Spatial-Temporal Transformer for Dynamic Scene Graph Generation

Yuren Cong[1], Wentong Liao[1], Hanno Ackermann[1], Bodo Rosenhahn[1], Michael Ying Yang[2]
[1]TNT, Leibniz University Hannover, [2]SUG, University of Twente

Abstract

*Dynamic scene graph generation aims at generating a scene graph of the given video. Compared to the task of scene graph generation from images, it is more challenging because of the dynamic relationships between objects and the temporal dependencies between frames allowing for a richer semantic interpretation. In this paper, we propose **Spatial-temporal Transformer (STTran)**, a neural network that consists of two core modules: (1) a spatial encoder that takes an input frame to extract spatial context and reason about the visual relationships within a frame, and (2) a temporal decoder which takes the output of the spatial encoder as input in order to capture the temporal dependencies between frames and infer the dynamic relationships. Furthermore, STTran is flexible to take varying lengths of videos as input without clipping, which is especially important for long videos. Our method is validated on the benchmark dataset Action Genome (AG). The experimental results demonstrate the superior performance of our method in terms of dynamic scene graphs. Moreover, a set of ablative studies is conducted and the effect of each proposed module is justified. Code available at:* https://github.com/yrcong/STTran.

1. Introduction

A scene graph is a structural representation that summaries objects of interest as nodes and their relationships as edges [26, 29]. Recently, scene graphs have been successfully applied in different vision tasks, such as image retrieval [26, 46], object detection, semantic segmentation, human-object interaction [15], image synthesis [24, 3], and high-level vision-language tasks like image captioning [13, 62] or visual question answering (VQA) [25]. It is treated as a promising approach towards holistic scene understanding and a bridge connecting the large gap between vision and natural language domains. Therefore, the task of scene graph generation has caught increasing attention in communities.

While the great progress made in scene graph generation from a single image (static scene graph generation), the
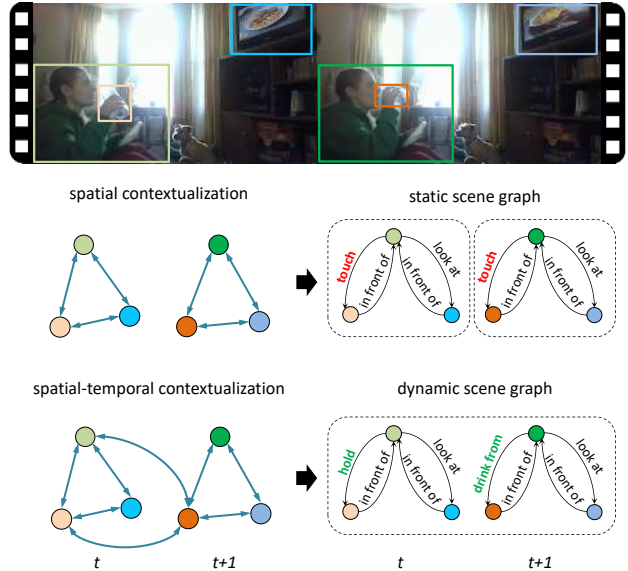

Figure 1: The difference between scene graph generation from image and video. In the video, the person is watching TV and drinking water from the bottle. Dynamic Scene graph generation can utilize both spatial context and temporal dependencies (3rd row) compared with image-based scene graph generation (2nd row). Nodes in different colors denote objects (`person`,`bottle`,`tv`) in the frames.

task of scene graph generation from a video (dynamic scene graph generation) is new and more challenging. The most popular approach of static scene graph generation is built upon an object detector that generates object proposals, and then infers their relationship types as well as their object classes. However, objects are not sure to be consistent in each frame of the video sequence and the relationships between any two objects may vary because of their motions, which is characterized by *dynamic*. In this case, temporal dependencies play a role, and thus, the static scene graph generation methods are not directly applicable to dynamic scene graph generation, which has been fully discussed in [22] and verified by the experimental results analyzed in Sec. 4. Fig. 1 showcases the difference between scene graph generation from image and video.

Action recognition is an alternative to detect the dynamic

relationships between objects. However, actions and activities are typically regarded as monolithic events that occur in videos in action recognition [4, 27, 47, 33]. It has been studied in Cognitive Science and Neuroscience that people perceive an ongoing activity by segmenting them into consistent groups and encoding into a hierarchical part structure [30]. Let's take the activity "drinking water" as an example, as shown in Fig. 1. The person starts this activity by holding the bottle in front of her, and then holds it up and takes water. More complex, the person is looking at the television at the same time. Decomposition of this activity is useful for understanding how it happens and what is going on. Associating with the scene graph, it is possible to predict what will happen: after the person picks up the bottle in front of her, we can predict that the person is likely to drink water from it. Representing temporal events with structured representations, *i.e.* dynamic scene graph, could lead to more accurate and grounded action understanding. However, most of the existing methods for action recognition are not able to decompose the activity in this way.

In this paper, we explore how to generate a dynamic scene graph from sequences effectively. The main **contributions** are summarized as: (1) We propose a novel framework, Spatial-Temporal Transformer (STTran), which encodes the spatial context within single frames and decodes visual relationship representations with temporal dependencies across frames. (2) Distinct from the majority of related works, multi-label classification is applied in relationship prediction and a new strategy to generate a dynamic scene graph with confident predictions is introduced. (3) With several experiments, we verify that temporal dependencies have a positive effect on relationship prediction and our model improves performance by understanding it. STTran achieves state-of-the-art results on Action Genome [22].

2. Related Work

Scene Graph Generation Scene graph has first been proposed in [26] for image retrieval and caught increasing attention in Computer Vision community [42, 61, 35, 9, 38, 51, 55, 62, 64, 37]. It is a graph-based representation describing interactions between objects in the image. Nodes in the scene graph indicate the objects while edges denote the relationships. The applications include image retrieval [46], image captioning [1, 45], VQA [51, 25] and image generation [24, 19]. In order to generate high-quality scene graphs from images, a series of works explore different directions such as utilizing spatial context [61, 65, 40], graph structure [60, 58, 34], optimization [8], reinforcement learning [36, 51], semi-supervised training [7] or a contrastive loss [66]. These works have achieved excellent results on image datasets [29, 42, 31]. Although it is universal for multiple relationships to co-occur between a subject-object pair in the real world, the majority of previous works defaults to

edge prediction as single-label classification. Despite the progress made in this field, all these methods are designed for static images. In order to extend the gain brought by scene graphs in images to video, Ji *et al.* [22] collect a large dataset of dynamic scene graphs by decomposing activities in videos and improve state of the art results for video action recognition with dynamic scene graph.

Transformer for Computer Vision The vanilla Transformer architecture was proposed by Vaswani *et al.* [54] for neural machine translation. Many transformer variants are developed and have achieved great performance in language modeling tasks, especially the large-scale pre-trained language models, like GPT [44] and BERT [10]. Then, Transformers have also been widely and successfully applied in many vision-language tasks, such as image captioning [59, 18], VQA [2, 63]. To further bridge the vision and language domains, different Bert-like large-scale pre-trained models are also developed, like Caption-Based Image Retrieval and Visual Commonsense Reasoning (VCR) [43, 32, 50]. Most recently, Transformers are attracting increasing attention in the vision community. DETR is introduced by Carion *et al.* [5] for object detection and panoptic segmentation. Moreover, Transformers are explored to learn vision features from the given image instead of the traditional CNN backbones and achieve promising performance [12, 52]. The core mechanism of Transformer is its self-attention building block which is able to make predictions by selectively attending to the input points (each point can be a word representation of a sentence or a local feature from an image), so that context is captured between different input points and the representation of each point is refined. Nonetheless, the above methods focus on learning spatial context with a transformer from a single image while temporal dependencies play a role in video understanding. Action Transformer is proposed by Girdhar *et al.* [14] that utilizes transformer to refine the spatio-temporal representations, which are learned by I3D model [6] and then pooled from the RoI given by a RPN network [45], for recognizing human actions in video clips. In fact, the transformer module is still used to learn spatial context. VisTR is introduced in [57] for video segmentation. The features of each frame that are extracted by a CNN backbone are fed to a transformer encoder to learn the temporal information of a video sequence.

Spatial-Temporal Networks Spatial-temporal information is the key to access video understanding [39, 28, 21] and has been long and well studied. To date, the most popular approaches are RNN/LSTM-based [20] or 3D ConvNets-based [23, 53] structures. The former takes features from each frame sequentially and learns the temporal information [49, 11]. The latter extends the traditional 2D convolution (height and width dimension) to time dimension for sequential inputs. Simonyan *et al.* [48] introduce a

two-stream CNN structure that spatial and temporal information is learned on different streams respectively. Residual connections are inserted between the two information streams to allow information fusion. Then, the 2D convolution in the two-stream structure is inflated into its counterpart 3D convolution, dubbed I3D model [6]. Non-local Neural Networks [56] introduce another kind of generic self-attention mechanism, non-local operation. It computes relatedness between different locations in the input signal and refines the inputs by weighted sum of different inputs based on the relatedness. Their method is easy to be applied in video input by extending the non-local operation along the time dimension. However, these works are applied for activity recognition and are not able to decompose the activity into consistent groups. In this work, we do not only utilize transformer to learn spatial context between objects within a frame, but also the temporal dependencies between frames to infer the dynamic relationships varying along the time axis.

3. Method

A dynamic scene graph $G_{dyn}(\mathcal{V}_t, \mathcal{E}_t)$ can be modeled as a static scene graph $G_{stat}(\mathcal{V}, \mathcal{E})$ with an extra index t representing the relations over time as an extra temporal axis. Inspired by the transformer characteristics: (1) the architecture is permutation-invariant, and (2) the sequence is compatible with positional encoding, we introduce a novel model, Spatial-Temporal Transformer (**STTran**), in order to utilize the spatial-temporal context along videos (see Fig. 2).

3.1. Transformer

First, we take a brief review on the transformer structure. The transformer is proposed by Vaswani *et al.* [54] and consists of a stack of multi-head dot-product attention based transformer refining layers. In each layer, the input $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ that has N entries of D dimensions, is transformed into queries ($\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q$, $\boldsymbol{W}_Q \in \mathbb{R}^{D \times D_q}$), keys ($\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K$, $\boldsymbol{W}_K \in \mathbb{R}^{D \times D_k}$) and values ($\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$, $\boldsymbol{W}_V \in \mathbb{R}^{D \times D_v}$) though linear transformations. Note that D_q, D_k and D_v are the same in the implementation normally. Each entry is refined with other entries through dot-product attention defined by:

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D_k}}\right)\boldsymbol{V}, \quad (1)$$

To improve the performance of the attention layer, multi-head attention is applied which is defined as :

$$\begin{aligned} MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) &= Concat(h_1, \ldots, h_h)\boldsymbol{W}_O, \\ h_i &= Attention(\boldsymbol{X}\boldsymbol{W}_{Q_i}, \boldsymbol{X}\boldsymbol{W}_{K_i}, \boldsymbol{X}\boldsymbol{W}_{V_i}). \end{aligned} \quad (2)$$

A complete self-attention layer contains the above self-attention module followed by a normalization layer with

residual connection and a feed-forward layer, which is also followed by a normalization layer with residual connection. For simplicity, we denote such a self-attention layer as $Att(.)$. In this work, we design a Spatio-Temporal Transformer based on $Att(.)$ to explore the spatial context, which works on a single frame, and temporal dependencies that work on sequence, respectively.

3.2. Relationship Representation

We employ Faster R-CNN [45] as our backbone. For the frame I_t at time step t in a given video with T frames $V = [I_1, I_2, \ldots, I_T]$, the detector provides visual features $\{\boldsymbol{v}_t^1, \ldots, \boldsymbol{v}_t^{N(t)}\} \in \mathbb{R}^{2048}$, bounding boxes $\{\boldsymbol{b}_t^1, \ldots, \boldsymbol{b}_t^{N(t)}\}$ and object category distribution $\{\boldsymbol{d}_t^1, \ldots, \boldsymbol{d}^{N(t)}\}$ of object proposals where $N(t)$ indicates the number of object proposals in the frame. Between the $N(t)$ object proposals there is a set of relationships $R_t = \{r_t^1, r_t^2, \ldots, r_t^{K(t)}\}$. The representation vector $\boldsymbol{x}_t^k$ of the relation r_t^k between the i-th and j-th object proposals contains visual appearances, spatial information and semantic embeddings, which can be formulated as:

$$\boldsymbol{x}_t^k = \left\langle \boldsymbol{W}_s\boldsymbol{v}_t^i, \boldsymbol{W}_o\boldsymbol{v}_t^j, \boldsymbol{W}_u\varphi(\boldsymbol{u}_t^{ij} \oplus f_{box}(\boldsymbol{b}_t^i, \boldsymbol{b}_t^j))), \boldsymbol{s}_t^i, \boldsymbol{s}_t^j \right\rangle \quad (3)$$

where $\langle, \rangle$ is concatenation operation, φ is flattening operation and $\oplus$ is element-wise addition. $\boldsymbol{W}_s, \boldsymbol{W}_o \in \mathbb{R}^{2048 \times 512}$ and $\boldsymbol{W}_u \in \mathbb{R}^{12544 \times 512}$ represent the linear matrices for dimension compression. $\boldsymbol{u}_t^{ij} \in \mathbb{R}^{256 \times 7 \times 7}$ indicates the feature map of the union box computed by RoIAlign [16] while f_{box} is the function transforming the bounding boxes of subject and object to an entire feature with the same shape as $\boldsymbol{u}_t^{ij}$. The semantic embedding vectors $\boldsymbol{s}_t^i, \boldsymbol{s}_t^j \in \mathbb{R}^{200}$ are determined by the object categories of subject and object. The relationship representations exchange spatial and temporal information in Spatial-Temporal Transformer.

3.3. Spatio-Temporal Transformer

The Spatio-Temporal Transformer maintains the original encoder-decoder architecture [54]. The difference is, the encoder and decoder are delegated the more concrete tasks.

Spatial Encoder concentrates on the spatial context within a frame whose input is a single $\boldsymbol{X}_t = \{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \ldots, \boldsymbol{x}_t^{K(t)}\}$. The queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ share the same input and the output of the n-th encoder layer is presented as:

$$\boldsymbol{X}_t^{(n)} = Att_{enc.}(\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{X}_t^{(n-1)}) \quad (4)$$

The encoder consists of N identical $Att_{enc.}$ layers that are stacked sequentially. The input of the (n)-th layer is the output of the $(n-1)$-th layer. For simplicity, we remove the superscript n in the following discussion. Unlike the majority of transformer methods, no additional position encoding
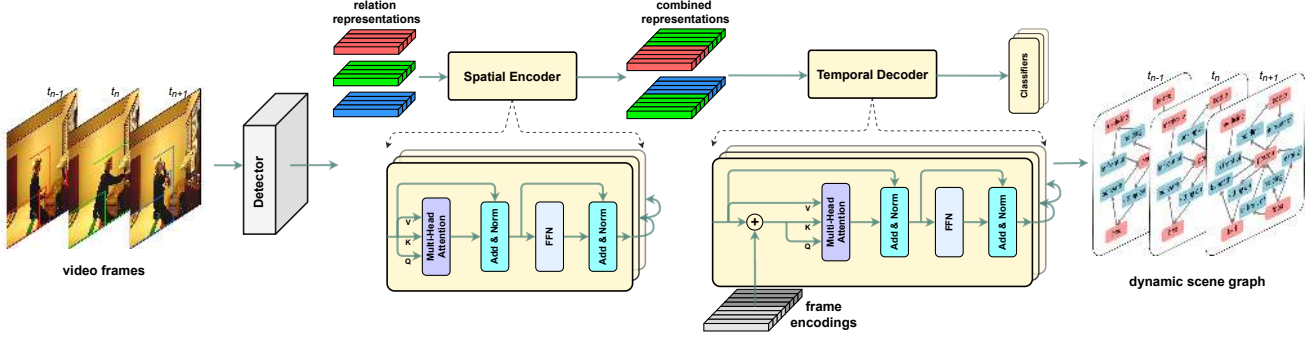
Figure 2: Overview of our method: the object detection backbone proposes object regions in RGB video frames and the relationship feature vectors are pre-processed (Sec. 3.2). The encoder of the proposed Spatial-Temporal Transformer (Sec. 3.3) first extracts the spatial context within single frames. The relation representations refined by encoder stacks from different frames are combined and added to learned frame encodings. The decoder layers capture temporal dependencies and relationships are predicted with linear classifiers for different relation type (such as *attention*, *spatial*, *contact*). $\oplus$ indicates element-wise addition while FFN stands for feed-forward network.

is integrated into the inputs since the relationships within a frame are intuitively parallel. Having said that, the spatial information hiding in the relation representations (see Eq. 10) plays a crucial role in the self-attention mechanism. The final output of the encoder stacks is sent to the Temporal Decoder.

Frame Encoding is introduced for the temporal decoder. Without convolution and recurrence, the knowledge of sequence order such as positional encoding must be embedded in the input for the transformer. In contrast to the word position in [54] or the pixel position in [5], we customize the frame encodings to inject the temporal position in the relationship representations. The frame encodings $\boldsymbol{E}_f$ are constructed with learned embedding parameters, since the amount of the embedding vectors depending on the window size η in the Temporal Decoder is fixed and relative short: $\boldsymbol{E}_f = [\boldsymbol{e}_1, \dots, \boldsymbol{e}_\eta]$, where $\boldsymbol{e}_1, \dots, \boldsymbol{e}_\eta \in \mathbb{R}^{1936}$ are the learned vectors with the same length as $\boldsymbol{x}_t^k$.

The widely used sinusoidal encoding method is also analyzed (see Table 5). We adopt the learned encoding method because of its overall better performance. The window size η is fixed and therefore the video length does not affect the length of frame encodings.

Temporal Decoder captures the temporal dependencies between frames. Not only the amount of calculation required and the memory consumption increase greatly, but also useful information is easily overwhelmed by a large number of irrelevant representations. In this work, we adopt a sliding window to batch the frames so that the message is passed between the adjacent frames in order to avoid interference with distant frames.

Different from [54], the self-attention layer of our temporal decoder is identical to the spatial encoder $Att_{enc.}()$, *i.e.* the masked multi-head self-attention layers are removed. A sliding window of size η runs over the sequence

of spatial contextualized representations $[\boldsymbol{X}_1, \dots, \boldsymbol{X}_T]$ and the i-th generated input batch is presented as:

$$\boldsymbol{Z}_i = [\boldsymbol{X}_i, \dots, \boldsymbol{X}_{i+\eta-1}], i \in \{1, \dots, T - \eta + 1\} \quad (5)$$

where the window size $\eta \leq T$ and T is the video length. The decoder consists of N stacked identical self-attention layer $Att_{dec}()$ similar as the encoder structure. Considering the first layer:

$$
\begin{aligned}
\boldsymbol{Q} = \boldsymbol{K} &= \boldsymbol{Z}_i + \boldsymbol{E}_f, \\
\boldsymbol{V} &= \boldsymbol{Z}_i, \\
\hat{\boldsymbol{Z}}_i &= Att_{dec.}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}).
\end{aligned}
\quad (6)
$$

Regarding the first line in Eq. 6, same encoding is added to the relation representations in the same frame as queries and keys. The output from the last decoder layer is adopted for final prediction. Because of the sliding widow, the relationships in a frame have various representation in different batches. In this work, we choose the earliest representation appearing in the windows.

3.4. Loss Function

We employ multiple linear transformations to infer different kinds of relationships (such as attention, spatial, contacting) with the refined representations. In reality, the same type of relationship between two objects is not unique in semantics, such as synonymous actions <person-holding-broom> and <person-touching-broom>. Thereby, we introduce the multi-label margin loss function for predicate classification as follows:

$$L_p(r, \mathcal{P}^+, \mathcal{P}^-) = \sum_{p \in \mathcal{P}^+} \sum_{q \in \mathcal{P}^-} max(0, 1 - \phi(r, p) + \phi(r, q))$$

$$(7)$$

For a subject-object pair r, $\mathcal{P}^+$ are the annotated predicates while $\mathcal{P}^-$ is the set of the predicates not in the annotation. $\phi(r, p)$ indicates the computed confidence score of the p-th predicate.

During training, the object distribution is computed by two fully-connected layers with a ReLU activation and a batch normalization in between. The standard cross entropy loss L_o is utilized. The total objective is formulated as:

$$L_{total} = L_p + L_o \qquad (8)$$

3.5. Graph Generation Strategies

There are two typical strategies to generate a scene graph with the inferred relation distribution in previous works: (a) **With Constraint** only allows each subject-object pair to have at most one predicate while (b) **No Constraint** allows a subject-object pair to have multiple edges in the output graph with multiple guesses. **With Constraint** is more rigorous and indicates the ability of models to predict the most important relationships, but it is incompetent for the multi-label task. Although **No Constraint** can reflect the ability of multi-label prediction, tolerant multiple guesses cause wrong information in the generated scene graph.

In order to make the generated scene graph closer to ground truth, we propose a new strategy named **Semi Constraint** allowing that a subject-object pair has multiple predicates such as <person-holding-food> and <person-eating-food>. The predicate is regarded as positive iff the corresponding relation confidence is higher than the threshold.

At test time, the score of each relationship triplet <subject-predicate-object> is computed as:

$$s_{rel} = s_{sub} \cdot s_p \cdot s_{obj}, \qquad (9)$$

where s_{sub}, s_p, s_{obj} are the confidence score of subject, predicate and object respectively.

4. Experiments

4.1. Dataset and Evaluation Metrics

Dataset We train and validate our model on the Action Genome (AG) dataset [22] which provides frame-level scene graph labels and is built upon the Charades dataset [47]. $476, 229$ bounding boxes of 35 object classes (without $person$) and $1, 715, 568$ instances of 25 relationship classes are annotated for $234, 253$ frames. These 25 relationships are subdivided into three different types: (1) $attention$ relationships denoting whether a person is looking at an object, (2) $spatial$ relationships and (3) $contact$ relationships which indicate the different ways the object is contacted. In AG, $135, 484$ subject-object pairs are labeled with multiple spatial relationships (*e.g.* <door-in front of-person>

and <door-on the side of-person>) or contact relationships (*e.g.* <person-eating-food> and <person-holding-food>).

Evaluation Metrics We follow three standard tasks from image-based scene graph generation [42] for evaluation : (1) predicate classification (PREDCLS): given ground truth labels and bounding boxes of objects, predict predicate labels of object pairs. (2) scene graph classification (SG-CLS): classify the ground truth bounding boxes and predict relationship labels. (3) Scene graph detection (SGDET): detect the objects and predict relationship labels of object pairs. The object detection is regarded as successful if the predicted box overlaps with the ground-truth box at least 0.5 IoU. All tasks are evaluated with the widely used $Recall@K$ metrics ($K = [10, 20, 50]$) following **With Constraint**, **Semi Constraint** and **No Constraint**. The threshold of confidence in the relationship is set to 0.9 in **Semi Constraint** for all experiments if no special instruction.

4.2. Technical Details

In this work, FasterRCNN [45] based on ResNet101 [17] is adopted as object detection backbone. We first train the detector on the training set of Action Genome [22] and get 24.6 mAP at 0.5 IoU with COCO metrics. The detector is applied to all baselines for fair comparison. The parameters of the object detector including RPN are fixed when training scene graph generation models. Per-class non-maximal suppression at 0.4 IoU is applied to reduce region proposals provided by RPN.

We use an AdamW [41] optimizer with initial learning rate $1e^{-5}$ and batch size 1 to train our model. Moreover, gradient clipping is applied with a maximal norm of 5. For all experiments on Action Genome, we set the window size $\eta = 2$ and $stride = 1$ for our STTran. The spatial encoder contains 1 layer while the temporal decoder contains 3 iterative layers. The self-attention module in both encoder and decoder has 8 heads with $d_{model} = 1936$ and $dropout = 0.1$. The 1936-d input is projected to 2048-d by the feed-forward network, then projected to 1936-d again after ReLU activation.

4.3. Quantitative Results and Comparison

Table 1 shows that our model outperforms state-of-the-art image-based methods in all metrics following **With Constraint**, **Semi Constraint** and **No Constraint**. For the fair comparison, all methods share the identical object detector which provides feature maps and region proposals of the same quality.

The bold numbers denote the best result in any column. With the help of temporal dependencies our model improves state-of-the-art (GPS-Net [40]) 1.9% on PredCLS-

Method	With Constraint									No Constraint								
	PredCLS			SGCLS			SGDET			PredCLS			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	R@10	R@20	R@50	R@10	R@20	R@50
VRD[42]	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
Motif Freq[65]	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[35]	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[51]	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
RelDN[66]	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[40]	66.8	69.9	69.9	45.3	46.5	46.5	24.7	33.1	35.1	76.0	93.6	99.5	53.6	63.3	66.0	24.4	35.7	47.3
STTran	**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**

Table 1: Comparison with state-of-the-art image-based scene graph generation methods on Action Genome [22]. The same object detector is used in all baselines for fair comparison. STTran has the best performance in all metrics. Note that the evaluation results of baselines are different from [22] since we adopted a more reasonable relationship output method, more details are provided in the supplementary material.

Method	Semi Constraint								
	PredCLS			SGCLS			SGDET		
	R@10	R@20	R@50	R@10	R@20	R@50	R@10	R@20	R@50
VRD[42]	55.5	64.9	65.2	36.2	39.7	40.1	19.0	27.1	32.4
Motif Freq[65]	65.7	74.1	74.5	45.5	49.3	49.5	22.9	33.7	39.0
MSDN[35]	69.6	78.9	79.9	48.3	54.1	54.5	23.2	34.2	41.5
VCTREE[51]	70.1	78.2	79.6	49.0	53.7	54.0	23.7	34.8	40.4
RelDN[66]	70.7	78.8	80.3	49.4	53.9	54.1	24.1	35.0	40.7
GPS-Net[40]	71.3	81.2	82.0	50.2	55.0	55.2	24.5	35.3	41.9
STTran	**73.2**	**83.1**	**84.0**	**51.2**	**56.5**	**56.8**	**24.6**	**35.9**	**44.0**

Table 2: Evaluation results of **Semi Constraint** which indicates the relationship between object pair is regarded as positive if the confidence score is higher than the threshold.

$R@20$, 1.0% on SGCLS-$R@20$ and 1.0% on SGDET-$R@20$ for the strategy **With Constraint**, which shows that STTran performs better than image-based baselines in predicting the most important relationships between an object pair. Our model also has excellent performance (see Table 2): 1.9% on PredCLS-$R@20$, 1.5% on SGCLS-$R@20$ and 0.6% improvement on SGDET-$R@20$ for **Semi Constraint** that allows multiple relationships between a subject-object pair. For **No Constraint**, STTran outperforms other methods in all settings except PredCLS-$R@50$. Due to the small number of object pairs and the large number (50) of chances to guess, the results in this column are unstable and unconvincing. Motif Freq [65] which is very dependent on statistics achieves the highest score. However, the results become reliable with the less prediction number $K = [10, 20]$.

Note that there is no difference between PredCLS-$R@20$ and PredCLS-$R@50$ for **With Constraint** because of a limited number of object pairs and edge restriction. This also happens on SGCLS. Compared with PredCLS or SGCLS, the gap of SGDET between STTran and other methods is narrowed since the increased false object proposals cause interference, especially for **Semi Constraint** and **No Constraint** using small K. Furthermore, the reproduced results of some methods are different from [22] since a more reasonable relationship output method is adopted and the object detectors are different.

In **Semi Constraint**, the threshold of confidence in the

relationship is set to a fixed number (0.9) in the experiments. In order to study the impact of such threshold in **Semi Constraint** on $Recall@K$, the $R@20$-Threshold curves of [35, 66, 40] and STTran are shown in Fig. 3. STTran consistently outperforms all three models at all threshold levels from 0.7 to 0.95. The high threshold suppresses the $R@20$ values except in SGDET since there are more pair proposals.
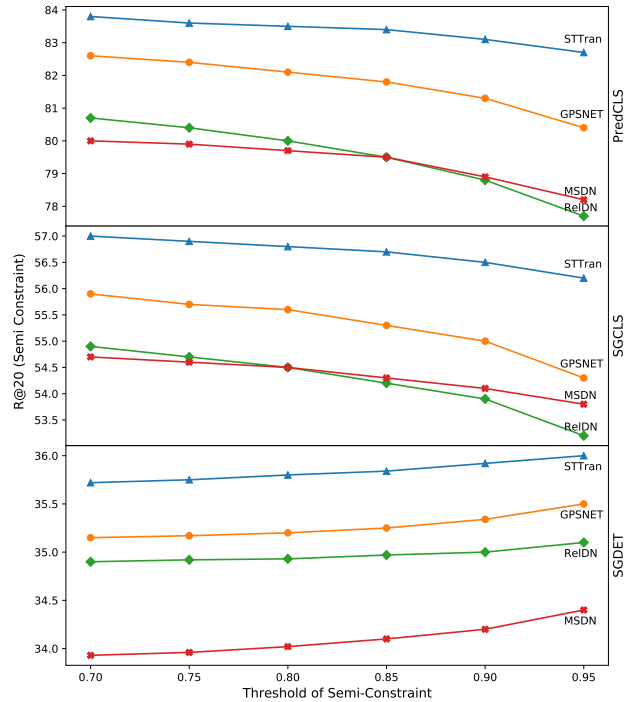

Figure 3: $R@20$-Threshold Curves in three standard tasks (PredCLS/SGCLS/SGDET) for **Semi Constraint**.

4.4. Temporal Dependency Analysis

Compared to the previous image-based scene graph generation, a dynamic scene graph has additional temporal dependencies that can be utilized. We discuss whether tempo-

ral dependencies can improve the relationship inference and validate that our proposed method utilize temporal dependencies. In this subsection, we measure PredCLS-R@20 (**With Constraint**) as the performance indicator that shows the ability of single relationship classification strictly.

Is temporal dependence easy to use? Spatial context plays a relevant role in scene graph generation as validated by several image-based methods [65, 40]. To explore the effectiveness of temporal dependencies, we graft the widely-used recurrent network, LSTM onto the baselines in Table 3 as follows. Before forwarding the feature vectors into the final classifiers, the entire vectors representing relationships in the video are organized as a sequence and processed by LSTM.

Table 3 shows all baselines can gain more or less from the temporal dependencies. For Motif Freq [65], PredCLS-R@20 increases from 65.1% to 65.2% slightly probably due to the relatively simple feature representation. Meanwhile, the score of GPS-Net [40] is improved from 69.9% to 70.4% significantly. The experiment shows that temporal dependencies are helpful for scene graph generation. However, the previous methods were designed for static images. This is why we propose Spatial-Temporal Transformer (STTran) to make better use of temporal dependencies.

Method	PredCLS-R@20	
	original	+LSTM
Motif Freq[65]	65.1	65.2
MSDN[35]	68.5	68.8
RelDN[66]	69.5	69.7
GPS-Net[40]	69.9	70.4

Table 3: We integrate LSTMs to process the relationship features before forwarding them into the classifier into some representative baselines. All baselines are improved with temporal dependencies but worse than our STTran.

(a) spatial encoder only   (b) complete STTran

Figure 4: Two relationship instances respectively generated by the spatial encoder and STTran. (a) Spatial encoder predicts the wrong relationship only with the spatial context in the second frame while (b) STTran can infer more accurate results with the help of temporal dependencies.

Can STTran really understand temporal dependencies? In order to verify that STTran really improves performance through temporal dependencies in the video, instead of using clearer feature representation or powerful multi-head attention module, we trained our model with the processed training set and show the results in Table 4.

We randomly sample $1/3$ videos in the training set and shuffle/reverse them. Meanwhile, the test set remains unchanged. As shown in Table 4, PredCLS-R@20 (**With Constraint**) drops significantly from 71.8% to 71.0%, when one-third of the training videos are reversed, which is equivalent to adding noise in the temporal information. Moreover, shuffled videos indicate the temporal information is completely broken and the noise is further amplified. The experimental result (first row) is in line with expectations: PredCLS-R@20 drops to 70.6%. The experiments demonstrate where the improvement comes from and validate that the temporal dependencies are learned in STTran.

Normal Video	Processed Video	Processing	PredCLS-R@20
2/3	1/3	shuffle	70.6
2/3	1/3	reverse	71.0
1	-	-	**71.8**

Table 4: We shuffle/reverse one-third of the videos in the training set to explore the sensitivity of the model to frame sequence. By disorganizing the temporal information via shuffling or reversing the video sequence, the performance of the model degrades accordingly as expected.

4.5. Ablation Study

In our Spatial-Temporal Transformer, two modules are proposed, a Spatial Encoder and Temporal Decoder. Furthermore, we integrate the temporal position into the relationship representations with the frame encoding in the Temporal Decoder. In order to clarify how these modules contribute to the performance, we ablate different components and present the results in Table 5. We adopt PredCLS-R@20 and SGDET-R@20 as the metrics with **With Constraint** and **Semi Constraint**. PredCLS shows the ability of relationship prediction intuitively while SGDET indicates the performance of scene graph generation.

When only the spatial encoder is enabled, the model works the same as the image-based method and also has a similar performance as RelDN [66]. The isolated temporal decoder (second row) boosts the performance significantly with the additional information from other frames. PredCLS-R@20 is improved slightly when the encoder and decoder both work whereas the improvement of SGDET-R@20 is limited by the object detection backbone. The learned frame encoding helps STTran fully understand the temporal dependencies and has a strong, positive effect both on PredCLS-R@20 and SGDET-R@20
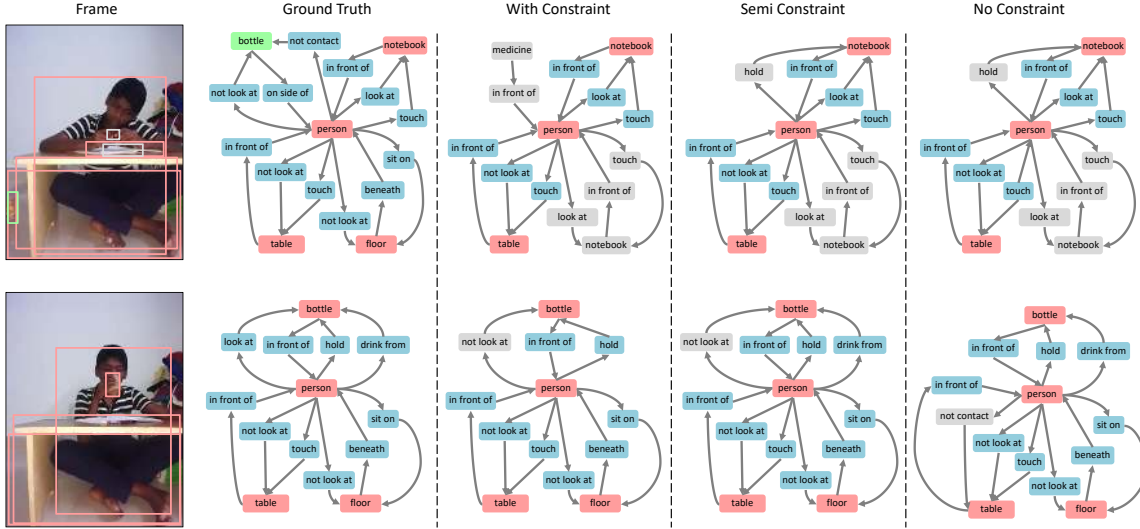
Figure 5: Qualitative results for dynamic scene graph generation. The scene graphs from STTran are generated with the top-10 confident relationship predictions with different Strategies. The green box is the undetected ground truth. The melon and gray colors indicate true positive and false positive respectively. Correct relationships are colored with light blue for clarity and *ing* is omitted for brevity. It shows poor object detection results reduce the performance and the result of **Semi Constraint** is closer to the ground truth.

while the fixed sinusoidal encoding performs unsatisfactorily. Two instances respectively predicted by the spatial encoder only and the complete STTran are shown in Fig. 4. Without temporal dependencies, the spatial encoder mistakenly predicts `<person-eating-food>` as `<person-touching-food>` in the second frame whereas STTran infers the relationship correctly. This explicitly proves that STTran can utilize temporal context to improve scene graph generation.

Spatial Encoder	Temporal Decoder	Frame Encoding	PredCLS-R@20 With	PredCLS-R@20 Semi	SGDET-R@20 With	SGDET-R@20 Semi
✓	-	-	69.6	78.7	32.9	35.1
-	✓	-	71.0	82.2	33.7	35.5
✓	✓	-	71.3	82.7	33.8	35.6
✓	✓	sinusoidal	71.3	82.8	33.9	35.7
✓	✓	learned	**71.8**	**83.1**	**34.1**	**35.9**

Table 5: Ablation Study on our STTran. ✓ indicates the corresponding module is enabled while − indicates disabled. We also compare the effectiveness of sinusoidal and learned positional encoding.

4.6. Qualitative Results

Fig. 5 shows the qualitative results for the dynamic scene graph generation. The five columns from left to right are RGB frame, scene graph generated by ground truth, scene graph generated with the top-10 confident relationship predictions with the Strategies **With Constraint**, **Semi Constraint** and **No Constraint**. The melon color indicates truth positive whereas gray indicates false positive. The green

box is the ground truth not detected by the detector. In the first row, two false positives with high object detection confidence (*medicine* and *notebook*) result in wrong predictions among the top-10 relationships. All the top-10 confident relationships following three strategies are of high quality in the second row when the object detection is successful. `<person-drinking from-bottle>` in the third column is lost because **With Constraint** only allows at most one relationship between each subject-object pair for each type of relationship while `<person-not contacting-bottle>` replaces the attention relationship between *person* and *bottle* in the top-10 confident list when using **No Constraint**. The two frames in Fig. 5 are not adjacent since the detected *person*s overlap with the ground truth IoU < 0.5 in the frames between them.

5. Conclusion

In this paper, we propose Spatial-Temporal Transformer (STTran) for dynamic scene graph generation whose encoder extracts spatial context within a frame and decoder captures the temporal dependencies between frames. Distinct from single-label losses in previous works, we utilize a multi-label margin loss and introduce a new strategy to generate scene graphs. Several experiments demonstrate that temporal context has a positive effect on relationship prediction. We obtain state-of-the-art results for the dynamic scene graph generation task on the Action Genome dataset.

6. Appendix

In this supplementary material, we provide additional implementation details for our method in Sec. 6.1 of this appendix. In Sec. 6.2, we present detailed analysis of the Action Genome dataset [22]. In Sec. 6.3, we show additional qualitative results. Failure cases of our method are shown in Sec. 6.4.

6.1. Implementation Details

In this section, we present some implementation details that were omitted in the main paper for brevity.

Box Function f_{box} It transforms the bounding boxes of the subject and object to the $256 \cdot 7 \cdot 7$ feature map. Following [65], the bounding boxes of the subject and object are firstly converted to a binary spatial mask of size $2 \cdot 27 \cdot 27$ which indicates the location of the subject and object in the frame. By forwarding the spatial mask into a convolutional network (see Fig. 6), the location representation is computed which can be added to the $256 \cdot 7 \cdot 7$ feature map of the union box.
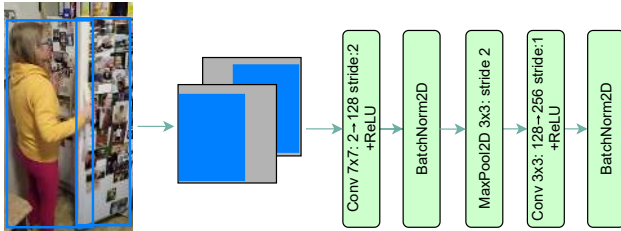
Figure 6: Illustration of the box function f_{box}

Queries and Keys in the Temporal Decoder   For the i-th batch in the decoder layers, the queries Q and keys K are computed by adding the learned frame encoding $E_f = [e_1, \ldots, e_\eta]$ to $Z_i = [X_i, \ldots, X_{i+\eta-1}]$. Note that E_f and Z_i have the same length. $X_i = \{x_i^1, \ldots, x_i^{K(i)}\}$ denotes all the relationship representations in the i-th frame. Here we use braces to emphasize that there is no order between relationships in the same frame and X_i is still a matrix (tensor) in our PyTorch code. Therefore, the first elements of Q and K can be formulated as:

$$q_1 = k_1 = e_1 + X_i = [x_i^1 + e_1, \ldots, x_i^{K(i)} + e_1] \quad (10)$$

which means the same encoding is added to the relation representations in the same frame.

Object Classification   FasterRCNN [45] based on ResNet101 outputs a `2048-d` feature vector and a class distribution for each object proposal box. With multiplying the class distribution by the linear matrix $W_e \in \mathbb{R}^{36 \times 200}$, a `200-d` semantic embedding is computed. Meanwhile, the

`4-d` box coordinate is forwarded into a feed-forward network (see Fig. 7) to achieve a `128-d` position embedding. We concatenate the feature vector, semantic embedding and position embedding, then project the concatenated vector to a `37-d` distribution (including the class $background$) with two linear layers and a ReLU function in between.

Data Pre-processing   When performing down-sampling in the backbone, the visual information of ultra-small objects is damaged. In the experiments for SGCLS/SGDET, we only keep bounding boxes with short edges larger than 16 pixels as [35] did.
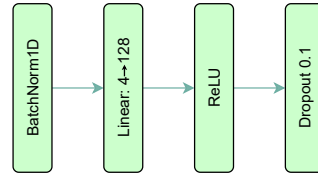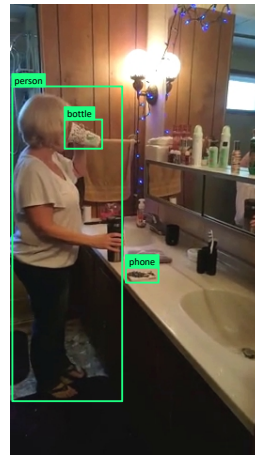

Figure 7: The box coordinate is forwarded into the feed-forward network to compute the position embedding.

Figure 8: An example of the data annotation in Action Genome dataset.

6.2. Benchmark from Action Genome

In the Action Genome (AG) dataset [22], each human-object pair is annotated with three types of relationships, namely *attention*, *spatial*, and *contact* relationships where *attention* and *contact* relationships are formulated in the order of `<person-predicate-object>`, and *spatial* relationships are in the order of `<object-predicate-person>`. Note that the *spatial*, and *contact* relationships can be annotated with multiple labels in Action Genome dataset. An annotation example is shown in Fig. 8.

A benchmark following **With Constraint** is provided by [22]. However, their evaluation code and object de-

tector have not been released. We also evaluate several advanced image-based models. Although the ranking of the model performances is consistent with [22] (VRD [42]<Motif Freq [65]<MSDN [35]<RelDN [66]), the values of $Recall@K$ are different. PredCLS-$R@K$ ($K = [10, 20, 50]$) computed by us are generally much higher, *e.g.*, PredCLS-$R@20$ from us = 69.5 whereas PredCLS-$R@20$ from [22] = 49.4 for RelDN [66]. The reason for the difference was found after discussing with the authors of [22]. Each person-object pair is allowed to have either an *attention* or *contact* relationship in [22]. Instead of, we allow each person-object to have:

- <person-attention relationship-object>

- <object-spatial relationship-person>

- <person-contact relationship-object>

for **With Constraint** so that *attention* and *contact* relationships can be detected simultaneously. Each human-object pair is allowed to have more than one *spatial* or *contact* relationship when the confidence score is higher than the threshold (0.9) following **Semi Constraint**. For **No Constraint**, the most confident top-K relationships are chosen no matter what kind of relationship. SGCLS/SGDET-$R@K$ ($K = [10, 20, 50]$) from [22] are slightly higher than ours. We argue that their object detector has a better performance which is crucial for SG-CLS/SGDET. Note that person boxes in the ground truth are annotated by the detector from [22] in the present version of the Action Genome dataset.

Furthermore, there are two kinds of $Recall@K$ metrics in [22]: image-wise and video-wise. The video-wise $Recall@K$ is not adopted in our work because the only difference is whether the per-frame measurements in each video are first averaged.

6.3. Additional Results

We also report the average precision of predicates AP_{pred} to evaluate the performance for single relationships. The AP_{pred} evaluates the average precision of the predicates where the subject and object boxes are given. The 10 most frequently occurring relationships in Action Genome dataset (2 *attention*, 4 *spatial* and 4 *contact* relationships) are evaluated with our model and GPS-Net [40], which performs best in the image-based scene graph generation methods. The results are shown in the Table 6. Compared with GPS-Net, our model has a great advantage in predicting *attention* relationships with temporal dependencies and also performs better for *spatial* relationships. However, GPS-Net outperforms STTran on the prediction of *holding* and *not contacting* for *contact* relationships.

Different performance of 3 generation strategies are demonstrated in Fig. 9. For **With Constraint**, *wearing* is
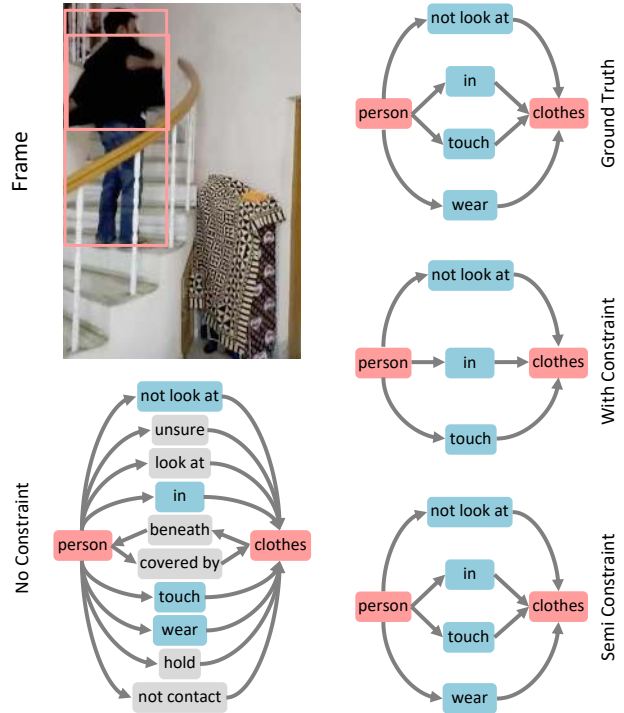

Figure 9: Qualitative instances in the PredCLS setting following different strategies. Light blue relationships are true positives predicted by STTran at the $R@10$ setting while gray are false positives. The graph from **Semi Constraint** is identical to the ground truth, whereas there are several false positives in the graph from **No Constraint** without restriction.

abandoned since only one contact relationship is allowed between each object pair. Although **No Constraint** allows multi-label prediction, the result contains a lot of noise when there are few pairs in the frame, especially bounding boxes are given in PredCLS and SGCLS.

Additional qualitative results for dynamic scene graph generation from the video are shown in Fig. 10. The dynamic scene graphs are generated with the top-10 confident predictions with different Strategies in the SGDET task. The green boxes denote the undetected truths. The melon and gray colors indicate true positive and false positive respectively. Correct relationships are colored with light blue whereas relationships not in the ground truth are colored with gray. In the video the person sitting on the bed holds the medicine and bottle. Then she takes the medicine and drinks water from the bottle.

6.4. Failure Cases

In order to clarify the limitation of the model, we analyze the results and summarize the following most common failure cases:
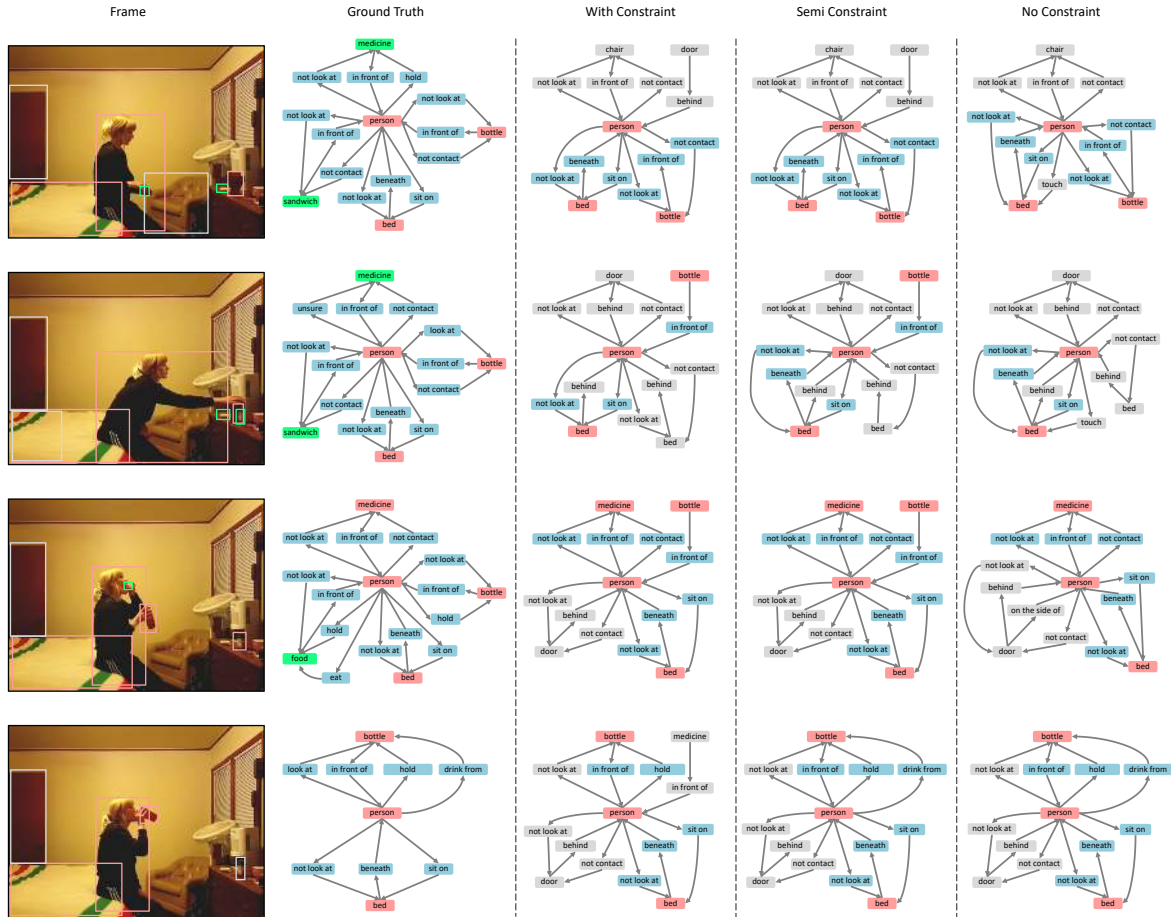
Figure 10: Qualitative results for dynamic scene graph generation. The scene graphs are generated with the top-10 confident predictions with different Strategies in the SGDET task. The green boxes denote the undetected ground truth. The melon and gray colors indicate true positive and false positive respectively. Correct relationships are colored with light blue whereas relationships not in the ground truth are colored with gray. In the video the person sitting on the bed holds the medicine and bottle. Then she takes the medicine and drinks water from the bottle.
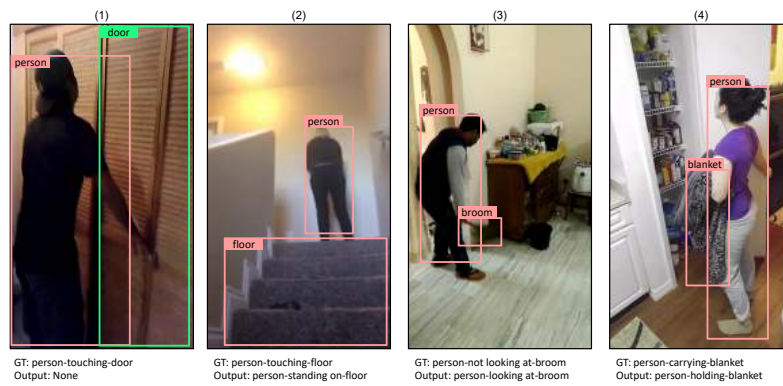

Figure 11: Instances of the most common failure cases. (1) The door is not detected by the object detector and there is no corresponding relationships in the output. (2) STTran predicts that the person is standing on the floor while the ground truth is incorrect. (3) Although the prediction from STTran is wrong, it is difficult for humans to identify whether the person is looking at the broom or not. (4) *carrying* which occurs less frequently in Action Genome is predicted as *holding* with a similar meaning and a higher frequency.

Method	AP_{pred}										
	not looking at	looking at	in front of	on the side of	beneath	hebind	holding	not contacting	touching	sitting on	*mean*
GPS-Net[40]	64.94	49.81	90.14	38.08	**88.98**	77.45	**88.38**	**81.30**	37.26	88.36	*70.47*
STTran	**79.73**	**67.07**	90.14	**40.52**	88.93	**81.01**	85.29	81.29	**37.50**	**90.67**	***74.22***

Table 6: The average precision of predicates AP_{pred} for the top-10 frequent relationships including 2 *attention*, 4 *spatial* and 4 *contact* relationships. We compare our model with GPS-Net [40] which performs best on the Action Genome dataset among the image-based baselines. With temporal dependencies, STTran has a great advantage in predicting *attention* relationships and also performs better for *spatial* relationships. For *contact* relationships, GPS-Net outperforms STTran on the prediction of *holding* and *not contacting*. The last column is the mean of AP_{pred} for these 10 relationships.

1. The object is not detected (IoU< 0.5), particularly small objects such as *phone* and *medicine*.

2. The predictions do not match the ground truth relationships which are annotated by mistake.

3. The relationship is ambiguous and difficult to be identified even by humans.

4. The model predicts the wrong majority relationship instead of the correct minority relationship.

The failure cases are shown in Fig. 11. We conjecture that Failure 1 can be improved by a better object detector. Failure 2 and Failure 3 are caused by the human-labeled annotations. Failure 4 is caused by the imbalanced relationship distribution both in the dataset and in the real world.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer, 2016. 2

[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. 2

[3] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4561–4569, 2019. 1

[4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 4

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2, 3

[7] Vincent S Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2580–2590, 2019. 2

[8] Yuren Cong, Hanno Ackermann, Wentong Liao, Michael Ying Yang, and Bodo Rosenhahn. Nodis: Neural ordinary differential scene understanding. In *ECCV*, pages 636–653, 2020. 2

[9] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *CVPR*, pages 3076–3086, 2017. 2

[10] 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. 2

[11] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 2

[12] 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. In *ICLR*, 2021. 2

[13] Lizhao Gao, Bo Wang, and Wenmin Wang. Image captioning with scene-graph based semantic concepts. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 225–229, 2018. 1

[14] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, pages 244–253, 2019. 2

[15] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, 2018. 1

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[18] Sen He, Wentong Liao, Hamed R Tavakoli, Michael Yang, Bodo Rosenhahn, and Nicolas Pugeault. Image captioning through image transformer. In *ACCV*, 2020. 2

[19] Sen He, Wentong Liao, Michael Ying Yang, Yongxin Yang, Yi-Zhe Song, Bodo Rosenhahn, and Tao Xiang. Context-aware layout to image generation with enhanced object appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15049–15058, 2021. 2

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[21] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning*, pages 4364–4375. PMLR, 2020. 2

[22] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020. 1, 2, 5, 6, 9, 10

[23] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2012. 2

[24] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 1, 2

[25] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2989–2998, 2017. 1, 2

[26] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015. 1, 2

[27] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2

[28] Florian Kluger, Hanno Ackermann, Michael Ying Yang, and Bodo Rosenhahn. Temporally consistent horizon lines. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3161–3167. IEEE, 2020. 2

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

[30] Christopher A Kurby and Jeffrey M Zacks. Segmentation in the perception and memory of events. *Trends in cognitive sciences*, 12(2):72–79, 2008. 2

[31] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, pages 1–26, 2020. 2

[32] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. What does bert with vision look at? In *ACL*, pages 5265–5275, 2020. 2

[33] Shijie Li, Jinhui Yi, Yazan Abu Farha, and Juergen Gall. Pose refinement graph convolutional network for skeleton-based action recognition. *IEEE Robotics and Automation Letters*, 6(2):1028–1035, 2021. 2

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

[35] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1261–1270, 2017. 2, 6, 7, 9, 10

[36] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 848–857, 2017. 2

[37] Wentong Liao, Cuiling Lan, Michael Ying Yang, Wenjun Zeng, and Bodo Rosenhahn. Target-tailored source-transformation for scene graph generation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2021. 2

[38] Wentong Liao, Bodo Rosenhahn, Ling Shuai, and Michael Ying Yang. Natural language guided visual relationship detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2

[39] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 2

[40] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020. 2, 5, 6, 7, 10, 12

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[42] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European conference on computer vision*, pages 852–869. Springer, 2016. 2, 5, 6, 10

[43] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, volume 32, 2019. 2

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2, 3, 5, 9

[46] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 1, 2

[47] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 2, 5

[48] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014. 2

[49] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015. 2

[50] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020. 2

[51] 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, 6

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 2

[53] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 2

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2, 3, 4

[55] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *CVPR*, pages 8188–8197, 2019. 2

[56] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 3

[57] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020. 2

[58] 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

[59] 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 *International conference on machine learning*, pages 2048–2057. PMLR, 2015. 2

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

[61] Michael Ying Yang, Wentong Liao, Hanno Ackermann, and Bodo Rosenhahn. On support relations and semantic scene graphs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 131:15–25, 2017. 2

[62] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. 1, 2

[63] 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. 2

[64] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *ICCV*, pages 1974–1982, 2017. 2

[65] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018. 2, 6, 7, 9, 10

[66] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019. 2, 6, 7, 10