VRDFormer: End-to-End Video Visual Relation Detection with Transformers

Sipeng Zheng
Renmin University of China
zhengsipeng@ruc.edu.cn

Shizhe Chen
Inria
shizhe.chen@inria.fr

Qin Jin[*]
Renmin University of China
qjin@ruc.edu.cn

Abstract

Visual relation understanding plays an essential role for holistic video understanding. Most previous works adopt a multi-stage framework for video visual relation detection (VidVRD), which cannot capture long-term spatio-temporal contexts in different stages and also suffers from inefficiency. In this paper, we propose a transformer-based framework called VRDFormer to unify these decoupling stages. Our model exploits a query-based approach to autoregressively generate relation instances. We specifically design static queries and recurrent queries to enable efficient object pair tracking with spatio-temporal contexts. The model is jointly trained with object pair detection and relation classification. Extensive experiments on two benchmark datasets, ImageNet-VidVRD and VidOR, demonstrate the effectiveness of the proposed VRDFormer, which achieves the state-of-the-art performance on both relation detection and relation tagging tasks. The code is released at https://github.com/zhengsipeng/VRDFormer_VRD.

1. Introduction

Video visual relation detection (VidVRD) [32] aims to detect all *relation instances* in the video. Each instance contains a subject, an object and their relationship, as well as the spatial and temporal locations of the subject and the object. This task has attracted more and more attention in recent years, as it serves as a bridge to connect basic vision tasks (*e.g*. object detection [5, 12, 54] and tracking [11, 47]) with more complicated video semantic understanding tasks (*e.g*. captioning [43] and VideoQA [25]).

One typical approach for VidVRD [30, 32, 36] is to decompose the task in a multi-stage pipeline. These works, as illustrated in Figure 1, firstly employ off-the-shelf object detectors [27, 54] to detect and track objects in a video, and then, enumerate every two object tracklets and use temporal sliding window to obtain tracklet pairs. Finally, they filter out invalid tracklet pairs, then predict relation types for the remained ones using region-of-interests (RoI) features cropped from pre-computed CNN feature maps [3, 27].

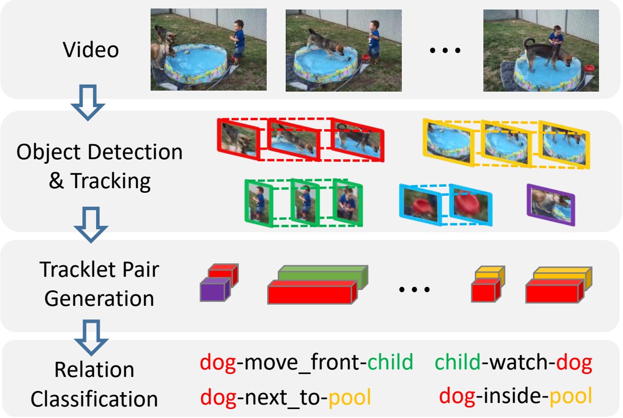

Figure 1. Existing VidVRD methods adopt a multi-stage pipeline. They suffer from limited spatio-temporal contexts, overly sampled tracklet pairs, and independently optimized modules.

We consider that there are three major limitations in such multi-stage framework. First, spatio-temporal contexts are not well exploited in tracklet pair generation. In fact, spatio-temporal contexts not only can enhance the model's ability to localize objects, but also provide valuable information to infer the presence or absence of a relation. For example, the detection of the subject/object when occlusion occurs can get help from the temporal context. The relation reasoning can benefit from the spatial context. Although context has been widely adopted for the final relation classification step [39, 44], it has not been well explored in the detection of relation instances in video. Therefore, object detection and tracking might not be very accurate in these methods, resulting in accumulated errors to the following stages. Second, in previous works, each module is independently trained. However, object detection, tracking and relation classification are highly correlated and could promote each other through joint learning. Last but not least, since tracklet pairs are exhaustively generated, many of them do not have meaningful relations, which not only harms

[*]Qin Jin is the corresponding author.

computational efficiency but also influences classification performances.

To address above limitations, in this work, we propose a unified transformer-based video visual relation detection framework named VRDFormer. It consists of a video encoding module and a query-based relation instance generation module to detect relations in an autoregressive manner. Specifically, we adopt a query-based approach to detect and track object pairs. We propose two types of queries for object pairs generation in videos, namely static and recurrent queries. The *static queries* detect new object pairs in each frame which can aggregate the spatial contexts via transformer attention mechanism, while the *recurrent queries* aggregate the temporal contexts across frames to track previously detected object pairs. We keep all tracklet pairs in a memory and use a transformer-based model to classify relations for each tracklet pair which reserves long-term spatio-temporal history. The whole model is end-to-end trained by the object pair detection and relation classification tasks jointly. We conduct extensive experiments on two benchmark datasets to evaluate the model. VRDFormer achieves the state-of-the-art performance in relation detection and relation tagging on the two datasets.

In summary, our contributions are as follows:

- We propose a unified one-stage model VRDFormer for video visual relation detection (VidVRD), which is able to perform tracklet pair generation and relation classification simultaneously.
- We design static queries and recurrent queries to aggregate spatio-temporal contexts, which enable more convenient temporal association for object pairs across frames and more effective relation classification.
- We carry out extensive experiments and analysis on two benchmark datasets and achieve the state-of-the-art performance on both datasets.

2. Related Work

Image Visual Relation Detection (ImgVRD). Relations among objects play an important role in image understanding, and thus the task of ImgVRD has received much attention in recent years [6, 22, 42, 45, 46]. Earlier works adopt a two-stage framework [16, 17, 44, 55], which first detects objects then predicts relations for every pair of objects in the image. They mainly focus on the second relation prediction stage, for example, employing graph neural networks to encode more contexts [39], fine-grained pose features [38,50], or language priors [22,49]. However, these approaches suffer from accumulated errors in two-stage processing and computation inefficiency. Recently, one-stage models [18] are emerging for ImgVRD to address these limitations. Different from previous works which adopt a CNN-based architecture, HOTR [14] and QPIC [35] instead utilize transformer architectures with query-based pairwise detection to benefit from global spatial contexts.

Video Visual Relation Detection (VidVRD): VidVRD [7, 26, 33, 34] is a more challenging task compared to ImgVRD, involving more diverse relation types and object spatial-temporal localization. Most existing works follow a multi-stage pipeline [32], such as object detection, object tracking, tracklet pair generation and relation classification. These works focus on improving relation classification by leveraging contextual knowledge [26,36], inter-dependency or long-range temporal information [20], while simply using off-the-shelf models like Faster-RCNN [27] for object detection or Deep-Sort [41] for tracking. 3DRN [1] is the only one-stage model that unifies object detection, tracking and relation classification based on I3D backbone [3]. Though improved efficiency, 3DRN shows poor performance on localization compared to multi-stage methods, since it fails to leverage rich localization knowledge from pre-trained object detectors or tracking models.

Transformers in Vision. Transformer [37] has achieved significant progress [13] in vision tasks including image classification [8], object detection [2,53] and image relation detection [14,35]. One typical approach is DETR [2], which decodes a set of queries into object proposals in parallel by regarding object detection as a set prediction problem. Recently, some works explore to extend such query-based architecture in video domain [4, 23, 40]. Among them, Meinhardt *et al.* [23] propose a new concept named track query that can follow an object over time for tracking. Inspired by the success of transformers in many vision tasks, we explore the transformer architecture for VidVRD.

3. Methodology

The video visual relation detection (VidVRD) task aims to detect all *relation instances* in the video. Each relation instance is denoted as $(s, r, o, \mathcal{T}^s_{t_1:t_n}, \mathcal{T}^o_{t_1:t_n})$, where s, r, o denote the triplet classes of subject, relation and object while $\mathcal{T}^s_{t_1:t_n}$, $\mathcal{T}^o_{t_1:t_n}$ denote tracklets of the subject and the object between the start and end timestamp t_1 and t_n in the video. $\mathcal{T}^s_{t_1:t_n}$ consists of $(b^s_{t_1}, \cdots, b^s_{t_n})$ where $b^s_{t_i}$ is the bounding box of subject at time t_i. Similarly, $\mathcal{T}^o_{t_1:t_n}$ is represented as $(b^o_{t_1}, \cdots, b^o_{t_n})$. In the following, we first present the overall framework of our model VRDFormer in Sec 3.1, then introduce its key ingredient, query-based relation instance generation in Sec 3.2. Finally, we describe training and inference algorithms for VRDFormer in Sec 3.3 and Sec 3.4 respectively.

3.1. Overall Framework

VRDFormer consists of a video encoding module and a query-based relation instance generation module.

The video encoding module encodes a video into a

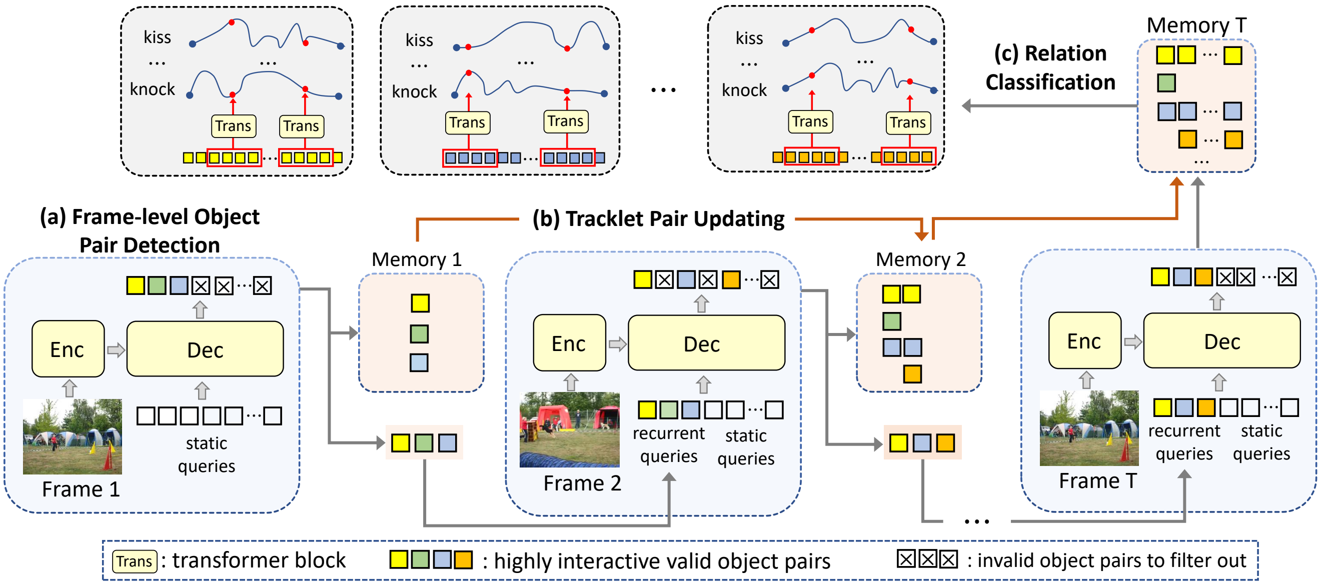
kiss
…
knock
Trans
Memory T
(c) Relation Classification
(a) Frame-level Object Pair Detection
(b) Tracklet Pair Updating
Memory 1
Memory 2
Enc
Dec
static queries
recurrent queries
Frame 1
Frame 2
Frame T
Trans : transformer block
: highly interactive valid object pairs
☒☒☒ : invalid object pairs to filter out

Figure 2. Overall framework of the query-based relation instance generation module in our VRDFormer. It consists of: **(a) frame-level object pair detection** (blue module), which detects object pairs based on static or recurrent queries for each frame; **(b) tracklet pair updating** (orange module), which autoregressively updates a tracklet pair memory frame by frame. **(c) relation classification** (grey module), which predicts relations per frame for each detected tracklet pair with temporal aggregation to model the long-term dependencies.

sequence of frame-level feature maps. It contains a CNN backbone and a multi-layer transformer [37]. For each frame in the video, the CNN backbone extracts feature maps with local spatial contexts, and then the transformer uses self-attention to capture global spatial contexts. Positional embeddings [24] are added to enhance spatial information.

The query-based relation instance generation module (illustrated in Figure 2) processes the encoded feature maps frame-by-frame and generates relation instances in an autoregressive manner. It consists of three sub-modules: 1) frame-level object pair detection; 2) tracklet pair updating; and 3) relation classification. The first sub-module (Sec 3.2.1) detects *object pairs* with a query-based approach [2] for each frame t. For each object pair in the form of (s, o, b_t^s, b_t^o), an interactiveness probability p_t^{intr} is predicted to denote whether there is certain type of relations between two objects. Then, the second sub-module (Sec 3.2.2) connects object pairs along the temporal axis into tracklet pairs. At each frame step, the module updates a tracklet pair given current predictions then stores its current status into a memory bank. A tracklet pair in the memory is formulated as $(s, o, \mathcal{T}^s_{<t}, \mathcal{T}^o_{<t})$, which includes all the spatio-temporal locations and features of the same s and o instance until frame t. The highly interactive object pair in frame t with a large p_t^{intr} will either initialize a new tracklet pair or expand the trajectories $(\mathcal{T}^s_{<t}, \mathcal{T}^o_{<t})$ of an existing tracklet pair. Finally, for each tracklet pair in the memory bank, the last sub-module (Sec 3.2.3) predicts its relation class at its every occurred frame t given temporal and spatial contexts. The whole model can be end-to-end trained with multiple tasks, such as object detection and relation classification.

3.2. Query-based Relation Instance Generation

3.2.1 Frame-level Object Pair Detection

This sub-module consists of three components: input queries, a transformer decoder and prediction heads.

Static and Recurrent Queries. We utilize query vectors to extract contextual information from the video for object pair detection. Similar to query-based ImgVRD [35], each query captures information to decode at most one object pair. However, unlike ImgVRD where an object pair only occurs in a single image, an object pair in the VidVRD task can evolve over time in the video, so associating object pairs across different images poses a great challenge to the model. To support convenient and effective temporal associations of object pairs, we propose a new type of queries inspired by [23], which recurrently gathers spatio-temporal contexts to track previous object pairs in subsequent frames.

To be specific, there are two types of query vectors, namely *static queries* and *recurrent queries*. The static queries contain a fixed number of learnable query vectors $(q_1^{\text{st}}, \cdots)$ to detect new object pairs in each frame, while the number of recurrent queries $(q_1^{\text{re}}, \cdots)$ is dynamic. As can be seen in Figure 2(b), each recurrent query inherits feature embeddings of an object pair in the previous frame, and thus carries previous semantic and location information to localize the same instance in the current frame. We will describe how to obtain and update recurrent queries in Sec 3.2.2. The static and recurrent queries are concatenated together as input to the decoder presented below.

Transformer Decoder. The decoder is a multi-layer transformer which transforms query vectors $(q_1^{\text{st}}, \cdots, q_1^{\text{re}}, \cdots)$ into contextualized output embeddings $(d_1^{\text{st}}, \cdots, d_1^{\text{re}}, \cdots)$ for object pair detection. There are two types of attention per transformer layer: 1) self-attention over queries to model their correlations, and 2) cross-attention between queries and feature maps of the current frame to incorporate each query with image-wide contexts of the frame.

Prediction Heads. Given the output embedding of an input query, we utilize five prediction heads implemented as multi-layer perceptron (MLP) to predict: 1) subject bounding box $b_i^s \in \mathbb{R}^4$; 2) object bounding box $b_i^o \in \mathbb{R}^4$; 3) subject class probability $p_i^s \in \mathbb{R}^{N_{\text{obj}}}$; 4) object class probability $p_i^o \in \mathbb{R}^{N_{\text{obj}}}$; and 5) interactiveness probability for each relation type $p_i^{\text{intr}} \in \mathbb{R}^{N_{\text{rel}}}$ with sigmoid activation function, where $N_{\text{obj}}, N_{\text{rel}}$ are the total number of object and relation classes respectively. The p_i^{intr} denotes an initial relation score during tracking. Therefore, each query represents an object pair (s_i, o_i, b_i^s, b_i^o), where $s_i = \arg\max p_i^s$, $o_i = \arg\max p_i^o$. To be noted, the notations of the queries and predictions are for each frame, whereas we omit the subscript t for notation simplicity.

3.2.2 Tracklet Pair Updating

We keep a memory bank to store tracklet pairs as illustrated in Figure 2(b). Thanks to our design of static and recurrent queries, it is straightforward to initialize new tracklet pairs or associate object pairs to existing tracklet pairs.

Initialization of New Tracklet Pairs. Object pairs predicted from static queries denote new tracklet pairs, which firstly occur in the frame (except for the re-activated pairs described in “Expansion of Existing Tracklet Pairs” below). For each static query q_i^{st}, we compute a confidence score θ_i^{st} as follows:

$$\theta_i^{\text{st}} = \max p_i^s \cdot \max p_i^{\text{intr}} \cdot \max p_i^o. \qquad (1)$$

We only select static queries with confidence score larger than a threshold θ_{intr} to filter out invalid object pairs without interaction. We use their predictions $\{s_i, o_i, b_i^s, b_i^o\}$ together with the corresponding output embedding d_i^{st} to initialize new tracklet pairs in the memory. The output embedding d_i^{st} is used as a recurrent query for the next frame.

Expansion of Existing Tracklet Pairs. A recurrent query corresponds to an existing tracklet pair in the memory bank by its definition. For each recurrent query q_i^{re}, we can calculate a confidence score θ_i^{re} at the current frame similar to Eq (1). If θ_i^{re} of the query is below threshold θ_{intr}, or its object class predictions are different from the classes of its corresponding tracklet pair, the recurrent query will be inactivated. We only add bounding box predictions of active recurrent queries into their aligned tracklet pairs in the memory. For inactivated recurrent queries, we do not discard them immediately. Instead, we wait for up to T_{re} frames as in [23] because the occlusions, shadows or other unexpected circumstances may make predictions at a frame unstable. If a static query at one frame shares the same subject and object classes with an inactive recurrent query and their IoU of bounding boxes is larger than a threshold θ_{re}, this recurrent query will be re-activated for tracking again. We use linear interpolation to fill the missing predictions during the inactive period. The output embedding d_i^{re} of a remained recurrent query continues to serve as a recurrent query for the next frame. Such autoregressive manner enables recurrent queries to record long-term temporal contexts of previous frames.

3.2.3 Relation Classification

For each completed tracklet pair in the memory, the relation classification module predicts its relation class for every frame as shown in Figure 2(c). Both spatial and temporal contexts are essential to recognize the relation of an frame-level object pair at any frame t. To this end, we utilize query output embedding d_t for the relation classification, which captures image-wide spatial contexts at frame t and its previous temporal contexts. In order to further smooth the frame-level relation prediction of a tracklet pair, we further consider d_t and query output embeddings of its previous T frames, denoted as $D_t = \{d_{t-T+1}, \cdots, d_t\}$, in prediction. We employ another transformer to aggregate D_t. Specifically, we concatenate a learnable token [cls] with D_t, and add temporal positional encoding to each token according to their frame index. The output embedding of the [cls] token is fed into a MLP head to predict a relation probability $p_t^r \in \mathbb{R}^{N_{\text{rel}}}$ for the tracklet pair at frame t. Compared to the initial relation score p_i^{intr} in Sec 3.2.1, p_t^r can improve relation prediction with more accurate spatial-temporal contexts.

3.3. Training VRDFormer

We jointly train VRDFormer with the object pair detection and relation classification tasks.

3.3.1 Task I: Object Pair Detection

In this task, we train video encoding and frame-level object pair detection modules. To improve training efficiency and reduce memory burden, we only sample two frames (t_1, t_2) with a short interval in the video to mimic the tracking procedure as [23]. For each frame, we first calculate a bipartite matching between predictions by the model and the groundtruth, and then compute losses for matched pairs.

Bipartite Matching. The groundtruth for each frame is a sequence of annotated object pairs. We pad the groundtruth with $\varnothing$ (no interaction), so that the groundtruth sequence has the same size as the queries to be matched. We use Hungarian algorithm [2] to find an optimal mapping with minimal matching cost between the groundtruth and the predictions of queries for each frame. Assume $y_i = (s_i, o_i, b_i^s, b_i^o)$ is the prediction of a query q_i, and $y_j^* = (s_j^*, o_j^*, b_j^{s*}, b_j^{o*})$ is the j-th object pair in groundtruth. In addition, p_i^{intr} is the interactiveness probability of q_i and $p_i^{\text{intr}*}$ denotes the relation label vector of y_j^*. The matching cost $\mathcal{C}_{\text{match}}(y_i, y_j^*)$ is computed as follows:

$$\begin{aligned} \mathcal{C}_{\text{match}}(y_i, y_j^*) = &- p_i^s[s_j^*] - p_i^o[o_j^*] - \frac{1}{2}\mathcal{C}_{\text{intr}}(p_i^{\text{intr}}, p_j^{\text{intr}*}) \\ &+ \lambda_{\text{box}}\mathcal{C}_{\text{box}}(b_i^s, b_j^{s*}, b_i^o, b_j^{o*}) \end{aligned} \tag{2}$$

where $p_i^s[s_j^*]$, $p_i^o[o_j^*]$ denote the probability of groundtruth class s_j^* and o_j^*. $\mathcal{C}_{\text{intr}}$ is an interaction cost defined as:

$$\mathcal{C}_{\text{intr}}(p_i^{\text{intr}}, p_j^{\text{intr}*}) = \frac{(p_j^{\text{intr}*})^{\mathrm{T}} p_i^{\text{intr}}}{||p_j^{\text{intr}*}||_1 + \epsilon} + \frac{(1 - p_j^{\text{intr}*})^{\mathrm{T}}(1 - p_i^{\text{intr}})}{||1 - p_j^{\text{intr}*}||_1 + \epsilon} \tag{3}$$

Eq (3) balances the number of positive and negative relation classes. $\mathcal{C}_{\text{box}}$ is the box cost with a balance factor λ_{box} to measure the alignment of object bounding boxes:

$$\begin{aligned} \mathcal{C}_{\text{box}}(b_i^s, b_j^{s*}, b_i^o, b_j^{o*}) = &\max\{||b_i^s - b_j^{s*}||_1, ||b_i^o - b_j^{o*}||_1\} \\ &+ \max\{-\text{GIoU}(b_i^s, b_j^{s*}), -\text{GIoU}(b_i^o, b_j^{o*})\} \end{aligned} \tag{4}$$

where $||\cdot||_1$ and GIoU denote the L1 norm and generalized IoU [28]. Same as [35], we minimize the larger one of the subject and object box costs to avoid undesirable matching bias when one cost is significant small than the other.

For the first frame t_1, as there is no recurrent query, we only align the groundtruth with predictions of static queries. For the second frame t_2, each recurrent query tracks an existing object pair detected in frame t_1, and thus it inherits the groundtruth alignment in t_1. Therefore, if the previous object pair instance still occurs in groundtruth of frame t_2, its recurrent query can be directly mapped with the groundtruth item, otherwise the recurrent query is mapped to $\varnothing$. The other unmatched groundtruth items are used in bipartite matching with static queries via the Hungarian algorithm. In this way, we obtain the matching with the groundtruth for each query prediction at frame t_1 and t_2.

Prediction Loss. After we find an optimal matching σ, the i-th groundtruth y_i^* can be mapped to the $\sigma(i)$-th query prediction, we compute a loss for $(y_{\sigma(i)}, y_i^*)$ as follows:

$$\begin{aligned} \mathcal{L}_{\text{det}}(y_{\sigma(i)}, y_i^*) = -\mu_{\text{cls}}\big(&\log p_{\sigma(i)}^s[s_i^*] + \log p_{\sigma(i)}^o[o_i^*]\big) + \\ &\mu_{\text{intr}}\mathcal{L}_{\text{intr}}(p_{\sigma(i)}^{\text{intr}}, p_i^{\text{intr}*}) + \\ &\mu_{\text{box}}\mathbb{I}_{i \notin \Omega}\big(\mathcal{L}_{box}(b_{\sigma(i)}^s, b_i^{s*}) + \mathcal{L}_{box}(b_{\sigma(i)}^o, b_i^{o*})\big) \end{aligned} \tag{5}$$

Ω denotes the set of groundtruth that corresponds to "no interaction", and $\mu_{\text{box}}, \mu_{\text{cls}}, \mu_{\text{inter}}$ are three scaling factors. The $\mathcal{L}_{\text{intr}}$ is the binary cross entropy loss. $\mathcal{L}_{\text{box}}(b_{\sigma(i)}^s, b_i^{s*})$ for the subject is computed as follows:

$$\mathcal{L}_{\text{box}}(b_{\sigma(i)}^s, b_i^{s*}) = ||b_{\sigma(i)}^s - b_i^{s*}||_1 - \text{GIoU}(b_{\sigma(i)}^s, b_i^{s*}) \tag{6}$$

$\mathcal{L}_{\text{box}}(b_i^o, b_i^{o*})$ for the object is computed similarly as Eq (6). The final loss is the average loss of all matched pairs in frame t_1 and t_2.

3.3.2 Task II: Relation Classification

In this task, we further train relation classification module with other parameters given groundtruth object pair locations. As we have groundtruth object pairs for each frame, we could use the exact number of queries as input per frame. Different from the inference as described in Sec 3.4, the input embedding of each query can be directly initialized by the RoI aligned features [9] of groundtruth object pair in the encoded feature maps. In this way, we reduce the number of queries for each frame, and are able to employ more frames with longer temporal duration for relation classification. Given the relation prediction p_t^r of a tracklet pair at frame t, and its groundtruth label r_t^*, the classification loss at frame t is $\mathcal{L}_{\text{rel}} = -\log\ p_t^r[r_t^*]$. We average the loss over all frames and traklet pairs.

3.4. Inference

During inference, VRDFormer detects object pairs frame-by-frame with static and recurrent queries, updates the tracklet pair memory, and predicts frame-level relations for each tracklet pair in the memory. Suppose there are N tracklet pairs in the memory, where each pair is $(s, o, \{p_t^r\}_{t=t_1}^{t_n}, \{b_t^s\}_{t=t_1}^{t_n}, \{b_t^o\}_{t=t_1}^{t_n})$. As shown in Figure 2(c), we can plot an interactive curve over time for each relation type of the tracklet pair. Therefore, we decompose these N tracklet pairs into $N \times N_{rel}$ relation tracklet pairs. We utilize watershed algorithm similar to [48] to obtain relation instance proposals for each relation tracklet pair. Finally, we select top K relation instances according to their relation probabilities. More details can be seen in the supplementary material.

4. Experiments

4.1. Experimental Setup

Datasets. We carry out experiments on two video relation benchmarks: ImageNet-VidVRD [32] and VidOR [30]. ImageNet-VidVRD is a subset of ILSVRC2016-VID [29], which contains 1,000 videos and labels of 35 object classes and 132 relation classes and 4,835 relation instances in total. It is split into a training set and a testing set with 800 and 200 videos respectively. VidOR is a larger dataset than ImageNet-VidVRD. It contains 10,000 videos and 378,546 relation instances with 80 object classes and 50 relation classes. It is split into 7,000 videos for training, 835 videos for validation, and 2,165 videos for testing.

Evaluation Protocol. We follow the standard evaluation protocol in [32] which includes two sub-tasks: relation detection and relation tagging. The relation detec-

Table 1. Comparison with previous works on the relation detection task. "Det Data" denotes additional datasets used by different models for object detection: COCO (MS-COCO [19]), DET (ILSVRC2016-DET [29]) and OPEN-IMG (OpenImage [15]).

Method	Det Data	Relation Detection		
		mAP	R@50	R@100
ImageNet-VidVRD				
VIDVRD [32]	COCO+DET	8.58	5.54	6.37
GSTEG [36]	COCO+DET	9.52	8.67	7.05
VRD-GCN [26]	COCO	14.23	7.43	8.75
3DRN [1]	-	14.68	5.53	6.39
MAGUS.Gamma [34]	-	6.56	6.89	8.83
VRD-STGC [20]	COCO+DET	18.23	11.21	13.69
VIDVRD II [31]	OPEN-IMG	29.37	19.63	22.92
VRDFormer (Ours)	COCO	**32.43**	**21.92**	**25.40**
VidOR				
RELAbuilder [51]	COCO+DET	1.47	1.58	1.85
3DRN [1]	-	2.47	2.58	2.75
VRD-STGC [20]	COCO+DET	6.85	8.21	9.90
VIDVRD II [31]	OPEN-IMG	8.65	8.59	10.69
VRDFormer (Ours)	COCO	**11.19**	**11.05**	**13.34**

tion sub-task evaluates the precision of relation instances $(s, r, o, \mathcal{T}^s_{t_1:t_n}, \mathcal{T}^o_{t_1:t_n})$. A predicted relation instance is considered as positive only when the vIoU (voluminal Intersection-over-Union) between its subject/object trajectory and the groundtruth subject/object trajectory is larger than 0.5 and classifications for s, r, o are all correct. This sub-task is evaluated by Mean Average Precision (mAP), Recall@50 (R@50) and Recall@100 (R@100). Recall@K denotes the fraction of positive relation instances in top K predictions. The relation tagging sub-task instead provides the groundtruth trajectories and only evaluates the model on the prediction of s, r, o. We use Precision@K (P@1, P@5 and P@10) as the metrics for relation tagging. We also propose a tracklet pair detection task in our ablations, which evaluates the precision of tracklet pairs $(s, o, \mathcal{T}^s_{t_1:t_n}, \mathcal{T}^o_{t_1:t_n})$ without considering the relation precision.

Implementation Details We use ResNet-101 [10] as the CNN backbone and the same transformer encoder and decoder architecture as Deformable-DETR [53] which consists of 6 transformer layers. The VRDFormer model is initialized from Deformable-DETR [53] pre-trained on MS-COCO dataset [19]. We train VRDFormer with AdamW [21] optimizer. The learning rate is set to 10^{-5} for CNN backbone and 10^{-4} for remaining modules in Task I training. The learning rate is divided by 10 in Task II training. The dimension of hidden layers in transformer is set to 256. We set the number of static queries and the maximum number of recurrent queries as 100. We adopt the full two-task training strategy for relation detection by alternately using one mini-batch to train Task I and then another mini-batch to train Task II, and we only train Task II for the relation tagging task. It takes 32 hours to train on VidOR using 8 V100 GPUs and the inference speed is 18.2 FPS. More details are presented in the supplementary material.

Table 2. Comparison results with previous state-of-the-art works on the relation tagging task.

Method	Det Data	Relation Tagging		
		P@1	P@5	P@10
ImageNet-VidVRD				
VIDVRD [32]	COCO+DET	43.0	28.9	20.8
GSTEG [36]	COCO+DET	51.5	39.5	28.23
3DRN [1]	-	57.89	41.80	29.15
VRD-GCN [26]	COCO	59.5	40.5	27.85
VRD-STGC [20]	COCO+DET	60.0	43.1	32.24
VIDVRD II [31]	OPEN-IMG	70.4	53.88	40.16
VRDFormer (ours)	COCO	**73.0**	**57.1**	**44.75**
VidOR				
RELAbuilder [51]	COCO+DET	33.05	35.27	-
VRD-STGC [20]	COCO+DET	48.92	36.78	-
MAGUS.Gamma [34]	-	51.20	41.73	-
3DRN [1]	-	52.59	42.33	29.89
VIDVRD II [31]	OPEN-IMG	57.4	44.54	33.3
VRDFormer (ours)	COCO	**63.71**	**51.07**	**39.89**

4.2. Comparison with State of the Art

Table 1 compares our VRDFormer with state-of-the-art methods on the relation detection task. Our model achieves the best performance under all the three evaluation metrics. Compared with VidVRD II [31], we use much less data for object detection but still achieve +3.06%, +2.29% and +2.48% improvements for mAP, R@50 and R@100 respectively on the ImageNet-VidVRD dataset. We obtain similar improvements on the large-scale VidOR dataset as well, with +2.54%, +2.46% and +2.65% for the three metrics respectively. We also compare VRDFormer with some graph-based methods such as VRD-STGC [20] or VRD-GCN [26], which focus on learning from contextualized information as well. Our performance is more encouraging, which shows more than +10% improvement on ImageNet-VidVRD and +3% improvement on VidOR using similar collection of data for object detection. Such results indicate that our model is more effective and efficient for localization of relation instance by leveraging both spatial and temporal contexts. Table 2 presents the comparison results for the relation tagging task. VRDFormer achieves state-of-the-art results on both datasets as well.

Please note that we also compare VRDFormer with another end-to-end method 3DRN [1]. VRDFormer outperforms 3DRN significantly in both relation detection and relation tagging tasks on both datasets. For example, VRDFormer achieves +17.75%, +16.39% and +19.01% improvement for relation detection on metrics mAP, R@50 and R@100 respectively on ImageNet-VidVRD. It achieves +15.11%, +15.3% and +15.6% improvement for relation

Table 3. Ablations of recurrent queries and re-activate strategies.

	Re-Activate	Recurrent Query	Relation Detection		Tracklet Pair Detection	
			mAP	R@50	R@50	R@100
1	×	×	27.25	17.04	26.91	30.13
2	×	✓	30.67	20.33	29.62	31.98
3	✓	✓	**32.43**	**21.92**	**30.47**	**34.25**

tagging on metrics P@1, P@5 and P@10 on ImageNet-VidVRD. Similar performance improvement trend is obtained on VidVRD as well. We believe these performance differences are mainly due to the fact that 3DRN directly generates object tracklets from the I3D backbone [3], therefore it abandons the knowledge to localize an object from the off-the-shell models like Faster-RCNN [27]. Instead, VRDFormer learns from such knowledge by pre-training it on object detection data. Additionally, VRDFormer leverages auxiliary contexts to help localization, while 3DRN localizes each tracklet independently.

4.3. Ablation Study

We carry out extensive ablation studies on ImageNet-VidVRD to evaluate different components in our model.

1) Recurrent queries. Recurrent queries enable our model to conveniently employ temporal contexts to associate object pairs in different frames. We compare our model with a variant without recurrent queries in Table 3. Row 1 only utilizes static queries to detect object pairs for each single frame, and we employ a greedy approach [48, 52] to associate predicted object pairs across frames in post-processing. We see that recurrent queries largely improve the performance to generate tracklet pairs (row 1 vs. 2).

2) Re-activate strategy for recurrent queries. We adopt a re-activate strategy to alleviate the influence of missing tracking results in a few frames. In Table 3, we further evaluate the effectiveness of the re-activate strategy as shown in row 2 and row 3. The results demonstrate that such strategy is beneficial for tracking.

3) Joint training of object pair detection and relation classification tasks. Previous works [20, 32] utilize independent modules for VidVRD. Nevertheless, relation classification can benefit the detection of interactive object pairs with relations, and vice versa. Table 4 compares models with and without such joint training. The model in row 1, without relation confidence score to select positive object pairs in frames, achieves much worse performance in both tracklet pair detection and relation detection.

Table 4. Ablations of joint training of object detection and relation classification.

	joint train	Relation Detection		Tracklet Pair Detection	
		mAP	R@50	R@50	R@100
1	×	29.85	19.36	28.11	31.85
2	✓	32.43	21.92	30.47	34.25

Table 5. Ablations of different number of queries, where N_q denotes the number of static queries. Row 1 shows the results using the VidVRD baseline [32].

	N_q	Relation Detection		Tracklet Pair Detection	
		mAP	R@50	R@50	R@100
1	-	5.54	6.37	10.87	12.18
2	20	21.67	13.48	20.58	22.50
3	50	30.12	17.05	25.56	27.94
4	100	32.43	21.92	31.35	35.14
5	200	28.62	20.50	30.47	34.25
6	300	24.68	19.44	28.75	33.65

Table 6. Ablations of different strategies to aggregate temporal contexts of tracklets pairs in the video.

	Aggregation	Relation Detection		Relation Tagging	
		mAP	R@50	P@1	P@5
1	Mean	31.17	20.39	71.13	55.82
2	LSTM	31.85	21.45	72.53	56.77
3	Self Att	32.43	21.92	73.11	57.19

4) Number of queries. Table 5 compares the influence of different numbers of static queries. A small number of static queries such as 20 in row 2 drastically decrease the performance especially for recalls, because it limits the generation of relation tracklets and may miss some positive relation pairs in groundtruth. On the contrary, a too large number of static queries such as 200 or 300 is harmful to the performance especially for precision (mAP) due to overly sampling the proposals. It decreases the performance of mAP with -3.81% when $N_q = 200$ and -7.75% when $N_q = 300$. The recalls also decrease a bit because when we increase N_q, it will be more challenging to capture positive proposals from the top ranks. Comparing different scales of the query set, we set N_q=100 in this work.

5) Temporal aggregation strategy in relation classification. In Table 6, we compare different strategies to aggregate the temporal contexts of tracklet pairs for relation classification. Using the mean pooling, VRDFormer has achieved encouraging performance. In addition, using a LSTM for temporal aggregation brings slight improvement. The best performance is achieved by adopting the transformer with self-attention. Comparing row 1 and 3, VRDFormer obtains +1.26% and +1.53% improvement on mAP and R@50 for the relation detection task. We think the limited improvement may relate to the limitation of datasets, as many relations such as “kiss” or “behind” in both datasets do not show complicated temporal variations. Therefore, a naive aggregation strategy could be good enough for these relation types.

Table 7. Ablations of different length for temporal aggregation.

	T length	Relation Detection		Relation Tagging	
		mAP	R@50	P@1	P@5
1	1	30.57	19.92	70.55	55.37
2	4	30.82	20.14	70.84	55.68
3	8	31.25	20.48	71.25	56.03
4	32	32.43	21.92	73.11	57.19

Table 8. Ablations of transformer components, where “Cross” and “Self” denote the cross- and self-attention in transformer decoder.

	Decoder		Relation Detection		Relation Tagging	
	Cross	Self	mAP	R@50	P@1	P@5
1	×	✓	28.38	18.84	69.60	53.49
2	✓	×	26.03	16.38	67.46	51.68
3	✓	✓	32.43	21.92	73.11	57.19

6) Length for temporal aggregation. In Table 7, we further explore a suitable length T to reserve the embedding memories of tracklet pairs for long-term relation prediction. When T equals to 1, our model directly uses the current recurrent query to predict the relation, which only contains short-term memory for relation classification. To our surprise, performance in this case is comparable with performance using T=32, showing both efficiency and effectiveness of the recurrent queries to model the short-term temporal contexts. While the long-term memory of temporal contexts is still beneficial comparing row 1 and 4.

7) Transformer architectures. Table 8 ablates different attentions in our transformer decoder for object pair generation. Row 1 utilizes the decoder without cross-attention. In such case, the query is initialized same as the two-stage version of [53] to obtain image cues. Adding the cross-attention (Row 3) brings stable significant improvement on both tasks, which proves that the external information beyond the cropped bounding box regions indeed help enhance the queries. Row 2 uses the transformer decoder without self-attention, which is much worse than results in row 3. It shows that the contextual relationships between different relation tracklets are beneficial to improve the localization and relation prediction. Combining cross- and self-attention in the decoder, we achieve the best performance on relation tagging and detection in row 3.

4.4. Quality Analysis

Some visualization examples are illustrated in Figure 3 for comparison between our model and the VidVRD baseline. In Figure 3(a), VRDFormer correctly detects the “adult-next_to-guitar relation instead of “adult-play(instr)-guitar” by considering the useful context of the adult. Besides, in Figure 3(b), our model is able to successfully localize the occluded relation pair “adult-watch-laptop” which is however missed by VidVRD baseline. Befitting from relation instances generation through aggregating contexts, our model is able to localize and predict some relations that cannot be detected in isolation.

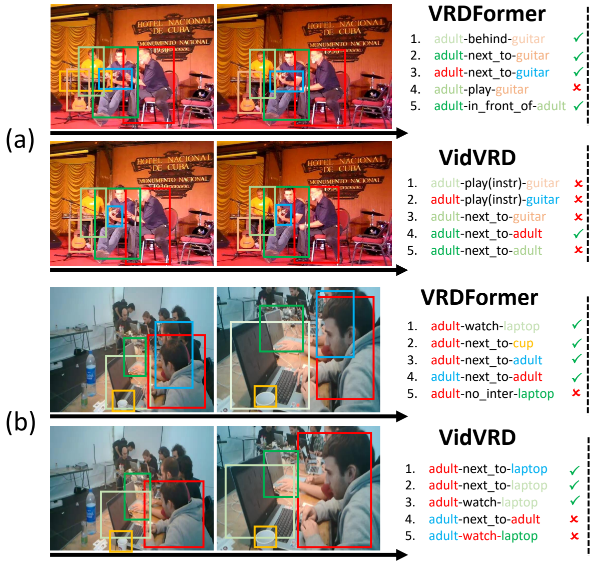

Figure 3. Visualization of two pairs of comparison in relation detection between VidVRD (**Bottom**) and our model VRDFormer (**Top**). The ✓ and × represent correct and false detection respectively. The figure is best viewed in color.

5. Conclusion

In this paper, we propose VRDFromer for video visual relation detection. VRDFormer is a transformer-based model to unify previous multi-stage components for VidVRD in an end-to-end manner. By first introducing the contextually enriched queries into VidVRD, our model can localize and predict a relation instance more precisely. In addition, modeling spatio-temporal contexts also provides auxiliary information to infer the negative relation status of queries to avoid exhaustive sampling. The experiments demonstrate that VRDFormer significantly surpasses state-of-the-art methods on two benchmark datasets. However, due to imbalanced class distribution in VidVRD datasets, there is a large gap for different relation classes. In the future, we will address the long-tail issue in our VRDFormer.

6. Acknowledgments

This work was partially supported by National Natural Science Foundation of China (No. 62072462), National Key R&D Program of China (No. 2020AAA0108600), and Large-Scale Pre-Training Program 468 of Beijing Academy of Artificial Intelligence (BAAI).

References

[1] Qianwen Cao, Heyan Huang, Xindi Shang, Boran Wang, and Tat-Seng Chua. 3-d relation network for visual relation recognition in videos. *Neurocomputing*, 432:91–100, 2021. 2, 6

[2] 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, 3, 5

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 2, 7

[4] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021. 2

[5] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10337–10346, 2020. 1

[6] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE conference on computer vision and Pattern recognition*, pages 3076–3086, 2017. 2

[7] Donglin Di, Xindi Shang, Weinan Zhang, Xun Yang, and Tat-Seng Chua. Multiple hypothesis video relation detection. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 287–291. IEEE, 2019. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[9] 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. 5

[10] 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. 6

[11] Hao Jiang, Sidney Fels, and James J Little. A linear programming approach for multiple object tracking. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1

[12] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 727–735, 2017. 1

[13] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021. 2

[14] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 74–83, 2021. 2

[15] 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. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 6

[16] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'ou Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1347–1356, 2017. 2

[17] Wentong Liao, Bodo Rosenhahn, Ling Shuai, and Michael Ying Yang. Natural language guided visual relationship detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[18] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 482–490, 2020. 2

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[20] Chenchen Liu, Yang Jin, Kehan Xu, Gongqiang Gong, and Yadong Mu. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10840–10849, 2020. 2, 6, 7

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[22] 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

[23] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 2, 3, 4

[24] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 3

[25] Min Peng, Chongyang Wang, Yuan Gao, Yu Shi, and Xiang-Dong Zhou. Temporal pyramid transformer with multimodal interaction for video question answering. *arXiv preprint arXiv:2109.04735*, 2021. 1

[26] Xufeng Qian, Yueting Zhuang, Yimeng Li, Shaoning Xiao, Shiliang Pu, and Jun Xiao. Video relation detection

with spatio-temporal graph. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 84–93, 2019. 2, 6

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 2, 7

[28] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 5

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5, 6

[30] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 279–287, 2019. 1, 5

[31] Xindi Shang, Yicong Li, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Video visual relation detection via iterative inference. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3654–3663, 2021. 6

[32] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1300–1308, 2017. 1, 2, 5, 6, 7

[33] Zixuan Su, Xindi Shang, Jingjing Chen, Yu-Gang Jiang, Zhiyong Qiu, and Tat-Seng Chua. Video relation detection via multiple hypothesis association. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3127–3135, 2020. 2

[34] Xu Sun, Tongwei Ren, Yuan Zi, and Gangshan Wu. Video visual relation detection via multi-modal feature fusion. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2657–2661, 2019. 2, 6

[35] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10410–10419, 2021. 2, 3, 5

[36] Yao-Hung Hubert Tsai, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10424–10433, 2019. 1, 2, 6

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 3

[38] Bo Wan, Desen Zhou, Yongfei Liu, Rongjie Li, and Xuming He. Pose-aware multi-level feature network for human object interaction detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9469–9478, 2019. 2

[39] Tiancai Wang, Rao Muhammad Anwer, Muhammad Haris Khan, Fahad Shahbaz Khan, Yanwei Pang, Ling Shao, and Jorma Laaksonen. Deep contextual attention for human-object interaction detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5694–5702, 2019. 1, 2

[40] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6847–6857, 2021. 2

[41] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 2

[42] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017. 2

[43] 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. 1

[44] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018. 1, 2

[45] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 322–338, 2018. 2

[46] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5678–5686, 2017. 2

[47] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1

[48] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 5, 7

[49] Sipeng Zheng, Shizhe Chen, and Qin Jin. Visual relation detection with multi-level attention. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 121–129, 2019. 2

[50] Sipeng Zheng, Shizhe Chen, and Qin Jin. Skeleton-based interactive graph network for human object interaction

detection. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020. 2

[51] Sipeng Zheng, Xiangyu Chen, Shizhe Chen, and Qin Jin. Relation understanding in videos. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2662–2666, 2019. 6

[52] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020. 7

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 6, 8

[54] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 1

[55] Yaohui Zhu and Shuqiang Jiang. Deep structured learning for visual relationship detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2