arXiv:2303.13209v1 [cs.CV] 23 Mar 2023

Taking A Closer Look at Visual Relation : Unbiased Video Scene Graph Generation with Decoupled Label Learning

Wenqing Wang[1] Yawei Luo[1]* Zhiqing Chen[1] Tao Jiang[1] Lei Chen[2] Yi Yang[1] Jun Xiao[1]
[1]Zhejiang University [2]FinVolution Group

Abstract

*Current video-based scene graph generation (VidSGG) methods have been found to perform poorly on predicting predicates that are less represented due to the inherent biased distribution in the training data. In this paper, we take a closer look at the predicates and identify that most visual relations (*e.g. `sit_above`*) involve both actional pattern (*`sit`*) and spatial pattern (*`above`*), while the distribution bias is much less severe at the pattern level. Based on this insight, we propose a decoupled label learning (DLL) paradigm to address the intractable visual relation prediction from the pattern-level perspective. Specifically, DLL decouples the predicate labels and adopts separate classifiers to learn actional and spatial patterns respectively. The patterns are then combined and mapped back to the predicate. Moreover, we propose a knowledge-level label decoupling method to transfer non-target knowledge from head predicates to tail predicates within the same pattern to calibrate the distribution of tail classes. We validate the effectiveness of DLL on the commonly used VidSGG benchmark,* i.e. *VidVRD. Extensive experiments demonstrate that the DLL offers a remarkably simple but highly effective solution to the long-tailed problem, achieving the state-of-the-art VidSGG performance.*

1. Introduction

Video-based scene graph generation (VidSGG) aims to represent video content as dynamic graphs constructed by ⟨`subject, predicate, object`⟩ triplets. It offers high-level understanding and summarization of video knowledge, which can benefit downstream tasks such as visual question answering [1, 37, 43], video captioning [44], and video retrieval [33, 8, 40]. Compared to its image-based counterpart ImgSGG [48, 24], VidSGG is considered a more challenging task, as the pairwise relations between visual entities are dynamic along the temporal dimension, making VidSGG a typical multi-label problem. Despite the vast body of literature on ImgSGG [48, 24], these characteristics prevent ImgSGG methods from being trivially applied to VidSGG. VidSGG remains a relatively under-explored problem with several unsolved issues at present.

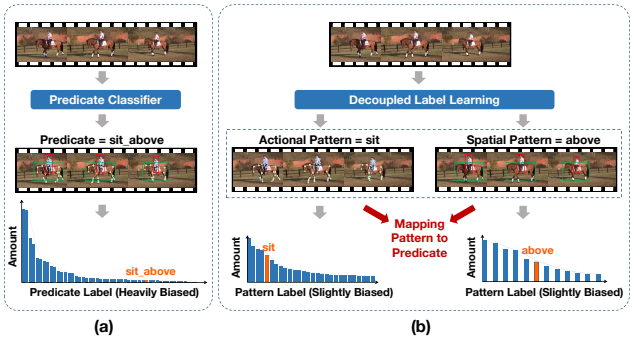


Figure 1. (a) Previous canonical visual relation prediction pipeline. (b) A new DLL paradigm, which transforms the vanilla relation prediction task into a pattern-level problem.

Several recent attempts have been made to solve VidSGG by exploiting the spatio-temporal information of the video [28, 22, 38, 6]. While these attempts have achieved some progress in pursuing overall performance by extracting short- and long-term information, they have ignored the inherent long-tailed nature of the data, leading to severely biased predicate predictions in the final results. For example, taking the commonly used VidVRD [32] dataset as an illustration, the head predicate categories nearly dominate the ground-truth annotations. Furthermore, even among all correctly detected predicates, a small number of head categories account for most of the overall performance. Additionally, the missing labels of VidSGG are inevitable during data annotation due to the fleeting temporal interaction or inconspicuous spatial relation of the objects. Compared to the head predicates, tail samples are more likely to be ignored by the annotators, further deteriorating the predicate bias in a video.

More recently, a few methods have noticed this phenomenon and endeavored to debias the visual relation. Li *et al.*[20] proposed a causality-inspired interaction to weaken the false correlation between input data and predicate la-

*Corresponding author

bels. Xu *et al.* [45] considered temporal, spatial, and object biases in a meta-learning paradigm. These implicit approaches mitigate the long-tail problem to some extent, but the performance of tail classes is still unsatisfactory.

In this paper, we propose a more explicit method to tackle the biased VidSGG problem from the decoupled learning perspective. By taking a closer look at the visual relations in the video, we identify that most visual relations (*e.g.* `sit_above`) involve both actional pattern (`sit`) and spatial pattern (`above`). Surprisingly, compared to the original predicate-level label, the distribution bias is much less severe at the pattern level. Based on this insight, we propose a decoupled label learning (DLL) paradigm to transform the intractable visual relation detection into a pattern-wise prediction problem. Specifically, DLL employs an adversarial learning strategy to decouple the original video features into actional and spatial ones and then forwards them to the actional and spatial classifiers separately, in order to learn the decoupled patterns. The output patterns are then aggregated and mapped back to predicates.

To further boost the performance on tail visual relations, we propose a knowledge-level label decoupling method to calibrate the distribution of the preliminary predicates from pattern aggregation. We assume that head predicates are well represented with sufficient training data and the predicates within the same pattern ought to contain similar non-target knowledge (*i.e.*, correlation with other predicates). Motivated by this thought, we propose to decouple label knowledge into target and non-target knowledge and calibrate the non-target knowledge of tail classes with the help of head classes that are within the same pattern. To this end, we build a learnable predicate correlation matrix and align the tail prediction distribution to that of head classes. Our knowledge-level label decoupling method imposes tail classes to learn more from non-target knowledge of the head classes that are more abundant and calibrated.

In summary, this paper makes following contributions:

- We propose the Decoupled Label Learning (DLL) method for VidSGG, which is the first attempt to address the intractable biased predicate prediction from a pattern-level perspective.
- We present a knowledge-level label decoupling approach to further boost the performance on tail visual relations, which calibrates the tail distribution using the non-target knowledge of head predicates.
- We conduct extensive experiments on a widely used VidSGG dataset: VidVRD [32]. The results demonstrate that DLL achieves state-of-the-art performance on various metrics across different scenarios, especially on those tail predicates.

2. Related Work

2.1. Image-based Scene Graph Generation

ImgSGG has been an intensively studied task. Existing works can be roughly divided into two groups: 1) **Two-stage methods** [24, 48, 36, 5]: conducting object detection and relationship prediction within two stages. 2) **One-stage methods** [23, 26, 21]: dealing with the object and relationship simultaneously in one go. Because of the long-tail problem, the performances of the traditional methods are far from satisfactory. Zeller *et al.* [48] first pointed out the imbalance of predicates in ImgSGG dataset. Chen *et al.* [5] and Tang *et al.* [36] also noticed this problem and proposed a new metric to measure the average performance *i.e.*, mean recall@K. Along this vein, various methods [14] have been proposed to solve biased relationship prediction, including depolarization strategies such as resampling [19], reweighting [46], unbiased representation from bias [35], *etc*.

2.2. Video-based Scene Graph Generation

Generally, video-based and image-based SGG share a similar goal of detecting the visual objects and relationships in given data. Nevertheless, employing existing ImgSGG approaches straightly to parse a video is non-trivial due to the multi-label and dynamic nature of VidSGG. The existing VidSGG work can be roughly divided into two groups according to the format of dataset annotation:

Frame-based VidSGG: Ji *et al.* [16] proposed the first large-scale frame-level data set named **Action Genome (AG)**. Following this seminal work, Cong *et al.* [6] proposed a new network structure called "space-time converter (STTran)", which consists of a space encoder and a time decoder. Chen *et al.* [4] proposed a novel method for weakly-supervised task with only single-frame weak supervision and to generate pseudo labels for unannotated frames. Both of the methods are designed for better capturing the spatio-temporal context information for relationship recognition.

Tracklet-based VidSGG: Shang *et al.* proposed a video-level dataset called **VidVRD** [32], which aims to detect all visual relationship instances in the video in the form of relational triplets $\langle$`subject, predicate, object`$\rangle$ and object tracks. In VidVRD [32], the authors also proposed a widely used three-stage segment-based detection framework. However, this method is not ideal for long video detection. To remedy this, Feng *et al.* [10] proposed a detection trajectory recognition paradigm by constructing consistent long-term object trajectories from videos, and then using transducers to capture the dynamic and visual relationships of objects. Besides, Liu *et al.* [22] also proposed a new sliding window scheme called "VRD-STGC" to simultaneously predict short- and long-term relations. Instead of preview segment-based and window-based methods, Woo *et al.* [41] presented a time span-suggested network "TSPN"

to determine what and when to look in task. More recently, Shang *et al.* [31] proposed "VidVRD-II", which achieves iterative relational reasoning and joint relation classification. Gao *et al.* [12] first proposed a compositional and motion-based relation prompt learning framework (RePro) in open-vocabulary VidVRD setting. Albeit with these prior arts, only a few work has realized the long-tail predicate distribution as the bottleneck issue for VidSGG task [20, 45].

Our work belongs to this tracklet-based vein. We adopt VidVRD-II [31] and VRD-STGC [22] as our baselines. Different from above methods, we target at long-tail predicate distribution problem and make pioneering attempt to address it from the perspective of decoupled label learning.

2.3. Disentangled Representation Learning

Disentangled Representation Learning (DRL) is an unsupervised learning technique aimed at identifying and separating the underlying factors of variation in observable data into representation form. The process of disentangling these factors into semantically meaningful variables aids in learning explainable data representations, imitating the meaningful reasoning process of humans. In recent years, numerous disentangling methods based on antagonism to decouple features have emerged in different fields, such as DADA, proposed by Peng *et al.* [27], which simultaneously disentangles domain-invariant, domain-specific, and class-irrelevant features. The disentanglement process is carried out in an adversarial manner, with the disentangler generating features that deceive the class identifier trained on the labeled source domain. In the field of representation learning, disentangled representation learning has been heavily studied [7, 9, 42], which involves assigning different factors of variation to distinct dimensions of representation vectors. Jozsef Nemeth *et al.* [25] also proposed an adversarial decoupling method based on group observation to separate content and style-related attributes. Yang *et al.* [47] used a gradient reversal layer (GRL) [11] based adversarial classifier to eliminate speaker information in latent space for voice conversion tasks, extracting features related to speaker identity using a common classifier for timbre. In our work, we adopt the adversarial paradigm to decouple video features into actional and spatial components, inspired by these prior works.

3. Methodology

In this section, we formally introduce **DLL**, the method that decouples labels in both pattern level and knowledge level with the goal of debiasing the scene graph generation. Firstly, we present the preliminaries in Sec.3.1. Secondly, we detail Pattern Decoupling Learning (PDL) in Sec.3.2, which decouples predicate labels into patterns, transforming the original predicate prediction into a less biased pattern-level classification problem. Thirdly, we present

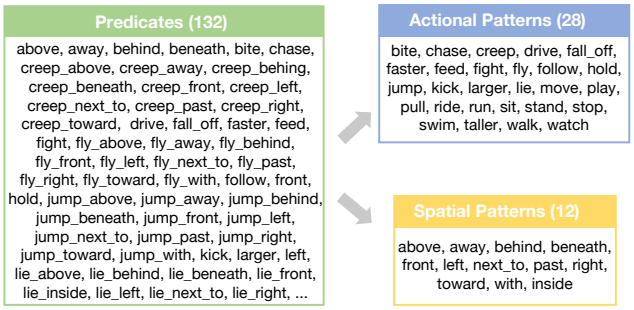


Figure 2. Illustration of predicate labels and actional & spatial patterns after label decoupling.

Knowledge Decoupling Learning (KDL) in Sec.3.3, which further boosts the tail performance by decoupling predicate knowledge into target and non-target knowledge and then calibrating the non-target knowledge of tail predicates using head classes within the same pattern. Finally, we describe the overall training objective of DLL in Sec. 3.4.

3.1. Preliminary

Problem Formulation. Given an entity category set $\mathcal{C}_e$ and predicate category set $\mathcal{C}_p$, a video scene graph can be represented as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. Each node in $\mathcal{N}$ is characterized by an entity category $c_i^e \in \mathcal{C}_e$ and a bounding box (bbox) sequence (tracklet). Each edge in $\mathcal{E}$ is characterized by the linkage from the i-th node (subject) to the j-th node (object), and a collection of multiple predicate categories $\mathcal{P}_{ij} = \{c_k^p \in \mathcal{C}_p | k = 1, \ldots, n_{ij}\}$ where n_{ij} is the total number of predicates between i-th and j-th node.

Actional and Spatial Pattern Sets. We depict the actional and spatial pattern sets decoupled from predicate labels in Fig. 2. These exemplar predicate labels are from VidVRD [32] dataset.

Pipeline Overview. Given a video segment, we first detect bbox in each frame using FasterRCNN [29] and apply Seq-NMS [13] to generate tracklet (*i.e.*, bbox sequence). For relation classification, we combine the RoI Aligned visual feature of tracklet regions and the relative position feature of subject-object pairs as the video feature f_v of this segment. The video feature f_v is then forwarded to the pattern decoupling learning (PDL) module to predict its actional and spatial patterns. After that, the actional and spatial patterns are integrated and mapped back to predicate space. Finally, the knowledge decoupling learning (KDL) module is employed to further calibrate the tail predicate distribution. The detailed architecture of our DLL (PDL + KDL) is depicted in Fig. 3.

3.2. Pattern-level Label Decoupling

The proposed PDL module takes the feature vector f_v as input. As a video feature, f_v consists of intertwined actional

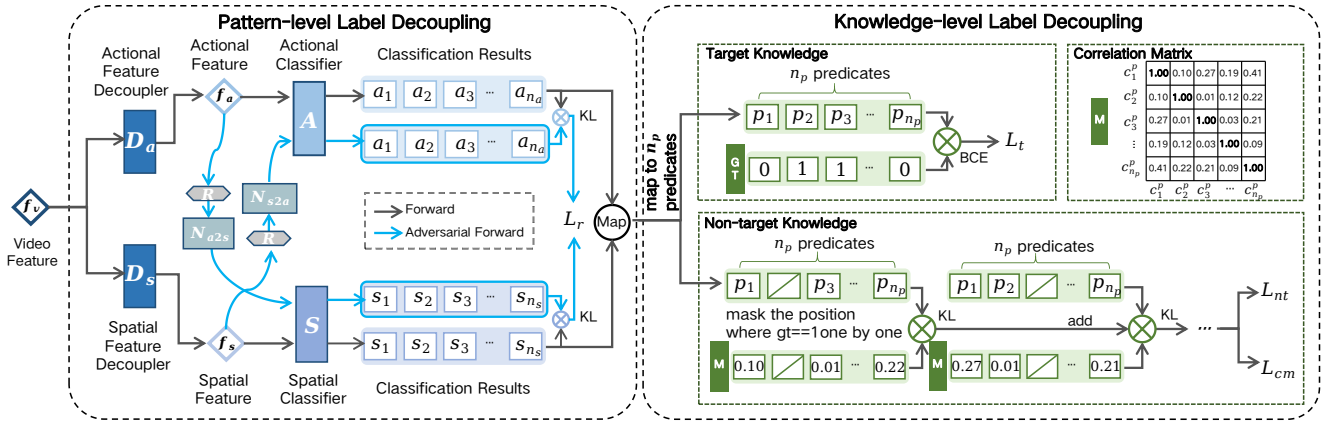


Figure 3. **The overall framework of DLL.** DLL is fed with the video feature and conducts Pattern Decoupling Learning and Knowledge Decoupling Learning on the feature in a sequential manner.

and spatial information. To decouple the highly entangled f_v to pattern space, we introduce two decouplers D_a, D_s to separate f_v into actional feature f_a and spatial feature f_s using adversarial disentangle training.

Adversarial Feature Disentanglement. As analyzed above, the aim of decouplers D_a, D_s is to decouple feature f_v into actional feature f_a and spatial feature f_s, which are then utilized to train two separate pattern classifier A and S. The specific output is as follows:

$$\begin{aligned} f_a &= D_a(f_v) \qquad f_s = D_s(f_v) \\ \boldsymbol{p_a} &= A(f_a) \qquad \boldsymbol{p_s} = S(f_s), \end{aligned} \tag{1}$$

where $\boldsymbol{p_a}$ and $\boldsymbol{p_s}$ is the output of A and S, respectively.

However, to decouple the highly entangled feature f_v to patterns is non-trivial, especially for those less supervised predicates. To tackle this challenge, we propose an adversarial learning method to achieve better disentanglement in a self-supervised manner: we design two extra actional-spatial and spatial-actional feature extraction network N_{a2s}, N_{s2a} to learn opposite pattern features f_{a2s}, f_{s2a} from f_s, f_a. Generally, if f_a and f_s are well disentangled, f_{a2s} and f_{s2a} should not contain any discriminative information about the opposite patterns. In this way, the specific output is as follows:

$$\begin{aligned} f_{a2s} &= N_{a2s}(f_a) \qquad f_{s2a} = N_{s2a}(f_s) \\ \boldsymbol{p_{s2a}} &= A(f_{s2a}) \qquad \boldsymbol{p_{a2s}} = S(f_{a2s}), \end{aligned} \tag{2}$$

where $\boldsymbol{p_{s2a}}$ and $\boldsymbol{p_{a2s}}$ is the output of A and S, respectively.

We train the disentanglers D_a, D_s by minimizing the opposite pattern information in f_{a2s}, f_{s2a}. Accordingly, the network parameter $\theta_{N_*}, \theta_{D_*}$ will play a two-player mini-max game with the following objective function $\mathcal{L}_r(\theta_{N_*}, \theta_{D_*})$:

$$\begin{aligned} &\min_{\theta_{N_{a2s}}} \max_{\theta_{D_a}} \mathcal{L}_r(\theta_{N_{a2s}}, \theta_{D_a}), \\ &\min_{\theta_{N_{s2a}}} \max_{\theta_{D_s}} \mathcal{L}_r(\theta_{N_{s2a}}, \theta_{D_s}). \end{aligned} \tag{3}$$

We use KL loss with a non-parametric Gradient Reversal Layer (GRL) [11] as R to achieve Eq. 3:

$$\begin{aligned} \mathcal{L}_{PLD} &= \mathcal{L}_r \\ &= -\lambda(D_{KL}(\boldsymbol{p_{a2s}}||\boldsymbol{p_s}) + D_{KL}(\boldsymbol{p_{s2a}}||\boldsymbol{p_a})), \end{aligned} \tag{4}$$

where λ is a hyper-parameter, $\boldsymbol{p_{a2s}}$ & $\boldsymbol{p_{s2a}}$ are the output of S & A using feature f_{a2s} & f_{s2a}. $\boldsymbol{p_a}$ & $\boldsymbol{p_s}$ are the gradient-blocked outputs of A & S using feature f_a and f_s, respectively. Following the training rule of GRL, R will multiply the incoming gradient by a negative value, so that the training objectives of the network before and after R are opposite. R is allocated between the feature extractor f_* and the opposite pattern learning network module N_*, as shown in Fig. 3. The actional-spatial and spatial-actional learning network parameters θ_{N_*} and disentangler parameter θ_{D_*} are updated as follows:

$$\begin{aligned} \theta_{N_*} &\leftarrow \theta_{N_*} - \mu \frac{\partial \mathcal{L}_r}{\partial \theta_{N_*}}, \\ \theta_{D_*} &\leftarrow \theta_{D_*} + \mu \frac{\partial \mathcal{L}_r}{\partial \theta_{D_*}}, \end{aligned} \tag{5}$$

where μ is the learning rate, $\theta_{N_*} \in \{\theta_{N_{a2s}}, \theta_{N_{s2a}}\}$ and $\theta_{D_*} \in \{\theta_{D_a}, \theta_{D_s}\}$.

Mapping Function. In this part, we detail the mechanism to map actional pattern and spatial pattern back to the predicate prediction. Specifically, we couple the actional logits and spatial logits by $Map(.,.)$:

$$\boldsymbol{p} = Map(\boldsymbol{p_a}, \boldsymbol{p_s}), \tag{6}$$

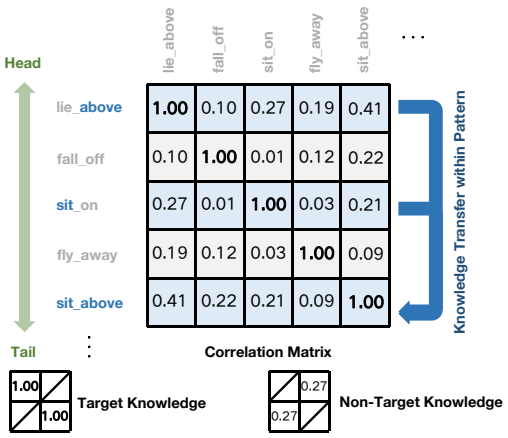


Figure 4. Illustration of **KDL**, which transfers the non-target knowledge of head classes to the tail ones that are within the same actional or spatial pattern, in order to calibrate the biased distribution of tail predicates.

where $\boldsymbol{p}$ denotes the probability of final coupling labels and $Map(.,.)$ denotes the mapping function from actional & spatial space to predicate space. Since the predicate-pattern correspondences are fixed and known, $Map(.,.)$ can be regarded as an inverse process of Fig. 2. For dual-pattern predicates such as `fly_away`, $Map(.,.)$ calculates its probability by average the corresponding actional and spatial pattern logits. While for those single-pattern predicates such as `bite` and `inside`, which only belong to either actional pattern or spatial pattern, $Map(.,.)$ would directly assign the output logits of classifiers A or S to it.

Mutual Calibration. In certain scenarios, classifiers A and S may generate high-probability pattern combinations that do not correspond to any predicate labels in the original label space, *e.g.*, `bite` + `away`. To enable these classifiers to better align with the data distribution of the final predicate labels, we propose an iterative mechanism for mutual calibration (MC) of pattern-level and predicate-level distributions as follows:

$$(\boldsymbol{p}'_{\boldsymbol{a}}, \boldsymbol{p}'_{\boldsymbol{s}}) = Map^{-1}(Map(\boldsymbol{p}_{\boldsymbol{a}}, \boldsymbol{p}_{\boldsymbol{s}})), \tag{7}$$

$$\begin{aligned} \boldsymbol{p}_{\boldsymbol{a}} &\leftarrow \eta \boldsymbol{p}_{\boldsymbol{a}} + (1-\eta)\boldsymbol{p}'_{\boldsymbol{a}} \\ \boldsymbol{p}_{\boldsymbol{s}} &\leftarrow \eta \boldsymbol{p}_{\boldsymbol{s}} + (1-\eta)\boldsymbol{p}'_{\boldsymbol{s}}. \end{aligned} \tag{8}$$

MC enhances the adaptability of pattern classifiers by iteratively updating the distribution of patterns and predicates. By refining the classification outputs at both levels, our proposed mechanism helps the patterns generated by the classifiers to align with the desired predicate labels.

3.3. Knowledge-level Label Decoupling

After acquiring the preliminary predicates, our next objective is to refine the distribution of the tail predicates with the assistance of the well-represented head ones. To achieve this, we divide the predicate label knowledge into two parts, namely target and non-target, and transfer the non-target knowledge from the head predicates to the tail ones that are within the same pattern, as shown in Fig. 4.

Target knowledge. Target knowledge refers to the categorical information of a sample that represents the ground truth and is specific to each class. In the context of multi-label learning task such as VidSGG, we employ the binary cross-entropy loss function to supervise the target knowledge.

$$\mathcal{L}_t = BCE(\boldsymbol{p}, \boldsymbol{q}), \tag{9}$$

where $\boldsymbol{p}$ denotes the prediction probabilities and $\boldsymbol{q}$ denotes the multi-hot encoded target labels.

Non-target knowledge. Non-target knowledge, also known as "dark knowledge" [15], contains valuable information that represents the intrinsic correlation between classes. To capture this correlation, we utilize a correlation matrix denoted as $\boldsymbol{M} \in \mathbb{R}^{n_p \times n_p}$, where n_p is the number of predicate labels. In our approach, we initialize the correlation matrix $\boldsymbol{M}$ as an identity matrix $\boldsymbol{I}$ and update it through mutual learning between the backbone model and $\boldsymbol{M}$.

Correlation Matrix Updating. We learn non-target knowledge from the model prediction probabilities to update the correlation matrix. For a sample of k-th predicate class with the probability vector $\boldsymbol{p}$, we employ KL loss between k-th position masked correlation vector $\boldsymbol{m}^k_{non}$ and gradient-blocked prediction probabilities $\boldsymbol{p}_{non}$ to update $\boldsymbol{M}$:

$$\mathcal{L}_{cm} = D_{KL}(\boldsymbol{m}^k_{non} || \boldsymbol{p}_{non}). \tag{10}$$

Algorithm 1: Decoupled Label Learning.

Input: disentangler D_a, D_s, actional-spatial and spatial-actional feature extraction network N_{a2s}, N_{s2a}, actional classifier A, spatial classifier S, correlation matrix $\boldsymbol{M}$.

initialization: $\boldsymbol{M} \leftarrow \boldsymbol{I}, \alpha = 0.1, \gamma = 1.0$

for *1:epoch* **do**
  for *1:iteration* **do**
    Pattern-level Label Decoupling:
    for *1:step* **do**
      Update $\boldsymbol{p_a}, \boldsymbol{p_s}$ using Eq. 7, 8;
    end
    Update D_a, D_s, A, S using Eq. 9;
    Update $D_a, D_s, N_{a2s}, N_{s2a}$ using Eq. 4 with R;
    Knowledge-level Label Decoupling:
    Update $\boldsymbol{M}$ using Eq. 10;
    Update D_a, D_s, A, S using Eq. 11;
    Update α;
  end
  Update γ;
end

Head-to-Tail Knowledge Transfer. We assume that the non-target knowledge of head predicates in $\boldsymbol{M}$ is well represented. To leverage the head knowledge, for a preliminary predicate, we search for the non-target knowledge of *header* predicates in $\boldsymbol{M}$ that are within the same pattern, as presented in Fig. 4. The pinpointed knowledge is denoted as $\boldsymbol{m_h}$, where h denotes the index of header predicate. Then we mask the k-th ground-truth position of $\boldsymbol{p}$ and $\boldsymbol{m_h}$ to learn the non-target knowledge with KL loss as follows:

$$\mathcal{L}_{nt} = D_{KL}(\boldsymbol{p}_{non} || \boldsymbol{m}^h_{non}), \tag{11}$$

where k represents the existing target class index, $\boldsymbol{p}_{non}$ is prediction probabilities with k-th position masked, $\boldsymbol{m}^h_{non}$ is the masked gradient-blocked correlation vector $\boldsymbol{m_h}$, which denotes the h-th row in $\boldsymbol{M}$. Since VidSGG is a multi-label task, such knowledge transfer would be continually proceeded until all the ground-truth positions are traversed.

The "Correlation Updating" and "Knowledge Transfer" form a mutual learning mechanism, enabling the VidSGG model and the correlation matrix $\boldsymbol{M}$ to learn from each other. Considering that the model learning is more reliable under the constraints of BCE loss in the early stage, we assign loss $\mathcal{L}_{nt}$ with a growing learning rate α, which is initialized to a small value. The growth of α depends on the training iterations i with hyper-parameter β. Accordingly, the overall loss of KDL is

$$\begin{aligned} \alpha &\leftarrow \alpha + \beta i, \\ \mathcal{L}_{KDL} &= \mathcal{L}_t + \mathcal{L}_{cm} + \alpha \mathcal{L}_{nt}. \end{aligned} \tag{12}$$

3.4. Training Objective

With the above loss terms, the overall loss function of DLL can be written as:

$$\mathcal{L} = \mathcal{L}_{PDL} + \gamma \mathcal{L}_{KDL}, \tag{13}$$

where γ is employed to control the relative scale of the two loss terms.

4. Experiments

4.1. Datasets

We adopt the video relation benchmark ImageNet-VidVRD [32] in our experiments: **VidVRD [32]** is the first dataset for benchmarking VidSGG and has been widely used in previous work [22, 38, 3, 41]. It contains $1,000$ videos (800 for training and 200 for evaluation).

4.2. Evaluation Metrics

Metrics. We use the official evaluation metrics [32, 31] of the VRU Challenge, including Relation Detection (RelDet) and Relation Tagging (RelTag). Besides the average precision **(mAP)**, Recall@K (**R@K**,K=$50, 100$) for RelDet and Precision@K (**P@K**,K=$5, 10$) for RelTag. Following [5, 36, 35] and [18], we first introduce the mean Recall@K **(mR@K)** and a comprehensive metric **Mean** as key metrics to VidSGG.

mR@K. mR@K is calculated by obtaining recall scores from the top-K triplet predictions in every video segment first and then averaging them *w.r.t* each predicate category. As discussed in [5, 36, 35], mR@K serves as a more canonical metric in the long-tailed learning scenario.

Mean. As discussed in [18], Mean is the average of mR@K and R@K. Since R@K favors head predicates and mR@K favors tail predicates, the Mean metric is better for evaluating the performance across all predicates.

4.3. Implementation Details

Relation Detection Details. VRD-STGC [22]: We employ a Faster-RCNN [29] model in video frames, then track frame-level detection results across the whole video using a Multiple Object Tracking (MOT) algorithm to obtain tracklets. We adopt RoI Aligned detection and I3D feature with relative motion features for each pair. Only pairs overlapping with the ground truth by more than 0.5 in vIoU(volume IoU) are selected. Then we construct a spatial graph and a temporal graph to filter incompatible proposals. **VidVRD-II [31]:** Given a video, we first split it into shot segments of 30 frames with 15 frames overlapped, then detect the bboxes in each frame using Faster-RCNN [29] and apply Seq-NMS [13] to generate tracklets (i.e., bounding box sequence). We adopt the RoI Aligned visual feature of tracklet regions and the relative position feature of subject-object pairs. Finally, we perform the simple greedy relation association as proposed in [32] to associate the detected relation instances across the segments.

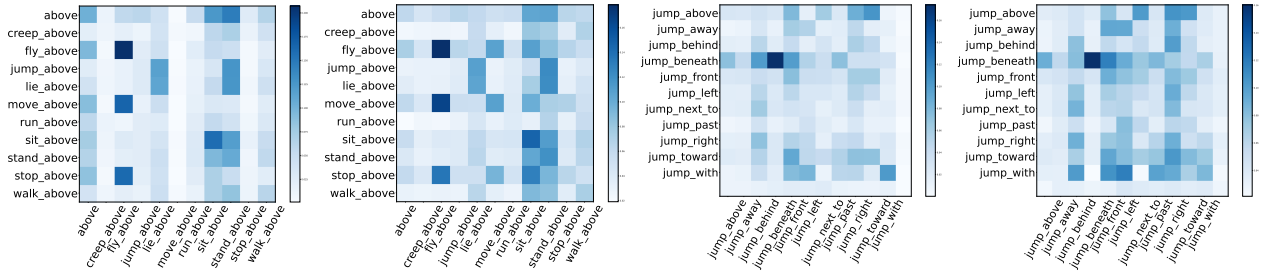
Hyper-paramaters. **VRD-STGC [22]:** We set λ=1e-1, η=1e-1, β=1e-4 and $\gamma = pow(0.99, e)$ where e denotes the epoch. We set α=1e-1 in the first 3 epochs. In order to better couple labels, we fine-tune the ratio of actional labels and spatial labels. Other hyper-parameters are set consistently with VRD-STGC [31]. Our model is trained for total of 20 epochs with the learning rate μ=1e-1 by using SGD [30] optimizer. **VidVRD-II [31]:** We set λ=0.13, η=1e-3, β=1e-4 and $\gamma = pow(0.99, e)$. We set α=1e-1 in the first 10 epochs. Other hyper-parameters are set consistently with VidVRD-II [31]. We train DLL for total of 100 epochs with the learning rate μ=1e-3 by using Adam [17] optimizer.

4.4. Comparison with State of the Arts

Performance Comparison on VidVRD. In Table. 1, we evaluate our DLL by incorporating it into two typical baseline models in VidSGG: VRD-STGC [22] and VidVRD-II [31]. However, since other state-of-the-art methods' goals are not to solve the long-tail problem, they have not recorded their performance on mR metrics, so we do

Table 1. Performance (%) on VidVRD [32] dataset. **Mean**: The average of mR@50/100 and R@50/100.

Method	Relation Detection						Relation Tagging	
	mR@50	mR@100	R@50	R@100	Mean	mAP	P@5	P@10
VidVRD [32] $_{ACM-MM2017}$	-	-	5.54	6.37	-	8.58	28.90	20.80
GSTEG [39] $_{CVPR2019}$	-	-	7.05	8.67	-	9.52	39.50	28.23
3DRN [2] $_{Neurocomputing2021}$	-	-	5.53	6.39	-	14.68	41.80	29.15
VRD-GCN [28] $_{ACM-MM2019}$	-	-	8.07	9.33	-	16.26	41.00	28.50
MHA [34] $_{ACM-MM2020}$	-	-	9.53	10.38	-	19.03	41.40	29.45
TRACE [38] $_{ICCV2021}$	7.55	9.37	9.08	11.15	9.29	17.57	45.30	33.50
TSPN [41] $_{Arxiv2022}$	-	-	11.56	14.13	-	18.90	43.80	33.73
Social Fabric [3] $_{ICCV2021}$	-	-	13.73	16.88	-	20.08	49.20	38.45
IVRD [20] $_{ACM-MM2021}$	-	-	12.40	14.46	-	22.97	49.87	35.75
VRD-STGC+MSVGG [45] $_{ECCV2022}$	-	-	12.62	15.78	-	20.76	44.90	33.15
VRD-STGC [22] $_{CVPR2020}$	**8.73**	10.21	11.21	13.69	10.96	18.38	43.10	32.24
+PDL (ours)	7.75	**10.76**	**12.43**	16.09	**11.76**	**18.40**	41.40	**32.31**
+KDL (ours)	8.59	10.52	12.41	15.47	11.75	18.30	**43.50**	31.80
+DLL (ours)	7.85	10.54	12.41	**16.15**	11.74	18.33	41.60	32.25
VidVRD-II [31] $_{ACM-MM2021}$	12.41	12.97	13.63	14.85	13.47	25.93	55.60	41.70
+PDL (ours)	13.09	14.12	13.80	15.39	14.10	25.93	**55.70**	41.30
+KDL (ours)	12.23	13.17	13.82	15.26	13.62	26.02	54.90	**41.80**
+DLL (ours)	**13.28**	**14.33**	**14.13**	**15.62**	**14.34**	**26.65**	53.70	41.20



(a) `above`-related in VidVRD-II (b) `above`-related in **DLL (ours)** (c) `jump`-related in VidVRD-II (d) `jump`-related in **DLL (ours)**

Figure 5. **Visualization of the prediction logits of the visual relations within same actional or spatial pattern.** We normalize the scores over the whole map and only display a part of the predicates. As can be seen, DLL can promote the correlations between same-pattern predicates thus transfers the well represented head knowledge to tail ones that are within the same pattern.

not have these relevant data. From Table. 1, we have the following observations: 1) Compared with the two baselines, our DLL consistently improves the model performance on both R@K and mR@K metrics, as well as the Mean and mAP. *e.g.* DLL achieves a significant improvement (0.85%/1.36%) in mR@50/100 based on VidVRD-II and 0.33% in mR@100 based on VRD-STGC. This demonstrates the effectiveness of DLL in dealing with the long-tail problem in predicate distribution.Unlike existing methods that have to trade-off between R@K and mR@K, DLL can improve both metrics simultaneously. This can be confirmed by the 1.21%/2.46% and 0.5%/0.77% improvement in R@50/100, 0.87% and 0.78% improvement in Mean compared to two baselines with a slight loss in P@5. 2) Our DLL method outperforms our single module methods PDL and KDL. According to our observation, we can see that KDL could enhance mR@K metric while PDL is more capable of solving the long-tail problem than KDL. Although both PDL and KDL may have some slight losses on mR@50, the combination of PDL and KDL can be complementary to each other and achieve synergistic improvement.

4.5. Ablation Study

Effectiveness of the Adversarial Disentanglement and Mutual Calibration. Table. 2 shows the performance of PDL without the KDL module on VidVRD-II [31], with or without adversarial disentanglement and mutual calibration. Compared with the baseline in row one, PDL with adversarial disentanglement can further improve the performance and enable the actional and spatial classifiers to learn the actional and spatial feature more effectively, *e.g.* 0.69%/0.73% in mR@50/100 and 0.7% in R@100. This indicates that adversarial disentanglement can make the classifier obtain pure and sufficient learning. Therefore, we adopt adversarial disentanglement in our later work. We also observed that MC is a good way to help the classifier align with the data distribution of the final predicate labels by row 4 of Table. 2, which is reflected in the overall im-

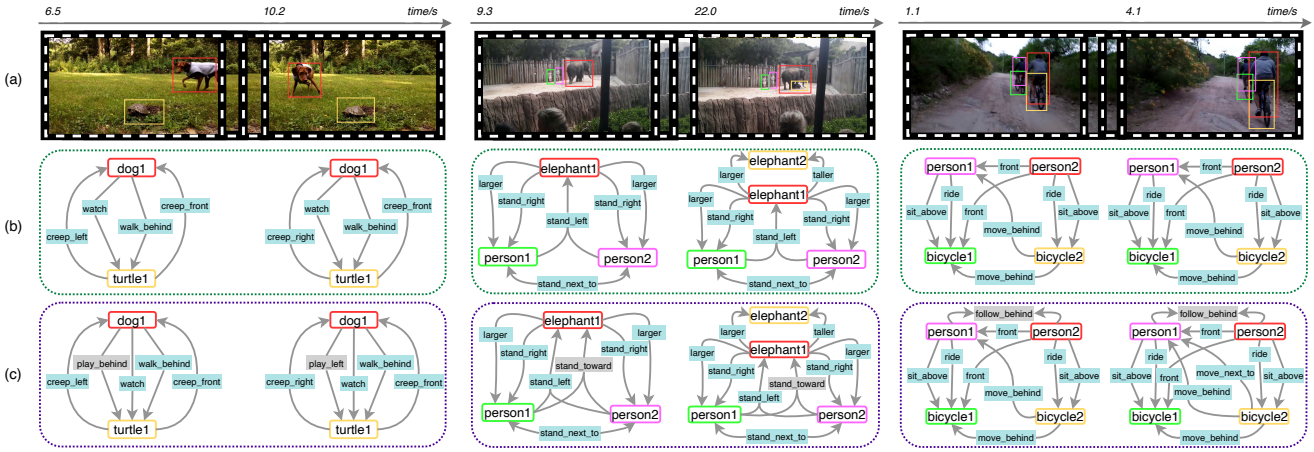


Figure 6. **Visualization of scene graph generation results and open-vocabulary relation prediction.** (a) Results of object detection. (b) Results of predicted triplets by **DLL**. (c) Possible open-vocabulary predicates captured by DLL, which are marked with grey background.

Table 2. Ablation (%) of w/o Adversarial Disentanglement (AD) and Mutual Calibration (MC) in PDL without KDL based on VidVRD-II [31].

AD	MC	mR@50/100	R@50/100	Mean	mAP	P@5/10
VidVRD-II		12.41/12.97	13.63/14.85	13.47	25.93	55.60/**41.70**
✓		12.56/13.70	13.59/**15.55**	13.88	**26.10**	54.50/41.25
	✓	12.26/13.09	**13.80**/15.18	13.58	25.12	53.20/38.75
✓	✓	**13.09/14.12**	**13.80**/15.39	**14.10**	25.93	**55.70**/41.30

Table 3. Ablation (%) of different steps of Mutual Calibration in PDL without KDL based on VidVRD-II [31].

step/η	mR@50/100	R@50/100	Mean	mAP	P@5/10
0/0	11.75/13.60	12.68/14.25	13.07	23.50	51.50/39.40
1/1e-1	11.79/12.68	12.84/14.60	12.98	24.45	53.40/40.80
1/1e-2	**13.58/14.62**	14.04/15.53	**14.44**	**26.00**	**56.10**/40.80
2/1e-2	12.23/13.13	13.22/14.62	13.30	24.04	54.30/39.95
3/1e-2	12.95/14.23	**14.21/15.82**	14.30	25.59	54.30/**41.00**
1/1e-3	12.40/13.53	13.34/14.91	13.55	26.31	53.80/39.80

provement of performance.

Number of the Mutual Calibration Steps. Table. 3 shows the performance of different steps of mutual calibration with different scales of hyper-parameter η. Compared with the baseline that has no mutual calibration (step=0), the model that iteratively calibrates once achieves better results. From the data in Table 3, we can see that mutual calibration improves not only R@K, but also mR@K, mAP and Mean, which indicates that mutual calibration can facilitate label coupling. However, as the hyper-parameter η and iteration steps increase, the performance decreases, which suggests that excessive calibration is detrimental.

Growth Rate of Knowledge Transfer. Table. 4 shows the performance of using different β in Knowledge Transfer without PDL module on VidVRD-II [31]. It can be seen that a certain growth rate can improve the performance and promote the mutual learning between the model and the correlation matrix $\boldsymbol{M}$, which also proves the effectiveness of the KDL module.

Table 4. Ablation (%) of different scales of β in Knowledge Transfer without PDL based on VidVRD-II [31].

β	mR@50/100	R@50/100	Mean	mAP	P@5/10
1e-1	**12.23**/13.17	**13.82/15.26**	**13.62**	**26.02**	**54.90/41.80**
1e-2	12.10/12.99	13.69/14.97	13.44	25.68	54.30/40.85
1e-3	11.99/**13.64**	12.84/14.60	13.27	25.52	53.40/39.90
1e-4	12.14/12.90	13.34/14.93	13.33	24.97	53.50/41.45

4.6. Qualitative Results

Visualization of correlations of predicates within the same actional or spatial pattern. Fig. 5 compares the correlations of predicates generated by baseline and DLL. As can be seen, DLL can promote the correlations between the similar predicates via non-target knowledge calibration.

Visualization of predictions in open-vocabulary scenario. Fig. 6 shows some detected predicates of DLL. Comparing to the vanilla scene graphs, our generated ones are more promising in predicting open-vocabulary predicates.

5. Conclusion

In this paper we target at VidSGG task, or more specifically, the long-tail problem inherently existed in the training data that hinders the VidSGG performance. To this end, we introduce **DLL**, a novel approach that decouples labels into actional and spatial patterns and learns them separately. DLL also transfers the non-target knowledge from head to tail predicates within the same pattern to further calibrate the tail predicate distribution. By combining these two decoupling method, we are able to create an "unbiased" scene graph, which accurately captures the visual relations in the video. Extensive results verify that DLL achieves state-of-the-art performance on various metrics across different scenarios, especially on tail predicates. Furthermore, the decoupled manner also improves the zero-shot learning ability of VidSGG model in open-vocabulary relation scenario.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, pages 2425–2433, 2015.

[2] Q. Cao, H. Huang, X. Shang, B. Wang, and T. S. Chua. 3-d relation network for visual relation recognition in videos. *Neurocomputing*, 432:91–100, 2021.

[3] Shuo Chen, Zenglin Shi, Pascal Mettes, and Cees G. M. Snoek. Social fabric: Tubelet compositions for video relation detection. In *ICCV*, 2021.

[4] Siqi Chen, Jun Xiao, and Long Chen. Video scene graph generation from single-frame weak supervision. In *The Eleventh International Conference on Learning Representations*, 2023.

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

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

[7] Guillaume Desjardins, Aaron Courville, and Yoshua Bengio. Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474*, 2012.

[8] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. Dual encoding for video retrieval by text. *TPAMI*, 2021.

[9] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2525–2534. PMLR, 2019.

[10] Shengyu Feng, Subarna Tripathi, Hesham Mostafa, Marcel Nassar, and Somdeb Majumdar. Exploiting long-term dependencies for generating dynamic scene graphs. *arXiv preprint arXiv:2112.09828*, 2021.

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[12] Kaifeng Gao, Long Chen, Hanwang Zhang, Jun Xiao, and Qianru Sun. Compositional prompt tuning with motion cues for open-vocabulary video relation detection. *arXiv preprint arXiv:2302.00268*, 2023.

[13] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016.

[14] Xianjing Han, Xuemeng Song, Xingning Dong, Yinwei Wei, Meng Liu, and Liqiang Nie. Dbiased-p: Dual-biased predicate predictor for unbiased scene graph generation. *IEEE Transactions on Multimedia*, 2022.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *stat*, 1050:9, 2015.

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

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[18] Lin Li, Long Chen, Hanrong Shi, Wenxiao Wang, Jian Shao, Yi Yang, and Jun Xiao. Label semantic knowledge distillation for unbiased scene graph generation. *arXiv preprint arXiv:2208.03763*, 2022.

[19] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, 2021.

[20] Yicong Li, Xun Yang, Xindi Shang, and Tat-Seng Chua. Interventional video relation detection. In *ACM MM*, pages 4091–4099, 2021.

[21] 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 *CVPR*, pages 482–490, 2020.

[22] Chenchen Liu, Yang Jin, Kehan Xu, Guoqiang Gong, and Yadong Mu. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In *CVPR*, pages 10840–10849, 2020.

[23] Hengyue Liu, Ning Yan, Masood S Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *CVPR*, 2021.

[24] Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *CVPR*, pages 2930–2939, 2016.

[25] J. Nemeth. Adversarial disentanglement with grouped observations. 2020.

[26] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *NIPS*, 2017.

[27] X. Peng, Z. Huang, X. Sun, and K. Saenko. Domain agnostic learning with disentangled representations. 2019.

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

[29] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[30] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[31] Xindi Shang, Yicong Li, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Video visual relation detection via iterative inference. In *ACM MM*, pages 3654–3663, 2021.

[32] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. Video visual relation detection. In *ACM MM*, pages 1300–1308, 2017.

[33] Cees GM Snoek, Marcel Worring, et al. Concept-based video retrieval. *FOUND TRENDS INF RET*, 2(4):215–322, 2009.

[34] Zixuan Su, Xindi Shang, Jingjing Chen, Yu-Gang Jiang, Zhiyong Qiu, and Tat-Seng Chua. Video relation detection via multiple hypothesis association. In *ACM MM*, pages 3127–3135, 2020.

[35] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pages 3716–3725, 2020.

[36] 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.

[37] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, pages 4631–4640, 2016.

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

[39] Yao-Hung Hubert Tsai, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *CVPR*, pages 10424–10433, 2019.

[40] Yinwei Wei, Xiang Wang, Weili Guan, Liqiang Nie, Zhouchen Lin, and Baoquan Chen. Neural multimodal cooperative learning toward micro-video understanding. *TIP*, 29:1–14, 2019.

[41] Sangmin Woo, Junhyug Noh, and Kangil Kim. What and when to look?: Temporal span proposal network for video visual relation detection. *arXiv preprint arXiv:2107.07154*, 2021.

[42] Sitao Xiang and Hao Li. Disentangling style and content in anime illustrations. *arXiv preprint arXiv:1905.10742*, 2019.

[43] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, pages 9777–9786, 2021.

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

[45] Li Xu, Haoxuan Qu, Jason Kuen, Jiuxiang Gu, and Jun Liu. Meta spatio-temporal debiasing for video scene graph generation. In *ECCV*, pages 374–390. Springer, 2022.

[46] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 265–273, 2020.

[47] SiCheng Yang, Methawee Tantrawenith, Haolin Zhuang, Zhiyong Wu, Aolan Sun, Jianzong Wang, Ning Cheng, Huaizhen Tang, Xintao Zhao, Jie Wang, et al. Speech representation disentanglement with adversarial mutual information learning for one-shot voice conversion. *arXiv preprint arXiv:2208.08757*, 2022.

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