Dynamic Scene Graph Generation via Anticipatory Pre-training

Yiming Li[1] Xiaoshan Yang[2,3,4] Changsheng Xu[2,3,4*]

[1]School of Information Engineering, Zhengzhou University (ZZU)

[2]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA)

[3]School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)

[4]PengCheng Laboratory (PCL)

liym.zzu@gmail.com, {xiaoshan.yang, csxu}@nlpr.ia.ac.cn

Abstract

Humans can not only see the collection of objects in visual scenes, but also identify the relationship between objects. The visual relationship in the scene can be abstracted into the semantic representation of a triple $\langle subject, predicate, object \rangle$ and thus results in a scene graph, which can convey a lot of information for visual understanding. Due to the motion of objects, the visual relationship between two objects in videos may vary, which makes the task of dynamically generating scene graphs from videos more complicated and challenging than the conventional image-based static scene graph generation. Inspired by the ability of humans to infer the visual relationship, we propose a novel anticipatory pre-training paradigm based on Transformer to explicitly model the temporal correlation of visual relationships in different frames to improve dynamic scene graph generation. In pre-training stage, the model predicts the visual relationships of current frame based on the previous frames by extracting intra-frame spatial information with a spatial encoder and inter-frame temporal correlations with a progressive temporal encoder. In the fine-tuning stage, we reuse the spatial encoder and the progressive temporal encoder while the information of the current frame is combined for predicting the visual relationship. Extensive experiments demonstrate that our method achieves state-of-the-art performance on Action Genome dataset.

1. Introduction

Scene graph abstracts the visual relationships as a graph structure, where the objects are represented as nodes and their relationships are represented as edges. It is a promising way to represent semantics of visual content, which can bridge the large gap between vision and natural language. In recent years, scene graph generation has attracted more and more attention and has been successfully applied in multiple tasks, e.g., image retrieval [19], image captioning [14, 21, 48, 49] and visual question answering [11, 12, 53].

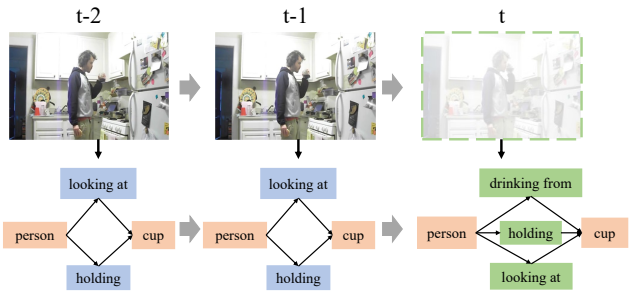


Figure 1. Given previous frames, humans can easily infer the visual relationship contained in the current frame. Because the temporal correlation of different relationships is a kind of common sense to humans. But this kind of temporal reasoning is difficult for computers.

Existing scene graph generation methods can be roughly grouped into two categories, the static scene graph generation, i.e, generating scene graph from a single image, and dynamic scene graph generation, i.e., generating scene graph from a video. For static scene graph generation, existing methods [4, 22, 51, 52] generally use popular object detectors, such as Faster R-CNN [34] and Mask R-CNN [15], to extract objects, and then predict the relationship between objects based on the visual and semantic features. Although static scene graph generation methods have achieved significant progress, the dynamic scene graph generation is less studied, which is more challenging because the objects in video are moving, and thus cause the change of the relationships. The static scene graph generation methods cannot be directly applied to solve dynamic scene graph generation since they ignore the temporal information in videos. To improve the prediction accuracy, existing dynamic scene graph generation methods focus on capturing temporal information by 3D convolution model [39] and transformer [7, 32].

*indicates corresponding author: Changsheng Xu.

Existing dynamic scene graph generation methods explore the temporal structure information in feature-level and model the dynamic scene graph generation as a classification task, which results in that they cannot explicitly capture the temporal correlation of visual relationships. In contrast, humans can easily infer the subsequent relationships based on the past relationships according to their temporal correlations. As shown in Figure 1, after observing $\langle person, looking\_at, cup\rangle$ and $\langle person, holding, cup\rangle$, humans can infer that the subsequent relationships are likely to be consistent with the previous relationships or change to $\langle person, drinking\_from, cup\rangle$. This kind of reasoning ability comes from humans' experience and commonsense in the real world. To make the dynamic scene graph generation model explicitly capture the temporal correlation of visual relationships like humans, there are at least two challenges to be resolved. **(1)** Since the temporal and spatial information in the videos are heavily entangled, it is difficult to explicitly capture the temporal correlations. **(2)** Existing datasets, e.g., Action Genome [18], only have scene graph annotations in key-frame level due to the high cost, which hinders the consecutive modeling of the temporal correlations.

In this paper, we propose an anticipatory pre-training paradigm to predict dynamic scene graph in videos to handle the above challenges. The anticipatory scene graph generation task is defined as using previous frames to predict the relationships in the current frame. Using the anticipatory pre-training paradigm has the two advantages. **(1)** Since the goal of the pre-training task is predicting visual relationships in unseen frames, it can induce the model to explicitly extract the temporal correlations in task-level. **(2)** Based on the pretext task, we can use a large amount of unlabeled data to train the anticipatory model with the supervision of key frame labels, thus can alleviate the problem of insufficient annotations.

The proposed anticipatory pre-training paradigm is instantiated as an anticipatory Transformer architecture. In pre-training stage, the model consists of a spatial encoder to extract intra-frame spatial information and a progressive temporal encoder to capture inter-frame temporal correlations based on both visual and semantic features. To enhance the perception of visual content in long-sequence frames, we design an efficient comprehensive short-term and long-term attention mechanism in the progressive temporal encoder to capture the long-term visual context from labeled and unlabeled frames for each relationship without adding too many parameters. Finally, we predict the visual relationship in current frame based on the output of the progressive temporal encoder. In the fine-tuning stage, we reuse the spatial encoder in the pre-training model to obtain the spatial information of the current frame, and sequentially combine it with the output of the progressive temporal encoder to predict the visual relationship in the current frame.

The main contributions of this work are summarized as follows:

1. We propose a novel anticipatory pre-training paradigm for dynamic scene graph generation, which explicitly models the temporal correlation of visual relationships in the task-level.

2. We instantiate the anticipatory pre-training paradigm with a Transformer architecture. which can not only capture the spatial and temporal information from the labeled training videos based on the visual and semantic features, but also efficiently capture the short-term and long-term visual context from unlabeled data for each relationship.

3. We evaluate the proposed pre-training paradigm on public Action Genome dataset. The extensive experiment results demonstrate that our model achieves state-of-the-art results.

2. Related Work

Scene Graph Generation for Image. Recently, a large number of approaches have been proposed for scene graph generation. A number of methods [6,22,26,38,45,46,52,52] focus on the structural-semantic object features to improve the prediction performance. Xu et al. [46] solve the scene graph generation task by carefully taking the spatial as well as statistical features in a scene. Inspired by this, many methods [6, 22, 38, 45, 52] focus on exploring better spatial context features. Furthermore, Zellers et al. [52] propose a strong baseline which predicts scene graph using only semantic labels of objects. They demonstrate that the semantic information plays an essential role in scene graph generation. To capture correlations among different predicates, Chen *et al.* [5] propose a two-stage predicate association network (PANet). The first stage is designed to extract instance-level and scene-level context information, while the second stage is mainly used to capture the association between predicate alignment features. Our method differs from above image scene graph generation methods in that we consider a more complex task of dynamic scene graph generation in videos. This task requires capturing both the spatial and temporal context in the video.

Scene Graph Generation for Video. Due to the successful application of the scene graph generation method in the field of image scene parsing, researchers turn to explore applying scene graph in video understanding. Video understanding requires reasoning about the relationships between actors and objects within a long video sequence. In SGVST [43], the image scene graph method is used to generate the story from an image stream. Zhuo *et al.* [55] fur-

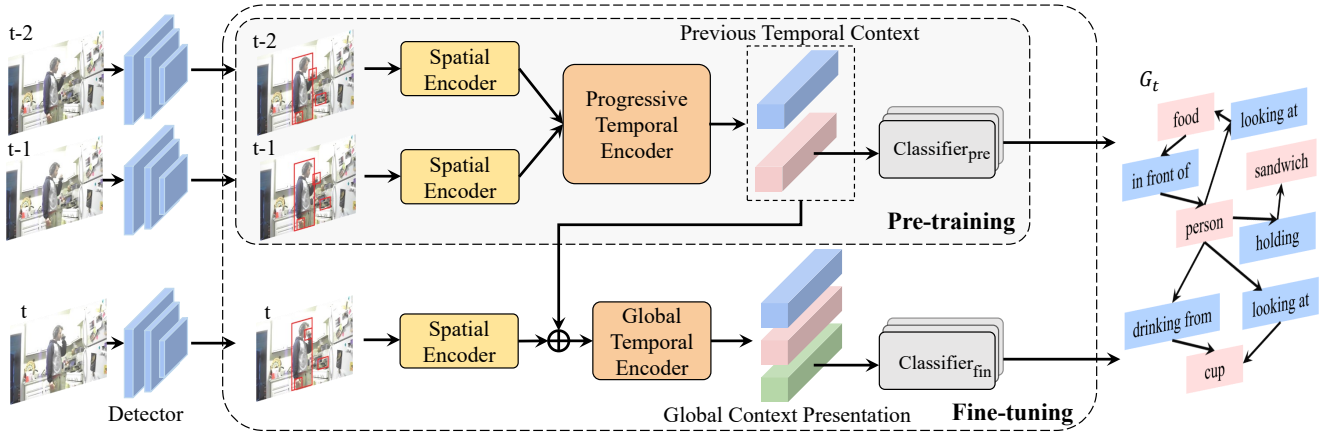


Figure 2. Framework of the proposed method. We employ the spatial encoder for extracting spatial context in a single frame, and a progressive temporal decoder for extracting temporal context. We pre-train the spatial encoder and the progressive temporal encoder for a anticipatory task to learn the temporal correlation. Then we fine-tune the whole model for dynamic scene graph generation by combining the information of the current frame.

ther propose the use of scene graphs to help action reasoning. Although these methods introduce scene graphs into video understanding, they ignore temporal information in the generation of scene graphs. There are very few methods [1, 7, 31, 32, 39] proposed for exploring the utilization of temporal information, and little attention has been paid to explore the temporal correlation of relationships in the prediction and inference. These methods on dynamic scene graph generation simply embed temporal information into visual features, while the temporal correlation between relationships is ignored. The work most related to ours is STTran [7], which adopts a Transformer architecture to explore the temporal dependence of relationships and achieves promising results. The major difference is that we propose an anticipatory pre-training paradigm to explicitly model the temporal correlation of the relationships, which results in the better performance of our model.

Transformer. The Transformer architecture was firstly proposed by Vaswani *et al.* [41] for translation task. Since Transformer has superior performance, a large number of improved models have been developed in the field of natural language processing. Devlin *et al.* [9] propose a large scale pre-trained model BERT, which performs well in a variety of natural language processing tasks. Then, the Transformer has also been successfully applied in vision-language tasks, e.g., VQA [2, 50] and image caption [17, 47]. More recently, Transformer has also be widely used in video-related tasks. For example, Girdhar *et al.* [13] propose Action Transformer that utilizes Transformer to refine the spatio-temporal representations, and Wang *et al.* [44] propose VisTR for video segmentation. Different from these methods, dynamic scene graph generation needs to pay more attention to the temporal changes of the relationships.

Pre-Trained Models. Pre-trained models are first proposed in the field of natural language processing, such as Word2Vec [28], GloVe [29], CoVe [27], ELMo [30], BERT [9], and GPT [3]. These methods use large-scale data for pre-training and achieve satisfactory performance in a variety of downstream tasks, such as object detection [15, 25, 33, 34], and image captioning [2, 42]. Inspired by these methods, more and more pre-trained models have been applied in vision task. A series of CNNs [16, 20, 35, 37] and Transformers [10, 40] are pre-trained on large-scale dataset ImageNet [8] and can provide robust visual features for downstream tasks. More recently, there are also pre-trained models designed for other modalities. VideoBERT [36] conducts pre-training on the Cooking312K video dataset [36] and applies the model in zero-shot action classification task and video captioning task. After pre-training, spoken question answering (SQA) task is used for evaluation. To our best knowledge, this is the first work of applying pre-training and fine-tuning paradigm in dynamic scene graph generation.

3. Method

In this section, we first introduce the problem formulation of dynamic scene graph generation and then describe the structure of the proposed method. Finally, the details of the pre-training and fine-tuning strategies will are given.

3.1. Problem Formulation

Given a video $V = \{I_1, I_2, ..., I_T\}$, dynamic scene graph generation aims to generate a scene graph sequence $G = \{G_1, G_2, ..., G_T\}$, where G_t is the corresponding scene graph of the frame I_t. We define $G_t = \{B_t, O_t, R_t\}$, where $B_t = \{b_{t,1}, b_{t,2}, \cdots, b_{t,N(t)}\}$, $O_t = \{o_{t,1}, o_{t,2}, \cdots, o_{t,N(t)}\}$ and $R_t = \{r_{t,1}, r_{t,2}, \cdots, r_{t,K(t)}\}$ indicate the bounding box set, object set and predicate set, respectively. $N(t)$ is the number of object in the t-th frame,

and $K(t)$ is the number of relationships.

In this work, we formulate the dynamic scene graph generation as an online prediction task based on pre-training paradigm. Since the spatial and temporal information are both important for the prediction of G_t, i.e., both the current frame I_t and the previous frames $\{I_1, I_2, \cdots, I_{t-1}\}$ contribute a lot to the prediction of G_t, the probability of G_t can be formulated as follows:

$$P(G_t|\{I_t\}) = P(G_t|\{I_{t-1}\})P(G_t|I_t), \quad (1)$$

where $P(G_t|\{I_{t-1}\})$ is designed to capture the temporal correlation and is learned by an anticipatory pre-training model. We use $\{I_{t-1}\}$ to denote the set of previous frames for current frame I_t, where $\{I_{t-1}\}$ contains both labeled and unlabeled frames. $P(G_t|I_t)$ is designed to predict the scene graph based on the spatial information from I_t, which is learned in fine-tuning.

Following the widely used definition [52], given I_t, the probability of G_t can be formulated as the multiplication of the probabilities of B_t, O_t, and R_t:

$$P(G_t|I_t) = P(B_t|I_t)P(O_t|B_t, I_t)P(R_t|O_t, B_t, I_t). \quad (2)$$

Similarly, $P(G_t|\{I_{t-1}\})$ can be defined as follows:

$$\begin{aligned} P(G_t|\{I_{t-1}\} =& P(B_t|\{I_{t-1}\})P(O_t|\{B_{t-1}\}, \{I_{t-1}\}) \\ & P(R_t|\{O_{t-1}\}, \{B_{t-1}\}, \{I_{t-1}\}). \end{aligned} \quad (3)$$

3.2. Framework Overview

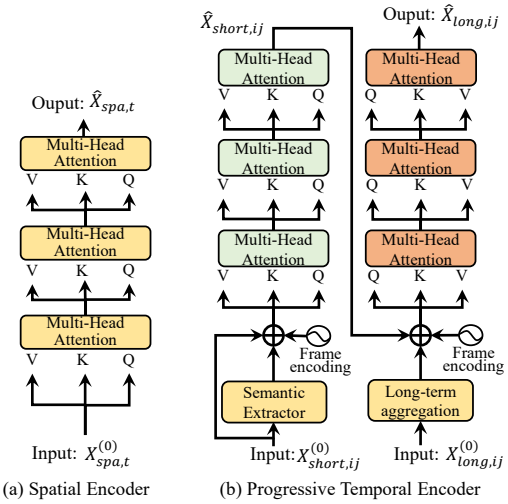
The overall framework of our model is shown in Figure 2. To predict the scene graph G_t for the t-th frame I_t, we firstly use the pre-trained detector to detect object boxes and recognize their categories in the current frame I_t and the previous frames $\{I_{t-1}\}$. Then, we use the spatial encoder to extract the context-aware visual representations of the object pairs in different frames. Next, the progressive temporal encoder is adopted to explore the long-term temporal correlations among object pairs in different frames, which is learned in an anticipatory pre-training network. In the fine-tuning stage, the spatial encoder and the progressive temporal encoder are reused to predict the relation categories for the object pairs in the current frame based on both the output of the spatial encoder and the progressive temporal encoder.

3.3. Detector Backbone

Following [7], we adopt Faster R-CNN as our backbone to detect objects from video frames, which is pre-trained on Action Genome [18] dataset. The representation of object o_i contains spatial information, visual feature and semantic feature, which can be formulated as follows:

$$f_{t,i} = [\mathbf{M}_o v_{t,i}, \phi(b_{t,i}), s_{t,i}], \quad (4)$$

where $[,]$ indicates concatenation operation, $\mathbf{M}_o$ indicates the trainable matrix of a linear transformation layer, and ϕ is a function transforming the bonding box $b_{t,i}$ to a continuous vector. The semantic embedding $s_{t,i}$ is determined by the object category $o_{t,i}$ with a trainable linear embedding layer.



(a) Spatial Encoder (b) Progressive Temporal Encoder

Figure 3. Illustration of the proposed Anticipatory Transformer. (a) is spatial encoder, which is utilized to capture the spatial context information in each frame. (b) is progressive temporal encoder, which captures temporal correlation from the representation of relationships in different frames.

3.4. Anticipatory Transformer

In this work, we design our model based on Transformer to capture spatial information and temporal correlation. Therefore, we firstly give a simple review of the general Transformer [41]. Given the queries Q, keys K and values V, the self-attention layer is defined as follows:

$$Attention(Q, K, V) = Softmax(\frac{QK^\top}{\sqrt{D_K}})V, \quad (5)$$

where $\sqrt{D_K}$ is the key dimensionality.

In classical model, the self-attention operation is followed by a normalization layer, a feed-forward layer and another normalization layer, and all the above constitutes a complete self-attention layer. Extending the self-attention layer into multi heads enable the mechanism to consider various attention distributions and make the model pay attention to different aspects of information, thus the multi-head attention can be generated, which is the main component of the Transformer.

$$\begin{aligned} & MultiHead(X) = Concat(h_1, h_2, \cdots, h_H)\mathbf{W}_o, \\ & h_i = Attention(XW_{Q_i}, XW_{K_i}, XW_{V_i}), \end{aligned} \quad (6)$$

where $X \in \mathbb{R}^{D_X \times D}$, $\mathbf{W}_o \in \mathbb{R}^{HD_V \times D}$ is the parameter matrice, $W_{Q_i} \in \mathbb{R}^{D \times D_{Q_i}}$, $W_{K_i} \in \mathbb{R}^{D \times D_{K_i}}$ and $W_{V_i} \in \mathbb{R}^{D \times D_{V_i}}$ are projection functions. For simplicity, we denote the multi-head attention layer as $MultiHead(\cdot)$, and focus on the description of the input X.

Spatial Encoder. We firstly design a spatial encoder to extract visual information contained in a single frame. As shown in Figure 3(a), Q, K and V share the same input

$X^0_{spa,t} \in \mathbb{R}^{N(t)\times N_{spa}}$, which is presented as:

$$X^{(0)}_{spa,t} = \{f_{t,1}, f_{t,2}, \cdots, f_{t,N(t)}\}, \tag{7}$$

where $N(t)$ denotes the number of objects detected in the frame I_t. The output of the n-th $MultiHead$ layer is computed as follows:

$$X^{(n)}_{spa,t} = MultiHead_{spa}(X^{(n-1)}_{spa,t}). \tag{8}$$

The output of the n-th layer will be used as the input of the (n+1)-th layer. Since $f_{t,i}$ already contains the position information of the corresponding object, there is no additional position coding operation. The final output of the spatial encoder is denoted as $\hat{X}_{spa,t} = \{\hat{f}_{t,1}, \hat{f}_{t,2}, \cdots, \hat{f}_{t,N(t)}\}$.

Relationship Representation. The representation of each relationship is computed based on $\hat{X}_{spa,t}$ and union box feature. We denote the relationship between the objects $o_{t,i}$ and $o_{t,j}$ as $r_{t,ij}$, which can be represented with the following feature:

$$e_{t,ij} = [\hat{f}_{t,i}, \hat{f}_{t,j}, M_u u_{t,ij}], \tag{9}$$

where $u_{t,ij}$ is union box feature of the i-th and j-th objects in frame I_t obtained by RoIAlign, M_u is linear metric for dimension compress.

Progressive Temporal Encoder. As shown in Figure 3(b), the progressive temporal encoder is designed to capture the temporal correlation of relationships, which consists of short-term encoder and long-term encoder. Since short-term temporal information is more relevant to the target frame in pre-training task, while the long-term temporal information contains rich temporal correlation knowledge, both of them are equally important. Simply using long sequence data as input can take into account both the short-term and long-term information at the same time. However, this will result in too many model parameters that are hard to train. To solve this problem, we propose an efficient way to explicitly explore the comprehensive temporal information by a short-term encoder and a long-term encoder. The **short-term encoder** captures short-term information most relevant to the target frame, which takes the relationship representations of the same subject-object pair in different frames as input. To find the same subject-object pair in different frames, we adopt the IoU (i.e., intersection over union) to match the subject-object pairs detected in frames $\{I_{t-\gamma}, \cdots, I_{t-1}\}$, where $\gamma - 1$ is the number of frames that can be processed by the short-term encoder. Specifically, we calculate the matching score between two object pairs $(o_{t',i}, o_{t',j})$ and $(o_{t',i'}, o_{t'-1,j'})$ in in the t'-th and $t'-1$-th frames as follows:

$$\epsilon = min\big(IoU(o_{t',i}, o_{t'-1,i'}), IoU(o_{t',j}, o_{t'-1,j'})\big). \tag{10}$$

The object pairs in adjacent frames are matched if the matching score $\epsilon > 0.8$. For each subject-object pair $(o_{t-1,i}, o_{t-1,j})$ detected in I_{t-1}, we establish the temporal sequence $Pr_{ij} = \{(o_{t-\gamma,i_\gamma}, o_{t-\gamma,j_\gamma}), \cdots, (o_{t-1,i}, o_{t-1,j})\}$ for this pair through frame-by-frame matching. For a frame which does not have the matched object pair, we create a placeholder object pair by simply coping the matched object pair in the nearest frame. The relationship representation sequence $A_{ij} = \{a_{t-\gamma,ij}, \cdots, a_{t-1,ij}\}$ is constructed based on Pr_{ij}. $a_{t',ij}$ is the relationship representation of $(o_{t',i}, o_{t',j})$ and $a_{t',ij} = e_{t',ij}$.

Since the temporal sequence of the relationships has an obvious impact on temporal correlation, we adopt frame position encoding to inject the temporal position in the relationship representation. Specifically, we adopt a trainable liner layer to learn the embeddings of the temporal order $Z^s = \{z^s_{t-\gamma}, \cdots, z^s_{t-1}\}$. The frame position encodings Z^s have the same dimension as the feature representation $e_{t',ij}$ of the relationship. Furthermore, we adopt a semantic extractor which is implemented as a fully connected layer for obtaining the semantic representation $c_{t',ij}$ of the relationship between $o_{t',i}$ and $o_{t',j}$.

The short-term encoder is also comprised of multiple $MultiHead$ layers. The input of the first $MultiHead$ layer is denoted as $X^{(0)}_{s,ij} = [A_{ij} + Z^s, C_{ij}]$, and the n-th $MultiHead$ layer can be formulated as follows:

$$X^{(n)}_{s,ij} = MultiHead_{Short}(X^{(n-1)}_{s,ij}). \tag{11}$$

The final output of short-term encoder is denoted as $\hat{X}_{s,ij}$.

The **long-term encoder** is adopted to capture long-term temporal correlations, which takes the output $\hat{X}_{s,ij}$ of the short-term encoder and the representations of the long relationship sequence as input. The frame encoding Z^l is also used in long-term encoder to indicate temporal order, which is calculated in the same way of Z^s. The input of the first $MultiHead$ layer is denoted as follows:

$$X^{(0)}_{l,ij} = \{f_\theta(U_{ij}), \phi(\hat{X}_{s,ij})\} + Z^l, \tag{12}$$

where ϕ is a 3-layer fully connected network with ReLU activation function, and $U_{ij} = \{u_{t-\lambda}, \cdots, u_{t-\gamma}\}$ denotes the relationship representations of the long-term sequence. The construction of U_{ij} is similar to A_{ij}, the difference is that U_{ij} has a longer sequence than A_{ij}. We denote the length of U_{ij} as λ, which is much larger than γ. f_θ is an aggregation function used to balance the performance and efficiency, which combines the representations of different relationships in the long-term sequence. In this work, we implement f_θ with linear layer according to the analysis of the experiment, and f_θ can be formulated as follows:

$$f_\theta(U_{ij}) = W_\theta(\varphi(u_{t-\lambda,ij}) \otimes \cdots \otimes \varphi(u_{t-1,ij})), \tag{13}$$

where W_θ is a fully connected layer, φ is a convolutional layer for dimension reshape and $\otimes$ denotes cross product operation.

After obtaining the input of the first $MultiHead$ layer in the long-term encoder, we can formulate the long-term encoder as follows:

$$X^{(n)}_{l,ij} = MultiHead_{Long}(X^{(n-1)}_{l,ij}). \tag{14}$$

In each batch, the progressive temporal encoder processes different subject-object pairs in parallel, and the final output of long-term encoder is denoted as $\hat{X}_{l,ij}$, which is the temporal context presentation of relationships.

3.5. Pre-training and Fine-tuning Strategy

In pre-training, as described in Sec. 3.1, we propose a pretext task, which is defined as an online anticipatory prediction. We take $\{I_{t-1}\}$ as model input to predict the scene graph of I_t. Since a large number of frames are unlabeled in dataset, we only use the labeled frames to calculate the loss of pre-training. Furthermore, since Action Genome [18] provides the type of relationship category, e.g., *attention*, *spatial* and *contacting* relationships, we adopt multiple linear classifiers instead of only one classifier to infer different kinds of relationships. The category distribution of relationship $r_{t,ij}$ between $o_{t,i}$ and $o_{t,j}$ can be predicted as follows:

$$y_{t,ij} = Classifiers_{\text{pre}}(\hat{x}_{l,ij}), \tag{15}$$

where $\hat{x}_{l,ij}$ is the last element in $\hat{X}_{l,ij}$. Since in reality there may be multiple correct relationships between two objects, e.g., $\langle person, touching, food\rangle$ and $\langle person, eating, food\rangle$, we adopt a multi-label margin loss in pre-training, which can be formulated as follows:

$$L_{pre}(y_{t,ij}, Y^+, Y^-) = \sum_{p\in Y^+}\sum_{q\in Y^-} max(0, 1-y^p_{t,ij}+y^q_{t,ij}), \tag{16}$$

where Y^+ denotes the set of ground-truth predicate labels, Y^- is the set of the negative predicate labels that are not in the annotation, and $y^p_{t,ij}$ indicates the predicted confidence score of the p-th predicate.

In fine-tuning, we reuse the spatial encoder in Sec. 3.4 to capture the spatial information of the current frame I_t. The representations of objects $\{f_{t,i}\}$ and relationships $\{e_{t,ij}\}$ are constructed following Eq. (4) and Eq. (9) respectively. Then, we adopt another global temporal encoder to capture the temporal correlation based on the output of the long-term encoder, which shares the parameters with the short-term encoder. The formulation of this encoder is defined as follows:

$$\begin{aligned} X^{(0)}_{g,ij} &= \{\hat{X}_{l,ij}, e_{t,ij}\} + Z^f, \\ X^{(n)}_{g,ij} &= MultiHead_{global}(X^{(n-1)}_{g,ij}), \end{aligned} \tag{17}$$

where Z^f is frame encoding, and the output of global temporal encoder is denoted as $\hat{X}_{f,ij}$. Similar as in pre-training, the multiple linear classifiers are also adopted in fine-tuning:

$$y^*_{t,ij} = Classifiers_{\text{fin}}(\hat{x}_{g,ij}), \tag{18}$$

where $\hat{x}_{g,ij}$ is the last element of $\hat{X}_{g,ij}$, and we use the same loss function as pre-training.

In inference, we only use the output of classifiers $Classifiers_{\text{fin}}$ in fine-tuning, while the pre-trained classifiers $Classifiers_{\text{pre}}$ are discarded.

4. Experiments

In this section, we firstly introduce the details of the experimental setting and dataset. Then we compare our model with state-of-the-art methods and report the results. Subsequently, we present ablation and qualitative studies.

4.1. Implementation Details

The proposed method is implemented by PyTorch. We employ Faster RCNN [34] with a ResNet-101 backbone as the object detector following previous work [7]. For the ϕ in Eq. 4, we implement it by a multi-layer perceptron with 3 fully connected layers, and the output dimension is set to 128. The object semantic embedding is obtained by mapping the object category distribution to a 200-dimensional vector with a linear matrix $M_w \in \mathbb{R}^{36\times 200}$. The dimension of object presentation is 840, while the relationship presentation is 2192. The spatial encoder contains 1 multi-head attention layer while the short-term encoder, long-term encoder and global temporal encoder contain 3 multi-head attention layers. The head number of all multi-head attention layers is 8.

During the pre-training stage, we use SGD optimizer with an initial learning rate of 0.001 and decay the learning rate by multiplying it with 0.9 after every epoch. The momentum is set to 0.9 and the size of mini-batch is set to 16. For hyper-parameters, we set the length of short-term sequence γ to 4 according to the validation results, while the long-term encoder takes $\lambda = 10$. The scene in consecutive frames may be unchanged and thus the temporal correlation cannot be reflected, we sample 1 frame in every 3 frames for pre-training. Furthermore, for batches with insufficient previous frames, we fill the sequence with a copy of the first frame.

For fine-tuning, we use SGD optimizer with an initial learning rate of $1e-5$ and decay the learning rate by multiplying it with 0.9 after every epoch. The momentum is set to 0.9 and the size of mini-batch is set to 16.

4.2. Dataset and Metrics

We train and test our method on Action Genome [18], which is the largest dynamic scene graph dataset. Since the goal of Action Genome is to decompose the actions, it focuses on annotating video clips where the action truly takes place and only the objects involved in the action are annotated. In the experiments, we use the same training and testing split as in [18]. Furthermore, we also utilize the unlabeled frames in the action genome for pre-training. We evaluate the performance of our model with the metric of Recall@K (R@K, $K = [10, 20, 50]$), which measures the ratio of correct instances among the top K predicted instances with the highest confidences.

Methods	With Constraint									No Constraint								
	Pred Cls			*SG Cls*			*SG Gen*			*Pred Cls*			*SG Cls*			*SG Gen*		
	R@10	R@20	R@50	R@10	R@20	R@50	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 [26]	51.7	54.7	54.7	32.4	33.3	33.3	19.2	24.5	26.0	59.6	78.5	99.2	39.2	49.8	52.6	19.1	28.8	40.5
MotifFreq [52]	62.4	65.1	65.1	40.8	41.9	41.9	23.7	31.4	33.3	73.4	92.4	**99.6**	50.4	60.6	64.2	22.8	34.3	46.4
MSDN [23]	65.5	68.5	68.5	43.9	45.1	45.1	24.1	32.4	34.5	74.9	92.7	99.0	51.2	61.8	65.0	23.1	34.7	46.5
VCTREE [38]	66.0	69.3	69.3	44.1	45.3	45.3	24.4	32.6	34.7	75.5	92.9	99.3	52.4	62.0	65.1	23.9	35.3	46.8
ReIDN [54]	66.3	69.5	69.5	44.3	45.4	45.4	24.5	32.8	34.9	75.7	93.0	99.0	52.9	62.4	65.1	24.1	35.4	46.8
GPS-Net [24]	66.8	69.9	69.9	45.3	46.5	46.5	24.7	33.1	35.1	76.2	93.6	99.5	53.6	63.3	66.0	24.4	35.7	47.3
STTran [7]	68.6	71.8	71.8	46.4	47.5	47.5	25.2	34.1	37.0	77.9	94.2	99.1	54.0	63.7	66.4	24.6	36.2	48.8
Ours	**69.4**	**73.8**	**73.8**	**47.2**	**48.9**	**48.9**	**26.3**	**36.1**	**38.3**	**78.5**	**95.1**	99.2	**55.1**	**65.1**	**68.7**	**25.7**	**37.9**	**50.1**

Table 1. Comparison with state-of-the-art scene graph generation methods on Action Genome. Best result is marked in **bold**.

Methods	*Pred Cls*		*SG Cls*		*SG Gen*	
	R@20	R@50	R@20	R@50	R@20	R@50
w/o Semantic	72.65	72.97	47.25	47.30	35.62	37.94
w/o long-term	72.24	72.98	47.81	47.15	35.67	37.71
w/o Pre-training	71.57	71.59	44.96	45.82	33.24	35.92
Full model	**73.81**	**73.84**	**48.94**	**48.94**	**36.11**	**38.28**

Table 2. Impact of the relationship semantic information, long-term encoder and pre-training paradigm in the proposed method. Evaluated on **With Constraint** strategy.

Methods	*Pred Cls*		*SG Cls*		*SG Gen*	
	R@10	R@20	R@10	R@20	R@10	R@20
With Constrain						
STTran [7]	68.6	71.8	46.4	47.5	25.2	34.1
STTran*	68.8	72.0	46.6	47.8	25.4	37.4
Ours	**69.4**	**73.8**	**47.2**	**48.9**	**26.3**	**38.3**
No Constrain						
STTran [7]	77.9	94.4	54.0	63.7	24.6	36.2
STTran*	77.9	94.4	54.3	64.5	24.7	36.9
Ours	**78.5**	**95.1**	**55.1**	**65.1**	**25.7**	**37.9**

Table 3. Ablation study of using unlabeled data for training.

We evaluate our model under three kinds of experiment setups: **Predicate Classification** (*Pred Cls*): predict the predicates between actors and objects with given ground-truth bounding boxes and category labels. **Scene Graph Classification** (*SG Cls*): predict both the predicates and the class labels of objects with given ground-truth bounding boxes. **Scene Graph Generation** (*SG Gen*): predict relationship labels of object pairs which are detected by detector. An object box is considered to be correctly detected only if the predicted box has at least 0.5 IoU (Intersection over Union) overlap with the ground-truth box. Since we cannot obtain the ground truth bounding box and object class of unlabeled frames used for training, the detector is utilized in *Pred Cls* and *SG Cls* to detect objects in unlabeled frames. Furthermore, we analyse the performance of dynamic scene graph generation based on two typical generation strategies. (1) **With Constraint:** Each subject-object pair is only allowed to have at most one predicate. (2) **No Constraint:** Each subject-object pair is allowed to have multiple predicates. Moreover, since Action Genome dataset annotates 3 types of relationships (attention, spatial and contact), our model outputs all the three relationships for each subject-object pair following [7].

4.3. Comparison with State-of-the-arts

As shown in Table 1, our model outperforms all static scene graph generation methods and state-of-the-art dynamic scene graph generation methods in all metrics. For fair comparison, all methods share the same object detector.

Since the rich temporal correlation information is obtained from pre-training, our model improves the previous state-of-the-art method [7] by 0.8%/2.0% on *Pred Cls*-R@10/20, 0.8%/1.8% on *SG Cls*-R@10/20 and 0.9%/2.0% on *SG Gen*-R@10/20 under the **With Constraint** strategy. This demonstrates that our model performs better in predicting the most important relationships. For **No Constraint**, our model outperforms other methods in all settings except *Pred Cls*-R@50. Since **No Constraint** allows a subject-object pair to have multiple relationships, and R@50 metric gives the model plenty of opportunities to guess, the results in this case are unstable. However, our model outperforms other methods in R@10 and R@20, where the results become more reliable with the less opportunities of guess.

4.4. Ablation Study

In this part, we conduct more experiments to analyze the impacts of the designed relationship semantic information, long-term encoder and pre-training paradigm.

Impact of the semantic and long-term encoder. The first two lines with the full model in Table 2 reflect the role of the semantic and long-term information in the dynamic scene graph generation. The semantic provides a high-level information of temporal correlation and the long-term encoder expands the ability of the model to perceive long temporal sequence.

Impact of the pre-training paradigm. To analyze the influence of pre-training, we retrain a model, which directly uses the pre-training framework to predict the scene graph

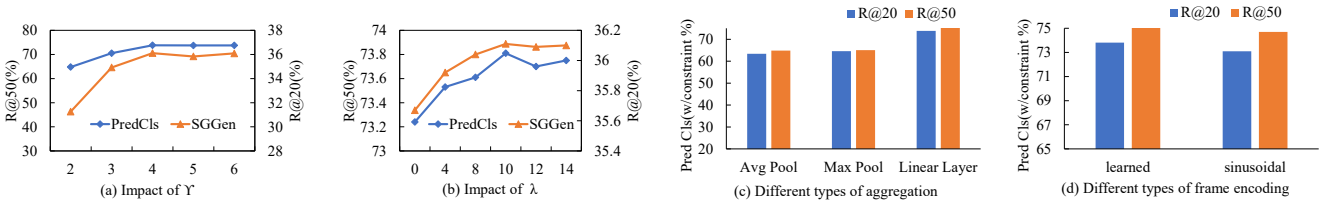


Figure 4. Parameter analysis. (a) and (b) show how the length of the long-term or short-term sequence affect the performance of our model. We analyze different types of temporal aggregation function and frame encoding in (c) and (d). Evaluated on **With Constraint**.

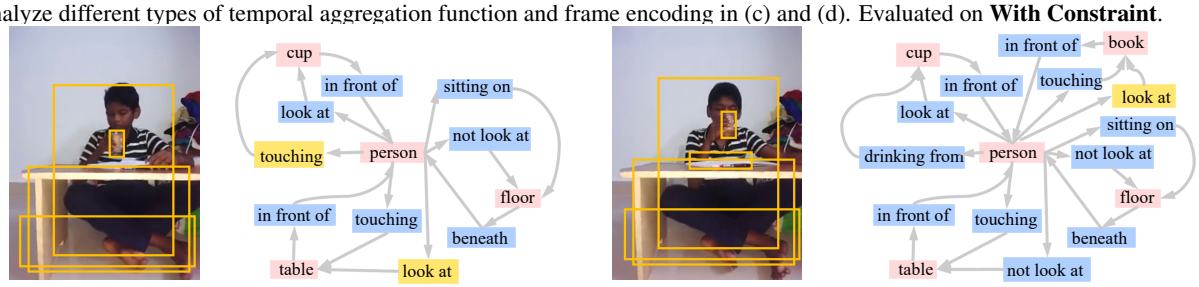


Figure 5. Qualitative results of our model. For the input RGB frames, we generate the scene graph (in *SG Gen* task) with top-10 confidence prediction under the strategy of **With Constraint**. The blue and pink boxes are correct relationships and objects respectively. The yellow boxes are wrong relationships.

of I_t with $\{I_t\}$ instead of $\{I_{t-1}\}$ as input. As shown in Table 2, the performance of our model improves significantly after adding the pre-training strategy, which proves that the pre-training indeed captures the temporal correlation.

Impact of using unlabeled data for training. Since we utilize the unlabeled data for pre-training, we compare our model with the variant model $STTran^*$ which is implemented based on the previous state-of-the-art method [7] and trained with same scale of data. The results shown in Table. 3 demonstrate that the additional data is helpful for capturing time correlation.

Impact of the hyperparameters. There are two important hyperparameters in our model, γ and λ which denote the length of short-term and long-term sequence respectively. As shown in Figure 4 (a) and (b), when λ and γ become larger, the performance of the model gradually improves until it stabilizes at $\lambda = 10$ and $\gamma = 4$.

Impact of the different functions of long-term aggregation and frame encoding. We analyze the performance with different types of long-term aggregation (i.e., f_θ) in Figure 4 (c). The learnable linear layer has better performances on the metrics of R@20 and R@50 than average pooling and maximum pooling, which demonstrates its effectiveness. As shown Figure 4 (d), the learned frame encoding performs better than sinusoidal method on both *Pred Cls*-R@20/50 and *SG Gen*-R@20/50.

4.5. Qualitative Results

The qualitative results are shown in Figure 5. We visualize the results in *SG Gen* metric under the strategy of **With Constraint**, which is a scenario closest to the practical use. The pink box is correct detection result and blue box is correct relationship prediction result. The yellow box is wrong relationship. As shown in Figure 5, our model performs satisfactorily in most of relationships. It is worth noting that when the object (e.g., *book*) is occluded, the detection performance is unstable. In this case, our model still can accurately predict the related relationship after detecting the target based on long-term information and temporal correlations.

5. Conclusion

In this work, we propose a novel pre-training paradigm for dynamic scene graph generation, which induces the model to explicitly extract the temporal correlation in task-level. The pre-training paradigm is instantiated with an Anticipatory Transformer architecture, which introduces the spatial encoder and progressive temporal encoder to extract the intra-frame spatial information and inter-frame temporal correlations. We comprehensively capture the visual context from labeled and unlabeled data for each relationship by the short-term and long-term attention mechanisms in the progressive temporal encoder. We conduct extensive experiments to show that the proposed method significantly outperforms the state-of-the-art methods. In future work, we would like to explore utilizing our method in scene graph-based video generation, which is more challenging.

Acknowledgements. This work was supported by National Key Research and Development Program of China (No. 2018AAA0100604), National Natural Science Foundation of China (No. 61720106006, 61721004, 62072455, U1836220, U1705262, 61872424), Key Research Program of Frontier Sciences of CAS (QYZDJ-SSW-JSC039), and Beijing Natural Science Foundation (L201001).

References

[1] Somak Aditya, Yezhou Yang, Chitta Baral, Yiannis Aloimonos, and Cornelia Fermüller. Image understanding using vision and reasoning through scene description graph. *CVIU*, 173:33–45, 2018. 3

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 3

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 3

[4] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, pages 6163–6171, 2019. 1

[5] Yunian Chen, Yanjie Wang, Yang Zhang, and Yanwen Guo. Panet: A context based predicate association network for scene graph generation. In *ICME*, pages 508–513. IEEE, 2019. 2

[6] Weilin Cong, William Wang, and Wang-Chien Lee. Scene graph generation via conditional random fields. *arXiv preprint arXiv:1811.08075*, 2018. 2

[7] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *ICCV*, pages 16372–16382, 2021. 1, 3, 4, 6, 7, 8

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 3

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[11] Noa Garcia and Yuta Nakashima. Knowledge-based video question answering with unsupervised scene descriptions. *arXiv preprint arXiv:2007.08751*, 2020. 1

[12] Shalini Ghosh, Giedrius Burachas, Arijit Ray, and Avi Ziskind. Generating natural language explanations for visual question answering using scene graphs and visual attention. *arXiv preprint arXiv:1902.05715*, 2019. 1

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, pages 244–253, 2019. 3

[14] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *ICCV*, pages 10323–10332, 2019. 1

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[17] Sen He, Wentong Liao, Hamed R Tavakoli, Michael Yang, Bodo Rosenhahn, and Nicolas Pugeault. Image captioning through image transformer. In *ACCV*, 2020. 3

[18] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *CVPR*, pages 10236–10247, 2020. 2, 4, 6

[19] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015. 1

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 3

[21] X. Li and S. Jiang. Know more say less: Image captioning based on scene graphs. *IEEE TMM*, 21(8):2117–2130, 2019. 1

[22] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'ou Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, pages 1347–1356, 2017. 1, 2

[23] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, pages 1261–1270, 2017. 7

[24] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, pages 3746–3753, 2020. 7

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 3

[26] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pages 852–869. Springer, 2016. 2, 7

[27] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *arXiv preprint arXiv:1708.00107*, 2017. 3

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013. 3

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

[30] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. 3

[31] Xufeng Qian, Yueting Zhuang, Yimeng Li, Shaoning Xiao, Shiliang Pu, and Jun Xiao. Video relation detection with spatio-temporal graph. In *ACM MM*, pages 84–93, 2019. 3

[32] Gorjan Radevski, Marie-Francine Moens, and Tinne Tuytelaars. Revisiting spatio-temporal layouts for compositional action recognition. *arXiv preprint arXiv:2111.01936*, 2021. 1, 3

[33] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 3

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1, 3, 6

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[36] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, pages 7464–7473, 2019. 3

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 3

[38] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pages 6619–6628, 2019. 2, 7

[39] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target adaptive context aggregation for video scene graph generation. In *ICCV*, pages 13688–13697, 2021. 1, 3

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*. PMLR, 2021. 3

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 3, 4

[42] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015. 3

[43] Ruize Wang, Zhongyu Wei, Piji Li, Qi Zhang, and Xuanjing Huang. Storytelling from an image stream using scene graphs. In *AAAI*, volume 34, pages 9185–9192, 2020. 2

[44] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, pages 8741–8750, 2021. 3

[45] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. *arXiv preprint arXiv:1811.06410*, 2018. 2

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

[47] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015. 3

[48] Ning Xu, An-An Liu, Jing Liu, Weizhi Nie, and Yuting Su. Scene graph captioner: Image captioning based on structural visual representation. *JVCIR*, 58:477–485, 2019. 1

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

[50] Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. Bert representations for video question answering. In *WACV*, pages 1556–1565, 2020. 3

[51] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *ECCV*, August 2020. 1

[52] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018. 1, 2, 4, 7

[53] Cheng Zhang, Wei-Lun Chao, and Dong Xuan. An empirical study on leveraging scene graphs for visual question answering. *arXiv preprint arXiv:1907.12133*, 2019. 1

[54] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, pages 11535–11543, 2019. 7

[55] Tao Zhuo, Zhiyong Cheng, Peng Zhang, Yongkang Wong, and Mohan Kankanhalli. Explainable video action reasoning via prior knowledge and state transitions. In *ACM MM*, pages 521–529, 2019. 2