AAAI 23

Unbiased Heterogeneous Scene Graph Generation with Relation-aware Message Passing Neural Network

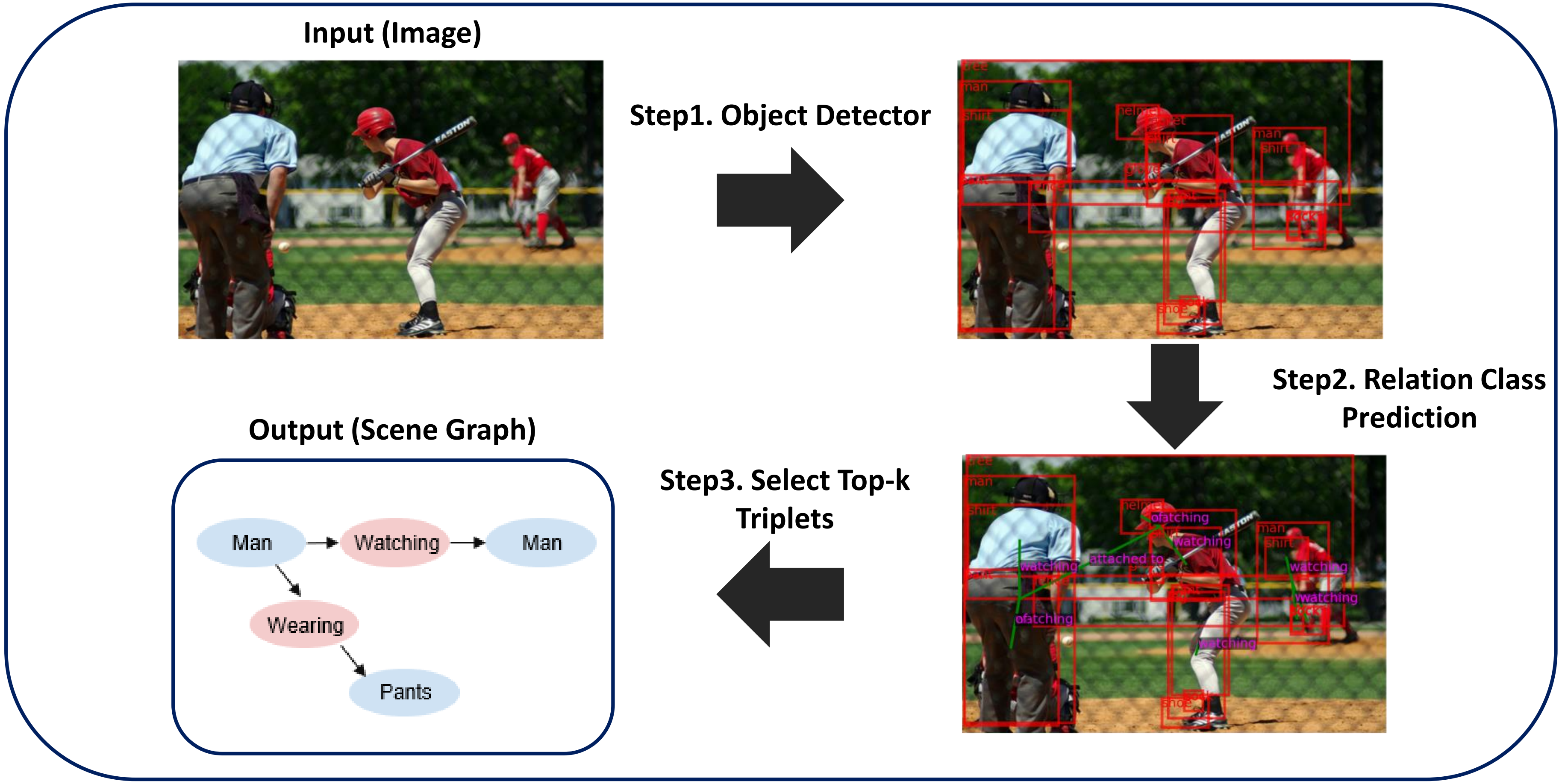
Kanghoon Yoon ,Kibum Kim
Jinyoung Moon and Chanyoung Park

Korea Advanced Institute of Science and Technology (KAIST)
Electronics and Telecommunications Research Institute (ETRI)

KAIST DSAIL Data Science & Artificial Intelligence ETRI Electronics and Telecommunications Research Institute

SCENE GRAPH GENERATION (SGG)

- **SGG** aims to represent observable knowledges in an image in the form of a graph
  - The Knowledges include 1) object information and 2) their relation information
    - E.g., Object information: *man, horse, glasses, …* Relation information between objects: *feeding, wearing, …*

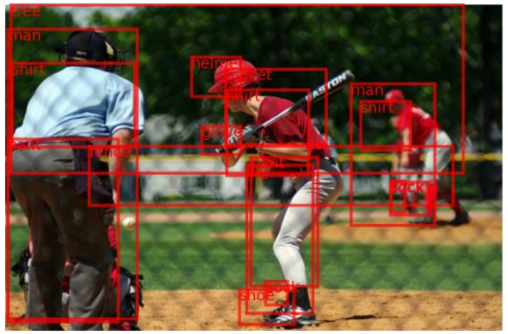


SCENE GRAPH GENERATION (SGG)

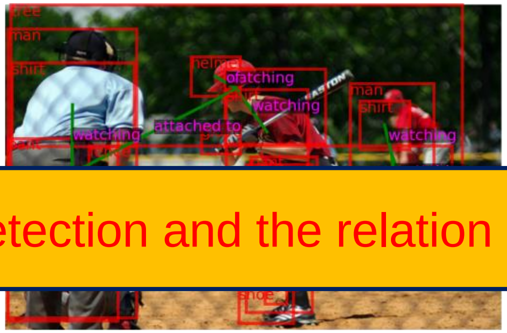
- **SGG** aims to represent observable knowledges in an image in the form of a graph
  - The Knowledges include 1) object information and 2) their relation information
    - E.g., Object information: *man, horse, glasses, …*   Relation information between objects: *feeding, wearing, …*

Input (Image)

Step1. Object Detector

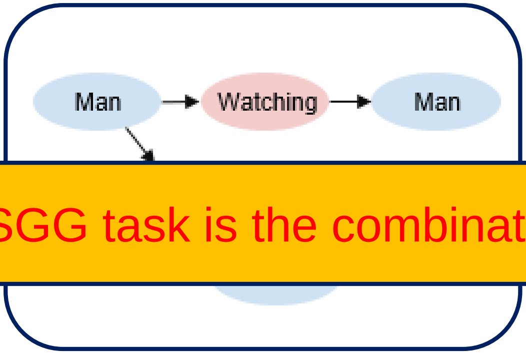


Step2. Relation Class Prediction



Step3. Select Top-k Triplets

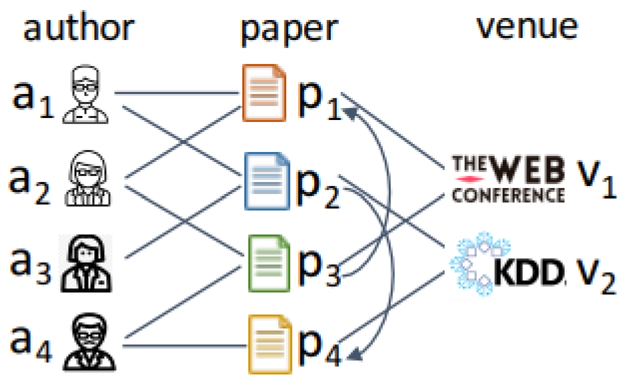
Output (Scene Graph)



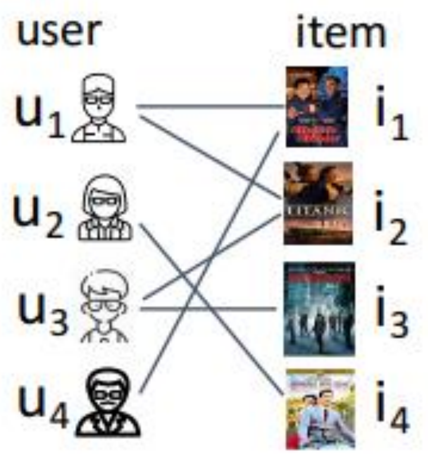
In summary, SGG task is the combination of the object detection and the relation classification!

HETEROGENEOUS GRAPH

- **Heterogeneous graph** is a graph-structured data with more than one type of nodes or edges
  - By considering associations between multiple types of nodes or edges, many works demonstrate that considering the heterogeneity of nodes/edges are helpful for learning the representations with the semantic information.



[Academic Graph]

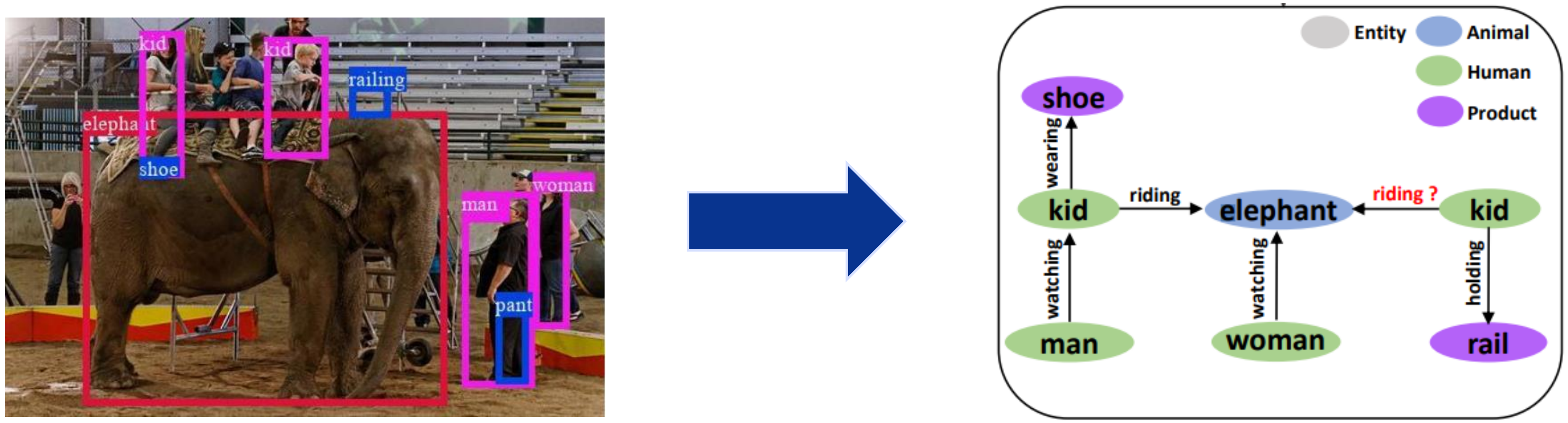


[Review Graph]

[KDD'19] Heterogeneous Graph Neural Network. Zhang et al.

PREVIOUS WORKS

- **In the literature of SGG, it's important to capture the context of neighborhood**
  - Considering *<kid, holding, rail>* and *<woman, watching, elephant>* is helpful for predicting *<kid, riding, elephant>*
    - Compared with when **kid** and **elephant** are considered independently

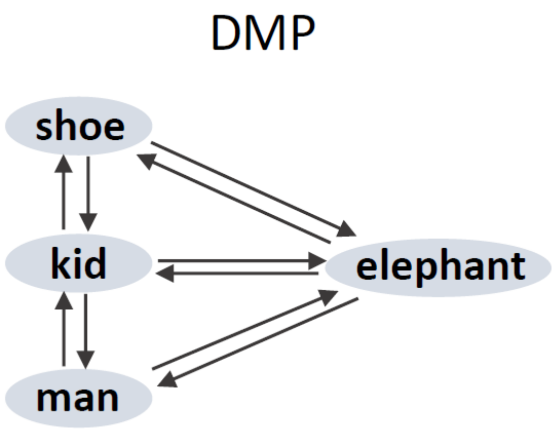


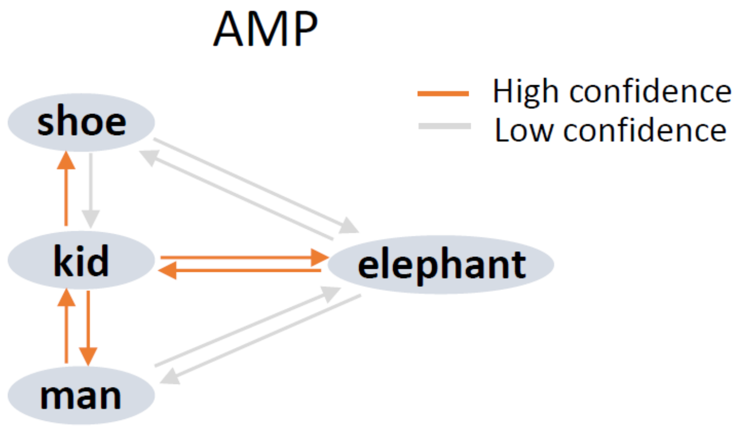
[Example of a context-aware model]

- Context-aware SGG employs RNN, GNN, …, Transformer to aggregate features of neighboring objects.

PREVIOUS WORKS

- **Moreover, recent works for context-aware SGG adopts Message-passing Neural Network**
  - Direction-aware MPNN (DMP) passes the messages according to the direction [1]
    - Treats messages of (subject → object), (object → subject) differently
  - Adaptive Message Passing (AMP) filters unnecessary messages based on the structure of a scene graph [2]



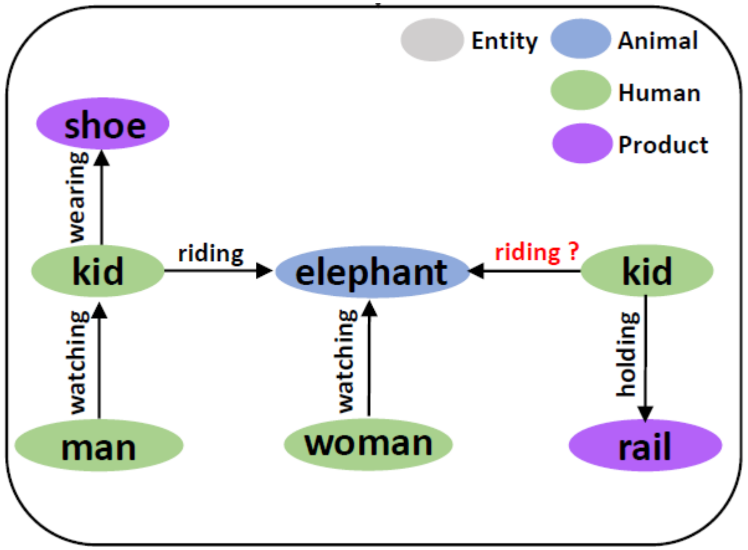


  - Other Models such as Transformer , …, etc.

[CVPR’20] GPS-Net: Graph Property Sensing Network for Scene Graph Generation. Lin et al. [1]
[CVPR’21] Bipartite Graph Network with Adaptive Message Passing for Unbiased Scene Graph Generation. Li et al. [2]

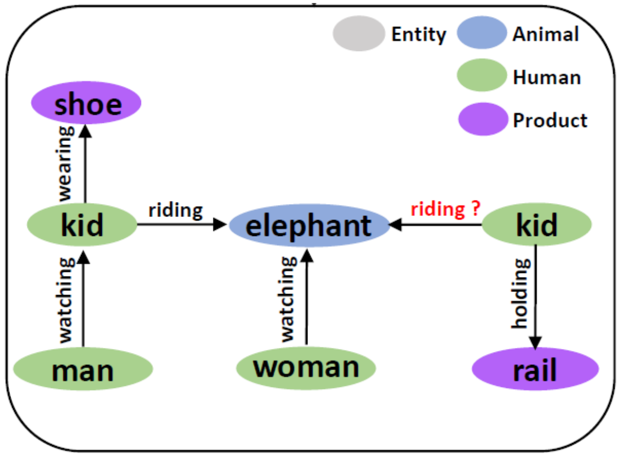
LIMITATIONS OF PREVIOUS WORKS

- **Previous works consider the scene graph as homogeneous graph**
  - The assumption of homogeneity restricts the context-awareness of the visual relations between objects.
    - Since it neglects the fact that predicates highly dependent on the objects where the predicates are associated.
    - For example, when we consider *<kid, riding, elephant>*, we know the opposite triplet *<elephant , riding, kid>* is not likely to appear.
    - Because it is usually "Human" that rides "Animal".

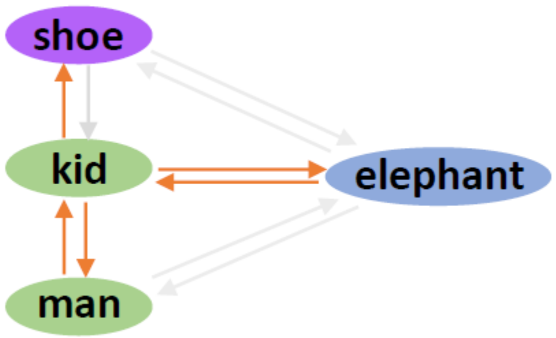


TACKLING PROBLEM

- **We propose the Heterogeneous scene graph generation (HetSGG) framework**
  - HetSGG generates a scene graph with relation-aware context
    - We consider both object types (e.g., Human, Animal, Product) & relation types (e.g., Human-Animal, Human-Human, …,).
  - We propose a novel message-passing called relation aware message-passing (RMP)
  - It can naturally capture the semantic between "Human" and "Animal" to predict <*kid, riding, elephant*>



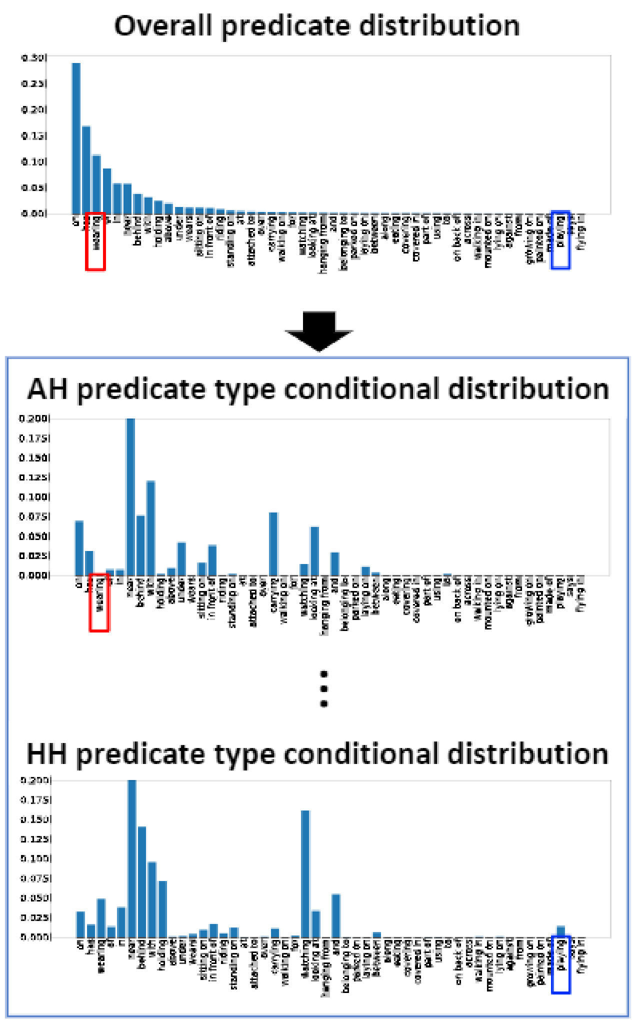
RMP



RELIEVING LONG-TAILEDNESS

- Overall predicate distribution is long-tailedness
  - Problem: Model primarily predicts the meaningless predicate (i.e., on, has)

- Observation of the reformulated distribution in condition of predicate types
  - **Animal-Human**(AH): head predicate (e.g., "wearing") in overall distribution becomes tail predicate in AH distribution
  - **Human-Human**(HH): tail predicate (e.g., "playing") makes up a small proportion of the overall distribution, but the proportion improves in HH distribution

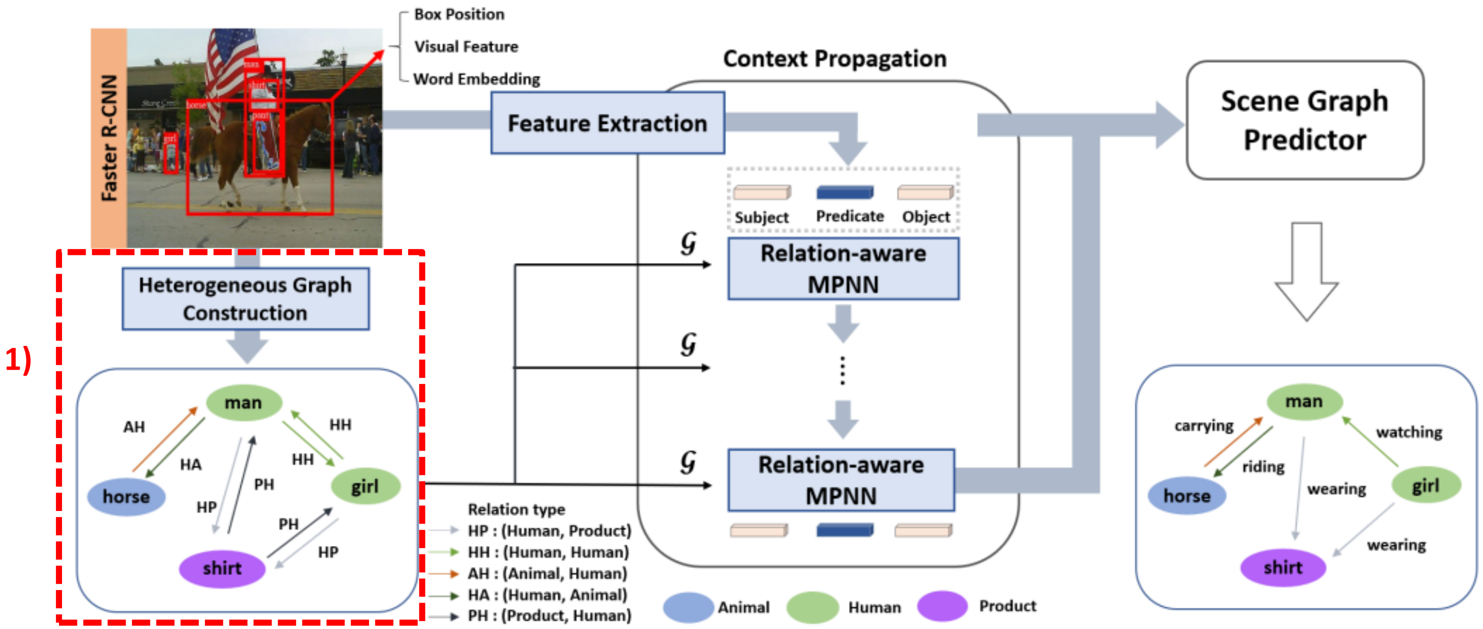


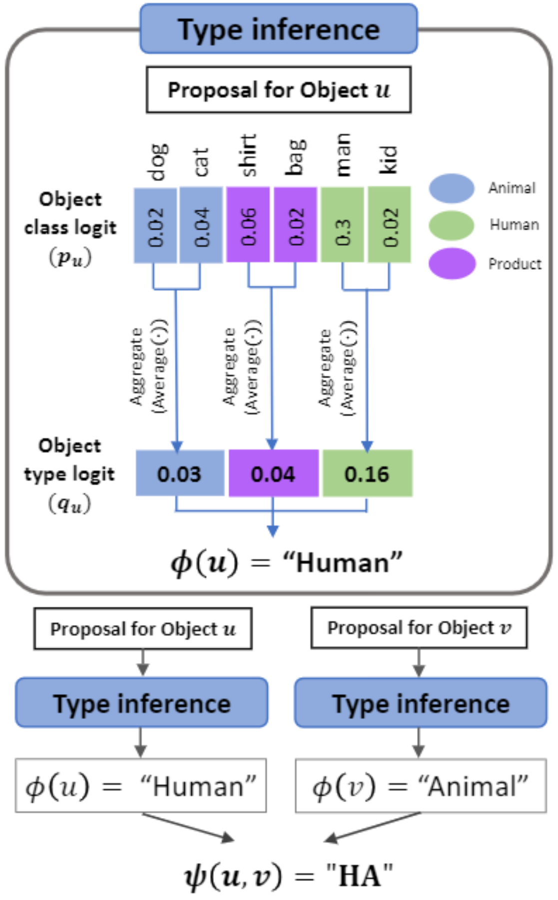
We expect the long-tailed problem is naturally alleviated in the formulation of heterogeneous graph distinguishing the relation type

HETSGG: (1) HETEROGENEOUS GRAPH CONSTRUCTION

- **1) Construct the heterogeneous graph based on the detector**

- Estimate the object type, utilizing the object class logit which is the output of Faster R-CNN
  - Assign the object type with the highest logit value by averaging the logits for each object type's corresponding class
  - Assign the relation type by Cartesian product of object type, e.g., Human, Animal => HA

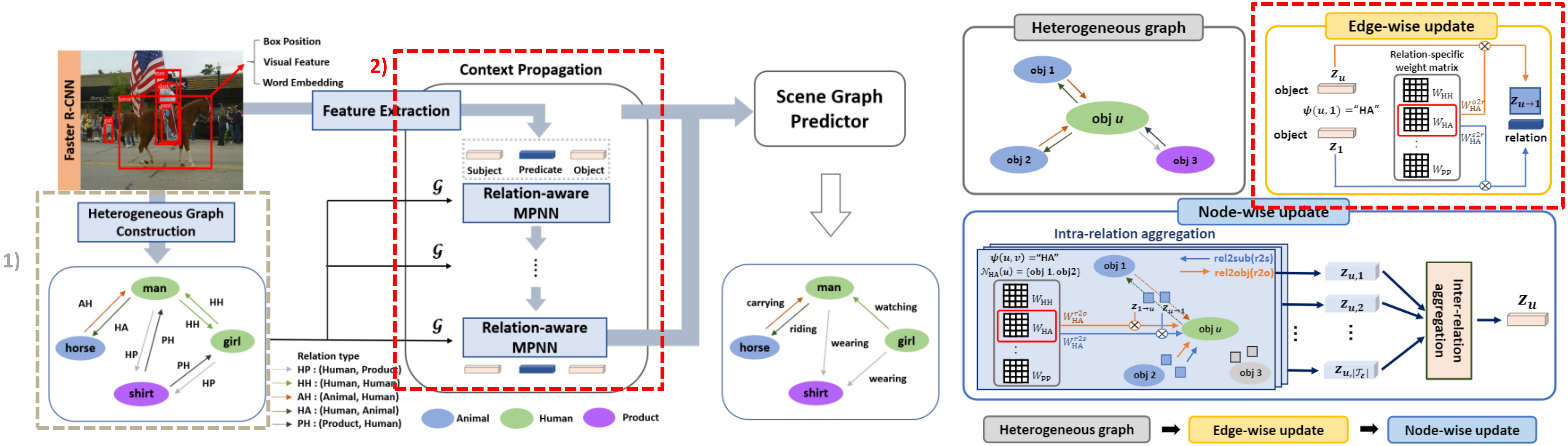




HETSGG: (2) RELATION-AWARE MPNN (RMP)

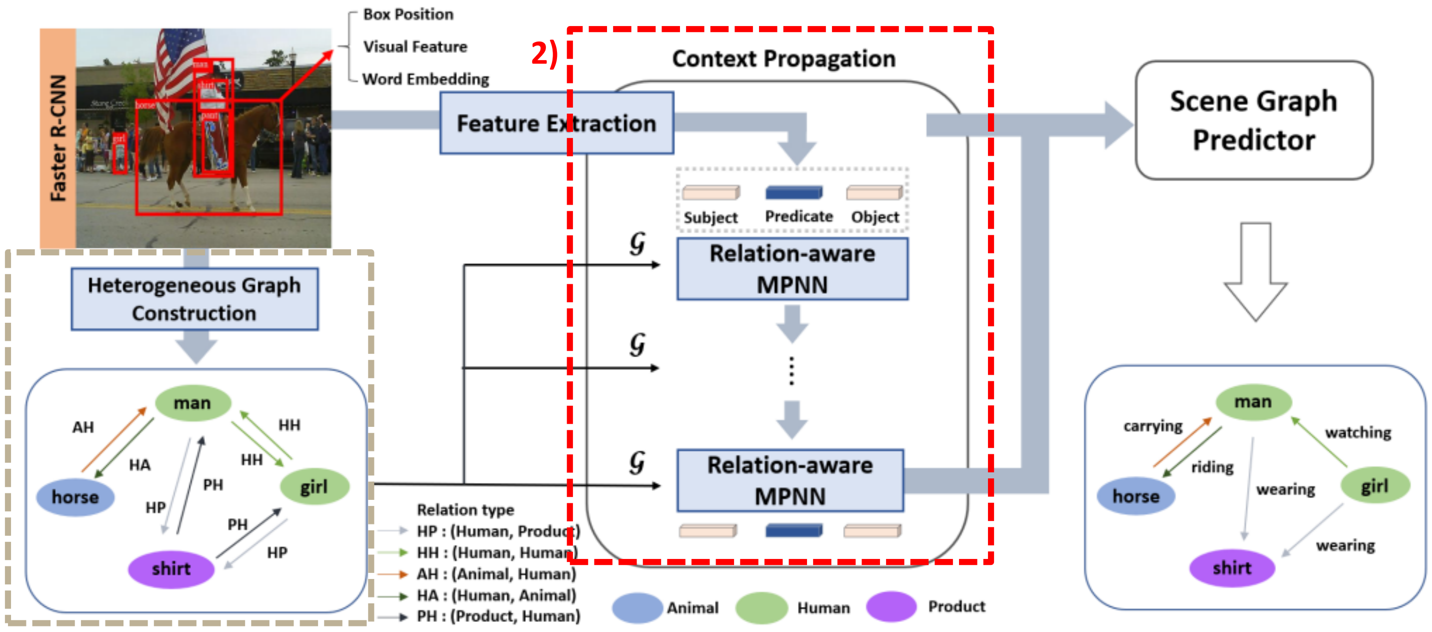
- **2) RMP (Relation-aware MPNN): Propagate the messages considering the relation type**

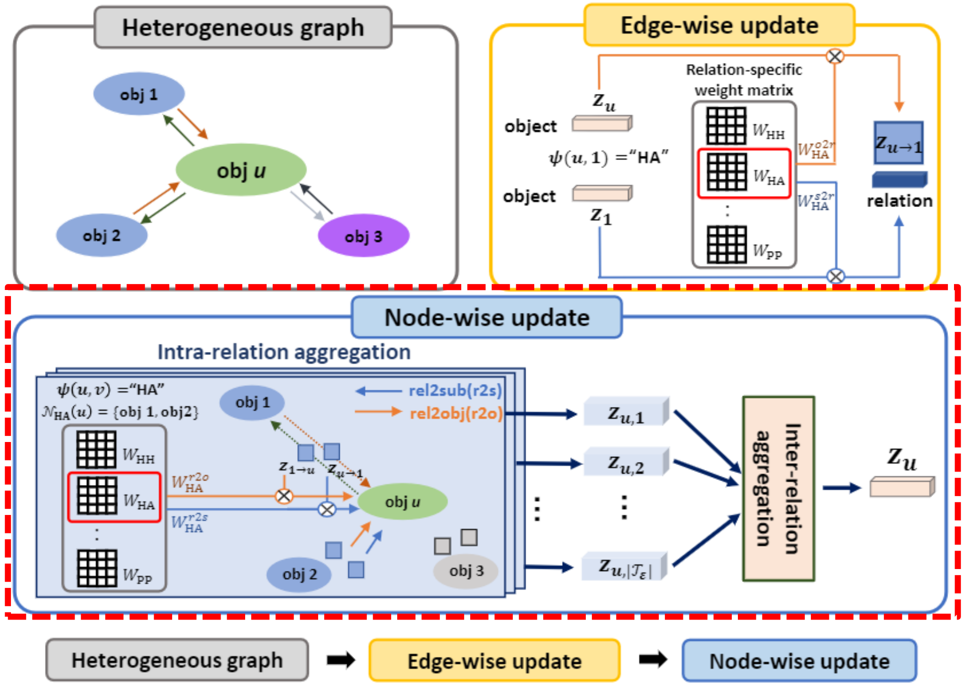
- Take 2 step for RMP: **1) Edge-wise update** 2) Node-wise update
  - To update the edge (relation) feature, propagate the subject→edge (sub2rel) and object→edge (obj2rel) messages
  - Utilize the different weight matrix to differentiate the relation type and propagate messages
    - E.g., “Human”, “Animal” ⇒ $W_{HA}^{sub2rel}$, $W_{AH}^{obj2rel}$ parameter recognize their relation type



HETSGG: (2) RELATION-AWARE MPNN (RMP)

- **2) RMP (Relation-aware MPNN): Aggregate and Propagate the messages considering the relation type**
- Take 2 step for RMP: 1) Edge-wise update **2) Node-wise update**
  - Aggregation Step: a) Intra-relation aggregation and b) Inter-relation aggregation. (Similarly, use the different weight matrix for relation types)
    - **a) Intra-relation aggregation**: Aggregate messages of the neighboring entity with the same relation type
    - **b) Inter-relation aggregation**: Aggregate messages that are generated through the intra-relation aggregation





HETSGG: (2) RELATION-AWARE MPNN (RMP)

- However, utilizing the different parameters increases our model complexity
  - E.g., $W^{sub2rel}$ parameter is split into $W_{HA}^{sub2rel}, W_{AH}^{sub2rel}, W_{PA}^{sub2rel}$, …
  - The model complexity increases 9 (3×3) times

- Solution: Use the relation-specific weight matrix that consist of bases ($b \ll 9$) as in [1]
  - $W_t = \sum_{i=1}^{b} a_{ti} B_i$, t denotes the relation types, e.g., HA
  - B_i is shared parameter across the relation type. a_{ti} coefficient is assigned to each relation type

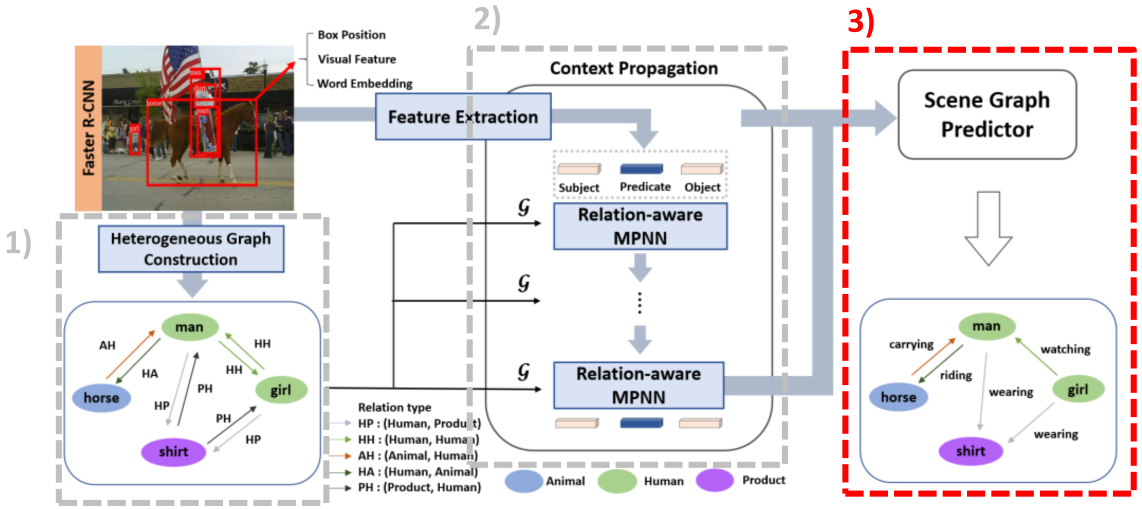
[ESWC'18] Modeling Relational Data with Graph Convolutional Networks [1]

HETSGG: TRAINING & INFERENCE

- **3) Training or Inference with refined object and relation representation**

- Training: $L_{final} = L_{obj} + L_{rel}$
  - L_{obj}: Classification loss of object
  - L_{rel}: Classification of relation

- Inference
  - Assign the object or relation class with highest logits

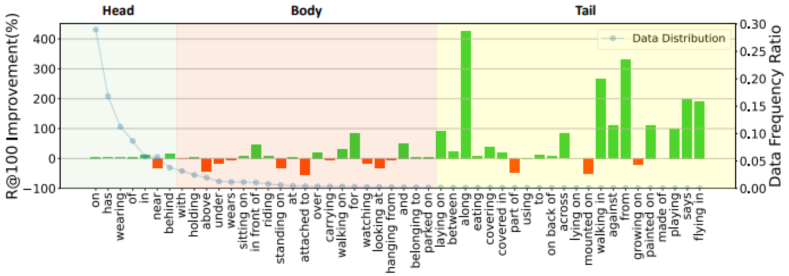


EXPERIMENT: COMPARISON WITH SOTA MODEL

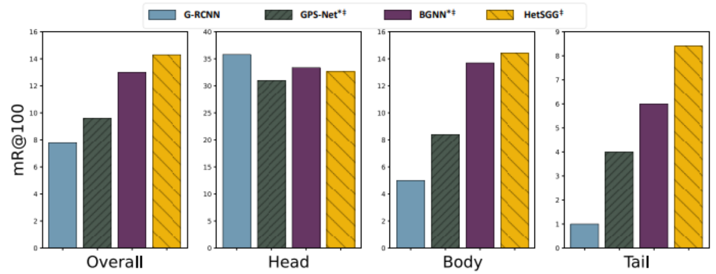
- Metric
  - Recall (R@K): Overall ratio of predicting the correct ground-truth triplet (Performance for head predicates)
  - Mean Recall (mR@K): Average of each predicate's recall (Performance for tail predicates)
- **HetSGG enhances mean mR@K while showing competitive R@K**
  - It improves performance for tail predicates, maintaining the performance for head predicates

Models	PredCls		SGCls		SGGen	
	mR@50/100	R@50/100	mR@50/100	R@50/100	mR@50/100	R@50/100
RelDN (Zhang et al. 2019b)	15.8/17.2	64.8/66.7	9.3/9.6	38.1/39.3	6.0/7.3	31.4/35.9
Motifs (Zellers et al. 2018)	14.6/15.8	66.0/67.9	8.0/8.5	39.1/39.9	5.5/6.8	32.1/36.9
VCTree (Tang et al. 2019)	15.4/16.6	65.5/67.4	7.4/7.9	38.9/39.8	6.6/7.7	31.8/36.1
G-RCNN (Yang et al. 2018)	16.4/17.2	65.4/67.2	9.0/9.5	37.0/38.5	5.8/6.6	29.7/32.8
MSDN (Li et al. 2017)	15.9/17.5	64.6/66.6	9.3/9.7	38.4/39.8	6.1/7.2	31.9/36.6
Unbiased (Tang et al. 2020)	25.4/28.7	47.2/51.6	12.2/14.0	25.4/27.9	9.3/11.1	19.4/23.2
GPS-Net (Lin et al. 2020)	15.2/16.6	65.2/67.1	8.5/9.1	37.8/39.2	6.7/8.6	31.1/35.9
GPS-Net‡ (Lin et al. 2020)	29.2/31.4	55.2/57.6	15.9/16.9	36.4/37.5	8.1/9.6	28.4/33.4
NICE-Motif(Li et al. 2022a)	29.9/32.3	55.1/57.2	16.6/17.9	33.1/34.0	**12.2/14.4**	27.8/31.8
PPDL(Li et al. 2022b)	32.2/33.3	47.2/47.6	17.5/18.2	28.4/29.3	11.4/13.5	21.2/23.9
BGNN‡ (Li et al. 2021)	30.4/32.9	59.2/61.3	14.3/16.5	37.4/38.5	10.7/12.6	31.0/35.8
BGNN*‡ (Li et al. 2021)	29.2/31.7	57.8/60.0	14.6/16.0	36.9/38.1	10.9/13.1	30.2/34.9
HetSGG‡	31.6/33.5	57.8/59.1	**17.2/18.7**	37.6/38.7	**12.2/14.4**	30.0/34.6
HetSGG$^{\ddagger}_{++}$	**32.3/34.5**	57.1/59.4	15.8/17.7	37.6/38.5	11.5/13.5	30.2/34.5
Improv.(%)	**10.6/8.8**	0.0/-1.0	**17.8/16.9**	1.9/1.6	**11.9/9.9**	0.0/-0.8

[Main Table]



[Improvement per class of HetSGG over BGNN]



[Results on the overall, head, body, tail predicates]

EXPERIMENT: OBJECT TYPES & ACCURACY OF TYPE INFERENCE

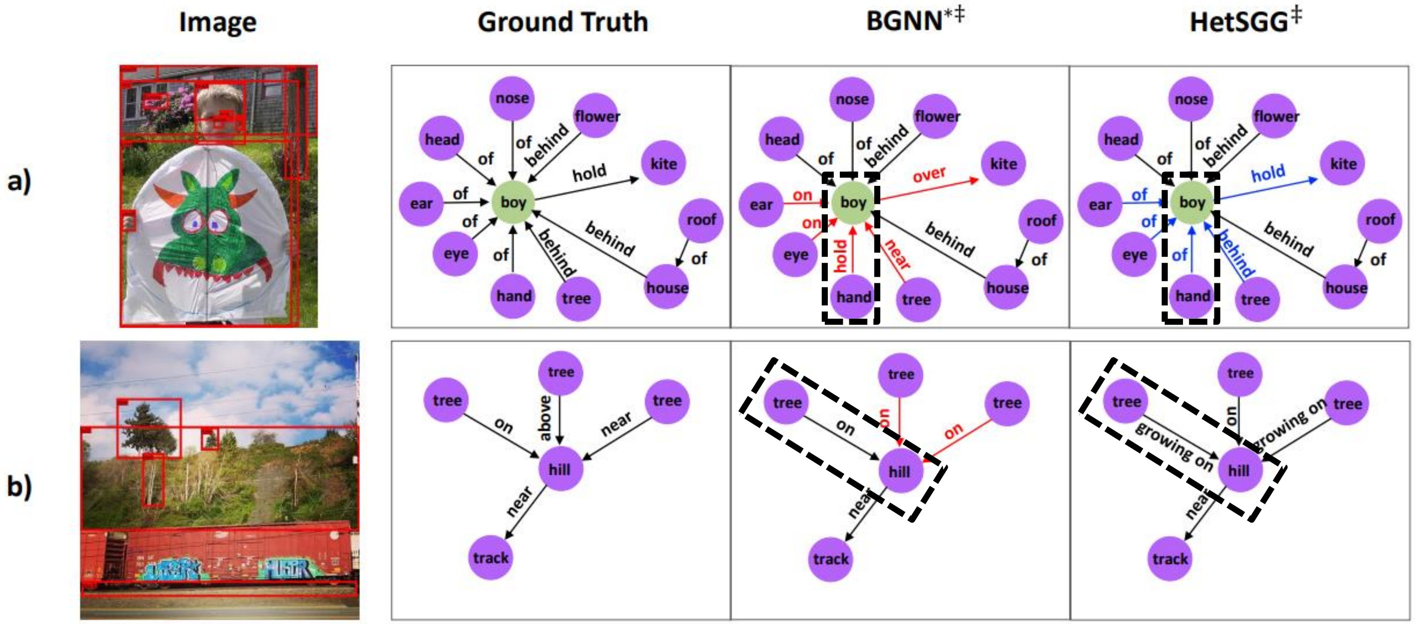
- **Analysis for object types and accuracy of object type prediction**
- 1) $\mathrm{HetSGG_{GT}}$ performs better on P,H,A,L than P,H,A object types
  - Add the **Landform** object type from Product, Human, and Animal types
  - The fine-grained heterogeneity information is helpful on scene graph
- 2) $\mathrm{HetSGG_{GT}}$ consistently outperforms HetSGG
  - Accurately inferring the object types is crucial
  - For this reason, HetSGG outperforms on P,H,A object types compared to P,H,A,L object types

Object Types	Model	SGCls		Type Inf. Acc.(%)
		mR@50/100	R@50/100	
P,H,A	$\mathrm{HetSGG}^{\ddagger}$	17.2 / 18.7	37.6 / 38.7	95.3
	$\mathrm{HetSGG}^{\ddagger}_{\mathrm{GT}}$	17.4 / 19.1	38.0 / 39.0	100
P,H,A,L	$\mathrm{HetSGG}^{\ddagger}$	15.9 / 18.2	37.5 / 38.4	90.9
	$\mathrm{HetSGG}^{\ddagger}_{\mathrm{GT}}$	18.2 / 19.4	39.4 / 40.5	100

[Object type and Accuracy of object type prediction]

EXPERIMENT: QUALITATIVE RESULTS

- a) BGNN predicts "hand hold boy", but HetSGG predicts "hand of boy"
  - HetSGG predicts the correct predicate by filtering the non-sense semantic relation, such as "hand hold boy"
- b) BGNN predicts "tree on hill", but HetSGG predicts the fine-grained predicate (i.e., growing on)
  - HetSGG alleviates the long-tailed predicate distribution, thus predicts the fine-grained predicate



[Qualitative Result]

Red predicate: Incorrect for BGNN
Blue predicate: Correct for HetSGG and Incorrect for BGNN

CONCLUSION

- In Summary,
  - As we verified, HetSGG is the first work, which shows that the semantic information captured through a heterogeneous graph is helpful for the scene graph generation.

- For , limitation of this study,
  - The object type assignment depends on the selection of object detectors.
    - Applying the state-of-art object detector further improve HetSGG !
  - Pre-defining all object types requires cost, and causes a new bias.
    - New framework that generates latent object types and assigns based on the image is necessary

- For additional experiments, please refer to paper.

- Code is available at https://github.com/KanghoonYoon/hetsgg-torch

Paper Code

THANK YOU