

Knowledge-aware Global Reasoning for Situation Recognition

Weijiang Yu, Haofan Wang, Guohao Li, Nong Xiao, Bernard Ghanem

Abstract—The task of situation recognition aims to solve the visual reasoning problem with the ability to predict the activity happening (salient action) in an image and the nouns of all associated semantic roles playing in the activity. This poses severe challenges due to long-tailed data distributions and local class ambiguities. Prior works only propagate the local noun-level features on one single image without utilizing global information. We propose a **Knowledge-aware Global Reasoning (KGR)** framework to endow neural networks with the capability of adaptive global reasoning over nouns by exploiting diverse statistical knowledge. Our KGR is a local-global architecture, which consists of a local encoder to generate noun features using local relations and a global encoder to enhance the noun features via global reasoning supervised by an external global knowledge pool. The global knowledge pool is created by counting the pairwise relationships of nouns in the dataset. In this paper, we design an action-guided pairwise knowledge as the global knowledge pool based on the characteristic of the situation recognition task. Extensive experiments have shown that our KGR not only achieves state-of-the-art results on a large-scale situation recognition benchmark, but also effectively solves the long-tailed problem of noun classification by our global knowledge.

Index Terms—Graph Reasoning, Knowledge-guided Learning, Situation Recognition, Long-tailed Problem.

1 INTRODUCTION

SITUATION recognition [2] has attracted more and more attention, as it provides a deeper understanding of the image and thus facilitates various vision tasks ranging from fundamental classification [3], [4] to high-level reasoning tasks [5]. The situation recognition task aims to address what is happening in an image, who is playing a part in the situation, and what the relevant nouns are. In many real-world applications, we need more global knowledge to understand the scene. For instance, pesticides can be used to kill insects in crops as shown in Fig. 1 (a, left).

When humans recognize a specific situation [6], each noun is not identified individually. Prior knowledge helps to make a correct identification by considering global semantic coherence. For example, humans can recognize the obscured object in Fig. 1 (c) as a “salt” based on two knowledge concepts: 1) the “salt” often appears with “bowl” in the “kitchen” (pairwise relationship knowledge); 2) the obscured object is similar to “salt” as we saw before in the “Cooking” situation (action-guided relationship knowledge).

Obviously, situation recognition can benefit from such structured knowledge. However, the dominant state-of-the-art situation recognition methods [7], [8], [9] either predict each noun by relying on end-to-end category-based supervision, or build local correlations of nouns implicitly within a constrained image [1], [10]. As shown in Fig. 1

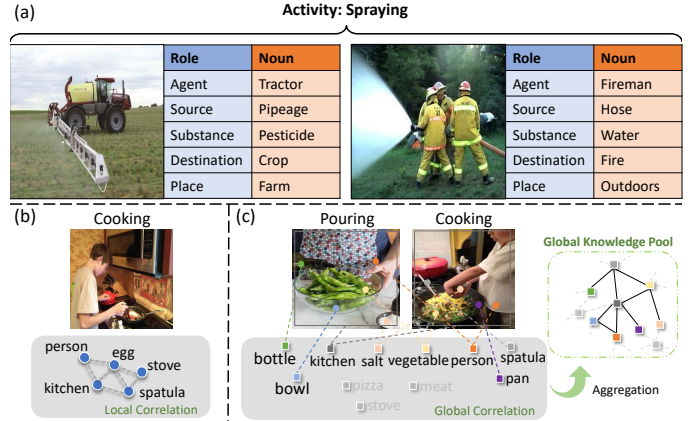


Fig. 1: (a) The example for situation recognition task, which aims to predict the salient action (e.g., spraying) and associated nouns (e.g., crop and farm) at the same time. (b) The previous methods [1] model local correlation of nouns in a single image without any global supervision. (c) Humans can still recognize the tiny object “salt” and location “kitchen” in the “Pouring” scene. This is because (1) this object looks similar to the “salt” as we saw before in the “Cooking” situation; (2) the “salt” often appears in the “kitchen” in the “Cooking” situation. Thus, it inspires our design of Global Knowledge Pool. Such rich human experience can be represented in a knowledge graph and guide our recognition pipeline. Our proposed global reasoning is supervised by the global knowledge pool which is constructed by noun-wise correlation statistic on the dataset.

(b), these works learn local relationships in an implicit and uncontrollable way, and so their performance boost is limited. In other words, these methods will still fail to

- Weijiang Yu and Nong Xiao are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, Guangdong, China. Haofan Wang is with the Carnegie Mellon University. E-mail: weijiangyu8@gmail.com; haofanwang.ai@gmail.com; xiaon6@syzu.edu.cn.
- Guohao Li and Bernard Ghanem are with King Abdullah University of Science and Technology. E-mail: guohao.li@kaust.edu.sa; Bernard.Ghanem@kaust.edu.sa.
- Nong Xiao is corresponding author

Manuscript received X X, 20XX; revised X X, 20XX.

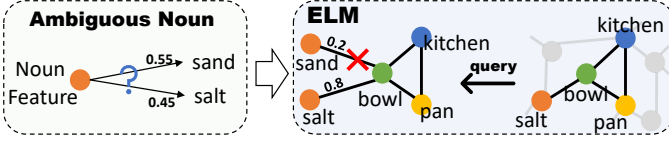


Fig. 2: Overview of our edge learning module (ELM) to solve the ambiguous noun by querying knowledge from the global knowledge pool. Although ambiguous noun feature tends to predict wrong result (e.g., sand), our ELM can correct it by using related relationship (salt, bowl and kitchen).

reason through a bad feature representation when heavy occlusions and class ambiguities exist in the image, which is very common in situation recognition. On the contrary, our work aims to develop a global reasoning framework, which can not only explicitly and adaptively learn multiple kinds of noun correlations from global knowledge pool, but also propagate noun-level information globally from all the categories to refine situation recognition.

In this paper, we study the application of global knowledge to the situation recognition task for the first time. As shown in Fig. 3 (a), we propose a novel Knowledge-aware Global Reasoning (KGR) framework consisting of a local encoder and a global encoder. We utilize the local encoder from GGNN [1] to generate noun features. The noun features and the noun relations are represented as nodes and edges respectively for the graph reasoning. Then, a global encoder is exploited to enhance the noun features via global reasoning. Our global encoder is composed of 1) an edge learning module which is an implicit knowledge graph to predict edges in noun features; 2) a node evolution module to enhance noun features via graph reasoning.

To get rid of the graph knowledge pool during test and enable our KGR to deal with ambiguous nouns, we design a differentiable edge learning module (ELM) to distill and fuse the hard graph representation into diffused knowledge. As shown in Fig. 2, if a noun gets softmax score of “sand 0.55” and “salt 0.45”, our ELM can learn to relate this ambiguous noun feature to other noun features (e.g., bowl, kitchen) in the image. Then, it can correct the noun feature from the wrong “sand” to the accurate “salt” by using related relationships queried from the global knowledge pool. Hence, our ELM can refine ambiguous noun features by producing more accurate edges. The hard graph knowledge cannot produce edges for refinement because the edges of ambiguous nouns can be absent, e.g., “sand” cannot be connected with “bowl”. So, we use our ELM to generate the soft edges by using external graph knowledge as an additional source of supervision during the training. To achieve end-to-end learning, we design an objective function for ELM to learn global correlations of nouns. Inspired by the human experience as shown in Fig. 1 (c), we fully consider the characteristics of our task and propose action-guided pairwise knowledge as shown in Fig. 3 (b) to be our global knowledge pool. This can not only show the concurrent relationships among nouns but also capture the essential connections between nouns and actions.

Contributions. (1) We propose a novel Knowledge-aware Global Reasoning (KGR) framework to improve the noun

representation by jointly reasoning the local-global correlations between nouns. (2) We present an edge learning module (ELM) to distill the global knowledge pool to deal with ambiguous nouns by generating soft edges. (3) We customize a global knowledge pool named action-guided pairwise knowledge to supervise the ELM. (4) Extensive experiments demonstrate that our proposed KGR achieves state-of-the-art performance in TABLE 1. In addition, we observe that our method can address the long-tailed problem to significantly improve the performance of low-frequency categories as shown in TABLE 5 and Fig. 6. The analysis of model complexity are in TABLE 6.

2 RELATED WORK

Situation Recognition. Visual reasoning tasks have attracted more and more attention in the vision community by targeting visual question answering [11], [12], visual grounding [13] and situation recognition [2]. Recently, attention was drawn to the situation recognition task under a more general and practical setting, where the roles can be any objects, subjects, and locations in the scene. Situation recognition involves classifying the salient action in one image and inferring the associated semantic nouns, which generalizes action recognition to other relevant roles playing in the activity. Yatskar et al. [2] constructed the imSitu dataset benchmark for situation recognition and provided a baseline method that employs a conditional random field to jointly predict the actions and semantic roles. Mallya et al. [14] improved the accuracy by using a specialized verb predictor and a Recurrent Neural Net (RNN) for noun prediction. Li et al. [1] use Gated Graph Neural Nets to implicitly capture joint dependencies between roles. Inspired by query-based visual reasoning from visual question answering, Cooray et al. [10] proposed a context-aware query network for inter-dependent semantic role prediction. The aforementioned methods overlook the statistical relationship of co-occurrences and action functionality between nouns and actions. Inspired by the human experience, we propose to use a global knowledge pool that exploits statistical relationships to explicitly enable our KGR to deal with ambiguous nouns.

Graph Networks. Many works put effort into incorporating graph knowledge to aid numerous vision tasks [15], [16], [17]. For instance, Marino et al. [15] built a knowledge graph based on WordNet [18] and the Visual Genome [19] dataset and learned the representation of this graph to enhance the feature representations for multi-label classification. Deng et al. [17] proposed semantic relationships including mutual exclusion, overlap, and subsumption as constraints in the loss function to train image classifiers. Graph networks are also widely used in visual reasoning. Yu et al. [20] proposed a heterogeneous graph learning method to seamlessly integrate the intra-graph and inter-graph reasoning in order to bridge multi-modal domains for visual commonsense reasoning. Li et al. [21] proposed to generate scene graphs via a subgraph-based model using bottom-up relationship inference of objects in images. In this work, we design an action-guided pairwise knowledge to represent the human experience for the task of situation recognition. This knowledge can build the relations between

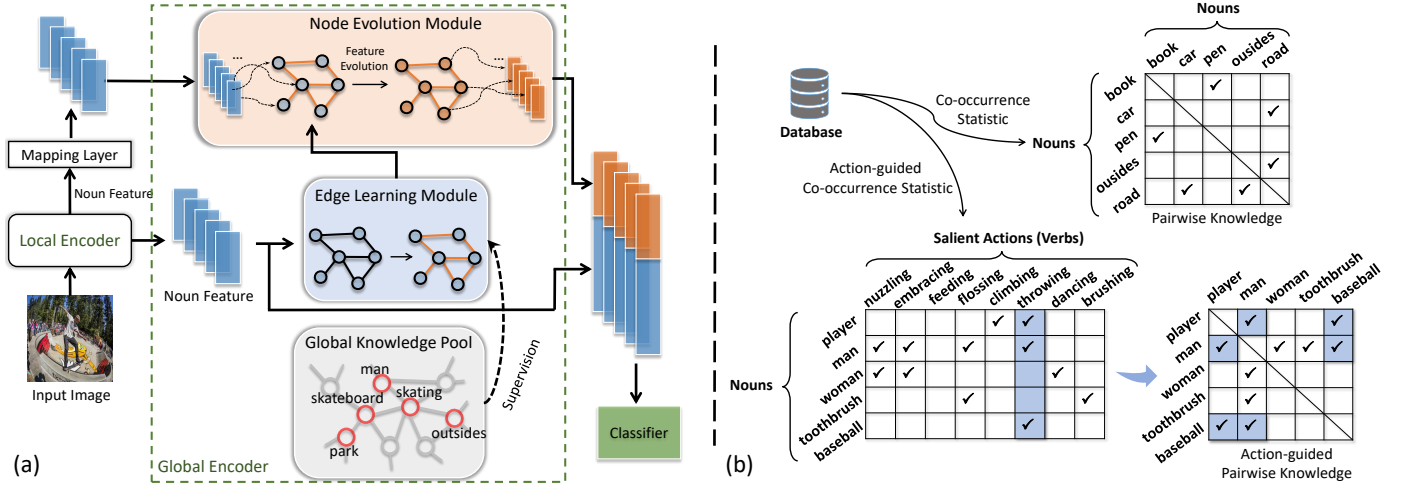


Fig. 3: (a) Overview of our knowledge-aware global reasoning (KGR) framework. (b) The processing progress of the normal pairwise knowledge and our action-guided pairwise knowledge. Our KGR framework consists of a local encoder and a global encoder. We adopt the GGNN [1] as our local encoder to extract the noun features. The global encoder aims to enhance the noun features via global reasoning, which is composed of an edge learning module (ELM) and a node evolution module. The ELM is optimized by an objective function to calculate the loss of the noun relations between the global knowledge pool and the generated soft edges from the ELM. To construct the global knowledge pool from the dataset, we design an action-guided pairwise knowledge, which can link every two nouns based on their common actions. For instance, the nouns called “player”, “man” and “baseball” can be connected together in our action-guided pairwise knowledge thanks to their common verb named “throwing”.

nouns and construct implicit connections between nouns and actions. The proposed knowledge graph can be used to regularize global relationships across different scenes.

3 KNOWLEDGE-AWARE GLOBAL REASONING

In this section, we describe the architecture of our proposed Knowledge-aware Global Reasoning (KGR). Our KGR framework consists of a local encoder to generate noun features by using local reasoning, and a global encoder that contains an edge learning module (ELM) and a node evolution module (NEM) to achieve global reasoning. Our ELM aims to generate informative edges for refining ambiguous nouns. Given the generated edges, the NEM achieves global reasoning via a graph convolutional operation to propagate the global correlation to features of each node. After the global reasoning, we obtain the enhanced noun features, which are concatenated with original noun features to feed into the predictor for the final role-noun prediction. As for the global knowledge pool, we analyze the basic pairwise knowledge and design action-guided pairwise knowledge as the global knowledge pool for our task.

Task Definition. Given an image I , the task of situation recognition aims to generate a structured description $S = (v, f_{(v,I)})$, which consists of a verb $v \in \mathcal{V}$ and its corresponding realized frame $f_{(v,I)} \in \mathcal{F}$. A realized frame $f_{(v,I)}$ contains a set of roles with their associated nouns: $f_{(v,I)} = \{(R_v, \mathbf{n}) : R_v \in \mathcal{F}, \mathbf{n} \in \mathcal{N} \cup \{\emptyset\}\}$ to describe the action v in image I . Each role is paired with a noun $\mathbf{n} \in \mathcal{N} \cup \{\emptyset\}$, where $\mathbf{n} = \emptyset$ represents that the noun of the role R_v does not exist in the situation. \mathcal{N} is the set of all nouns.

3.1 Overview of Framework

Our KGR framework is a local-global architecture composed of a local and global encoder. We use a similar local encoder as GGNN [1] to generate noun features. In this paper, we mainly introduce our global encoder that contains an edge learning module (ELM) and a node evolution module (NEM). In Fig. 3 (a), our framework first generates original noun features from the local encoder. Then, ELM is used to predict edge connections between each pair of noun features. This is supervised by a mapping of the ground truth from an external sub-graph \mathbf{K} queried from a global knowledge pool. We define $\mathbf{K} = \langle \mathcal{N}, \mathcal{E} \rangle$ as a class-to-class graph with N class nodes ($N=6$) and prior edge weights $e_{ij} \in \mathcal{E}$. \mathbf{K} can be formed from different types of pairwise knowledge. Based on the predicted edges, ELM executes global reasoning on the original noun features to produce enhanced noun features.

3.2 Global Knowledge Pool Construction

As mentioned above, we distill the global knowledge pool by our ELM. In this section, we present the construction of our global knowledge pool. First of all, we follow the same statistical rules in [1] to count the correlation of the most frequent 2,000 noun categories in the full imSitu dataset [2], including train, validation and test data. We only use the global knowledge pool as supervision to train our ELM in the training processing. In the test and validation, we don't use the global knowledge pool. We assume the model can't access the knowledge pool in the test and validation for fair comparison. Because other methods also don't access the statistical information in the test and validation. In our KGR, the only way to obtain the edges of noun categories in test and validation is to use the ELM's weights to predict.

To construct the global knowledge pool, we represent noun-to-noun pairwise relationships as a knowledge graph, which can be referred to as pairwise knowledge. To build the pairwise knowledge graph, we count the number of co-occurrences between noun categories and set them as the edge values in our graph, which is shown in Fig. 3 (b, top). However, such pairwise knowledge only builds the co-occurrence relationships between nouns and ignores the relationships between nouns and actions. In the task of situation recognition, we consider actions an important factor in noun prediction. We propose an action-guided pairwise knowledge for our global knowledge pool, which also takes the correlations between nouns and actions into account.

Pairwise Knowledge. We denote the pairwise knowledge as $\mathbf{K}^R \in \mathbb{R}^{2000 \times 2000}$, which has 2,000 pair relationships between nouns including place relationships (e.g., *outside*, *outdoor*), character relationships (e.g., *husband*, *wife*) and other co-occurrence relationships. As shown in Fig. 3 (b, top) we obtain \mathbf{K}^R by calculating frequency statistics from the occurrence among all noun class pairs. Specifically, we first create a $C \times C$ squared matrix \mathbf{M} with counts from the pairwise nouns, where C is the number of nouns ($C=2000$). Then, a column-row normalization [22] is performed to get \mathbf{K}^R :

$$\mathbf{K}_{ij}^R = \frac{\mathbf{M}_{ij}}{\sqrt{\mathbf{D}_{ii}\mathbf{D}_{jj}}}, \quad \mathbf{D}_{ii} = \sum_{j=1}^C \mathbf{M}_{ij}, \quad (1)$$

where \mathbf{M}_{ij} is the co-occurrence count between the i -th noun class and the j -th noun class and \mathbf{D} is the degree matrix, following from [22].

Although the pairwise knowledge can represent the global noun-to-noun correlations, it fails to capture the relationships between nouns and actions. We argue that incorporating relationships between nouns and actions can improve the performance on situation recognition. To achieve this goal, we design an action-guided pairwise knowledge.

Action-guided Pairwise Knowledge. We design the action-guided pairwise knowledge to have the same form as the original pairwise knowledge. The action-guided pairwise knowledge is defined as $\mathbf{K}^A \in \mathbb{R}^{2000 \times 2000}$. In Fig. 3 (b, bottom), there are two steps to construct \mathbf{K}^A . In the first step, we get frequency statistics for L types of actions and C noun categories, through counting. Then, a $C \times L$ frequency distribution table for each noun-action pair can be obtained, which has frequencies of all noun-action pairs. In the second step, the edge weights are defined by the pairwise Jensen-Shannon (JS) divergence between probability distributions P_{c_i} and P_{c_j} of noun classes c_i and c_j :

$$e_{c_i, c_j}^A = 1 - JS(P_{c_i} || P_{c_j}), \quad (2)$$

where e_{c_i, c_j}^A indicates the edge weight between two classes in the action-guided pairwise knowledge. The JS divergence is used to measure the similarity instead of Kullback-Leibler (KL) divergence, since we expect a symmetric undirected graph. The obtained action-guided pairwise knowledge is used as supervision for our KGR to learn the capability of global noun-to-noun reasoning for enhancing noun features. In this work, $C = 2000$ and $L = |\mathcal{V}| = 504$ are the numbers of noun categories and action categories, respectively.

Algorithm 1 PyTorch-style pseudocode for KGR.

```

1  # role number=6; noun number=3 (per role)
2  # img:[batch, 3, 224, 224]; verb (action):[batch];
   labels:[batch, 6, 3]
3  # load external knowledge, total 2000 noun classes
4
5  statistic_matrix = load.adj() # [2000, 2000]
6  for (img, verb, labels) in loader:
7      # extract sub-graph GT:[batch, 6*3, 6*3]
8      adj_gt = extract(statistic_matrix, labels)
9      # noun feature f:[batch, 6, feat_dimension]
10     f = net.local_encoder(img, verb)
11
12     # Edge Learning Module (ELM)
13     f1 = f.unsqueeze(2)
14     f2 = transpose(f1, 1, 2)
15     # adj_pred:[batch, 6, 6]
16     adj_pred = net.MLPk(transpose(abs(f1-f2), 1, 3))
17     adj_pred = adj_pred.squeeze(1)
18     # adj_p:[batch, 6*3, 6*3]
19     adj_p = adj_pred.repeat(1, 3, 3)
20     Loss1 = MSE(adj_p, adj_gt)
21
22     # Node Evolution Module (NEM)
23     # f_e:[batch, 6, hidden_dimension]
24     f_e = net.We(bmm(softmax(adj_pred), f))
25
26     # logits:[batch, 6, 2000]
27     logits = net.classifier(cat([f, f_e]))
28     # Total Loss
29     loss = CrossEntropyLoss(logits, labels) + Loss1
30     # SGD update
31     loss.backward()
32     update(net.params)

```

bmm: batch matrix multiplication.

3.3 Knowledge Reasoning

In this section, we introduce how to predict graph edges guided from sub-graph that is queried from action-guided pairwise knowledge pool. It contains an edge learning module and a node evolution module.

Edge Learning Module (ELM). Since we do not have access to the ground-truth nouns for querying the knowledge graph during inference time, we propose ELM to predict edges by distilling knowledge into the network during the training phrase. We denote the original noun features as $\mathbf{f} = \{f_i\}_{i=1}^N$, $f_i \in \mathbb{R}^D$, which are extracted from the backbone network. Here, N is the number of role nodes (each role should be full with a noun value) and D is the dimension of features. We define $\hat{e}_{ij} \in \hat{\mathcal{E}}$ as the predicted edge for each pair of noun features. Given an external knowledge graph, distinct edge connections \mathcal{E} can be accordingly extracted as sub-graphs to characterize the specific context information for each noun in the context of specific knowledge. Formally, the ELM learns an implicit knowledge graph supervised by the knowledge graph pool \mathbf{K} to generate edges \hat{e}_{ij} between nouns through a stacked Multi-layer Perceptron:

$$\hat{e}_{ij} = \text{MLP}_{\mathbf{K}}(\alpha(f_i, f_j)), \quad (3)$$

where \hat{e}_{ij} is the predicted edge and $\alpha(\cdot)$ is chosen to be the pairwise absolute difference between features of each noun pair (f_i, f_j) . Given different prior graphs \mathbf{K} , $\text{MLP}_{\mathbf{K}}$ would be parameterized with $\mathbf{W}_{\mathbf{K}}$ to generate different noun-to-noun edges for different images, leading to a personalized knowledge reasoning. The pseudo-code is in Algorithm 1.

Since we cannot use the ground-truth adjacency matrix (i.e. graph edges) during test, we design an objective function to assist the edge learning during the training. We learn $\text{MLP}_{\mathbf{K}}$ by directly enforcing the predicted \hat{e}_{ij} to be consistent with the edge weight e_{ij} of a prior graph \mathbf{K} . During

TABLE 1: **Quantitative comparison in three metrics on the development and test sets against the state-of-the-art dataset.** Each model was run on the test set only once. † denotes the results of our implementation using official code strictly following the settings from the papers. We highlight the best results in **bold**. Note that the results of different methods on *verb* are the same since the same verb predictor are used.

		top-1 predicted verb			top-5 predicted verbs			ground truth verbs		mean
		verb	value	value-all	verb	value	value-all	value	value-all	
dev	TDA† [10]	37.12	26.17	12.37	65.02	44.78	20.06	66.30	27.05	37.36
	CAQ† [10]	37.12	26.58	12.98	65.02	45.45	20.87	67.22	28.12	37.92
	GGNN† [1]	37.12	28.72	17.03	65.02	48.67	26.89	70.60	34.86	41.11
	KGR	37.12	29.40	17.88	65.02	50.03	28.69	73.52	38.38	42.51
test	TDA† [10]	37.51	26.39	12.70	65.27	45.06	20.42	66.21	27.09	37.58
	CAQ† [10]	37.51	26.85	13.35	65.27	45.77	21.38	67.16	28.34	38.16
	GGNN† [1]	37.51	28.95	16.99	65.27	48.76	26.56	70.21	34.35	41.07
	KGR	37.51	29.57	17.96	65.27	50.11	28.48	73.12	37.85	42.48

training and since we know the ground-truth categories of each noun, we can extract a sub-graph from the whole pool as our supervision. The edge \hat{e}_{ij} of two noun nodes is learned to regress the edge weights e_{ij} of the ground truth categories of noun nodes in \mathbf{K} . The $\text{MLP}_{\mathbf{K}}$ learns to distill the noun-wise knowledge correlations during training and serves as an implicit knowledge graph to produce edges for nouns during the testing phase. The training loss of learned edge weights $\{\hat{e}_{ij}\}$ for all N noun values are defined as:

$$\mathcal{L}(\mathbf{f}, \mathbf{W}_{\mathbf{K}}, \mathbf{K}) = \sum_{i=1}^N \sum_{j=1}^N \frac{1}{2} (\hat{e}_{ij} - e_{ij})^2, \quad (4)$$

where $\mathcal{L}(\mathbf{f}, \mathbf{W}_{\mathbf{K}}, \mathbf{K})$ is the objective function with supervision from the external knowledge \mathbf{K} .

Node Evolution Module (NEM). Given the predicted edges from the edge learning module, our node evolution module performs global feature update via a graph convolutional operation. Inspired by [22], we first map the initial noun features $\mathbf{f}^{N \times D}$ into a embedding space via a mapping layer (i.e.MLP). Then, a row normalization over learned edges $\hat{\mathcal{E}}^{N \times N} = \{\hat{e}_{ij}\}$ is performed. The NEM propagates the learned relational knowledge of connected nouns to enhance noun features through a generated adjacency matrix with weighted edges, which is implemented as follows:

$$\mathbf{f}_e = \hat{\mathcal{E}} \mathbf{f} \mathbf{W}_e, \quad (5)$$

where $\mathbf{f}^{N \times D}$ denotes the original noun features, $\mathbf{W}_e \in \mathbb{R}^{D \times E}$ is a transformation weight matrix and $\mathbf{f}_e \in \mathbb{R}^{N \times E}$ denotes the enhanced noun features. Those rare noun classes with few samples that contain occlusions and class ambiguities can benefit from the global knowledge across diverse scenes. Finally, we concatenate the enhanced features \mathbf{f}_e with the original features \mathbf{f} together to predict the nouns for each role via a classifier.

We present the objective function \mathcal{L} of our KGR framework. We denote the cross-entropy loss for action and noun prediction for every annotation as \mathcal{L}_{CE} . The objective function of global knowledge supervision is shown in Eq. 4. The joint objective function is $\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}(\mathbf{f}, \mathbf{W}_{\mathbf{K}}, \mathbf{K})$.

4 EXPERIMENTS

4.1 Dataset and Evaluation

We evaluate the proposed KGR and existing state-of-the-art competitors on the imSitu dataset [2]. We follow the experiment setup from [10]. The imSitu contains 75,702 training images, 25,200 validation images, and 25,200 test images. Each image corresponds to a verb and up to 6 role-noun tuples, where each role contains 3 candidate nouns. There is a total of 504 verbs, 190 roles, and 2,000 nouns except the \emptyset for the dataset (top 2000 nouns from the whole imSitu dataset). It is a challenging and the most widely used benchmark for situation recognition. The evaluation criteria are consistent with [2] and consist of three metrics: 1) *verb*: the performance of action prediction model; 2) *value*: the role-noun tuples prediction performance that is considered to be correct if it matches any of the three ground truth annotations; 3) *value-all*: the prediction is correct when all verb-role-value pairs match at least one ground truth annotation.

4.2 Implementation Details

We implement our model and re-implement several state-of-the-art methods [1], [10] using PyTorch [23] framework under the same experimental setting. We re-implement the methods from [1], [10] using the official code or pre-trained models and strictly follow the setting mentioned in the original papers. We adopt a pre-trained ResNet-50 [3] without the last fully-connected and softmax layers for extracting image features and build our local encoder based on the GGNN [1]. The whole model is trained with Adamax optimizer [24] with mini batches of 64 samples. The initial learning rate is 0.0005 for ResNet-50 and 0.001 for other modules. The learning rate of each parameter group decays at a rate of 0.9 at every epoch.

4.3 Results and Comparisons

We report the results on the imSitu dataset [2] in TABLE 1. Our KGR outperforms all other methods on both sets. The *predicted verb* means that we use the predicted verb to assist the role-noun prediction. The *ground truth verb* denotes that the ground truth verb is used to help the role-noun

prediction. To fairly compare the performance on the task of role-noun prediction, we use the same verb predictor (e.g., ResNet-50 [3]) for all methods to obtain consistent verb prediction results. Our KGR achieves an overall mean accuracy for each sample of 42.51% compared to 41.11% by GGNN [1] on the development set and 42.48% compared to 41.07% on the test set. Our method consistently outperforms existing methods on all three metrics on both the development and test sets thanks to the ability to reason about the local-global correlations between nouns. Moreover, our model achieves significantly higher performance on both top-1 and top-5 accuracy metrics compared with other state-of-the-art methods such as TDA [10], CAQ [10] and GGNN [1]. Compared with GGNN, the performance gains of our method benefit from our knowledge-aware global reasoning guided by rich external knowledge supervision.

Visualization of the learned noun features. To better understand the underlying feature representations that our KGR framework actually learns, we record 1,000 outputs from the node evolution module and the corresponding real labels. Then, we take the average according to the labels and use the t-SNE [25] method to visualize them in Fig. 4 (b). Similarly, we cluster the noun features from our baseline model [1] and visualize the results in Fig. 4 (a). If the features of some classes are close to each other, the edges between those close classes are more likely to be activated. We also show a sub-graph of our global knowledge pool in Fig. 4 (c) to demonstrate that our KGR can successfully learn the knowledge from the global knowledge pool. Here, we pick examples with two different types of semantic relationship to compare the semantic distance of learned noun feature between baseline and our method. In Fig. 4 (b), the “football” noun and the “football player” noun are close to each other due to the high co-occurrence frequency. And the “playing” action can successfully bridge the relation between the “football player” and the “football field”. In another semantic relationship, the “dentition” and “toothbrush” are closely connect due to the “brushing” action as the bridge. However, the noun features from the baseline (i.e., Fig. 4 (a)) cannot establish such a global structured connection due to the lack of supervision from the global knowledge pool.

4.4 Ablation Studies

We conduct ablation studies to demonstrate the effectiveness of our KGR framework and the action-guided pairwise knowledge. In TABLE 2, our KGR achieves the best performance and exceeds the baseline model (GGNN) by 2.92% (73.52 vs 70.60) on the development set and 2.91% (73.12 vs 70.21) on the test set in the *value* metric. Without using external knowledge (w/o $\mathcal{L}(\mathbf{f}, \mathbf{W}_K, \mathbf{K})$), the performance of KGR drops by 0.36% in the the *value* metric on the development set, which suggests the effectiveness of the action-guided pairwise knowledge for improving the KGR capability. Using the ELM instead of a dot product operation to generate graph edges reaches higher accuracy (72.66% vs 73.12% on the test set). These results show the advantages of the proposed ELM. After replacing the learned edge weights with the ground-truth edge weights (KGR w/ GT), we get 73.24% and 38.08% w.r.t value metric and value-all

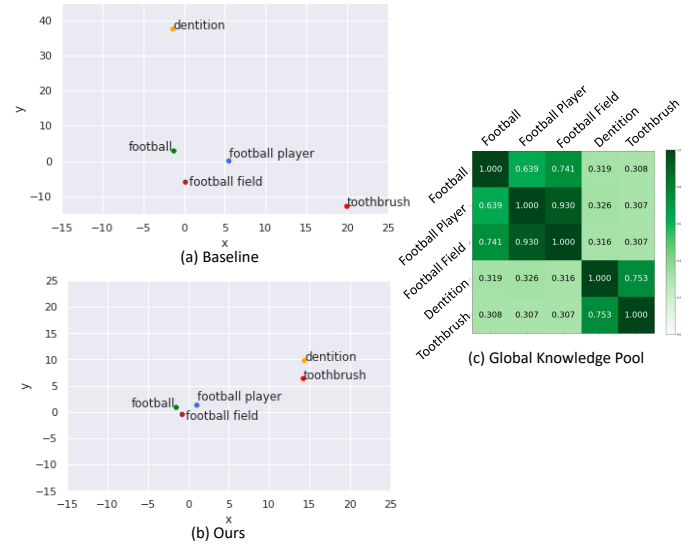


Fig. 4: The 2D visualization of the noun-level feature by the t-SNE method [25]. The categories shared with the similar correlation are closed to each other. This verifies that our KGR can learn the corresponding knowledge.

TABLE 2: Ablation study on the development and test set. The KGR w/o $\mathcal{L}(\mathbf{f}, \mathbf{W}_K, \mathbf{K})$ means training the KGR without global knowledge supervision. The ELM means the edge learning module. We replace the learned edges with the ground-truth edges, which is denoted as KGR w/ GT. KGR[†] means the global knowledge pool of KGR only built from training data. KGR* indicates replacing ELM with sub-graph that is queried from knowledge graph pool built by pure train data.

		value	value-all
dev	KGR w/ GT	73.57	38.45
	KGR	73.52	38.38
	KGR [†]	73.27	38.16
	KGR*	72.57	37.25
	KGR w/o $\mathcal{L}(\mathbf{f}, \mathbf{W}_K, \mathbf{K})$	73.16	37.96
	KGR w/o ELM	72.99	37.60
	Baseline (GGNN)	70.60	34.86
test	KGR w/ GT	73.24	38.08
	KGR	73.12	37.85
	KGR [†]	73.00	37.67
	KGR*	72.26	36.81
	KGR w/o $\mathcal{L}(\mathbf{f}, \mathbf{W}_K, \mathbf{K})$	72.82	37.46
	KGR w/o ELM	72.66	37.21
	Baseline (GGNN)	70.21	34.35

metrics on test set, which outperforms our KGR (73.12% and 37.85%). It shows the quality of learned edges is similar to the ground truth of knowledge, which demonstrates the reasonable of our design.

To construct our global knowledge pool, We follow the same statistical rules in [1] to count the correlation of the most frequent 2,000 noun categories in the full imSitu dataset [2] including train, val and test data. This knowledge pool is only used as supervised label during the training.

We do not use the knowledge edge as input or supervision during the validation and test. To get rid of the effect of correlation from test and val data, we rebuild our knowledge pool by only using train data. Then we train our model on the reconstructed knowledge pool, and test it on val and test set, which is marked as KGR^\dagger in TABLE 2. Without using the data from test and val data to build the global knowledge pool, our model meets slight performance decrease (73.27% on dev set and 73.00% on test set). While the KGR^\dagger still performs better than the baseline (GGNN) on dev set and test set over all metrics. Hence, the key gain of our model comes from the human experience simulation using statistical information about noun categories rather than using test and dev data to build the graph knowledge pool.

We have conducted experiments to apply a multi-head attention module ($input_dim = 128, embed_dim = 576, num_heads = 8$) to adj_p ($6*3, 6*3$) to explore the effect that how a multi-head attention module to capture similarities across different noun class. It achieves 73.00 value and 37.78 value-all on dev set and 72.58 value and 37.06 value-all on test set. While our KGR performs 73.52 value and 38.38 value on dev set, gets 73.12 value and 37.85 value-all on test set. It demonstrates the similarities across different nouns from each role cannot improve the performance. In contrast, it brings unexpected noise to disturb the prediction. Besides, we have implemented experiments to discuss the influence by only using f_e . It achieves 72.44 value and 36.97 value-all on dev set, and performs 71.99 value and 36.33 value on test set. It still outperforms the baseline. After concatenating f with f_e , it brings improvement of 0.83% and 1.19% w.r.t. value and value-all on dev set. As we can see, only using f_e can bring improvement from baseline. While the better way is to combine the basic feature f with refined feature f_e to get better performance. Such empirical combination operation usually can be seen in fundamental vision tasks, such as image classification, semantic segmentation, etc.

To further explore the effect of pre-computed knowledge pool constructed by pure train data, we have conducted additional experiments to replace the ELM with the sub-graph that is queried from pre-computed graph knowledge pool constructed by train data during testing, which is indicated as KGR^* in TABLE 2. After replacing the ELM with the pre-computed graph weights (KGR^*), we get 72.26% and 36.81% w.r.t value metric and value-all metrics on test set, which underperforms our KGR^\dagger (73.00% and 37.67%). It can demonstrate our ELM can achieve better generalization than directly querying pre-computed graph from train set.

To further demonstrate the effectiveness and generalization of our knowledge graph, we have done more ablation studies as shown in TABLE 3 to analyze the superior performance of our action-aware knowledge graph. We execute the action-aware knowledge graph on the several state-of-the-art methods to further verify our knowledge graph routing mechanism. We analyze the effect of action-aware knowledge graph on TDA [10] framework. The action-aware knowledge graph can significantly promote the TDA by 4.73% accuracy (value) and 7.20% accuracy (value-all) on development set, 4.48% (value) and 6.97% (value-all) on test set, respectively. Moreover, as can be seen in TABLE 3, the action-aware graph is apparently to boost the

development accuracy by around 3.88% (value) and 6.52% (value-all) from the CAQ [10]. On the test set, our action-aware graph can also outperform the CAQ by 3.45% value accuracy and 5.62% value-all accuracy. The proposed graphs collaborated with GGNN [1] are evaluated on development set and test set, which gets great scores on overall metrics as validating the availability of the graphs. In conclusion, our proposed knowledge graph routing mechanism can be inserted seamlessly into any situation recognizing pipeline to remarkably improve their performance by incorporating external commonsense knowledge into the neural networks.

TABLE 3: Ablation study on development and test set. The upper right number means the improvement benefited from the graph. K^A means the action-guided knowledge graph.

		value	value-all
dev	TDA [10]	66.30	27.05
	TDA + K^A	71.03 ^{+4.73}	34.25 ^{+7.20}
	CAQ [10]	67.22	28.12
	CAQ + K^A	71.10 ^{+3.88}	34.64 ^{+6.52}
	GGNN [1]	70.60	34.86
	GGNN + K^A	73.52 ^{+2.92}	38.38 ^{+4.52}
test	TDA [10]	66.21	27.09
	TDA + K^A	70.69 ^{+4.48}	34.06 ^{+6.97}
	CAQ [10]	67.16	28.34
	CAQ + K^A	70.61 ^{+3.45}	33.96 ^{+5.62}
	GGNN [1]	70.21	34.35
	GGNN + K^A	73.12 ^{+2.91}	37.85 ^{+3.5}

4.5 Knowledge Pool Transfer Learning

In this section, we want to test our KGR on another dataset using zero-shot setting to show transfer ability of our knowledge pool. However, there is no other same dataset like imSitu to support situation recognition. Hence, we decide to construct a dataset. Firstly, we find an action recognition dataset named Stanford Action40 [26]. Then we randomly select images from its test set that have same action categories with imSitu dataset. There are seven action categories existing in two datasets (Stanford Action40 and imSitu), including drinking, gardening, cooking, applauding, phoning, running and fishing. We randomly collect 15 images for each category. Next, we label such 105 images with noun categories by using the imSitu tool*. We call our constructed dataset as Action40-105. Then we test our KGR and our baseline (GGNN) on the Action40-105, which results are shown in TABLE 4. As we can see, our KGR outperforms GGNN by 6.51% (57.94% vs 51.43%) on Action40-105. It verifies our KGR indeed learns some general high-level knowledge. It also demonstrates the effectiveness and superior transfer ability of our knowledge pool.

Although our KGR performs better than GGNN on another dataset under zero-shot setting, we wonder the gap between imSitu and other dataset. For fair comparison, we also randomly choose 105 images for seven categories (15 images per category) from imSitu test set, which constructs a subset indicated as imSitu-105. Then we test KGR

*<http://imsitu.org/browse/>

TABLE 4: **Generalization of our knowledge pool.** We test our model on another dataset named Action40-105 while keeping the current knowledge pool built from the imSitu dataset, which aims to explore transfer ability of our knowledge pool.

Datasets	Methods	value	value-all
imSitu-105	GGNN	72.70	33.33
	KGR	74.37	36.19
Action40-105	GGNN	51.43	7.62
	KGR	57.94	12.38



	Role	Noun (GGNN)	Noun (KGR)
	Agent	People	Woman
	Tool	Null	Spoon
	Container	Null	Pot
	Food	Null	Null
	Heatsource	Null	Null
	Place	Outdoors	Outdoors
Cooking			
	Role	Noun (GGNN)	Noun (KGR)
	Agent	Woman	Person
	Tool	Null	Shovel
	Place	Outdoors	Garden
Gardening			

Fig. 5: **Qualitative results between baseline and our KGR on Action40-105.**

and GGNN on imSitu-105, and compare their results with Action40-105’s test results. We find there is a gap (16.43%, 74.37%→57.94%) between two datasets. One of the possible reasons is that our KGR trained on imSitu rather than Action40. Compared with our KGR, the GGNN has bigger gap with 21.27% (72.70%→51.43%). It means our knowledge pool can help network to effectively narrow the gap between two different dataset even though there are some data bias. Without our knowledge pool, the GGNN can not support data bias well and exists bigger gap.

We also some some visualization comparison between GGNN and our KGR on another dataset (Action40-105) in Fig. 5. In the first example, our KGR can well recognize “spoon” and “pot” in the “Cooking” activity. While GGNN can not recognize any tools related with “Cooking”. In the second example, based on the “Gardening” action, our KGR infers the place with “Garden”, and recognizes the “Shovel” that usually connects with Garden in our knowledge pool. The GGNN predicts ambiguity place (i.e., outdoors) rather than a precise place (e.g., garden). Hence, our KGR can transfer to another dataset and remains learned knowledge.

4.6 Analysis of the Long-tailed Problem

To understand how KGR alleviates the long-tailed distribution problem, we apply KGR on different frequent noun classes and compare the accuracy between the high-frequency nouns and the low-frequency nouns. As shown

in TABLE 5, our method achieves the best mean accuracy across noun classes. Specifically, we estimate the generalization of the long-tailed data by evaluating the model on the data with different frequency classes rather than the samples. We test our KGR on samples of the top-10 most frequent categories of nouns. Our method achieves 58.51% and 58.13% in top-1 accuracy on the development and test sets, respectively. This is slightly lower than TDA (61.85%) and CAQ (61.12%). However, on the samples of the less frequent categories (denoted as “rest”), TDA and CAQ both suffer from a substantial performance drop (decreasing to 31.26% and 33.12%) due to the long-tailed distribution problem. In contrast, our method can effectively ease the influence of the uneven data distribution and achieves 43.37% and 42.95% on the development and test sets, respectively.

We find that current sample-based metrics are dominated by the performance of the most frequent noun classes. As shown in Fig. 6 (a), methods such as TDA and CAQ perform well on the top-10 most frequent noun categories, but perform poorly on the other less frequent classes. As such, we compare our KGR with the baseline model on the top-20 most frequent noun classes (e.g. Fig. 6 (c)) and the top-20 low-frequency noun classes randomly extracted from other less frequent categories (e.g. Fig. 6 (b)). Our KGR performs comparably on the high-frequency classes in Fig. 6 (c), and achieves better results for those less frequent noun classes like “lizard” and “rake” compared to other methods.

Furthermore, we show some qualitative results in Fig. 7 to analyze how the external knowledge helps the long-tailed problem. At the first example, the GGNN can not predict the right place, while our KGR recognize the precise “Shower” place thanks to the model weights that learned from global knowledge pool can connect “shower” with “shampoo”. The “Shower” belongs to low-frequency noun classes in imSitu dataset. Besides, at the second example, our KGR can predict the “Gas Tank” rather than “Car”. Because “gas tank” has strong connection with “gasonline”, “storage”, “gas pump” and “gasonline station” in global knowledge pool. This can further demonstrate our KGR can store the knowledge into model’s weights and transfer to test set to help the long-tailed problem.

We provide more metrics in TABLE 6 to show that our method significantly outperforms others on rare categories in both precision and recall. We also compare the parameter efficiency and computational efficiency of the models. Our model has slightly more parameters and FLOPs than TDA and GGNN, and less parameters and FLOPs than CAQ, while achieving significantly better performance on rare categories. These results support the claim that our KGR alleviates the long-tailed distribution problem of noun prediction and improves the prediction of rare categories.

4.7 Statistic Distribution for Long-tailed Problem.

To further demonstrate the effectiveness of knowledge, we analyze the difference between the distributions of the top 100 frequent noun classes and top 100 frequent noun-level relationships in the imSitu [2] dataset. As exhibited in Fig. 8 (a), we show the statistic results of high-frequency noun classes, which is a long-tailed distribution where rare categories have few samples. The long-tailed problem leads

TABLE 5: **Quantitative comparison w.r.t top-1 accuracy metric of the noun classes with different frequencies.** The *top-10* means the samples of top 10 most frequent categories of nouns. The rest indicates the samples of the rest less frequent categories. Similar meanings for *top-20* and *top-50*. The *average classes* means the top-1 accuracy for each noun class.

		top-10 classes (top-10/rest)	top-20 classes (top-20/rest)	top-50 classes (top-50/rest)	average classes (mean)
dev	TDA [†] [10]	61.69/31.27	58.78/28.22	58.62/22.55	44.70
	CAQ [†] [10]	61.07/33.07	59.34/30.23	58.60/24.48	45.42
	GGNN [†] [1]	57.48/40.68	56.13/39.31	55.31/36.63	48.09
	KGR	58.48/43.37	57.26/42.15	56.66/39.52	50.04
test	TDA [†] [10]	61.85/31.26	59.34/28.79	58.44/22.82	44.66
	CAQ [†] [10]	61.12/33.12	58.82/30.86	58.44/24.68	45.38
	GGNN [†] [1]	57.41/40.42	55.62/39.48	55.12/36.36	47.86
	KGR	58.51/42.95	56.85/42.11	56.53/39.04	49.76

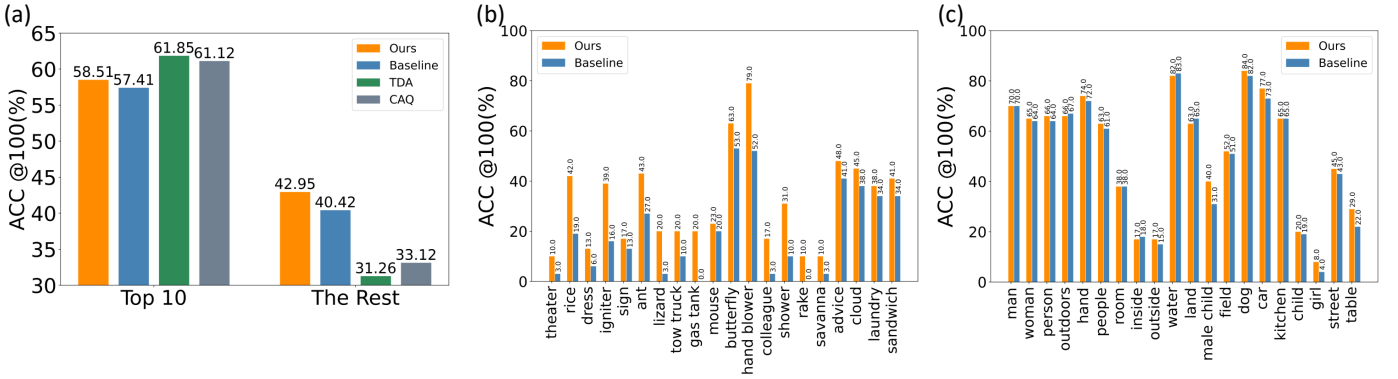


Fig. 6: (a) The results of comparison between our KGR and the other approaches on the top 10 most frequent noun classes and the rest less frequent categories. (b) Comparison between our model and our baseline on the data of the extracted low-frequency noun classes. (c) Comparison between our KGR and the baseline on the data of high-frequency noun classes. Please zoom in the colored PDF version of this paper for more details.

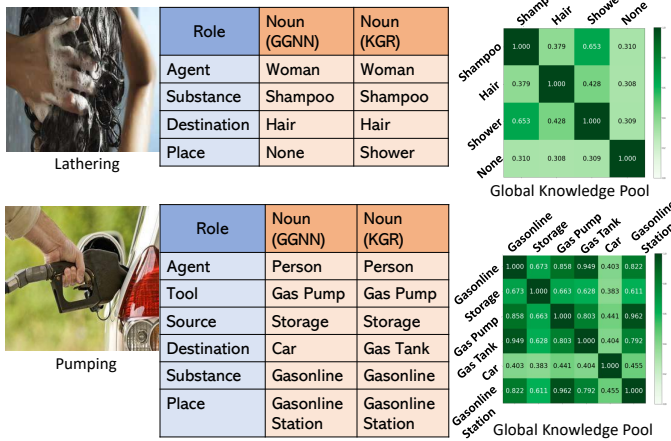


Fig. 7: Visualization comparison in long-tailed categories between baseline (GGNN) and our KGR.

the model trained by a data-driven way with pure class-level supervision to wrongly predict rare classes as high-frequency classes. To mitigate the influence from the long-tailed problem, we introduce the external knowledge supervision, which can be extracted by statistical relationship. As shown in Fig. 8 (b), we analyze the distribution of noun relationships in the dataset. Different from the distribution

TABLE 6: **Analysis of the average precision (P), recall (R), parameters and flops.** The *rare-10* means the top 10 rarest categories of nouns. Similar meaning for *rare-20*, *rare-50* and *rare-100*. The *All* means the results over all nouns. The GFLOPs is computed by using the fvcare tool.

	TDA (P/R)	CAQ (P/R)	GGNN (P/R)	KGR (P/R)
<i>rare-10</i>	0.00/0.00	0.00/0.00	0.00/0.00	0.07/0.10
<i>rare-20</i>	0.00/0.00	0.00/0.00	0.00/0.00	0.04/0.08
<i>rare-50</i>	0.00/0.00	0.00/0.00	0.02/0.02	0.04/0.05
<i>rare-100</i>	0.00/0.00	0.00/0.00	0.02/0.02	0.05/0.06
<i>All</i>	0.06/0.07	0.07/0.08	0.21/0.19	0.26/0.22
Params (M)	43.74	93.77	35.71	51.08
GFLOPs	4.21	6.69	4.26	4.52

of top 100 frequent noun classes as shown in Fig. 8 (a), the different frequency between each pair-wise relationship is very small (roughly 0.17). To some extent, the distribution of pair-wise relationship is relatively balanced. Therefore, the rare classes can be benefited from more frequent ones by using such balanced distribution. For instance, the “football player” and “student” are rare categories where have few samples in the dataset. However, the relationships such as “football player” and “football”, “student” and “teacher”

are high-frequency relations in imSitu dataset. Hence, the rare classes can be benefited from more frequent ones by bridging structured pair-wise relation knowledge between classes. We subtly utilize such intrinsic characteristic of the relationships as external knowledge to solve the long-tailed distribution problem.

4.8 Feature Analysis for Different Knowledge

To better understand the underlying feature representations that our knowledge graph routed network actually learns for graph reasoning, we record the output from the node evolution module and its corresponding real labels from each noun category of 1000 imSitu [2] images. Then we take average according to the labels and use the t-SNE [25] clustering method to visualize them as shown in Fig. 9. Note that if the features of some classes are closed to each other, the edges between those closed classes are more likely to be activated. As shown in Fig. 9 (a), the “woman” and “sewing machine” are closed to each other due to the similar action attribute “sewing”. The man named “factory” can be connected with the “wrench” in the industry by using the commonsense knowledge, although the “factory” and “wrench” have different attributes. Similarly, the noun-level class called “woman” has action-aware relationship in the kitchen such as “using knife and glass” for cooking. In Fig. 9 (b), the “sheep” and “cow pen” occur frequently in many scenes so that their features are closed to each other. Besides, the “construction worker” often goes to the “outdoors” for cutting the “wood”. The “bulldozer” and “crane” often show up together for building a house. Benefiting from the explicit knowledge supervision, our model can be benefited from incorporating external graph knowledge to better discriminate the features of different noun classes from a global perspective.

4.9 Interpretability of the Learned Graph

In Fig. 10, we visualize the edge weights generated by the baseline (GGNN [1]) and our ELM trained with action-guided pairwise knowledge. The edge values generated by the GGNN tend to be very similar, which fail to represent the global relations between nouns.

Since the GGNN predicts graph edges based on local image features without any global reasoning, it fails to build the relationship among the “bride”, “groom”, “wedding” and “unveiling” (Fig. 10 (b)). Hence, the baseline model can not precisely predicts the “man” as “groom” due to the global relationship absence. Furthermore, our methods can correctly predict the example in Fig. 10 (a) by distinguishing the “Teacher”. Since the “teacher” and “student” is highly co-occurrent in normal situations, our proposed knowledge-aware global reasoning method benefits from such global co-occurrence to bridge the “student” and “teacher” in a lecturing situation. However, the purely data-driven baseline wrongly predict the “teacher” as “man” for lacking of global knowledge.

In Fig. 10 (d), we can see that our method can successfully connect the noun-wise correlation among the “baseball”, “ball filed” and “ballplayer” to better predict the noun value (i.e. Ballplayer) of the agent role, which can further help to refine the ambiguous noun prediction. By

taking a look at the instance in Fig. 10 (c), the baseline only predict the apparent noun like “man” and “kitchen” in the “cooking” situation. Different from the baseline, thanks to the global relationship, our method can predict the “knife” and “chopping board”, “chief” and “kitchen” in the “cooking” situation. In a conclusion, the precise prediction of our action-guided pairwise graph thanks to the proposed edge learning module (ELM) with a customized pairwise knowledge pool. Compared with the GGNN graph, our ELM can generate a more reasonable action-guided pairwise graph. This further demonstrates that our ELM can refine ambiguous noun features by learning more accurate relationships.

5 CONCLUSION

We propose a knowledge-aware global reasoning (KGR) network to distill our explicit-globally knowledge for improving the situation recognition task. The distillation way is executed by our edge learning module cooperated with our node evolution module. Both of them are optimized by two objective functions in an end-to-end training mechanism. Our KGR achieves state-of-the-art performance as well as addresses the long-tailed problem of noun classification.

6 ACKNOWLEDGEMENTS

This work was supported in part by National Natural Science Foundation of China under Grant No.U1811461, in part by the Major Program of Guangdong Basic and Applied Research No.2019B030302002, in part by Natural Science Foundation of Guangdong Province, China under Grant No.2018B030312002.

REFERENCES

- [1] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, “Situation recognition with graph neural networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4173–4182.
- [2] M. Yatskar, L. Zettlemoyer, and A. Farhadi, “Situation recognition: Visual semantic role labeling for image understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5534–5542.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs,” in *ICLR*, 2015.
- [5] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, “From recognition to cognition: Visual commonsense reasoning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6720–6731.
- [6] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz, “Scene perception: Detecting and judging objects undergoing relational violations,” *Cognitive psychology*, vol. 14, no. 2, pp. 143–177, 1982.
- [7] M. Yatskar, V. Ordonez, L. Zettlemoyer, and A. Farhadi, “Commonly uncommon: Semantic sparsity in situation recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7196–7205.
- [8] A. Mallya and S. Lazebnik, “Recurrent models for situation recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 455–463.
- [9] S. Pratt, M. Yatskar, L. Weihs, A. Farhadi, and A. Kembhavi, “Grounded situation recognition,” *arXiv preprint arXiv:2003.12058*, 2020.

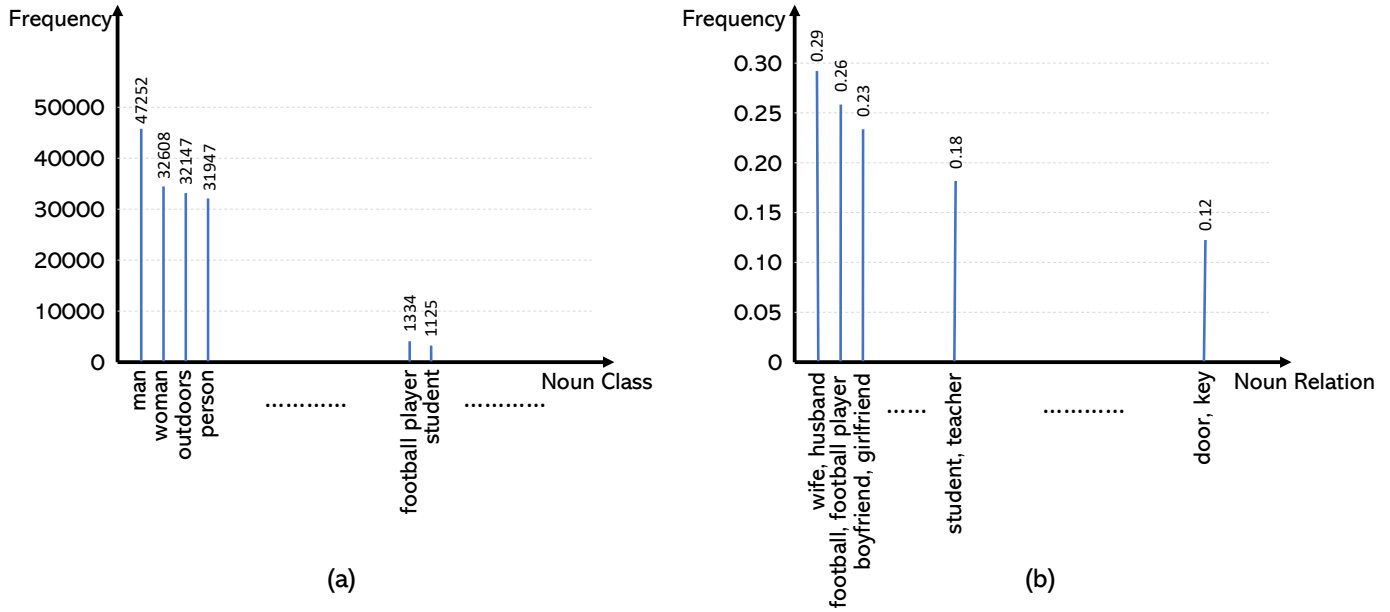


Fig. 8: (a) The distribution of the top 100 frequent noun classes on the imSitu dataset [2]. This distribution suffers from severe long-tailed problem due to a huge different number between the highest frequent classes and lowest frequent classes such as above 47,000 quantities. The long-tailed problem leads the model trained by a data-driven way with pure class-level supervision to wrongly predict rare classes as high-frequency classes. (b) The distribution of the top 100 frequent noun relationships of action-aware graph on the imSitu dataset [2]. Different from (a), the different frequency between each pair-wise relationship is very small (roughly 0.17). Hence, the rare classes can be benefited from more frequent ones by bridging structured pair-wise relation knowledge between classes.

- [10] T. Cooray, N.-M. Cheung, and W. Lu, "Attention-based context aware reasoning for situation recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4736–4745.
- [11] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.
- [12] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.
- [13] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *ICCV*, 2015.
- [14] A. Mallya and S. Lazebnik, "Recurrent models for situation recognition," 2017.
- [15] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," *arXiv preprint arXiv:1612.04844*, 2016.
- [16] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1576–1585.
- [17] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *European conference on computer vision*. Springer, 2014, pp. 48–64.
- [18] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [19] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: <https://arxiv.org/abs/1602.07332>
- [20] W. Yu, J. Zhou, W. Yu, X. Liang, and N. Xiao, "Heterogeneous graph learning for visual commonsense reasoning," in *Advances in Neural Information Processing Systems*, 2019, pp. 2769–2779.
- [21] Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang, "Factorizable net: An efficient subgraph-based framework for scene graph generation," *arXiv preprint arXiv:1806.11538*, 2018.
- [22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.
- [23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.
- [24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [25] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [26] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," *2011 International Conference on Computer Vision*, pp. 1331–1338, 2011.

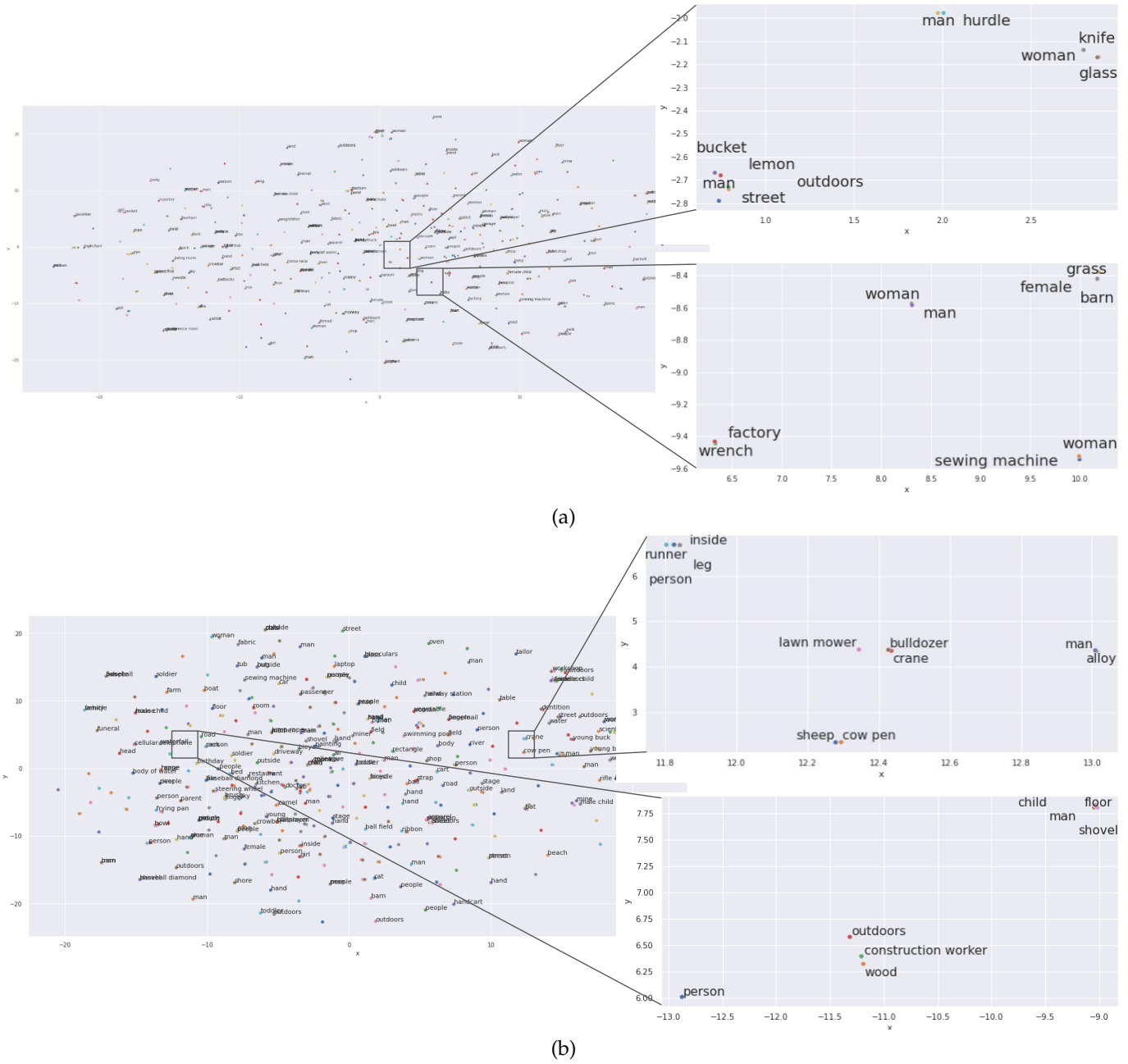
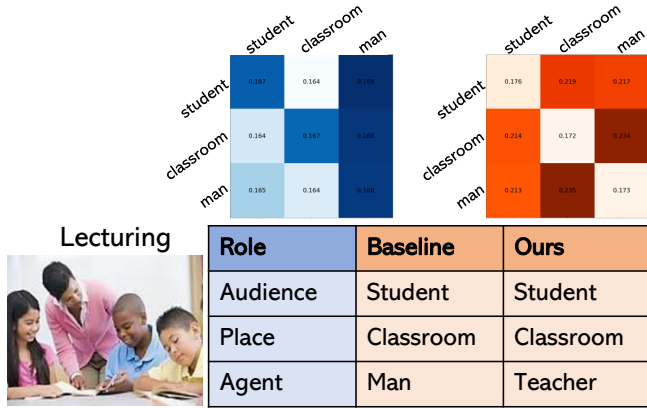


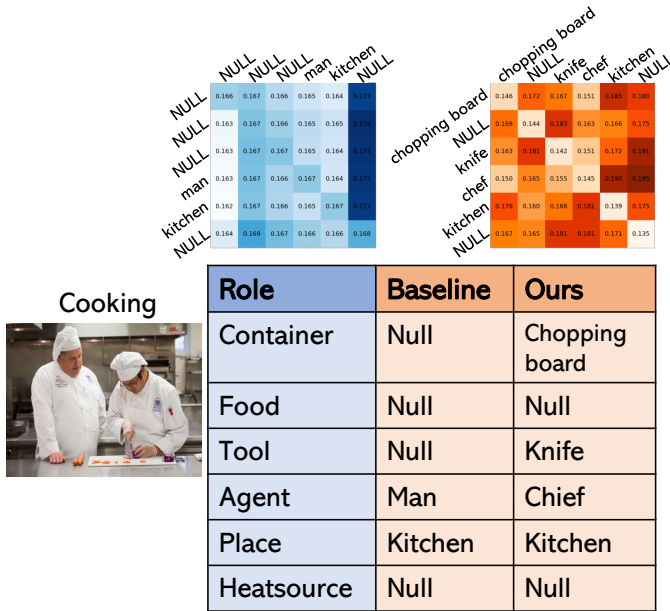
Fig. 9: The 2D visualization of noun-level feature by t-SNE method [25]. (a) and (b) are the two examples for the output features of the node evolution module. The right regions are enlarged in left panels. The categories shared with the similar knowledge are closed to each other. This verifies that our knowledge variants learn the corresponding knowledge. Please zoom in the colored PDF version of this paper for more details.



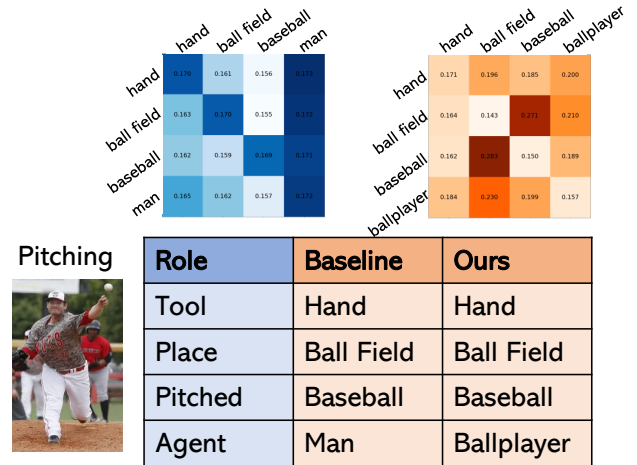
(a)



(b)



(c)



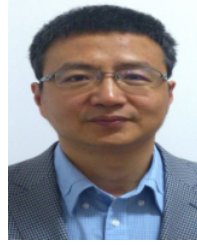
(d)

Fig. 10: Visualization of the learned graph between the baseline (blue matrix) and our method (orange matrix). We compare different graphs to show the superior performance of our action-guided pairwise graph generated by our ELM. Our method can predict more precise noun values (e.g., Man vs Teacher, Man vs Groom, Man vs Chief, Man vs Ballplayer).



Weijiang Yu is a Ph.D. student from the School of Computer Science and Engineering, Sun Yat-sen University, advised by Nong Xiao. He has closely collaborated with Prof. Bernard Ghanem in the KAUST. His research interests mainly include interactive vision system, structural representation learning, and multimodality like visual question answering, visual commonsense reasoning, video captioning and understanding. Currently, he prefers to explore the pre-trained model (Transformer-based) for cognitive reasoning in video-based multimodality.

ing in video-based multimodality.



Nong Xiao is a Professor in the School of Computer Science and Engineering, Sun Yat-sen University. He is the Deputy Director of working Committee of China Computer Society, Deputy Director of Information Storage Professional Committee of China Computer Society, Standing member of Big Data Expert Committee and member of High Performance Computing Professional Committee. His research interests include grid computing, cloud computing, big-data processing and machine learning. Prof.

Xiao's awards and honors include National Science Foundation for Distinguished Young Scholars, Chang Jiang Scholars Program, and the second prize of the National Science and Technology Progress award.

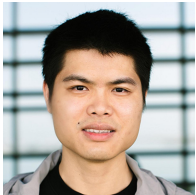


Haofang Wang is a Master student from Department of Electrical and Computer Engineering, Carnegie Mellon University. He has also worked with Prof. Bernard Ghanem at KAUST and Prof. Xia (ben) Hu at Texas A&M University. His research interests include cross-modal learning, natural language understanding and interpretability.



Bernard Ghanem is currently an Associate Professor in the CEMSE division, a theme leader at the Visual Computing Center (VCC), and the Deputy Director of the AI Initiative at King Abdullah University of Science and Technology (KAUST). His research interests lie in computer vision and machine learning with emphasis on topics in video understanding, 3D recognition, and theoretical foundations of deep learning. He received his Bachelor's degree from the American University of Beirut (AUB) in 2005 and his

MS/PhD from the University of Illinois at Urbana-Champaign (UIUC) in 2010. His work has received several awards and honors, including four Best Paper Awards for workshops in CVPR 2013&2019 and ECCV 2018&2020, a Google Faculty Research Award in 2015 (1st in MENA for Machine Perception), and a Abdul Hameed Shoman Arab Researchers Award for Big Data and Machine Learning in 2020. He has co-authored more than 120 peer reviewed conference and journal papers in his field as well as three issued patents. He serves as an Associate Editor for IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) and has served as Area Chair (AC) for CVPR 2018/2021, ICCV 2019/2021, ICLR 2021, and AAAI 2021.



Guohao Li obtained his BEng degree in Communication Engineering from Harbin Institute of Technology in 2015. In 2018, he received his Master Degree in Communication and Information Systems from Chinese Academy of Science. He was a research intern at SenseTime and Intel ISL. He is currently a CS PhD student at King Abdullah University of Science and Technology. His primary research interests are Computer Vision, Robotics and Deep Learning.