Meta Spatio-Temporal Debiasing for Video Scene Graph Generation

Li Xu[1⋆], Haoxuan Qu[1⋆], Jason Kuen[2], Jiuxiang Gu[2], and Jun Liu[1⋆⋆]

[1] Singapore University of Technology and Design
{li_xu, haoxuan_qu}@mymail.sutd.edu.sg, jun_liu@sutd.edu.sg
[2] Adobe Research
{kuen, jigu}@adobe.com

Abstract. Video scene graph generation (VidSGG) aims to parse the video content into scene graphs, which involves modeling the spatio-temporal contextual information in the video. However, due to the long-tailed training data in datasets, the generalization performance of existing VidSGG models can be affected by the **spatio-temporal conditional bias** problem. In this work, from the perspective of meta-learning, we propose a novel Meta Video Scene Graph Generation (**MVSGG**) framework to address such a bias problem. Specifically, to handle various types of spatio-temporal conditional biases, our framework first constructs a support set and a group of query sets from the training data, where the data distribution of each query set is different from that of the support set w.r.t. a type of conditional bias. Then, by performing a novel **meta training and testing** process to optimize the model to obtain good testing performance on these query sets after training on the support set, our framework can effectively guide the model to learn to well generalize against biases. Extensive experiments demonstrate the efficacy of our proposed framework.

Keywords: VidSGG, Long-tailed bias, Meta learning

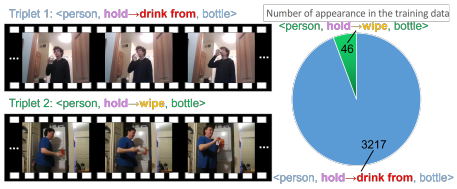
1 Introduction

A scene graph is a graph-based representation, which encodes different visual entities as nodes and the pairwise relationships between them as edges, i.e., in the form of `subject predicate object` relation triplets [13, 29]. Correspondingly, the task of video scene graph generation (VidSGG) aims to parse the video content into a sequence of spatio-temporal relationships between different objects of interest [40, 29]. Since it can provide refined and structured scene understanding, the video scene graph representation has been widely used in various higher-level video tasks, such as video question answering [15, 35], video captioning [8, 38], and video retrieval [16, 7].

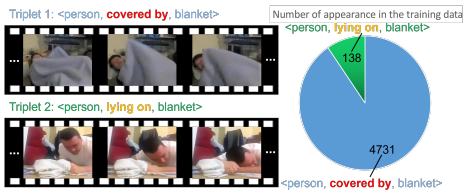
⋆ Both authors contributed equally to the work.
⋆⋆ Corresponding Author

However, despite of the great progress of VidSGG [36, 5, 2, 22], most existing approaches tackling this task may suffer from the problem of *spatio-temporal conditional biases*. Specifically, as shown by previous works [33, 4, 19], there exist long-tailed training data issues in existing SGG datasets. While in the context of VidSGG, given the complex spatio-temporal nature of this task, such long-tailed issues can lead to spatio-temporal conditional biases that affect the model generalization performance. Here conditional biases mean the problem that once the model detects certain context information (i.e., conditions) in the visual content, it is likely to directly predict certain labels (i.e., biased prediction), which however may contradict with the ground-truth. Some works [27, 43] also refer to this problem as spurious correlation. In particular, in the VidSGG task, this conditional bias issue can be further divided into two sub-problems: temporal conditional bias and spatial conditional bias.

For **temporal** conditional bias, as shown in the example of Fig. 1 (a), if `person hold bottle` appears first (i.e., **temporal** context) in the video, and in most cases, `person drink from bottle` happens next, then a conditional bias can be established towards `person drink from bottle` conditioned on `person hold bottle` along the temporal axis in the video. Due to such temporal conditional bias, once the previous video part involves `person hold bottle`, the trained model is very likely to simply predict `person drink from bottle` for the next part, which however can contradict with the ground truth as in the example.



(a) Example of temporal conditional bias.



(b) Example of spatial conditional bias.

Fig. 1. Illustration of spatio-temporal conditional biases examples from Action Genome dataset [12]. (a) illustrates an example of temporal conditional bias towards `person drink from bottle` conditioned on `person hold bottle` in consecutive video parts, and (b) presents an example of spatial conditional bias towards the predicate of `covered by` conditioned on the subject-object pair of `person` and `blanket`. Such spatio-temporal conditional biases can affect the performance of VidSGG models.

Such a conditional bias problem also exists in the **spatial** domain when tackling the VidSGG task. For example, as shown in Fig. 1 (b), there are many more videos containing `person covered by blanket` than `person lying on blanket` in the VidSGG dataset: Action Genome [12]. This can lead to a conditional bias towards the predicate (`covered by`) given the **spatial** contexts of `person` and `blanket`. Due to such spatial conditional bias, once `person` and `blanket` appear in the video, the model tends to directly predict `person covered by blanket` that however can be incorrect.

We observe that such spatio-temporal conditional bias problems widely exist in VidSGG datasets [29, 12]. Meanwhile, to correctly infer the relation triplets, VidSGG models need to effectively model the spatio-temporal context information in the video [36, 5, 29]. Thus the models can be easily prone to exploiting the conditional biases w.r.t. the relation triplet components (e.g., the predicate) based on the spatio-temporal contexts during training, and then fail to generalize to data samples in which such conditional biases no longer hold as shown in Fig. 1. Therefore, to address this issue for obtaining better generalization performance, we propose a novel meta-learning based framework, Meta Video Scene Graph Generation (**MVSGG**).

Meta learning, also known as learning to learn, aims to enhance the model generalization capacity by incorporating *virtual testing* during model training [6, 23, 11]. Inspired by this, to improve the generalization performance of VidSGG models against the conditional biases, our framework incorporates a *meta training and testing* scheme. More concretely, we can split the training set to construct a support set for *meta training* and a query set for *meta testing*, which have different data distributions w.r.t. the conditional biases, i.e., creating a virtual testing scenario where simply exploiting the conditional biases during training would lead to poor testing performance. For example, given the same subject-object pair of `person` and `blanket`, if the support set contains more `person covered by blanket` relation triplets, the query set can contain more `person lying on blanket` triplets on the contrary. We first use the support set to train the model (i.e., *meta training*), and then evaluate the trained model on the query set (i.e., *meta testing*). According to the evaluation performance (loss) on the query set, we can further update the model to obtain better generalization performance. Since the query set is distributionally different from the support set w.r.t. the conditional biases, by improving the testing performance on the query set after training on support set via *meta training and testing*, our model is driven to learn to capture the "truly" generalizable features in the data instead of relying on the biases. Thus our model can "learn to generalize" well, even when handling the "difficult" testing samples that contradict the biases in training data.

Moreover, there can exist various types of conditional biases besides the ones shown in Fig. 1. For example, there can also exist a conditional bias w.r.t. the object based on subject-predicate pair in relation triplets, if an object appears more frequently given the same subject-predicate pair in the training data. Thus to better handle such a range of conditional biases, we can construct a group of query sets, where each query set is distributionally different from the support set w.r.t. one type of conditional bias. In this manner, by utilizing all these query sets to improve the model generalization performance via *meta training and testing*, our framework can effectively address various types of conditional biases in the video, and enhance the robustness of the VidSGG model.

Our MVSGG framework is general since it only changes the model training scheme (i.e., via *meta training and testing*), and thus can be flexibly applied to

various off-the-shelf VidSGG models. We experiment with multiple models, and achieve consistent improvement of model performance.

The contributions of our work are summarized as follows. 1) We propose a novel *meta training and testing* framework, MVSGG, for effectively addressing the spatio-temporal conditional bias problem in VidSGG. 2) By constructing a support set and multiple query sets w.r.t. various types of conditional biases, our framework can enable the trained model to learn to generalize well against various types of conditional biases simultaneously. 3) Our framework achieves significant performance improvement when applied on state-of-the-art models on the evaluation benchmarks [29, 12].

2 Related Work

Scene Graph Generation (SGG). Being able to provide structured graph-based representation of an image or a video, scene graph generation (SGG) has attracted extensive research attention [13, 34, 40, 18, 31, 10, 3, 21, 29, 36]. For image SGG (ImgSGG), a variety of methods [34, 46, 45, 44] have been proposed. Suhail et al. [31] proposed an energy-based framework to improve the model performance by learning the scene graph structure. Yang et al. [41] investigated the diverse predictions for predicates in SGG from a probabilistic view.

Besides ImgSGG, there are also increasing research efforts exploring the task of video scene graph generation (VidSGG) [29, 20, 2, 36]. This task provides two task settings based on the granularity of the generated video scene graphs: video-level [29, 37, 24, 28, 20, 2] and frame-level [36, 5]. For video-level VidSGG, models generate scene graphs based on the video clip, where each node encodes the spatio-temporal trajectory of an object, and the connecting edge denotes the relation between two objects. Shang et al. [29] first investigated this problem setting, and proposed to extract improved Dense Trajectories features [39] for handling this problem. Later on, some other methods have been proposed to solve this video-level VidSGG problem from different perspectives, including the fully-connected spatio-temporal graph [37], and iterative relation inference [28]. For frame-level VidSGG, a scene graph is generated for each video frame [36, 5]. To handle this problem setting, Teng et al. [36] proposed to use a hierarchical relation tree to capture the spatio-temporal context information. Cong et al. [5] proposed to solve this problem via a spatio-temporal transformer.

For SGG, there often exists the long-tailed data bias issue that hinders models from obtaining better performance. To solve this problem, various debiasing methods have been proposed. Tang et al. [33] introduced a debiasing framework by utilizing the Total Direct Effect (TDE) analysis. Guo et al. [10] proposed a balance adjustment method to handle this issue. Li et al. [19] explored a causality-inspired interventional approach to reduce the data bias in VidSGG. Differently, to cope with the spatio-temporal conditional bias problem in SGG, from the perspective of meta learning, we propose a novel learning framework that can train the SGG model to learn to better generalize against biases.

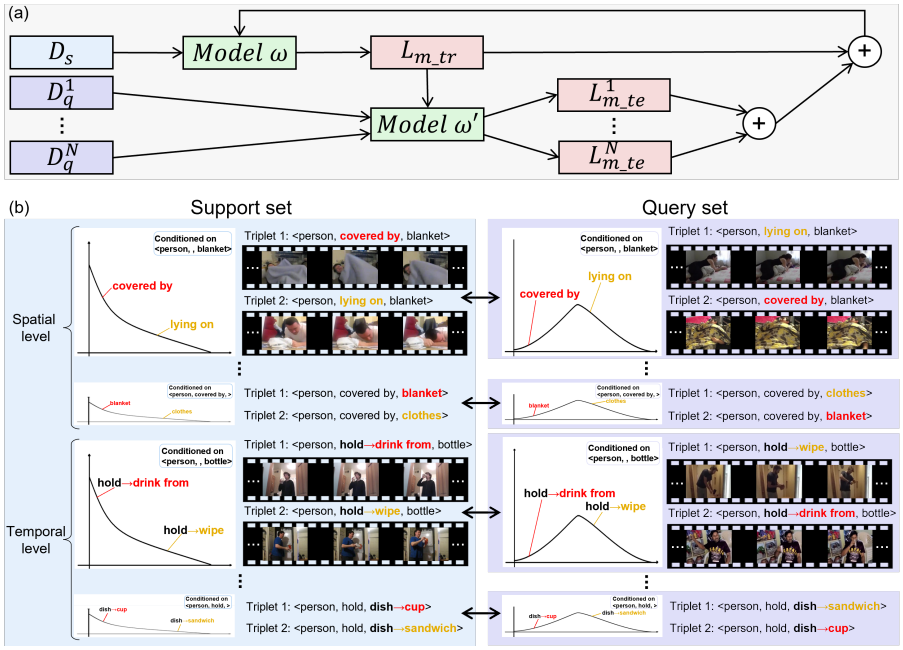


Fig. 2. Overview of our framework. (a) illustrates our *meta training and testing* scheme. 1) We first train the model (with parameters ω) on the support set (D_s) by optimizing the loss function (L_{m_tr}), and thus obtain the model with updated parameters (ω'), i.e., *meta training* process. 2) We evaluate the updated model on a group of query sets ($\{D_q^n\}_{n=1}^N$), by computing the losses ($\{L_{m\_te}^n(\omega')\}_{n=1}^N$) on these query sets (i.e., *meta testing* process). 3) Finally, based on the evaluation losses, we perform meta-optimization to update the model to improve its generalization performance. (b) shows that to address various types of **spatial level** and **temporal level** conditional biases, we construct a support set and a group of query sets for *meta training and testing*. The data distribution of each query set (see right side of (b)) is different from that of the support set (see left side of (b)) w.r.t. a type of conditional bias.

Meta Learning. As a group of representative works in meta learning, MAML [6] and its following works [23, 32, 26] mainly tackle the few-shot learning problem. These approaches often need to perform test-time model update for fast adaptation to new few-shot tasks. While recently, meta learning techniques have also been explored in other tasks [1, 17, 11, 9, 25] to improve the model performance without the need of test-time update, such as in domain generalization [1] and point cloud classification [11]. Different from existing works, here to address the challenging spatio-temporal conditional bias problem in SGG, we propose a framework that optimizes the model via *meta training and testing* over the constructed support set and query sets with different data distributions w.r.t. the conditional biases.

3 Method

As discussed above, affected by the conditional biases in the dataset, the VidSGG model can fail to generalize to the data samples where the biases do not hold. To address this problem, we aim to train the model to learn generalizable features in the data instead of exploiting biases. Here generalizable features refer to the learned features that can enable the model to make unbiased predictions, i.e., obtaining robust performance. How to achieve this goal? We notice that some meta-learning works [6, 23, 32] propose to boost the model learning ability via *meta training and testing*. Concretely, these works use *meta training and testing* to mimic the model training and testing for improving model generalization performance. Inspired by this, we propose a novel MVSGG framework, that optimizes the VidSGG model via *meta training and testing* for robust generalization performance against the biases.

More specifically, our framework first splits the training set (D_{train}) into a support set (D_s) for *meta training*, and a group of (N) query sets ($\{D_q^n\}_{n=1}^N$) for *meta testing*, where each query set is distributionally different from the support set w.r.t. one type of conditional bias. Then we first train the model using the support set (i.e., *meta training*), and then evaluate the model testing performance on each of the query sets (i.e., *meta testing*). Since the biases in meta-training data (support set) do not hold in meta-testing data (query sets) due to their different data distributions, if the model trained on the support set can still obtain good testing performance under this condition, it indicates the model has learned more generalizable features rather than biases during the training process. As a result, we can optimize the model performance in *meta testing*, which can serve as a *generalization feedback*, to drive and adjust the model training on the support set towards learning more generalizable features. Below, we first introduce the *meta training and testing* scheme in our framework, and then describe how to construct the support set and query sets.

3.1 Meta Training and Testing

Meta training. Using the support set D_s, we first train a VidSGG model (with parameters ω), via conventional gradient update. Specifically, we compute the model loss on the support set as:

$$L_{m\_tr}(\omega) = L(D_s; \omega) \tag{1}$$

where $L(\cdot)$ denotes the loss function (e.g., cross-entropy loss) for training the VidSGG model. Then we update the model parameters via gradient descent as:

$$\omega' = \omega - \alpha \nabla_\omega L_{m\_tr}(\omega) \tag{2}$$

where α is the learning rate for *meta training*. Note that the parameters update in this step is virtual (i.e., the updated parameters ω' is merely intermediate parameters), and the actual update for parameters ω will be performed in the meta-optimization step.

Meta testing. After *meta training* on the support set (D_s), we then evaluate the generalization performance of the model with the updated parameters (ω'),

on the query sets ($\{D_q^n\}_{n=1}^N$). In particular, for each query set D_q^n, we compute the model loss $L_{m\_te}^n$ on this query set as:

$$L_{m\_te}^n(\omega') = L(D_q^n; \omega') \tag{3}$$

This computed loss can measure the model generalization performance on the query set after training on the support set, and will be used to provide feedback on *how the model should be updated so that it can generalize to different data distributions against the biases* in the following meta-optimization step.

Meta-optimization. As discussed above, we aim to optimize the model parameters (ω), so that *after the training (update) on the support set (i.e.,* $\omega \rightarrow \omega'$*), it can also obtain good testing performance (i.e., lower* $L_{m\_te}^n(\omega')$*) on all the query sets against the biases in training data.* Towards this goal, inspired by MAML [6], the meta-optimization objective can be formulated as:

$$\begin{aligned} &\min_{\omega} \; L_{m\_tr}(\omega) + \sum_{n=1}^{N} L_{m\_te}^n(\omega') \\ =&\min_{\omega} \; L_{m\_tr}(\omega) + \sum_{n=1}^{N} L_{m\_te}^n\big(\omega - \alpha \nabla_{\omega} L_{m\_tr}(\omega)\big) \end{aligned} \tag{4}$$

where the first term denotes the model training performance, while the second term denotes the model *generalization* performance (with the updated parameters ω'). Note that the above meta-optimization is performed over the initial model parameters ω, while ω' is merely intermediate parameters for evaluating the model generalization performance ($L_{m\_te}^n(\omega')$) during *meta testing*. Based on the meta-optimization objective in Eqn. 4, we can update the model parameters ω as:

$$\omega \leftarrow \omega - \beta \nabla_{\omega} \Big(L_{m\_tr}(\omega) + \sum_{n=1}^{N} L_{m\_te}^n\big(\omega - \alpha \nabla_{\omega} L_{m\_tr}(\omega)\big) \Big) \tag{5}$$

where β denotes the learning rate for meta-optimization. Via such optimization, the model is driven to learn to capture more generalizable features to generalize well against biases.

Here we provide an intuitive analysis of such meta-optimization. During the above "learning to learn" process, the model is first trained (updated) over the support set (i.e., $\omega \rightarrow \omega'$). In this step, the biases in meta-training data (support set) can be learned by the model, since such biases can contribute to model performance on meta-training data. However, to generalize well to the meta-testing data (query sets) where biases in meta-training data (support set) no longer hold, the model needs to learn to avoid learning biases and instead capture more generalizable features during *meta training*. This means that the second term in Eqn. 5 that involves the second-order gradients of ω (i.e., meta-gradients): $\nabla_{\omega} L_{m\_te}^n\big(\omega - \alpha \nabla_{\omega} L_{m\_tr}(\omega)\big)$, serves as a *generalization feedback* to the model learning process ($\omega \rightarrow \omega'$) on the support set about how to learn more generalizable features.

From the above analysis, we can also conclude that the efficacy of our framework for debiasing lies in the simulated difficult testing scenarios where training

data biases no longer hold. This also implies that we are not using the query sets to simulate the data distribution of the real testing set, which is unknown during model training. Instead, we only need to make the difference between the data distribution of meta-testing data (query sets) and that of meta-training data (support set) to be as large as possible w.r.t. the biases, so as to drive the model learning to learn more generalizable features. We also provide theoretical analysis of the efficacy of this framework for alleviating the bias learning in the supplementary. We perform the above three steps (i.e., meta training, meta testing and meta-optimization) iteratively until the model converges.

3.2 Dataset Split

As mentioned above, to handle various types of conditional biases, we split the original training set to construct a support set and a group of N query sets for *meta training and testing*. In this way, the purpose of the following dataset split strategy is to make each query set distributionally different from the support set w.r.t. one type of conditional bias. Under the guidance of this strategy, we can easily construct the support set and query sets. Below we first discuss the details of the support set and query sets, and then introduce the strategy for constructing each query set by selecting the data samples, of which the data distributions have the largest KL divergences to the support set w.r.t. the corresponding type of conditional bias. Some visualization examples of data distributions of the support set and query sets can refer to supplementary.

Support Set and Query Sets. We first randomly select a part of the training set data as the support set (D_s), and the remaining part of the training set will be used to construct various query sets ($\{D_q^n\}_{n=1}^N$), where each query set is designed to address one type of conditional bias. Since the conditional biases in VidSGG can be roughly grouped into the spatial level and the temporal level, we correspondingly construct our query sets based on these two levels, as follows.

Spatial level. There can exist conditional biases between a part of the relation triplet (e.g., the predicate) and the remaining parts (i.e., the spatial contexts). For example, as shown in Fig. 1 (b), given the same subject-object pair of `human` and `blanket`, the corresponding predicate is `covered by` in most triplets. To reduce such spatial conditional bias w.r.t. the **predicate conditioned on subject-object pair**, we can construct a query set, in which the distribution of the predicates conditioned on the same subject-object pair is different from the support set, as shown in Fig. 2.

Similarly, conditional bias can also exist w.r.t. the **predicate conditioned on subject**. For instance, if there are many more triplets containing `bear play` than the triplets containing `bear bite` in the dataset, a conditional bias can be established towards the predicate `play` given the subject `bear`. Thus we can build a query set where the distribution of the predicates (e.g., `play`, `bite`) conditioned on the same subject (e.g., `bear`) is different from that of the support set. In a similar way, we can also construct a query set to handle the conditional bias w.r.t. the **predicate conditioned on object**.

Therefore, 3 query sets can be constructed to handle the corresponding 3 types of conditional biases w.r.t. the *predicate* (*predicate-centered*) conditioned on other parts of the relation triplet (i.e., the subject-object pair, or the subject, or the object), as discussed above. Similarly, when considering the conditional biases w.r.t. the *subject* (*subject-centered*) conditioned on other parts of the triplet, we can also construct 3 query sets, and the same goes for the *object-centered* scenario. Thus we will construct a total of **9** query sets for handling these different types of spatial conditional biases.

Temporal level. In VidSGG, when predicting a relation triplet, besides spatial contexts, there can also exist conditional biases between the current triplet and its temporal contexts. Specifically, temporal conditional bias can exist between the current triplet and the triplets that appear before it, and for simplicity, we refer to this case as *forward case*. Similarly, conditional bias can also exist between the current triplet and the triplets that appear after it (*backward case*). For these two cases, the query set construction procedures are similar, and below we take the forward case as the example to describe such procedures.

For example, as shown in Fig. 1 (a), if `human hold bottle` happens first in the video, and then `human drink from bottle` follows in most cases, then there can exist temporal conditional bias between the **previous predicate** (e.g., `hold`) **and current predicate** (e.g., `drink from`), based on the subject-object pair (e.g., `human` and `bottle`). To handle such temporal conditional bias, we can construct a query set, in which the distribution w.r.t. the temporal change of predicates, conditioned on the same subject-object pair, is different from that in the support set. For example, if the support set has more videos containing `human hold bottle`→`human drink from bottle`, the query set will involve more videos containing other cases w.r.t. the temporal change of predicates, such as `human hold bottle`→`human wipe bottle`.

Similarly, temporal conditional bias can also exist between the **previous subject and current subject**, and we can construct a query set for handling this type of conditional bias. In a similar manner, we can also construct a query set for handling the temporal conditional bias between the **previous object and current object**. Therefore, we construct 3 query sets to handle the above 3 types of temporal conditional biases in the *forward case*. Similarly, we can also construct 3 query sets for the *backward case*, and thus a total of **6** query sets for handling various types of temporal conditional biases can be obtained.

As a result, considering both spatial-level and temporal-level conditional biases, we construct **9+6=15** query sets (N=15) in total from the training data.

Query Sets Construction Strategy. For constructing each query set, we need to select suitable video samples from the candidate video samples, so that the difference between the data distribution of the triplets in the selected videos (i.e., query sets) and that of the support set is as large as possible w.r.t. the corresponding type of conditional bias. Here to achieve this goal, we adopt an efficient and generalizable strategy by maximizing the KL divergence between the data distributions of the query set and support set w.r.t. the biases, which can be applied to construct each of the 15 query sets. Below we take the process

of constructing the query set for handling the spatial conditional biases w.r.t. the **predicate conditioned on subject** as an example, to describe such a strategy.

As mentioned before, for handling this conditional bias, we aim to construct a query set, of which the distribution of the **predicates** (e.g., `play`, `bite`) given the same **subject** (e.g., `bear`), is different from the support set. For simplicity, we use ϕ_q to denote such a distribution of the query set, and ϕ_s to denote this distribution of the support set. Then since the KL divergence can be used to measure the difference of two distributions, we here aim to construct a query set, so that the KL divergence between ϕ_q and ϕ_s (i.e., $D_{KL}(\phi_q \parallel \phi_s)$) is large.

To this end, we perform the following four steps. (1) We first compute the distribution ϕ_s, i.e., computing the probability of the occurrence of each **predicate** conditioned on the same **subject** (e.g., $p(\texttt{play}|\texttt{bear})$, $p(\texttt{bite}|\texttt{bear})$) in the support set. (2) Then, assuming we have a total of N_c candidate video samples for constructing the query sets, since each candidate video sample (i) contains multiple relation triplets, we can also compute its corresponding data distribution (ϕ_c^i, $i \in \{1, ..., N_c\}$). (3) Since we aim to select a set of video samples to construct the query set, so that ϕ_q is different from ϕ_s (i.e., large $D_{KL}(\phi_q \parallel \phi_s)$), we compute the KL divergence between ϕ_c^i of each candidate video sample and ϕ_s (i.e., $D_{KL}(\phi_c^i \parallel \phi_s)$) that can be computed efficiently. (4) Finally, we can select the set of video samples that have the largest KL divergences w.r.t. ϕ_s, to construct the query set.

In a similar manner, we can apply the above strategy to automatically construct other query sets. Note that different query sets can share common data samples. Moreover, to help cover the wide range of possible conditional biases in the dataset, instead of fixing the support set and query sets during the whole training process, at the beginning of each training epoch, we *randomly* select a part of the training set to re-construct the support set, and use the remaining part to automatically re-construct various query sets via the above strategy. In this way, by performing *meta training and testing*, during the whole training process, our model can learn to effectively handle various types of possible conditional biases.

3.3 Training and Testing

We can flexibly apply our framework to train the off-the-shelf VidSGG models. During training, at each epoch, we first split the training set to construct a support set and a group of query sets as discussed above. Then we perform *meta training and testing* over the support set and query sets, to iteratively optimize the VidSGG model. During testing, we can evaluate the trained model on the testing set in the conventional manner.

4 Experiments

We evaluate our framework on two datasets for two evaluation settings in VidSGG respectively: ImageNet-VidVRD [29] for video-level VidSGG, and Action Genome [12] for frame-level VidSGG. More experiment results are in supplementary.

ImageNet-VidVRD (VidVRD). VidVRD dataset [29] contains 1000 video samples with 35 object categories and 132 predicate categories. For each video in VidVRD dataset, the model needs to predict a set of relation instances, and each relation instance contains a relation triplet with the subject and object trajectories. Following [29, 28], we use two evaluation protocols on this dataset: relation detection and relation tagging. For relation detection, we count a predicted relation instance as a correct one, if its relation triplet is the same with a ground truth, and their trajectory vIoU (volume IoU) of the subject and object are both larger than the threshold of 0.5. In the same way as [29, 28], we adopt Mean Average Precision (mAP), Recall@50 (R@50) and Recall@100 (R@100) to evaluate the model performance on relation detection. While in relation tagging, for a predicted relation instance, following [29, 28] we only consider the correctness of its relation triplet, and ignore the precision of its subject and object trajectories. The evaluation metrics of Precision@1 (P@1), Precision@5 (P@5) and Precision@10 (P@10) are used in relation tagging [29, 28].

Action Genome (AG). AG dataset [12] provides scene graph annotation for each video frame, i.e., the model needs to predict the scene graph of each frame. AG dataset contains 234253 video frames with 35 object categories and 25 predicate categories. Following [36, 5, 12], we evaluate models on three standard sub-tasks on this dataset: predicate classification (PredCls), scene graph classification (SGCls) and scene graph detection (SGDet). For these three sub-tasks, in line with [36], we use Recall (R@20, R@50), Mean Recall (MR@20, MR@50), mAP_{rel} and wmAP_{rel} to measure model performance.

4.1 Implementation Details

We conduct our experiments on an RTX 3090 GPU. For experiments of video-level VidSGG on VidVRD dataset, we use the VidVRD-II network [28] as the backbone of our framework, which exploits the spatio-temporal contexts via iterative relation inference. For experiments of frame-level VidSGG on AG dataset, we use TRACE network [36] as the backbone of our framework, which adaptively aggregates contextual information to infer the scene graph.

On these two datasets, at each training epoch, we randomly select 60% of the training samples as the support set, and the remaining training samples are used to construct the query sets. We set the size of each query set to 100 on VidVRD, and 200 on AG. Note that in AG dataset, since the subject of all relation triplets is fixed to "person", we skip the query sets for handling the conditional biases w.r.t. the prediction of subject (e.g., *subject-centered* group) in this dataset. We set the learning rate (α) for *meta training* to 0.0005, and the learning rate (β) for meta-optimization to 0.01.

4.2 Experimental Results

On VidVRD dataset, compared to existing approaches, our method achieves the best performance across all metrics on both relation detection and relation tagging as shown in Table 1. This demonstrates that by reducing various types of

Table 1. Comparison with state-of-the-arts on VidVRD dataset.

Method	Relation Detection			Relation Tagging		
	mAP	R@50	R@100	P@1	P@5	P@10
VidVRD [29]	8.58	5.54	6.37	43.00	28.90	20.80
GSTEG [37]	9.52	7.05	8.67	51.50	39.50	28.23
VRD-GCN [24]	14.23	7.43	8.75	59.50	40.50	27.85
VRD-GCN+siamese [24]	16.26	8.07	9.33	57.50	41.00	28.50
VRD-STGC [20]	18.38	11.21	13.69	60.00	43.10	32.24
VidVRD+MHA [30]	15.71	7.40	8.58	40.00	26.70	18.25
VRD-GCN+MHA [30]	19.03	9.53	10.38	57.50	41.40	29.45
TRACE [36]	17.57	9.08	11.15	61.00	45.30	33.50
Social Fabric [2]	20.08	13.73	16.88	62.50	49.20	38.45
IVRD [19]	22.97	12.40	14.46	68.83	49.87	35.57
VidVRD-II [28]	29.37	19.63	22.92	70.40	53.88	40.16
VidVRD-II [28] + Reweight [33]	29.52	19.80	22.96	71.50	54.30	40.20
VidVRD-II [28] + TDE [33]	29.78	19.90	23.04	72.50	54.50	40.65
VidVRD-II [28] + DLFE [4]	29.92	19.98	23.16	73.50	54.90	41.10
Ours	**31.57**	**21.16**	**24.57**	**79.00**	**57.60**	**43.20**

Table 2. We apply our framework on various models, and obtain consistent performance improvement on VidVRD dataset.

Method	Relation Detection			Relation Tagging		
	mAP	R@50	R@100	P@1	P@5	P@10
VRD-STGC[20]	18.38	11.21	13.69	60.00	43.10	32.24
VRD-STGC + Ours	20.76	12.62	15.78	65.50	44.90	33.15
Independent baseline[28]	27.49	18.18	21.28	67.10	50.18	38.02
Independent baseline + Ours	30.02	19.86	23.10	75.50	53.60	40.80
VidVRD-II[28]	29.37	19.63	22.92	70.40	53.88	40.16
VidVRD-II + Ours	31.57	21.16	24.57	79.00	57.60	43.20

Table 3. We apply our framework on different SOTA models for image SGG, and obtain consistent performance improvement.

Method	SGGen			SGCls			PredCls		
	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
VCTree [34]	5.2	7.1	8.3	9.1	11.3	12.0	14.1	17.7	19.1
VCTree+Ours	10.1	13.1	15.4	16.9	19.6	20.6	25.7	29.8	31.4
BGNN [18]	-	10.7	12.6	-	14.3	16.5	-	30.4	32.9
BGNN + Ours	11.1	14.2	16.4	15.9	17.4	18.6	27.3	31.6	34.1

conditional biases, our method can effectively enhance the model performance. Moreover, we also compare our method to other debiasing methods in SGG, including two representative methods (Reweight [33] and TDE [33]) and a recently proposed one (DLFE [4]). For Reweight, we follow the idea in [33]. These methods use the same backbone (VidVRD-II [28]) with ours. The results in Table 1 show that compared to these methods, our method achieves superior performance, demonstrating that by considering the spatio-temporal structure of VidSGG, our method can better handle the biases in this task.

On AG dataset, as shown in Table 4 and Table 5, our method outperforms other methods on all metrics. Our method also outperforms other debiasing methods [33, 4] that use the same backbone (TRACE [36]) with ours. Moreover, note that the metric of Mean Recall is designed to measure the model performance considering the imbalanced data distribution [33, 36], and our method achieves more performance improvements on this metric, demonstrating that our framework can effectively mitigate the spatio-temporal conditional bias problem caused by biased data distribution in the dataset.

4.3 Ablation Studies

We conduct extensive ablation experiments to evaluate our framework on VidVRD dataset.

Impact of different backbone networks. To validate the general effectiveness of our framework, we apply it on different models [20, 28], and obtain

Table 4. Recall (%) of various models on AG dataset following the setting in [36].

Top k Predictions for Each Pair	Method	SGDet				SGCls				PredCls			
		image		video		image		video		image		video	
		R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50
k=7	Freq Prior [45]	34.41	44.34	32.50	41.11	45.10	48.87	44.47	46.39	87.95	93.02	86.01	88.59
	G-RCNN [42]	34.28	44.47	32.60	41.29	45.57	49.75	45.11	47.22	88.73	93.73	86.28	88.93
	RelDN [46]	34.92	45.27	33.18	42.10	46.47	50.31	45.87	47.78	90.89	96.09	88.77	91.43
	TRACE [36]	35.09	45.34	33.38	42.18	46.66	50.46	46.03	47.92	91.60	96.35	89.31	91.72
	TRACE [36] + Reweight [33]	35.15	45.37	33.42	42.24	46.68	50.50	46.07	47.94	91.61	96.35	89.32	91.74
	TRACE [36] + TDE [33]	35.20	45.41	33.49	42.30	46.71	50.55	46.12	48.00	91.63	96.36	89.32	91.76
	TRACE [36] + DLFE [4]	35.29	45.47	33.58	42.41	46.75	50.63	46.18	48.04	91.64	96.36	89.35	91.77
	Ours	**36.59**	**47.00**	**34.88**	**43.81**	**47.40**	**51.06**	**46.71**	**48.56**	**91.74**	**96.43**	**89.44**	**91.85**
k=6	Freq Prior [45]	34.47	43.69	32.38	40.24	44.90	47.15	43.57	44.63	85.89	89.43	83.33	84.99
	G-RCNN [42]	34.60	43.98	32.75	40.65	45.82	48.31	44.60	45.77	87.03	90.60	84.02	85.74
	RelDN [46]	35.22	44.94	33.39	41.64	46.76	49.11	45.48	46.57	89.63	93.56	87.01	88.86
	TRACE [36]	35.41	45.06	33.59	41.76	47.00	49.32	45.71	46.79	90.34	93.94	87.56	89.24
	TRACE [36] + Reweight [33]	35.44	45.10	33.64	41.83	47.01	49.35	45.73	46.82	90.36	93.95	87.58	89.27
	TRACE [36] + TDE [33]	35.49	45.16	33.68	41.90	47.04	49.36	45.76	46.89	90.37	93.96	87.61	89.27
	TRACE [36] + DLFE [4]	35.56	45.28	33.76	41.99	47.08	49.41	45.83	46.92	90.39	93.99	87.65	89.29
	Ours	**36.80**	**46.73**	**34.99**	**43.39**	**47.66**	**49.96**	**46.41**	**47.47**	**90.49**	**94.11**	**87.78**	**89.50**

Table 5. Mean Recall (%) and Average Precision (%) of various models on AG dataset following the setting in [36].

Method	SGDet				SGCls				PredCls			
	Mean Recall		Average Precision		Mean Recall		Average Precision		Mean Recall		Average Precision	
	@20	@50	mAP_r	$wmAP_r$	@20	@50	mAP_r	$wmAP_r$	@20	@50	mAP_r	$wmAP_r$
Freq Prior [45]	24.89	34.07	9.45	15.58	34.30	36.96	14.29	22.68	55.17	63.67	33.10	65.92
G-RCNN [42]	27.79	34.99	11.76	15.90	36.19	38.29	17.64	22.53	56.32	61.31	41.21	70.89
RelDN [46]	30.39	39.53	12.93	15.94	39.92	41.93	20.07	23.88	59.81	63.47	50.08	72.26
TRACE [36]	30.84	40.12	13.43	16.56	41.19	43.21	20.71	24.61	61.80	65.37	53.27	75.45
TRACE [36] + Reweight [33]	30.87	40.21	13.44	16.59	41.31	43.44	20.75	24.63	61.97	65.77	53.30	75.46
TRACE [36] + TDE [33]	31.01	40.40	13.47	16.60	41.56	43.70	20.79	24.66	62.12	65.89	53.34	75.50
TRACE [36] + DLFE [4]	31.24	40.75	13.48	16.62	41.77	43.98	20.83	24.70	62.44	66.31	53.35	75.52
Ours	**32.43**	**43.13**	**14.00**	**17.47**	**43.43**	**47.26**	**21.25**	**25.32**	**67.67**	**75.72**	**53.88**	**75.96**

consistent performance improvement as shown in Table 2, showing our framework can be flexibly applied on various models to improve their performance.

Impact of spatio-temporal conditional biases. To investigate the impact of spatial and temporal conditional biases on model performance, we evaluate the following variants. For spatial conditional biases, we test 4 variants.

Specifically, one model variant (*w/o Spatial Level (all)*) ignores *all* groups of spatial conditional biases and handles only temporal conditional biases, i.e., optimizing the model without the query sets for handling *all* types of spatial conditional biases. Moreover, as discussed in 3.2, we have 3 groups of spatial conditional biases, i.e., the conditional bias between *predicate*/*subject*/*object* and their corresponding spatial contexts. Thus to explore the impact of each group of conditional biases, we correspondingly implement 3 variants (*w/o Predicate-centered*, *w/o Subject-centered* and *w/o Object-centered*), and each variant ignores the corresponding group of query sets during the model training.

Table 6. We evaluate various variants to investigate the impact of each group of spatio-temporal conditional biases.

Method	Relation Detection			Relation Tagging		
	mAP	R@50	R@100	P@1	P@5	P@10
Baseline (VidVRD-II)	29.37	19.63	22.92	70.40	53.88	40.16
w/o Spatial Level (all)	30.49	20.35	23.61	75.50	55.30	41.50
w/o Predicate-centered	30.98	20.67	23.94	76.50	56.40	42.35
w/o Subject-centered	31.10	20.87	24.04	78.00	56.80	42.75
w/o Object-centered	31.05	20.80	23.98	77.50	56.60	42.60
w/o Temporal Level (all)	30.47	20.48	23.63	75.00	55.60	41.70
w/o Forward Case	30.94	20.78	23.95	76.50	56.50	42.60
w/o Backward Case	31.00	20.75	24.01	77.50	56.80	42.45
Ours	**31.57**	**21.16**	**24.57**	**79.00**	**57.60**	**43.20**

Similarly, for temporal conditional biases, we have 3 variants. One model variant (*w/o Temporal Level (all)*) ignores *all* groups of temporal conditional biases, and handles only spatial conditional biases. Furthermore, since we have 2 groups of temporal conditional biases, i.e., the conditional bias between the current triplet and the triplets appear before or appear after, we evaluate 2 more variants (*w/o Forward Case* and *w/o Backward Case*).

As shown in Table 6, ignoring any group of conditional biases would lead to performance drop compared to our framework, showing that each group of conditional biases can affect the model performance. More ablation study and qualitative results are in our supplementary.

4.4 Experiments on Image SGG

Besides the VidSGG task, there can also exist spatial conditional biases in the task of image SGG. Thus if we remove the query sets for handling the temporal conditional biases, our framework can then be adapted to handle the image SGG task. Therefore, we also evaluate our method on the widely used SGG benchmark: Visual Genome [14], by constructing and incorporating only the query sets for handling the spatial conditional biases. As shown in Table 3, we apply our framework on different SGG models [18, 34], and consistently enhance their performances. Besides, as shown in Table 7, based on the same backbone (VCTree [34]), our method achieves better performance than other debiasing strategies.

Table 7. Experiment results of ours and other debiasing methods in image SGG.

Method	SGGen		SGCls		PredCls	
	mR@20	mR@50	mR@20	mR@50	mR@20	mR@50
VCTree [34]	5.2	7.1	9.1	11.3	14.1	17.7
VCTree+Reweight [33]	6.6	8.7	10.6	12.5	16.3	19.4
VCTree+TDE [33]	6.8	9.5	11.2	15.2	19.2	26.2
VCTree+DLFE [4]	8.6	11.8	15.8	18.9	20.8	25.3
VCTree+Ours	13.1	15.4	19.6	20.6	29.8	31.4

5 Conclusion

To address the spatio-temporal conditional bias problem in VidSGG, we propose a novel Meta Video Scene Graph Generation (**MVSGG**) framework. By constructing a support set and various query sets w.r.t. various types of conditional biases, and optimizing the model on these constructed sets via *meta training and testing*, our framework can effectively train the model to handle various types of conditional biases. Moreover, our framework is general, and can be flexibly applied to various models. Our framework achieves superior performance.

Acknowledgement

This work is supported by National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-100E-2020-065), Ministry of Education Tier 1 Grant and SUTD Startup Research Grant.

References

1. Bai, Y., Jiao, J., Ce, W., Liu, J., Lou, Y., Feng, X., Duan, L.Y.: Person30k: A dual-meta generalization network for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2123–2132 (2021)
2. Chen, S., Shi, Z., Mettes, P., Snoek, C.G.: Social fabric: Tubelet compositions for video relation detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13485–13494 (2021)
3. Chen, T., Yu, W., Chen, R., Lin, L.: Knowledge-embedded routing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6163–6171 (2019)
4. Chiou, M.J., Ding, H., Yan, H., Wang, C., Zimmermann, R., Feng, J.: Recovering the unbiased scene graphs from the biased ones. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 1581–1590 (2021)
5. Cong, Y., Liao, W., Ackermann, H., Rosenhahn, B., Yang, M.Y.: Spatial-temporal transformer for dynamic scene graph generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16372–16382 (2021)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning. pp. 1126–1135. PMLR (2017)
7. Gao, J., Sun, C., Yang, Z., Nevatia, R.: Tall: Temporal activity localization via language query. In: Proceedings of the IEEE international conference on computer vision. pp. 5267–5275 (2017)
8. Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., Saenko, K.: Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 2712–2719 (2013)
9. Guo, J., Zhu, X., Zhao, C., Cao, D., Lei, Z., Li, S.Z.: Learning meta face recognition in unseen domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6163–6172 (2020)
10. Guo, Y., Gao, L., Wang, X., Hu, Y., Xu, X., Lu, X., Shen, H.T., Song, J.: From general to specific: Informative scene graph generation via balance adjustment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16383–16392 (2021)
11. Huang, C., Cao, Z., Wang, Y., Wang, J., Long, M.: Metasets: Meta-learning on point sets for generalizable representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8863–8872 (2021)
12. Ji, J., Krishna, R., Fei-Fei, L., Niebles, J.C.: Action genome: Actions as compositions of spatio-temporal scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10236–10247 (2020)
13. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3668–3678 (2015)
14. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123**(1), 32–73 (2017)
15. Lei, J., Yu, L., Bansal, M., Berg, T.L.: Tvqa: Localized, compositional video question answering. arXiv preprint arXiv:1809.01696 (2018)

16. Lei, J., Yu, L., Berg, T.L., Bansal, M.: Tvr: A large-scale dataset for video-subtitle moment retrieval. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16. pp. 447–463. Springer (2020)
17. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.: Learning to generalize: Meta-learning for domain generalization. Proceedings of the AAAI Conference on Artificial Intelligence **32** (Apr 2018)
18. Li, R., Zhang, S., Wan, B., He, X.: Bipartite graph network with adaptive message passing for unbiased scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11109–11119 (2021)
19. Li, Y., Yang, X., Shang, X., Chua, T.S.: Interventional video relation detection. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 4091–4099 (2021)
20. Liu, C., Jin, Y., Xu, K., Gong, G., Mu, Y.: 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. pp. 10840–10849 (2020)
21. Liu, H., Yan, N., Mortazavi, M., Bhanu, B.: Fully convolutional scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11546–11556 (2021)
22. Lu, Y., Rai, H., Chang, J., Knyazev, B., Yu, G., Shekhar, S., Taylor, G.W., Volkovs, M.: Context-aware scene graph generation with seq2seq transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15931–15941 (2021)
23. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)
24. Qian, X., Zhuang, Y., Li, Y., Xiao, S., Pu, S., Xiao, J.: Video relation detection with spatio-temporal graph. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 84–93 (2019)
25. Qu, H., Li, Y., Foo, L.G., Kuen, J., Gu, J., Liu, J.: Improving the reliability for confidence estimation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
26. Rajeswaran, A., Finn, C., Kakade, S.M., Levine, S.: Meta-learning with implicit gradients. Advances in neural information processing systems **32** (2019)
27. Seo, S., Lee, J.Y., Han, B.: Information-theoretic bias reduction via causal view of spurious correlation. arXiv preprint arXiv:2201.03121 (2022)
28. Shang, X., Li, Y., Xiao, J., Li, W., Chua, T.S.: Video visual relation detection via iterative inference. In: Proceedings of the 29th ACM international conference on Multimedia (2021)
29. Shang, X., Ren, T., Guo, J., Zhang, H., Chua, T.S.: Video visual relation detection. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1300–1308 (2017)
30. Su, Z., Shang, X., Chen, J., Jiang, Y.G., Qiu, Z., Chua, T.S.: Video relation detection via multiple hypothesis association. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 3127–3135 (2020)
31. Suhail, M., Mittal, A., Siddiquie, B., Broaddus, C., Eledath, J., Medioni, G., Sigal, L.: Energy-based learning for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13936–13945 (2021)

32. Sun, Q., Liu, Y., Chua, T.S., Schiele, B.: Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 403–412 (2019)
33. Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3716–3725 (2020)
34. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6619–6628 (2019)
35. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4631–4640 (2016)
36. Teng, Y., Wang, L., Li, Z., Wu, G.: Target adaptive context aggregation for video scene graph generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13688–13697 (2021)
37. Tsai, Y.H.H., Divvala, S., Morency, L.P., Salakhutdinov, R., Farhadi, A.: Video relationship reasoning using gated spatio-temporal energy graph. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10424–10433 (2019)
38. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence-video to text. In: Proceedings of the IEEE international conference on computer vision. pp. 4534–4542 (2015)
39. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)
40. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5410–5419 (2017)
41. Yang, G., Zhang, J., Zhang, Y., Wu, B., Yang, Y.: Probabilistic modeling of semantic ambiguity for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12527–12536 (2021)
42. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph r-cnn for scene graph generation. In: Proceedings of the European conference on computer vision (ECCV). pp. 670–685 (2018)
43. Ye, N., Tang, J., Deng, H., Zhou, X.Y., Li, Q., Li, Z., Yang, G.Z., Zhu, Z.: Adversarial invariant learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12441–12449. IEEE (2021)
44. Yu, J., Chai, Y., Hu, Y., Wu, Q.: Cogtree: Cognition tree loss for unbiased scene graph generation. arXiv preprint arXiv:2009.07526 (2020)
45. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5831–5840 (2018)
46. Zhang, J., Shih, K.J., Elgammal, A., Tao, A., Catanzaro, B.: Graphical contrastive losses for scene graph parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11535–11543 (2019)