Supplementary Material
for
Few-Shot Referring Relationships in Videos

A. Tracklet-based Method

In this section, we provide a detailed explanation of the tracklet-based method presented in section 4.3 of the main paper. Consider notations independent of the main method.

A.1. Tracklet Detector.

For each video, using a pre-trained detector, we extract all tracklets (represented as a set T_e) features and spatio-temporal locations. Each entity $e_i \in T_e$ having length l_i as number of frames is categorized by a time slot (s_i^e, e_i^e) and bounding box coordinate $b_i \in R^{l_i \times 4}$. For each entity e_i, tracklet feature $feat_i$ is a combination of spatial and appearance features. Spatial feature $feat_i^s \in R^{l_i \times 8}$ is formed as the concatenation of all box offset Δb_i and coordinate b_i, where $\Delta b_{i,j}$ is defined between two consecutive frames as box coordinate offset, i.e., $\Delta b_{i,j} = b_{i,j+1} - b_{i,j}$. For appearance $feat_i^a \in R^{l_i \times d_a}$ each frame is extracted using RoIAlign [**?**] based on box locations. For entity e_i, the tracklet feature $feat_i \in R^{l_i \times d_e}$ is formed as,

$$feat_i = f_a[f_b(feat_i^a); f_c(f_i^s)]. \tag{1}$$

where f_a, f_b and c are learnable linear models and [;] is concatenation operation.

A.2. Relationship Pair Proposal.

The goal of this module is to propose relationship pair proposals. Given a video v, the first tracklet set T_e is extracted using a pre-trained detector. An attention score is calculated for each tracklet for finding whether a tracklet contains a subject similar for the object. A tracklet having a high attention score is more likely to contain the desired object/subject category. To this end, corresponding to subjects, an attention vector is computed $\alpha_{s_{n\times 1}} = SubAtt(T_e, w2v(s))$, where $SubAtt$ is attention unit which takes all tracklet features along with word2vec features of query subject and predicate. Similarly for objects, we have $\alpha_{o_{n\times 1}} = ObjAtt(T_e, w2v(o)$. To calculate attention vectors α first a score for each entity (subject/object) is computed as,

$$s_j = f_d(f_e[feat_j; w2v(s)]). \tag{2}$$

where f_d and f_e are learnable linear functions. Finally, the attention vector is computed as,

$$\alpha_s = SubAtt(T_e, w2v(s)), where, \alpha_j = \frac{e^{s_j}}{\sum_{i=1}^{n} e^{s_i}}. \tag{3}$$

Similarly, an attention vector α_o corresponding to an object is computed. To make entity tracklet features aware of each other we use an attention-shifting mechanism [1], where attention is shifted from one entity to every other entity. Attention shift from subject to object computed as, $\alpha_{so} = W_{so}\alpha_s$, similarly object to subject $\alpha_{os} = W_{so}\alpha_o$, where W_{so} and W_{os} are learnable parameters. Then entity tracklet features aware of each other computed as, $feat_s = f_{os}[\alpha_{os}; w2v(p)]$ and $feat_o = f_{so}[\alpha_{so}; w2v(p)]$, where $feat_s$ and $feat_s$ are objects and subject features respectably and f_{os} and f_{so} are linear learnable modules. Finally, probable query relationship pair proposals are computed as,

$$feat_p = f_e(ReLu(f_d([\phi(feat_s; feat_o)], w2v(p)))). \tag{4}$$

where $feat_p$ is a set pair proposals features, and $\phi()$ is function to enumerate all possible pair proposals.

A.3. Metric Learner.

This module aims to rank the pair proposals produced by the pair proposal module. A deep metric is learned with the help of support set videos in a few-shot way. Deep metric learns to project positive pair proposals (pair proposals with the same predicate) close in the feature space while negative pair proposals (pair proposals with different predicates) are far away.

Feature Extraction. Given a k^{th} support set video, we have ground truth bbox tracklets of entities (subject) $feat_{k_s}^s \in R^{l_k \times 8}$ of query predicate. We extracted appearance feature $feat_{k_s}^{l_k \times d_a}$ based on bbox using a pre-trained detector. To incorporate spatial and visual fea-

		VidVRD					VidOR				
Approach	Support Set Size	A^{sub}_{s-t}	$mIoU_s$	Acc^{obj}_{s-t}	$mIoU_o$	A^{r}_{s-t}	A^{sub}_{s-t}	$mIoU_s$	A^{obj}_{s-t}	$mIoU_o$	Acc^{r}_{s-t}
tracklet-based	2	12.3	12.1	12.2	11.7	8.8	10.9	10.7	10.1	10.3	7.3
tracklet-based	4	12.6	12.9	12.8	12.0	9.6	11.1	11.0	10.9	10.4	8.2
tracklet-based	8	13.4	13.2	13.7	13.3	10.4	12.8	12.6	13.2	13.0	9.7
w/o vis.sim. potential, w/o GSA, w/o LLA	2	19.2	18.2	17.8	19.9	15.9	19.3	18.1	18.4	19.8	16.7
w/o vis.sim. potential, w/o GSA, w/o LLA	4	19.6	19.1	18.2	20.8	16.3	19.8	18.7	18.5	20.4	17.3
w/o vis.sim. potential, w/o GSA, w/o LLA	8	20.2	19.6	18.5	21.6	17.1	20.2	19.4	18.7	20.9	18.4
w/o vis.sim. potential	2	21.4	20.6	22.1	21.3	16.9	19.3	21.4	21.4	21.8	17.2
w/o vis.sim. potential	4	22.6	21.9	22.3	22.9	17.2	20.9	22.5	22.7	22.1	17.6
w/o vis.sim. potential	8	22.9	22.3	22.8	23.7	17.8	21.3	22.8	23.4	23.7	18.6
Greedy Solver	2	23.8	23.2	22.9	25.2	22.4	23.7	21.7	23.4	24.7	20.4
Greedy Solver	4	25.4	24.7	24.3	26.4	22.8	24.2	23.8	24.5	25.8	21.7
Greedy Solver	8	25.7	25.4	25.1	26.7	23.1	24.8	23.8	24.7	26.4	22.3
Full model	2	24.5	26.3	25.7	26.7	24.3	24.8	25.7	26.1	26.8	22.6
Full model	4	26.8	27.1	26.0	27.9	25.1	25.3	26.2	26.3	27.0	23.8
Full model	8	27.4	28.3	26.8	28.2	25.7	26.1	27.3	26.8	27.6	24.4

Table 1. Comparison of baseline and proposed method on different support set sizes. For every method (baseline and all variants of our method) the performance is increased with an increase in support set size.

tures, the final subject features $feat_{k_s}$ is computed using equation 1. Similarly, object feature $feat_{k_o}$ is computed. Then positive predicate feature is computed as $feat_{p(+ve)} = MLP[f_{k_s}; f_{k_s}; w2v(p)]$. Similarly, for each supported video, negative predicate feature $feat_{p(-ve)}$ is computed where the predicate is not the same as in the query relationship.

A metric learner **ML** is defined as a linear differentiable model and trained using the ground truth positive and negative predicate feature in a supervised setting. The loss function for this module is defined as,

$$\begin{aligned}\mathcal{L}_1 = \sum_{i=1}^{k} \sum_{j} ((ML(f_{p_i(+ve)}) - ML(f_{p_j(+ve)})) \\ + (ML(f_{p_i(-ve)}) - ML(f_{p_j(-ve)})) \\ - (ML(f_{p_i(+ve)}) - ML(f_{p_j(-ve)})) \\ - (ML(f_{p_i(-ve)}) - ML(f_{p_j(+ve)}))).\end{aligned} \tag{5}$$

The pair proposals produced by the relationship proposal module are ranked by the learned deep metric ML. The top-ranked pair proposal $feat_t$ has the lowest l_2 distance from all the positive ground truth predicates. The top-ranked pair is returned by the following equation.

$$feat_t = \min_{i=1}^{n} (\sum_{j=1}^{k} ML(f_i) - ML(f_{k(+ve)})). \tag{6}$$

A.4. Reconstruction module.

This module aims to ensure the effective use of word embedding of query relationships in the aforementioned modules. We took the inspiration of reconstruction as weak supervision [1, 2] to use query relationship embedding effectively. The first few (set as hyperparameters) top-ranked predicate feature returns from the previous module are stacked together to form an encoding for the LSTM, i.e., $f_{emb} = MLP(\psi(f))$, where ψ return the top-ranked stacked predicate features. LSTM considers relation as a textual phrase and reconstructs it using a single layer of LSTM. This module is trained using the following loss function,

$$\mathcal{L}_2 = -\sum_{i=1}^{l_r} logP(w_i | w_{0:i-1}, f_{emb}). \tag{7}$$

A.5. Training and Inference.

The top-ranked feature contains the tracklet information of the subject and object related to the query predicate. But the query visual relation might not span the whole duration of the subject and object and can be in multiple time slots. To correctly find the duration and probable multiple slots we learn another linear model to predict each frame whether it belongs to a query relationship or not. The linear model is defined as,

$$r_d = softmax(MLP([f_{emb}; f_t])) \tag{8}$$

r_d has a dimension equal to twice the max frame length in the dataset. For each common frame of subject and object, we have a score for starting of the relationship and similarly for the end of a relationship. This module is trained using the following supervised loss,

$$\mathcal{L}_3 = -tIoU(bbox_{gt}, bbox_{pred}) \tag{9}$$

$tIOU$ computes the temporal intersection over the union of predicted predicates with respect to the ground truth bounding boxes. The start and end of the query relation are extracted as,

$$F_s = \{i\,; r_d[i] > \epsilon\}_{i=1}^{d/2}, F_e = \{i\,; r_d[i] > \epsilon\}_{i=d/2}^{d} \tag{10}$$

The frames which are not contained within the common duration of the subject and object are removed. The lowest frame number i_s is selected from F_s as the start of the relationship and the lowest frame from i_e such that $i_e > i_s$ is selected from F_e as the end of the relation. The frames contained in between i_s and i_e are removed from F_s and F_e to find another query relation instant in a similar way.
Finally, all modules are combined during training as an end-to-end differential function. The overall model is trained using the following loss function,

$$\mathcal{L} = \mathcal{L}_1 + \lambda_1 \times \mathcal{L}_2 + \lambda_2 \times \mathcal{L}_3 \tag{11}$$

where λ_1 and λ_2 are hyperparameters.

References

[1] Junbin Xiao, Xindi Shang, Xun Yang, Sheng Tang, and Tat-Seng Chua. Visual relation grounding in videos. In *Proc. ECCV*, 2020. 1, 2

[2] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *Proc. ECCV*, 2016. 2