COMPOSITIONAL PROMPT TUNING WITH MOTION CUES FOR OPEN-VOCABULARY VIDEO RELATION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Prompt tuning with large-scale pretrained vision-language models empowers open-vocabulary prediction trained on limited base categories, *e.g.*, object classification and detection. In this paper, we propose *compositional* prompt tuning with *motion cues*: an extended prompt tuning paradigm for compositional predictions of video data. In particular, we present Relation Prompt (RePro) for Open-vocabulary Video Visual Relation Detection (Open-VidVRD), where conventional prompt tuning is easily biased to certain subject-object combinations and motion patterns. To this end, RePro addresses the two technical challenges of Open-VidVRD: 1) the prompt tokens should respect the two different semantic roles of subject and object, and 2) the tuning should account for the diverse spatio-temporal motion patterns of the subject-object compositions. Our RePro achieves a new state-of-the-art performance on two VidVRD benchmarks of not only the base training object and predicate categories, but also the unseen ones. Extensive ablations also demonstrate the effectiveness of the proposed compositional and multi-mode design of prompt.

1 INTRODUCTION

Video visual relation detection (VidVRD) aims to detect the visual relationships between object tracklets in videos as ⟨subject, predicate, object⟩ triplets (Shang et al., 2017; Chen et al., 2021; Gao et al., 2022), *e.g.*, `dog-towards-child` shown in Figure 1. Compared to its counterpart in still images (Chen et al., 2019), due to extra temporal axis, there are usually multiple relationships with different temporal scales, and a subject-object pair can have several predicates with ambiguous boundaries. For example, as shown in Figure 1, the action `feed` of `child` to `dog` co-occurs with several other predicates (*e.g.*, `away`, `towards`). This characteristic makes VidVRD have more plentiful and diverse relations between objects than its image counterpart. As a result, it is impractical to collect sufficient annotations for all categories for VidVRD. Therefore, to make VidVRD practical, we should know how to generalize the model, trained on limited annotations, to *new* object and predicate classes *unseen* in training data.

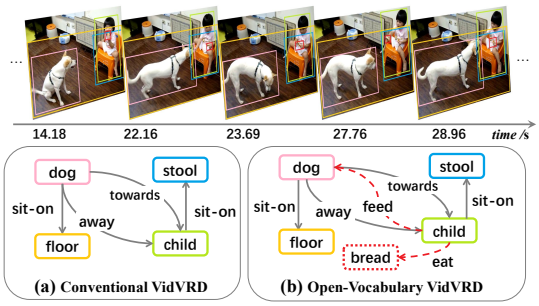


Figure 1: Examples of VidVRD. The relation graphs are w.r.t the whole video clip. Dashed line denotes *new* categories unseen in the training data.

To this end, we propose a new task: *Open-vocabulary* VidVRD (Open-VidVRD). In particular, "open" does not only mean unseen relationship combinations, *e.g.*, `dog-sit_on-floor`, but also unseen objects and predicates, *e.g.*, `bread` and `feed`, as shown in Figure 1. Recent works on such generalization only focus on the unseen combinations (Chen et al., 2021; Shang et al., 2021) in VidVRD, or zero-shot transfer among semantically related objects in zero-shot object detection (Huang et al., 2022), *e.g.*, the seen `dog` class can help to recognize the unseen `wolf`. However, they fail

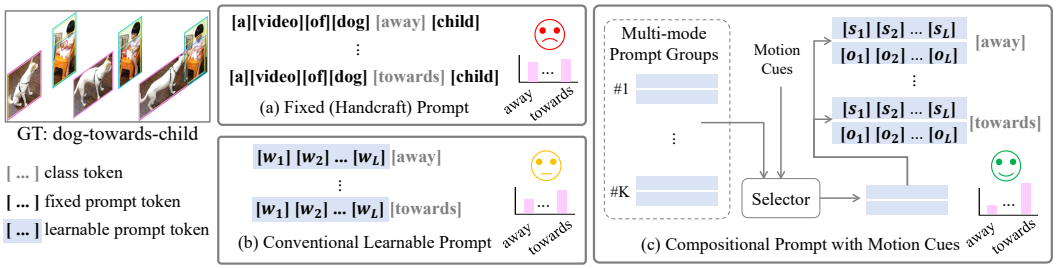


Figure 2: Comparisons of different prompt tuning methods for Open-VidVRD.

to generalize to the categories totally unrelated to the limited seen ones, where the transfer gap is unbridgeable, *e.g.*, `bread` in testing has no visual similarity with `dog` and `child` in training.

Thanks to the encyclopedic knowledge acquired by large vision-language models (VLM) pre-trained on big data (Radford et al., 2021; Li et al., 2022), we can achieve open-vocabulary relation detection with only training data of limited base categories. To bridge the gap between the pre-trained and downstream tasks without extra fine-tuning the whole VLM model, a trending technique named **prompt tuning** is widely adopted (Liu et al., 2021; Jin et al., 2022; Zhou et al., 2022b). For example, we can achieve zero-shot relation classification for the tracklets pair in Figure 2. We first crop the object tracklet regions in the video, and feed them into the visual encoder of VLM to obtain corresponding visual embeddings. Then we use a simple prompt like "a video of [CLASS]" , feed it to the VLM's text encoder to obtain the text embedding, and classify the object based on the similarities between visual and text embeddings. Based on the tracklet classification results, for the example of the pair `dog` and `child`, we can craft a prompt like "a video of dog [CLASS] child", as shown in Figure 2(a), and similarly classify their predicates based on the predicate text embeddings. Furthermore, we can replace the fixed prompt tokens with learnable continuous tokens, as shown in Figure 2(b), known as prompt representation learning, which has been widely applied to many vision tasks, *e.g.*, image classification (Zhou et al., 2022b;a) and open-vocabulary object detection (Gu et al., 2021; Du et al., 2022; Ma et al., 2022).

Learning the prompt representation is actually introducing some priors for describing the context of target classes, and it excludes some impossible classes with the constraint of the context. However, the prompt representation (either handcrafted or learned) in above approaches is monotonous and static, and learning the prompt sometimes might break the "open" knowledge due to overfitting to the base category training data. Modeling the prompt representation for video visual relations has some specific characteristics need to be considered:

- **Compositional**: The prompt context for predicates is highly related to the semantic roles of subject and object. A holistic prompt representation might be sub-optimal for predicates. For example, as shown in Figure 1, even the same predicate (`sit_on`) in different relation triplets (`dog-sit_on-floor` and `child-sit_on-stool`) have totally different visual context.
- **Motion-related**: Predicates with different motion patterns naturally should be prompted with different context tokens. The naive prompt representation fails to consider the spatio-temporal motion cues of tracklet pairs. For example, the predicate `towards` shown in Figure 1 can be prompted as "a relation of [CLASS], moving closer". In contrast, `eat` and `sit_on` can be prompted as "a relation of [CLASS], relative static".

In this paper, we propose a compositional and motion-based **Re**lation **Pro**mpt learning framework: RePro, as shown in Figure 2(c). To deal with the **compositional** characteristic of relation prompt, we set compositional prompt representations specified with subject and object respectively. Thanks to the subject-object compositions, we can model the prompt context w.r.t. semantic roles (*i.e.*, `subject` or `object`). For example, a possible prompt can be "sth. doing [CLASS]" for subject and "sth. being [CLASS]" for object. To consider the **motion-related** context of predicates, we design multi-mode prompt groups, as shown in Figure 2(c), where multi-mode means *with multiple motion patterns*. Each group has the compositional prompts for subject and object, and we select a proper group according to the motion cues (patterns) in the subject-object tracklet pairs (cf. Sec. 3.3). Compared to some prompt tuning works which focus on category-based context (Zhou et al., 2022b) or instance-conditioned context (Zhou et al., 2022a), our motion-cue-based grouping

has better cross-category generalization ability, and can avoid the over-fitting to base categories. We evaluate our RePro on the VidVRD (Shang et al., 2017) and VidOR (Shang et al., 2019) benchmarks. Our experiment results show that RePro trained with only the samples of base relation categories has a good generalizability to detect novel relations, and achieves the new state-of-the-art. For example, it outperforms the top-performing method, *i.e.*, VidVRD-II (Shang et al., 2021), by 2.54 and 3.91 percentage points of absolute mAP for SGDet and SGCls, respectively.

Our contributions in this paper are thus three-fold. 1) A new open-vocabulary setting for video visual relation detection task, *i.e.*, Open-VidVRD. 2) A compositional prompt representation learning method that models the prompt contexts for subject and object separately. 3) A motion-cue-based multi-mode prompt groups that achieves a strong generalization ability.

2 Related Works

Video visual relation detection (VidVRD) was defined in Shang et al. (2017; 2019) together with the proposals of the VidVRD and VidOR benchmarks. The task aims to spatio-temporally localize visual relations between object tracklets. Existing methods mainly focus on modeling better visual or spatio-temporal contexts (Qian et al., 2019; Shang et al., 2021), and detecting visual relations with more granularity either by sliding windows (Liu et al., 2020) or temporal grounding (Gao et al., 2022). They mainly worked on the pre-defined visual relations with closed sets of object and predicate categories (in both training and testing phases), and only achieve zero-shot transfer on the unseen combinations, while ignore the model's generalizability to unseen categories. Different from them, our work is the first one to research on the open-vocabulary VidVRD setting. The aim is to generalize the model to recognize the categories totally unrelated to the seen (training) ones.

Prompt tuning for open-vocabulary visual recognition. Prompt tuning (Liu et al., 2021) has been widely adopted in zero-shot and few-shot image classification (Zhou et al., 2022b;a), and open-vocabulary object detection (OV-Det) (Gu et al., 2021; Du et al., 2022; Ma et al., 2022). For image classification, existing works are focused on learning the prompt conditioned on different image categories (Zhou et al., 2022b), or different image instances (Zhou et al., 2022a). For OV-Det, recent works mainly focus on knowledge distillation from VLM and simply using handcrafted prompt (Gu et al., 2021; Ma et al., 2022; Rasheed et al., 2022), or focus on prompt representation learning for object regions (Du et al., 2022; Feng et al., 2022). Compared to existing works which focus on prompt tuning for static image or objects, we are the first one to model prompt representation for dynamic visual relations. There is also a recent work (He et al., 2022) using prompt tuning for visual relations in images. However, their prompt representations are learned from the visual cues of static subject and object. In contrast, our RePro is to learn the compositional prompt by leveraging the motion cues in subject and object tracklets. It is expected to perform better than static prompts for detecting the dynamic visual relations in videos.

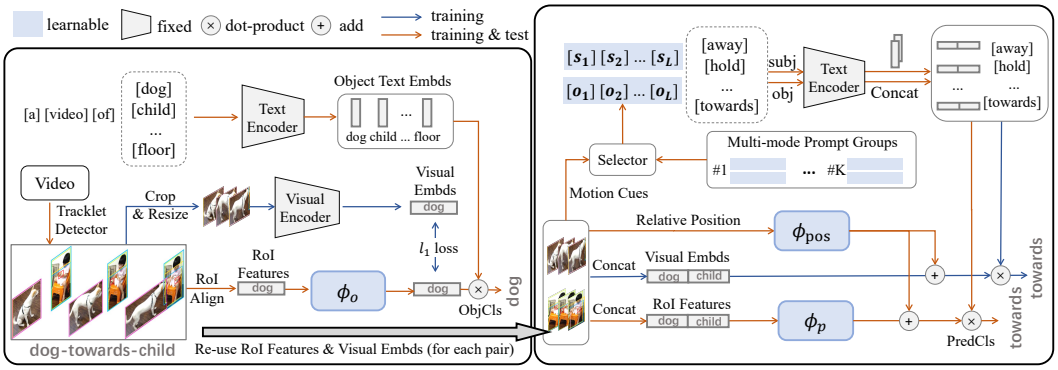
3 Method

We build the Open-Vocabulary setting for VidVRD as follows. We first divide the object and predicate categories of an existing dataset into *base* and *novel* splits. Specifically, we denote $\mathcal{C}_b^O$ and $\mathcal{C}_n^O$ as the sets of *base* and *novel* object categories, respectively. We use $\mathcal{C}_b^P$ and $\mathcal{C}_n^P$ to denote the sets of *base* and *novel* predicate categories, respectively. In the training stage, we use all visual relation triplet samples from $\mathcal{C}_b^O \times \mathcal{C}_b^P \times \mathcal{C}_b^O$ to train the model. In the testing stage, we evaluate the model with the triplets sampled from all categories, *i.e.*, $\mathcal{C}_b^O \cup \mathcal{C}_n^O$ and $\mathcal{C}_b^P \cup \mathcal{C}_n^P$.

Based on this new setting, we first briefly introduce the preliminaries for open-vocabulary classification with pre-trained VLMs (Sec. 3.1). Then, we introduce the proposed Open-VidVRD method RePro, as illustrated in Figure 3, in which we first extend open-vocabulary object detection methods (Gu et al., 2021; Du et al., 2022) to open-vocabulary tracklet detection (Sec. 3.2), and then perform open-vocabulary relation classification for each tracklet pair (Sec. 3.3).

3.1 Preliminaries: Open-Vocabulary Classification with Prompt

Fixed Prompt. Pre-trained VLMs have a strong open-vocabulary classification ability (Li et al., 2022). They first extract text embeddings for all categories by feeding handcrafted prompt (*e.g.*, "a



(a) Tracklet Detection (b) Relation Classification

Figure 3: The overall pipeline of RePro. (a) We first detect the object tracklet regions and classify them by training a V2L projector ϕ_o based on visual embeddings and RoI Aligned features. (b) For each tracklet pair, we learn the compositional prompt group based on the motion cues in the tracklet pair, and classify their predicates by learning a position projector ϕ_{pos} and a V2L projector ϕ_p based on the concatenated embeddings and features. We show only one predicate in the figure for clarity.

video of [CLASS]") into the text encoder of VLM, where [CLASS] can be replaced with the class name of an arbitrary object or predicate. Their output text embedding $\boldsymbol{t}_c \in \mathbb{R}^d$ for each class c is

$$\boldsymbol{t}_c = \text{VLM}_{\text{txt}}(\boldsymbol{W}_c),\ \boldsymbol{W}_c = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_L, \tilde{\boldsymbol{w}}_c], \forall c \in \mathcal{C}_b^O \cup \mathcal{C}_n^O \text{ or } \mathcal{C}_b^P \cup \mathcal{C}_n^P, \tag{1}$$

where $\boldsymbol{W}_c$ is the prompt representation with L context token vectors and the class token vector $\tilde{\boldsymbol{w}}_c$. Then, for each object tracklet with cropped video region, the corresponding visual embedding can be extracted by the visual encoder of VLM, denoted as $\boldsymbol{v}_i^o \in \mathbb{R}^d$. Similarly, the visual embedding of tracklet pair can also be extracted (*e.g.*, based on the union region), denoted as $\boldsymbol{v}_i^p$. Therefore, the i-th region (generally denoted as $\boldsymbol{v}_i$) can be classified by the cosine similarities w.r.t $\{\boldsymbol{t}_c\}$:

$$\hat{c}_i = \arg\max_c \cos(\boldsymbol{v}_i, \boldsymbol{t}_c),\ \forall c \in \mathcal{C}_b^O \cup \mathcal{C}_n^O \text{ or } \mathcal{C}_b^P \cup \mathcal{C}_n^P,\ \text{where } \cos(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}/(\|\boldsymbol{x}\|\|\boldsymbol{y}\|). \tag{2}$$

Learnable Prompt. Manually tuning the words in the prompt requires domain expertise and is time-consuming or not robust (Radford et al., 2021). The substitute method is to learn the prompt representations from the training data (Zhou et al., 2022b;a). Specifically, the context vector $\boldsymbol{w}_i$ in $\boldsymbol{W}_c$ can be set as a learnable vector while $\tilde{\boldsymbol{w}}_c$ is kept as fixed. In the training stage, samples and $\{\tilde{\boldsymbol{w}}_c\}$ are from base categories. In the testing stage, the L learned vectors in each $\boldsymbol{W}_c$ are fixed and then the model performs classification in the same way as in Eq. (2).

3.2 Open-Vocabulary Object Tracklet Detection

Tracklet Proposal Generation. Given a video, we first detect all the class-agnostic object tracklets using a pre-trained tracklet detector, denote as $\mathcal{T} = \{T_i\}_{i=1}^N$, as shown in Figure 3(a). Specifically, each tracklet T_i is characterized with a bounding box sequence $\boldsymbol{b}_i \in \mathbb{R}^{l_i \times 4}$ and the corresponding RoI Aligned (He et al., 2017) visual feature $\boldsymbol{f}_i^0 \in \mathbb{R}^{l_i \times 2048}$, where l_i is the length (number of frames) of T_i. To reduce the computational over-head, we average the RoI feature along the temporal axis following (Shang et al., 2021), and denote it as $\boldsymbol{f}_i \in \mathbb{R}^{2048}$.

Tracklet Classification. Instead of directly classifying object tracklets using VLM as Eq. (2), we train a visual-to-language (V2L) projection module $\phi_o(\cdot)$ to further utilize the annotations of base classes. In particular, $\phi_o(\cdot)$ maps the RoI Aligned feature $\boldsymbol{f}_i$ of each tracklet to the same semantic space $\mathbb{R}^d$, *i.e.*, $\boldsymbol{v}_i^{o\prime} = \phi_o(\boldsymbol{f}_i)$. Let $\boldsymbol{t}_c^o$ be the text embedding of object class $c \in \mathcal{C}_b^O$. The probability of tracklet T_i being classified as class c can be calculated as

$$p_i(c) = \frac{\exp(\cos(\boldsymbol{v}_i^{o\prime}, \boldsymbol{t}_c^o)/\tau)}{\sum_{c' \in \mathcal{C}_b^O} \exp(\cos(\boldsymbol{v}_i^{o\prime}, \boldsymbol{t}_{c'}^o)/\tau)}, \forall c \in \mathcal{C}_b^O, \tag{3}$$

where τ is a temperature parameter for softmax.

Training Objectives. To train the object tracklet classification module, we assign base category labels to detected tracklets according to the IoU w.r.t ground-truth tracklets. We call those tracklets

with assigned labels as positive tracklets, otherwise negative tracklets. Note that two cases can be recognized as negative tracklets: 1) the content is background; and 2) the content contains a novel object category. For these negative tracklets, we follow the loss used by Du et al. (2022) that forces the prediction (from any negative tracklet) on each base class to be $1/|\mathcal{C}_b^O|$, *i.e.*, unlike any base category. Therefore, the classification loss for positive and negative tracklets can be calculated as

$$\mathcal{L}_{\text{cls-pos}} = -\frac{1}{|\mathcal{T}_p|}\sum_{T_i\in\mathcal{T}_p}\sum_{c\in\mathcal{C}_b^O}\mathbb{1}_{\{c=c_i^*\}}\log p_i(c), \quad \mathcal{L}_{\text{cls-neg}} = -\frac{1}{|\mathcal{T}_n|}\sum_{T_i\in\mathcal{T}_n}\sum_{c\in\mathcal{C}_b^O}\frac{1}{|\mathcal{C}_b^O|}\log p_i(c), \quad (4)$$

where $\mathcal{T}_p$ and $\mathcal{T}_n$ are the sets of positive and negative tracklets, respectively (*i.e.*, $\mathcal{T}_p \cup \mathcal{T}_n = \mathcal{T}$), and c_i^* is the ground-truth label for the i-th positive tracklet. We empirically found (in Sec. 3.2) that using above negative tracklet loss works better than using the loss with a unique "background" class (Zareian et al., 2021; Gu et al., 2021). Besides, following (Gu et al., 2021), we distil the knowledge from a pre-trained visual encoder to $\phi_o(\cdot)$ by aligning $\boldsymbol{v}_i^{o\prime}$ to $\boldsymbol{v}_i^o$ using l_1 loss, *i.e.*,

$$\mathcal{L}_{\text{distil}} = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{v}_i^{o\prime} - \boldsymbol{v}_i^o\|_1 \quad (5)$$

Therefore, the overall loss for object tracklet classification is $\mathcal{L}_{\text{cls}} = \mathcal{L}_{\text{cls-pos}} + \mathcal{L}_{\text{cls-neg}} + \lambda\mathcal{L}_{\text{distil}}$, where λ is a hyper-parameter to weight the classification and distillation.

3.3 Open-Vocabulary Visual Relation Classification

Based on the classified object tracklets, we perform open-vocabulary relation classification for each tracklet pair, as shown in Figure 3(b). First, we learn the prompt representations based on the pre-extracted visual embeddings of tracklet pairs, for which we introduce the *compositional prompt representations* and the *motion-based prompt groups*. Second, we utilize the pre-extracted RoI Aligned features to train a visual-to-language (V2L) projection module based on the learned prompt representations. Finally for testing, we extract all predicate text embeddings and classify the predicates of each tracklet pair by using the RoI Aligned features and the trained V2L projection module.

Compositional Prompt Representations. The compositional prompt consists learnable prompt representations $\boldsymbol{S}_c$ and $\boldsymbol{O}_c$ (of predicate class c) for subject and object, respectively:

$$\boldsymbol{S}_c = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L, \tilde{\boldsymbol{w}}_c], \quad \boldsymbol{O}_c = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_L, \tilde{\boldsymbol{w}}_c], \quad (6)$$

where $\boldsymbol{s}_i$ and $\boldsymbol{o}_i$ is the learnable context vector and $\tilde{\boldsymbol{w}}_c$ is the fixed class token for predicate c (for $c \in \mathcal{C}_b^P$ in training phase and for all c in testing phase). Then, the predicate text embedding $\boldsymbol{t}_c^p$ is generated by concatenating the two outputs of VLM given two prompts (respectively) as inputs, *i.e.*,

$$\boldsymbol{t}_c^p = [\text{VLM}_{\text{txt}}(\boldsymbol{S}_c), \text{VLM}_{\text{txt}}(\boldsymbol{O}_c)], \text{ and } \boldsymbol{t}_c^p \in \mathbb{R}^{2d}. \quad (7)$$

Motion-based Prompt Groups. We vary the prompt contexts based on the motion cues, *i.e.*, the relative spatio-temporal motion patterns, between each pair of subject and object. In specific, we take the generalized IoU Rezatofighi et al. (2019) (*i.e.*, GIoU) as the metric to calculate the motion patterns. For each tracklet pair $\langle T_i, T_j\rangle$, we use a vector to represent a motion pattern:

$$\boldsymbol{m}_{i,j} = \text{sign}([G_{i,j}^s - \gamma, G_{i,j}^e - \gamma, G_{i,j}^e - G_{i,j}^s]), \text{ and } \boldsymbol{m}_{i,j} \in \{+,-\}^3, \quad (8)$$

where $G_{i,j}^s, G_{i,j}^e$ are the GIoU between subject-object for the start and end bounding boxes of their temporal intersection, respectively, and γ is a threshold for GIoU. This definition considers two perspectives: 1) whether the two tracklets are near or far (*i.e.*, the first two terms of Eq. (8)), and 2) whether they move toward or away to each other (*i.e.*, the third term of Eq. (8). Overall, we have 6 motion patterns (cf. Sec. A.2 for more details), and build 6 prompt groups correspondingly. Each group consists its own compositional prompt representations $\boldsymbol{S}_c$ and $\boldsymbol{O}_c$ as defined in Eq. (6).

It's worth noting that we aim to build a framework of learning motion-based multi-mode prompts. The used GIoU-based approach (in our framework) is a simple and intuitive way to calculating motion cues. This approach is not perfect, *e.g.*, it is poor to capture the motion pattern of tracklets moving back and forth. We leave other fancier (motion capturing) approaches as future work.

Training Objectives. Based on the above definition, we train the prompt representations with the visual embeddings and relative position features. For simplicity, we show the training process of

a single group (and in the end, we derive the final loss by averaging the losses across all groups). For each tracklet pair $\langle T_i, T_j \rangle$, we first calculate its motion cue, select the corresponding prompt group, and extract the class text embeddings $\boldsymbol{t}_c^p$ for each predicate class $c \in \mathcal{C}_b^P$. Then, we take the pre-extracted visual embeddings $\boldsymbol{v}_i^o$ and $\boldsymbol{v}_j^o$, and concatenate them as the pair's visual embedding $\boldsymbol{v}_{i,j}^p = [\boldsymbol{v}_i^o, \boldsymbol{v}_j^o] \in \mathbb{R}^{2d}$. Following Shang et al. (2021), we additionally compute the relative position feature between bounding boxes of T_i and T_j, denoted as $\boldsymbol{f}_{i,j}^s \in \mathbb{R}^{12}$ (cf. Sec. A.2 in the Appendix for more details). The predicted probability of predicate class c in this tracklet pair is thus

$$p_{i,j}(c) = \text{Sigmoid}(\cos(\boldsymbol{v}_{i,j}^p + \phi_{\text{pos}}(\boldsymbol{f}_{i,j}^s), \boldsymbol{t}_c^p)), \ \forall c \in \mathcal{C}_b^P. \tag{9}$$

where ϕ_{pos} projects $\boldsymbol{f}_{i,j}^s$ to the same dimension as $\boldsymbol{v}_{i,j}^p$. Going through all *base* classes (in $\mathcal{C}_b^P$), the probability vector of tracklet pair $\langle T_i, T_j \rangle$ is generated and can be denoted as $\boldsymbol{p}_{i,j}$, *i.e.*, each dimension is calculated by Eq. (9). In the training time, we assign the predicate labels according to the IoU for each tracklet pair w.r.t the ground-truth tracklet pair by following Shang et al. (2021). We denote the sets of positive and negative tracklet pairs as $\mathcal{P}_p$ and $\mathcal{P}_n$, respectively. Due to the multi-label setting of VidVRD, we use binary cross-entropy loss for relation classification. The ground truth for positive tracklet pair in $\mathcal{P}_p$ is a binary vector of dimension $|\mathcal{C}_b^P|$, denoted as $\boldsymbol{p}_{i,j}^*$. For those negative tracklet pairs in $\mathcal{P}_n$, we optimize the probability of each *base* class to zero, *i.e.*, the ground truth is an all-zero vector. The classification loss is thus calculated as

$$\mathcal{L}_{\text{pred-cls}} = \frac{1}{|\mathcal{P}_p|} \sum\nolimits_{(T_i,T_j) \in \mathcal{P}_p} \text{BCE}(\boldsymbol{p}_{i,j}, \boldsymbol{p}_{i,j}^*) + \frac{1}{|\mathcal{P}_n|} \sum\nolimits_{(T_i,T_j) \in \mathcal{P}_n} \text{BCE}(\boldsymbol{p}_{i,j}, \boldsymbol{0}). \tag{10}$$

Training V2L Projection Module. Once the prompt representations are learned, we train a visual-to-language (V2L) projection module to use RoI Aligned features $\{\boldsymbol{f}_i\}$ as training data, and get rid of VLM's visual encoder at inference time. Given the learned prompt representations, we pre-extract the predicate class text embeddings (denoted as $\{\tilde{\boldsymbol{t}}_c^p\}$) for each prompt group and fix them. Formally, for each tracklet pair $\langle T_i, T_j \rangle$, we concatenate their RoI features as $\boldsymbol{f}_{i,j}^p = [\boldsymbol{f}_i, \boldsymbol{f}_j] \in \mathbb{R}^{4096}$. Then, we use a V2L projection module ϕ_p to project it to the same dimension as text embeddings. Similar to Eq. (9), the probability of predicate class c is predicted as

$$p_{i,j}(c) = \text{Sigmoid}(\cos(\phi_p(\boldsymbol{f}_{i,j}^p) + \phi_{\text{pos}}(\boldsymbol{f}_{i,j}^s), \tilde{\boldsymbol{t}}_c^p)), \ \forall c \in \mathcal{C}_b^P. \tag{11}$$

where ϕ_{pos} is the learned spatio-temporal projection layer and is fixed. Then, we apply the same loss as defined in Eq. (10), and compute the final total loss by averaging across all groups.

Discussions. Intuitively, we can train the prompt representations together with the V2L projection module ϕ_p, and use the l_1 loss to align $\phi_p(\boldsymbol{f}_{i,j}^p)$ to $\boldsymbol{v}_{i,j}^p$, *i.e.*, distill the knowledge from the pre-trained visual encoder to the V2L module. We name this variant as RePro†. We justify that our RePro works better than RePro† due to 1) directly using the teacher (*i.e.*, $\boldsymbol{v}_{i,j}^p$) to train the prompt is intuitively better than using student (*i.e.*, projected visual embedding), and 2) the distillation makes the V2L module focus too much on the static visual alignment, rather than the dynamic relation information learned in the prompt. In experiments, we empirically show the superiority of RePro over RePro†.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We evaluated our method on the VidVRD (Shang et al., 2017) and VidOR (Shang et al., 2019) benchmarks: 1) VidVRD consists of 1,000 videos, and covers 35 object categories and 132 predicate categories. We used official splits: 800 videos for training and 200 videos for testing. 2) VidOR consists of 10,000 videos, which covers 80 object categories and 50 predicate categories. We used official splits: 7,000 videos for training, 835 videos for validation, and 2,165 videos for testing. Since the annotations of VidOR-test are not released, we only evaluated on the validation set.

Evaluation Settings. To build the open-VidVRD setting, we manually split *base* and *novel* categories by selecting common object and predicate categories as *base* split, and selecting rare ones as *novel* split. The detailed splits are given in Sec. A.4 of the Appendix. We trained the model on the triplet samples of both *base* object and predicate categories in the training set. During testing, we evaluated the model on two settings: 1) **Novel-split**: triplet samples with all object categories and

novel predicate categories, and 2) **All-splits**: triplet samples with all object and predicate categories, in the testing set of VidVRD (or the validation set of VidOR).

Metrics. We follow three standard evaluation tasks in scene graph generation (Zellers et al., 2018): scene graph detection (SGDet), scene graph classification (SGCls), and predicate classification (PredCls). We apply these metrics to VidVRD: a detected triplet is considered to be correct if there is a same triplet tagged in ground truth, and both subject and object tracklets have a sufficient volume IoU (*e.g.*, 0.5) with the ground truth. Following the standard setting (Shang et al., 2017), we use mAP and Recall@K (R@K, K=50,100) as evaluation metrics.

4.2 Implementation Details

Tracklet Detector & Pre-trained VLM. We used the Faster-RCNN (Ren et al., 2015)-based VinVL model (Zhang et al., 2021) to detect frame-level object bounding boxes and extracted corresponding RoI Alinged features, and then adopted Seq-NMS (Han et al., 2016) to generate class-agnostic object tracklets. The VinVL model was trained on out-of-domain image data without seeing any VidVRD data. For pre-trained VLM, we used ALPro (Li et al., 2022), which was pre-trained on a wide range of video-language data, and learned the fine-grained visual region to text entity alignment.

Relation Detection Details. Following the popular segment-based methods (Qian et al., 2019; Shang et al., 2017; 2021), we first detected visual relations in short video segments, and then adopted relation association algorithm (Su et al., 2020) to merge the same relation triplets.

Hyperparameters. We set ϕ_o, ϕ_p and ϕ_{pos} all as two-layer MLPs with hidden dimension 768. The λ for weighting the distillation loss (*i.e.*, Eq. (5)) is set as 5.0. The prompt length L is set as 10. The softmax temperature τ is set as learnable. The GIoU threshold γ is chosen based on the statistics of the training set, by making the tracklet pairs evenly distributed w.r.t different motion patterns. In our implementation, γ is set as -0.3 for VidVRD and -0.25 for VidOR. We trained our RePro using Adam (Kingma & Ba, 2014) with learning rate 1e-4, and stop the training when SGDet mAP drops.

4.3 Evaluate Open-vocabulary Object Tracklet Detection

We evaluated the tracklet detection part of RePro on *novel* object categories, as shown in Table 1. We first verify the effectiveness of negative tracklet classification loss $\mathcal{L}_{\text{cls-neg}}$ in Eq. (4), and the visual distillation in Eq. (5). Then, we compare RePro with the pre-trained ALPro (Li et al., 2022).

Negative tracklet classification. In contrast to Eq. (4), an alternative way is to set a unique background embedding (BG-Embd) (Zareian et al., 2021; Gu et al., 2021). By comparing row **#3** and row **#4**, we find that without using background embedding (*i.e.*, as $\mathcal{L}_{\text{cls-neg}}$) achieves better recall, and outperforms the other by a large margin, especially on the more challenging VidOR benchmark. This is because the tracklets recognized as negative may be due to the fact that they contain novel objects (rather than backgrounds), and aligning their embeddings (*i.e.*, different novel class embeddings) to a unique background embedding hurts the model's recognition ability on novel objects.

Distillation. We varify the effectiveness of visual distillation (*i.e.*, Eq. (5)) by comparing rows: **#2** vs. **#3**. Obviously, the distillation helps RePro improve the detection recall by a large margin, especially for the more challenging VidOR benchmark. For row #1, we can observe that optimizing the negative tracklets' classification probability of each base class to $1/|\mathcal{C}_B|$ without distillation makes the model has poor generalize ability to novel category and has extremely low performance.

Table 1: Performance (%) of tracklet detection on objects with novel categories.

Methods		Distillation	BG-Embd	VidVRD-test		VidOR-val	
				R@5	R@10	R@5	R@10
ALPro (Li et al., 2022)		-	-	41.38	53.81	34.26	41.72
RePro	#1	×	×	2.17	3.71	2.33	3.48
	#2	×	✓	32.43	33.36	7.58	12.37
	#3	✓	✓	43.84	**53.00**	12.61	16.85
	#4	✓	×	**46.34**	50.42	**31.62**	**37.08**

Comparison to ALPro. We applied ALPro to Open-VidVRD by directly inputting the tracklet regions into its visual encoder to perform classification (as in Eq.(2)). Thanks to the distillation, our

RePro (row **#4**) can achieve comparable performances (to ALPro) on both datasets, with much less computational cost, since we use only one projection layer, *i.e.*, $\phi_o(\cdot)$.

4.4 Evaluate Open-vocabulary Relation Classification

Comparison to Conventional VidVRD SOTA Methods. We compare our RePro with several SOTA methods in the conventional VidVRD setting, and show the results in Table 2. We can observe that even when our RePro is trained with only base category samples (while others with both base and novel category samples), our performance on SGDet tasks is comparable to others'.

Table 2: Performance (%) comparision to conventional methods on VidVRD-test. Relation Tagging (RelTag) only considers the precision of relation triplets and ignores the localization of tracklets.

Methods	Training Data	SGDet			RelTag		
		mAP	R@50	R@100	P@1	P@5	P@10
Su et al. (2020)	base+novel	19.03	9.53	10.38	57.50	41.40	29.45
Liu et al. (2020)	base+novel	18.38	11.21	13.69	60.00	43.10	32.24
Li et al. (2021)	base+novel	**22.97**	12.40	14.46	**68.83**	**49.87**	**35.57**
Gao et al. (2022)	base+novel	17.67	9.63	11.29	56.00	43.80	32.85
RePro (Ours)	base	21.72	**12.61**	**15.84**	61.50	43.70	31.92

Comparisons in the setting of Open-VidVRD. We compare the model performances in the setting of Open-VidVRD and show results in Table 3. As our RePro is the first Open-VidVRD method, we compare it to the SOTA method VidVRD-II (Shang et al., 2021) trained on the samples of base categories. We re-implemented VidVRD-II and replaced its classifier with text embeddings extracted by the ALPro's text encoder (for both objects and predicates). We also report the results of using ALPro directly to perform relation classification as in Eq. (2). From the results in Table 3, we can observe that our RePro outperforms ALPro and VidVRD-II by a large margin on both Novel-split and All-splits. We also compare our RePro with its intuitive variant RePro† as mentioned in the "**Discussion**" of Sec. 3.3. The results show that RePro outperforms RePro† by a large margin on both splits, validating the effectiveness of our training scheme for RePro.

Table 3: Performance (%) comparision of Open-VidVRD methods on VidVRD-test.

Split	Methods	SGDet			SGCls			PredCls		
		mAP	R@50	R@100	mAP	R@50	R@100	mAP	R@50	R@100
Novel	ALPro	1.33	3.47	4.46	3.45	7.43	8.92	4.09	9.42	10.41
	VidVRD-II	4.09	9.58	11.90	5.52	14.21	18.84	7.35	18.84	26.44
	RePro†	3.89	8.26	10.74	7.08	14.71	19.83	9.34	18.67	24.13
	RePro	**6.63**	**12.06**	**15.20**	**9.43**	**20.00**	**24.46**	**12.74**	**25.12**	**33.88**
All	ALPro	3.08	2.60	3.16	3.79	3.78	4.92	4.97	4.50	5.79
	VidVRD-II	12.65	9.98	12.96	16.95	14.82	20.06	19.73	18.17	24.90
	RePro†	16.48	11.14	14.43	21.97	16.73	22.31	25.43	21.36	28.04
	RePro	**21.72**	**12.61**	**15.84**	**29.79**	**19.95**	**25.37**	**34.90**	**25.50**	**32.49**

4.5 Ablation Studies

We conducted careful ablation studies to show the effectiveness of our design of prompt tuning, as shown in Table 4. **#1**: training with neither compositional prompt nor motion cues. Specifically, we learn a single prompt representation $\boldsymbol{W}_c$ as in Eq. (1). The obtained predicate text embedding has the half dimensions of the $\boldsymbol{t}_c^p$ in Eq. (6). Therefore, we calculate the visual embeddings of a tracklet pair as[1] $\boldsymbol{v}_{i,j}^p = \boldsymbol{v}_i^o - \boldsymbol{v}_j^o$ (different from the concatenated vector $\boldsymbol{v}_{i,j}^p$ in Eq. (9)). **#2**: training with compositional prompt but without motion cues. **#3 (Ens)**: training with compositional prompt, but the prompt is randomly selected from the 6 groups without considering motion cues. For testing,

[1]To avoid introducing extra parameters (and also avoid over-fitting to base categories), we do not use the concatenated $\boldsymbol{v}_{i,j}^p$ and train an MLP to reduce its dimension to d.

the prompts are ensembled by averaging. **#4 (Rand)**: training as in **#3**, but the prompt is randomly selected at testing. **#5**: our RePro, where the prompt is chosen according to the motion cues.

Compositional Prompt. By comparing the results in **#1** and **#2**, we can observe that the compositional prompt can effectively improve the performance on Novel-split. Meanwhile, the improvement on All-splits is not significant. We conjecture that the base relations, the majority of All-splits, require less compositional semantic contexts for prompt learning.

Table 4: Ablations (%) for RePro with different prompt design in VidVRD-test, where **C** stands for Compositional, and **M** stands for Motion cues. **Ens**: ensemble all the learned prompts by averaging their representations. **Rand**: reandomly select a prompt without considering motion cues.

		C	M	SGDet			SGCls			PredCls		
				mAP	R@50	R@100	mAP	R@50	R@100	mAP	R@50	R@100
Novel-split	#1	×	×	3.35	9.25	12.39	7.33	15.70	21.15	8.63	20.33	27.43
	#2	✓	×	6.45	10.74	13.55	9.12	18.51	23.80	11.83	22.31	30.90
	#3	✓	Ens	5.45	11.23	13.38	**9.84**	17.19	22.31	12.36	21.32	29.91
	#4	✓	Rand	6.28	11.23	14.04	9.37	17.19	23.63	12.42	22.64	30.90
	#5	✓	✓	**6.63**	**12.06**	**15.20**	9.43	**20.00**	**24.46**	**12.74**	**25.12**	**33.88**
All-splits	#1	×	×	19.44	12.28	15.01	26.55	18.38	24.36	30.80	23.70	30.42
	#2	✓	×	18.80	11.95	15.09	25.28	18.28	24.28	29.45	23.39	30.17
	#3	✓	Ens	19.74	12.47	15.42	27.60	18.75	24.34	31.68	23.61	30.29
	#4	✓	Rand	21.02	**12.67**	15.73	28.84	19.31	25.12	33.11	24.38	31.49
	#5	✓	✓	**21.72**	12.61	**15.84**	**29.79**	**19.95**	**25.37**	**34.90**	**25.50**	**32.49**

Motion-based Prompt Groups. By comparing our RePro (**#5**) vs. **#2**, we can observe that with the help of motion cues, our RePro achieves significant improvement of recall on all tasks, and also achieves considerable improvement of mAP on most tasks. We can see that the improvement on Novel-split is more significant than that that on All-splits, *e.g.*, 13.55%→15.20% vs. 15.09%→15.84% on R@100 of SGDet, showing that the motion-based prompt has a better generalizability for detecting novel relations. Besides, if comparing RePro (**#5**) to **Ens** (**#3**), we can see that RePro outperforms **Ens** in Novel-split on most tasks, and achieves considerable improvements for All-splits on all tasks. Compared to **Rand** (**#4**), RePro achieves clear improvements on most metrics for both Novel-split and All-splits.

Table 5: Ablations (%) on VidOR-val.

	C	M	SGCls		PredCls	
			R@50	R@100	R@50	R@100
Novel-split	×	×	0.86	0.86	2.30	2.88
	✓	×	1.72	2.59	6.62	8.06
	✓	Ens	**2.30**	**2.59**	7.20	8.35
	✓	Rand	2.01	2.30	5.76	7.20
	✓	✓	2.01	2.30	**7.20**	**8.35**
All-splits	×	×	9.49	12.85	25.62	34.83
	✓	×	**10.06**	**13.40**	27.00	**36.73**
	✓	Ens	9.49	12.68	25.66	35.16
	✓	Rand	10.03	13.13	26.94	36.48
	✓	✓	10.03	12.91	**27.11**	35.76

Ablations on VidOR. We conducted the same ablation studies on VidOR-val, as shown in Table 5. Firstly, we can observe that the compositional prompt representation shows its efficiency on both Novel-split and All-splits, *e.g.*, 0.86%→1.72% and 9.49%→10.06% on R@50 of SGCls. For the motion-based prompt groups, the improvement of RePro is small due to the biased data distribution (Li et al., 2021), *i.e.*, the predicate categories strongly depend on the visual cues of subject and object tracklets, making the model predict relations simply based on object appearances without considering motion cues. More results on VidOR are given in the Appendix (Sec. A.3).

5 Conclusions

In this paper, we introduced a challenging video visual relation detection task on open-vocabulary data — Open-VidVRD. We analyzed two key characteristics, *i.e.*, compositional and motion-related, when applying the method of prompt tuning in this new setting, and proposed a novel method called RePro that learns compositional prompt representations while considering motion-based contexts in the video. Our evaluations on both conventional and open-vocabulary datasets show a clear superiority of RePro for tackling video visual relation detection tasks.

ETHICS AND REPRODUCIBILITY STATEMENTS

Ethics statement. The open-vocabulary video visual relation detection (Open-VidVRD) that we introduced in this paper is a general extension of conventional VidVRD, and there are no known extra ethical issues in terms of the Open-VidVRD task and the proposed RePro model. As for the pre-trained visual-language model (VLM) used for Open-VidVRD, the large-scale pre-training data might contain some videos and captions involved with discrimination/bias issues. When applying pre-trained VLM to Open-VidVRD, the model tends to predict relations based more on the pre-trained knowledge, and focus less on the visual cues. For example, when the pre-trained data involved with unethical videos and captions, a model might predict "person-punch-dog" given a video of person caressing dog, which implies a person is abusing animals. To avoid the potential ethical issues, we can design algorithms to filter out those unethical training data for VLMs. For Open-VidVRD models, we can also introduce some common sense knowledge and design some rule-based methods to filter out those unreasonable relation triplets that involve ethical issues.

Reproducibility Statement. Our RePro is mainly implemented based on the realsed code of ALPro (Li et al., 2022), VinVL (Zhang et al., 2021), and VidVRD-II (Shang et al., 2021). We first modified the code of VinVL to fit the video data and to extract object tracklets in each video. Then We modified the code of VidVRD-II to be compatible with the visual and text encoder of ALPro, and to fit the Open-VidVRD setting. We provide the the detailed base/novel split information of object and predicate categories in the Appendix (Sec. A.4) to ensure all experiments can be reproduced. When training the RePro model and its variants, we manually set the random seed and fixed the seed for all experiments to ensure they can be reproduced. We also provide the code of our RePro model and the training/evaluate scripts in the supplementary materials.

REFERENCES

Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*, pp. 4613–4623, 2019.

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

Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pp. 14084–14093, 2022.

Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, 2022.

Kaifeng Gao, Long Chen, Yulei Niu, Jian Shao, and Jun Xiao. Classification-then-grounding: Reformulating video scene graphs as temporal bipartite graphs. In *CVPR*, pp. 19497–19506, 2022.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2021.

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.

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

Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. Towards open-vocabulary scene graph generation with prompt-based finetuning. *arXiv preprint arXiv:2208.08165*, 2022.

Peiliang Huang, Junwei Han, De Cheng, and Dingwen Zhang. Robust region feature synthesizer for zero-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7622–7631, June 2022.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. In *ACL*, pp. 2763–2775, 2022.

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

Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*, pp. 4953–4963, 2022.

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

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*, pp. 10840–10849, 2020.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

Zongyang Ma, Guan Luo, Jin Gao, Liang Li, Yuxin Chen, Shaoru Wang, Congxuan Zhang, and Weiming Hu. Open-vocabulary one-stage detection with hierarchical visual-language knowledge distillation. In *CVPR*, pp. 14074–14083, 2022.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *36th Conference on Neural Information Processing Systems (NIPS)*, 2022.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In *ICMR*, pp. 279–287, 2019.

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

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*, pp. 3127–3135, 2020.

Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pp. 14393–14402, 2021.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *CVPR 2021*, 2021.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, pp. 1–12, 2022b.

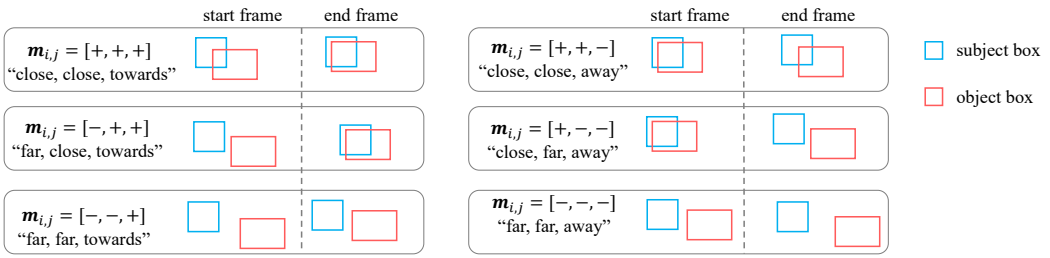


Figure 4: The schematic of the 6 motion patterns defined in Eq. (8).

A Appendix

This Appendix has the following contents:

- More details about the relative position feature are in Sec. A.1.
- More details about the motion pattern are in Sec. A.2.
- More experiment results on VidOR are provided at Sec. A.3.
- The detailed base/novel split information of object and predicate categories are in Sec. A.4.

A.1 Relative Position Feature for Tracklet Pairs

We compute the relative position between bounding boxes of subject-object tracklet pair $\langle T_i, T_j \rangle$ by following Shang et al. (2021). Specifically, we compute the relative position feature between subject-object for the beginning and ending bounding boxes of their temporal intersection. For the beginning bounding boxes of the subject and object, the position feature is calculated as:

$$\boldsymbol{f}^B_{i,j} = \left[\frac{x_i - x_j}{x_j}, \frac{y_i - y_j}{y_j}, \log \frac{w_i}{w_j}, \log \frac{h_i}{h_j}, \log \frac{w_i h_i}{w_j h_j}, \frac{t_i - t_j}{L_{\text{seg}}}. \right], \tag{12}$$

where (x_i, y_i) is the central coordinates of T_i's beginning bounding box, (w_i, h_i) is its width and height, and t_i is the frame ID of this beginning bounding box. $(x_j, y_j, w_j, h_j, t_j)$ is defined similarly for T_j. L_{seg} is the number of frames in each video segment, and following Shang et al. (2021), we set $L_{\text{seg}} = 30$. The relative position feature between the ending bounding boxes of $\langle T_i, T_j \rangle$ is defined similarly as $\boldsymbol{f}^B_{i,j}$, and is denoted as $\boldsymbol{f}^E_{i,j}$. The final relative position of $\langle T_i, T_j \rangle$ is concatenated as $\boldsymbol{f}^s_{i,j} = [\boldsymbol{f}^B_{i,j}, \boldsymbol{f}^E_{i,j}]$, and $\boldsymbol{f}^s_{i,j} \in \mathbb{R}^{12}$.

A.2 Details About the Motion Patterns

We provide a schematic of the motion patterns defined in Eq. (8), as shown in Figure 4. According to the definition in Eq. (8), $\boldsymbol{m}_{i,j}$ has only 6 possible values, *i.e.*, $[+,-,+]$ and $[-,+,-]$ are impossible.

A.3 More Experiment Results on VidOR

Table 6: Performance (%) on the validation set of VidOR.

Methods	Novel-split				All-splits			
	SGCls		PredCls		SGCls		PredCls	
	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
ALPro	3.17	3.74	8.35	9.79	0.95	1.32	2.61	3.66
VidVRD-II	1.44	2.01	4.32	4.89	9.40	12.78	24.81	34.11
RePro†	1.72	2.30	6.62	8.06	8.88	11.52	23.84	31.57
RePro	**2.01**	**2.30**	**7.20**	**8.35**	**10.03**	**12.91**	**27.11**	**35.76**

We provide more experiment results for our RePro on the validation set of VidOR, as shown in Table 6. We first compare our RePro with using ALPro directly perform relation classification as Eq. (2). We find that ALPro performs slightly better than RePro on novel-split, because ALPro

doesn't have the trend of fitting base categories. However, ALPro performs much worse than RePro in All-splits due to not trained on base categories. Furthermore, we also compare RePro with the VidVRD-II (Shang et al., 2021) baseline and the variant RePro†. We can observe that RePro outperfoms both VidVRD-II and RePro† by a large margin on both novel-split and all-split.

A.4 Detailed base/novel categories of object and predicate for VidVRD and VidOR

We list the base/novel categories of object and predicate for training and evaluating our RePro and other baselines in all experiments. We also provide more statistics information in the supplementary materials

VidVRD Object

25 base object categories:

"airplane", "bicycle", "bird", "bus", "car", "dog", "domestic_cat", "elephant", "hamster", "lion", "monkey", "rabbit", "sheep", "snake", "squirrel", "tiger", "train", "turtle", "whale", "zebra", "ball", "frisbee", "sofa", "skateboard", "person"

10 novel object categories

"horse", "watercraft", "giant_panda", "fox", "red_panda", "cattle", "motorcycle", "bear", "antelope", "lizard"

VidVRD Predicate

71 base predicate categories:

"behind", "chase", "creep_behind", "creep_beneath", "creep_front", "creep_left", "creep_right", "fall_off", "faster", "fly_above", "fly_next_to", "fly_past", "fly_toward", "fly_with", "follow", "front", "jump_beneath", "jump_front", "jump_left", "jump_next_to", "jump_right", "jump_toward", "larger", "left", "lie_behind", "lie_front", "lie_left", "lie_next_to", "lie_right", "move_behind", "move_beneath", "move_front", "move_left", "move_right", "move_with", "next_to", "play", "ride", "right", "run_behind", "run_front", "run_left", "run_past", "run_right", "run_with", "sit_above", "sit_front", "sit_left", "sit_right", "stand_behind", "stand_front", "stand_left", "stand_next_to", "stand_right", "stop_behind", "stop_front", "stop_left", "stop_right", "swim_front", "swim_left", "swim_right", "swim_with", "taller", "touch", "walk_behind", "walk_front", "walk_left", "walk_next_to", "walk_right", "walk_with", "watch"

61 novel predicate categories:

"above", "away", "beneath", "bite", "creep_above", "creep_away", "creep_next_to", "creep_past", "creep_toward", "drive", "feed", "fight", "fly_away", "fly_behind", "fly_front", "fly_left", "fly_right", "hold", "jump_above", "jump_away", "jump_behind", "jump_past", "jump_with", "kick", "lie_above", "lie_beneath", "lie_inside", "lie_with", "move_above", "move_away", "move_next_to", "move_past", "move_toward", "past", "pull", "run_above", "run_away", "run_beneath", "run_next_to", "run_toward", "sit_behind", "sit_beneath", "sit_inside", "sit_next_to", "stand_above", "stand_beneath", "stand_inside", "stand_with", "stop_above", "stop_beneath", "stop_next_to", "stop_with", "swim_behind", "swim_beneath", "swim_next_to", "toward", "walk_above", "walk_away", "walk_beneath", "walk_past", "walk_toward"

VidOR Object

50 base object categories:

"adult", "child", "toy", "dog", "baby", "car", "chair", "table", "sofa", "ball/sports_ball", "screen/monitor", "cup", "bicycle", "guitar", "bottle", "backpack", "handbag", "baby_seat", "camera", "cat", "cellphone", "bird", "sheep/goat", "laptop", "ski", "stool", "watercraft", "duck", "bus/truck", "bench", "fruits", "baby_walker", "horse", "bat", "dish",

"electric_fan", "kangaroo", "motorcycle", "lion", "hamster/rat", "refrigerator", "elephant", "faucet", "cake", "penguin", "sink", "piano", "microwave", "cattle/cow", "aircraft"

30 novel object categories:

"antelope", "vegetables", "panda", "rabbit", "fish", "train", "snowboard", "suitcase", "squirrel", "leopard", "chicken", "skateboard", "traffic_light", "surfboard", "camel", "racket", "bread", "bear", "oven", "scooter", "frisbee", "stop_sign", "turtle", "stingray", "pig", "crab", "crocodile", "toilet", "tiger", "snake"

VidOR Predicate

30 base predicate categories:

"next_to", "in_front_of", "watch", "behind", "away", "towards", "beneath", "above", "hold", "lean_on", "speak_to", "ride", "hug", "touch", "carry", "hold_hand_of", "bite", "push", "pull", "play(instrument)", "grab", "release", "pat", "inside", "lift", "caress", "point_to", "press", "hit", "use"

20 novel predicate categories:

"kick", "chase", "wave", "smell", "throw", "feed", "kiss", "wave_hand_to", "shout_at", "drive", "clean", "lick", "squeeze", "shake_hand_with", "get_off", "knock", "cut", "open", "get_on", "close"