

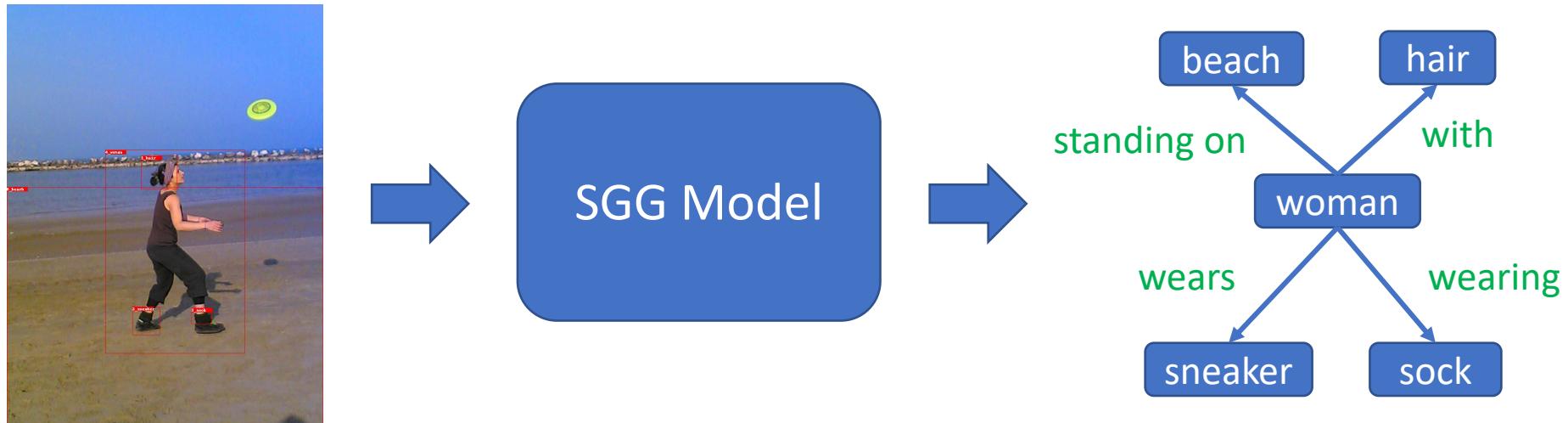
Recovering the Unbiased Scene Graphs from the Biased Ones

Meng-Jiun Chiou¹, Henghui Ding², Hanshu Yan¹, Changhu Wang²,
Roger Zimmermann¹ and Jiashi Feng¹

¹National University of Singapore ²ByteDance AI Lab

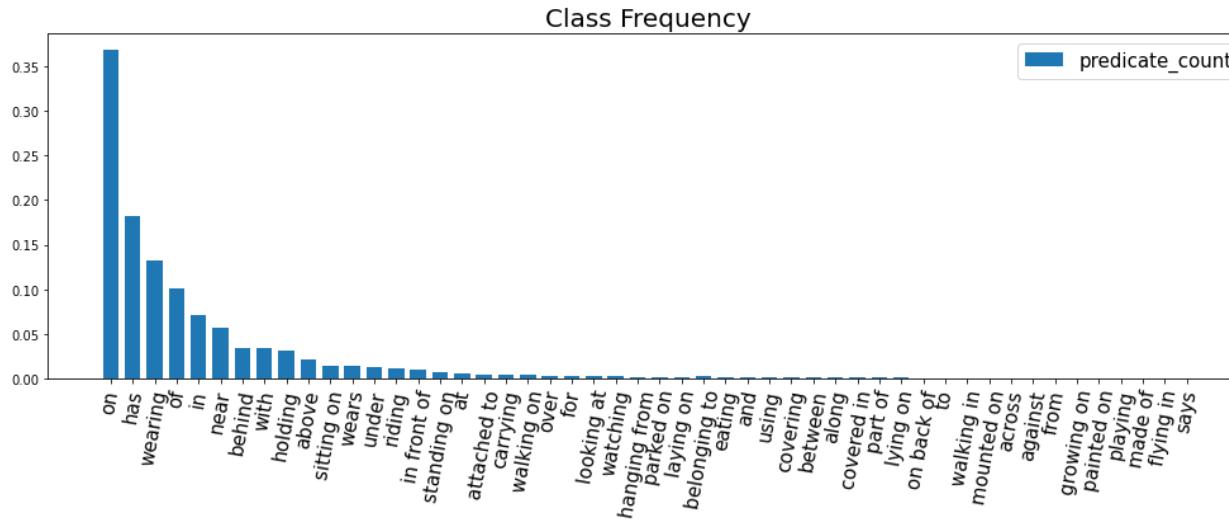
Scene Graph¹ Generation (SGG)

- Given images, SGG aims to predict visual relationships among salient objects

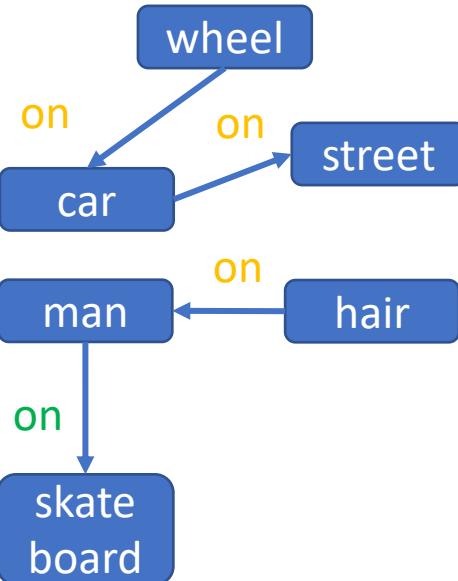
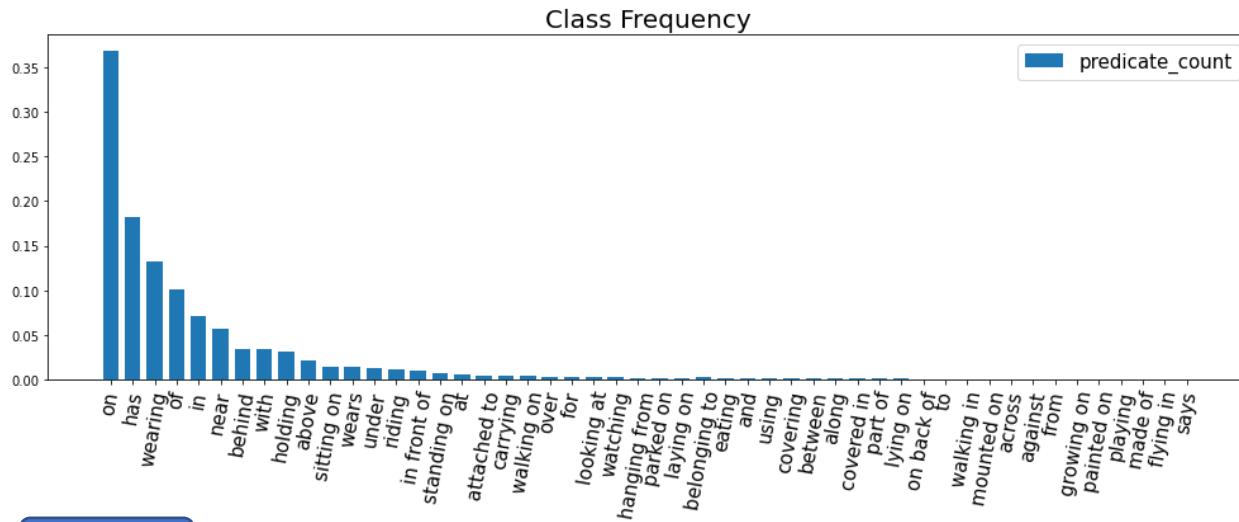


¹Johnson et al. “image retrieval using scene graphs.” In CVPR 2018

Long Tail Problem in SGG with the VG dataset



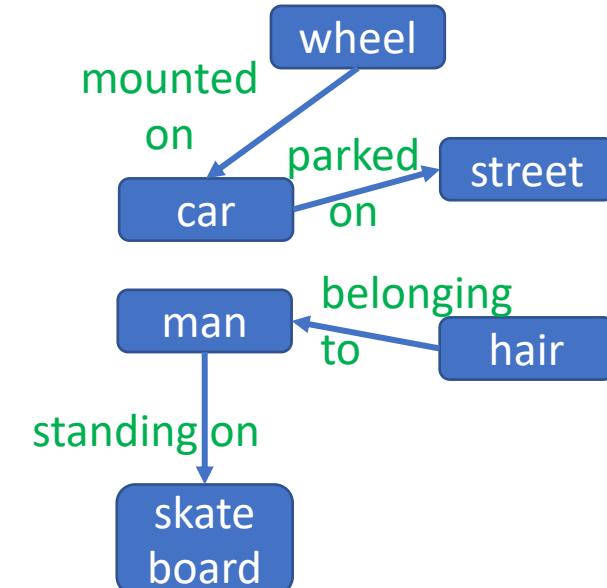
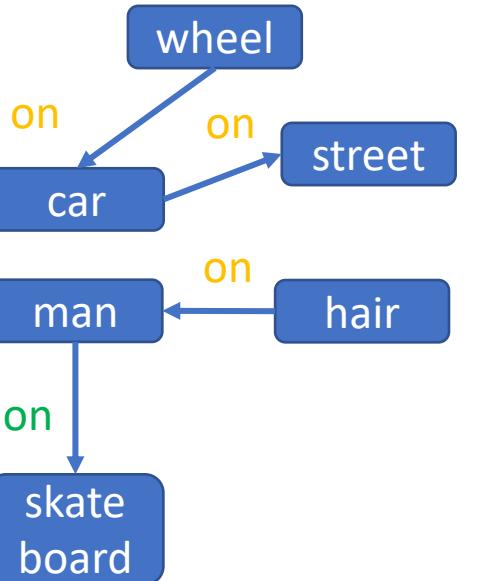
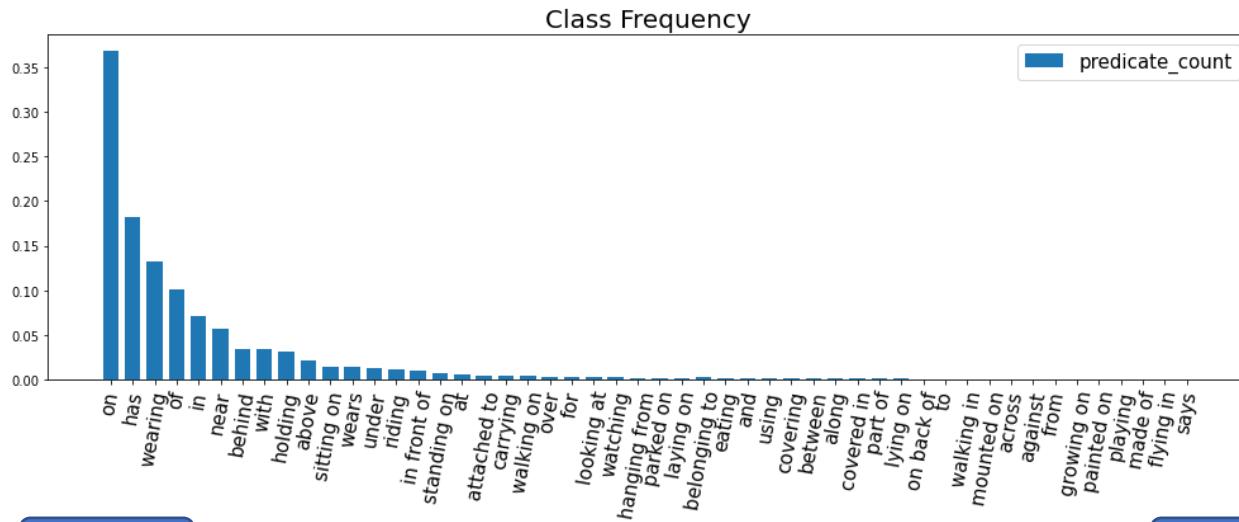
Long Tail Problem in SGG with the VG dataset



What we get with
a biased model:



Long Tail Problem in SGG with the VG dataset



Not All Long Tails are Equal

- Unlike long tails in other vision tasks like image classification, the long tail in SGG is significantly affected by **the imbalance in missing labels**

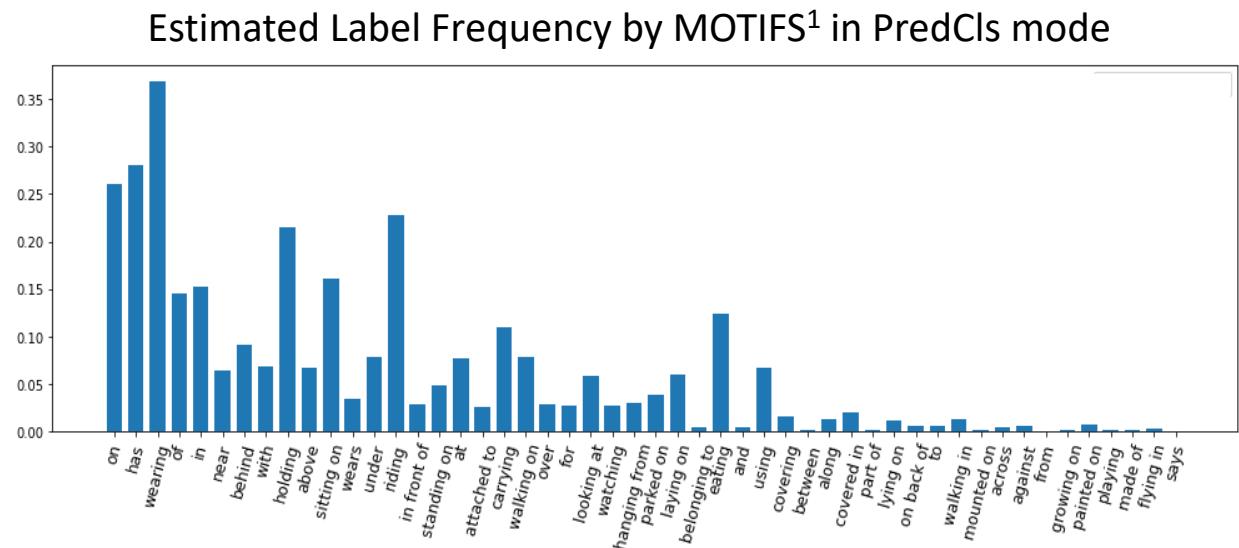
Not All Long Tails are Equal

- Unlike long tails in other vision tasks like image classification, the long tail in SGG is significantly affected by **the imbalance in missing labels**
- **Missing Labels** due to the cubic number of predicates
- For each image:
 - N objects, K predicate classes
 - $KN(N - 1)$ Possibilities
- **Missing label bias** causes the predicted probabilities to be under-estimated



Not All Long Tails are Equal

- Unlike long tails in other vision tasks like image classification, the long tail in SGG is significantly affected by **the imbalance in missing labels**
- **Label frequency:** the per-class fraction of labeled, positive examples in all the examples
- **The imbalance in missing labels, or reporting bias:** easier predicates (e.g., *on*) get annotated more than harder ones (e.g., *parked on*)
- The predicted probabilities of the hard ones are thus **under-estimated more** than the easy ones, causing the long tail in the VG dataset



¹Zellers et al. “Neural Motifs: Scene Graph Parsing with Global Context.” CVPR’18

Learning from Positive and Unlabeled Data¹

- Examples in VG dataset are a set of triplets $\{(x, y, s)\}$.
 - x be an example (candidate object pair)
 - $y \in \{0, \dots, K\}$ be its true class, where K is the number of predicate classes.
 - $s \in \{0, \dots, K\}$ is a label and $s = 0$ if x is unlabeled.
 - When $s = r \Rightarrow y = r$; When $s = 0$, y can be 0 (background) or any natural number.

¹Elkan et al. “Learning classifiers from only positive and unlabeled data”. In SIGKDD 2008.

Learning from Positive and Unlabeled Data¹

- Examples in VG dataset are a set of triplets $\{(x, y, s)\}$.
 - x be an example (candidate object pair)
 - $y \in \{0, \dots, K\}$ be its true class, where K is the number of predicate classes.
 - $s \in \{0, \dots, K\}$ is a label and $s = 0$ if x is unlabeled.
 - When $s = r \Rightarrow y = r$; When $s = 0$, y can be 0 (background) or any natural number.
- Biased probability of a non-background class r ($r \neq 0$):

$$\begin{aligned} p(s = r|x) &= p(y = r, s = r|x) \\ &= p(y = r|x)p(s = r|y = r, x) \end{aligned}$$

¹Elkan et al. “Learning classifiers from only positive and unlabeled data”. In SIGKDD 2008.

Learning from Positive and Unlabeled Data¹

- Examples in VG dataset are a set of triplets $\{(x, y, s)\}$.
 - x be an example (candidate object pair)
 - $y \in \{0, \dots, K\}$ be its true class, where K is the number of predicate classes.
 - $s \in \{0, \dots, K\}$ is a label and $s = 0$ if x is unlabeled.
 - When $s = r \Rightarrow y = r$; When $s = 0$, y can be 0 (background) or any natural number.
- Biased probability of a non-background class r ($r \neq 0$):

$$\begin{aligned} p(s = r|x) &= p(y = r, s = r|x) \\ &= p(y = r|x)p(s = r|y = r, x) \end{aligned}$$

- Unbiased probability:

$$p(y = r|x) = \frac{p(s=r|x)}{p(s=r|y=r,x)}$$

Propensity score

¹Elkan et al. "Learning classifiers from only positive and unlabeled data". In SIGKDD 2008.

Learning from Positive and Unlabeled Data¹

- Examples in VG dataset are a set of triplets $\{(x, y, s)\}$.
 - x be an example (candidate object pair)
 - $y \in \{0, \dots, K\}$ be its true class, where K is the number of predicate classes.
 - $s \in \{0, \dots, K\}$ is a label and $s = 0$ if x is unlabeled.
 - When $s = r \Rightarrow y = r$; When $s = 0$, y can be 0 (background) or any natural number.
- Biased probability of a non-background class r ($r \neq 0$):

$$\begin{aligned} p(s = r|x) &= p(y = r, s = r|x) \\ &= p(y = r|x)p(s = r|y = r, x) \end{aligned}$$

- Unbiased probability:

$$p(y = r|x) = \frac{p(s=r|x)}{p(s=r|y=r,x)} \approx \frac{p(s=r|x)}{p(s=r|y=r)}$$

Label frequency of class r

(by SCAR assumption)

¹Elkan et al. "Learning classifiers from only positive and unlabeled data". In SIGKDD 2008.

Estimator of Label Frequencies

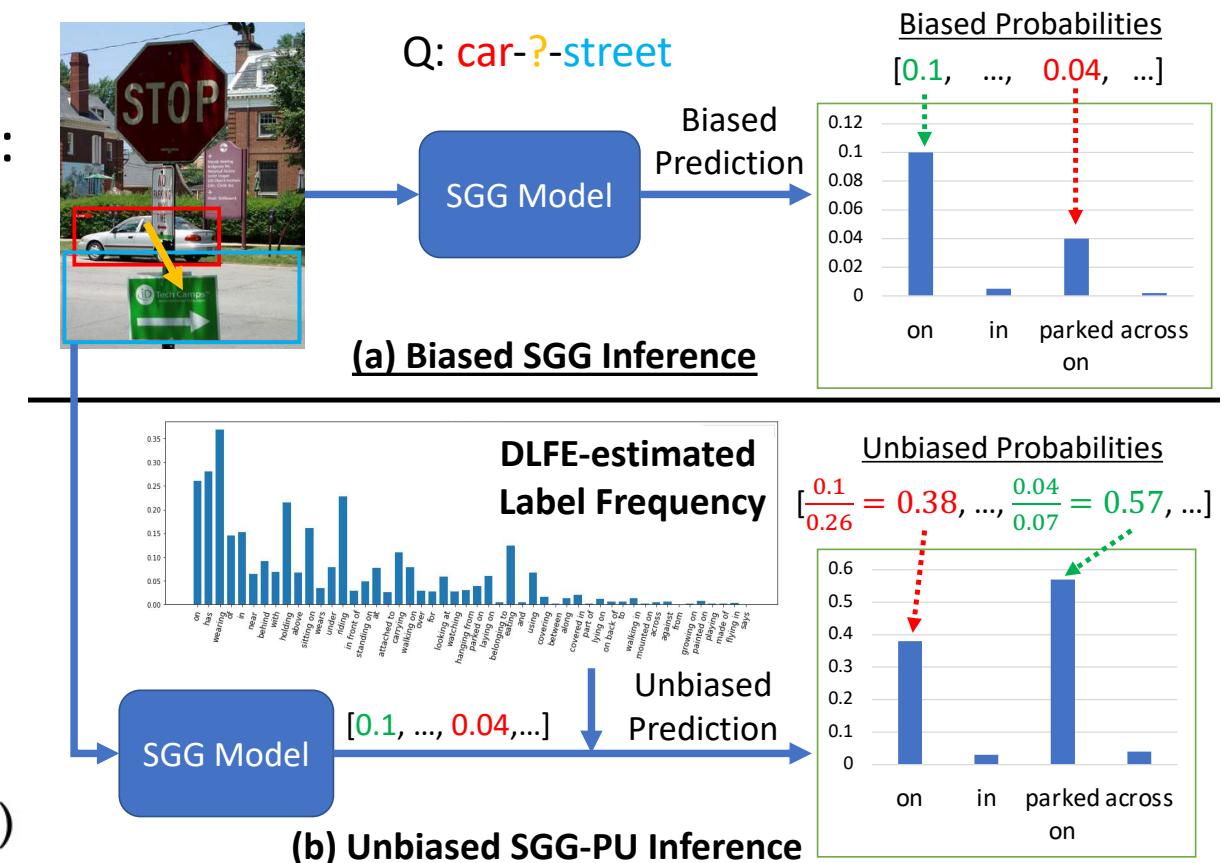
- Label frequency can be estimated by averaging over per-class biased probability predicted on a train/val set¹:

$$(\text{Train-Est}) \quad c_r = P(s = r | y = r)$$

$$\approx \frac{1}{N_r} \sum_{(x, y=r) \in D} p(s = r | x),$$

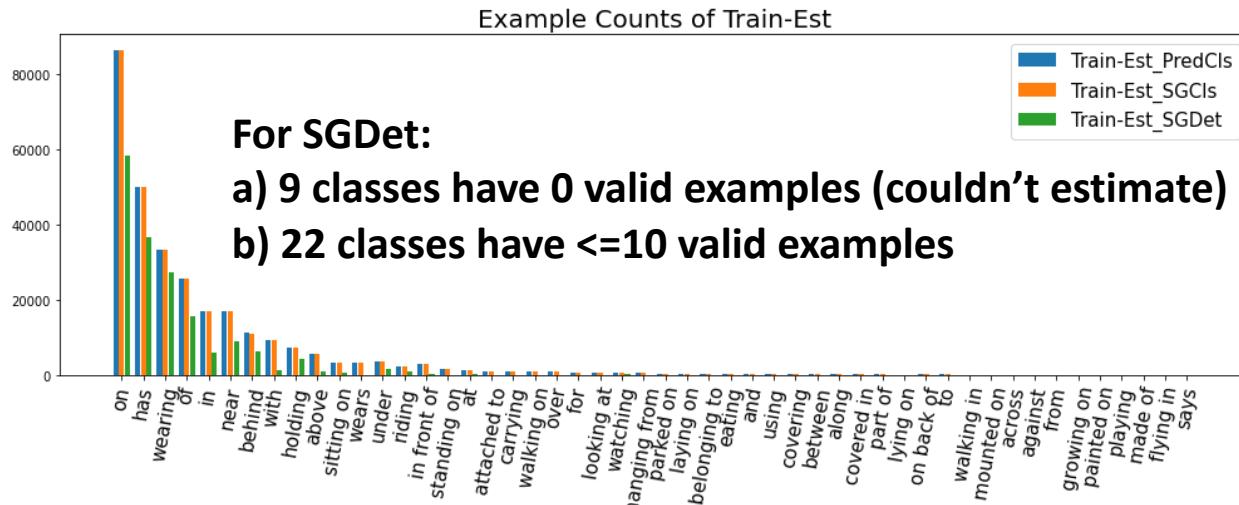
- Full Derivation

$$\begin{aligned} p(s = r | x) &= p(s = r | x, y = r)p(y = r | x) \\ &\quad + p(s = r | x, y \neq r)p(y \neq r | x) \\ &= p(s = r | x, y = r) \times 1 + 0 \times 0 \quad (\text{since } y=r) \\ &= p(s = r | y = r), \quad (\text{the SCAR assumption}) \end{aligned}$$



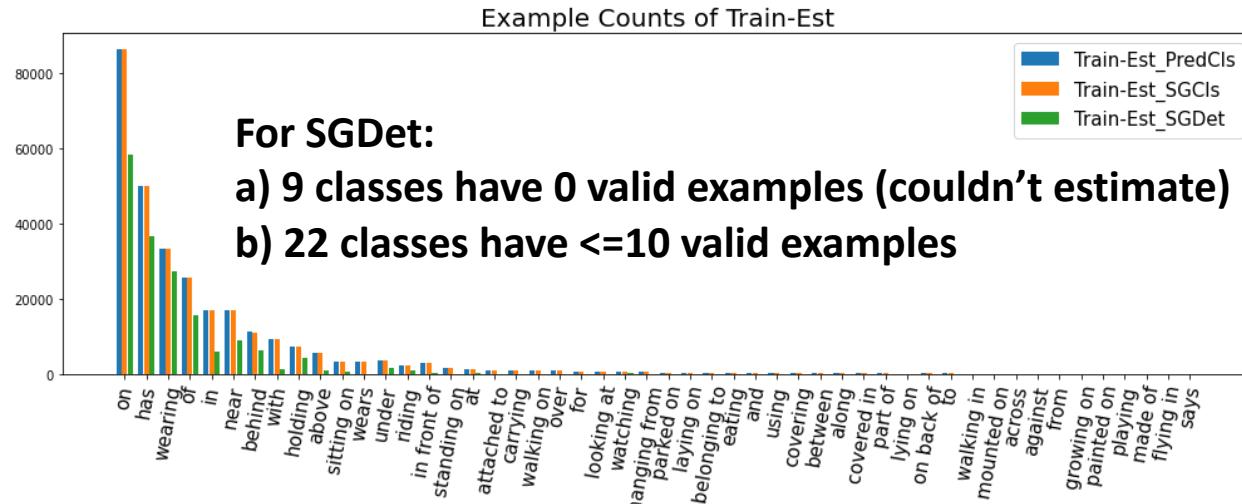
Towards a Better Label Frequency Estimator

- Traditional estimator of label frequency is not feasible in the harder SGG setting (i.e., SGDet), as there's no *valid* example for some classes
- More valid examples are also missing for tail classes: because they are concentrated in a much smaller number of images, not matching a bounding box could invalidate lots of examples.



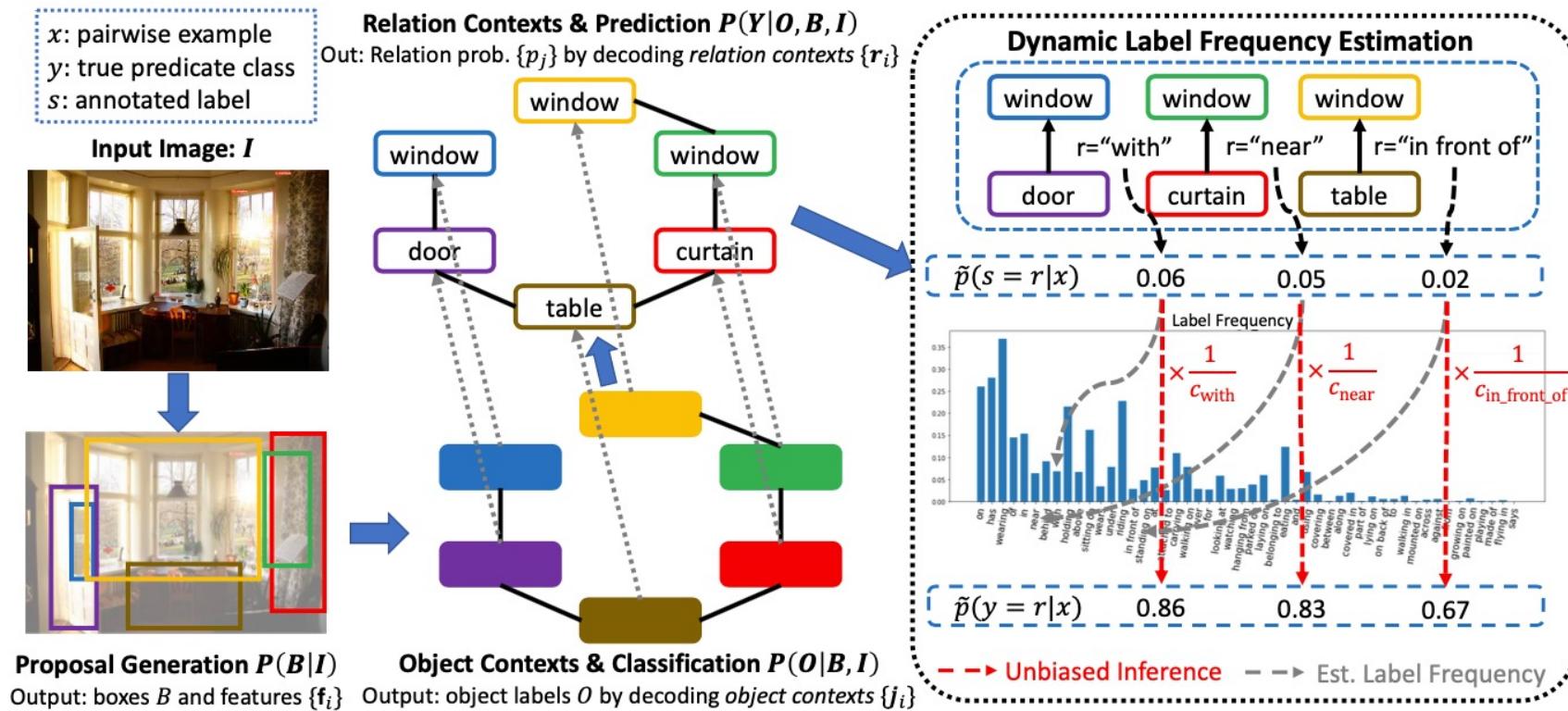
Towards a Better Label Frequency Estimator

- Traditional estimator of label frequency is not feasible in the harder SGG setting (i.e., SGDet), as there's no *valid* example for some classes
- More valid examples are also missing for tail classes: because they are concentrated in a much smaller number of images, not matching a bounding box could invalidate lots of examples.



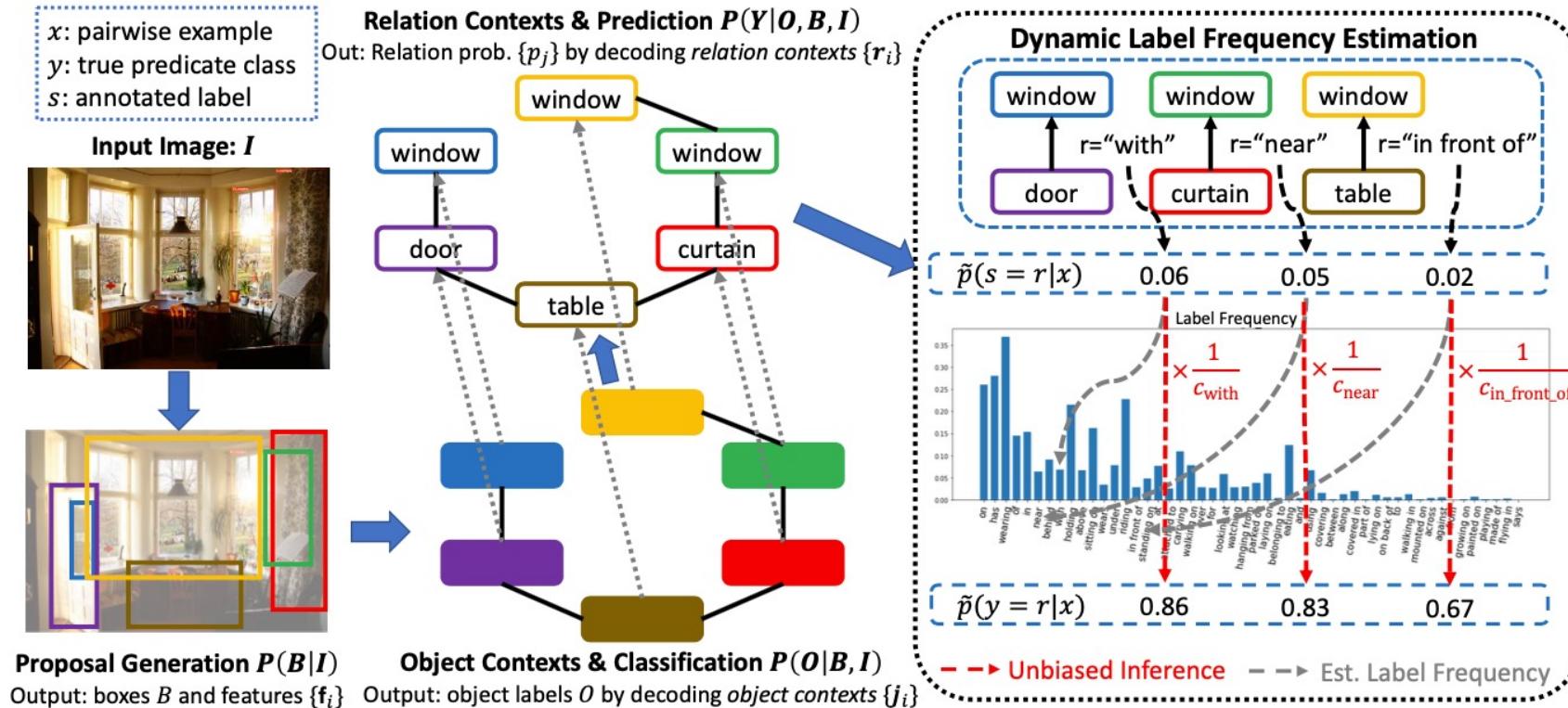
- We propose to take advantage of **data augmentation** such as **random flipping** and **averaging over multiple epochs** to introduce more samples

Dynamic Label Frequency Estimation (DLFE)



SGG Model from (Zellers et al. "Neural Motifs: Scene Graph Parsing with Global Context." CVPR'18)

Dynamic Label Frequency Estimation (DLFE)



Algorithm 1: DLFE during training-time

Input : Training dataset D^t and momentum α
Output: Biased model $g(\cdot)$ and estimated label frequency c

```

for each mini batch  $S = \{(x_i, y_i)\} \in D^t$ : do
    Forward model to obtain the biased probabilities  $g(x)$ ;
    // in-batch average of biased probabilities
    for each predicate class  $k \in \{1, \dots, K\}$ : do
         $S' \leftarrow \{(x_i, s_i) \in S | s_i = k\}$ ;
         $c'_k \leftarrow \frac{1}{|S'|} \sum_{(x_i, s_i) \in S'} g(s = s_i | x_i)$ ;
        // Update the exponential moving average
         $\tilde{c} \leftarrow \alpha \times c' + (1 - \alpha) \times \tilde{c}$ ;
    end
end
// Save for inference use
 $c \leftarrow \tilde{c}$ ;

```

Inference: $\tilde{p}(y|x) = \frac{1}{c} \odot \tilde{p}(s|x)$

SGG Model from (Zellers et al. "Neural Motifs: Scene Graph Parsing with Global Context." CVPR'18)

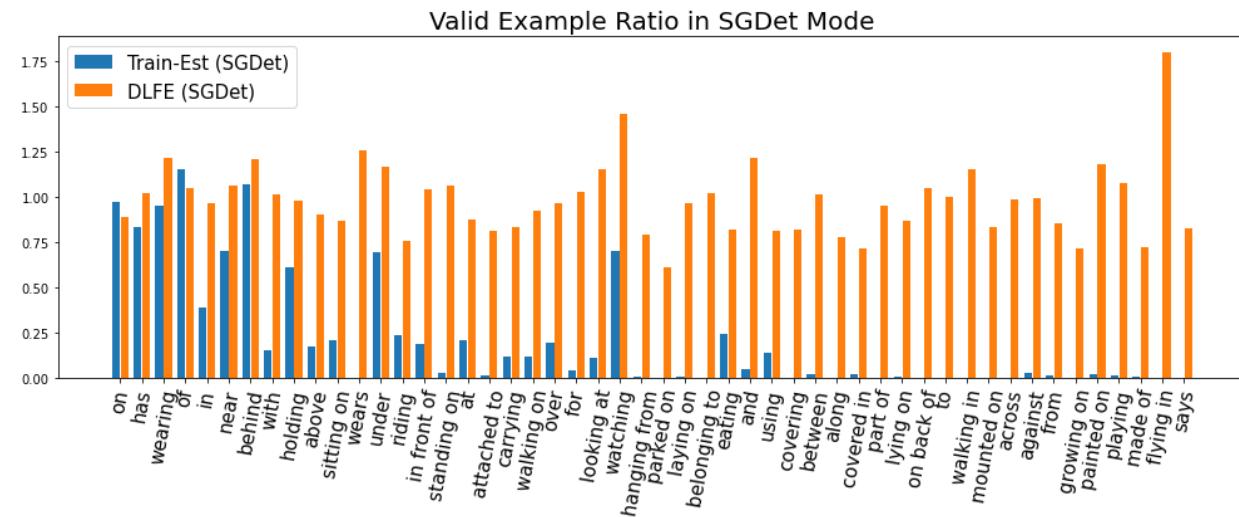
Experiments & Metrics

- Experimented with two popular SGG models: Motifs¹ & VCTree²
- The VG150 split of Visual Genome dataset
 - 62,723 images for training, 5,000 for validation and 26,446 for testing
 - 150 object categories, 50 predicate classes
- Recalls@K, where $K \in [20, 50, 100]$
 - Plain Recall ($R@K$)
 - Mean Recall ($mR@K$)
 - Non-graph constraint Recalls ($ng-R@K$, $ng-mR@K$)
 - Head (top-15 frequent predicates), tail (last-15), and middle (the others)
- Faster R-CNN with ResNext-101-FPN backbone as object detector

¹Zellers et al. “Neural Motifs: Scene Graph Parsing with Global Context.” CVPR’18

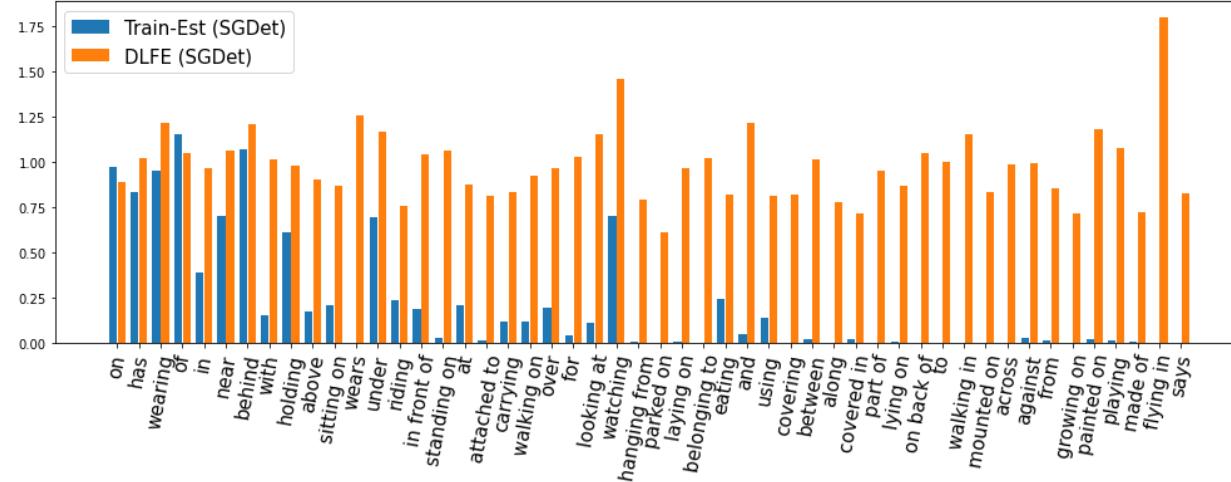
²Tang et al. “Learning to compose dynamic tree structures for visual contexts.” CVPR’19

Is DLFE more effective in estimating label frequency?

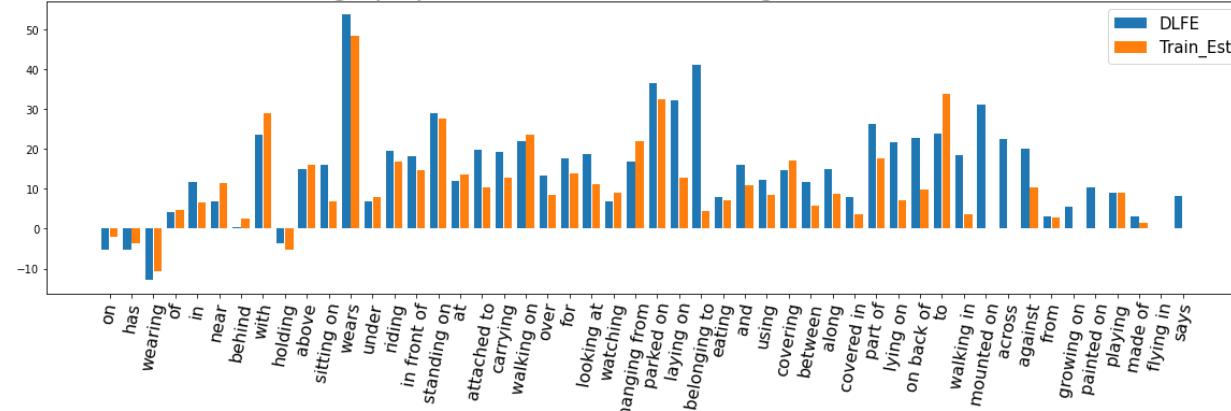


Is DLFE more effective in estimating label frequency?

Valid Example Ratio in SGDet Mode

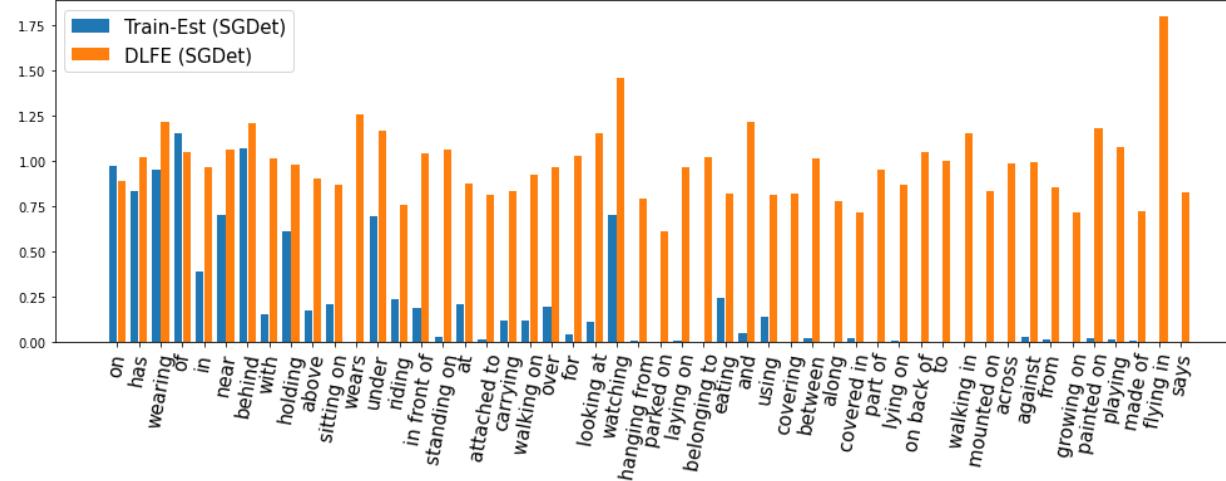


Predicate-wise non-graph per-class recall@100 Change w.r.t. the biased motif backbone

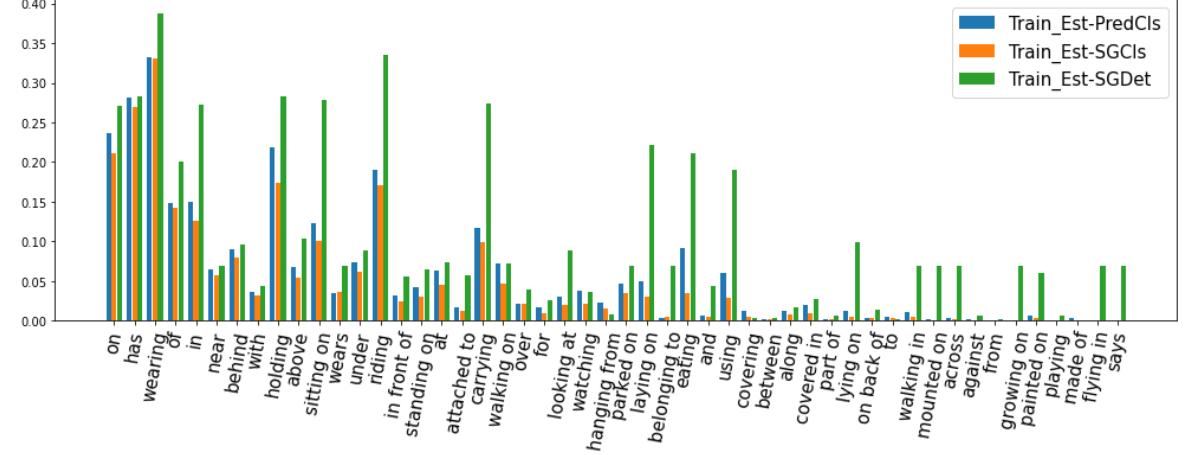


Is DLFE more effective in estimating label frequency?

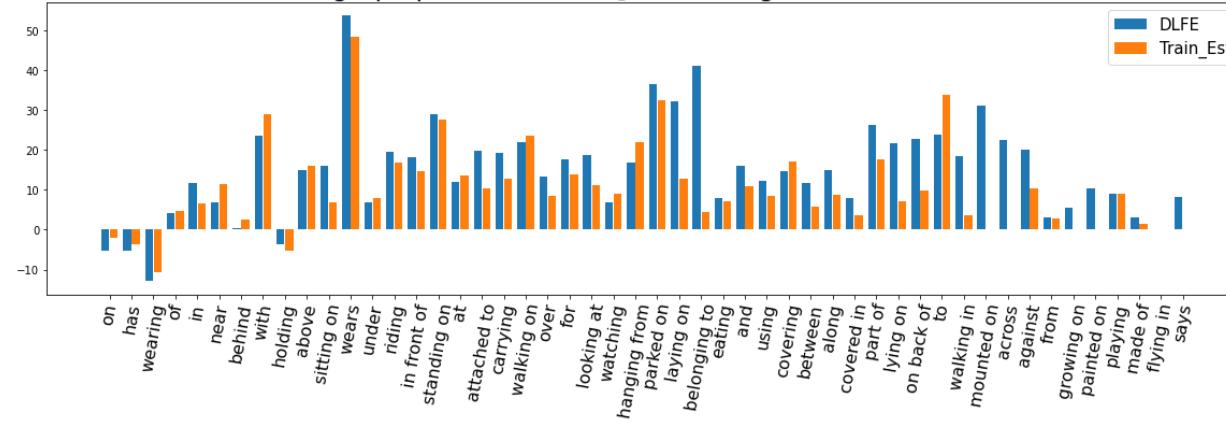
Valid Example Ratio in SGDet Mode



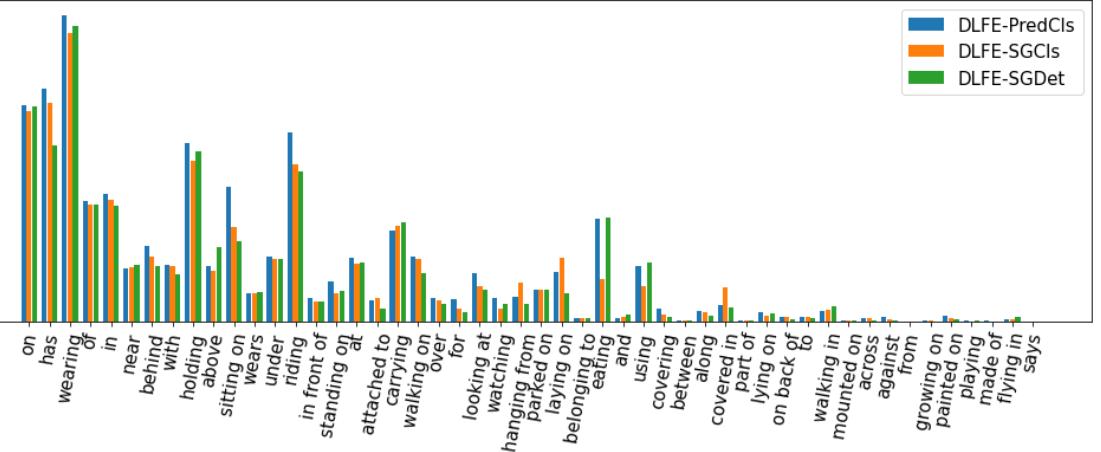
Train_Est-estimated Label Frequency



Predicate-wise non-graph per-class recall@100 Change w.r.t. the biased motif backbone



DLFE-estimated Label Frequency



Is DLFE more effective in estimating label frequency?

Model	Predicate Classification (PredCls)			Scene Graph Classification (SGCls)			Scene Graph Detection (SGDet)		
	ng-mR@20	ng-mR@50	ng-mR@100	ng-mR@20	ng-mR@50	ng-mR@100	ng-mR@20	ng-mR@50	ng-mR@100
MOTIFS [37, 53]	19.9	32.8	44.7	11.3	19.0	25.0	7.5	12.5	16.9
MOTIFS-Train-Est [12]	24.4	38.9	50.5	17.1	26.1	32.8	8.9	14.1	18.9
MOTIFS-DLFE	30.0	45.8	57.7	17.6	25.6	32.0	11.7	18.1	23.0
VCTree [37, 38]	21.4	35.6	47.8	12.4	19.1	25.5	7.5	12.5	16.7
VCTree-Train-Est [12]	25.0	39.1	52.4	21.0	32.2	39.4	8.1	13.0	17.1
VCTree-DLFE	29.1	44.6	56.8	21.6	31.4	38.8	11.7	17.5	22.5

Table 4: Comparison of non-graph constraint mean recalls (ng-mR@K) between Train-Est and our DLFE, in PredCls, SGCls and SGDet.

Compare DLFE to other debiasing methods

Model	Predicate Classification (PredCls)			Scene Graph Classification (SGCls)			Scene Graph Detection (SGDet)		
	ng-mR@20	ng-mR@50	ng-mR@100	ng-mR@20	ng-mR@50	ng-mR@100	ng-mR@20	ng-mR@50	ng-mR@100
KERN [4]	-	36.3	49.0	-	19.8	26.2	-	11.7	16.0
GB-Net- β^\diamond [39]	-	44.5	58.7	-	25.6	32.1	-	11.7	16.6
MOTIFS [†] [31, 41]	19.9	32.8	44.7	11.3	19.0	25.0	7.5	12.5	16.9
MOTIFS-Reweight [‡]	20.5	33.5	44.4	12.6	19.1	24.3	8.0	12.9	16.8
MOTIFS-L2+uKD [‡] [31]	-	36.9	50.9	-	22.7	30.1	-	14.0	19.5
MOTIFS-L2+cKD [‡] [31]	-	37.2	50.8	-	22.1	29.6	-	14.2	19.8
MOTIFS-TDE [†] [28]	18.7	29.0	38.2	10.7	16.1	21.1	7.4	11.2	14.9
MOTIFS-PCPL [†] [34]	25.6	38.5	49.3	13.1	19.9	25.6	9.8	14.8	19.6
MOTIFS-STL [†] [3]	15.7	29.4	43.2	10.3	18.4	27.2	6.4	10.6	15.0
MOTIFS-DLFE	30.0	45.8	57.7	17.6	25.6	32.0	11.7	18.1	23.0
VCTree [†] [29, 31]	21.4	35.6	47.8	14.3	23.3	31.4	7.5	12.5	16.7
VCTree-Reweight [‡]	20.6	32.5	41.6	14.1	21.3	27.8	8.0	12.1	15.9
VCTree-L2+uKD [‡] [31]	-	37.7	51.7	-	26.8	35.2	-	13.8	19.1
VCTree-L2+cKD [‡] [31]	-	38.4	52.4	-	26.8	35.8	-	13.9	19.0
VCTree-TDE [†] [28]	20.9	32.4	41.5	12.4	19.1	25.5	7.8	11.5	15.2
VCTree-PCPL [†] [34]	25.1	38.5	49.3	17.2	25.9	32.7	9.9	15.1	19.9
VCTree-STL [†] [3]	16.8	31.8	45.1	12.7	22.0	32.7	6.0	10.0	14.1
VCTree-DLFE	29.1	44.6	56.8	21.6	31.4	38.8	11.7	17.5	22.5

Table 1: Performance comparison in ng-mR@K on VG150 [14, 33]. Models in the first section are with VGG backbone [26]. [†] models implemented or reproduced ourselves with ResNeXt-101-FPN [17] backbone. [‡] models also with the same ResNeXt-101-FPN backbone while their performance are reported by the respective papers. \diamond model using external knowledge bases.

Compare DLFE to other debiasing methods

Model	Predicate Classification (PredCls)			Scene Graph Classification (SGCls)			Scene Graph Detection (SGDet)		
	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
IMP+ [4, 33]	-	9.8	10.5	-	5.8	6.0	-	3.8	4.8
FREQ [29, 41]	8.3	13.0	16.0	5.1	7.2	8.5	4.5	6.1	7.1
MOTIFS [29, 41]	10.8	14.0	15.3	6.3	7.7	8.2	4.2	5.7	6.6
KERN [4]	-	17.7	19.2	-	9.4	10.0	-	6.4	7.3
VCTree [29]	14.0	17.9	19.4	8.2	10.1	10.8	5.2	6.9	8.0
GPS-Net [19]	17.4	21.3	22.8	10.0	11.8	12.6	6.9	8.7	9.8
GB-Net- β^\diamond [39]	-	22.1	24.0	-	12.7	13.4	-	7.1	8.5
MOTIFS [†] [28, 41]	13.0	16.5	17.8	7.2	8.9	9.4	5.3	7.3	8.6
MOTIFS-Focal [‡] [18, 28]	10.9	13.9	15.0	6.3	7.7	8.3	3.9	5.3	6.6
MOTIFS-Resample [‡] [2, 28]	14.7	18.5	20.0	9.1	11.0	11.8	5.9	8.2	9.7
MOTIFS-Reweight [†]	14.3	17.3	18.6	9.5	11.2	11.7	6.7	9.2	10.9
MOTIFS-L2+uKD [‡] [31]	14.2	18.6	20.3	8.6	10.9	11.8	5.7	7.9	9.5
MOTIFS-L2+cKD [‡] [31]	14.4	18.5	20.2	8.7	10.7	11.4	5.8	8.1	9.6
MOTIFS-TDE [†] [28]	17.4	24.2	27.9	9.9	13.1	14.9	6.7	9.2	11.1
MOTIFS-PCPL [†] [34]	19.3	24.3	26.1	9.9	12.0	12.7	8.0	10.7	12.6
MOTIFS-STL [†] [3]	13.3	20.1	22.3	8.5	12.8	14.1	5.4	7.6	9.1
MOTIFS-DLFE	22.1	26.9	28.8	12.8	15.2	15.9	8.6	11.7	13.8
VCTree [†] [28, 29]	14.1	17.7	19.1	9.1	11.3	12.0	5.2	7.1	8.3
VCTree-Reweight [†]	16.3	19.4	20.4	10.6	12.5	13.1	6.6	8.7	10.1
VCTree-L2+uKD [‡] [31]	14.2	18.2	19.9	9.9	12.4	13.4	5.7	7.7	9.2
VCTree-L2+cKD [‡] [31]	14.4	18.4	20.0	9.7	12.4	13.1	5.7	7.7	9.1
VCTree-TDE [†] [28]	19.2	26.2	29.6	11.2	15.2	17.5	6.8	9.5	11.4
VCTree-PCPL [†] [34]	18.7	22.8	24.5	12.7	15.2	16.1	8.1	10.8	12.6
VCTree-STL [†] [3]	14.3	21.4	23.5	10.5	14.6	16.6	5.1	7.1	8.4
VCTree-DLFE	20.8	25.3	27.1	15.8	18.9	20.0	8.6	11.8	13.8

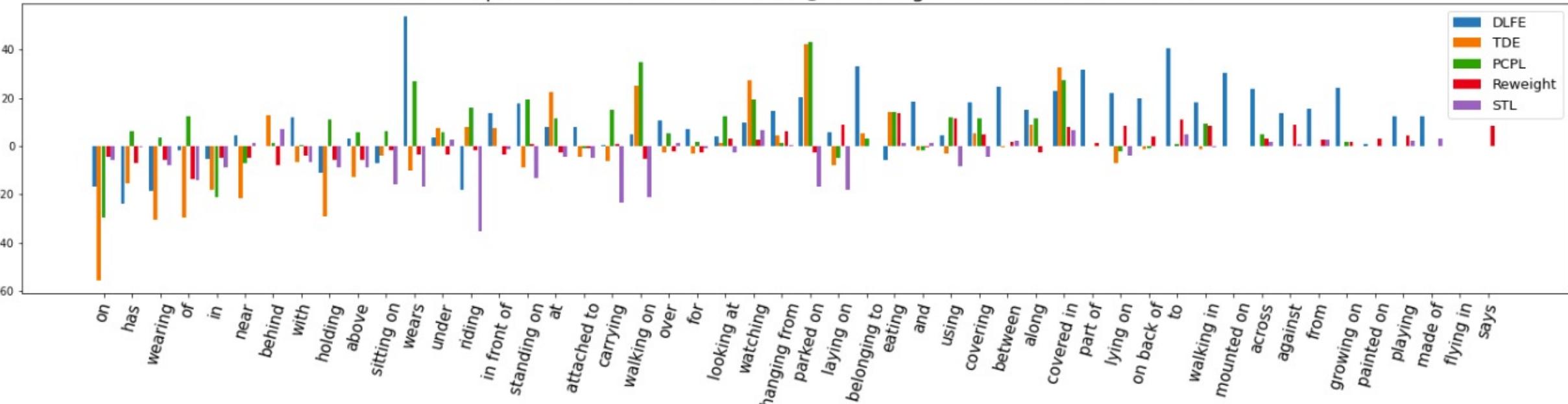
Table 2: Performance comparison of SGG models in graph-constraint mR@K on VG150 [14, 33] testing set. Models in the first section are with VGG backbone [26]. [†], [‡] and \diamond are with the same meanings as in Table 1.

Compare DLFE to other debiasing methods

Model	Head Recalls		Middle Recalls		Tail Recalls	
	R@50	R@100	R@50	R@100	R@50	R@100
MOTIFS [†] [28, 41]	65.9	78.6	30.0	45.4	3.3	9.7
MOTIFS-Reweighting [†]	57.4	69.2	30.7	43.0	13.3	21.5
MOTIFS-TDE [†] [28]	48.3	60.8	34.9	46.1	1.8	5.3
MOTIFS-PCPL [†] [34]	66.5	77.6	41.8	55.2	6.0	13.2
MOTIFS-STL [†] [3]	56.4	70.0	24.1	39.8	9.6	21.2
MOTIFS-DLFE	61.9	72.4	42.8	54.2	31.8	44.6
VCTree [†] [28, 29]	67.5	79.8	34.3	50.0	5.5	12.7
VCTree-Reweighting [†]	61.6	73.4	28.3	38.3	9.0	14.3
VCTree-TDE [†] [28]	54.8	67.5	37.9	49.1	2.5	5.4
VCTree-PCPL [†] [34]	64.5	75.9	42.6	54.2	6.9	16.1
VCTree-STL [†] [3]	57.6	71.1	26.1	41.8	13.8	23.5
VCTree-DLFE	57.5	68.3	36.0	48.2	26.5	38.1

Table 3: Non-graph constraint head, middle and tail recalls (PredCls). \dagger , \ddagger and \diamond are with the same meanings as in Table 1. DLFE improves the tail recalls by a large margin.

Non-Graph Constraint Per-class Recall@20 Change w.r.t. MOTIFS Backbone



Qualitative Results

Scene Graph Visualization

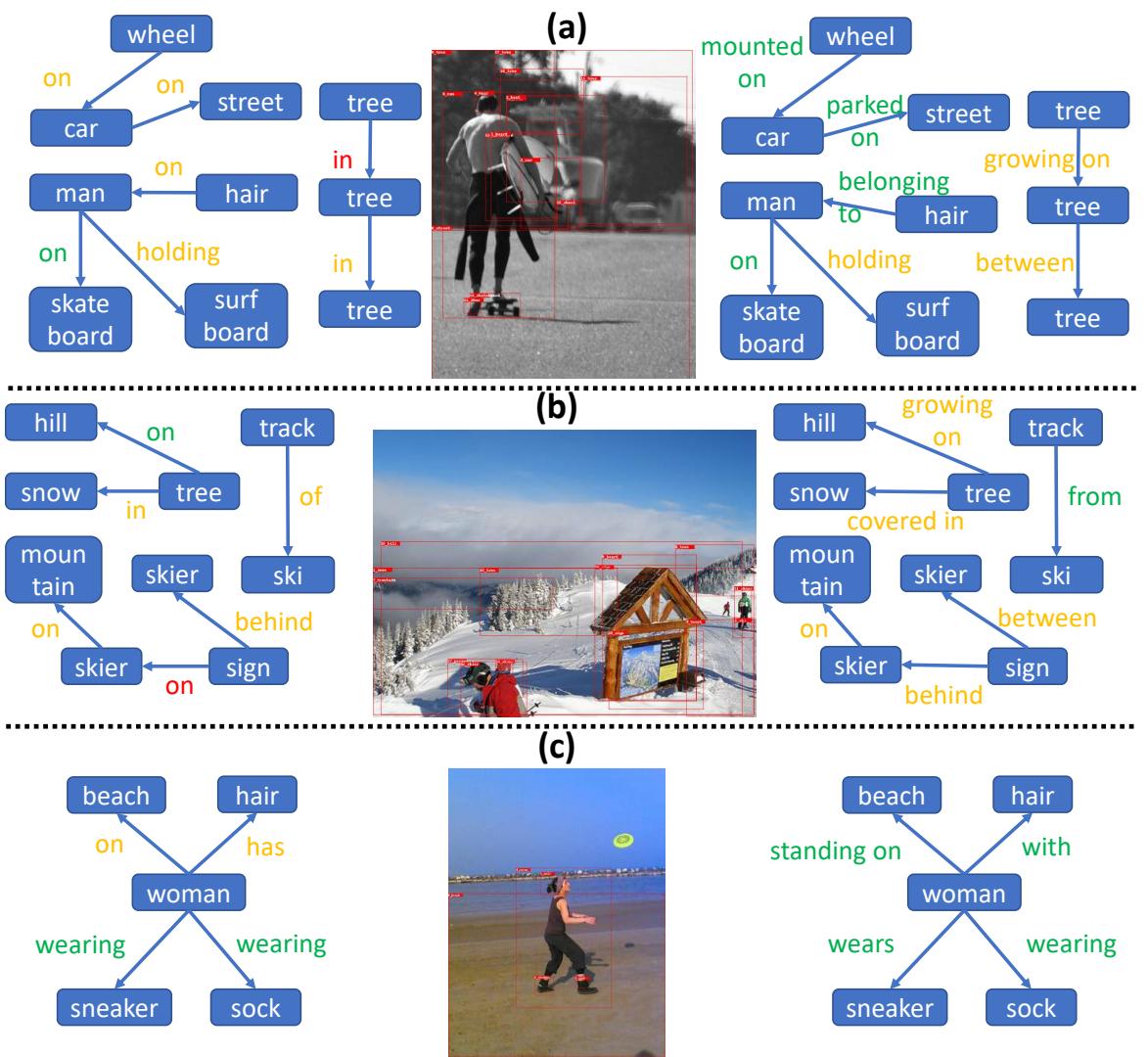
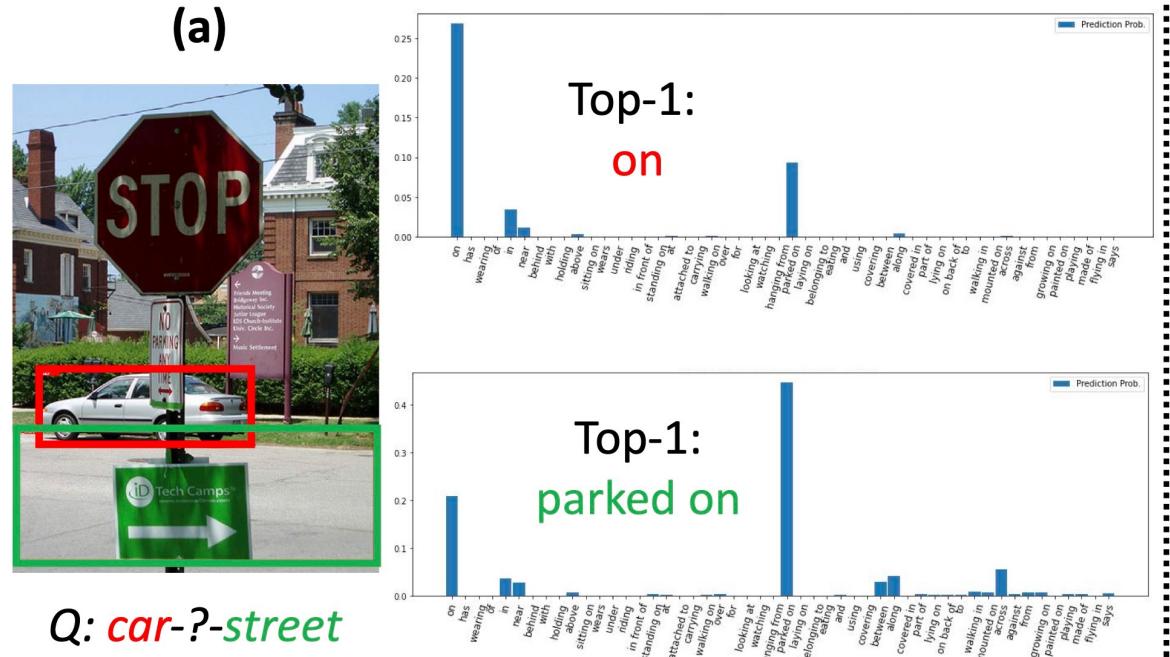
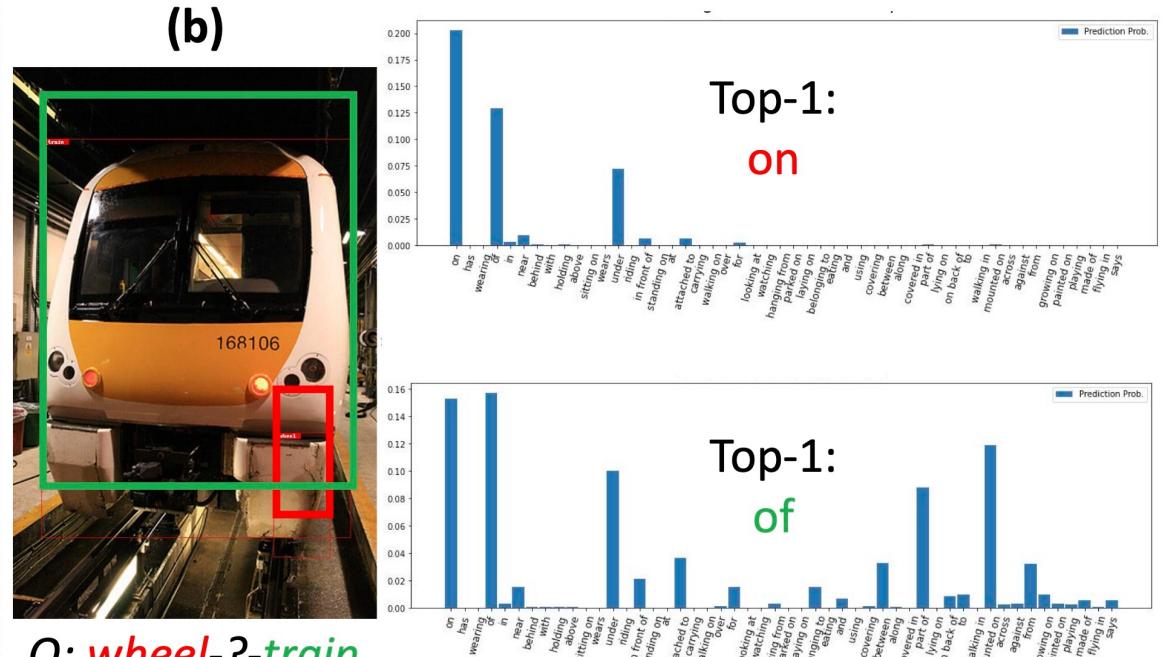


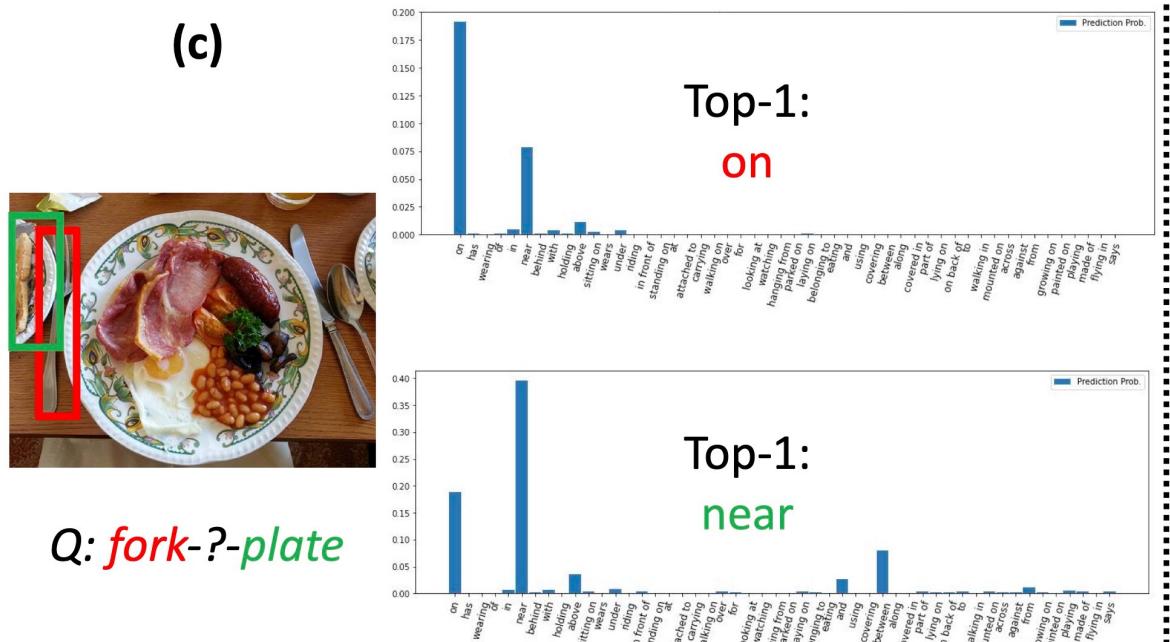
Figure 7: Example scene graphs generated by MOTIFS (left) and MOTIFS-DLFE (right) in PredCls. Green, red and tangerine color denote correct (GT), incorrect (Non-GT and weird) and acceptable (Non-GT but reasonable) predicate prediction, respectively. Only the top 1 prediction is shown for each object pair.



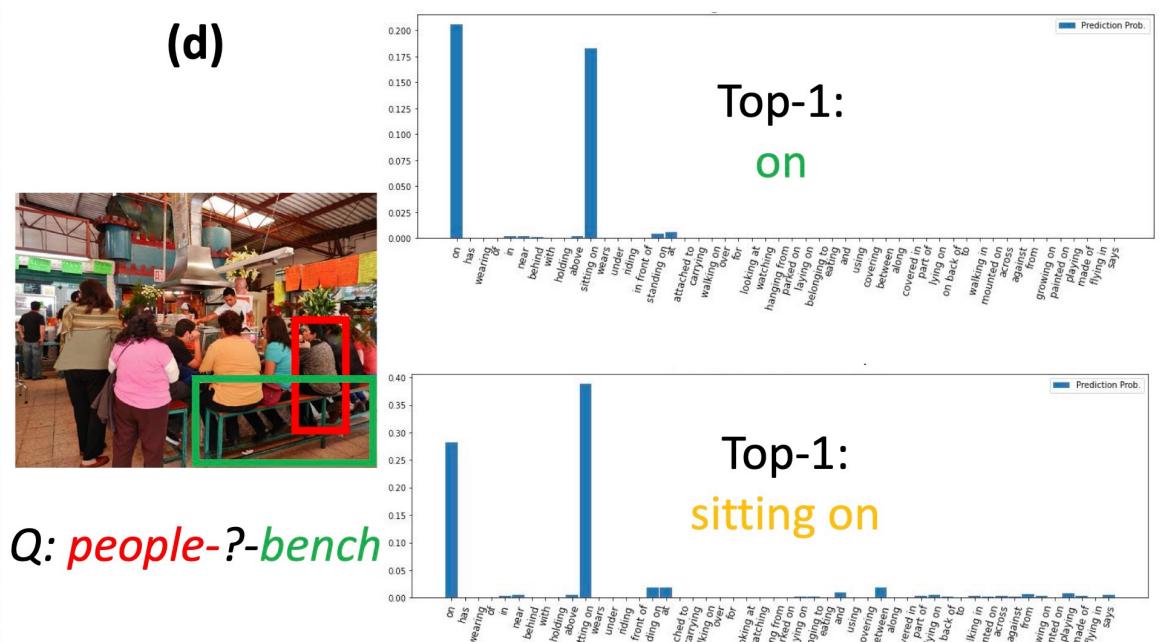
Q: car-?-street



Q: wheel-?-train



Q: fork-?-plate



Conclusion

- we are among the first to deal with the long tail problem in SGG with the cause (unbalanced missing labels) instead of its superficial effect (long-tailed distribution).
- We view SGG as a PU problem and we remove the reporting bias by recovering the per-class unbiased probabilities from the biased ones.
- We propose DLFE which provides more reliable label frequency estimates using augmented data and averages over multiple epochs
- We show that DLFE is more effective in estimating label frequencies, and SGG models with DLFE achieves SOTA debiasing performance in VG dataset and produce significantly more balanced scene graphs.

Thank you for your attention ☺

Source code will be available at <https://github.com/coldmanck/recovering-unbiased-scene-graphs>