Supplementary for Representation Learning on Visual-Symbolic Graphs for Video Understanding

Effrosyni Mavroudi[1], Benjamín Béjar Haro[1], and René Vidal[1]

Johns Hopkins University, Baltimore, MD, 21218, USA
{emavrou1,bbejar,rvidal}@jhu.edu

A Appendix

This Appendix provides additional implementation details, ablations and qualitative results.

- In Section A.1, we provide more details for our experiments on the CAD-120 dataset.
- In Section A.2, we provide more details for our experiments on the Charades dataset.
- In Section A.3, we describe the implementation details for our experiments on the ActivityNet Entities dataset.
- In Section A.4, we provide additional ablations on the CAD-120 and Charades datasets, as well as additional qualitative results on the Charades and ActivityNet Entities datasets.

A.1 CAD-120: Implementation Details

In Sec. 4.1 of our main paper, we describe the implementation details of our model used for sub-activity and object affordance detection on the CAD-120 datasets. Here we provide more details about the visual and symbolic graphs.

Visual st-graph. We instantiate a visual st-graph on the actor and objects of each temporal segment of an input sequence. Actor node features correspond to human skeleton joint positions, body pose and hand position features. Object node features correspond to the location of the object and its trajectory in the temporal segment. Edge features describe the geometric relationship between nodes i and j, such as difference in centroids and distance between them. (For an analytic description of the available hand-crafted features see [4]). There are 5 edge types: edges connecting objects in the same temporal segment (*obj-obj-sp*), edges connecting objects with the actor within a temporal segment (*obj-act-sp*), edges connecting the actor with objects within a temporal segment (*act-obj-sp*), edges connecting actors between two consecutive temporal segments (*act-act-t*) and edges connecting objects between two consecutive temporal segments (*obj-obj-t*).

Symbolic Graph. We construct a symbolic graph that has 22 nodes corresponding to the 10 sub-activity and 12 affordance classes, with edge weights capturing per-frame class co-occurrences in training data (Fig. 7). The attribute of each symbolic node is obtained by using off-the-shelf word2vec [6] embeddings of size $K = 300$ to represent the semantic class. Actor/object visual nodes are connected to sub-activity/affordance symbolic nodes, respectively.

A.2 Charades: Implementation Details

In Sec. 4.2 of our main paper, we describe the two models whose predictions were fused for temporal action localization on the Charades dataset: a) a global (coarse-grained) model based on whole frame convolutional features and a long-term temporal model and b) a local (fine-grained) model based on our Visual Symbolic - Spatio Temporal - Message Passing Neural Network (VS-ST-MPNN). We briefly described the instantiation of the visual st-graph and symbolic graph, the architecture hyper-parameters and the choice of loss for the task of temporal action localization on the Charades dataset. Here we discuss in more detail the construction of the visual and symbolic graphs, as well as the training and hyperparameters of the global model.

Global model. The global model is an I3D RGB model [1] fine-tuned on Charades by the authors of [7], combined with a two-layer bidirectional Gated Recurrent Unit (biGRU) of hidden size 256. In particular, we extract per frame features from the `Mixed_5c` layer of the I3D, which are then fed as an input to the biGRU.

Visual st-graph. The nodes of the visual st-graph correspond to bounding boxes of people (actors) and objects. A Faster-RCNN [3] architecture pretrained on the MSCOCO [5] dataset is used for obtaining these bounding boxes. We detect objects from the 80 categories of the COCO dataset, retaining all object detections with a confidence score above 0.15 and from those we keep the two highest scoring human detections and 10 object detections per frame. Bounding boxes are enlarged by a relative margin of 30% at each side. Note that we do not use the predicted classes of the detected objects in our model. Rather than using the object detector features for describing the actors and objects, we exploit the rich spatio-temporal feature maps of the I3D action recognition model, by pooling features from the `Mixed_4f` feature map of the I3D, which has a spatial output stride of 16 pixels, a temporal output stride of 4 frames and 832 channels. In particular, we first temporally downsample the spatio-temporal feature map to obtain an effective temporal downsampling by a factor of 16 frames (1.5FPS, maximum 109 frames per video) and then we apply RoIAlign [3] to pool features from each detected object at each downsampled frame. This leads to a feature map of $832 \times 7 \times 7$ per region per frame. To obtain a single feature vector for each actor and object node, we perform max-pooling over space. The edge features are obtained by embedding the relative position of the bounding boxes corresponding to the nodes connected by the edge.

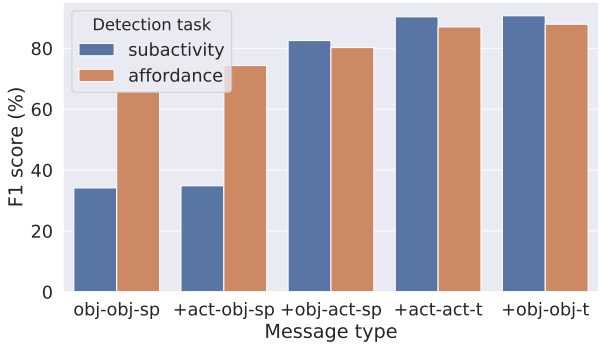

Fig. 1: Ablation on CAD-120 [4] sub-activity and object affordance detection performance by incrementally adding edge types, starting from using only *object-to-object spatial* edges.

Symbolic graph. Our symbolic graph has nodes corresponding to the 157 action classes and edge weights corresponing to per-frame label co-occurrences in training data, with a temporal downsampling of one every 100 frames. Fig. 8 shows the adjacency matrix for a subset of action classes of the Charades dataset. To obtain the linguistic embedding of each action class, we first map the action class name to a verb and object pair, then use off-the-shelf word2vec [6] embeddings of size $K = 300$ to represent the verb and the object and finally we average them.

Hyperparameter selection Given the input visual and symbolic graphs, the proposed VS-ST-MPNN has 4 hyperparameters: the message sizes (d_L, D_s) and the number of layers (L,R). L and d_L were chosen via cross-validation, while the rest of the hyperparameters (including those of backbone and recognition networks) were either set to default values or chosen visually (e.g. the object detector threshold).

A.3 ActivityNet Entities: Implementation Details

As explained in the main paper, for the grounded video description experiments on the ActivityNet Entities dataset we augment the model of Zhou et al. [9] (GVD) with our visual and/or symbolic graph message passing modules. The GVD model uses a hierarchical LSTM decoder that generates a descriptive sentence based on global video features along with local region features of 100 region proposals extracted in 10 equidistant frames of the video segment and it utilizes the attention coefficients to ground the nouns in the image. One of the components of the GVD model is a multi-head self-attention mechanism used to refine local region features, akin to our visual graph message passing module. In our main paper, we explored ways of replacing (or augmenting) that component with our VS-ST-MPNN (or symbolic graph module), respectively. The rest of

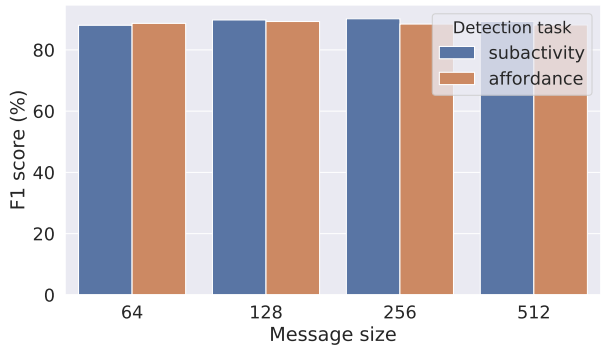

Fig. 2: Effect of varying message size on CAD-120 [4] sub-activity and object affordance detection.

the GVD architecture is the same as in [9], including the training procedure and losses.

Model hyperparameters. The hyperparameters of our model used in this dataset are: $L = 2$, $d_L = 1024$, $R = 2$, $D_s = 256$, $\lambda_v = 1, \lambda_e = 0$ and $\lambda_{ea} = 0$ (no edge features because of memory limitations). We used *obj-obj-sp* and *act-obj-sp* edges. Our batch size is 80 video clips, the learning rate is set to 0.0003 and we train for 30 epochs.

Visual st-graph. The actor nodes for each frame of the clip correspond to the top 10 object detections that belong to one of the 42 manually defined actor classes: *adult*, *baby*, *biker*, *bride*, *boy*, *catcher*, *chef*, *child*, *couple*, *cyclist*, *driver*, *fire extinguisher*, *girl*, *guy*, *groom*, *kid*, *lady*, *little girl*, *male*, *man*, *men*, *mother*, *motorcyclist*, *officer*, *passenger*, *pedestrian*, *person*, *player*, *pitcher*, *police officer*, *policeman*, *racer*, *referee*, *rider*, *she*, *skateboarder*, *skater*, *skier*, *tennis player*, *umpire*, *woman*, *worker*, *young man*. The object nodes correspond to the rest 90 object detections per frame, including background detections.

Symbolic graph. Our symbolic graph has nodes corresponding to the 431 object classes and edge weights corresponing to per-sentence object label co-occurrences in training data. The adjacency matrix is binarized by thresholding co-occurence frequency values with a threshold of 0.2, removing spurious edges between object classes with very few co-occurrences. Fig. 9 shows a part of the adjacency matrix. To obtain the linguistic embedding of each object, we use off-the-shelf word2vec [6] embeddings of size $K = 300$. These input symbolic node embeddings are visualized in Fig. 10. In the main paper, we discussed two variants of the semantic context module: a) learn visual-to-symbolic node assignment weights and vice-versa from scratch (SCM) and b) use fixed visual-to-symbolic node assignment weights and vice versa (SCM-VG). To obtain the latter we

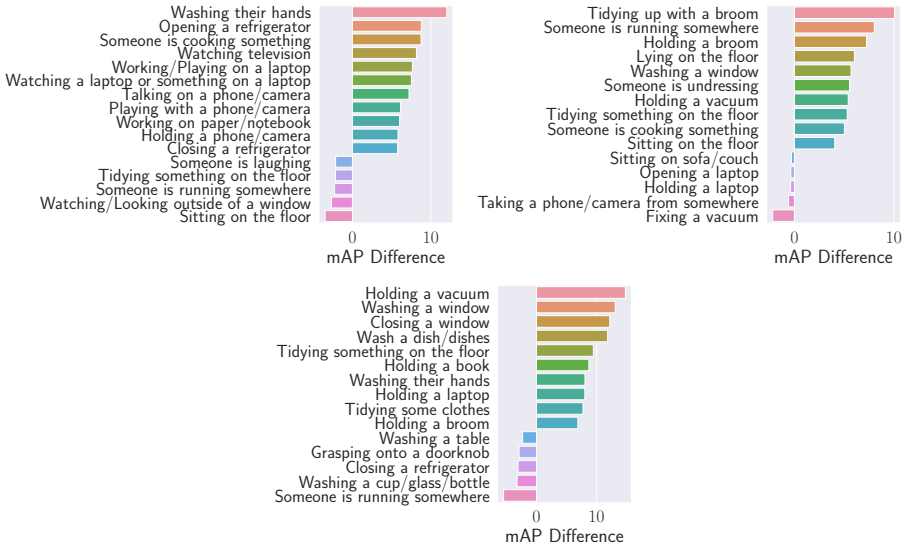

Fig. 3: Extending Fig. 6 of the main paper, we visualize the classes with the highest positive and negative performance difference after adding *object-to-actor spatial* messages. Incorporating spatial structure benefits actions that involve interactions with objects far away from the actor, such as *watching television* or *cooking*. (*bottom*) Adding *actor-to-actor temporal* messages helps with long actions, such as *running*, and actions involving objects that are hard to detect (*Holding a broom*). (c) Adding symbolic graph benefits actions that have a few training examples, such as *Holding a vacuum* or have strong co-occurrences, such as *Holding a book*.

transfer detection knowledge from an object detector that is pre-trained on Visual Genome (VG), inspired by [9]. We find the nearest neighbor from the VG object classes for each of the 431 object classes according to their distances in the embedding space and then we assign visual nodes to symbolic nodes (object classes) by using the corresponding classifier, i.e., the weights and biases, from the last linear layer of the detector.

A.4 Additional Ablation Studies and Qualitative Results

CAD-120. In Figure 1 we show the **contribution of each edge type** on the final sub-activity and object affordance detection performance on CAD-120. Note how adding the object-to-actor-spatial message leads to a significant improvement in the sub-activity detection, due to the usage of improved, context-aware actor features. Object affordance detection is also improved as we keep refining the actor features, since the refinement of object and actor features is performed jointly by taking into account the different edge types. In Fig. 2, we show the **effect of the visual message size** on CAD-120, where we see that our model is not very sensitive with respect to that hyperparameter.

Charades. In Fig. 5 we study the **contribution of each component of the visual context module** to the final performance, in order to validate their necessity. We start with a baseline model that classifies actions per frame based on the local actor features (actor node attributes). Adding a single round of *obj-act-sp* and *act-act-t* messages yields a first significant improvement in the performance (more than 1%). Per frame mAP keeps improving as we perform more rounds of node and edge updates. Adding an edge-type specific attention mechanism for adapting the graph connectivity also benefits our model. Importantly, using the edge features in messages and the attention computation leads to further improvements. Also, comparing our Visual ST-MPNN with the vanilla GNN of Hamilton et al. [2], we get an improvement of 2% (12.1% $\rightarrow$ 13.7%). The result clearly demonstrates the benefit of using attention, edge features, and edge-type specific propagation weights. Evaluating our model with shared weights across edge types for Charades (while still separately normalizing attention over edges of different types) yields an mAP of 13.3%. As we can see, even with shared weights our model outperforms the vanilla GNN, but learning separate weights for the attention mechanism and feature projection leads to further mAP improvement. Finally, the performance is further improved to 15.3% by adding the semantic context module, leading to an absolute improvement of 3% over the vanilla GNN on the challenging Charades dataset.

Ablation results regarding **symbolic graph connectivity and node information** are summarized in Table 1. We find that adding the semantic context module, even with a fully-connected symbolic graph, improves performance compared to only using the visual context module (13.7% $\rightarrow$ 14.9%). Training with more informative edges connecting the symbolic nodes, such as edges based on class co-occurrence or linguistic similarity, slightly improves performance. From this we can conclude that our semantic graph reasoning model can adapt to different types of input symbolic graphs. We further investigate the contribution of the input symbolic graph connectivity and node embeddings. To achieve this, we train our model using as input a co-occurrence symbolic graph with node attributes initialized with word embeddings and then test it using a) the same graph; b) a graph with the same node embeddings but random adjacency matrix (each edge is a Bernoulli trial, with edge probability 0.1) and c) a graph with random node embeddings and random binary adjacency matrix (300-dimensional node embeddings drawn from a $\mathcal{N}(0, I)$ normal distribution). As shown in Table 1, our model's performance significantly degrades when using random edges and/or random node embeddings (15.3% $\rightarrow$ 13.55%), and therefore it has learned to utilize both symbolic graph connectivity and node information.

In Fig. 4 we **visualize the attention** computed along the *object-to-actor spatial* edges, by showing the two object detections that have the highest attention coefficients. As it can be seen, attention focuses on regions that contain relevant context, such as the television, chairs, tables, pots etc. In the second row, we can also see how attention shifts from the kitchen stove to the table, as the person moves. However, not all attended regions are relevant to the action performed by the actor. Furthermore, our model has the tendency to attend

Table 1: Symbolic graph ablation analysis on the Charades [8] dataset. (*left*) Comparison of three symbolic graph types: 1) *co-occurrence*: Co-occurrence adjacency matrix (default), 2) *linguistic similarity*: adjacency matrix based on the cosine similarity of linguistic embeddings, 3) *dense*: fully-connected (complete) symbolic graph. (*right*) Contribution of symbolic graph connectivity and node embeddings. After training our model with an symbolic graph, whose edges encode co-occurrence and whose node attributes are word embeddings, we test with a) the same graph; b) a graph with the same node attributes but random adjacency matrix and c) a graph with random node attributes and random connectivity.

Symbolic graph type	mAP (%)
co-occurrence	15.3
linguistic similarity	15.1
dense	14.9

Symbolic Graph Adjacency/Nodes	mAP (%)
co-occurrence/GloVe embeddings	15.3
random/GloVe embeddings	14.29
random/random	13.55

to large regions, since they provide more context. Since our approach depends on actor/object detections, it might miss relevant small regions, such as the cupboard door, leading to failure cases in activities such as *opening cupboard*, *grasping a doorknob*, *turning off a light*. To address this we combine our method with global scene features.

Furthermore, in Fig. 6 we provide some **sample action predictions** (scores) for 9 frames of 3 videos from the Charades dataset. These predictions are obtained by using only our VS-ST-MPNN, without biGRU and global scene features/temporal dynamics. The proposed model is able to detect fine-grained actions that involve human-object interaction, such as *Drinking from a cup*, *Opening a door*, *Looking outside*, *Walking through a doorway* etc.

ActivityNet Entities. In Fig. 11 and Fig. 12, we show how the initial symbolic node embeddings (Fig. 10), which are shared among all videos, get refined by our Semantic Context Module (*sb-vg* model) for various video segments. Interestingly, symbolic nodes end up with **refined symbolic node features**, which have only a few non-zero feature entries. These entries are similar for video segments of the same scene/event (Fig. 11), but different for video segments of different events (Fig. 12). These symbolic node features, although not directly interpretable, indicate that our semantic context module seems to be performing global semantic reasoning that results in symbolic node features that can help discriminate between different scenes/events.

Fig. 13 illustrates **video captioning results** on sample video segments from the ActivityNet Entities validation set. Adding the semantic context module to the GVD [9] seems to lead to richer captions, capturing more details about the objects in the image.

References

1. Carreira, J., Zisserman, A.: Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4724–4733 (2017). https://doi.org/10.1109/CVPR.2017.502
2. Hamilton, W.L., Ying, R., Leskovec, J.: Representation Learning on Graphs: Methods and Applications. CoRR (2017)
3. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2018). https://doi.org/10.1109/TPAMI.2018.2844175
4. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. International Journal Robotics Research **32**(8), 951–970 (2013). https://doi.org/10.1177/0278364913478446
5. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer International Publishing, Cham (2014)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Neural Information Processing Systems, pp. 3111–3119 (2013)
7. Piergiovanni, A., Ryoo, M.S.: Learning Latent Super-Events to Detect Multiple Activities in Videos. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5304–5313 (2018). https://doi.org/10.1109/CVPR.2018.00556
8. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: European Conference on Computer Vision. pp. 510–526 (2016)
9. Zhou, L., Kalantidis, Y., Chen, X., Corso, J.J., Rohrbach, M.: Grounded video description. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)

Fig. 4: Visualization of attention over objects for updating the feature of an actor on sample frames from Charades dataset. Each pair of images shows: the original frame with the actor detection in green and object detections in blue (*left*) and the actor and the two objects with largest attention coefficients (*right*).

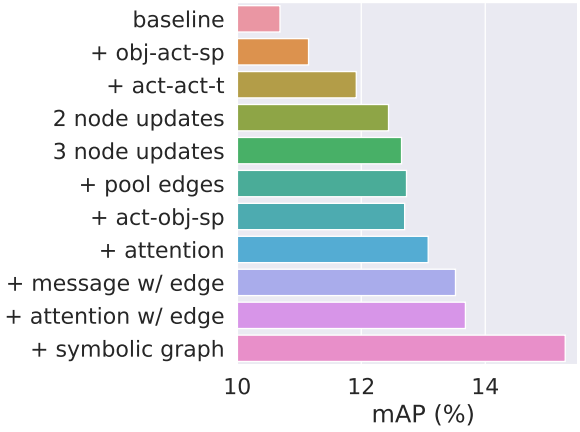

Fig. 5: Performance ablation on Charades when incrementally adding components of our full model, starting with early stage RGB I3D features pooled from actor regions.

Fig. 6: Temporal action predictions of our model (ID: 3) for three videos from the Charades validation set. Action scores for 9 equidistant frames shown for each video.

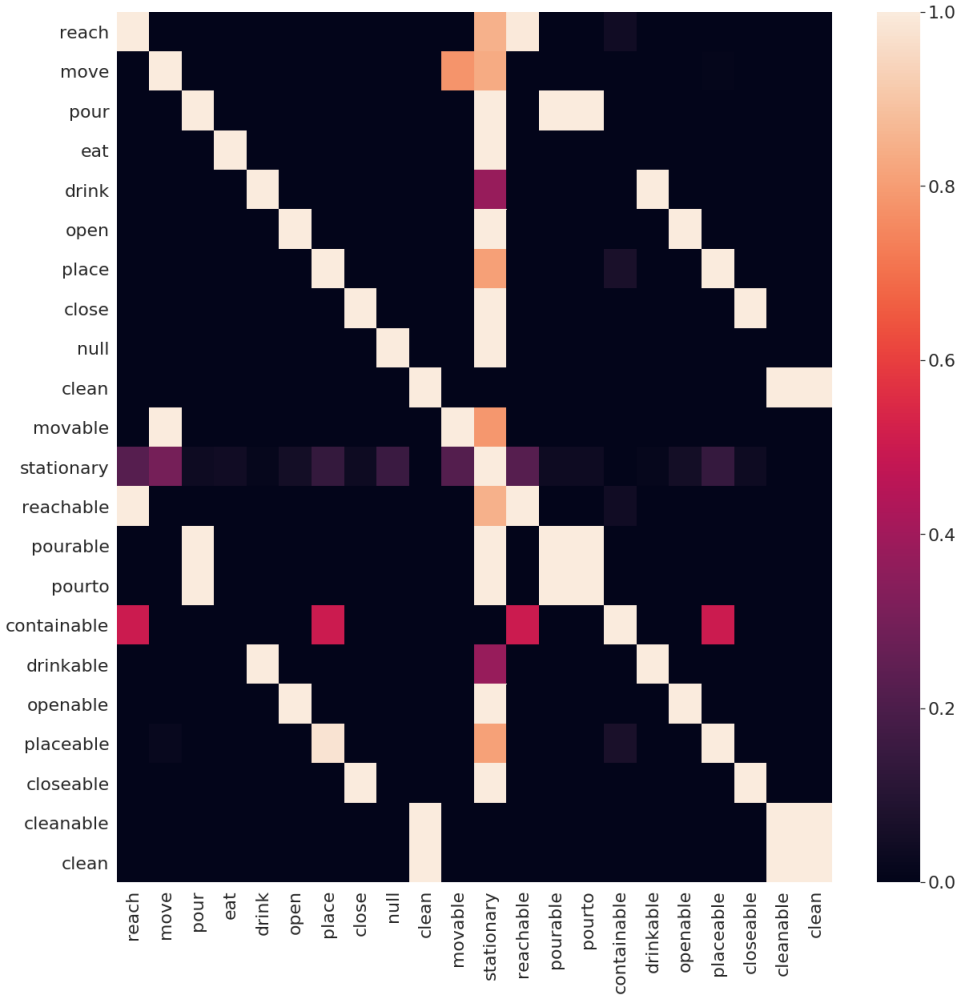

Fig. 7: Symbolic graph adjacency matrix for CAD-120 dataset.

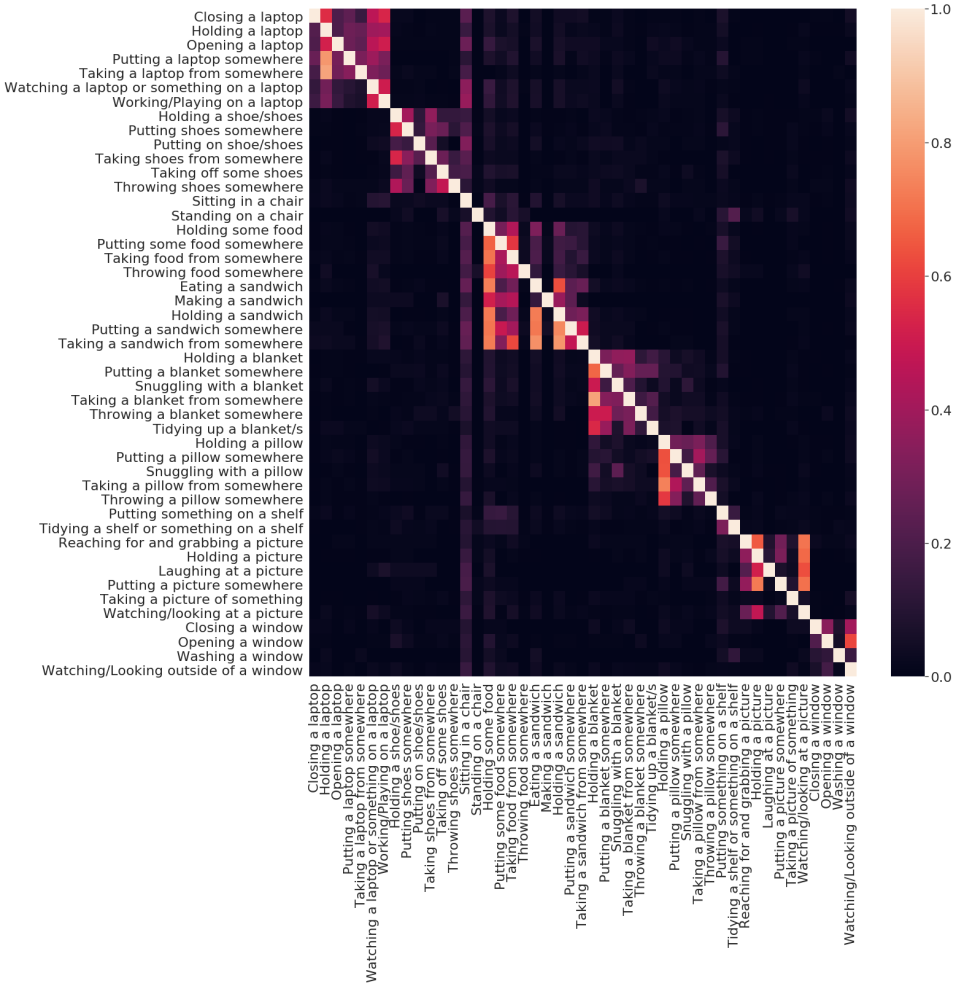

Fig. 8: Illustration of per-frame co-occurrences of a subset of action classes from the training annotations of the Charades dataset.

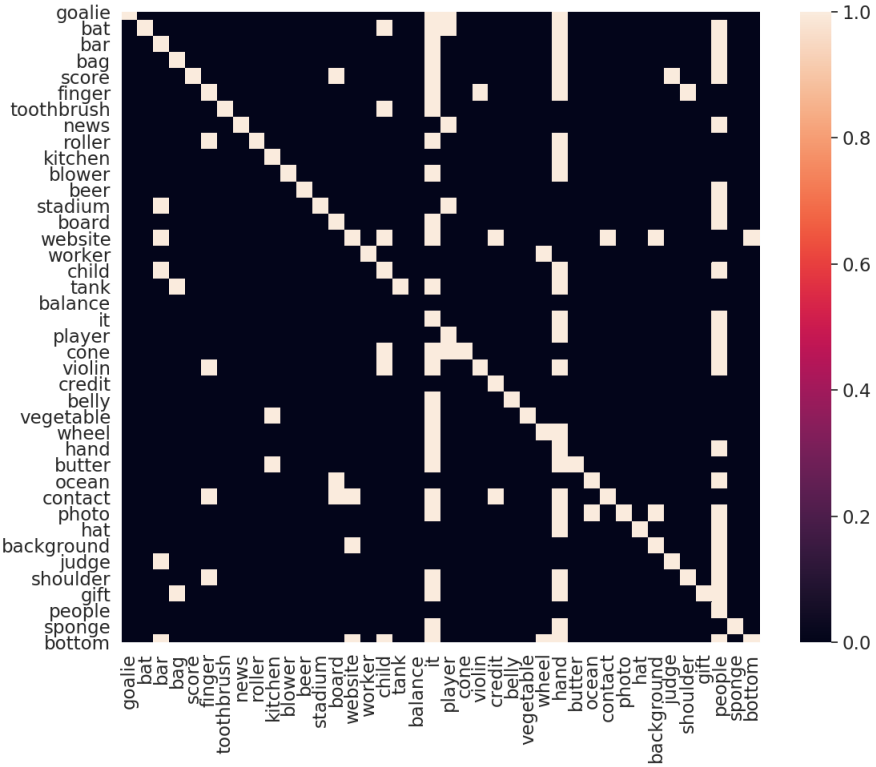

Fig. 9: Illustration of per-sentence co-occurrences of a subset of object classes from the training set of the ActivityNet Entities dataset.

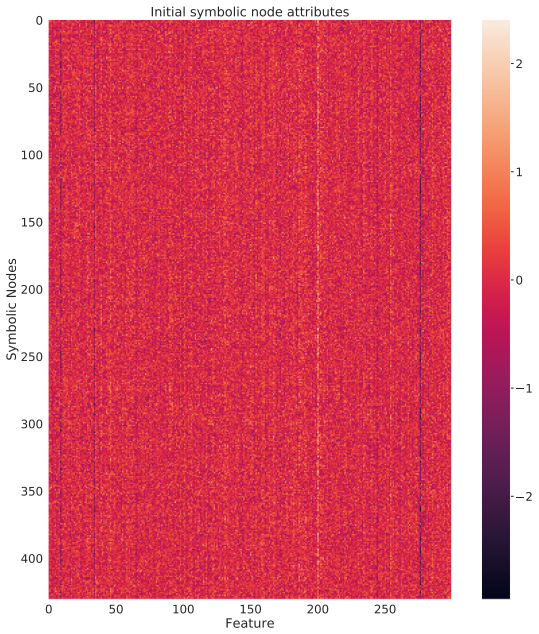

Fig. 10: Visualization of initial symbolic graph node embeddings (word embeddings).

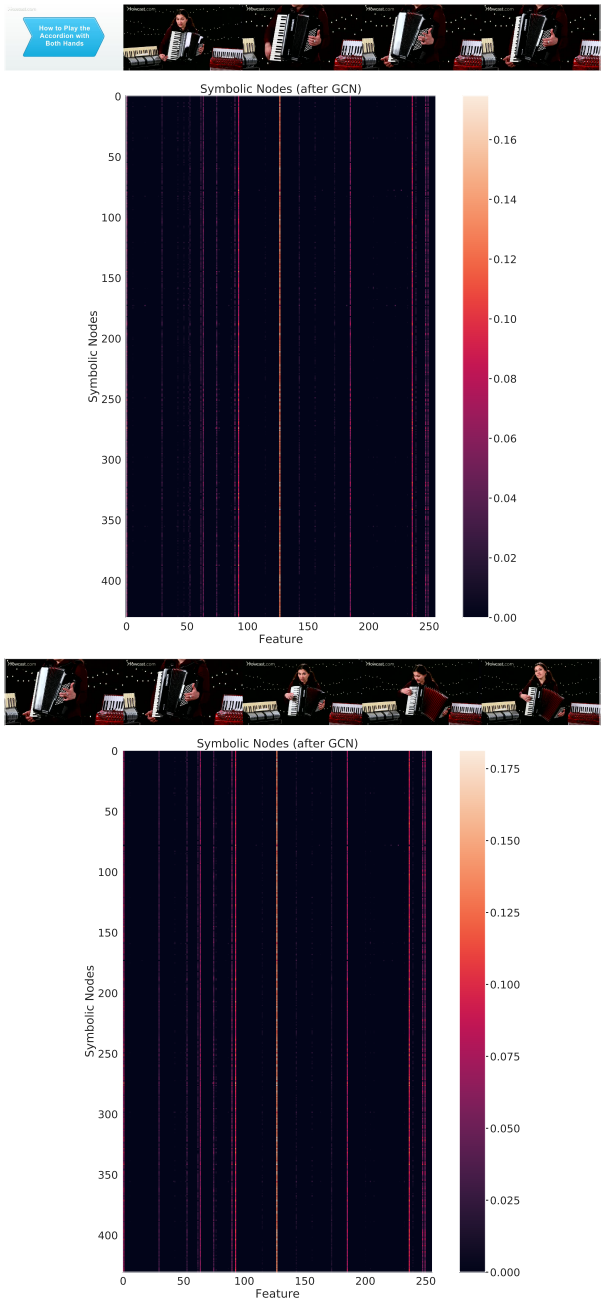

Fig. 11: Visualization of refined symbolic graph node embeddings for two sample ActivityNet Entities video segments that contain similar events (5 frames shown from each segment).

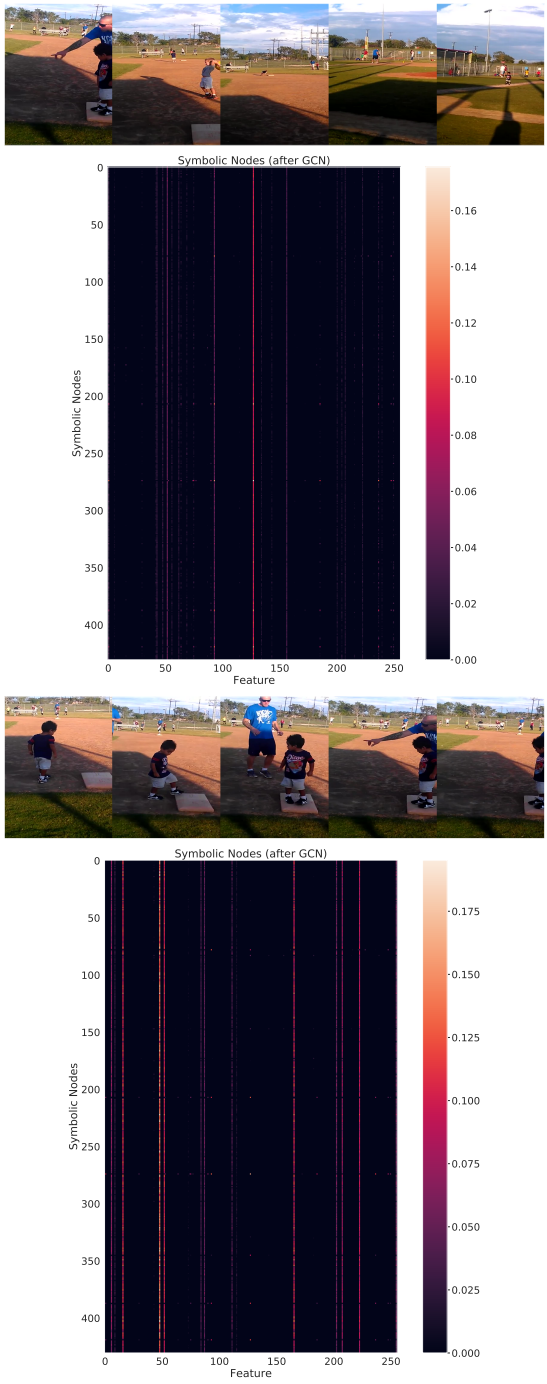

Fig. 12: Visualization of refined symbolic graph node embeddings for two sample ActivityNet Entities video segments that contain different events (5 frames shown from each segment).

GT: A woman is standing in a kitchen while holding a mop.
GVD: A woman is standing in a kitchen
GVD w/ symb: A woman is standing in a kitchen with a mop in her hand

GT: The man fills a white bowl with food.
GVD: The man in the kitchen takes the knife and the man in the black shirt takes the cone and the
GVD w/ symb: The man then grabs a piece of metal and begins to peel the object with the stick

GT: "The boy then jumps on the beam grabbing the bars and doing several spins across the balance beam."
GVD: A man is seen standing on a horse and begins performing a gymnastics routine
GVD w/ symb: He mounts a beam and begins performing a gymnastics routine

GT: He then moves into a hand stand and jumps off the bar into the floor.
GVD: He dismounts , then dismounts and lands on the mat
GVD w/ symb: He dismounts , raising his arms into the air and lands on the mat

GT: A man and a woman sail in a boat down the river while moving the oars.
GVD: People are paddling down a river in a canoe
GVD w/ symb: A group of people paddle in canoes canoes in a canoe

GT: A man is shown tight roping across a small pond and eventually falls in the water.
GVD: A man is standing in a lake
GVD w/ symb: A man is standing in a pool holding a rope

Fig. 13: Video captioning results on sample video segments from the ActivityNet Entities validation set. Adding the symbolic graph reasoning module to the GVD [9] seems to yield richer captions, capturing more details about the objects in the image. *GT*: Ground Truth. *GVD*: Grounded Video Description model. *GVD w/ symb (ours)*: GVD with symbolic graph (5 frames shown from each segment.)