Appendix

A. Details of frequency-based predicate grouping and proportional query grouping.

In this section, we provide a detailed algorithm for dividing $C^p = \{c_l^p\}_{l=1}^{N_p}$, a set of N_p predicates, and $\mathcal{Q}$, a set of N_q queries into N_g groups, as introduced in Sec. 3.2 of the main paper. First, $f_p(c_l^p)$, a proportion of a predicate c_l^p is defined as the number of training samples having c_l^p as a predicate label divided by the total number of training samples. Similarly, the frequency of a predicate group $\mathcal{G}_i^p$ is then defined as the sum of frequencies of predicates in the group:

$$f_g(\mathcal{G}_i^p) = \sum_{c_l^p \in \mathcal{G}_i^p} f_p(c_l^p). \tag{12}$$

Then, the algorithm iteratively assigns a predicate to a predicate group from the most frequent predicates to rare ones. A predicate c_l^p is assigned to the current predicate group $\mathcal{G}_i^p$ if the value of $f_g(\mathcal{G}_i^p)$ does not exceed a threshold of $(\frac{1}{2})^i$, where i is initialized as 1. Otherwise, the predicate group index is incremented to move to the next group, and then the predicate is assigned to the new group. This process continues until all predicates have been allocated or until the $N_g - 1$'th group is filled. If there still exist remaining predicates, those are allocated to the final predicate group $\mathcal{G}_{N_g}^p$. After predicate groups are formed, the number of queries in each group $|\mathcal{G}_k^q|$ is determined as a floored result of $f_g(\mathcal{G}_k^p)$ multiplied by the number of queries N_q, where $k = 1, ..., N_g$. Similarly, the remaining queries after the allocation is done are assigned to the last query group $\mathcal{G}_{N_g}^q$. The pseudocode is presented in Algorithm 1.

B. Implementation details.

For Scene Graph Generation, we implement the proposed SpeaQ on ISG [14] and HOTR [16]. Both architectures adopt a ResNet-101 [9] backbone and a 6-layer Transformer encoder. ISG adopts three separate 6-layer decoders for subject, predicate, and object where each decoder takes 300 queries as input, respectively. HOTR consists of a 6-layer instance decoder and a 12-layer predicate decoder, where each decoder takes 100 and 160 decoder queries as input. For Human-Object Interaction Detection, we implement SpeaQ on the smallest model of GEN [25], GEN-VLKT$_s$. GEN-VLKT$_s$ adopts ResNet-50 as a backbone and consists of an instance decoder and a predicate decoder, where both consist of 3 decoder layers and take 64 decoder queries as input. Default hyperparameters for proposed components and for training are presented in Tab. 11. Following baselines, a Non-Maximum Suppression (NMS) is applied to remove duplicate detections. The number of predicate decoder queries N_q is multiplied twice in all three architectures compared to the baselines. Model weights for the backbone, encoder, and decoder are initialized with the DETR weight pre-trained on the Visual Genome and MS-COCO dataset for Scene Graph Generation models and Human-Object Interaction Detection models, respectively. All experiments are conducted with 4 NVIDIA RTX 3090 GPUs.

C. Details about frequency-based predicate groups from previous works.

In Tab. 8 from the main paper, results under adopting predicate groups split on a frequency-basis from previous works BGNN [21] and SHA [7] as $\{\mathcal{G}_i^p\}_{i=1}^{N_g}$ are reported. In BGNN, predicates are split into three groups ($N_g = 3$) named head, body, and tail by the number of training samples where predicates in each of the three groups have more than 10k samples ('head'), between 0.5k and 10k samples ('body'), and less than 0.5k samples ('tail'), respectively. In SHA, predicates are split into multiple groups in the way that the number of training samples of the most common predicate in a group does not exceed the pre-defined threshold μ multiplied by the number of the least common predicate in the same group. Regarding the reference to 'SHA' in Tab. 8 of the main paper, we adopt predicate groups constructed under $\mu = 5$.

D. A running example of quality-aware multi-assignment.

In this section, we demonstrate the necessity of the proposed 'quality-aware' multi-assignment **(Ours)** over the conventional single assignment **(single)** and quality-agnostic multi-assignment **(agnostic)** with the qualitative results in Fig. 4. To be specific, quality-agnostic multi-assignment denotes that d_i, the number of predictions a GT t_i is assigned to, is set equal for every GT, which is two in this example. With regard to an 'ideal' assignment, GT 1 should be assigned to predictions 1 to 3, and GT 2 should be assigned to prediction 4 where predicted bounding boxes and classes equal to that of the corresponding GT. For prediction 5, 'no relation' should be assigned since both the classification and localization results on the subject largely differ from the GT. As illustrated in the figure, ours succeeded in finding the ideal assignment. In contrast, the conventional assignment fails to assign GT 1 to predictions 2 and 3 due to a constraint that a GT can only be assigned to a single GT, and quality-agnostic multi-assignment wrongly assigns GT 2 to prediction 5 due to a constraint that d_i is set equal to every GT. The qualitative examples demonstrate the importance of adaptively assigning a GT to multiple predictions, since insufficient or wrong training signals may be provided otherwise.

Algorithm 1 Frequency-based Predicate Grouping and Proportional Query Grouping

Initialize a predicate group index $i = 1$
Create an empty predicate group $\mathcal{G}_i^p$
Sort the predicate classes by frequency in descending order using the training set
while $i < N_g$ **do**
  for predicate c_l^p in the sorted predicate list **do** ▷ Frequency-based Predicate Grouping
    if $f_g\left(\mathcal{G}_i^p\right) \leq \left(\frac{1}{2}\right)^i$ **then**
      Add c_l^p to $\mathcal{G}_i^p$
    else
      Increment i by 1
      Create an empty predicate group $\mathcal{G}_i^p$
      Add c_l^p to $\mathcal{G}_i^p$
    end if
  end for
end while
if there are predicates remaining **then**
  Add remaining predicates to a predicate group $\mathcal{G}_{N_g}^p$
end if
for query group index k in $k = 1, 2, ...N_g$ **do** ▷ Proportional Query Grouping
  Create an empty query group $\mathcal{G}_k^q$
  Assign $\lfloor N_q f_g(\mathcal{G}_k^p) \rfloor$ queries to a query group $\mathcal{G}_k^q$
end for
if there are queries remaining **then**
  Assign remaining queries to a query group $\mathcal{G}_{N_g}^q$
end if

Model	N_g	k	λ_{rel}	$\mathcal{R}$	Optimizer	Training steps	Initial lr	lr decay step	Decayed lr	Batch size
ISG [14]	4	5	-0.5	max	AdamW [26]	150k iters	10^{-4}	96k'th iter	10^{-5}	20
HOTR [16]	5	4	-0.5	max	AdamW	150k iters	10^{-4}	96k'th iter	10^{-5}	20
GEN [25]	2	5	-0.5	max	AdamW	50 epochs	10^{-4}	40'th epoch	10^{-5}	16

Table 11. **Training details and hyperparemeters.**

Method	IoU > 0.6	IoU > 0.7	IoU > 0.8
Baseline	44.53%	42.72%	42.61%
Ours	**35.38%**	**33.58%**	**33.61%**

Table 12. **Ratio of promising predictions assigned 'no relation(∅)' as a GT.** The lower the percentage, the better.

E. Ratio of 'no relation' assigned to promising predictions.

In Tab. 12, we report the ratio of promising predictions labeled as 'no relation (∅)' to the total number of promising predictions, where a promising prediction is defined as a prediction that is correctly classified and overlaps with the GT in both subject and object with IoU over the certain threshold. The results show that SpeaQ consistently reduces the ratio, which implies that abundant positive training signals are provided to promising predictions.

F. Analysis of training signals provided to a query.

In this section, we validate that a query receives more specialized and abundant training signals under SpeaQ. First, we provide statistics about the predicate group of a predicate assigned to a query belongs to. In Fig. 5, a matrix is plotted where an element in the i-th column and the j-th row denotes an average number of predicates from an i'th predicate group assigned to a query in a j'th query group. As shown in the figure, a GT in a specific predicate group is only assigned to a query in the corresponding query group under SpeaQ, given that every entry in the matrix except for diagonal entries is set to zero. Second, we provide an average number of GTs assigned to a query. Under the conventional assignment, 612 samples are assigned to a query on average. In contrast, 1,540 samples are assigned to a query on average under the SpeaQ, which results in richer training signals being provided to queries. Considering both

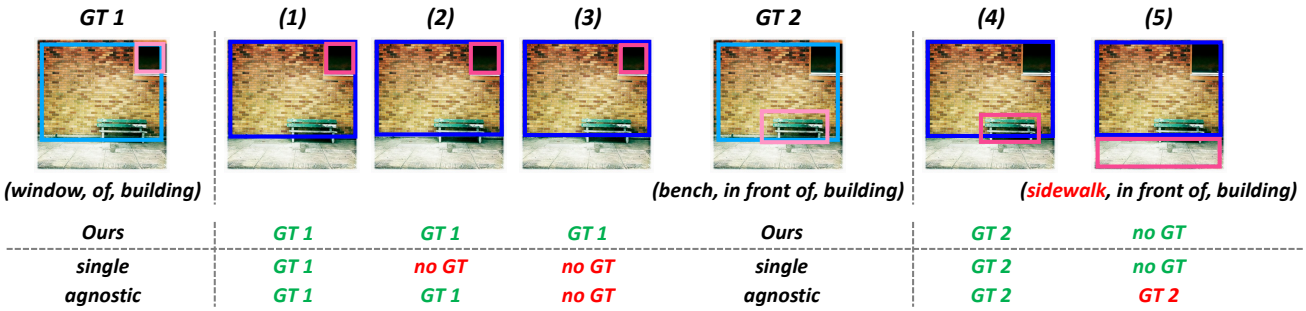

Figure 4. **Qualitative examples of various assignment strategies.** Bounding boxes and labels of two ground-truths (GT 1, 2) and five prediction results (1-5) are illustrated. Note that a prediction label is only specified in case it differs from the most relevant GT. Ideal assignment results are colored green, while wrong assignment results are colored red.

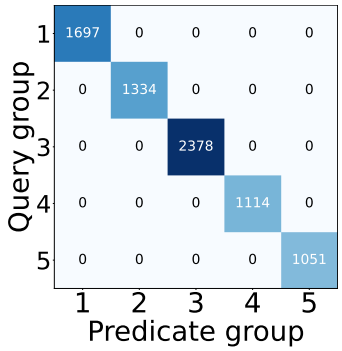

Figure 5. **Assignment Results between predicate and query groups under SpeaQ.** An element in the i-th column and the j-th row denotes an average number of predicates in $\mathcal{G}_i^p$ assigned to a query in $\mathcal{G}_j^q$.

statistics provided, we conclude that a query receives more specialized and abundant training signals under SpeaQ.

G. Further qualitative examples.

In Fig. 6, further qualitative examples are provided. Examples demonstrate that the model trained with SpeaQ succeeds in correctly detecting challenging samples that require both semantic and visual understanding compared to the baseline.

H. Failure cases.

In Fig. 7, we provide qualitative results where the model trained with SpeaQ fails. As shown in the figure, although the predicted label is considered incorrect based on the GT annotations, some predictions are not completely wrong due to the ambiguity of the GT or language. Therefore, we suggest developing a more accurate evaluation metric or annotations addressing the ambiguity of GTs may be an interesting direction for future research.

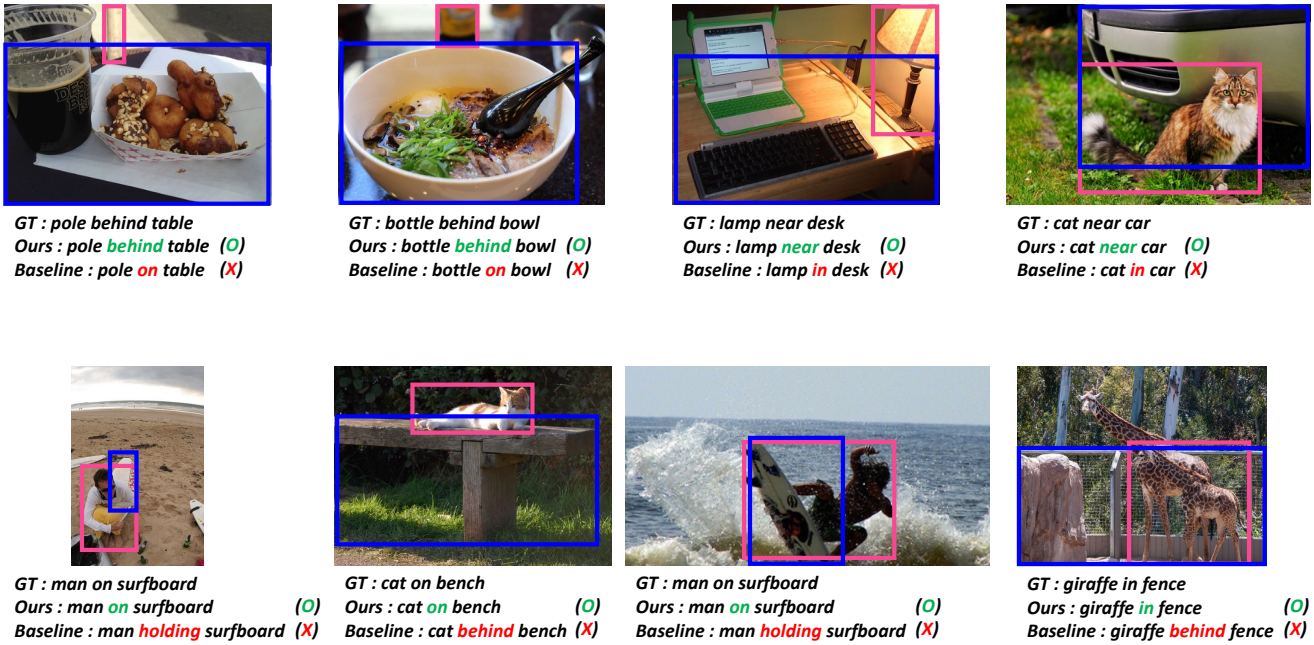

Figure 6. **Further qualitative results on Visual Genome dataset.** Prediction results of the baseline and the model trained with SpeaQ are visualized. Predicates classified correctly are marked green, while predicates that are misclassified are marked red.

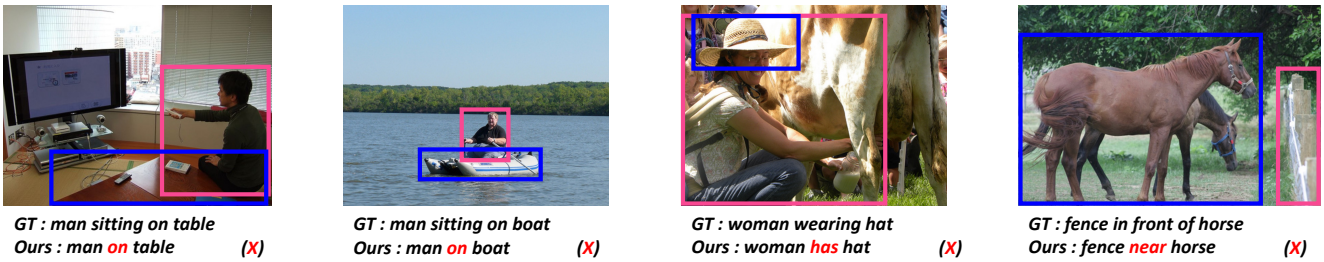

Figure 7. **Failure cases on Visual Genome dataset.** Prediction results of the model trained with SpeaQ are visualized.