GTR: A Grafting-Then-Reassembling Framework for Dynamic Scene Graph Generation

Jiafeng Liang¹, Yuxin Wang¹, Zekun Wang¹, Ming Liu¹,²*, Ruiji Fu³, Zhongyuan Wang³ and Bing Qin¹,²
¹Harbin Institute of Technology, Harbin, China
²Peng Cheng Laboratory, Shenzhen, China
³Kuaishou Technology, Beijing, China
{jfliang, yuxinwang, zkwang, mliu, qinb}@ir.hit.edu.cn, {furuiji, wangzhongyuan}@kuaishou.com

Abstract

Dynamic scene graph generation aims to identify visual relationships (subject-predicate-object) in frames based on spatio-temporal contextual information in the video. Previous work implicitly models the spatio-temporal interaction simultaneously, which leads to entanglement of spatio-temporal contextual information. To this end, we propose a Grafting-Then-Reassembling framework (GTR), which explicitly extracts intra-frame spatial information and inter-frame temporal information in two separate stages to decouple spatio-temporal contextual information. Specifically, we first graft a static scene graph generation model to generate static visual relationships within frames. Then we propose the temporal dependency model to extract the temporal dependencies across frames, and explicitly reassemble static visual relationships into dynamic scene graphs. Experimental results show that GTR achieves the state-of-the-art performance on Action Genome dataset. Further analyses reveal that the reassembling stage is crucial to the success of our framework.

1 Introduction

Scene graph is a structured representation of an image that clearly represents entities (nodes) and relationships between them (edges) through a series of triples. Such structured representations play an important role in many downstream tasks, such as visual question-answering [Garcia and Nakashima, 2020; Luo et al., 2022], visual reasoning [Shi et al., 2019], image captioning [Zhang et al., 2021] and vision-and-language navigation (VLN) [Hong et al., 2020]. Scene graph can be applied to individual images (as static scene graph) or to each frame of a video (as dynamic scene graph). Most methods for generating static scene graphs begin by detecting objects in the image using an object detector, and then obtain the relationships between the objects. However, these methods cannot be directly applied to dynamic scene graph generation because they neglect the natural temporal dependence of relationships across frames (shown in Figure 1).

Most previous methods [Cong et al., 2021; Li et al., 2022] for dynamic scene graph generation utilize the Transformer [Vaswani et al., 2017] to encode the visual features of entity pairs and decode their relationships. Although these methods achieve remarkable performance, they implicitly model spatio-temporal interaction simultaneously and require a large amount of video data for training, which presents two challenges. First, the one-stage modeling process cannot extract spatial and temporal contextual information separately, leading to entanglement between them (shown in Figure 2 (a)). Second, these methods require large amounts of video data to learn spatio-temporal interactions, resulting in expensive annotation and training costs.

To address the aforementioned challenges, we propose a novel two-stage Grafting-Then-Reassembling framework (GTR) for dynamic scene graph generation. In the first stage, we graft a pre-trained static scene graph generation model to
generate static visual relationships. Since the model already has the basic static information generation ability, we only need a small amount of the video data and take each frame as an image to fine-tune the model for learning consecutive action relationship types. With the help of grafting, the intra-frame static visual relationships can be obtained without the need for expensive training and manual annotation of large amounts of video data.

In the second stage, to resolve the entanglement of spatio-temporal contextual information, we propose the temporal dependency model (TDM), which contains a temporal attention module and a context attention module. Specifically, after obtaining the visual and semantic features of each frame, we extract temporal dependencies across frames based on these frame features using the temporal attention module. In addition, the mask strategy is designed to capture fine-grained temporal dependencies in the temporal attention module, which can effectively distinguish the cross-frame temporal dependencies of different entity pairs. The context attention module aims to explicitly reassemble static visual relationships into the dynamic scene graph based on temporal dependencies. We consider that there is a positive inductive bias in videos, that is consecutive visual relationships often occur in sequence (i.e., \( \langle \text{person} - \text{holding} - \text{cup} \rangle \) and \( \langle \text{person} - \text{drinking from} - \text{cup} \rangle \) have a high probability of occurring in sequence). For the reason, when static visual relationships within frames are incorrect due to a lack of temporal contextual information, the context attention module can rectify the error by replacing it with the correct relation predicate that has similar visual features in other frames based on the temporal dependence (shown in Figure 2 (b)). In addition, to increase the availability of relation predicates, we design a noise filter (NFT) based on visual feature similarity, which can effectively filter out redundant relation predicates.

To evaluate the performance of the proposed framework, we conduct extensive experiments on Action Genome [Ji et al., 2020]. Our experiments show that GTR achieves state-of-the-art results using only 60% of the video data for training, (i.e., 71.2% Recall@10 for predicate classification task), which is 1.8% higher than the previous best result. Besides, extensive analysis experiments demonstrate that GTR has excellent performance in capturing the spatio-temporal interaction.

Our contributions are summarized as follows:

- We propose GTR, a novel two-stage framework to explicitly capture spatio-temporal interactions for accurate dynamic scene graphs generation.
- Our framework does not require a large amount of video data for training, saving expensive manual video data annotation costs.
- Experimental results show that our framework has significant performance improvements compared to the one-stage approach. Further analysis indicates that our proposed reassembling stage is the key to success.

2 Related Work

2.1 Static Scene Graph Generation

Scene graph generation task was first proposed by Johnson et al. [2015], advancing the state of the art in downstream computer vision tasks, natural language processing tasks and multimodal tasks. Currently, the methods of static scene graph generation are mainly based on CNN [Zhang et al., 2017b; Li et al., 2017a; Woo et al., 2018], RNN [Chen et al., 2019; Tang et al., 2019] and TransE [Zhang et al., 2017a; Wan et al., 2018; Gkanatsios et al., 2019; Hung et al., 2019]. CNN-based methods attempt to extract visual features of entity pairs in images by convolution and classify relationships based on these features. Samy Bengio et al. [2018] propose a relational embedding module to improve scene graph generation by explicitly modeling inter-dependency among the entire object instances. RNN-based methods attempt to infer relationships of entity pairs based on visual contextual information. Tang et al. [2019] propose to compose dynamic tree structures that place the objects in an image into a visual context, helping scene graph generation. TransE-based methods attempt to infer relationships between subject and object by computing the distance between them in the semantic vector space. Wan et al. [2018] provide a fully convolutional module to extract the visual embeddings of a visual triple and apply hierarchical projection to combine the structural and visual embeddings of a visual triple. However, as downstream video tasks are widely studied, the static scene graph is not sufficient for their needs. As a result, dynamic scene graph generation has started to be gradually studied by scholars.

2.2 Dynamic Scene Graph Generation

The difference between dynamic scene graph generation and static scene graph generation is that dynamic scene graph generation is for videos, which have an additional time dimension compared to images, making the task more challenging. Currently, there is not much research work [Cong et al., 2021; Li et al., 2022; Gao et al., 2022; Qian et al., 2019; Teng et al., 2021] on this task. Cong et al. [2021] proposed a Spatial-Temporal Transformer, which encodes spatial context within single frames and decodes relationships based on temporal
dependencies. Li et al. [2022] propose a Transformer-based anticipatory pre-training paradigm that uses unlabeled frames for pre-training to improve dynamic scene graph generation. However, these methods implicitly model the spatio-temporal interaction simultaneously and require a large amount of video data for training. Thus, we propose a novel two-stage framework to improve the problem of spatio-temporal contextual information entanglement by explicitly reassembling static visual relationships into dynamic scene graph based on temporal dependencies. We also graft a pre-trained static scene graph generation model into our framework, leading to outstanding performance without the need for extensive video data training.

2.3 Transformer for Time Series Modeling

Currently, most dynamic scene graph generation models are based on the Transformer [Vaswani et al., 2017] and demonstrate a powerful understanding of the dependencies between long sequences of data [Zhou et al., 2022; Tuli et al., 2022; Zerveas et al., 2021]. Recently, Transformer has also started to be widely applied to computer vision tasks. Girdhar et al. [2019] propose an Action Transformer model for recognizing and localizing human actions in video clips. Arnab et al. [2021] propose a pure Transformer-based model to classify videos by encode spatio-temporal tokens from the video. Due to the powerful time-series modeling capabilities of Transformer, our framework models temporal dependencies in the video based on the Transformer architecture.

3 Method

In this section, we first present the definition of dynamic scene graph generation task (Section 3.1). Then, we describe our Grafting-Then-Reassembling framework (GTR) in detail. As shown in Figure 3, our framework consists of two stages: the grafting stage (Section 3.2) and the reassembling stage (Section 3.3). In the first stage, we graft a static scene graph generation model to generate static visual relationships within frames. In the second stage, we extract the temporal dependencies between frames by proposed Temporal Dependency Model (TDM) and reassemble static information into dynamic scene graphs. Meanwhile, we introduce a Noise Filter (NFT) to remove redundant candidate static relation predicates.

3.1 Task Definition

Given a video \( V = \{F_1, F_2, ..., F_t\} \), the goal of the dynamic scene graph generation task is to take the video content as a set of graph nodes \( G^{vid} = \{G_1^{vid}, G_2^{vid}, ..., G_t^{vid}\} \). The \( G_t^{vid} \) is the scene graph based on the \( t \)-th frame \( F_t \), defined as \( G_t^{vid} = \{B_t, E_t, R_t\} \), where \( B_t = \{b_1, b_2, ..., b_l\} \) denotes the bounding box set, \( E_t = \{e_1, e_2, ..., e_f\} \) denotes the entity set and \( R_t = \{r_1, r_2, ..., r_k\} \) denotes the relation predicate set.

3.2 Grafting Stage

In this stage, we adopt a static scene graph generation model as initialization and fine-tune it with video data to generate static visual relationships within frames in the video. We convert each frame in the video to an image as data for fine-tuning, i.e., \( \{F_1, F_2, ..., F_t\} \rightarrow \{I_1, I_2, ..., I_t\} \), which allows the model to learn consecutive action relationship types that are unique to the video. In this way, we extend the image pre-training model from image task to video task to leverage its extensive pre-training knowledge. Specifically, for a given video \( V \), we obtain a set of static scene graphs \( G^{img} = \{G_1^{img}, G_2^{img}, ..., G_t^{img}\} \) by static scene graph generation model and extract the static visual relationships \( V R^{img} = \{\{S_1, R_1, O_1\}, \{S_2, R_2, O_2\}, ..., \{S_t, R_t, O_t\}\} \) from them, where \( S_i, O_i, R_i \) denotes the categories of the entity pairs (subject and object) and static relation predicates between them in \( t\)-th frame. Moreover, to increase the amount of relation predicates available for the next stage, we generate the top-k possible predicates between entity pairs based on the likelihood score during the inference process.

3.3 Reassembling Stage

This stage aims to model the inter-frame dependencies and reassemble the static visual relationships into dynamic scene graphs. To this end, we propose the Temporal Dependency Model (TDM) and the Noise Filter (NFT). The TDM consists of two parts: a temporal attention module and a context attention module.

Feature Extractor For the given video, we use a pre-trained Faster R-CNN to extract frame-level feature following [Cong et al., 2021]. To comprehensively describe the entity pairs, we consider both visual and semantic features. Specifically, as depicted in the bottom middle part of Figure 3, the visual feature of the entity pair \( p \) in \( t \)-th frame contains subject’s feature \( v_i^t \), object’s feature \( v_j^t \) and their union bounding box feature \( v_{ij}^t \). The semantic feature of entity pair \( p \) in \( t \)-th frame contains semantic embeddings of the subject and object categories, i.e., \( c_i^t, c_j^t \in \mathbb{R}^{d_s} \). Then, the feature \( f_p^t \in \mathbb{R}^{d_f} \) for entity pair \( p \) is:

\[
f_p^t = [\text{MLP}_v([v_i^t; v_j^t; v_{ij}^t]); \text{MLP}_s([c_i^t; c_j^t])]
\]

where \( \text{MLP}_v \) and \( \text{MLP}_s \) are two trainable MLPs, \([;]\) denotes the concatenation.

Temporal Dependency Model (TDM) It is a natural feature of video that the relationships between entity pairs in different frames are correlated. The temporal attention module aims to extract potential temporal dependencies across frames in the video. Specifically, the video feature \( F_p \in \mathbb{R}^{N(t) \times d_f} \) is presented as:

\[
F_p = \{f_{p,1}^1, f_{p,2}^1, ..., f_{p,N(t)}^1\}
\]

where \( N(t) \) denotes the number of entity pairs in the \( t \)-th frame. The correlation score between frames can be calculated as:

\[
Q_{frm} = K_{frm} = F_p + E_p, V_{frm} = F_p
\]

\[
S_{frm} = \text{softmax}(\frac{Q_{frm}(K_{frm})^T}{\sqrt{d_k}})
\]

where \( E_p \) is constructed with learned embedding parameters that is used to inject time positions in entity pair features. Intuitively, temporal dependencies only occur between same entity pairs in different frames. For example, in \( t \)-th frame, there is a visual relationship...
Thus, to obtain fine-grained temporal dependence, we mask
that is, consecutive visual relationships often occur in se-

doing the training loss for both object distribution and
relation predicate classification. For frame $F_i$, which contains
$\langle person − drinking from − cup \rangle$, which is only temporal de-
from visual relationship $\langle person − holding − cup \rangle$
in (t-1)-th frame, and temporal independent from visual rela-
tionship $\langle person − watching − television \rangle$ in (t-1)-th frame.

To obtain fine-grained temporal dependence, we mask
the correlation scores between different entity pairs, i.e.,
$S^{\text{frm}}_p \rightarrow \text{Mask}_p [S^{\text{frm}}_p]$. Then, we utilize the masked corre-
lation scores to weight temporal dependencies across frames:

$$H^{\text{frm}}_p = \text{Mask}_p [S^{\text{frm}}_p] V^{\text{frm}}_p$$

(5)

After obtaining the temporal contextual information in
the video, we reassemble the static visual relationships into
dynamic scene graphs by context attention module. We
consider that there is a positive inductive bias in videos,
that is, consecutive visual relationships often occur in se-
quence (e.g., high probability of $\langle person − holding − cup \rangle$
and $\langle person − drinking from − cup \rangle$ sequence occurring
in one video). Therefore, we treat static relation predic-
ates in all frames and entity pairs in current frame as candidates and targets respectively to explicitly model
the correlation between them based on temporal de-
pendencies (e.g., target: $\langle person − ? − cup \rangle$, candidates:
$\langle holding, drinking from, touching, \ldots \rangle$). For the entity
pair $p_n$, its candidate static relation predicates $R_n$ are
presented as:

$$R_n = \{r_{p_n,1}, r_{p_n,2}, \ldots, r_{p_n,c} \}$$

(6)

where $c$ denotes the number of candidate static relation predic-
ates in t-th frame. The matching scores between $p_n$ and $R_n$
is obtained by:

$$Q^{\text{frm}}_p = H^{\text{frm}}_p K^{\text{stc}} = V^{\text{stc}} = W_R R_n$$

(7)

$S^{\text{stc}}_p = \text{softmax}(Q^{\text{frm}}_p (K^{\text{stc}})^T / \sqrt{d_k})$

(8)

where $W_R$ is a trainable weight. We decompose the represen-
tation $S^{\text{stc}}_p$ into three parts based on the three different types of
relation predicates (attention, spatial, contacting) by mask
operation. The weighted representation after matching is:

$$H^{\text{frm}}_{\text{att}} = \text{Mask}_{\text{att}} [S^{\text{stc}}_p] V^{\text{stc}}$$

(9)

$$H^{\text{frm}}_{\text{spa}} = \text{Mask}_{\text{spa}} [S^{\text{stc}}_p] V^{\text{stc}}$$

(10)

$$H^{\text{frm}}_{\text{con}} = \text{Mask}_{\text{con}} [S^{\text{stc}}_p] V^{\text{stc}}$$

(11)

Noise Filter (NFT) Static relation predicates are generated
from a static scene graph generation model and fed to the temporal
dependency model. To improve the availability of static
relation predicates, we design NFT to remove redundant ones.
Specifically, for an entity pair $p$ in frame $F_i$. We calculate
the cosine similarity score between the visual content in the
bounding box of entity pair $p$ in frame $F_i$ and in frame $F_j$:

$$\text{Sim}(F_i, F_j) = \frac{p_{i,\text{bbox}} \cdot p_{j,\text{bbox}}}{\| p_{i,\text{bbox}} \| \cdot \| p_{j,\text{bbox}} \|}$$

(12)

The relation predicates in $F_j$ are regarded positive if the cor-
responding similarity confidence is greater than the threshold
and can be used as candidate relation predicates for the entity
pair $p$.

3.4 Training

We consider the training loss for both object distribution and
relation predicate classification. For frame $F_i$, which contains
$n_o$ entities and $n_p$ entity pairs, there are $n_r$ predicates be-
tween an entity pair, (e.g., $\langle person − touching − food \rangle$) and
$\langle person − holding − food \rangle$ occur simultaneously). Specifi-
cally, we optimize the model parameters $\theta$ by minimizing cost
as follows:

$$\hat{\theta} = \arg\min_{\theta} \sum_{p_n=1}^{n_p} \sum_{m=1}^{n_r} L_{\text{match}} (r^*_{p_n,m}, r_{\theta(p_n,m)}) + \sum_{l=1}^{n_o} L_{\text{match}} (o^*_l, o_{\theta(l)})$$

(13)

where $r^*_{p_n,m}$ and $o^*_l$ denote ground-truth relation predicates and entity categories respectively. $L_{\text{match}} (r^*_{p_n,m}, r_{\theta(p_n,m)})$ and $L_{\text{match}} (o^*_l, o_{\theta(l)})$ are entropy-based log-likelihood matching cost functions, which are defined as:

$$L_{\text{match}} (r^*_{p_n,m}, r_{\theta(p_n,m)}) = - \mathbb{1}_{\{c^*_{p_n,m} \neq \emptyset\}} \log P (c_{\theta(p_n,m)} = c^*_{p_n,m})$$

(14)

$$L_{\text{match}} (o^*_l, o_{\theta(l)}) = - \mathbb{1}_{\{c^*_l \neq \emptyset\}} \log P (c_{\theta(l)} = c^*_l)$$

(15)

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function, $c^*_{p_n,m}$ and $c^*_l$ denote logits of relation predicates and entities distribution.

4 Experiments

4.1 Dataset

We train and evaluate our model on the Action Genome (AG) dataset [Ji et al., 2020], which describes relationships over time. It regularly selects some frames in the video and annotates the position information of entities and the relationships between entity pairs. AG contains 17M human-object-relationship instances with 35 object categories and 25 relation predicates categories. These 25 relation predicates are subdivided into three different types, which are attention relationships, spatial relationships and contact relationships.

4.2 Evaluation Metrics

Following the scene graph generation task, we evaluate our framework in three different modes: Predicate classification (PREDcls): given the ground-truth bounding box and entity categories, predict relation predicates between entities. Scene Graph classification (SGcls): given the ground-truth bounding box, predict entity categories and relation predicates between entities. Scene graph detection (SGdet): given an image, predict bounding boxes and categories of entities, and relation predicates between entities. For SGcls and SGdet, we can not obtain the full ground-truth entities information directly, so we utilize a detector to detect objects. The detection strategy is that the predicted box is considered correctly when it has at least 0.5 IoU (Intersection over Union) overlap with the ground-truth box. We adopt the widely used Recall@K metrics ($K = [10, 20, 50]$) to evaluate our model, which calculates the recall in the most important (top-1) relation predicates.

4.3 Implementation Details

In this section, we introduce the details of the experimental setting and dataset.

**Detector.** Following previous work, we adopt Faster R-CNN with ResNet-101 as the backbone network to detect objects.

**Parameter Settings.** For the feature detector, we map the visual features to a vector of dimension 512 and the semantic features of the object categories to a vector of dimension 300. The MLPs in the paper are three-layer fully connected network and the hidden layer dimension is set to 512.

**Training Details.** In the grafting stage, we adopt the original RelTR model [Cong et al., 2022], changing only the output number of the classifier. The Action Genome dataset [Ji et al., 2020] is converted to COCO-format for fine-tuning. The RelTR model is fine-tuned for total 20 epochs with mini-batch size 8 in this stage. The initial learning rates of the classifier are unchanged and the learning rates of the other layers are multiplied by 0.9 of the initial learning rate. In the reassembling stage, we train the temporal dependency model (TDM) by SGD optimizer for total 15 epochs with mini-batch size 1. The initial learning rate is set to 1e-5 and adjusted to 5e-6 after the 5 epochs of training and to 1e-6 after 10 epochs of training. In the noise filter (NFT), we set the similarity threshold to 0.9.

4.4 Comparisons with State-of-the-Arts

To verify the superiority of GTR, we compared it with 8 state-of-the-art scene graph generation methods on the Action Genome dataset [Ji et al., 2020].

**VRD** [Lu et al., 2016] propose a relationship detection method, training two separate vision models, one to recognise objects and the other to recognise relationships.

**Motif Freq** [Zellers et al., 2018] investigate the problem of producing structured graph representations of visual scenes and propose a new stacked motif networks for capturing higher order motifs.

**MSDN** [Li et al., 2017b] propose a new end-to-end neural network model to exploit the interconnections across different semantic levels.

**VCTREE** [Tang et al., 2019] propose to compose dynamic tree structures that place the objects in an image into a visual context, helping scene graph generation.

**RelDN** [Zhang et al., 2019] improve the relationship detection network and propose a corresponding contrastive loss construction method that accurately identifies the specific relationship between two entities.

**GPS-Net** [Lin et al., 2020] propose a graph property sensing network that fully explores the edge direction information, the difference in priority between nodes and the long-tailed distribution of relationships.

**STTran** [Cong et al., 2021] propose a spatial-temporal transformer model to identify the relationships between entities.

**AP-Net** [Li et al., 2022] propose anticipatory pre-training paradigm based on transformer to model the temporal correlation of visual relationships, consider both global and local information.

As shown in Table 1, our framework outperforms the previous state-of-the-art method in all evaluation metric, improves it 1.8% on PREDcls-R@10, 1.5% on SGcls-R@10 and 1.6% on SGdet-R@10. From the experimental results, we can observe that our framework shows greater improvement with R@10 than with R@20 and R@50, which can indicate that our framework has high prediction efficiency in both PREDcls, SGcls and SGdet, i.e., more correct relation predicates can be
generated with a small number of recalls. It is worth noting that the complete GTR has a significant performance improvement compared to the GTR without the reassembling stage phase, which indicates that the modeling temporal contextual information is the key to our framework.

### 4.5 Ablation Study

In this section, we conduct experiments to verify the effectiveness of the components in the reassembling stage.

**Impact of Noise Filter (NFT).** NFT is proposed to filter the redundant static relation predicates generated by the grafting stage. We investigate the impact of NFT by removing it. The results in the second row of Table 2 demonstrate the effectiveness of using NFT to remove redundant static relation predicates, leading to a significant increase in their availability.

**Impact of Mask Strategy.** To capture the fine-grained temporal dependencies in the video, we propose the mask strategy in the temporal attention module. As shown in the third row of Table 2 result, the performance of the framework degrades to a certain extent when removing mask strategy, demonstrate that mask strategy can improve the extraction process of temporal dependencies, leading to enhancement of temporal attention module performance.

### 4.6 Analysis

In this section, we conduct further analytical experiments to evaluate our framework.

**Performance of Spatio-temporal Interaction.** We evaluate the performance of spatio-temporal interaction by having the GTR distinguish consecutive actions with similar visual content. We selected two most common samples for evaluation in Action Genome [Ji et al., 2020], where each sample contains two consecutive actions with highly similar visual feature, i.e., holding → drinking from and holding → eating. The results are shown in Table 4. Compared to STTran [Cong et al., 2021], our GTR achieves superior performance in distinguishing consecutive actions with temporal dependencies, demonstrating that our framework can better capture spatio-temporal interactions.

**Number of Video Data.** We investigate the effect of the training video data magnitude on the performance of GTR. We initially started the experiment using 40% of the data. Video data magnitude below 40% can result in entity and relation predicate categories not being fully covered.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>STTran</td>
<td>21/30</td>
</tr>
<tr>
<td>GTR</td>
<td>25/30</td>
</tr>
</tbody>
</table>

Table 4: The precision of distinguish similar consecutive actions. We select two samples (i.e., holding and drinking from, holding and eating), each containing 30 sets. Compared to STTran [Cong et al., 2021], GTR can predict the consecutive actions more accurately based on the accurate spatio-temporal interaction.
Increasing it by 10% each time. As shown in Figure 5, GTR outperform the previous state-of-the-art method in SGdet with 50% data training. When the training data reaches 60%, GTR can outperform the previous state-of-the-art method in all modes. These observations validate our motivation that GTR can achieve excellent performance without the need for extensive video data training.

Number of Candidate Static Relation Predicates. We investigate the effect of the number of candidate relation predicates on our framework by adjusting the number of predicates recalled in the grafting stage. The results are shown in Figure 6. As the number of candidates $K$ increases (the relation predicates may be recurring), the performance of our framework (green fold line) gradually improves and the best performance is achieved at $K=30$. However, when the number of recalled candidates $K=40$, the accuracy of the static relation predicates (red fold line) still improves but the performance of our framework decreases, indicating that the framework captures redundant relation predicates, which will weaken the performance of our framework.

4.7 Qualitative Results

The qualitative results are shown in Figure 4. We select two consecutive frames from the video, where the blue boxes are the correct detection results and the pink boxes are the correct relation predicates and the gray boxes are the incorrect relation predicates. The scene graphs are generated with the top-1 confidence relation predictions. The case shows that our model can detect most of the relationships. Some relation predictions could not be predicted due to unclear visual features (e.g., look at). Compared to previous model [Cong et al., 2021], our framework is able to accurately predict actions based on time dependence even the action is not obvious (e.g., touch).

5 Conclusion

In this paper, we propose a Grafting-Then-Reassembling (GTR) framework for dynamic scene graph generation to decouple spatio-temporal contextual information in video. We firstly graft a static scene graph generation model to generate static visual relationships within frames. Then, we introduce the temporal dependency model to extract temporal dependencies across frames. Finally, we explicitly reassemble the static visual relationships into dynamic scene graphs. Experimental results on the benchmark dataset demonstrate the effectiveness of our proposed framework.
Acknowledgements
We thank anonymous reviewers for their insightful feedback that helped improve the paper. The research in this article is supported by the National Key Research and Development Project (2021YFF0901600), the National Science Foundation of China (U22B2059, 61976073, 62276083), and Shenzhen Foundational Research Funding (JCYJ20200109113441941). Major Key Project of PCL (PCL2021A06).

References


