Graph R-CNN for Scene Graph Generation

- ECCV 2018, Georgia Institute of Technology & Facebook AI Research

Boeun Kim

Seoul National University

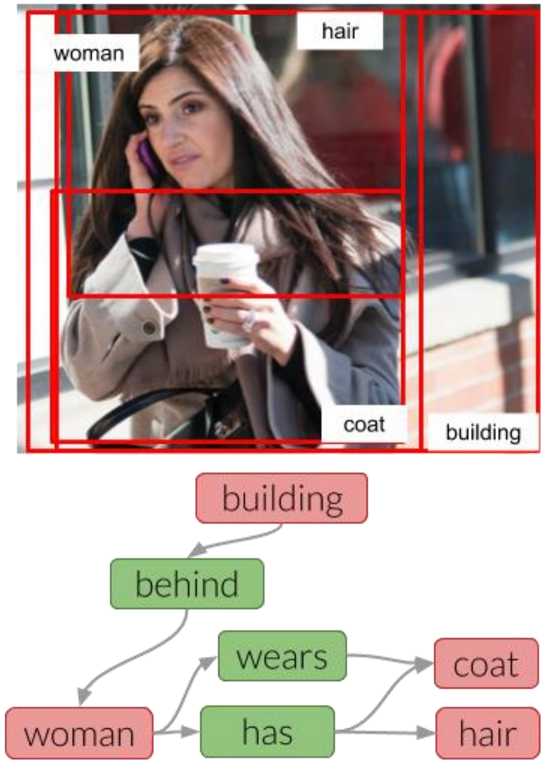
Introduction

- typical Visual Scene Understanding tasks
  - image classification
  - object detection
  - image segmentation

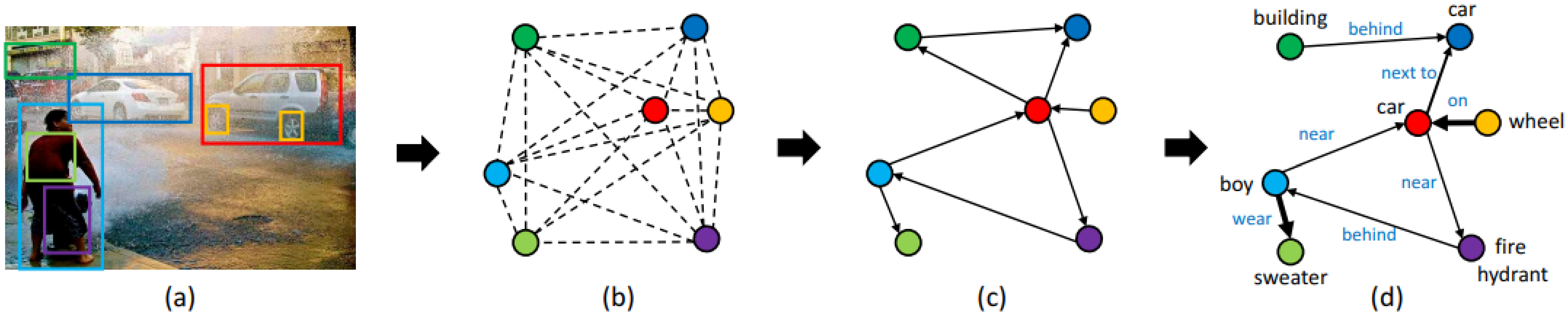
- Scene Graph Generation
  - objects & relationships

- support higher-level tasks
  - image captioning
  - visual question answering
  - image grounded-dialogue

scene graph



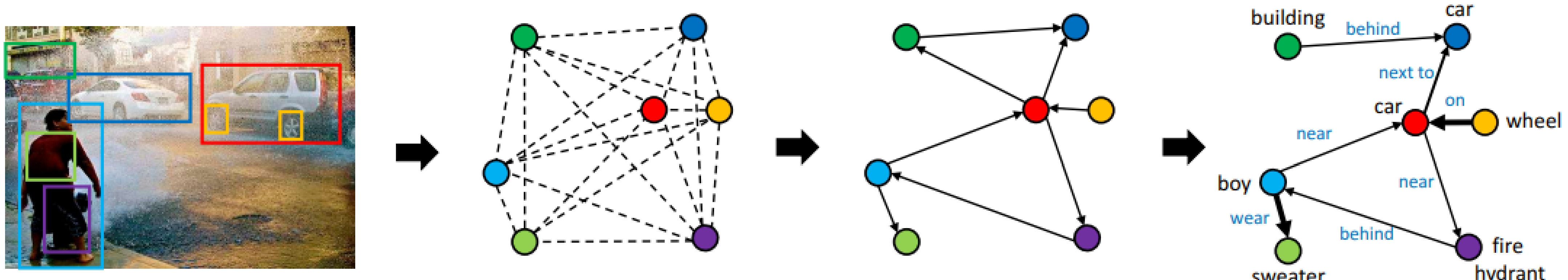
Graph R-CNN



- How can we effectively deal with fully-connected graph?
  → Naïve approach : random sampling

- (a)→(b) : object node extraction (R-CNN)
- (b)→(c) : relationship pruning (RePN)
- (c)→(d) : graph context integration (aGCN)

Graph R-CNN



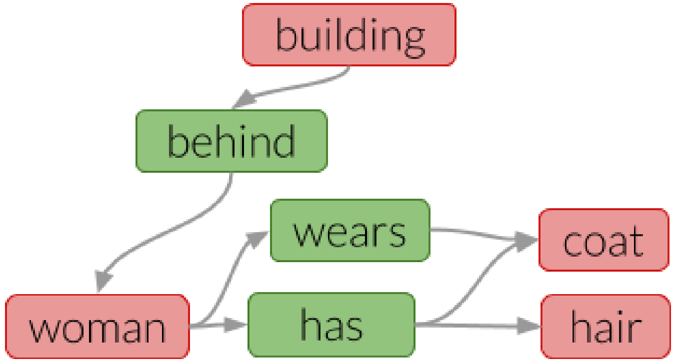
(a) (b) (c) (d)

- How can we effectively deal with fully-connected graph?
  → Naïve approach : random sampling

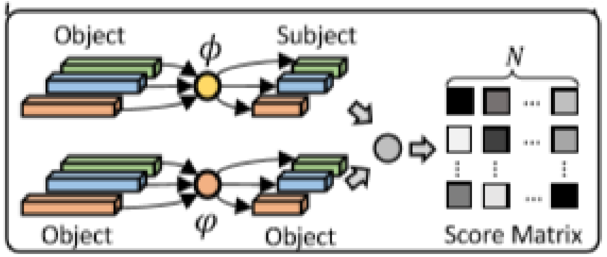
- (a)→(b) : object node extraction (R-CNN)
- (b)→(c) : relationship pruning (RePN)
- (c)→(d) : graph context integration (aGCN)

Graph R-CNN

- graph G
  - nodes : N object regions + m relationships
  - edges : relationship ↔ subject
    object ↔ relationship
    object ↔ object
  - object directional pairs : <subject, relationship, object>



Relation Proposal Network (RePN)



Relational Proposal Network

- estimate relatedness for all pairs

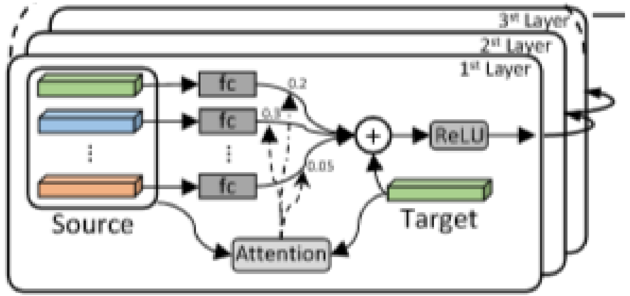
$$s_{ij} = f\left(\boldsymbol{P}_i^o, \boldsymbol{P}_j^o\right) = < \Phi(\boldsymbol{P}_i^o), \Psi\left(\boldsymbol{P}_j^o\right) >$$

where $\boldsymbol{P}_i^o$: class distribution of object i

$\Phi(\cdot), \Psi(\cdot)$: projection functions for subjects and objects

- leave top K pairs

Attentional Graph Convolution Network (aGCN)



Attentional GCNs

- layer-wise propagation

$$z_i^{(l+1)} = \sigma\left(z_i^{(l)} + \sum_{j \in N(i)} \alpha_{ij} W z_j^{(l)}\right)$$

where $\boldsymbol{z}_{\boldsymbol{i}}^{(l)}$: node representation of i in the layer l

- Learning to adjust $\boldsymbol{\alpha}$
  - In conventional GCN, connections in the graph are known → $\boldsymbol{\alpha}$ is preset

$$u_{ij} = w_h^T \sigma\left(W_a [z_i^{(l)}, z_j^{(l)}]\right)$$

$$\boldsymbol{\alpha}_i = softmax(\boldsymbol{u}_i)$$

Attentional Graph Convolution Network (aGCN)

$$z_i^{(l+1)} = \sigma\left(z_i^{(l)} + \sum_{j\in N(i)} \alpha_{ij} W z_i^{(l)}\right)$$

$$u_{ij} = w_h^T \sigma\left(W_a[z_i^{(l)}, z_j^{(l)}]\right)$$

$$\boldsymbol{\alpha}_i = softmax(\boldsymbol{u}_i)$$

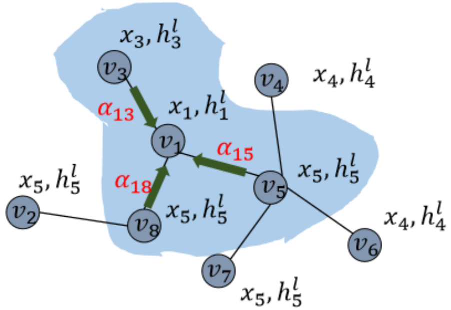
Remind

GCN: Filter in GAT (Graph ATtention Networks)

Aggregation:

$$\boldsymbol{h}_i^{(l+1)} = \boldsymbol{\sigma}\left(\boldsymbol{\Theta} \sum_{v_j\in N(v_i)} \boxed{\alpha_{ij}} \boldsymbol{h}_j^{(l)}\right)$$

$$\boxed{\alpha_{ij}} = \frac{exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{\Theta}\cdot\boldsymbol{h}_j^{(l)}\right]\right)\right)}{\sum_{v_k\in N(v_i)} exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{\Theta}\cdot\boldsymbol{h}_k^{(l)}\right]\right)\right)}$$



$\boldsymbol{a}, \boldsymbol{\Theta}$: parameters of a single layer neural network

GAT: Graph Attention Networks (https://arxiv.org/pdf/1710.10903.pdf)
(Petar Velickovic et al. ICLR 2018)

J. Y. Choi. SNU

Attentional Graph Convolution Network (aGCN)

$$z_i^{(l+1)} = \sigma\left(z_i^{(l)} + \sum_{j \in N(i)} \alpha_{ij} W z_i^{(l)}\right)$$

$$u_{ij} = w_h^T \sigma\left(W_a[z_i^{(l)}, z_j^{(l)}]\right)$$

$$\boldsymbol{\alpha}_i = softmax(\boldsymbol{u}_i)$$

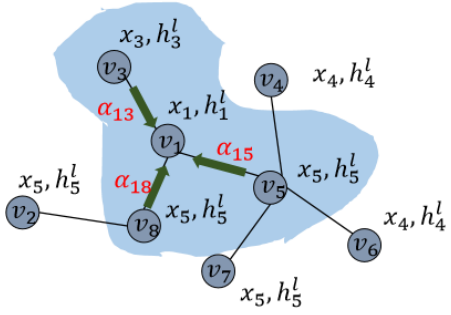
Remind

GCN: Filter in GAT (Graph ATtention Networks)

Aggregation:

$$\boldsymbol{h}_i^{(l+1)} = \boldsymbol{\sigma}\left(\boldsymbol{\Theta} \sum_{v_j \in N(v_i)} \boxed{\alpha_{ij}} \boldsymbol{h}_j^{(l)}\right)$$

$$\boxed{\alpha_{ij}} = \frac{exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{\Theta}\cdot\boldsymbol{h}_j^{(l)}\right]\right)\right)}{\sum_{v_k \in N(v_i)} exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{\Theta}\cdot\boldsymbol{h}_k^{(l)}\right]\right)\right)}$$



$\boldsymbol{a}, \boldsymbol{\Theta}$: parameters of a single layer neural network

GAT: Graph Attention Networks (https://arxiv.org/pdf/1710.10903.pdf)
(Petar Velickovic et al. ICLR 2018)

J. Y. Choi. SNU

Attentional Graph Convolution Network (aGCN)

$$z_i^{(l+1)} = \sigma\left(z_i^{(l)} + \sum_{j \in N(i)} \alpha_{ij} W z_i^{(l)}\right)$$

$$u_{ij} = w_h^T \sigma\left(W_a[z_i^{(l)}, z_j^{(l)}]\right)$$

$$\boldsymbol{\alpha}_i = softmax(\boldsymbol{u}_i)$$

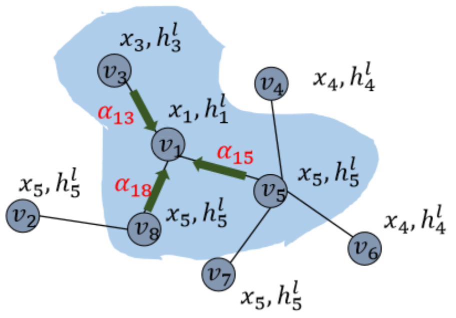
Remind

GCN: Filter in GAT (Graph ATtention Networks)

Aggregation:

$$\boldsymbol{h}_i^{(l+1)} = \boldsymbol{\sigma}\left(\boldsymbol{\Theta} \sum_{v_j \in N(v_i)} \boxed{\alpha_{ij}} \boldsymbol{h}_j^{(l)}\right)$$

$$\boxed{\alpha_{ij}} = \frac{exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{\Theta}\cdot\boldsymbol{h}_j^{(l)}\right]\right)\right)}{\sum_{v_k \in N(v_i)} exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \,\|\, \boldsymbol{\Theta}\cdot\boldsymbol{h}_k^{(l)}\right]\right)\right)}$$



$\boldsymbol{a}, \boldsymbol{\Theta}$: parameters of a single layer neural network

GAT: Graph Attention Networks (https://arxiv.org/pdf/1710.10903.pdf)
(Petar Velickovic et al. ICLR 2018)

J. Y. Choi. SNU

Attentional Graph Convolution Network (aGCN)

$$z_i^{(l+1)} = \sigma\left(z_i^{(l)} + \sum_{j \in N(i)} \alpha_{ij} W z_i^{(l)}\right)$$

$$u_{ij} = w_h^T \sigma\left(W_a[z_i^{(l)}, z_j^{(l)}]\right)$$

$$\boldsymbol{\alpha}_i = softmax(\boldsymbol{u}_i)$$

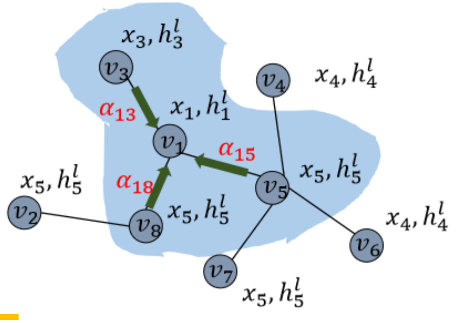
Remind

GCN: Filter in GAT (Graph ATtention Networks)

Aggregation:

$$\boldsymbol{h}_i^{(l+1)} = \boldsymbol{\sigma}\left(\boldsymbol{\Theta} \sum_{v_j \in N(v_i)} \boxed{\alpha_{ij}} \boldsymbol{h}_j^{(l)}\right)$$

$$\boxed{\alpha_{ij}} = \frac{exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \| \boldsymbol{\Theta}\cdot\boldsymbol{h}_j^{(l)}\right]\right)\right)}{\sum_{v_k \in N(v_i)} exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{\Theta}\cdot\boldsymbol{h}_i^{(l)} \| \boldsymbol{\Theta}\cdot\boldsymbol{h}_k^{(l)}\right]\right)\right)}$$



$\boldsymbol{a}, \boldsymbol{\Theta}$: parameters of a single layer neural network

GAT: Graph Attention Networks (https://arxiv.org/pdf/1710.10903.pdf)
(Petar Velickovic et al. ICLR 2018)

J. Y. Choi. SNU

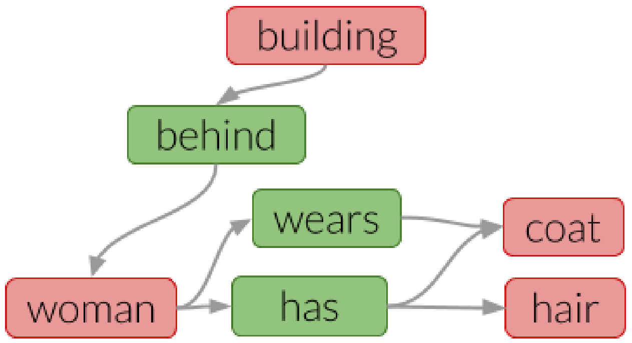
Attentional Graph Convolution Network (aGCN)

$$z_i^o = \sigma(\underbrace{W^{skip} Z^o \alpha^{skip}}_{\text{Message from other objects}} + \underbrace{W^{sr} Z^r \alpha^{sr} + W^{or} Z^r \alpha^{or}}_{\text{Messages from neighboring relationships}})$$

$$z_i^r = \sigma(z_i^r + \underbrace{W^{rs} Z^o \alpha^{rs} + W^{ro} Z^o \alpha^{ro}}_{\text{Messages from Neighboring objects}})$$

where $\boldsymbol{Z^o}, \boldsymbol{Z^r}$: node representation of object and relationship
s: subject, o: objects, r: relationships

- edges : object ↔ relationship
  relationship ↔ subject
  object ↔ object



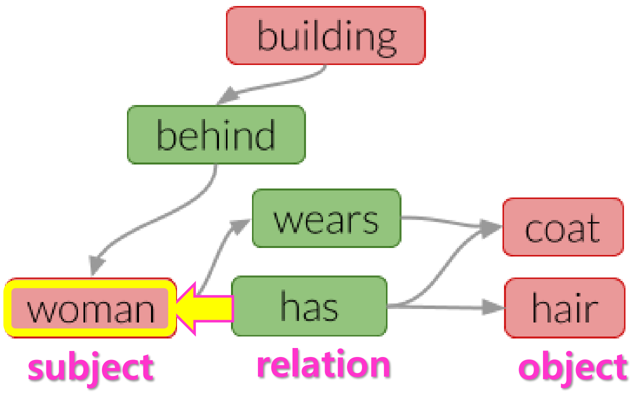
Attentional Graph Convolution Network (aGCN)

$$z_i^o = \sigma(\underbrace{W^{skip} Z^o \alpha^{skip}}_{\text{Message from other objects}} + \underbrace{\boxed{W^{sr} Z^r \alpha^{sr}} + W^{or} Z^r \alpha^{or}}_{\text{Messages from neighboring relationships}})$$

$$z_i^r = \sigma(z_i^r + \underbrace{W^{rs} Z^o \alpha^{rs} + W^{ro} Z^o \alpha^{ro}}_{\text{Messages from Neighboring objects}})$$

where $\boldsymbol{Z^o}, \boldsymbol{Z^r}$: node representation of object and relationship
s: subject, o: objects, r: relationships

- edges : object ↔ relationship
  relationship ↔ subject
  object ↔ object



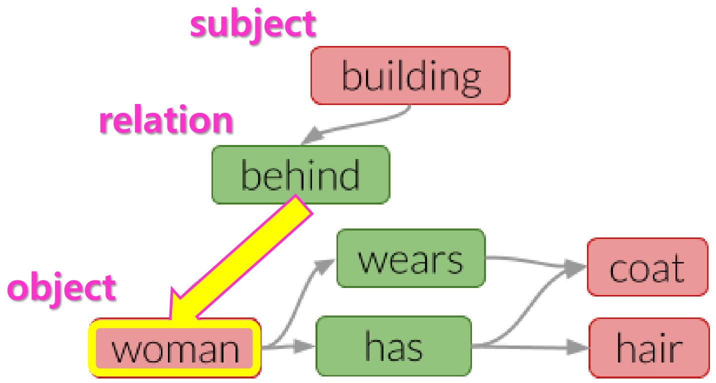
Attentional Graph Convolution Network (aGCN)

$$z_i^o = \sigma(\underbrace{W^{skip} Z^o \alpha^{skip}}_{\text{Message from other objects}} + \underbrace{W^{sr} Z^r \alpha^{sr} + \boxed{W^{or} Z^r \alpha^{or}}}_{\text{Messages from neighboring relationships}})$$

$$z_i^r = \sigma(z_i^r + \underbrace{W^{rs} Z^o \alpha^{rs} + W^{ro} Z^o \alpha^{ro}}_{\text{Messages from Neighboring objects}})$$

where $\boldsymbol{Z^o}, \boldsymbol{Z^r}$: node representation of object and relationship
s: subject, o: objects, r: relationships

- edges : <u>object ↔ relationship</u>
  relationship ↔ subject
  object ↔ object



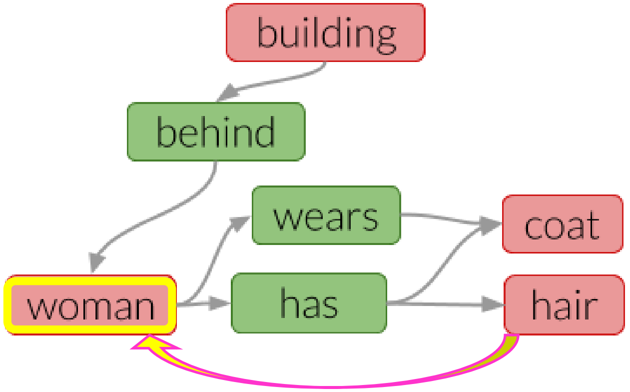
Attentional Graph Convolution Network (aGCN)

$$z_i^o = \sigma(\underbrace{\boxed{W^{skip} Z^o \alpha^{skip}}}_{\text{Message from other objects}} + \underbrace{W^{sr} Z^r \alpha^{sr} + W^{or} Z^r \alpha^{or}}_{\text{Messages from neighboring relationships}})$$

$$z_i^r = \sigma(z_i^r + \underbrace{W^{rs} Z^o \alpha^{rs} + W^{ro} Z^o \alpha^{ro}}_{\text{Messages from Neighboring objects}})$$

where $\boldsymbol{Z^o}, \boldsymbol{Z^r}$: node representation of object and relationship
s: subject, o: objects, r: relationships

- edges : object ↔ relationship
  relationship ↔ subject
  object ↔ object



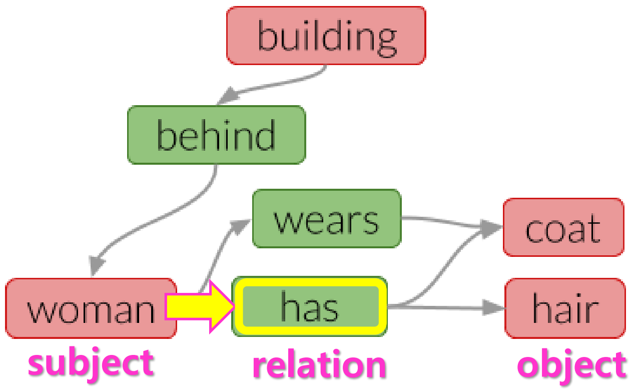
Attentional Graph Convolution Network (aGCN)

$$z_i^o = \sigma(\underbrace{W^{skip} Z^o \alpha^{skip}}_{\text{Message from other objects}} + \underbrace{W^{sr} Z^r \alpha^{sr} + W^{or} Z^r \alpha^{or}}_{\text{Messages from neighboring relationships}})$$

$$z_i^r = \sigma(z_i^r + \underbrace{\boxed{W^{rs} Z^o \alpha^{rs}} + W^{ro} Z^o \alpha^{ro}}_{\text{Messages from Neighboring objects}})$$

where $\boldsymbol{Z^o}, \boldsymbol{Z^r}$: node representation of object and relationship
s: subject, o: objects, r: relationships

- edges : object ↔ relationship
  relationship ↔ subject
  object ↔ object



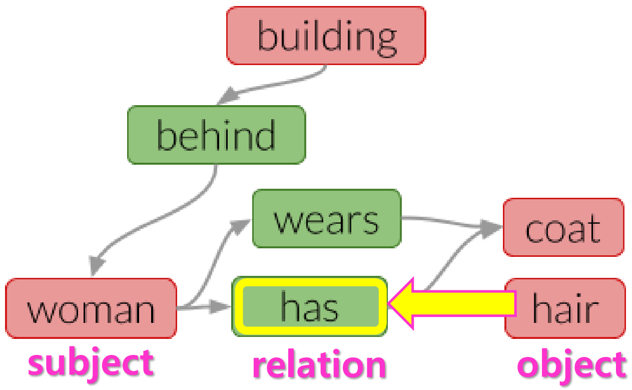
Attentional Graph Convolution Network (aGCN)

$$z_i^o = \sigma(\underbrace{W^{skip}Z^o\alpha^{skip}}_{\text{Message from other objects}} + \underbrace{W^{sr}Z^r\alpha^{sr} + W^{or}Z^r\alpha^{or}}_{\text{Messages from neighboring relationships}})$$

$$z_i^r = \sigma(z_i^r + \underbrace{W^{rs}Z^o\alpha^{rs} + \boxed{W^{ro}Z^o\alpha^{ro}}}_{\text{Messages from Neighboring objects}})$$

where $\boldsymbol{Z^o}, \boldsymbol{Z^r}$: node representation of object and relationship
s: subject, o: objects, r: relationships

- edges : object ↔ relationship
  relationship ↔ subject
  object ↔ object

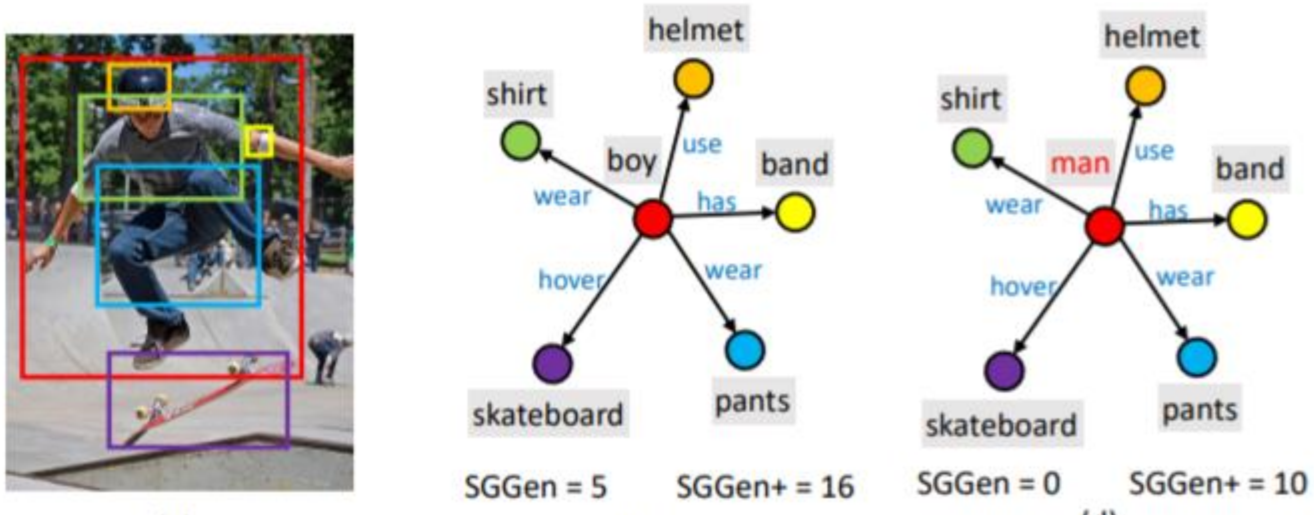


Evaluation metrics

- Recall

- SSGEN
  - exact triplet match
    - triplet <object, relationship, subject> labels
    - object and subject location

- SSGEN+
  - exact triplet match
  - + object & subject match
  - + relationship match



(a) (b) (d)

Experiments

- Visual Genome dataset
  - Training :75,000 / Test: 32,000 images
  - Top frequent 150 object classes / 50 relations

	SGGen+		SGGen		PhrCls		PredCls	
Method	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
IMP [40]	-	-	3.4	4.2	21.7	24.4	44.8	53.0
MSDN [18]	-	-	7.7	10.5	19.3	21.8	63.1	66.4
Pixel2Graph [26]	-	-	9.7	11.3	26.5	30.0	68.0	75.2
IMP† [40]	25.6	27.7	6.4	8.0	20.6	22.4	40.8	45.2
MSDN† [18]	25.8	28.2	7.0	9.1	27.6	29.9	53.2	57.9
NM-Freq† [42]	26.4	27.8	6.9	9.1	23.8	27.2	41.8	48.8
Graph R-CNN (Us)	**28.5**	**35.9**	**11.4**	**13.7**	**29.6**	**31.6**	**54.2**	**59.1**

Experiments

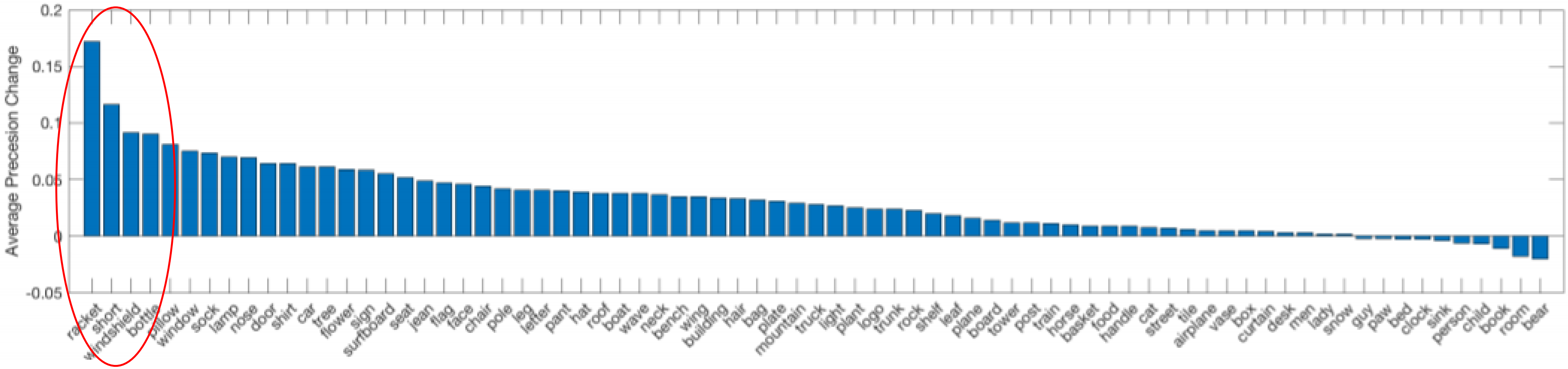
- Ablation study

RePN	GCN	aGCN	Detection	SGGen+		SGGen		PhrCls		PredCls	
			mAP@0.5	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
-	-	-	20.4	25.9	27.9	6.1	7.9	17.8	19.9	33.5	38.4
✓	-	-	**23.6**	27.6	34.8	8.7	11.1	18.3	20.4	34.5	39.5
✓	✓	-	23.4	28.1	35.3	10.8	13.4	27.2	29.5	52.3	57.2
✓	-	✓	23.0	**28.5**	**35.9**	**11.4**	**13.7**	**29.4**	**31.6**	**54.2**	**59.1**

Experiments

- Ablation study

RePN	GCN	aGCN	Detection	SGGen+		SGGen		PhrCls		PredCls	
			mAP@0.5	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
-	-	-	20.4	25.9	27.9	6.1	7.9	17.8	19.9	33.5	38.4
✓	-	-	**23.6**	27.6	34.8	8.7	11.1	18.3	20.4	34.5	39.5
✓	✓	-	23.4	28.1	35.3	10.8	13.4	27.2	29.5	52.3	57.2
✓	-	✓	23.0	**28.5**	**35.9**	**11.4**	**13.7**	**29.4**	**31.6**	**54.2**	**59.1**



Experiments

- Ablation study

RePN	GCN	aGCN	Detection	SGGen+		SGGen		PhrCls		PredCls	
			mAP@0.5	R@50	R@100	R@50	R@100	R@50	R@100	R@50	R@100
-	-	-	20.4	25.9	27.9	6.1	7.9	17.8	19.9	33.5	38.4
✓	-	-	**23.6**	27.6	34.8	8.7	11.1	18.3	20.4	34.5	39.5
✓	✓	-	23.4	28.1	35.3	10.8	13.4	27.2	29.5	52.3	57.2
✓	-	✓	23.0	**28.5**	**35.9**	**11.4**	**13.7**	**29.4**	**31.6**	**54.2**	**59.1**

Conclusion

- RePN intelligently prunes out pairs of objects that are unlikely to be related.
- aGCN effectively propagates contextual information across the graph.