Mixture-Kernel Graph Attention Network for Situation Recognition

Mohammed Suhail[1,2] Leonid Sigal[1,2,3]
[1]University of British Columbia [2]Vector Institute for AI [3]Canada CIFAR AI Chair
suhail33@cs.ubc.ca lsigal@cs.ubc.ca

Abstract

Understanding images beyond salient actions involves reasoning about scene context, objects, and the roles they play in the captured event. Situation recognition has recently been introduced as the task of jointly reasoning about the verbs (actions) and a set of semantic-role and entity (noun) pairs in the form of action frames. Labeling an image with an action frame requires an assignment of values (nouns) to the roles based on the observed image content. Among the inherent challenges are the rich conditional structured dependencies between the output role assignments and the overall semantic sparsity. In this paper, we propose a novel mixture-kernel attention graph neural network (GNN) architecture designed to address these challenges. Our GNN enables dynamic graph structure during training and inference, through the use of a graph attention mechanism, and context-aware interactions between role pairs. It also alleviates semantic sparsity by representing graph kernels using a convex combination of learned basis. We illustrate the efficacy of our model and design choices by conducting experiments on imSitu benchmark dataset, with accuracy improvements of up to 10% *over state-of-the-art.*

1. Introduction

There has been enormous progress in recent years on standard computer vision tasks that reason about objects or actions in isolation, including object categorization [6, 15, 16, 19], object detection [8, 9, 13], and even single image action recognition [33]. However, a more detailed understanding and comprehension of image content, which is needed for many real-world applications, remains a significant challenge. The problem of *situation recognition*, initially proposed by Yatskar *et al*. [32], is an attempt at exploring such more detailed understanding. In situation recognition, the task is to jointly reason about the verbs and a set of semantic-role and entity pairs. Effectively, the goal is to label an image with a set of *action frames*, where each verb-specific frame consists of a fixed set of roles that define the context of the action. Instantiating a frame requires an assignment of values (nouns) to the roles based on observed image content. The resulting frames allow easy structured access to much needed semantic information, *e.g*., who is performing the action, where the action is taking place and what may be the outcome. Figure 1 illustrates an example.

JUMPING	
ROLE	VALUE
AGENT	JOCKEY
SOURCE	LAND
OBSTACLE	FENCE
DESTINATION	LAND
PLACE	OUTDOORS

Figure 1. **Situation recognition** involves understanding the image beyond the salient action. Given the above image of `jumping`, the task is to identify who is jumping `(jockey)`, where is the `agent` jumping to and from `(land)`, what `obstacle` is the `agent` jumping over `(fence)` and where the action is taking place `(outdoors)`.

Yatskar *et al*. [32] proposed the initial large-scale dataset for the task and a baseline model. Later, in [31], they extended this baseline to include a compositional Conditional Random Field model that aims to address one of the major challenges in situation recognition, *semantic sparsity*[1]. This was achieved by encouraging sharing between nouns and data augmentation. In [17], the authors explore the structured nature of the problem and propose a Graph Neural Network (GNN) based architecture that learns to capture pairwise dependencies between roles. Their model, however, is limited and assumes that the interactions between different roles are global, *i.e*. given a pair of roles, the interaction between the roles are independent of the verb. The model also relies on a static fully connected graph structure for training and inference, which fails to account for variable interactions between various role-pairs.

[1] Semantic sparsity here refers to inability of a training dataset to span combinatorial number of possible action frame outputs.

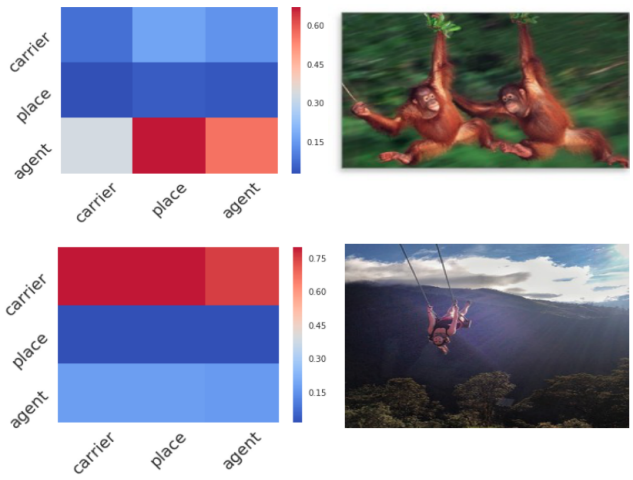

Figure 2. **Variable Conditional Dependencies.** Roles associated with a verb (*e.g.*, `swinging` here) need to propagate information differently depending on the image (right). Note that the predicted graph attention, learned by our model, is indeed different in the two cases (left); suggesting that our model is flexible enough to accommodate such variable interactions within the same verb.

Consider examples illustrated in Figure 2 (right). For the first image, given that the `agent` is a `monkey` and `action` is `swinging`, it is more likely that the `place` where the action is taking place is a `forest` and the `carrier` is a `vine`. Alternatively, for the second image, the presence of a `swing` should intuitively increase the probability that the `agent` is a `person` and that the action is taking place `outdoors`. In other words, for the same action frame instantiated by a verb `swinging`, in the top image the more salient visual component is the `agent`; in the bottom image it is `carrier`. As such, intuitively, the propagation of contextual information should be different in the two cases. However, prior models [17] are not able to accommodate such flexibility and assume fixed flow of contextual information among the roles.

To address these limitations, we propose an extension and generalization of the GNN approach introduced in [17]. In doing so, we make a number of core algorithmic contributions. First, our GNN architecture enables dynamic graph structure during training and inference, through the use of a graph attention mechanism, and context-aware interactions between role pairs. As a result, the graph structure learned by our model can accommodate the intuitions discussed in the last paragraph; see the attention maps in Figure 2 (left). Second, to alleviate semantic sparsity, we construct the kernel matrix (for a given image) through convex combination of a set of basis kernels that are shared across all the images during learning. Intuitively, this facilitates sharing of kernels among related verbs through amortized learning and inference. Additionally, the set of disentangled learned basis kernels aids in modelling the varied flow of information between the nodes in the graph depending on the input. The resulting end-to-end approach results in substantial overall improvement of upto 10% over the state-of-the-art on the *imSitu* benchmark dataset [32].

Contributions: Our main contribution is a novel mixture-kernel graph attention network architecture designed specifically for the problem of situation recognition. Our GNN architecture enables dynamic graph structure during training and inference, through the use of a graph attention mechanism, and context-aware interactions between role pairs. It also alleviates semantic sparsity by learning a set of basis kernels that are shared across all images. We further show that [17] is a special case of our model. Finally, we illustrate efficacy of our architecture and design choices by conducting experiments on *imSitu* benchmark dataset [32]; illustrating overall improvement of upto 10% over the state-of-the-art. We also conduct ablation studies to show the role of individual components.

2. Related Work

Image and video understanding has been widely studied in computer vision. Various tasks such as scene classification [34], activity recognition [20, 28], visual question answering [2], and image captioning [1, 12, 24, 29] have aimed at better understanding of image content. Of recent interest has been the task of situation recognition in images [17, 31, 32] and videos [27], which is the focus of this paper. We review most relevant literature below.

Situation recognition. Situation recognition generalizes the task of activity recognition to include information regarding participating actors, objects, location and their interactions with each other. Yatskar *et al*. [32] introduced the *imSitu* dataset that associates images with a verb, representing the salient action, and a set of semantic role-pair nouns, with entities derived from WordNet [7] and FrameNet [3] respectively. They proposed a baseline CRF model that learns to jointly predict a tuple containing the verb and verb-role-pair triplets. In a follow-up work [31], they address the issue of semantic sparsity in the dataset and propose a tensor composition function along with methods to augment the dataset to further enhance performance. Mallaya *et al*. [21], pose the situation recognition task as a sequential prediction of noun entities corresponding to the image verb in an arbitrary but fixed order. Under such a setup, they train an LSTM network to predict the sequence of nouns related to given roles in the frame and show how a model trained for the task of situation recognition can be used to caption images and answer questions regarding the image. Li *et al*. [17] further generalize this setup and use a Graph Neural Network to propagate information between roles and to remove the sequential dependencies between them.

Graph Neural Networks. Graph neural networks are neural nets that operate on graphs and structure their computation accordingly. Applying convolution on graphs is non-trivial due to the large variance in their structure. Various approaches have been proposed that aim to solve this task. Henaff *et al*. [10], makes use of graph convolutions from spectral graph theory to define parametrized filters similar to Convolutional Neural Networks. Defferard *et al*. [5] approximate the filters in the spectral domain using Chebyshev polynomials in order to bridge the gap between fast heuristics and slow spectral approach. Kipf and Welling [14] introduce further simplifications to spectral convolution framework, which allows convolutions on a graph by applying feedforward neural networks to the nodes in a recurrent manner. Velickovic *et al*. [26] leverage masked self-attention layers that learn to attend to neighborhood information and thereby address previous shortcomings.

Recently there has been a rapid growth in the popularity and scope of graph neural networks in the field of computer vision for tasks like object detection [11] and image classification [4]. In our work, we build on Gated Graph Neural Networks [18] that address the issue of contraction map assumption in [23], by updating the node states using a gating function. We also apply attention [26] over local computation, to allow more flexible propagation. Finally, we introduce a novel mixture-kernel which facilitates information sharing and amortized inference and learning.

3. Approach

The *imSitu* dataset assumes discrete sets of verbs $\mathcal{V}$, nouns $\mathcal{N}$ and frames $\mathcal{F}$ for situation recognition. Each frame $f \in \mathcal{F}$ is associated with a set of semantic roles $\mathcal{E}_f$. Semantic roles, $e \in \mathcal{E}_f$, are associated with a noun $n_e \in \mathcal{N} \cup \{\phi\}$, where ϕ specifies that the noun is either unknown or not applicable. The set of pairs of semantic roles and nouns are referred to as a realized frame, $\mathcal{R}_f = \{(e, n_e) : e \in \mathcal{E}_f\}$. Given an image, the task then is to predict $S = (v, \mathcal{R}_f)$, where $v \in \mathcal{V}$ is the salient action corresponding to the image and $\mathcal{R}_f$ its corresponding realized frame. For example, consider the image in Figure 1. The realized frame corresponding to the verb `jumping` consists of five role-value pairs *i.e*. {`(agent, jockey), (source, land), (obstacle, fence), (destination, land), (place, outdoor)`}.

Situation recognition as graph inference task. Verbs and semantic roles corresponding to an image are strongly dependant on one another. In order to model such dependencies, we present the task of situation recognition as a graph based inference problem. Given an instance from the dataset, we instantiate a graph $\mathcal{G} = (\mathcal{A}, \mathcal{B})$. The nodes in the graph $a \in \mathcal{A}$ represent the roles associated with the image and assume values from $\mathcal{N}$. The edges in the graph $b \in \mathcal{B}$, directed or undirected, encode the dependencies between the roles.

3.1. Graph Neural Networks

In our work, we use Gated Graph Neural Networks, for predicting the situation given an image. Figure 3 shows an overview of the proposed model architecture. Given an image, I, with associated verb $v \in \mathcal{V}$ we first instantiate a graph $\mathcal{G}_I = (\mathcal{A}, \mathcal{B})$, where the number of nodes, $|\mathcal{A}|$, is equal to the number of roles in the frame $\mathcal{E}_f$ corresponding to v. The hidden states of the nodes, $a \in \mathcal{A}$, are initialized as

$$h_a^0 = \text{ReLU}(W_{in}\phi_n(i) \odot W_e e \odot W_v \hat{v}), \tag{1}$$

where $\phi_n(i)$ are the features obtained from the penultimate fully connected layer of a VGG-16 network trained to predict the nouns in a given image, $\hat{v}$ and e correspond to the one-hot encoding of the predicted verb and role corresponding to the node. W_e and W_v are the role and verb embedding matrices respectively. W_{in} is a transformation matrix that maps the features obtained from the CNN to the space of hidden representation of the graph and $\odot$ is an element wise multiplication operation. The hidden states of each node are updated in a recurrent manner. At time t, the propagation of information between the nodes is governed by:

$$\begin{aligned} x_a^t &= \text{ACCUMULATE}^{(t)}\left(\{h_{a'}^{t-1} : a' \in \mathcal{N}_a\}\right) \\ h_a^t &= \text{COMBINE}^{(t)}(h_a^{t-1}, x_a^t), \end{aligned} \tag{2}$$

where $\mathcal{N}_a$ is the set of neighbours of a. The choice of ACCUMULATE$^t(\cdot)$ and COMBINE$^t(\cdot)$ dictate the power of the Graph Neural Network [30] and is therefore crucial. In our model, we formulate the ACCUMULATE function as:

$$\begin{aligned} x_a^t &= \sum_{(a,a')\in B} \alpha_{aa'} \left(\sum_{k=1}^{d} c_k W_k\right) h_{a'}^{t-1} \\ &= \sum_{k=1}^{d} c_k \left(\sum_{(a,a')\in B} \alpha_{aa'} W_k h_{a'}^{t-1}\right) \end{aligned} \tag{3}$$

where W_k are basis kernels that are used for modeling the interaction between the nodes, c_k are the associated weights, d is the dimension of the kernel vector space and $\alpha_{aa'}$ are weights corresponding to the edge between the nodes a and a'. The edge weights are derived using an attention mechanism similar to [26]. Representing the kernel as a mixture of basis kernels allows us to decompose our graph into a set of independent graphs that learns a disjoint set of embeddings for the nodes. The details for learning the basis kernels are described in Section 3.3.

The COMBINE step is formulated using a gated mecha-

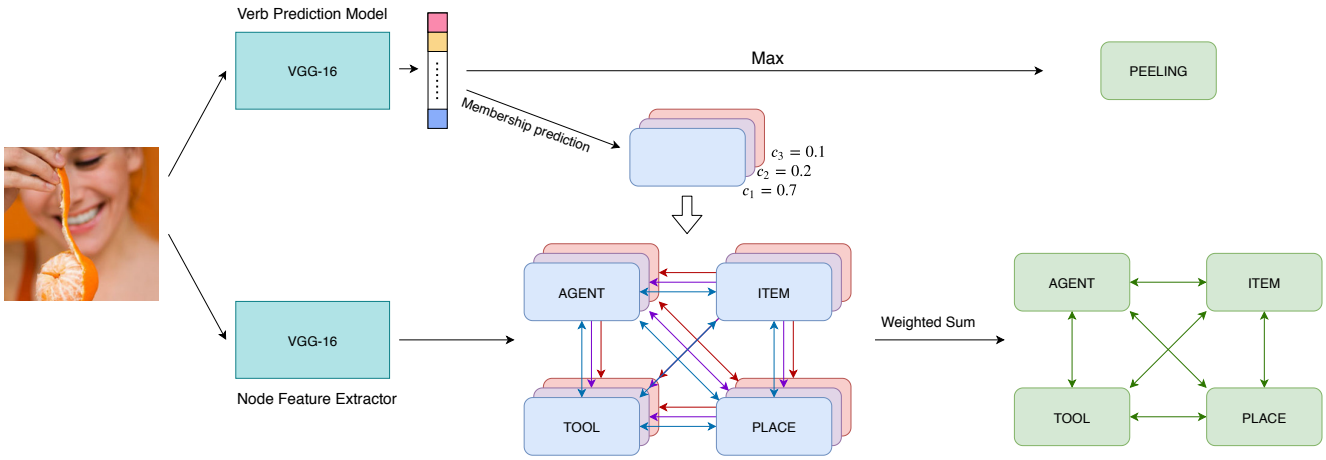

Figure 3. **Model Architecture.** Figure show the overall architecture of the model (for $d = 3$). The image is passed though a verb prediction model to obtain the probability distribution over possible actions (verbs) in the image. We then transform into the kernel vector space dimension to obtain the membership weights of the basis. A graph is instantiated with each of the basis kernel followed by information propagation between the nodes. The graphs are then summed up using the learned membership weights to obtain the final node features.

nism similar to [18] as:

$$\begin{aligned} z_a^t &= \sigma(W_z \cdot [h_a^{t-1}, x_a^t]) \\ r_a^t &= \sigma(W_r \cdot [h_a^{t-1}, x_a^t]) \\ \tilde{h}_a^t &= \tanh(W_h \cdot [r_a^t \odot h_a^{t-1}, x_a^t] \\ h_a^t &= (1 - z_a^t) \odot h_a^{t-1} + z_a^t \odot \tilde{h}_a^t \end{aligned} \tag{4}$$

where r_a and z_a are the reset and the update gates, W_z, W_r, W_h are the weights of the update functions. Such a state update mechanism allows information to be combined slowly, ensuring that information from previous time steps is not lost. After some steps, T, of propagation, the updated hidden states are used to predict the nouns corresponding to each of the roles as

$$p_{e:n} = \sigma(W_c h_{a_e}), \tag{5}$$

where $p_{e:n}$ is the noun associated with the role e, σ is a softmax function and W_c are the weights for the noun classifier. The classifier is shared across all the nodes to address the issue of semantic sparsity inherent to the problem.

Loss. The *imSitu* dataset provides three sets of annotations for each image. We accumulate the cross-entropy at the noun nodes for all annotations, where $y_{e:n}$ is the ground truth noun for role e corresponding to image i:

$$L = \sum_i \sum_{j=1}^{3} \left(\frac{1}{|E_f|} \sum_e y_{e:n} \log(p_{e:n}) \right). \tag{6}$$

During inference, we first predict the distribution over all possible verbs and pick the output with the highest probability as the salient action. The nodes in the graph corresponding to the image are then constructed by retrieving the set of roles associated with its verb and their states are initialized using (1). The kernel for the graph is then constructed from a convex combination of the basis kernels with their weights derived from the softmax layer of the verb prediction model. The features of the hidden states are refined by propagating and accumulating information from the neighboring nodes for a fixed number of time steps T in accordance with (3). The final output is then obtained by selecting the nouns corresponding to the highest score from the softmax output of the noun classifier.

3.2. Dynamic Graph Structure

The interaction between the nodes in the graph vary depending on what roles are associated with it. For example, in Figure 2, given the presence of a `monkey` in the image, it is highly likely that the `carrier` of the action is a `vine`. Similarly, we can deduce that the `agent` is most likely to be a `person` given that the `carrier` is a `swing`. Such conditional dependencies between role pairs can be hard to characterize as they vary even for a single verb. By modelling the edge weights to be function of node features that they connect, we can model such dependencies easily. Specifically, in our model, we learn the edge weights $\alpha_{aa'}$ by using an attention mechanism similar to [26]. The weights $\alpha_{aa'}$ are calculated as:

$$\begin{aligned} e_{aa'} &= a(W_{attn} h_a, W_{attn} h_{a'}) \\ \alpha_{aa'} &= \frac{\exp(e_{aa'})}{\sum_{\{a'' \in \mathcal{N}_a\}} \exp(e_{aa''})} \end{aligned} \tag{7}$$

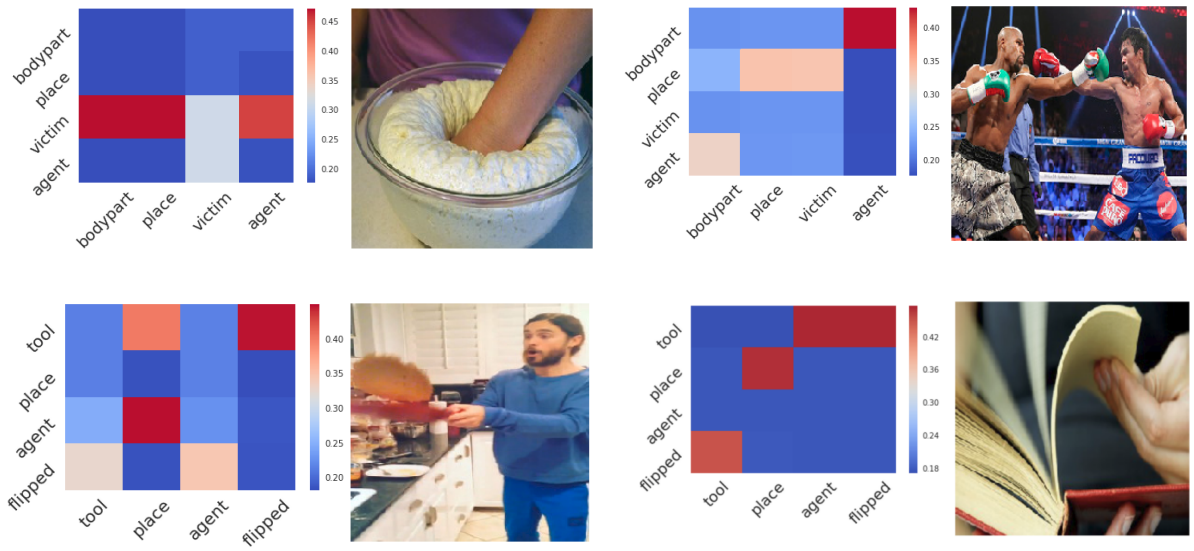

Figure 4. **Learned Adjacency Matrices.** The variable interaction between the role nodes is modeled explicitly using an attention mechanism where the attention score correspond to the entries in the adjacency matrix. In the figure above we visualize the adjacency matrix generated during inference for two different instances of the verbs punching (top) and flipping (bottom).

where W_{attn} is the attention kernel, a is the attention mechanism and $\mathcal{N}_a$ is the set of all neighbors of a in the graph. The weights $\alpha_{aa'}$ force the model to learn the relative importance of nodes w.r.t. a given node and thereby propagating information that is only relevant to it.

The attention mechanism learned by the model is visualized in Figure 4. We observe that the learned edge weights correspond to the relative importance of the nodes w.r.t. each other. Also the dependencies learned vary depending on the image context. For instance, the top row shows two examples for the verb `punching`. In the first case, given that the `victim` is `dough`, it is highly likely that the `place` is `kitchen` and that the `agent` is a `cook` as opposed to the second image, where information about the `place` – `boxing ring` suggest that the `victim` is a `boxer` and the `body part` a `face` suggest `agent` is a `boxer`. The bottom row depicts two examples of `flipping`. In the first image, knowing that a `pancake` is being `flipped` increases the probability of the `tool` being `pan` and `agent` being a `man`. Also, the presence of a `pan` indicates that the `place` is most likely a `kitchen`. Similarly, in the second image, knowing that the `book` is being `flipped` makes it very likely that the `tool` used is a `hand`. Likewise, the information about `tool` helps conclude that the `agent` is a `person`.

It can be observed that the diagonal entries of the adjacency matrix are generally low as compared to the rest. This is due to the soft-update that is applied to the hidden state features. Each node initially contains its own features and retains them while incorporating new information from the neighboring nodes.

3.3. Context-aware interaction

Depending on the action in the image, the interaction between the role nodes can differ. For example, the interaction between `agent` and `place` for `skidding` is different from what it would be in case of `repairing`. While different edge weights reason about the relative importance of nodes with respect to one another, the actual transformation of features into valuable information is achieved by the graph kernel. Modeling the graph neural network with a fixed global propagation matrix/kernel will force the model to learn a generalized view of information entanglement between the role nodes. It is possible that the attention mechanism could learn weights to adjust for variable information propagation, but as seen in Figure 2 the variance in interaction is high even within a given verb set and hence the model can benefit from the decoupling of image based and role-based interactions as shown in Section 4.1.

To incorporate such image dependant interaction, we model the kernel matrix as a convex combination of basis kernels. For a given image, the model is then trained to learn a set of membership weights for each of the basis kernel. To determine the dimension of the kernel vector space, we study the semantic similarity between the 512 verbs in the dataset. Based on the similarity of the role frames, we find that the verbs can be divided into 252 groups, with verbs in each group sharing the same role frame. Such a grouping is logical as role frames in the imSitu dataset are based on FrameNet [3] which is derived from theory of meaning (a.k.a., frame semantics) in computational linguistics. Frames in the FrameNet are, by design, meant to provide generic semantic representation. As such, verbs that share

frame definitions tend to be semantically similar. Given the kernel vector space, the propagation kernel (W_I) for an image (I) is obtained as

$$W_I = \sum_{k=1}^{252} c_k W_k, \tag{8}$$

where W_k are the basis kernels and c_k are their corresponding weights. The weights c_k are obtained from the softmax layer of the verb-prediction model followed by a differentiable transformation to a 252 dimensional space. This allows us to train the entire network (verb and frame prediction model) in an end-to-end manner.

While one can, as an alternative, use a single kernel for each verb, this is not advisable. First, semantic sparsity is a problem intrinsic to the task of situation recognition; as such, sharing parameters across similar verbs is beneficial. Second, certain images may not be categorized as belonging to a single verb. In both cases, constructing a dynamic kernel conditioned on the image content is very helpful.

Relationship to Other Models. The graph neural network based model proposed in [17] can be viewed as a special case of our model where the graphs are generated statically with all edge-weights, $\alpha_{aa'}$ set to one and a single kernel is shared over all the images in the dataset, *i.e.* $W_k = W_p$ and $c_k = \frac{1}{d}$ for $1 \leq k \leq d$. Under such restrictions, the information propagation then takes the form

$$x_a^t = \sum_{(a',a)\in B} W_p h_{a'}^{t-1}. \tag{9}$$

4. Experiments

Implementation Details. To initialize the graph network we have to extract features that are relevant to the verb and nouns present in the image. For this, we use two pre-trained VGG-16 networks [25], and fine-tune all the layers. The first network is trained to predict verbs given an image from the *imSitu* dataset which is then used to determine the structure of the graph network. The second network is fine-tuned on the task of multi-label classification, where given an image the model predicts all the role-values associated with it. We then remove the last fully connected layer to obtain the feature vector $\phi_n(i)$, in (1). The network, including the verb prediction and role-frame prediction model are trained together in an end-to-end manner.

For all experiments, we fix the number of propagation steps T to 4 throughout training and testing. Since all the nodes are connected to each other initially and the graphs can have a maximum of 6 nodes, propagating information for more number of step will not provide a significant boost in the model performance; see the effect of T in Table 1.

Optimization. All the models are implemented in PyTorch [22]. The models are trained using Adam, with a learning rate of $3e-4$ and mini-batch size of 64 for the single kernel models. For training the mixture-kernel model, we use an initial learning rate of $3e-4$ while pre-training and $3e-6$ while fine-tuning for the basis kernels.

	T= 0	T= 1	T= 2	T= 3	T= 4
Value	58.13	67.72	70.56	72.18	72.93

Table 1. Performance as a function of propagation steps T.

Evaluation. We use the standard data split for *imSitu* with 75k annotated images in the train set, 25k in development and 25k in the test. Each image is associated with three different annotations. During testing, we compare with all the available annotations and report the final score.

Following [32], for each of top-1 and top-5 predicted verbs, we report three metrics: (a) *verb:* the performance of the verb prediction model, (b) *value:* measures the performance of role-value tuple prediction which is considered to be correct if it matches any one of the tuples among the three ground truth annotations, (c) *value-all:* measures the realized-frame prediction performance, which is considered correct if all the role-value pairs match any one of the given ground truth annotations completely. When the ground truth verb is provided we report the *value* and *value-all* metric. This provides an upper bound on the performance of the role frame prediction model. Finally, we report the *mean*, the average of all the scores over top-1 predicted, top-5 predicted, and ground truth verbs.

4.1. Results and Analysis

Comparison with state-of-art We compare the performance of our model against previous approaches, mainly [17, 21, 31, 32], on both the dev and test set in Table 2.

Our verb-prediction model performance is significantly better than all the previous methods. We attribute this to the implicit use of loss from frame prediction as an auxiliary loss to train the verb-prediction model. We achieve the best performance on all the metrics when using top-1 predicted, top-5 predicted and ground truth verbs. For the value metric, our model yields 8% improvement when using top-1 predicted verb, 10% with top-5 predicted verb and nearly 2% when the ground truth verbs are provided along with the test image. Our model also achieves a 4% improvement in the mean score.

We note that *Fully-connected Graph* uses beam search over the predicted verbs in order to improve performance for value metric when ground truth is not available and *Tensor-Composition + reg* uses semantic augmentation to obtain more training data. Despite not doing either of these, our model achieves the best performance. This improvement can be attributed to the use of a dynamic graph structure, that is defined by the roles and verb, and a context-aware kernel construction that allows for more efficient propagation of information.

		top-1 predicted verb			top-5 predicted verb			ground truth verb		
		verb	value	value-all	verb	value	value-all	value	value-all	mean
dev	CNN + CRF [32]	32.25	24.56	14.28	58.64	42.68	22.75	65.90	29.50	36.32
	Tensor-Composition + reg [31]	34.20	25.39	15.61	62.21	46.72	25.66	*70.80*	34.82	39.57
	Fusion, VGG + RNN [21]	36.11	*27.74*	16.60	*63.11*	*47.09*	26.48	70.48	35.56	40.40
	Fully-connected Graph [17]	*36.93*	27.52	*19.15*	61.80	45.23	*29.98*	68.89	*41.07*	*41.32*
	Ours	**43.21**	**35.18**	**19.46**	**68.55**	**56.32**	**30.56**	**73.14**	**41.68**	**46.01**
test	CNN + CRF [32]	32.34	24.64	14.19	58.88	42.67	22.55	65.66	28.69	36.25
	Tensor Composition + reg [31]	34.12	26.45	15.51	62.59	46.88	25.46	*70.44*	34.38	39.48
	Fusion, VGG + RNN [21]	35.90	27.45	16.36	*63.08*	*46.88*	26.06	70.27	35.25	40.16
	Fully-connected Graph [17]	*36.72*	*27.52*	*19.25*	61.90	45.39	*29.96*	69.16	*41.36*	*41.40*
	Ours	**43.27**	**35.41**	**19.38**	**68.72**	**55.62**	**30.29**	**72.92**	**42.35**	**45.91**

Table 2. We compare our situation recognition model against current state-of-the-art on the development and test set. Our model achieves the best performance on all metric for top-1 predicted, top-5 predicted and ground truth verbs. We also showcase a significant improvement in the mean score over all the previously proposed models. The best performances are shown in **bold** and second best in *italics*.

		top-1 predicted verb		top-5 predicted verb		ground truth verb		
		value	value-all	value	value-all	value	value-all	mean
test	GGNN	31.16	14.34	53.69	25.23	67.32	37.32	38.19
	GGNN + attn	33.64	16.00	54.59	26.90	69.40	38.83	39.89
	GGNN + multi-kernel	33.35	15.78	54.04	26.32	67.96	38.21	39.27
	GGNN + mutli-kernel + attn	34.83	17.52	54.91	27.85	71.19	39.68	40.99
	GGNN + mixture-kernel + attn	**35.41**	**19.38**	**55.62**	**30.29**	**72.92**	**42.35**	**42.66**

Table 3. We study the impact of different components on the model performance. The best performances are shown in **bold**

Ablation Study. We study the effect of different components of our model in Table 3. As expected the model performs the best with both the mixture-kernel and attention component and provides a boost of 5% in value metric when ground truth verbs are provided and an increase of nearly 4% on the mean metric as compared to the naive GGNN. Removing either of components leads to performance loss.

We also experiment on a different variant (multi-kernel model) in which we use a hard assignment of kernel given the verb in an image. For the multi-kernel model, we first group together verbs based on their role-frame similarity and assign a kernel for each group. The ACCUMULATE function under such a setup is given by:

$$x_a^t = \sum_{(a,a')\in B} \alpha_{aa'} W_{p_v} h_{a'}^{t-1} \tag{10}$$

where p_v refers to the verb associated with the image and W_{p_v} is the kernel assigned to the verbs in the verb group p_v. This corresponds to the case where $c_i = 1$ when $i = p_v$ and 0 otherwise in (3). The boost in performance when adding the multi-kernel is less than that of the attention mechanism. This is due to a large number of kernels in the model and insufficient data to optimize some of the kernels. The mixture kernel model is oblivious to this issue as every basis kernel is trained using all the examples in the training set.

	Basis weights	
	Predicted	Random
Value	72.96	64.26
Value-all	42.35	26.85

Table 4. Effect of using the random weights for the basis kernel.

In order to ensure that the basis kernels do learn something that is semantically meaningful, we conduct an ablation experiment where we randomly permute the weights (c_i) of the basis during test time and study its impact on the performance. As shown in table 4, we observe a drop in value and value all metric of about 6% and 16% respectively when the ground truth verbs are provided.

Finally, we measure the impact of removing random edges from the graph. We show the value score obtained after randomly removing 25%, 50% and 75% of the edges from the graph with ground truth verbs provided in Table 5.

Edges removed	25%	50%	75%
Value	68.34	65.81	61.18

Table 5. Effect of removing edges from the graph.

Qualitative Analysis. Figure 5 shows some of the predicted situations for instances in the test set. The top two rows showcase several examples of instances where our model

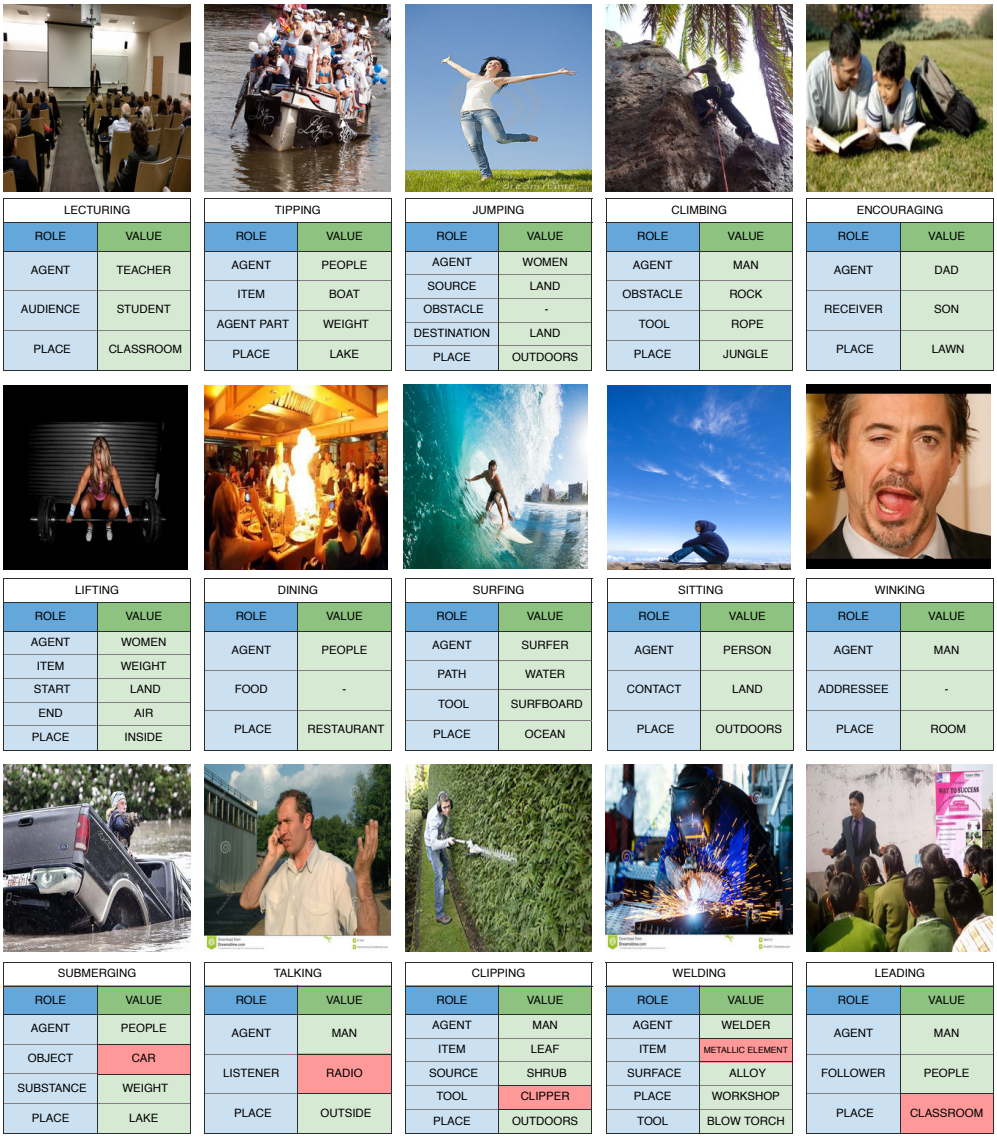

Figure 5. **Qualitative results.** The **top two row** show results where our model gets all the role-pair predictions correct. The **bottom row** depicts examples which contain typical errors in prediction of the role values.

predicts the frame correctly. The bottom row contains examples where our model makes some errors in predicting the role values. While the first two instances do predict nouns incorrectly, the predictions for the latter three are in fact reasonable. For example in the case of `clipping`, our model predicts that the `tool` used is a `clipper`, however, the ground truth annotation labels it as a `hedge trimmer`. Similarly, for `welding` our model predicts that the item being used as a `metallic element` which is not too far from the ground truth label, `alloy`, in terms of meaning. For the last image, our model predicts the `place` as `classroom` which is more informative given the image as opposed to `outside` or `outdoor` in the ground truth.

5. Conclusion

We present a model that learns to recognize the situation in a given image by predicting its salient action along with participating actor, objects, locations, and their interactions. Our approach learns to model varying interactions between role nodes depending on the roles themselves and the image context through the use of mixture kernel graph attention network. On *imSitu* dataset our model leads to improvements of up to 10% in value metric and a 4% improvement overall on average. We also present analysis of how different components in the model impact the performance.

Acknowledgments: This work was funded in part by the Vector Institute for AI, Canada CIFAR AI Chair, NSERC CRC and an NSERC DG and Discovery Accelerator Grants.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6, 2018. 2

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 2

[3] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998. 2, 5

[4] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7239–7248, 2018. 3

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016. 3

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1

[7] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. 2

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[10] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015. 3

[11] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2018. 3

[12] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 2

[13] Kye-Hyeon Kim, Sanghoon Hong, Byungseok Roh, Yeongjae Cheon, and Minje Park. Pvanet: deep but lightweight neural networks for real-time object detection. *arXiv preprint arXiv:1608.08021*, 2016. 1

[14] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. 3

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 1

[16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 1

[17] Ruiyu Li, Makarand Tapaswi, Renjie Liao, Jiaya Jia, Raquel Urtasun, and Sanja Fidler. Situation recognition with graph neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4173–4182, 2017. 1, 2, 6, 7

[18] Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR'16*, April 2016. 3, 4

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[20] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950, 2016. 2

[21] Arun Mallya and Svetlana Lazebnik. Recurrent models for situation recognition. *arXiv preprint arXiv:1703.06233*, 2017. 2, 6, 7

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[23] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 3

[24] Parth Shah, Vishvajit Bakrola, and Supriya Pati. Image captioning using deep neural architectures. In *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017 International Conference on*, pages 1–4. IEEE, 2017. 2

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. 3, 4

[27] Paul Vicol, Makarand Tapaswi, Lluís Castrejón, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8581–8590, 2018. 2

[28] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: region convolutional 3d network for temporal activity detection. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 5794–5803, 2017. 2

[29] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 2

[30] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. 3

[31] Mark Yatskar, Vicente Ordonez, Luke Zettlemoyer, and Ali Farhadi. Commonly uncommon: Semantic sparsity in situation recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 6, 7

[32] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5534–5542, 2016. 1, 2, 6, 7

[33] Zhichen Zhao, Huimin Ma, and Shaodi You. Single image action recognition using semantic body part actions. In *2017 IEEE International Conference on Computer Vision (ICCV), Venice*, pages 3411–3419, 2017. 1

[34] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 2