostic visual relationship features. In *ECCV*, 2018. 1
- [51] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018. 1
- [52] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. C. Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018. 2, 3
- [53] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 3
- [54] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 2, 3, 6, 7, 8, 11
- [55] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 1, 2
- [56] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal. Relationship proposal networks. In *CVPR*, 2017. 2
- [57] L. Zhang, F. Sung, F. Liu, T. Xiang, S. Gong, Y. Yang, and T. M. Hospedales. Actor-critic sequence training for image captioning. In *NeurIPS Workshop*, 2017. 3
- [58] Y. Zhang, J. Hare, and A. Prügel-Bennett. Learning to count objects in natural images for visual question answering. In *ICLR*, 2018. 11
- [59] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2
- [60] Y. Zhu and S. Jiang. Deep structured learning for visual relationship detection. In *AAAI*, 2018. 2
- [61] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition for visual relationship detection. In *ICCV*, 2017. 2

## Appendix

This supplementary document is organized as follows:

- Section A provides the details of some simplified functions in Agent Communication and Visual Relationship Detection.
- Section B provides the detailed proof of the CMAT convergence, which guarantees that the proposed CMAT method can converge to a locally optimal policy.
- Section C provides the detailed derivation of Eq. (6), i.e.,  $\nabla_\theta J \approx \sum_{i=1}^n \nabla_\theta \log \mathbf{p}_i^T(v_i^T | h_i^T; \theta) Q(H^T, V^T)$ .
- Section D shows more qualitative results of CMAT compared with the strong baseline MOTIFS [54] in SGDet setting.

### A. Details of Some Simplified Functions

We demonstrate the details of some omitted functions in Eq. (1), Eq. (2), Eq. (3) and Eq. (4).

#### A.1. $F_s$ and $F_e$ in Extract Module

$$\begin{aligned} \mathbf{h}_i^t &= \text{LSTM}(\mathbf{h}_i^{t-1}, [\mathbf{x}_i^t, \mathbf{e}_i^{t-1}]), \\ \mathbf{s}_i^t &= \mathbf{s}_i^{t-1} + \mathbf{W}_h \mathbf{h}_i^t, \\ \mathbf{v}_i^t &\sim \mathbf{p}_i^t = \text{softmax}(\mathbf{s}_i^t), \\ \mathbf{e}_i^t &= \sum_{\tilde{v}} \mathbf{p}_i^t(\tilde{v}) \mathbf{E}[\tilde{v}], \end{aligned} \quad (11)$$

where  $\mathbf{h}_i^t \in \mathbb{R}^h$  is the hidden state of LSTM,  $\mathbf{x}_i^t \in \mathbb{R}^d$  is the time-step input feature and  $\mathbf{s}_i^t \in \mathbb{R}^{|\mathcal{C}|}$  is the object class confidence.  $\mathbf{E}[\tilde{v}] \in \mathbb{R}^e$  is the embedding of class label  $\tilde{v} \in \mathcal{C}$  and  $\mathbf{e}_i^t \in \mathbb{R}^e$  is the soft-weighted embedding of class label based on probabilities  $\mathbf{p}_i^t$ .  $\mathbf{W}_h \in \mathbb{R}^{h \times |\mathcal{C}|}$  is a learnable matrix. and  $[,]$  is concatenate operation.

#### A.2. $F_{m*}$ in Message Module

$$\mathbf{m}_j^t = \mathbf{W}_u \mathbf{h}_j^t, \quad \mathbf{m}_{ij}^t = \mathbf{W}_p \mathbf{h}_{ij}^t \quad (12)$$

where  $\mathbf{m}_j^t \in \mathbb{R}^h$  and  $\mathbf{m}_{ij}^t \in \mathbb{R}^h$  is the unary message and pairwise message, respectively.  $\mathbf{h}_{ij}^t \in \mathbb{R}^d$  is the pairwise feature between agent  $i$  and agent  $j$ .  $\mathbf{W}_u \in \mathbb{R}^{h \times h}$ ,  $\mathbf{W}_p \in \mathbb{R}^{d \times h}$  are learnable mapping matrices.

#### A.3. $F_{att*}$ and $F_{u*}$ in Update Module

$$\begin{aligned} u_j^t &= \mathbf{w}_u[\mathbf{h}_i^t, \mathbf{h}_j^t], \quad \alpha_j^t = \exp(u_j^t) / \sum_k \exp(u_k^t), \\ u_{ij}^t &= \mathbf{w}_p[\mathbf{h}_i^t, \mathbf{h}_{ij}^t], \quad \alpha_{ij}^t = \exp(u_{ij}^t) / \sum_k \exp(u_{ik}^t) \\ \mathbf{x}_i^{t+1} &= \mathbf{W}_x(\text{ReLU}(\mathbf{h}_i^t + \sum_j \alpha_j^t \mathbf{m}_j^t + \sum_j \alpha_{ij}^t \mathbf{m}_{ij}^t)) \\ \mathbf{h}_{ij}^{t+1} &= \text{ReLU}(\mathbf{h}_{ij}^t + \mathbf{W}_s \mathbf{h}_i^{t+1} + \mathbf{W}_e \mathbf{h}_j^{t+1}) \end{aligned} \quad (13)$$

where  $\alpha_j^t$  and  $\alpha_{ij}^t$  are attention weights to fuse different messages,  $\mathbf{w}_u \in \mathbb{R}^{2h}$ ,  $\mathbf{w}_p \in \mathbb{R}^{h+d}$ ,  $\mathbf{W}_x \in \mathbb{R}^{h \times d}$ ,  $\mathbf{W}_s \in \mathbb{R}^{h \times d}$ , and  $\mathbf{W}_e \in \mathbb{R}^{h \times d}$  are learnable mapping matrices.

#### A.4. $F_r$ in Visual Relationship Detection

$$\begin{aligned} \mathbf{z}_i &= \mathbf{W}_o[\mathbf{h}_i^T, \mathbf{E}[v_i^T]], \quad \mathbf{z}_j = \mathbf{W}_o[\mathbf{h}_j^T, \mathbf{E}[v_j^T]], \\ \mathbf{p}_{ij} &= \text{softmax}([\mathbf{z}_i, \mathbf{z}_j] \odot \mathbf{W}_r \mathbf{z}_{ij} + \mathbf{w}_{v_i^T, v_j^T}), \\ r_{ij} &= \arg \max_{r \in \mathcal{R}} \mathbf{p}_{ij}(r), \end{aligned} \quad (14)$$

where  $\mathbf{W}_o \in \mathbb{R}^{(h+e) \times z}$ ,  $\mathbf{W}_r \in \mathbb{R}^{z \times 2z}$  are transformation matrices,  $\mathbf{z}_{ij} \in \mathbb{R}^z$  is the predicated visual feature between agent  $i$  and  $j$ ,  $\odot$  is a fusing function<sup>2</sup>, and  $\mathbf{w}_{v_i^T, v_j^T} \in \mathbb{R}^{|\mathcal{C}|}$  is the bias vector specific to head and tail labels as in [54].

**Predicate Visual Features**  $\mathbf{z}_{ij}$ . For the predicate visual features, we used RoIAlign to pool the union box of subject and object, and resized the union box feature to  $7 \times 7 \times 512$ . Following [54, 4], we used a  $14 \times 14 \times 2$  binary feature map to model the geometric spatial position of subject and object, with one channel per box. We applied two convolutional layers on this binary feature map and obtained a new  $7 \times 7 \times 512$  spatial position feature map. We added this position feature map with the previous resized union box feature, and applied two fully-connected layers to obtain the final predicate visual feature.

---

<sup>2</sup>Different functions get comparable performance. In our experiments, we follow [58]:  $\mathbf{x} \odot \mathbf{y} = \text{ReLU}(\mathbf{W}_x \mathbf{x} + \mathbf{W}_y \mathbf{y}) - (\mathbf{W}_x \mathbf{x} - \mathbf{W}_y \mathbf{y})^2$ .

## B. Proof of the Convergence of CMAT

*Proof.* We denote  $\pi_i$  as the policy of agent  $i$ , *i.e.*,  $\pi_i = \mathbf{p}_i^T$  and  $\boldsymbol{\pi}$  as the joint policy of all agents, *i.e.*,  $\boldsymbol{\pi} = \{\mathbf{p}_1^T, \dots, \mathbf{p}_n^T\}$ . Then, the expected gradient of CMAT is given by (cf. Eq. (11)):

$$\begin{aligned}\nabla_{\theta} J &= \mathbb{E}_{\boldsymbol{\pi}} \left[ \sum_{i=1}^n \nabla_{\theta} \log \pi_i(v_i^T) A^i(H^T, V^T) \right], \\ &= \mathbb{E}_{\boldsymbol{\pi}} \left[ \sum_{i=1}^n \nabla_{\theta} \log \pi_i(v_i^T) (R(H^T, V^T) - b(H^T, V_{-i}^T)) \right].\end{aligned}\tag{15}$$

where the expectation  $\mathbb{E}_{\boldsymbol{\pi}}$  is with respect to the state-action distribution induced by the joint policy  $\boldsymbol{\pi}$ ,  $b(H^T, V_{-i}^T)$  is the counterfactual baseline in CMAT model, *i.e.*,  $b(H^T, V_{-i}^T) = \sum \mathbf{p}_i^T(\tilde{v}_i^T) R(H^T, (V_{-i}^T, \tilde{v}_i^T))$ .

First, consider the expected contribution of this counterfactual baseline  $b(H^T, V_{-i}^T)$ ,

$$\nabla_{\theta} J_b = \mathbb{E}_{\boldsymbol{\pi}} \left[ \sum_{i=1}^n \nabla_{\theta} \log \pi_i(v_i^T) b(H^T, V_{-i}^T) \right].\tag{16}$$

Let  $d^{\boldsymbol{\pi}}(s)$  be the discounted ergodic state distribution as defined by [43]:

$$\begin{aligned}\nabla_{\theta} J_b &= \sum_s d^{\boldsymbol{\pi}}(s) \sum_{i=1}^n \sum_{V_{-i}^T} \boldsymbol{\pi}(V_{-i}^T) \sum_{v_i^T} \pi_i(v_i^T) \nabla_{\theta} \log \pi_i(v_i^T) b(H^T, V_{-i}^T) \\ &= \sum_s d^{\boldsymbol{\pi}}(s) \sum_{i=1}^n \sum_{V_{-i}^T} \boldsymbol{\pi}(V_{-i}^T) \sum_{v_i^T} \nabla_{\theta} \pi_i(v_i^T) b(H^T, V_{-i}^T) \\ &= \sum_s d^{\boldsymbol{\pi}}(s) \sum_{i=1}^n \sum_{V_{-i}^T} \boldsymbol{\pi}(V_{-i}^T) b(H^T, V_{-i}^T) \nabla_{\theta} 1 \\ &= 0\end{aligned}\tag{17}$$

Thus, this counterfactual baseline does not change the expected gradient. The remainder of the expected policy gradient is given by:

$$\nabla_{\theta} J = \mathbb{E}_{\boldsymbol{\pi}} \left[ \sum_{i=1}^n \nabla_{\theta} \log \pi_i(v_i^T) R(H^T, V^T) \right],\tag{18}$$

$$= \mathbb{E}_{\boldsymbol{\pi}} \left[ \nabla_{\theta} \log \prod_{i=1}^n \pi_i(v_i^T) R(H^T, V^T) \right].\tag{19}$$

Writing the joint policy into a product of the independent policies:

$$\boldsymbol{\pi}(V^T) = \prod_{i=1}^n \pi_i(v_i^T),\tag{20}$$

we have the standard single-agent policy gradient:

$$\nabla_{\theta} J = \mathbb{E}_{\boldsymbol{\pi}} [\nabla_{\theta} \log \boldsymbol{\pi}(V^T) R(H^T, V^T)].\tag{21}$$

Konda *et al.* [18] proved that this gradient converges to a local maximum of the expected return  $J$ , given that: 1) the policy  $\boldsymbol{\pi}$  is differentiable, 2) the update timescales for  $\boldsymbol{\pi}$  are sufficiently slow. Meanwhile, the parameterization of the policy (*i.e.*, the single-agent joint-action learner is decomposed into independent policies) is immaterial to convergence, as long as it remains differentiable.  $\square$

## C. Derivation of Eq. (6)

Based on the policy gradient theorem we provide the detailed derivation of Eq. (6) as follows. We denote the action sequence for agent  $i$  as  $\hat{A}_i = \{\hat{a}_i^1, \hat{a}_i^2, \dots, \hat{a}_i^T\}$ , and value function  $V_\theta(\hat{A}_i)$  as the expected future reward of sequence  $\hat{A}_i$ . Then the gradient of agent  $i$  is:

$$\begin{aligned}
\nabla_\theta J_i &= \frac{dV(\hat{A}_i)}{d\theta} = \frac{d}{d\theta} \mathbb{E}_{\hat{A}_i \sim \pi_i^t(\hat{A}_i)} R(\hat{A}_i) \\
&= \sum_{\hat{A}_i} \frac{d}{d\theta} [\pi_i^t(\hat{a}_i^1) \pi_i^t(\hat{a}_i^2 | \hat{a}_i^1) \dots \pi_i^t(\hat{a}_i^T | \hat{a}_i^1 \dots \hat{a}_i^{T-1})] R(\hat{A}_i) \\
&= \sum_{t=1}^T \sum_{\hat{A}_i} \pi_i^t(\hat{A}_i^{1..t-1}) \frac{d\pi_i^t(\hat{a}_i^t | \hat{A}_i^{1..t-1})}{d\theta} \pi_i^t(\hat{A}_i^{t+1..T} | \hat{A}_i^{1..t}) R(\hat{A}_i) \\
&= \sum_{t=1}^T \sum_{\hat{A}_i^{1..t}} \pi_i^t(\hat{A}_i^{1..t-1}) \frac{\pi_i^t(\hat{a}_i^t | \hat{A}_i^{1..t-1})}{d\theta} \sum_{\hat{A}_i^{t+1..T}} \pi_i^t(\hat{A}_i^{t+1..T} | \hat{A}_i^{1..t}) \sum_{\tau=1}^T r_i^\tau(\hat{a}_i^\tau; \hat{A}_i^{1..\tau-1}) \\
&= \sum_{t=1}^T \sum_{\hat{A}_i^{1..t}} \pi_i^t(\hat{A}_i^{1..t-1}) \frac{\pi_i^t(\hat{a}_i^t | \hat{A}_i^{1..t-1})}{d\theta} \left[ r^t(\hat{a}_i^t; \hat{A}_i^{1..t-1}) + \sum_{\hat{A}_i^{t+1..T}} \pi_i^t(\hat{Y}_i^{t+1..T} | \hat{Y}_i^{1..t}) \sum_{\tau=t+1}^T r_i^\tau(\hat{a}_i^\tau; \hat{A}_i^{1..\tau-1}) \right] \\
&= \sum_{t=1}^T \mathbb{E}_{\hat{A}_i^{1..t-1} \sim \pi_i^t(\hat{A}_i^{1..t-1})} \sum_{a_i^t \in \mathcal{A}} \frac{d\pi_i^t(a_i^t | \hat{A}_i^{1..t-1})}{d\theta} Q(a_i^t; \hat{A}_i^{1..t-1}) \\
&= \mathbb{E}_{\hat{A}_i \sim \pi_i^t(\hat{A}_i)} \sum_{t=1}^T \sum_{a_i^t \in \mathcal{A}} \frac{\pi_i^t(a_i^t | \hat{A}_i^{1..t-1})}{d\theta} Q(a_i^t; \hat{A}_i^{1..t-1})
\end{aligned} \tag{22}$$

Further, the gradient for agent  $i$  can be simplified as:

$$\begin{aligned}
\nabla_\theta J_i &= \mathbb{E} \left[ \sum_{t=1}^T \sum_{a_i^t \in \mathcal{A}} \nabla_\theta \pi_i^t(a_i^t) Q(s_i^t, a_i^t) \right] \\
&= \mathbb{E} \left[ \sum_{t=1}^T \sum_{a_i^t \in \mathcal{A}} \pi_i^t(a_i^t) \nabla_\theta \log \pi_i^t(a_i^t) Q(s_i^t, a_i^t) \right] \\
&\approx \sum_{t=1}^T \nabla_\theta \log \pi_i^t(a_i^t) Q(s_i^t, a_i^t)
\end{aligned} \tag{23}$$

Therefore, for the time step  $t$ , the gradient for agent  $i$  is  $\nabla_\theta \log \pi_i^t(a_i^t) Q(s_i^t, a_i^t)$ . For multi-agent in a cooperative environment, the state-action function  $Q$  should estimate the reward based on the set of all agent state and actions, i.e.,  $Q(S^t, A^t)$ . Then, the gradient for all agents is:

$$\nabla_\theta J \approx \sum_{i=1}^n \nabla_\theta J_i = \sum_{i=1}^n \nabla_\theta \log \pi_i^t(a_i^t | s_i^t) Q(S^t, A^t). \tag{24}$$

In our CMAT, we CMAT samples actions after  $T$ -round agent communication, and the action for agent  $i$  is  $v_i^T$ , the policy function is  $p_i^T$ , and the state of agent is  $h^t$ , i.e.,  $S^t = H^t$ ,  $A^t = V^t$ . Therefore, the gradient for the cooperative multi-agent in CMAT is:

$$\nabla_\theta J \approx \sum_{i=1}^n \nabla_\theta \log p_i^T(v_i^T | h_i^T; \theta) Q(H^T, V^T) \tag{25}$$

## D. More Qualitative Results

Figure 7 and 8 show more qualitative results of CMAT and MOTIFS in SGDet setting. From the rows where CMAT is better than MOTIFS, we can see that CMAT rarely mistakes at the important hub nodes such as the “surfboard” or “laptop”.

This is because CMAT directly optimizes the graph-coherent objective. However, the rows show that the mistakes made by CMAT always come from the incomplete annotation of CMAT can detect more false positive (the blue color) objects and relationship than MOTIFS. Since the evaluation metric (i.e., Recall@K) is based on the ranking of labeled triplet confidence, thus, detecting more reasonable false positive results with high confidence can worsen the performance.

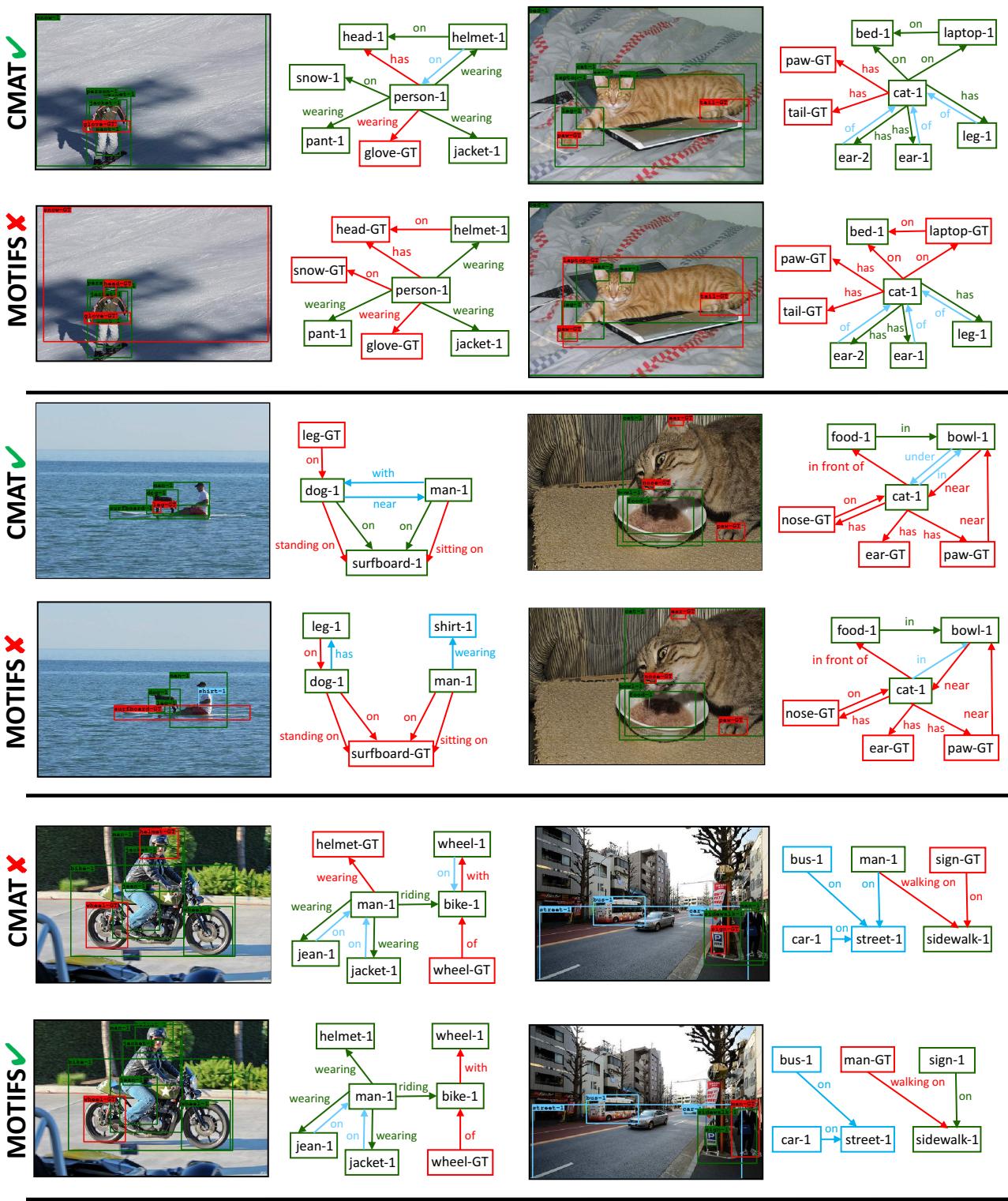


Figure 7: More qualitative results showing comparisons between CMAT and MOTIFS in the SGDet setting. Green boxes are detected boxes with IoU large 0.5 with the ground truth, blue boxes are detected but not labeled, red boxes are ground-truth with no match. Green edges are true positive predicted by each model at the R@20 setting, red edges are false negatives, and blue edges are false positives. Only detected boxes overlapped with ground-truth are shown.

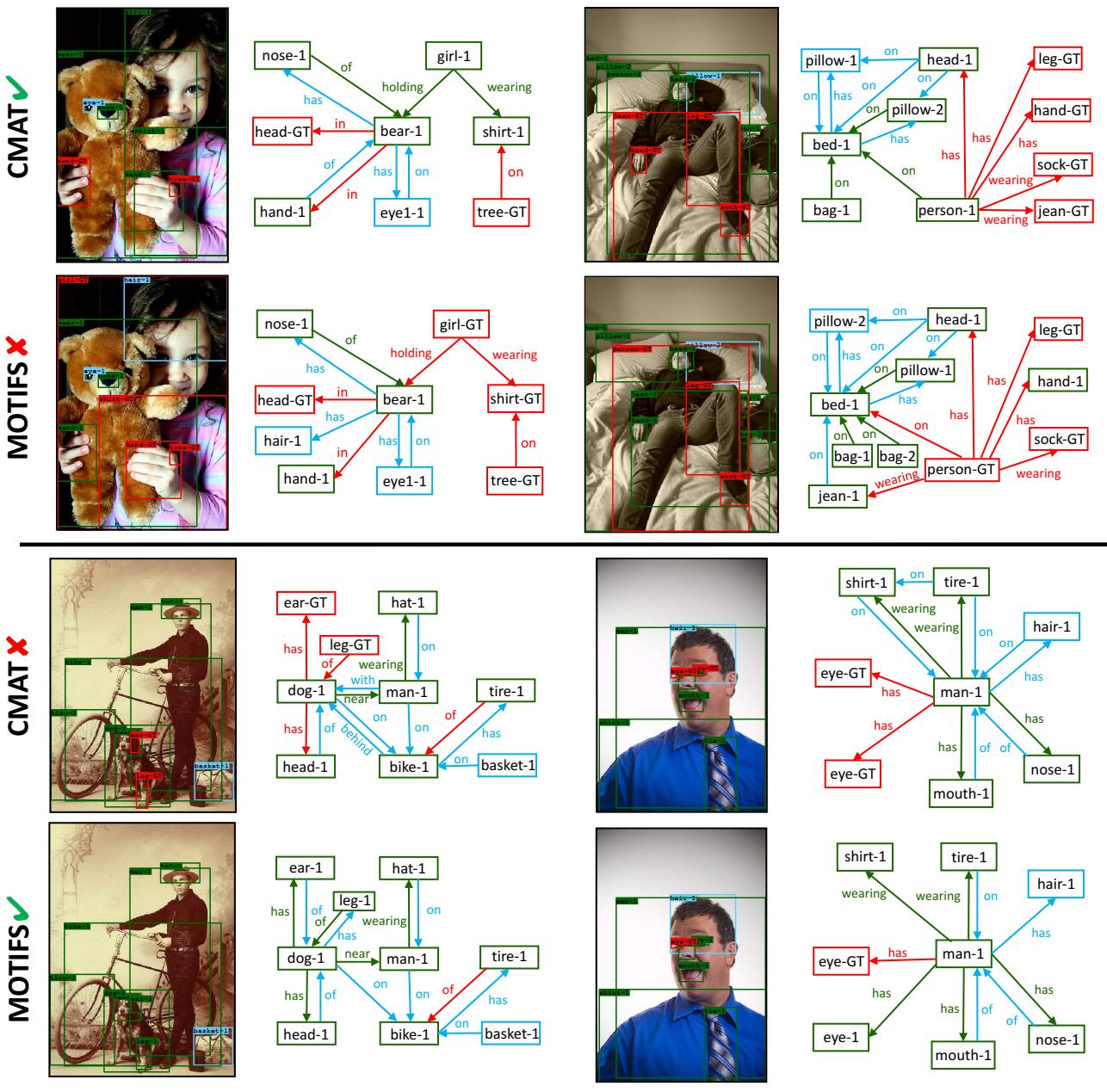


Figure 8: The caption is same as the one in Figure 8. More qualitative results showing comparisons between CMAT and MOTIFS in the SGDet setting. Green boxes are detected boxes with IoU large 0.5 with the ground truth, blue boxes are detected but not labeled, red boxes are ground-truth with no match. Green edges are true positive predicted by each model at the R@20 setting, red edges are false negatives, and blue edges are false positives. Only detected boxes overlapped with ground-truth are shown.