Appendix: Visual Distant Supervision for Scene Graph Generation

1. The Denoising Framework: Pseudo-Code

In this section, we provide the pseudo-code of the denoising framework in distantly supervised and semi-supervised settings respectively.

Algorithm 1 Distantly Supervised Denoising Framework

Require: \( \Lambda \): commonsense knowledge base
Require: \( D_S \): distantly labeled image data
Require: \( f(\cdot; \theta) \): any scene graph model, with parameter \( \theta \)
Optional: \( \Phi \): external semantic signal
1: Randomly initialize the model parameter \( \theta^0 \)
2: // Initial E step: estimate the probabilistic distant labels
3: Obtain distant labels \( d = \Psi(s, o, \Lambda) \)
4: if external signal \( \Phi \) available then
5: Initialize \( r^1 = e \)
6: else
7: Initialize \( r^1 = d \)
8: end if
9: // Initial M step: optimize model parameter
10: Optimize \( \theta^1 = \arg \max_\theta \mathcal{L}(D^1_S; \theta^0) \)
11: while not done do
12: // E step: estimate the probabilistic distant labels
13: if external signal \( \Phi \) available then
14: Estimate \( r^t_1 = \omega f_i(s, o; \theta^{t-1}) + (1 - \omega) e_i \)
15: else
16: Estimate \( r^t_1 = f_i(s, o; \theta^{t-1}) \)
17: end if
18: Eliminate noisy object pairs
19: // M step: optimize model parameter
20: Optimize \( \theta^t = \arg \max_\theta \mathcal{L}_p(D^t_S; \theta^{t-1}) \)
21: end while

Algorithm 2 Semi-Supervised Denoising Framework

Require: \( \Lambda \): commonsense knowledge base
Require: \( D_S \): distantly labeled image data
Require: \( D_L \): human-labeled image data
Require: \( f(\cdot; \theta) \): any scene graph model, with parameter \( \theta \)
1: Initialize \( f(\cdot; \theta^0) \) with fully supervised model
   \( \theta^0 = \arg \max_\theta \mathcal{L}_q(D_L; \theta) \)
2: while not done do
3: // E step: estimate the probabilistic distant labels
4: Estimate \( r^t_1 = f_i(s, o; \theta^{t-1}) \)
5: Eliminate noisy object pairs
6: // M1 step: pre-train on distantly labeled data \( D^t_S \)
7: Optimize \( \theta^t_1 = \arg \max_\theta \mathcal{L}_q(D^t_S; \theta) \)
8: // M2 step: fine-tune on human-labeled data \( D_L \)
9: Optimize \( \theta^t_2 = \arg \max_\theta \mathcal{L}_q(D_L; \theta^t_1) \)
10: end while

Scene Graph Model. For the base scene graph model, we follow the implementation of Neural Motif [8], with ResNeXt-101-FPN [4, 7] as the backbone. For predicate classification and scene graph classification, the ratio of positive relation samples and negative relation samples in each image is at most 1:3. For scene graph detection, a relational triplet is considered as positive only if the detected object pairs match ground-truth annotations, i.e., with identical object categories and bounding box IoU > 0.5. We find that this strict constraint leads to sparse positive supervision in experiments, especially in our distantly supervised setting. To address the issue, we change the ratio of positive and negative relation samples in distantly supervised setting to strictly 1:1. During evaluation, we only keep 64 object bounding box predictions. The models are trained using SGD optimizer on 2 NVIDIA GeForce RTX 2080 Ti, with momentum 0.9, batch size 12 and weight decay \( 5 \times 10^{-4} \).

Our Model. All the hyperparameters of our model are selected by grid search on the validation set. (1) In distantly supervised setting, for the denoising framework, the weighting hyperparameter \( \omega \) is 0.9, and 75% noisy object pairs are discarded. In the first iteration, we train the model with learning rate 0.12, and decrease the learning rate 3 times after the plateaus of validation performance. In the second iteration, the learning rate is 0.012, and decays 1 times after the validation performance plateaus. Note that following Devlin et al. [2], the learning rate in the second iteration

2. Implementation Details

In this section, we provide implementation details of our model and baseline methods. For fair comparisons, all the neural models in our experiments are implemented using the same object detector, scene graph model and backbone.

Object Detector. We adopt the object detector implementation from Tang et al. [6]. Specifically, the object detector is trained using SGD optimizer with learning rate \( 8 \times 10^{-3} \) and batch size 8. During the training process, the learning rate is decreased two times by 10 in 30,000 and 40,000 iterations respectively.
The learning rate decays $0.12$ for pre-training, and $0.012$ for fine-tuning. The learning rate decays $2$ and $1$ times in the first and second iterations respectively. To fine-tune the pre-trained distantly supervised model without semi-supervised denoising, we optimize with learning rate $0.012$ that decays $2$ times. The decay rate is $10$ for all models.

**Baselines.** For the Limited Labels [1], we train the decision trees for 200 different trails on 10 randomly sampled human-labeled seed instances for each relation, and select the best models according to the performance on the validation set. For the weakly supervised model, since Visual Genome does not have image-level captions, we utilize all the images in Visual Genome training set that have captions from COCO [5], resulting in 35,340 images with captions in total. Then we train the weakly supervised model with all Visual Genome object annotations from these images, and relation labels parsed from the corresponding captions. For the Cleanness Loss [3], we denoise with soft weight given by the confidence of the scene graph model.

### 3. Data statistics

In our main experiments, we adopt the refined relation schemes from Chen et al. [1], which removes hypernyms (e.g., near and on), redundant synonyms (e.g., lying on and laying on), and unclear relations (e.g., and) in the most frequent 50 relation categories in Visual Genome, resulting in 20 well-defined relation categories. The relation dependencies from Chen et al. [1] are shown in Figure 1. The dataset contains 10,986, 1,566 and 3,025 images in training, validation and test set respectively, where each image contains an average of 13.58 objects, 2.10 human-labeled relation instances and 15.60 distantly labeled relation instances.

### 4. Supplementary Experiments

**Case Study.** We provide qualitative examples in Figure 2 for better understanding of different scene graph models.

**Results on 50 Visual Genome Predicates.** We report the experimental results on 50 Visual Genome predicates in Table 1. We observe that although reasonable performance can be achieved, the improvement from distant supervision and the denoising framework shrinks. This is because that the 50 relations are not well-defined, where the major relations are problematic hypernym (e.g., near and on), redundant synonym (e.g., lying on and laying on), and unclear (e.g., and) relations, as pointed out by Chen et al. [1]. The problematic relation schemes can bring difficulties to denoising distant supervision.

**Results on 1,700 Visual Genome Predicates.** Visual distant supervision can alleviate the long-tail problem and therefore enables large-scale visual relation extraction. To
<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy</th>
<th>Mean Accuracy</th>
<th># Non-zero Predicates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>top-1</td>
<td>top-5</td>
<td>top-10</td>
</tr>
<tr>
<td>FS Motif [8]</td>
<td>64.05</td>
<td>81.35</td>
<td>85.05</td>
</tr>
<tr>
<td>SS Motif + DNS (Ours)</td>
<td><strong>66.10</strong></td>
<td><strong>84.26</strong></td>
<td><strong>87.72</strong></td>
</tr>
</tbody>
</table>

Table 2. Results of visual distant supervision on Visual Genome 1,700 predicates (%). FS: fully supervised, SS: semi-supervised.

investigate the effectiveness of visual distant supervision in handling large-scale visual relations, we refine the full Visual Genome predicates following the principles proposed by [1], resulting in 1,700 well-defined predicates. In addition to top-K accuracy, to better focus on the long-tail performance, we also report top-K mean accuracy and the number of non-zero predicates (i.e., predicates with at least one correctly predicted instance). From the experimental results in Table 2, we observe that our model significantly outperforms its fully supervised counterpart. Notably, our model nearly doubles the top-K mean accuracy and the number of non-zero predicates, demonstrating the promising potential of visual distant supervision in handling large-scale visual relations in the future.

References


Figure 2. Qualitative examples of model predictions in predicate classification task. We show top 10 predictions from (1) models that do not utilize human-labeled data, including weakly supervised and distantly supervised model, and (2) models that leverage human-labeled data, including fully supervised model and our semi-supervised model. Green edges: predictions that match Visual Genome annotations, blue edges: plausible predictions that are not labeled in Visual Genome, red edges: implausible predictions.