-Supplementary Material-
Grounding Consistency: Distilling Spatial Common Sense for Precise Visual Relationship Detection

1. Outline

In order to provide more insight into our main paper, we present additional information regarding:

- implementation details to aid the reproducibility of our results (section 2)
- the grounder employed for our experiments and the positive attributes of two-step grounding (section 3)
- the mathematical formulation of entropy ranking analysis that motivates the focus on proximal predicates (section 4)
- negative graph completion statistics (section 5)
- the problem of annotation redundancy in VG200 causing artificial Precision bounds (section 6)
- additional quantitative and qualitative results of experimentation with GCD (section 7)

2. Implementation details

Software/Hardware used All code is written in Python using the PyTorch framework. We perform all experiments in an Ubuntu 16.04 machine with 64GB RAM and a single NVIDIA 2080Ti GPU. In our Grounding Consistency Distillation (GCD) scheme, teacher training lasts on average 9 minutes for VRD and 75 minutes for VG200 per epoch. These times decrease to 3 and 55 minutes respectively in student training since inference on grounder is not performed and teacher's predictions can be pre-computed. On average, models need 16 epochs to converge on VRD and 8 on VG.

Hyperparameters We apply the Adam optimization algorithm [5] with weight decay equal to 5×10^{-4} for VRD and 5×10^{-5} for VG200. We consider batches of 16 images, that correspond to an average of 128 positive examples of relationships. The learning rate is initialized to 0.002 and dynamically adjusted during training. We find this setup reasonably efficient for all tested models.

Re-implementations We utilize the PyTorch implementation of Faster-RCNN [7] as a per-ROI feature extractor and freeze its parameters in order to isolate the effects of GCD with respect to different model architectures. Models are re-implemented according to the respective authors' publicly released code. When this is not available, we make the appropriate assumptions according to their architecture descriptions. As shown in Table 1, our re-implementations are consistent with the originally reported results, despite optimization parameters are not always given by the authors. We observe a large improvement for VTransE due to finetuning and a drop for RelDN, mainly ascribed to not jointly training the backbone network.

Tasks Depending on the information of the ground truth graph provided to a SGG model the following tasks are defined:

- Predicate Detection (PredDet): Given objects' boxes, categories as well as which of them interact, classify relationships of interacting object pairs.
- Predicate Classification (PredCls): Given objects' boxes and categories predict relationships for all object pairs.
- Scene Graph Classification (SGCls):Given objects' boxes, classify them and predict relationships for all object pairs.
- Scene Graph Generation (SGGen): Nothing is given. Detect and classify objects, then predict relationships for all object pairs. A detection is considered positive if the subject's and object's boxes have $IoU > 0.5$ with ground truth.
- Phrase Detection (PhrDet): Same as SGGen but a detection is considered positive when the predicate box (minimum box containing subject's and object's boxes) has $IoU > 0.5$ with ground truth.

Since GCD focuses on solving context bias induced by the annotations' statistics, incorporating object detection/classification accuracy as part of relationship detection

Model	Original	Ours	Dataset
VTransE [10]	44.76	53.17	VRD
Motis-Net [9]	65.20	62.54	VG200
RelDN [11]	68.30	57.83	VG200
ATR-Net [2]	58.40	57.69	VRD
UVTransE [4]	55.50	56.88	VRD
HGAT-Net [6]	59.54	57.00	VRD

Table 1. Comparison of originally reported R@50 performance to our re-implementations.

would be to no purpose. This is the reason why we choose to evaluate our method on PredDet and PredCls avoiding interference with object detection errors. Typically, micro-Recall@50 on PredDet is reported for VRD and macro-Recall@50 on PredCls for VG200. In micro-Recall, the true positives to positives ratio is calculated across all samples while in macro-Recall it is averaged across all images.

3. Grounder

Description The task of grounding is to locate the referring entities of a subject-predicate-object triplet to the corresponding image regions. For example, given the triplet `dog wear shirt`, the grounder has to locate, if existent, the `dog` and the `shirt` it `wears`.

A triplet can refer to more than one pairs of an image. For example, in Fig. 1b, grounding the triplet `person on street` could be achieved by both `persons`. In order to resolve this ambiguity, we split the subject/object localization into two independent problems by conditioning the object's grounding to the subject's ground-truth box and vice versa. An example is depicted in Fig. 1a. First, given the `jeans` (red box), we detect the `person` (green box) who is `wearing` them. Simultaneously, conditioning on the `person`, we ground the `jeans`. This way, despite the fact that both `persons wear jeans`, only the ground-truth are grounded.

Architecture The overall Grounder's architecture is presented in Fig. 3. Let us consider the case where we predict the object's bounding box (the inverse is symmetric). We represent the normalized coordinates of the subject's box center as $s_c = [s_{c,x}, s_{c,y}] \in \mathbb{R}^2$ and the normalized width and height as $s_b = [s_{b,w}, s_{b,h}] \in \mathbb{R}^2$. The subject's and object's semantic information s_{sem}, o_{sem} along with the relationship detection network's prediction r compose the predicted triplet $t =< s_{sem}, r, o_{sem} >$. The probability distribution $Pr(o_b, o_c)$ of the object's dimensions and center can be modeled as:

$$\begin{aligned} Pr(o_b, o_c) &= Pr(o_b, o_c|t, s_c, s_b) \\ &= Pr(o_c|t, s_c, s_b, o_b)Pr(o_b|t, s_c, s_b) \end{aligned}$$

and we assume that the object's dimensions o_b are independent of the subject's location s_c, leading to:

$$Pr(o_b, o_c) = Pr(o_c|t, s_c, s_b, o_b)Pr(o_b|t, s_b)$$

Inspired by [1], we model $Pr(o_b|t, s_b)$ as a sequence of linear layers with ReLU activation units which, given t and s_b, regresses the normalized height and width of the object's bounding box $\hat{o_b}$. This stage is supervised using an MSE Loss.

To model $Pr(o_c|t, s_c, s_b, \hat{o_b})$, we regress a heatmap assessing the probability that the object's box is centered at each position. We first encode the image with ResNet-50 [3] into an $H \times W \times D$ feature map $\mathbf{F}$, where H, W are the spatial dimensions and D the feature dimension. Then, we calculate a language-guided attention [8] mask A_{att} of size $H \times W$, as the inner product of the feature map $\mathbf{F}$ and a learnable D-dimensional vector constructed from the object's word embedding. This mask is applied on the feature map to obtain $\mathbf{F}_{att} = \mathbf{F} \star A_{att}$.

In order to evaluate all possible locations for the object, we convolve each channel of $\mathbf{F}_{att}$ with an all-ones kernel of spatial dimensions determined by $\hat{o_b}$. This operation results in an $H \times W \times D$ feature map $\mathbf{F}_{gath}$ where each feature vector represents the gathered visual information of a bounding box being centered at the respective location. Lastly, we concatenate each D-dimensional feature vector with an S-dimensional vector of spatial information extracted from the subject's box s_b and the object's box, as if the latter is centered at the corresponding position. This $H \times W \times (D+S)$ feature map is used to regress the spatial distribution of the box's center, resulting in the heatmap $\mathbf{h}_o$ of the main paper.

Qualitative Examples Fig. 1b showcases that the Grounder is able to locate all pairs the input triplet can refer to. However, it often struggles with disambiguation of same instance entities that lie too close. As an example, in Fig. 1c multiple `persons` are `on` different pairs of `skis`. When conditioning on the `skis` (in red box), the grounder identifies the correct `person` but also the one next to him. Similarly, both the correct pair of `skis` and the one next to it are detected. Although such errors are not significant when evaluating a grounder, they negatively impact teacher's training. This behavior explains the gap between GCD and oracle negative mining supported with NCE (see Table 3 of main paper).

4. Entropy Ranking

We define a context-conditioned predicate distribution as:

$$p_{ij}(k) = Pr(predicate = k|subj = i, obj = j)$$

We do not consider the background (unlabeled) samples since background is not treated as a class. Furthermore, to reduce noise, we exclude some contexts that are

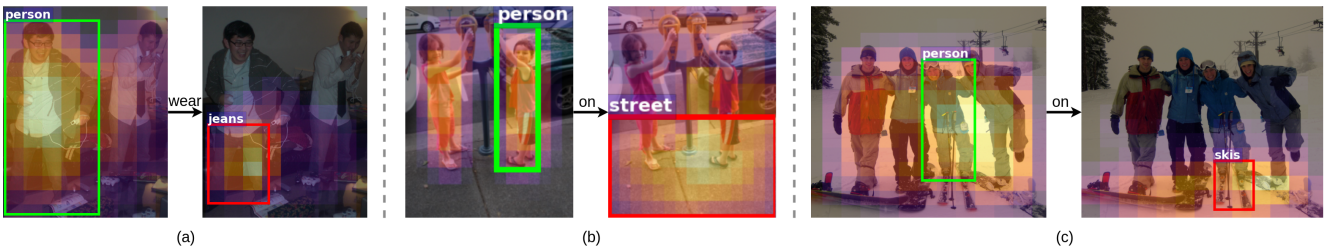

Figure 1. Grounding examples. (a) Conditioning on a ground-truth box to estimate the other mitigates conflicts such as the coexistence of multiple `persons wearing jeans`. (b) Our grounder is able to locate all entities related to a ground-truth box, such as two `persons` both standing `on` the `street`. (c) Objects of the same category lying closely are a common grounding pitfall. Here, the grounder is confused by two `persons` both `on skis` and incorrectly associates the `person` in green box with the `skis` next to the red box.

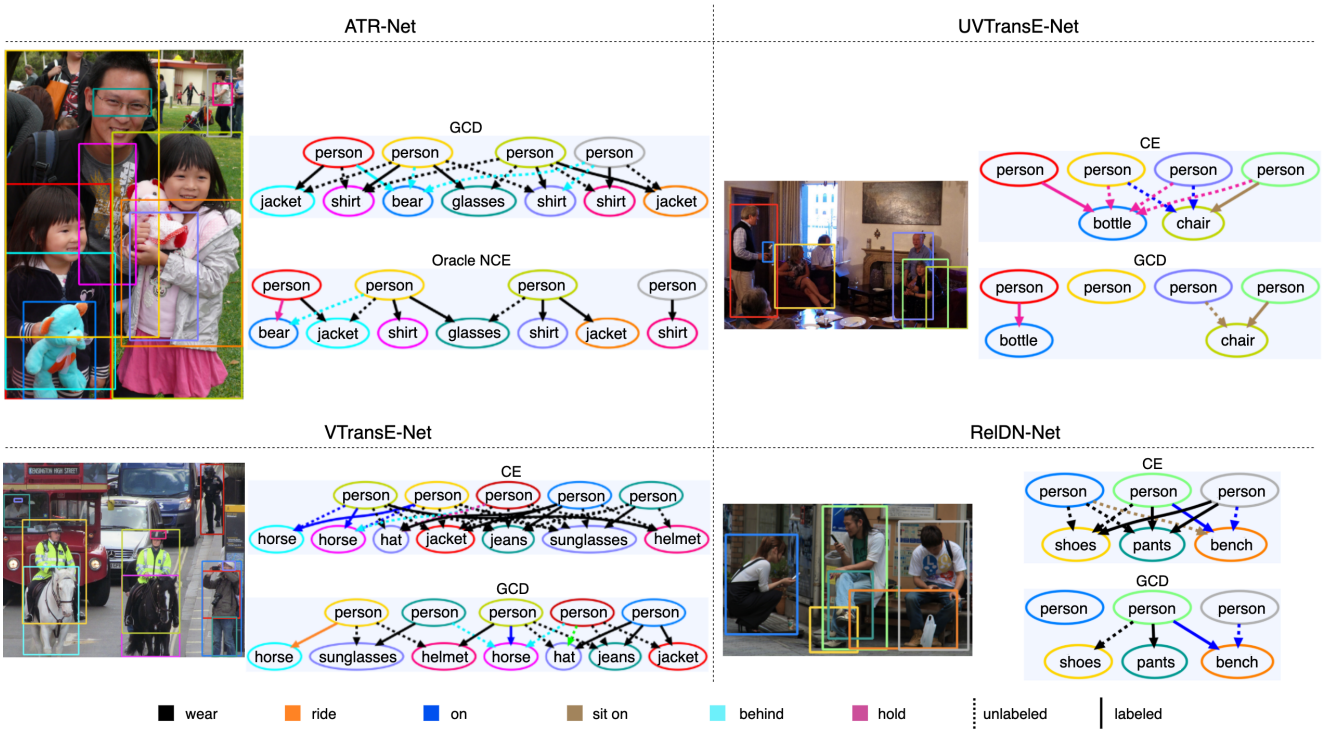

Figure 2. Left column: qualitative comparison of GCD with the oracle NCE of Table 3 in main paper. In cases where objects are in close proximity GCD has a hard time disambiguating the graph due to grounder imperfections. On the other hand the oracle model, since it is trained with mined negatives, develops a more precise understanding of relationships' spatial attributes. Right: supplementary qualitative results for UVTransE [4]. For clarity only proximal relationships are shown. Best viewed in color.

naturally biased and do not capture reporting bias e.g. `sky-above-person` and filter out contexts with a very small amount of samples. Each context is represented by the class with the most samples:

$$\hat{k}(i,j) = \underset{k}{argmax}\ p_{ij}(k)$$

and the set of contexts that are represented by a class k is

$$\mathcal{C}_k = \{i,j : \hat{k}(i,j) = k\}$$

Finally we calculate the mean entropy of each predicate

$$e(k) = \frac{1}{|\mathcal{C}_k|} \sum_{i,j \in \mathcal{C}_k} E(p_{ij})$$

where $E(\cdot)$ is the Entropy function. In Fig. .4 we present the entropy ranking for VG200 predicates. Similarly to VRD, proximal predicates tend to be more biased on context (lower entropy).

5. Negative Graph Completion

Table 2 contains all proximal predicates for VRD and VG200, as well as the way they are categorized to Possessive and Belonging. Additionally, Fig. 5 compares the subject-object overlap distribution of proximal predicates to that of all predicates, validating an increased overlap between the boxes of the referring entities that are connected

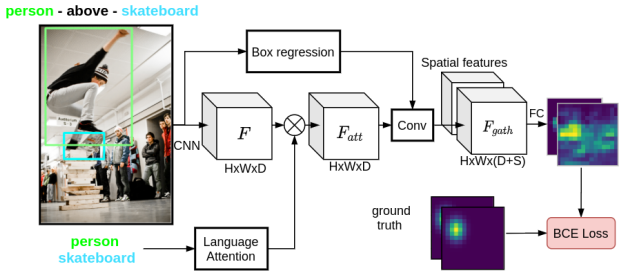

Figure 3. Architecture of the grounding network. We first regress the box's size. Then, we employ attention and convolutional layers to create a feature-map representing spatially gathered visual information. Lastly, we concatenate spatial and visual information and assess the probability heatmap of the entity to be grounded.

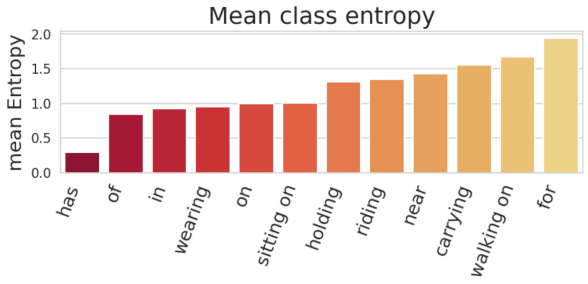

Figure 4. Entropy ranking for VG200. We notice a similar pattern to VRD, with the proximal predicates, e.g. `has`, `of`, `in`, displaying the lowest entropy and thus creating the most biased per-context distributions.

Rules	VRD	VG200
Possessive	carry, contain, cover, drive, eat, feed, fly, has, hit, hold, kick, play with, pull, ride, touch, use, wear, with	carrying, eating, has, holding, playing, riding, using, wearing, wears, with
Belonging	at, drive on, in, inside, lean on, lying on, on, park on,rest on, sit on, skate on, sleep on, stand on	at, attached to, belonging to, flying in,for,from, growing on, hanging from, in, laying on, looking at, lying on, made of,mounted on, of, on, painted on, parked on, part of, says, sitting on, standing on, to,walking in, walking on, watching

Table 2. Sets of predicates that each rule applies to in order to mine negative samples.

with a proximal predicate.

Proximal predicates are further used to extract negative samples and augment test sets with challenging examples. In Fig. 6 we compare the number of positive samples with the generated negatives for the top proximal predicates in VRD's test set. We observe that negative samples surpass almost an order of magnitude positives, due to the combinatorial nature of our rules, thus overfitting context is heavily penalized in our mP$^+$ and f-mP$^+$ metrics.

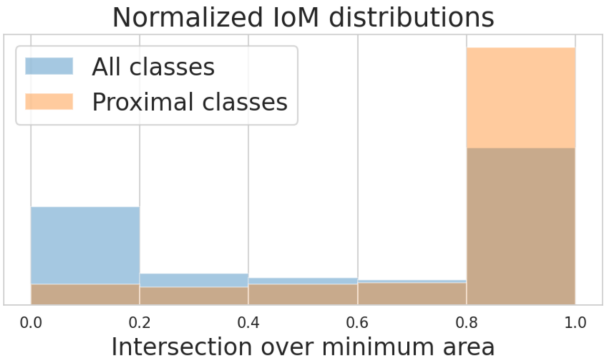

Figure 5. Distributions of Intersection over Minimum area (IoM) for all classes and proximal classes. By definition, proximal predicates display a characteristic larger IoM.

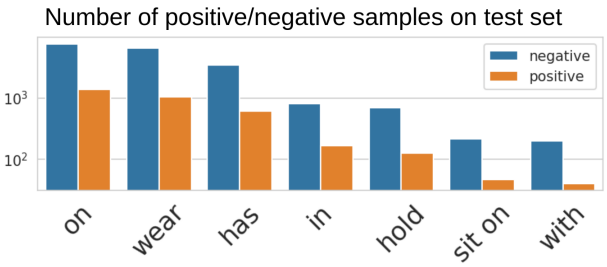

Figure 6. Comparison of the number of negative samples, generated from our rules, to positives on VRD test set. Our rules populate VRD in a way that each one of the dominant classes takes advantage of a significant number of negatives. Best viewed in color.

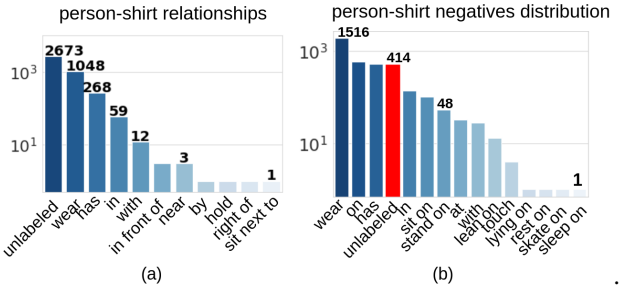

Figure 7. (a): Predicate distribution for `person-shirt` context. (b): Re-distribution of previously unlabeled samples after populating the graph with negatives. Unlabeled samples that do not acquire a negative label are shown in red. Best viewed in color.

Laslty, Fig. 7 depicts the predicate distribution shift caused by negative examples. Previously biased contexts, such as `person-shirt`, are calibrated with a multitude of generated negatives that span the predicate space. Most importantly, more frequent classes have more negative examples, facilitating the learning of their meaning.

Models	VRD (PredDet)			
	R@50	mP$^+$	f-mP$^+$	HarMean
VTransE [10]	53.17	17.42	26.95	35.77
Motis-Net [9]	55.06	20.67	32.38	40.78
RelDN [11]	55.02	22.94	36.63	43.98
ATR-Net [2]	57.69	23.87	38.78	46.38
UVTransE [4]	56.88	21.63	34.69	43.10
HGAT-Net [6]	57.00	22.46	36.26	44.32
VTransE + GCself-D	51.98	18.35	28.76	37.03
Motifs-Net + GCself-D	54.81	22.92	37.45	44.50
RelDN + GCself-D	53.82	25.95	42.90	47.74
ATR-Net + GCself-D	57.59	28.98	48.33	52.56
UVTransE + GCself-D	55.42	27.61	46.16	50.37
HGAT-Net + GCself-D	56.87	23.49	38.23	45.72

Table 3. Instead of using ATR-Net [2] as a universal teacher, both teacher and student share the same architecture (GCself-D). Results are reported in the same format as Table 1 of main paper for the VRD dataset.

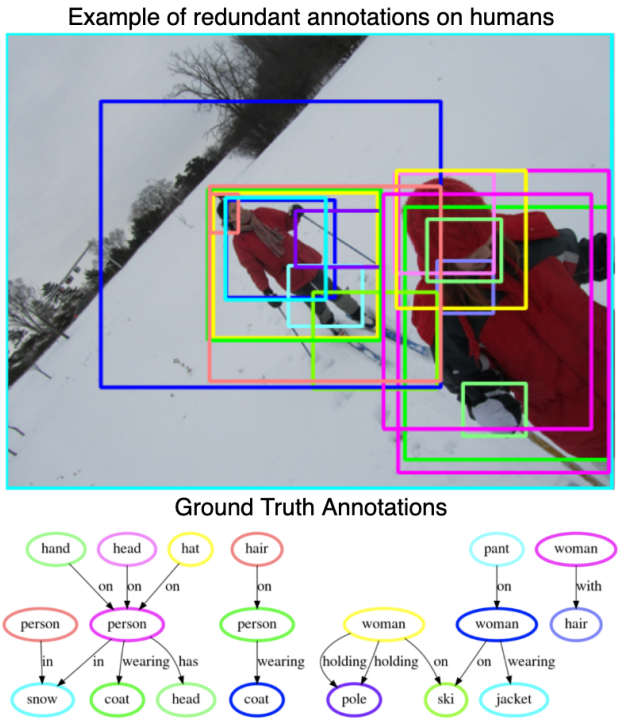

Figure 8. Example of redundant object annotations in VG200. Even though only two people are depicted, there are six separate object annotations assigned to a person, with each being related to other objects in different ways. In the test set, we measure 12% redundancy in annotations of people ($IoU > 0.5$) when we consider all object categories representing a person equivalent (e.g. `person`, `woman`, `man`, `kid` etc). Best viewed in color.

6. The reason behind VG200's seemingly low precision

In Table 1 of the main paper Precision achieved on VG200 appears to be noticeably lower in comparison to VRD. This is the result of VG200's annotations often being redundant, meaning that an entity can at times be labeled more than once and even with a different category e.g. a human is assigned two bounding boxes: one labeled as `person` and an other as `woman`. An example of such redundancy is presented in Fig. 8. Adding to this, redundant objects do not always share the same relationships with other objects. For example, in Fig. 8 the same human (on the left) is annotated four times as either `woman` or `person` and each time having different relationships to other objects in the image. As a result, our *negative graph completion* rules (section 4 of main paper) will occasionally *falsely* generate negative labels that will bound Precision metrics to deceptively low values. Fixing annotation redundancy is outside the scope of this work.

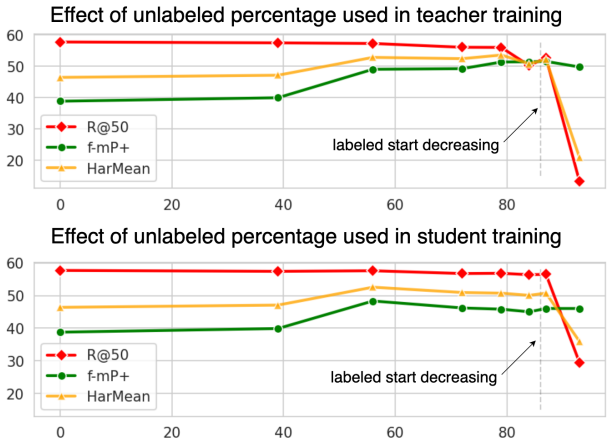

Figure 9. The ratio of unlabeled to labeled samples seems to have a similar effect in both teacher and student training. The more unlabeled data we use, the higher precision (f-mP$^+$) and Harmonic Mean get, with a non-substantial recall (R@50) drop. We observe a break point where we use all the available unlabeled data and decrease the labeled samples. Although there is a steep R@50 drop, the plentiful amount of unlabeled data is able to retain a high Precision. Showing results for ATR-Net [2].

7. Supplementary experiments and results

Self-Distillation An alternative to using a single teacher for all models is to utilize the same architecture both as teacher and student (GCself-D). In Table 3 we present the results of such a scheme. Different teacher architectures showcase different capabilities in developing spatial common sense. This means that weaker models (e.g. VTransE-Net) will not achieve maximum gains as they would with a better teacher.

Qualitative results In Fig. 2 we present supplementary qualitative results for UVTransE-Net [4], VTransE-Net [10] and RelDN-Net [11]. It is worth noting that VTransE seems to be in a disadvantage compared to other models. This is due to the fact that it uses simpler spatial features that are

Models	R@50	mP$^+$	f-mP$^+$	HarMean	R@50	mP$^+$	f-mP$^+$	HarMean	R@50	mP$^+$	f-mP$^+$	HarMean
	+GCD-D				+GraphL				+GCD			
VTransE	-3.67	+0.06	+2.19	+0.02	-0.88	+16.53	+30.46	+17.92	+1.59	+17.45	+35.88	+22.02
Motis-Net	-2.20	+19.59	+26.28	+13.99	+1.54	+22.11	+29.71	+17.63	+0.10	+23.75	+31.04	+17.58
RelDN	-2.04	+17.52	+21.62	+11.11	+0.65	+9.37	+12.26	+7.31	-1.90	+9.94	+13.13	+6.60
ATR-Net	-1.46	+21.16	+26.28	+13.48	-0.28	+15.93	+20.22	+10.76	-0.17	+19.31	+24.63	+13.31
UVTransE	-3.34	+36.25	+42.20	+20.66	-0.74	+29.91	+32.20	+17.44	-0.29	+30.37	+34.82	+18.95
HGAT-Net	-1.77	+3.61	+4.63	+2.05	+1.32	+13.85	+14.64	+9.07	-1.33	+14.87	+17.65	+9.46
	+GCD-G				SpatDistill				oracle with NCE			
VTransE	+0.17	+12.11	+13.17	+8.44	-0.53	+11.08	+13.73	+8.50	+0.47	+35.30	+41.97	+24.65
Motis-Net	+2.92	+11.47	+14.18	+9.73	+1.63	+21.96	+26.41	+15.94	+2.40	+49.25	+59.11	+32.03
RelDN	-1.09	-4.93	-4.45	-3.14	-0.67	+5.71	+7.89	+4.30	-0.78	+24.98	+29.70	+15.52
ATR-Net	-0.72	-0.45	+0.77	+0.17	-1.88	+6.34	+8.69	+4.20	-1.25	+35.61	+42.32	+20.57
UVTransE	+0.76	+7.91	+8.50	+5.43	-0.23	+17.61	+19.78	+11.32	-0.55	+46.05	+52.18	+26.73
HGAT-Net	+0.95	+5.30	+5.24	+3.53	+1.11	+13.31	+14.12	+8.68	+0.25	+42.97	+47.55	+24.67

Table 4. Expanded parts of Tables 2 and 3 of main paper showing relative improvements for GCD-D, GraphL[11], GCD, GCD-G, SpatDistill and oracle NCE on all models.

not able to fully capture the spatial configuration between objects, something also portrayed by its lower precision relative to other models (Table 1 of main paper). We also provide a comparison of oracle NCE (Table 3 in main paper) with GCD (Fig. 2 top left). In cases where objects are in close proximity oracle NCE can outperform GCD. As we explain in the main paper this is a result of the grounder's imperfections (Fig. 1c).

How many unlabeled data? As shown in Fig. 9, by increasing the ratio of unlabeled samples used during training f-mP$^+$ and HarMean improve up to a saturation point. This behavior is observed in both teacher and student training. When labeled samples start decreasing (dashed vertical line in Fig. 9) R@50 rapidly drops while Precision stays on the same level as before. This shows that even with a smaller amount of labeled samples, GCD is able to develop and distill spatial common sense using the abundance of unlabeled samples provided.

Expanded averaged metrics In Table 4 we present the per-model relative improvements that produce Tables 2 and 3 of the main paper when averaged. From this, we can confirm that GCD outperforms GraphL in terms of precision (blue and orange) while Distillation manages to consistently reduce the negative effect on recall (gray), even though the exact boost of both methods depends on the model architecture.

References

[1] Guillem Collell, Luc Van Gool, and Marie-Francine Moens. Acquiring Common Sense Spatial Knowledge through Implicit Spatial Templates. In *Proc. AAAI*, 2018.

[2] Nikolaos Gkanatsios, Vassilis Pitsikalis, Petros Koutras, and Petros Maragos. Attention-Translation-Relation Network for Scalable Scene Graph Generation. In *Proc. ICCV Workshops*, 2019.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385, 2015.

[4] Zih-Siou Hung, Arun Mallya, and Svetlana Lazebnik. Contextual Translation Embedding for Visual Relationship Detection and Scene Graph Generation. *PAMI*, 2020.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.

[6] Li Mi and Zhenzhong Chen. Hierarchical Graph Attention Network for Visual Relationship Detection. In *Proc. CVPR*, 2020.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *Proc. NeurIPS*, 2015.

[8] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, 2015.

[9] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural Motifs: Scene Graph Parsing with Global Context. In *Proc. CVPR*, 2018.

[10] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual Translation Embedding Network for Visual Relation Detection. In *Proc. CVPR*, 2017.

[11] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical Contrastive Losses for Scene Graph Generation. In *Proc. CVPR*, 2019.