Exploiting Long-Term Dependencies for Generating Dynamic Scene Graphs

Shengyu Feng[1*] Hesham Mostafa[2] Marcel Nassar[2] Somdeb Majumdar[2] Subarna Tripathi[2]
[1]Carnegie Mellon University [2]Intel Labs
shengyuf@andrew.cmu.edu,
{hesham.mostafa, marcel.nassar, somdeb.majumdar, subarna.tripathi}@intel.com

Abstract

Dynamic scene graph generation from a video is challenging due to the temporal dynamics of the scene and the inherent temporal fluctuations of predictions. We hypothesize that capturing long-term temporal dependencies is the key to effective generation of dynamic scene graphs. We propose to learn the long-term dependencies in a video by capturing the object-level consistency and inter-object relationship dynamics over object-level long-term tracklets using transformers. Experimental results demonstrate that our Dynamic Scene Graph Detection Transformer *(DSG-DETR) outperforms state-of-the-art methods by a significant margin on the benchmark dataset Action Genome. Our ablation studies validate the effectiveness of each component of the proposed approach. The source code is available at* `https://github.com/Shengyu-Feng/DSG-DETR`.

1. Introduction

A scene graph is a directed graph where each node represents a labelled object and each edge represents an inter-object relationship, also known as a *predicate*. Learning visual relations in static images is a difficult problem due to its combinatorial nature. The underlying spatio-temporal dynamics and temporal fluctuations of predictions make the dynamic scene graph generation from video even harder. The naive solution to dynamic scene graph generation is simply applying the static scene graph generation method on each video frame without considering the temporal context. Recently a line of work [2, 5, 30, 20] emerged that demonstrated the importance of capturing the spatial as well as the temporal dependencies for dynamic scene graph generations.

The predominant ways to realize spatio-temporal consistencies focus on the construction of the spatio-temporal graph. Arnab *et al.* [2] construct a unified graph structure, utilizing a fully connected graph over the foreground nodes from the frames in a sliding window, and the connections between the foreground nodes and context nodes in each frame. Although this spatio-temporal graph can successfully perform message passing over both the spatial and temporal domains, such a fully connected graph is computationally expensive. In practice, such models resort to using a small sliding window consisting of 3 to 5 key frames, making them incapable of reasoning over long-term sequences. Another recent work, Spatial-Temporal Transformer (STTran) [5] grounds the model on the adjacent key frames. As a result, these methods can only achieve short-term consistency and fail to capture long-term dependencies.

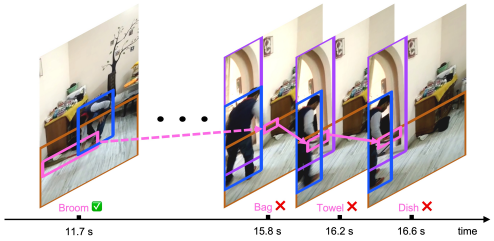


Figure 1: An example where the short-term temporal dependencies fail. The broom, bounded by the pink bounding boxes, is quite challenging to recognize on the rightmost three frames. Previous methods capturing only the short-term dependencies (indicated by the solid pink arrows) fail to make the correct prediction, while our method capturing the long-term dependencies (over more than 4 seconds), can recognize the broom and predict the human-broom relationship.

Fig 1 shows an example where occlusion and fast movement make it extremely difficult for any static image-based object detector to recognize the broom (bounded by the pink bounding box) in the rightmost three video frames. Any model that relies on capturing only short-term dependencies will fail to detect objects correctly in scenarios such as this example. This will result in incorrect dynamic scene graph generation. Predictions in frames, where an object might be

*Work partially done during an internship at Intel Labs.

occluded, may be improved by leveraging correct predictions from frames where it is easily detectable and recognizable. Since such "good" frames may be several frames away in the past or the future (4 seconds in this example), capturing long-term dependencies is crucial to improving the overall scene graph generation performance.

In this paper, we study the benefits of utilizing long-term temporal dependencies for objects and relations in dynamic scene graph generation tasks. We quantify it via estimating the hypothetical best case by exploiting the ground-truth object-tracks information when evaluated on scene graph generation tasks using Recall metric. This hypothetical best case significantly outperforms existing methods on dynamic scene graph generation. Next, we propose a paradigm for consistent video object detections without the access to ground truth object tracks. To this end, we construct the temporal sequences by tracking each object instance using the Hungarian matching algorithm [18], and apply a transformer encoder to leverage the temporal consistency within all such sequences. We also model the relationship transitions through the sequences of predicated subject-object classes using another transformer network. Our framework, named **DSG-DETR** (Dynamic Scene Graph Detection Transformer), performs comparably with the hypothetical best case explained above. The experimental results on the Action Genome dataset [14] also demonstrate that **DSG-DETR** can achieve significant improvements over the state-of-the-art methods for video scene graph generation. The main contributions of our work can be summarized as:

- We hypothesize that the key to improving dynamic scene graph generation is in capturing *long-term* temporal dependencies of objects and visual relationships.
- We quantify the benefit of capturing long-term dependencies by estimating the hypothetical best case (an upper-bound) on the scene graph generation task performances by utilizing the ground-truth object tracks.
- We then show that by capturing object consistencies and inter-object relationship dynamics within predicted object tracklets, our method DSG-DETR approaches the hypothetical best case performance.
- DSG-DETR significantly outperforms existing state-of-the-art methods on the video scene graph generation benchmark dataset Action Genome.

2. Related work

Scene graph generation. Scene graph generation (SGG) has become an important problem in computer vision since Johnson *et al.* [15] introduced the concept of graph-based image representation. A large body of work has focused on scene graph generation from images [33, 19, 29, 34, 23, 21]. These methods focus on either sophisticated architecture design or contextual feature fusion strategies, such as message passing or recurrent neural networks, to optimize SGG performance on the image scene graph benchmark dataset [17]. Such static SGG methods do not consider the dynamics of a video. Video SGG is significantly more challenging than image SGG due to the underlying spatio-temporal dynamics involving objects and inter-object relationships. A line of recent and concurrent work [2, 5, 30, 13] looks at the problem of video scene graph generation via modeling the spatio-temporal dynamics of relationships. While [2] takes an approach of message passing in a spatio-temporal graph for capturing the relationship dynamics, others rely on visual transformers for it. Some recent works [16, 7, 31] also utilize the tracking to boost the temporal context aggregation. [16] and [7] are based on the track-to-detect paradigm which first tracks the objects across the whole video and then figures out the pairwise relationship among tracklet pairs. However, this paradigm is highly sensitive to the tracking results and not flexible for the frame-level scene graph generation. TRACE [31], in contrast, utilizes a detect-to-track paradigm, but it is still limited to the short-term dependency and faces the aggregation problem of different prediction results for the same frame in different video segments. One concurrent work [20] leverages anticipatory prediction as pre-training and combines it with the fine-tuning strategies. Our coarse tracking method assimilates the complementary strengths of tracking based methods to allow the flexibility of long-term dependencies and frame-level prediction.

Transformer models in video analysis. Following the immense success of transformers [32] in natural language processing, they have been shown to be effective for image perception tasks [6, 3, 24] and video understanding tasks [28, 9, 8]. A recent work studies transformers for video SGG [5], which is also the theme of this paper. Another closely related problem is Human-Object-Interaction (HOI) detection from video where a Human-Object Relationship transformer has been utilized [13].

We hypothesize and experimentally show that the temporal fluctuation of object-level predictions hinders the performance of dynamic SGG tasks significantly. While [5] aims to capture the relationship dynamics via spatial encoder and temporal decoder, the impact of temporally consistent object predictions largely remain unaddressed. To the best of our knowledge, none of the methods aims to systematically capture the long-term dynamics at object-level. To summarize, we utilize a long-term temporal dependency via an online tracklet construction framework. Next, an object-centric transformer is employed on these sequences resulting in temporally consistent object recognition, followed by a spatio-temporal relationship transformer on the predicted sequences of the same subject-object classes.

3. Problem statement and notations

3.1. Dynamic scene graph generation

Given a video as a sequence of I key frames, we want to predict the objects for each frame, in terms of their positions and classes, and the relationships among them. Use $\mathcal{C}$ and $\mathcal{P}$ to denote the object class set and the predicate set respectively. We define each object as a tuple comprising its bounding box $\mathbf{b}$ and object class $\mathbf{c}$, i.e., $\mathcal{O} = \langle \mathbf{b}, \mathbf{c} \rangle$. Here, $\mathbf{b} \in [0,1]^4$ is a vector composed of the object center coordinates and its width and height relative to the image size. $\mathbf{c} \in \{0,1\}^{|\mathcal{C}|}$ is a one-hot vector with $\mathbf{c}[i] = 1$ and all other dimensions 0, where the i-th element of $\mathcal{C}$ corresponds to the class of this object.

The relationship tuple for a subject-object pair is defined as $\langle \mathcal{O}_s, p, \mathcal{O}_o \rangle$, which correspond to the subject, predicate and object respectively, and $p \in \mathcal{P}$. There could be multiple relationships for a subject-object pair and we represent these predicates as a vector $\mathbf{p} \in \{0,1\}^{|\mathcal{P}|}$, where $\mathbf{p}[i] = 1$ indicates the appearance of the i-th predicate in $\mathcal{P}$ and the corresponding relationship triplet $\langle \mathcal{O}_s, \mathcal{P}_i, \mathcal{O}_o \rangle$. Furthermore, we denote the distributions of the classes and predicates as $\tilde{\mathbf{c}} \in [0,1]^{|\mathcal{C}|}$ and $\tilde{\mathbf{p}} \in [0,1]^{|\mathcal{P}|}$, where $\sum_i \tilde{\mathbf{c}}[i] = 1$.

4. Methodology

In this section, we first identify the main challenges in modeling the temporal dynamics, then we discuss how DSG-DETR addresses them.

4.1. Temporal dynamics

Dynamic SGG requires reasoning over both spatial and temporal information. However, existing literature lacks a methodical analysis on what kind of information is needed for the temporal and spatial consistencies. In our experiments, we find the following aspects the main challenges for dynamic SGG.

Temporal object consistency. Static image based object detectors fail to detect video objects consistently due to factors like motion blur, fast movement, occlusion, compression artifacts, and temporal variation of predictions. Challenging cases like severe occlusion pose great difficulty in identifying an object from a single frame. We show that grounding the predictions over a long-term temporal context and enforcing it to be temporally consistent - i.e., avoiding sudden appearances or disappearances of object representations - results in more accurate and consistent object detections in video.

Temporal relationship transition. Besides the temporal object consistency, the other challenge for dynamic SGG is the temporal relation transition. Modeling relation transition allows different relationships among the same object-pair over time. We aim to maximize the conditional probability of a relationship given the previous relationships and the current observation.

Fig 2 describes the overall framework of DSG-DETR, which consists of an object transformer and a spatio-temporal relationship transformer, addressing the object consistency and relationship transition respectively.

4.2. Temporal object consistency

4.2.1 Online tracklet construction.

Unlike [2] that connects all objects in neighboring frames in a sliding window fashion, we only connect the objects which exhibit *apparent* similarity in either the visual feature or the spatial location. Our relatively sparser connections allow us to reason over long-term temporal contexts for a given computational and memory budget. To this end, we ground our method on a coarse tracking algorithm. It is worth noting that the purpose of our tracking is to **make the transformer only attend to the relevant features efficiently rather than directly extract the correct tracklets**, which is a significant difference between DSG-DETR and previous tracking based methods.

Prior to the construction of the tracklets, we pass all the frames in a video to Faster R-CNN [25] to obtain the object bounding boxes (if not available), object class distributions and object features. We use $\langle \mathbf{b}, \tilde{\mathbf{c}}, \mathbf{f} \rangle$ to denote detection, where $\mathbf{b}$ and $\tilde{\mathbf{c}}$ correspond to the bounding box and class distribution respectively and $\mathbf{f} \in \mathbb{R}^{2048}$ is the visual feature vector of the bounding box from Faster R-CNN.

Starting from the first frame, we iteratively match the detections with previous tracklets [10], where the tracking results in i-th frame could be represented as a permutation $\sigma_i(\cdot)$, for example, $\sigma_i(j)$ assigns the j-th detection in the i-th frame to $\sigma_i(j)$-th tracklet.

We denote the j-th detection in the i-th frame as $\mathcal{D}_{ij} = \langle \mathbf{b}_{ij}, \tilde{\mathbf{c}}_{ij}, \mathbf{f}_{ij} \rangle$, and the set of detections in the i-th frame as $\mathcal{D}_i = \{\mathcal{D}_{ij} | j\}$. We refer to the k-th tracklet up to the i-th frame as a set $\mathcal{T}_{(i-1)k} = \{\mathcal{D}_{i'j} | i' \leq i-1, j \leq |\mathcal{D}_{i'}|, \sigma_{i'}(j) = k\}$, which consists of all the detections matched to this tracklet in the previous frames. The set of tracklets up to the i-th frame is denoted as $\mathcal{T}_{i-1} = \{\mathcal{T}_{(i-1)k} | k\}$.

For each tracklet $\mathcal{T}_{(i-1)k}$, we define its position $\hat{\mathbf{b}}_{(i-1)k}$ as the bounding box of the last added detection, its class distribution and features as the average of the distributions and features over all detections in it:

$$\hat{\mathbf{c}}_{(i-1)k} = \frac{1}{|\mathcal{T}_{(i-1)k}|} \sum_{i' \leq i-1, j \leq |\mathcal{D}_{i'}|} \mathbb{1}_{[\sigma_{i'}(j)=k]} \tilde{\mathbf{c}}_{i'j} \quad (1)$$

$$\hat{\mathbf{f}}_{(i-1)k} = \frac{1}{|\mathcal{T}_{(i-1)k}|} \sum_{i' \leq i-1, j \leq |\mathcal{D}_{i'}|} \mathbb{1}_{[\sigma_{i'}(j)=k]} \mathbf{f}_{i'j}, \quad (2)$$

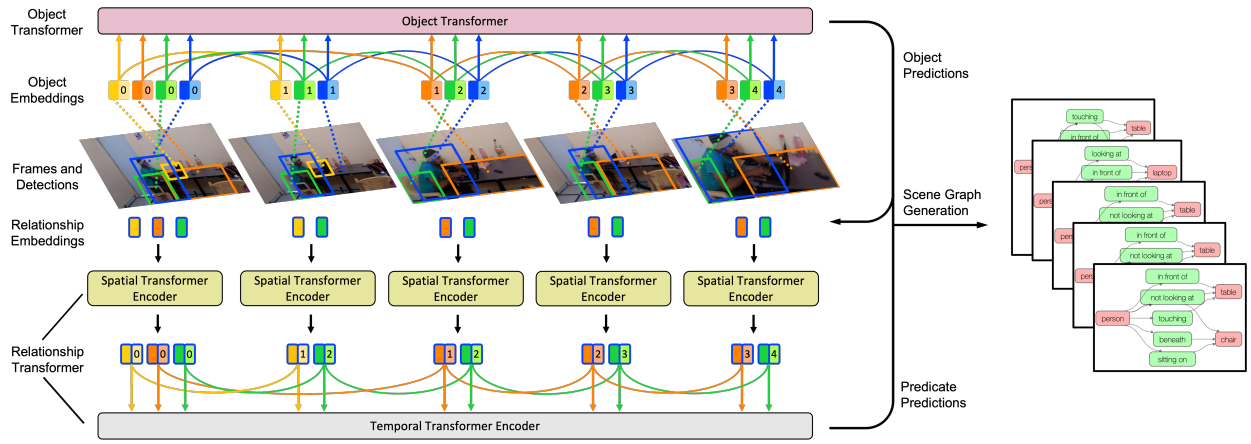


Figure 2: Visualization of the DSG-DETR model. The upper half visualizes the *object transformer* and the lower half sketches the *spatio-temporal relationship transformer*. Each color (orange, tangerine, green and blue) corresponds to one object. The token boxes without borderline represent object embeddings, and those with the borderline represent relationship embeddings, where the colors of the borderline and token box indicate the subject and object respectively. The token boxes with numbers inside are the positional encoding, e.g., the blue box with "1" inside denotes its position in the blue tracklet. The solid lines connecting the token boxes stand for the tracking results.

where $\mathbb{1}$ outputs 1 if the condition holds and 0 otherwise.

A tracklet is regarded as inactive if it does not get any new detection added in the past m frames. Here, m is the number of frames which corresponds to the time interval in the original Charades [27] videos which were annotated with scene graphs in Action Genome [14] dataset.

At the i-th frame, the detections of $\mathcal{D}_i$ are only matched with the active tracklets in $\mathcal{T}_{i-1}$. Let $n_i = \max\{|\mathcal{D}_i|, |\mathcal{T}_{i-1}|\}$, we pad with $\emptyset$ (empty set) for detections as $\mathcal{D}'_i = \{\mathcal{D}'_{ij} | j \leq n_i, \mathcal{D}'_{ij} = \mathcal{D}_{ij} \text{ if } j \leq |\mathcal{D}_i| \text{ else } \emptyset\}$ and active tracklets as $\mathcal{T}'_{i-1} = \{\mathcal{T}'_{(i-1)j} | j \leq n_i, \mathcal{T}'_{(i-1)j} = \mathcal{T}_{(i-1)j} \text{ if } j \leq |\mathcal{T}_{i-1}| \text{ and } \mathcal{T}_{(i-1)j} \text{ is active else } \emptyset\}$.

We use the Hungarian matching algorithm [18] to assign the detections to candidate tracklets based on their class distributions, features and positions, which aims to find the permutation of n_i elements $\sigma_i \in \mathfrak{S}_{n_i}$ with the lowest cost, where $\mathfrak{S}_{n_i}$ is the set of all permutations of size n_i:

$$\begin{aligned} &\arg\min_{\sigma_i \in \mathfrak{S}_{n_i}} \mathcal{L}_{\text{HM}}(\mathcal{D}'_i, \mathcal{T}'_{i-1}) = \\ &\arg\min_{\sigma_i \in \mathfrak{S}_{n_i}} \sum_{j=1}^{|D_i|} \mathbb{1}_{[\mathcal{T}'_{(i-1)\sigma_i(j)} \neq \emptyset]} [\mathcal{L}_{\text{dist}}(\tilde{\mathbf{c}}'_{ij}, \hat{\mathbf{c}}'_{(i-1)\sigma_i(j)}) \\ &+ \mathcal{L}_{\text{feat}}(\mathbf{f}'_{ij}, \hat{\mathbf{f}}'_{(i-1)\sigma_i(j)}) + \mathcal{L}_{\text{box}}(\mathbf{b}'_{ij}, \hat{\mathbf{b}}'_{(i-1)\sigma_i(j)})], \end{aligned} \tag{3}$$

where $\mathcal{L}_{\text{dist}}$, $\mathcal{L}_{\text{feat}}$ and $\mathcal{L}_{\text{box}}$ correspond to the loss of the class distributions, features and boxes respectively.

For the class distribution and feature losses, we use the cosine cost such that

$$\mathcal{L}_{\text{dist}}(\tilde{\mathbf{c}}'_{ij}, \hat{\mathbf{c}}'_{(i-1)\sigma_i(j)}) = (1 - \cos(\tilde{\mathbf{c}}'_{ij}, \hat{\mathbf{c}}'_{(i-1)\sigma_i(j)})) \tag{4}$$

$$\mathcal{L}_{\text{feat}}(\mathbf{f}'_{ij}, \hat{\mathbf{f}}'_{(i-1)\sigma_i(j)}) = \lambda_{\text{feat}}(1 - \cos(\mathbf{f}'_{ij}, \hat{\mathbf{f}}'_{(i-1)\sigma_i(j)})), \tag{5}$$

where $\cos(\cdot, \cdot)$ represents the cosine similarity and λ_{feat} is a non-negative scalar controlling the weight between the two loss components.

Following DETR [4], we combine the L_1 loss and the generalized IoU loss [26], denoted as $\mathcal{L}_{\text{iou}}(\cdot, \cdot)$, for the box loss:

$$\begin{aligned} \mathcal{L}_{\text{box}}(\mathbf{b}'_{ij}, \hat{\mathbf{b}}'_{(i-1)\sigma_i(j)}) &= \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(\mathbf{b}'_{ij}, \hat{\mathbf{b}}'_{(i-1)\sigma_i(j)}) \\ &+ \lambda_{L_1} \|\mathbf{b}'_{ij} - \hat{\mathbf{b}}'_{(i-1)\sigma_i(j)}\|_1, \end{aligned} \tag{6}$$

where λ_{iou} and λ_{L_1} control the weights of the generalized IoU loss and L_1 loss, respectively.

We create a new tracklet for an unmatched object, e.g., when $\mathcal{T}'_{(i-1)\sigma_i(j)} = \emptyset$. The Hungarian matching algorithm will always assign a detection to a tracklet, but it is not guaranteed that the detection indeed has a matching tracklet in $\mathcal{T}'_{i-1}$. For example, let's assume a case where the active tracklets correspond to two objects *person* and *table*; but the detections correspond to the objects *person* and *sofa*. The matching algorithm will match the *table* to *sofa*, although they are different. To mitigate this incorrect assignment problem, we ignore the matching if the cosine similarity between the features and class distributions are both less than a threshold τ. In such a case, we mark the corresponding tracklet as empty in the padded tracklet set and create a new tracklet for this detection.

Algorithm 1 Coarse tracking algorithm

Input data: Detections $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_I$ and video timestamps
Input hyperparameters: m, λ_{feat}, λ_{iou}, λ_{L_1} and τ
Let $\mathcal{T}_0 = \emptyset$
for iteration $i = 1, 2, \cdots, I$ **do**
  Construct the padded set $\mathcal{D}'_i$ and $\mathcal{T}'_{i-1}$ from $\mathcal{D}_i$ and $\mathcal{T}_{i-1}$
  Compute the optimal matching σ_i using Equation 3
  $k \leftarrow |\mathcal{T}_{i-1}| + 1$
  for iteration $j = 1, 2, \cdots, |\mathcal{D}_i|$ **do**
    if $\mathcal{T}'_{(i-1)\sigma_j(i)} \neq \emptyset$ and $\cos(\tilde{\mathbf{c}}'_{ij}, \hat{\mathbf{c}}'_{(i-1)\sigma_i(j)}) < \tau$ and $\cos(\tilde{\mathbf{f}}'_{ij}, \hat{\mathbf{f}}'_{(i-1)\sigma_i(j)}) < \tau$ **then**
      $\mathcal{T}'_{(i-1)\sigma_i(j)} \leftarrow \emptyset$
    end if
    if $\mathcal{T}'_{(i-1)\sigma_i(j)} = \emptyset$ **then**
      Create $\mathcal{T}_{ik} \leftarrow \{\mathcal{D}_{ij}\}$ and update $k \leftarrow k + 1$
    end if
  end for
  Update the tracklets in $\mathcal{T}_{i-1}$ according to Equation 7
  Update the tracklet set as $\mathcal{T}_i = \{\mathcal{T}_{ik'} | k' = 1, 2, \cdots, k-1\}$
end for
return: $\mathcal{T}_I$

Finally, the existing tracklets in $\mathcal{T}_{i-1}$ can be updated as

$$\mathcal{T}_{ik} = \mathcal{T}_{(i-1)k} \bigcup \{\mathcal{D}_{ij} | \sigma_i(j) = k, \mathcal{D}'_{ij} \neq \emptyset, \mathcal{T}'_{(i-1)k} \neq \emptyset\}. \tag{7}$$

The entire coarse tracking algorithm is summarized in Algorithm 1.

4.2.2 Object transformer for long-term consistency.

We build a transformer on top of these tracklets to realize the temporal object consistency. For each detection, we represent it as a concatenation of the box embedding, class distribution embedding and object features, which can be written as

$$\mathbf{o} = \text{Concat}(g^{\text{box}}(\mathbf{b}), g^{\text{dist}}(\tilde{\mathbf{c}}), \mathbf{f}), \tag{8}$$

where g^{box} and g^{dist} stand for the embedding functions of the box and class distribution respectively, and $\mathbf{o} \in \mathbb{R}^{d_o}$.

For each tracklet, $\mathcal{T}_{Ik} = \{\mathcal{D}_{ij} | \sigma_i(j) = k\}$, we represent all of its detections as a matrix $\mathbf{O}_k \in \mathbb{R}^{|\mathcal{T}_{Ik}| \times d_o}$. Then we apply an object transformer with positional encoding $PE(\cdot)$ followed by a feedforward network to output the new object class distributions $\tilde{\mathbf{C}}_k \in [0,1]^{|\mathcal{T}_{Ik}| \times |\mathcal{C}|}$ as:

$$\tilde{\mathbf{F}}_k = \text{Encoder}_{\text{object}}(\mathbf{O}_k + PE(\mathbf{O}_k)) \tag{9}$$

$$\tilde{\mathbf{C}}_k = \text{Softmax}(\text{FFN}(\tilde{\mathbf{F}}_k)). \tag{10}$$

The standard cross entropy loss $\mathcal{L}_{obj}$ is used for the object classification.

4.3. Temporal relationship transition

To model the relationship transition, we still ground our model on those tracklets, with their predicted subject-object classes, i.e., the relationships sharing the same subject-object classes [1] across the key frames are in the same sequence.

To simultaneously model the spatial dependency, we first feed all relationships into a spatial encoder, which aggregates the information in each frame, then we apply a temporal encoder for the same subject-object pairs across frames.

The relationships are defined over a detected subject-object pair $\langle \mathcal{D}_s, \mathcal{D}_o \rangle$. Similar to STTran [5], we represent the relationships as a combination of three embeddings, visual embedding, spatial embedding and semantic embedding:

$$\mathbf{r}^{vs} = \text{Concat}(g^s(\tilde{\mathbf{f}}_s), g^o(\tilde{\mathbf{f}}_o)) \tag{11}$$

$$\mathbf{r}^{sp} = g^{sp}(\mathbf{u}_{so} \oplus g^{\text{boxes}}(\mathbf{b}_s, \mathbf{b}_o) \tag{12}$$

$$\mathbf{r}^{se} = \text{Concat}(g^{se}(\mathbf{c}_s), g^{se}(\mathbf{c}_o)) \tag{13}$$

$$\mathbf{r} = \text{Concat}(\mathbf{r}^{vs}, \mathbf{r}^{sp}, \mathbf{r}^{se}), \tag{14}$$

where $\mathbf{r}^{vs}$, $\mathbf{r}^{sp}$, $\mathbf{r}^{se}$ correspond to the visual embedding, spatial embedding and semantic embedding, respectively. $\tilde{\mathbf{f}}$ is the spatial-temporal visual feature computed in Equation 9. g^s and g^o are the visual feature embedding functions for the subject and object. g^{sp} is the spatial embedding function, whose input is the sum of the union feature $\mathbf{u}_{so}$ for the subject and object extracted by ROIAlign [11] and a boxes embedding encoded by g^{boxes}, where $\oplus$ stands for the element-wise addition. g^{se} is the word embedding of the object class. Please refer to the Supplementary material for more details about each embedding function.

We stack all relationships $\mathbf{r}$ in the i-th frame into a matrix $\mathbf{R}_i$. The output of the spatial transformer encoder is:

$$\mathbf{R}'_i = \text{Encoder}_{\text{spatial}}(\mathbf{R}_i). \tag{15}$$

Then we rearrange all the output relationship representations $\mathbf{r}'$ according to their subject and object classes. We stack the relationship representations from the spatial encoder with the subject-object classes $\langle s, o \rangle$ into a matrix $\mathbf{R}'_{so}$, then the logits of the predicates are output by the temporal transformer encoder

$$\mathbf{Z}_{so} = \text{Encoder}_{\text{temporal}}(\mathbf{R}'_{so} + PE(\mathbf{R}'_{so})). \tag{16}$$

The predicates logits $\mathbf{z}$ of the corresponding relationship representation $\mathbf{r}'$ go through different linear projections to obtain the final predicates distribution $\tilde{\mathbf{p}}$ for different HOI

[1] In Action Genome [14], each object class is unique in one frame.

types belonging to *attention*, *spatial* and *contact* as defined in the Action Genome dataset [14]. We use the multi-label margin loss for the predicate classification,

$$\mathcal{L}_p(\tilde{\mathbf{p}}) = \sum_{i \in \mathcal{P}^+} \sum_{j \in \mathcal{P}^-} \max(0, 1 - \tilde{\mathbf{p}}[j] + \tilde{\mathbf{p}}[i]), \qquad (17)$$

where $\mathcal{P}^+$ denotes the indices of the annotated predicates and $\mathcal{P}^-$ denotes the indices of the predicates not in the annotation. The final loss combines both the object loss and predicate loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{obj} + \mathcal{L}_p$.

5. Experiments

5.1. Experimental setup

Dataset We evaluate our method on Action Genome dataset [14] containing 35 object categories and 25 relationship categories. The relationships are categorized into three human-object categories: attention, spatial and contact relationships, where multiple relationships may appear in spatial and contact categories.

Training We use one NVIDIA Tesla V100S GPU with 32G memory for training. Similar to [5], we utilize Faster R-CNN with Resnet101 [12] backbone for the object detector and pretrain it on Action Genome [14]. All layers in the backbone for the object detector feature extraction are frozen when training our method. We use AdamW [22] to optimize with batch size 1. The initial learning rate is set to 10^{-5}.

Evaluation The edge (predicate) classification conditioned on nodes (subject & object) is a relatively easy problem to solve. Most methods reach the ballpark of 99% in the unconstrained Recall@K=50 for this *PredCls* task. The problem becomes challenging when the joint object classification / detection is involved in SGCls / SGDet tasks. We evaluate performances on these two tasks. (1) scene graph classification (SGCls): where the video frames and bounding boxes are provided and the task is to predict the predicates and subject/object classes. (2) scene graph detection (SGDet): The task is to detect the objects and predict the predicates for object pairs, where only the video frames are provided. Following the convention of object detection, in case of SGDet, an entity (subject or object) is regarded as successfully detected if the Intersection-Over-Union (IOU) between the predicted bounding box and the ground-truth bounding box is larger than 0.5 and the predicted and ground-truth class labels match. Please refer to the supplementary material for additional results on PredCls. We use Recall@K (R@K, K=[10,20,50]) [23, 14] as the evaluation metric, which measures the fraction of the ground-truth relationship triplets in the top K predictions.

For the relationship tuple of a subject-object pair $\langle \mathcal{D}_s, \mathcal{D}_o \rangle$, we define the score of each detected object as the highest class score in its distribution, $\max\{\tilde{\mathbf{c}}\}$. Then the score of i-th relationship triplet is estimated as the product of three scores:

$$\max\{\tilde{\mathbf{c}}_s\} \cdot \tilde{\mathbf{p}}[i] \cdot \max\{\tilde{\mathbf{c}}_o\}. \qquad (18)$$

For the calculation of R@K, the relationship triplets are ordered according to their scores among all relationship triplets of that category in a frame.

5.2. Comparison with SOTA

Table 1 shows the main result of the proposed DSG-DETR. We use STTran [5] as our one of the strongest baselines, and develop our DSG-DETR atop their source code [1]. Besides, we also select some powerful scene graph generation methods on the static images such as VRD [23], M-FREQ [33], MSDN [29], RelDN [34] and GBS-Net [21]. For a fair comparison, we use the same object detector, a pretrained Faster R-CNN fine-tuned on Action Genome for all the baselines. Please note that the results for TRACE [31] correspond to the same evaluation criteria as with others; in their original paper the setup was different. The results show that DSG-DETR outperforms the strongest baseline of STTran in both SGCls and SGDet tasks where long-term dependencies are essential for consistent object recognition. DSG-DETR clearly outperforms the state-of-art by $\sim$10% and $\sim$20%-30% in terms of R@10 under constraint and no constraint criteria for SGCls and SGDet, respectively. The long-term dependencies in DSG-DETR brings significant improvement.

We observe that the improvement of DSG-DETR becomes less significant when it comes to larger K for SGDet, this is in fact a tradeoff between the consistency and the diversity. Please note, Recall at lower values of K are more significant than the larger values of K when such models are expected to be used for downstream applications.

5.3. Object-level consistency on object-tracks

To study the best possible effect of modeling object-level consistency, we apply the same framework on ground-truth object tracks instead of the constructed tracklets. Such sequences can be treated as the hypothetical *best case* for exploiting such long-term dependencies. We see in Table 2 that the *best case* scenario of exploiting the long-term dependencies largely improves the performance over the baseline; Specifically, SGCls improves by $\sim$13%-14% and SGDet improves by $\sim$ 45%-46% over the baseline. We also show that our proposed tracking algorithm in DSG-DETR helps in reducing the gap between the baseline and the ground-truth tracklet upper-bound in all cases. For example, the *best case* for SGCls performs only $\sim$4% better than DSG-DETR in terms of R@10. For SGDet, DSG-DETR is

Table 1: Comparison with state-of-the-art scene graph generation methods on Action Genome [14]. Note that ⋆ denotes results reproduced from the official model [31] for the same evaluation setup as others.

Method	With Constraint						No Constraints					
	SGCls			SGDet			SGCls			SGDet		
	R@10	R@20	R@50	R@10	R@20	R@50	R@10	R@20	R@50	R@10	R@20	R@50
VRD [23]	32.4	33.3	33.3	19.2	24.5	26.0	39.2	49.8	52.6	19.1	28.8	40.5
M-FREQ [33]	40.8	41.9	41.9	23.7	31.4	33.3	50.4	60.6	64.2	22.8	34.3	46.4
MSDN [19]	43.9	45.1	45.1	24.1	32.4	34.5	51.2	61.8	65.0	23.1	34.7	46.5
VCTree [29]	44.1	45.3	45.3	24.4	32.6	34.7	52.4	62.0	65.1	23.9	35.3	46.8
RelDN [34]	44.3	45.4	45.4	24.5	32.8	34.9	52.9	62.4	65.1	24.1	35.4	46.8
GBS-Net [21]	45.3	46.5	46.5	24.7	33.1	35.1	53.6	63.3	66.0	24.4	35.7	47.3
TRACE [31]⋆	14.8	14.8	14.8	13.9	14.5	14.6	37.1	46.7	50.5	26.5	35.1	45.3
STTran [5]	46.4	47.5	47.5	25.2	34.1	37.0	54.0	63.7	66.4	24.6	36.2	48.8
APT [20]	47.2	48.9	48.9	26.3	**36.1**	**38.3**	55.1	65.1	68.7	25.7	37.9	**50.1**
DSG-DETR(Ours)	**50.8**	**52.0**	**52.0**	**30.3**	34.8	36.1	**59.2**	**69.1**	**72.4**	**32.1**	**40.9**	48.3

able to to reduce the performance gap from the *best case* by almost half comparing with the baseline (∼45% vs ∼21%) in terms of R@10.

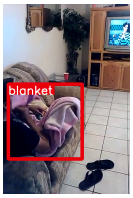
5.4. Ablation studies

In DSG-DETR, we propose to capture long-term dependencies primarily by consistent and effective object tracklets construction. We employ two transformers - one for consistent object prediction and the other for relationship transitions. For the object transformer, we additionally integrate the temporal position with the object representations into the transformer encoder through positional encoding, denoted by "Pos enc" in the ablation Table 3. We use the sinusoidal encoding from [32]. Our relationship transformer architecture shares the same spatial encoder as STTran [5], but it replaces the temporal decoder in STTran with a temporal encoder operating on the predicted classes sequences for capturing the long-term dependencies. In Table 3, we replace our relationship transformer with STTran for ablation. We show the results of ablating our model for SGCls task.

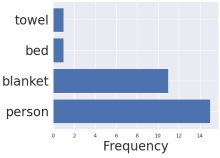
The heavy lifting is done by the object transformer employed on the constructed tracklets. For relationship transformer, the first two rows in the table demonstrate that even with the predicted classes sequences based on Faster R-CNN results, capturing the long-term dependencies still brings the benefits with 0.3 point improvement compared with STTran in R@K under constraint. Finally, the positional encoding in the object transformer boosts the performance by additional 0.2 point in terms of R@K under constraint and even significant improvement for no constraint evaluation. The ablative studies for SGDet also exhibits similar trend, and available in the Supplementary material.

(a) Correctly classified by all the models. (b) DSG-DETR correctly predicts 'dish' objects. (c) STTran (F-RCNN) misclassifies 'dish' as 'food'.

Figure 3: DSG-DETER predicts temporally consistent objects



(a) Correct object prediction.



(b) Histogram of Faster R-CNN predictions on the *blanket* object in the tracklet.

Figure 4: DSG-DETR recovers from the F-RCNN misclassification leveraging context from the tracklet.

Table 4 shows how long-term temporal context helps improve the video object detection and dynamic SGG performances. Long-term temporal context significantly improves SGG by 3-6 points over the baseline (row 3 vs row 1). For short-term temporal context, the performance drops by 3-4 points (row 2 vs row 3), yet beating the baseline. It also shows how object classification accuracy improves as we move from no-temporal context to short-term context to long-term context. Object classification accuracy significantly contributes to the fraction of correct triplet predictions as measured by Recall.

Table 2: Hypothetical *best case* when exploiting long-term dependencies captured via long-term object tracks. GTTrack includes all components of DSG-DETR but uses the ground-truth object tracks instead. DSG-DETR relies on online tracklet construction. DSG-DETR outperforms the baseline and significantly minimizes the performance gap between baseline and GTTrack.

Method	With Constraint				No Constraints			
	SGCls		SGDet		SGCls		SGDet	
	R@10	R@20	R@10	R@20	R@10	R@20	R@10	R@20
Baseline(STTran [1])	45.7	46.8	25.2	34.1	54.2	63.5	24.6	36.2
GTTrack (Upper-bound)	**52.2**	**53.4**	**36.8**	**37.9**	**60.9**	**71.0**	**43.8**	**51.0**
DSG-DETR(Ours)	50.8	52.0	30.3	34.8	59.2	69.1	32.1	40.9

Table 3: Ablation on different components of DSG-DETR for SGCls on Action Genome.

Obj-trans	Pos-enc	Rela-trans	With Constraint		No Constraint	
			R@20	R@50	R@20	R@50
-	-	-	46.8	46.8	63.5	66.0
-	-	✓	47.1	47.1	63.5	65.9
✓	✓	-	51.4	51.4	68.7	71.9
✓	✓	✓	**52.0**	**52.0**	**69.1**	**72.4**

Table 4: Effect of temporal context on SGCls. Short-term context corresponds to tracklet construction over 5 key-frames only, whereas the long-term context uses an order of magnitude higher number of key-frames (usually 5 to 40 key-frames), estimated by our tracklet construction algorithm.

Time context	With Constraint			No Constraint			Obj Acc
	R@10	R@20	R@50	R@10	R@20	R@50	
no-context	45.7	46.8	46.8	54.2	63.5	66.0	70.2
short-term	47.4	48.6	48.6	55.6	65.3	68.1	73.0
long-term	**50.8**	**52.0**	**52.0**	**59.2**	**69.1**	**72.4**	**73.8**

5.5. Qualitative results

Fig 3 is an example where DSG-DETR successfully constructs the sequence of the blue bowl (Figs 3(a) and 3(b)) from the temporally ordered key frames (top to bottom) and makes the correct prediction "dish" for all of them. While Faster R-CNN and STTran will mis-classify as shown in Fig 3(c). Fig 4 shows an example frame where DSG-DETR constructed a tracklet of a blanket shown in the red bounding box. Most of the detections in the tracklet are predicted as "person" by Faster R-CNN shown in Fig 4 (b). However, the object transformer makes a correct prediction "blanket" for all of them in the sequence. This reveals that the object transformer in fact learns to reason over temporal dependencies beyond a simple majority voting.

In Fig 5, we sample three key frames for an action where a person gets up from the bed and walks towards the doorway. The first frame is 33 frames away from the third one in the original video. Thanks to its capability to exploit long-term dependencies, DSG-DETR successfully understands the whole action with only one mistake which is mis-classifying "touching" as "sitting on" in the first frame. However, STTran makes many mistakes including predicting that the human is still "sitting on" the bed in the second frame and the bed is "on the side of" the human in the third frame rather than "behind".

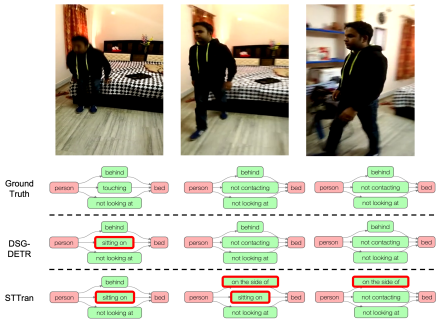


Figure 5: Scene graphs generated by DSG-DETR and STTran for three key frames sampled from an action which embeds long-term dependencies.

6. Conclusions

We hypothesized that capturing long-term temporal context is crucial for dynamic scene graph generation. We presented a framework called Dynamic Scene Graph Detection Transformer (DSG-DETR) that is capable of exploiting long-term dependencies within object-tracklets constructed in an online fashion. We also estimated an upper-bound on the performance of dynamic SGG tasks leveraging such notion of long-term consistencies by utilizing the ground-truth object tracks, and show that DSG-DETR is able to noticeably minimize the performance gap between this upper-bound and the baseline. We demonstrated the efficacy of DSG-DETR on the Action Genome dataset, where it significantly outperforms the state-of-the-art methods.

References

[1] Spatial-temporal transformer for dynamic scene graph generation. https://github.com/yrcong/STTran. Accessed: 2021-08-30.

[2] Anurag Arnab, Chen Sun, and Cordelia Schmid. Unified Graph Structured Models for Video Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[5] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-Temporal Transformer for Dynamic Scene Graph Generation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, October 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[7] Kaifeng Gao, Long Chen, Yifeng Huang, and Jun Xiao. Video relation detection via tracklet based visual transformer. *Proceedings of the 29th ACM International Conference on Multimedia*, Oct 2021.

[8] Noa Garcia and Yuta Nakashima. Knowledge-based video question answering with unsupervised scene descriptions. In *ECCV (18)*, volume 12363 of *Lecture Notes in Computer Science*, pages 581–598. Springer, 2020.

[9] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, pages 244–253. Computer Vision Foundation / IEEE, 2019.

[10] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5299–5309, June 2021.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[13] Jingwei Ji, Rishi Desai, and Juan Carlos Niebles. Detecting human-object relationships in videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, October 2021.

[14] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action Genome: Actions as Composition of Spatio-temporal Scene Graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015.

[16] Gayoung Jung, Jonghun Lee, and Incheol Kim. Tracklet pair proposal and context reasoning for video scene graph generation. *Sensors*, 21(9), 2021.

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, May 2017.

[18] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[19] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1270–1279, 2017.

[20] Yiming Li, Xiaoshan Yang, and Changsheng Xu. Dynamic scene graph generation via anticipatory pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13874–13883, June 2022.

[21] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3746–3753, 2020.

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[23] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016.

[24] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064. PMLR, 10–15 Jul 2018.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[26] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding, 2016.

[28] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video

and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[29] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[30] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target Adaptive Context Aggregation for Video Scene Graph Generation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, October 2021.

[31] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target adaptive context aggregation for video scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13688–13697, 2021.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[33] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. *CoRR*, abs/1711.06640, 2017.

[34] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019.