Adaptive Self-Training Framework for Fine-grained Scene Graph Generation
-ICLR 2024 Poster-

Kibum Kim*, Kanghoon Yoon*, Yeonjun In, Jinyoung Moon, Donghyun Kim, Chanyoung Park†

Presenter: Kibum Kim

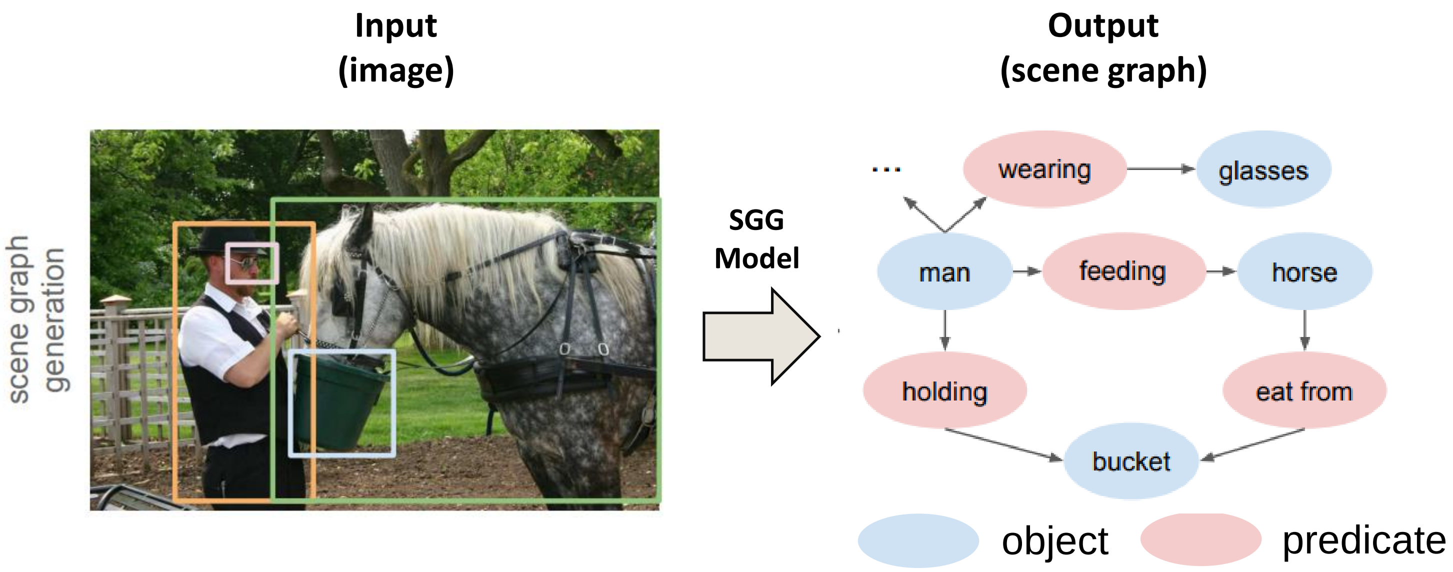
Ph.D Student
Department of Industrial & Systems Engineering
KAIST

*Equal Contribution / †Corresponding Author

CONTENT

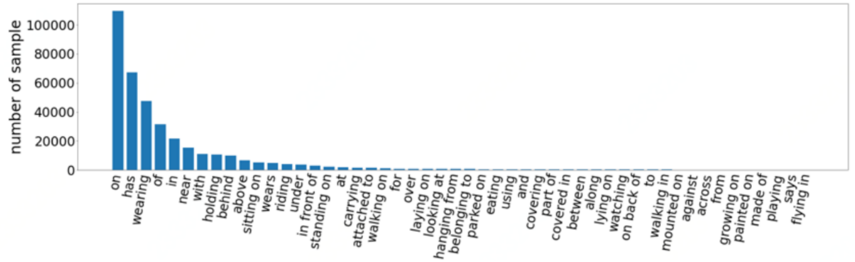
WHAT IS SCENE GRAPH GENERATION (SGG) ?

- SGG aims to represent observable knowledges in an image in the form of a graph
- The knowledge includes 1) object information and 2) their relation information, which is mapped to a scene graph
  - E.g., Object information: {*man, horse, glasses, bucket*}
  - E.g., Relationship information between objects: {*feeding, wearing, …, holding, eat from*}

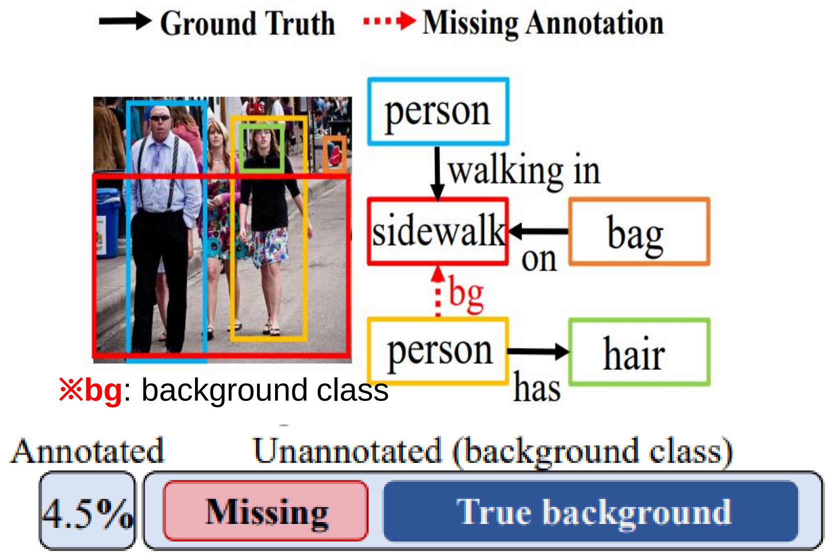


MOTIVATION: INHERENT PROBLEMS IN SGG DATASETS

- Inherent Problems in SGG datasets (e.g., Visual Genome [1])
  - **1. Long-tailed Predicate Distribution**
    - It leads to biased predictions towards head classes which are uninformative
  - **2. Missing Annotations of Predicate**
    - In right figure*, walking in* is annotated for one instance of *person* → *sidewalk*, but not for the other instance.
    - Among overall relationships, only 4.5% relationship is annotated = 95.5% relationship is not annotated



1. Long-tailed Predicate Distribution

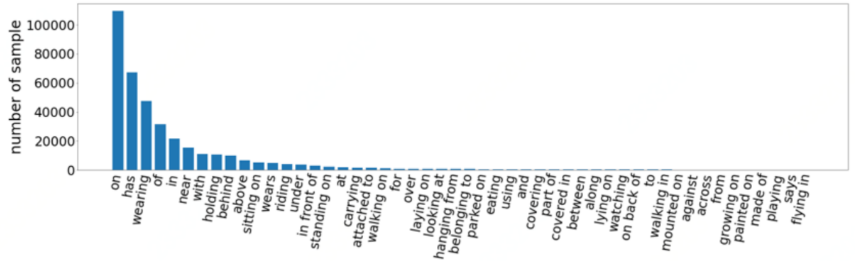


2. Missing Annotations of Predicate

[1] Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. Krishna et al. IJCV'17

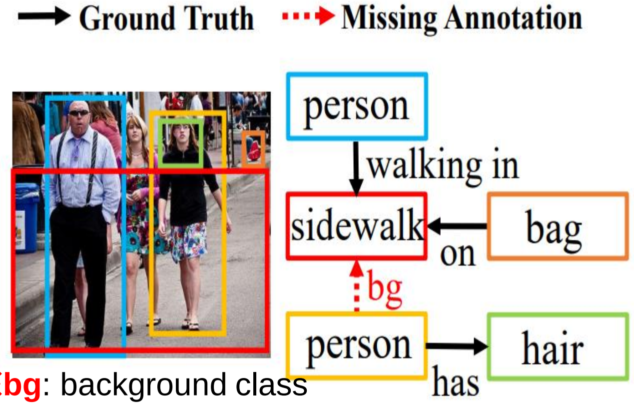
MOTIVATION: INHERENT PROBLEMS IN SGG DATASETS

- Inherent Problems in SGG datasets (e.g., Visual Genome [1])
  - **1. Long-tailed Predicate Distribution**
    - It leads to biased predictions towards head classes which are uninformative



1. Long-tailed Predicate Distribution

  - **2. Missing Annotations of Predicate**
    - In right figure, walking in is annotated for one instance of person → sidewalk, but not for the other instance.
    - Among overall relationships, only 4.5% relationship is annotated = 95.5% relationship is not annotated



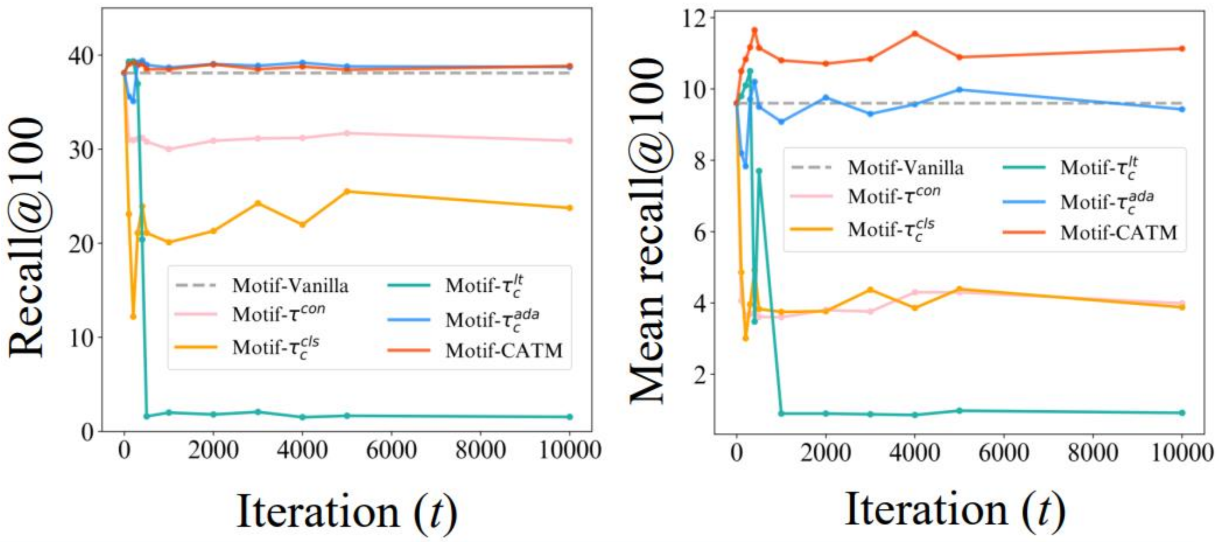
※bg: background class

We aim to assign pseudo-labels to missing annotations to address the long-tailed problem via Self-Training Framework

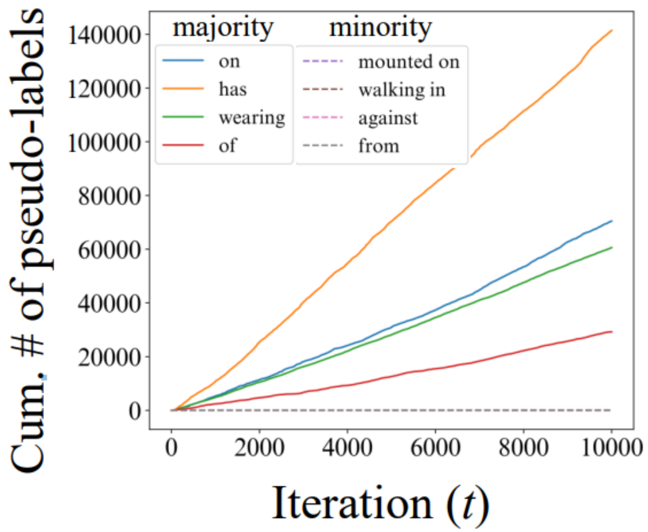
[1] Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. Krishna et al. IJCV'17

CHALLENGES OF APPLYING SELF-TRAINING FRAMEWORK FOR SGG

- **It is challenging to apply existing self-training framework from image classification to SGG task.**
  - **1. Extreme Long-tailedness:** Biased SGG models are likely to assign pseudo-labels of head classes.
  - **2. Semantic Ambiguity:** To assign pseudo-labels through the model's prediction probability, it is necessary to recognize "**Confident Samples**". Semantic Ambiguity makes it difficult to define confident samples.
    - E.g., in image classification task, the samples above 0.95 probability are assigned with pseudo-labels, but it is difficult on SGG



Failure of adapting existing self-training framework to SGG task



Pseudo-labels biased towards head class

METHOD: ADAPTIVE SELF-TRAINING FRAMEWORK FOR SGG (1/3)

- **Goal: We find suitable thresholds for each predicate class to apply self-training to SGG task.**
  - **(1) Class-specific Adaptive Thresholding – Increase threshold for each class**
    - Through Exponential Moving Average (EMA), we set thresholds based on the overall prediction probability.

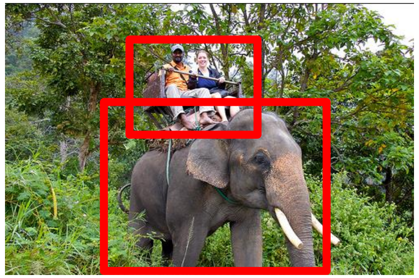
Momentum of EMA

Confidence average of predicate *c*

$$\tau_c^t = \left(1 - \lambda^{inc}\right) \cdot \tau_c^{t-1} + \lambda^{inc} \cdot \mathbb{E}_{i \in P_c^U}[\hat{q}_i]$$

Threshold of predicate *c* at *t* iteration

τ_{riding}^{t-1}



Confidence 0.3 (> 0.2)

Confidence 0.4 (> 0.2)

$$\tau_{riding}^{t} = 0.25 \approx (1 - 0.3) \cdot 0.2 + 0.3 \cdot \boxed{0.35}$$

τ_{riding}^{t-1} = 0.2 Increase τ_{riding}^{t} = 0.25

METHOD: ADAPTIVE SELF-TRAINING FRAMEWORK FOR SGG (2/3)

- **Goal: We find suitable thresholds for each predicate class to apply self-training to SGG task.**
  - **(2) Class-specific Momentum (Increase): It rapidly increase the threshold for head classes, while slowly increasing the threshold for tail classes.**

    → It enables pseudo-labeling primarily for tail predicate classes.

$$\underbrace{\tau_c^t}_{\text{Threshold of predicate } c \text{ at } t \text{ iteration}} = (1 - \overbrace{\lambda^{inc}}^{\text{Momentum of EMA}}) \cdot \tau_c^{t-1} + \lambda^{inc} \cdot \overbrace{\mathbb{E}_{i \in P_c^U}[\hat{q}_i]}^{\text{Confidence average of predicate } c}$$

$$\lambda_c^{inc} = \left(\frac{N_c}{N_1}\right)^{\alpha^{inc}}$$

Head: $\lambda_1^{inc} = \left(\frac{N_1}{N_1}\right)^{\alpha^{inc}} = 1^{\alpha^{inc}}$

Tail: $\lambda_3^{inc} = \left(\frac{N_3}{N_1}\right)^{\alpha^{inc}} = 0.2^{\alpha^{inc}}$

- N_c: # instances of *c* classes
- E.g., $N_1(Head)$=50, N_2=40, $N_3(Tail)$=10

METHOD: ADAPTIVE SELF-TRAINING FRAMEWORK FOR SGG (1/3)

- **Goal: We find suitable thresholds for each predicate class to apply self-training to SGG task.**
  - **(1) Class-specific Adaptive Thresholding – Decrease threshold for each class**
    - **If predictions of predicate c are made at the current step t while no pseudo-label is assigned with those instances, we decrease thresholds for predicate c**

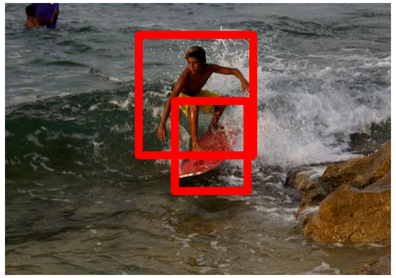
Momentum of EMA

Confidence average of predicate *c*

$$\tau_c^t = \left(1 - \lambda^{dec}\right) \cdot \tau_c^{t-1} + \lambda^{dec} \cdot \mathbb{E}_{i \in P_c^U}[\hat{q}_i]$$

Threshold of predicate *c* at *t* iteration

$\boldsymbol{\tau_{riding}^{t-1}}$



Confidence 0.1 (< 0.25)

Confidence 0.15 (< 0.25)

$$\boldsymbol{\tau_{riding}^{t} = 0.19 \approx (1 - 0.5) \cdot 0.25 + 0.5 \cdot \boxed{0.125}}$$

$\boldsymbol{\tau_{riding}^{t-1}}$ **= 0.25** Increase $\boldsymbol{\tau_{riding}^{t}}$ **= 0.19**

METHOD: ADAPTIVE SELF-TRAINING FRAMEWORK FOR SGG (3/3)

- **Goal: We find suitable thresholds for each predicate class to apply self-training to SGG task.**
  - **(2) Class-specific Momentum (Decrease): It slowly decreases the threshold for head classes, while rapidly decreasing the threshold for tail classes.**

Momentum of EMA

Confidence average of predicate *c*

$$\tau_c^t = (1 - \lambda^{dec}) \cdot \tau_c^{t-1} + \lambda^{dec} \cdot \mathbb{E}_{i \in P_c^U}[\hat{q}_i]$$

Threshold of predicate *c* at *t* iteration

$$\lambda_c^{dec} = \left(\frac{N_c}{N_1}\right)^{\alpha^{dec}}$$

Head: $\boldsymbol{\lambda_1^{dec} = \left(\frac{N_1}{N_1}\right)^{\alpha^{dec}} = 0.2^{\alpha^{dec}}}$

Tail: $\boldsymbol{\lambda_3^{dec} = \left(\frac{N_3}{N_1}\right)^{dec} = 1^{\alpha^{dec}}}$

- N_c: # instances of *c* classes
- E.g., $N_1(Head)$=50, N_2=40, $N_3(Tail)$=10

EXPERIMENT WITHIN VISUAL GENOME DATASET

- **Metric**
  - R@K: Performance of Head classes ↑ → R@K ↑
  - mR@K: Performance of Tail classes ↑ → mR@K ↑
  - F@K: Harmonic average of R@K and mR@K

	Method	PredCls R@50 / 100	PredCls mR@50 / 100	PredCls F@50 / 100	SGCls R@50 / 100	SGCls mR@50 / 100	SGCls F@50 / 100	SGDet R@50 / 100	SGDet mR@50 / 100	SGDet F@50 / 100
Specific	DT2-ACBS [4]	23.3 / 25.6	35.9 / 39.7	28.3 / 31.1	16.2 / 17.6	24.8 / 27.5	19.6 / 21.5	15.0 / 16.3	22.0 / 24.0	17.8 / 19.4
	PCPL [6]	50.8 / 52.6	35.2 / 37.8	41.6 / 44.0	27.6 / 28.4	18.6 / 19.6	22.2 / 23.2	14.6 / 18.6	9.5 / 11.7	11.5 / 14.4
	KERN [34]	65.8 / 67.6	17.7 / 19.2	27.9 / 29.9	36.7 / 37.4	9.4 / 10.0	15.0 / 15.8	27.1 / 29.8	6.4 / 7.3	10.4 / 11.7
	GBNet [33]	66.6 / 68.2	22.1 / 24.0	33.2 / 35.5	37.3 / 38.0	12.7 / 13.4	18.9 / 19.8	26.3 / 29.9	7.1 / 8.5	11.2 / 13.2
Model-Agnostic	Motif [22]	**65.3 / 67.1**	17.8 / 19.2	28.0 / 29.9	**36.9 / 38.1**	9.0 / 9.6	14.5 / 15.3	**31.9 / 36.4**	6.4 / 7.6	10.7 / 12.6
	+ST-SGG	63.4 / 65.4	22.4 / 24.1	33.1 / 35.2	36.8 / 37.8	12.1 / 12.8	18.2 / 19.1	29.7 / 34.8	8.5 / 10.1	13.2 / 15.7
	+Resam. [3]	62.3 / 64.3	26.1 / 28.5	36.8 / 39.5	36.1 / 37.0	13.7 / 14.7	19.9 / 21.0	30.4 / 34.8	10.5 / 12.3	15.6 / 18.2
	+Resam.+ST-SGG	53.9 / 57.7	28.1 / 31.5	36.9 / 40.8	33.4 / 34.9	16.9 / 18.0	22.4 / 23.8	26.7 / 30.7	11.6 / 14.2	16.2 / 19.4
	+TDE [8]	46.2 / 51.4	25.5 / 29.1	32.9 / 37.2	27.7 / 29.9	13.1 / 14.9	17.8 / 19.9	16.9 / 20.3	8.2 / 9.8	11.0 / 13.2
	+DLFE [10]	52.5 / 54.2	26.9 / 28.8	35.6 / 37.6	32.3 / 33.1	15.2 / 15.9	20.7 / 21.5	25.4 / 29.4	11.7 / 13.8	16.0 / 18.8
	+NICE [36]	55.1 / 57.2	29.9 / 32.3	38.8 / 41.3	33.1 / 34.0	16.6 / 17.9	22.1 / 23.5	27.8 / 31.8	12.2 / 14.4	17.0 / 19.8
	+IE-Trans [42]	54.7 / 56.7	30.9 / 33.6	39.5 / 42.2	32.5 / 33.4	16.8 / 17.9	22.2 / 23.3	26.4 / 30.6	12.4 / 14.9	16.9 / 20.0
	+I-Trans [42]	55.2 / 57.1	29.1 / 31.9	38.1 / 40.9	32.5 / 33.4	15.7 / 16.9	21.2 / 22.4	27.0 / 31.3	11.4 / 14.0	16.0 / 19.3
	+I-Trans+ST-SGG	50.5 / 52.8	**32.5 / 35.1**	**41.7 / 42.5**	31.2 / 32.1	**18.0 / 19.3**	**22.8 / 24.1**	25.7 / 29.8	**12.9 / 15.8**	**17.2 / 20.7**
	VCTree [23]	**65.5 / 67.2**	17.2 / 18.6	27.3 / 29.1	**38.1** / 38.8	9.6 / 10.2	15.3 / 16.2	**31.4 / 35.7**	7.3 / 8.6	11.9 / 13.9
	+ST-SGG	64.2 / 66.2	21.5 / 22.9	32.2 / 34.0	37.5 / 38.4	12.0 / 12.5	18.2 / 18.9	30.4 / 34.7	8.7 / 10.1	13.5 / 15.6
	+Resam. [3]	61.2 / 63.5	27.2 / 29.2	37.7 / 40.0	35.7 / 36.5	13.8 / 14.4	19.9 / 20.7	29.7 / 33.9	10.2 / 11.8	15.2 / 17.5
	+Resam.+ST-SGG	54.0 / 57.0	32.2 / 34.6	**40.3 / 43.0**	32.2 / 33.4	16.9 / 18.3	22.2 / 23.6	24.6 / 29.6	12.3 / 14.8	**16.4 / 19.7**
	+TDE [8]	47.2 / 51.6	25.4 / 28.7	33.0 / 36.9	25.4 / 27.9	12.2 / 14.0	16.5 / 18.6	19.4 / 23.2	9.3 / 11.1	12.6 / 15.0
	+DLFE [10]	51.8 / 53.5	25.3 / 27.1	34.0 / 36.0	33.5 / 34.6	18.9 / 20.0	24.2 / 25.3	22.7 / 26.3	11.8 / 13.8	15.5 / 18.1
	+NICE [36]	55.0 / 56.9	30.7 / 33.0	39.4 / 41.8	37.8 / **39.0**	19.9 / 21.3	26.1 / 27.6	27.0 / 30.8	11.9 / 14.1	16.5 / 19.3
	+IE-Trans [42]	53.0 / 55.0	30.3 / 33.9	38.6 / 41.9	32.9 / 33.8	16.5 / 18.1	22.0 / 23.6	25.4 / 29.3	11.5 / 14.0	15.8 / 18.9
	+I-Trans [42]	54.0 / 55.9	30.2 / 33.1	38.7 / 41.6	37.2 / 38.3	19.0 / 20.6	25.1 / 26.8	25.5 / 29.4	11.2 / 13.7	15.6 / 18.7
	+I-Trans+ST-SGG	52.5 / 54.3	**32.7 / 35.6**	**40.3 / 43.0**	36.3 / 37.3	**21.0 / 22.4**	**26.6 / 27.9**	20.7 / 24.9	**12.6 / 15.1**	15.7 / 18.8

Performance Comparison with baselines

- ST-SGG is applicable to other model
  - Performance increase on Motif+ST-SGG / VCTree + ST-SGG
- Combining ST-SGG with debiasing method outperforms SOTA baselines
  - ST-SGG is adopted to Resampling or I-Trans method

For more experiments, please refer to main paper

CONCLUSION

- ST-SGG aims to address long-tailed problem by annotating pseudo-labels on unannotated relationships via self-training framework

- We identify challenges of applying existing self-training framework to SGG task
  - It stems from 1) extreme long-tailed problem and 2) semantic ambiguity

- To this end, we propose novel self-training framework for SGG, which consists of *class-specific adaptive thresholding* with *class-specific momentum*

- Our proposed framework outperforms state-of-the-arts baseline in terms of F@K and mR@K

THANK YOU

- Paper: https://openreview.net/pdf?id=WipsLtH77t
- Code: https://github.com/rlqja1107/torch-ST-SGG

Paper

Code