Supplementary Material for "BGT-Net: Bidirectional GRU Transformer Network for Scene Graph Generation"

Naina Dhingra Florian Ritter Andreas Kunz

Innovation Center Virtual Reality, ETH Zurich

{ndhingra, kunz}@iwf.mavt.ethz.ch, ritterf@ethz.ch

Abstract

The supplementary material is organized in the following manner: 1) section 1: a comprehensive review of BGT-Net without BiGRU layer; 2) section 2: more ablation study results; 3) section 3: hyper-parameter study; 4) section 4: more qualitative results

1. BGT-Net without Bi-directional GRU

BGT-Net (no BiGRU) is the BGT-Net with no BiGRU in it. We experimented with it to see how Bi-directional GRU effects the performance of the network.

The performance of the BGT-Net (no BiGRU) for PredCls is lower than the other models. Even when compared to the MOTIFS, a performance decrease can be seen. In SGCls, there is slight improvements compared to the other models. Most important factor is that the performance increase compared to MOTIFS (baseline) is seen. Therefore, the effectiveness of the BGT-Net (no BiGRU) is shown in Table 5 in paper.

In SGDet protocol, the BGT-Net (no BiGRU) can show an impressive performance. It outperforms every other model. The performance difference is also quite significant. The Recall@K is improved by over 1 point and reaches up to an increase of more than 3 points over the next best model. The results in mean Recall@K are worse than others. The short-comings of this BGT-Net (no BiGRU) is removed and improved by changing the model structure to BGT-Net by adding a Bi-GRU.

2. Ablation Study

As mentioned in the paper, we performed ablation study on several factors. We provide additional results on those experiments. They are discussed below:

2.1. Different Combination of Modules

Figure 1, Figure 2, Figure 3, and Figure 4 illustrate the graphical representation of the performance achieved using different modules by evaluating them on different performance recall metric.

2.2. Number of Transformer Heads

Figure 5, Figure 6, Figure 7, and Figure 8 show the graphical representation of the performance achieved by using different number of transformer heads and by varying the performance recall metric.

2.3. Number of Bidirectional GRU Layers

Figure 9, Figure 10, Figure 11, and Figure 12 show the graphical representation of the performance achieved by using different number of BiGRU layers and by varying the performance recall metric.

3. Hyper-parameter Study

The influence of the most important hyper-parameters on the model performance were tested. Batch size, learning rate, and number of solver iterations were varied for multiple experiments.

PredCls was the main protocol on which the performance was compared. One hyper-parameter at a time was varied to understand its influence. The effect of changing parameters and therefore showing which parameter set performs the best can be seen below.

Leraning Rate. Batch size was fixed at 24 and number of solver iterations at 24000. Learning rates 0.0001, 0.0005, 0.001 and 0.002 were tested.

As Fig. 13, shows the best performance learning rate when evaluated using Recall@K. Evaluating on no graph constraint Recall@K illustrates no significant difference by the influence of learning rate.

Similarly, in Fig. 14, the influence of the learning rate for evaluation on zero shot and mean Recall@K can not clearly be seen. Performance of learning rates 0.002, 0.001 and 0.00005 are almost the same. But learning rate 0.002 seems to have a slight edge on the other two. But this difference is marginal. Only the smallest learning rate 0.0001 seems to be under-performing.

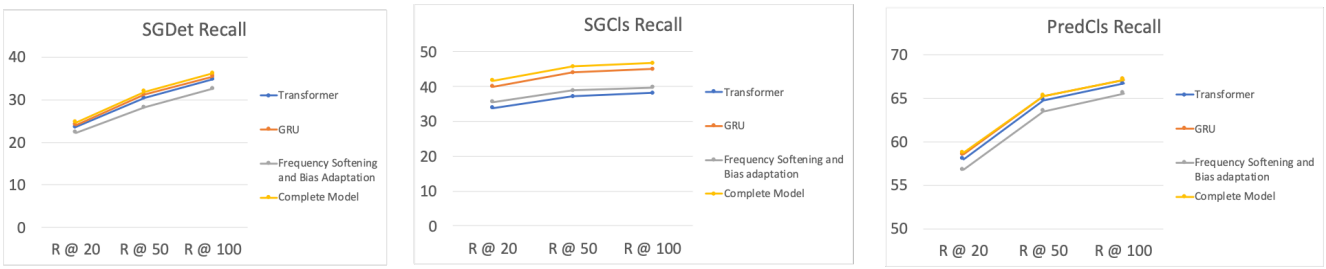

Figure 1: Graphical representation of Recall Results for SGDet (left), SGCls (middle), PredCls (right) comparing the effects of different modules of the BGT-Net made during ablation studies on the effectiveness of different modules.

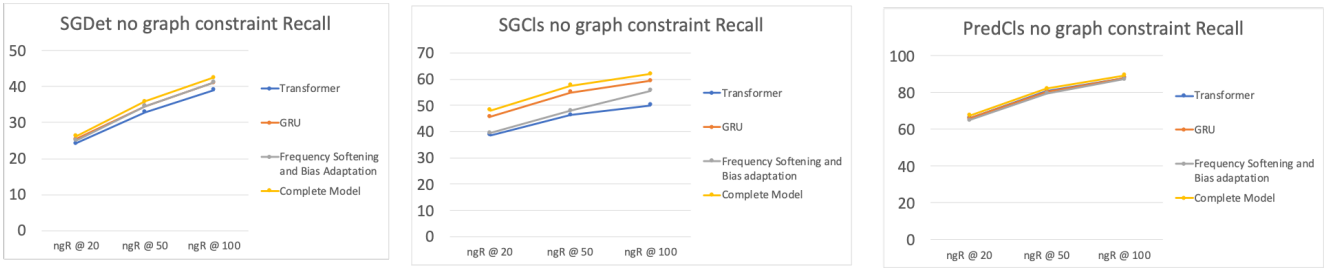

Figure 2: Graphical representation of no graph constraint Recall Results for SGDet (left), SGCls (middle), PredCls (right) comparing the effects of different modules of the BGT-Net made during ablation studies on the effectiveness of different modules.

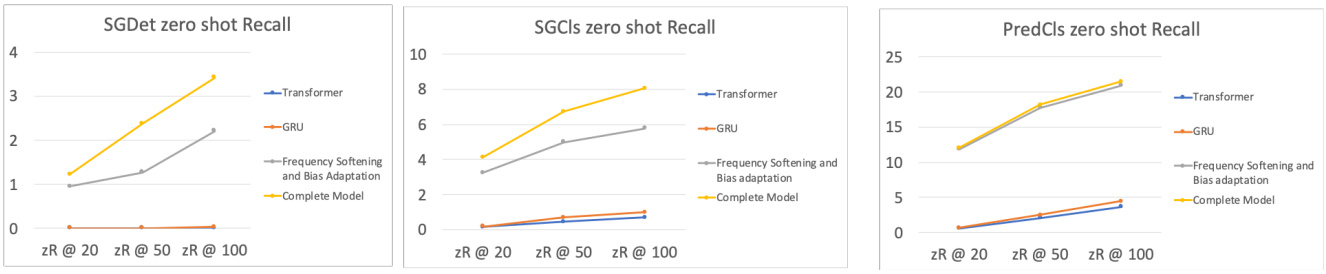

Figure 3: Graphical representation of zero shot Recall Results for SGDet (left), SGCls (middle), PredCls (right) comparing the effects of different modules of the BGT-Net made during ablation studies on the effectiveness of different modules.

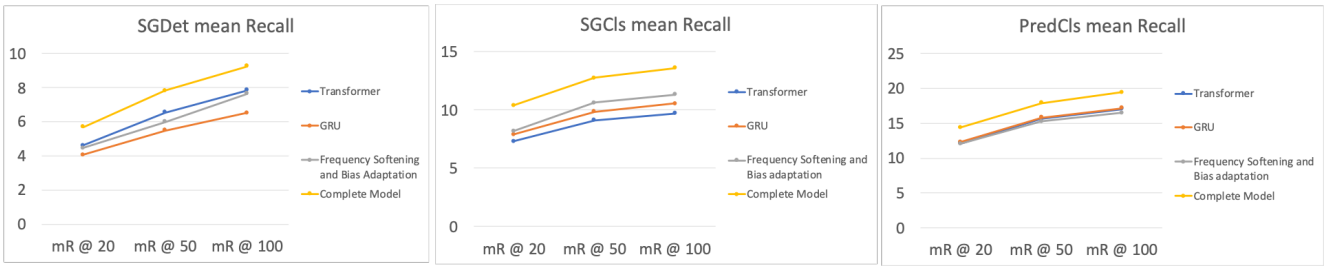

Figure 4: Graphical representation of mean Recall Results for SGDet (left), SGCls (middle), PredCls (right) comparing the effects of different modules of the BGT-Net made during ablation studies on the effectiveness of different modules.

Batch Size. The non-changed hyper-parameters are set to 0.002 for the learning rate and 24000 to solver iterations. The examined batch sizes are 6, 12, 18 and 24.

Only the smallest batch size 6 shows the lower performance in Recall@K as it can be seen in Fig. 15. This difference disappears for the no graph constraint Recall@K (vis-

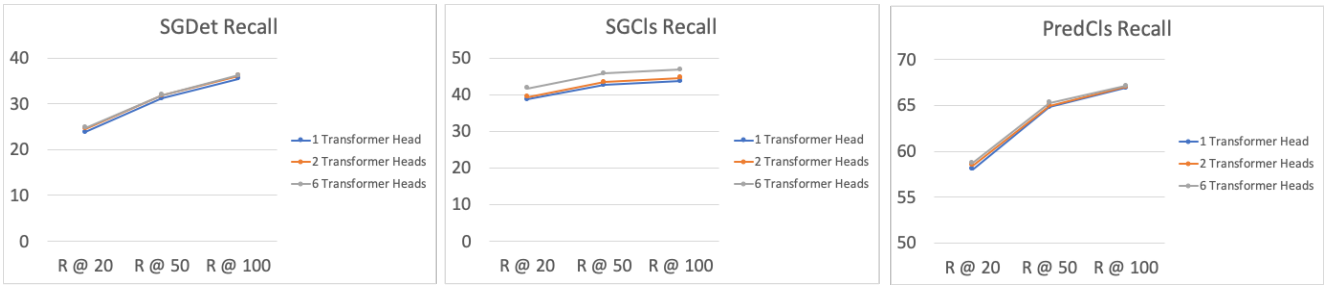

Figure 5: Graphical representation of Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluating the performance changes evoked by changing the number of Transformer heads of Transformer Encoders for object and edge information.

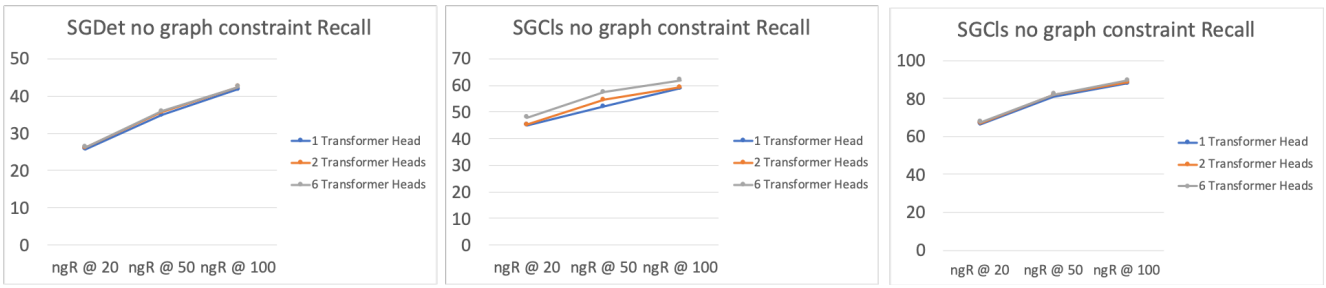

Figure 6: Graphical representation of no graph constraint Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluating the performance changes evoked by changing the number of Transformer heads of Transformer Encoders for object and edge information.

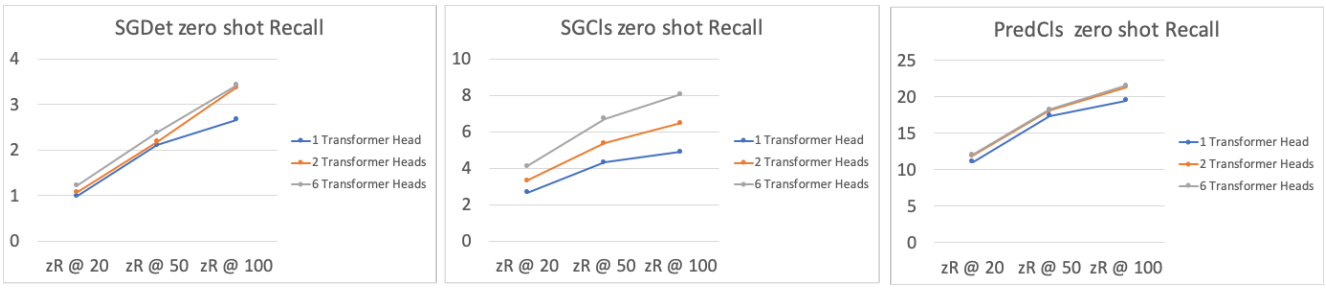

Figure 7: Graphical representation of zero shot Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluating the performance change evoked by changing the number of Transformer heads of Transformer Encoders for object and edge information.

ible in Fig. 15 on the right) and for the zero shot Recall@K (shown in Fig. 16 on the left).

Significant and most evident difference in performance can be seen in the mean Recall@K in Fig. 16. Clearly, the largest batch size is performing better in this metric. Also in the other metrics, batch size 24 has the highest values.

Solver Iterations. Keeping the learning rate at 0.002 and the batch size at 24, while changing the solver iterations to 6000, 12000, 18000 and 24000 shows the following influence of the solver iterations on the model performance.

Throughout all the results in Fig. 17 and Fig. 18, 6000 solver iterations under-perform significantly. Having only 6000 iterations does not allow the model to converge. As before with batch size, the difference in performance while changing the solver iterations is really small. Only for mean Recall@K the highest solver iteration does improve the results by almost 1 point.

After these results, the best performing set of hyper-

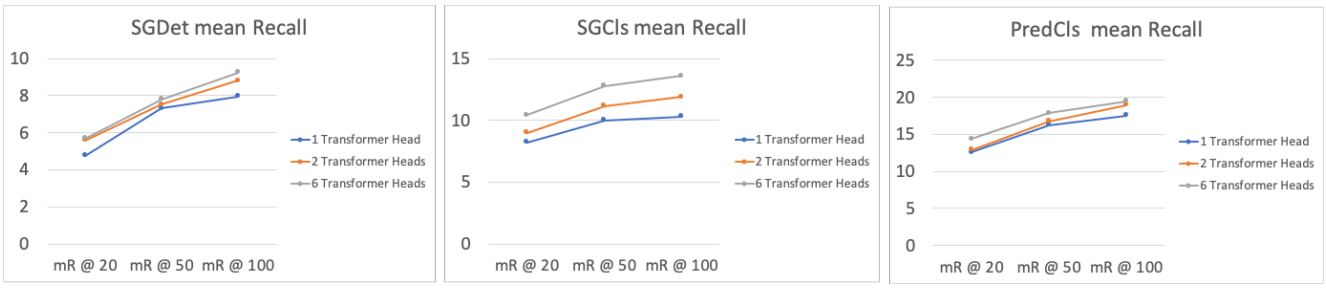

Figure 8: Graphical representation of mean Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluating the performance change evoked by changing the number of Transformer heads of Transformer Encoders for object and edge information.

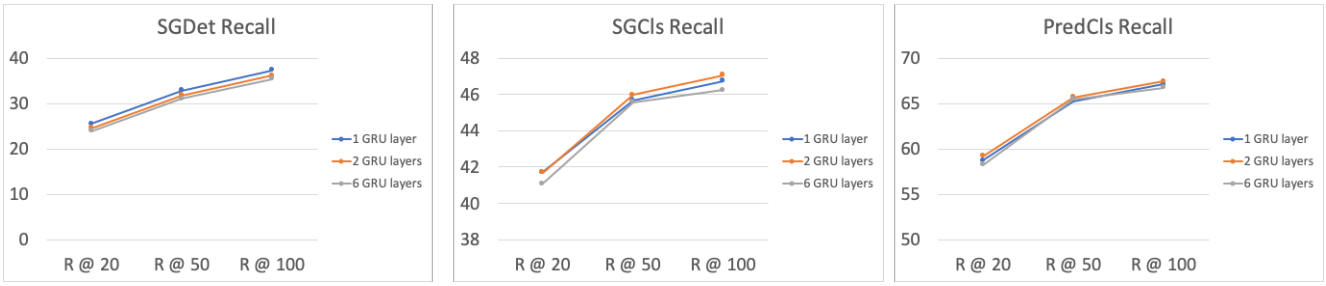

Figure 9: Graphical representation of Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluation of performance change influenced by using 1, 2 or 6 layers of bidirectional GRUs in the BGT-Net model.

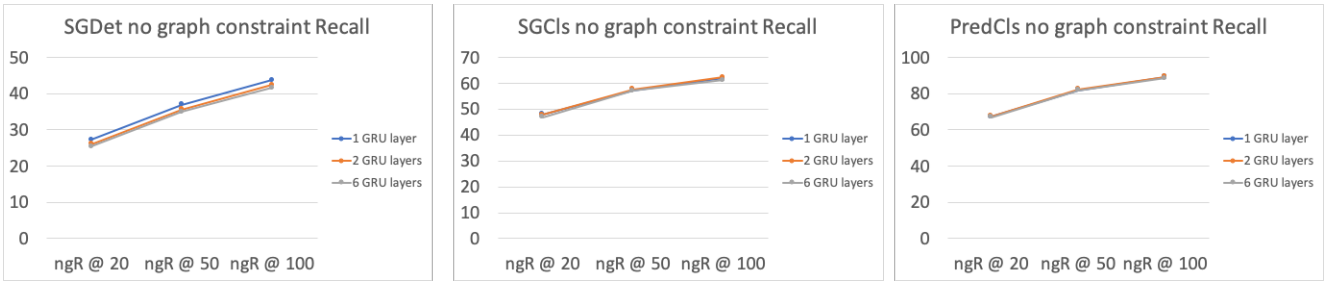

Figure 10: Graphical representation of no graph constraint Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluation of performance change influenced by using 1, 2, or 6 layers of bidirectional GRUs in the BGT-Net model.

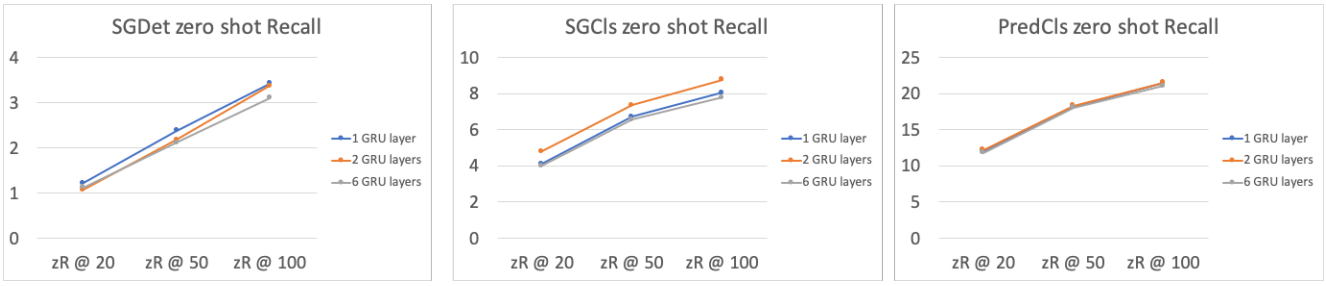

Figure 11: Graphical representation of zero shot Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluation of performance change influenced by using 1, 2, or 6 layers of bidirectional GRUs in the BGT-Net model.

parameters can be found. Combining the results for batch size, learning rate and solver iterations leads to the choice of

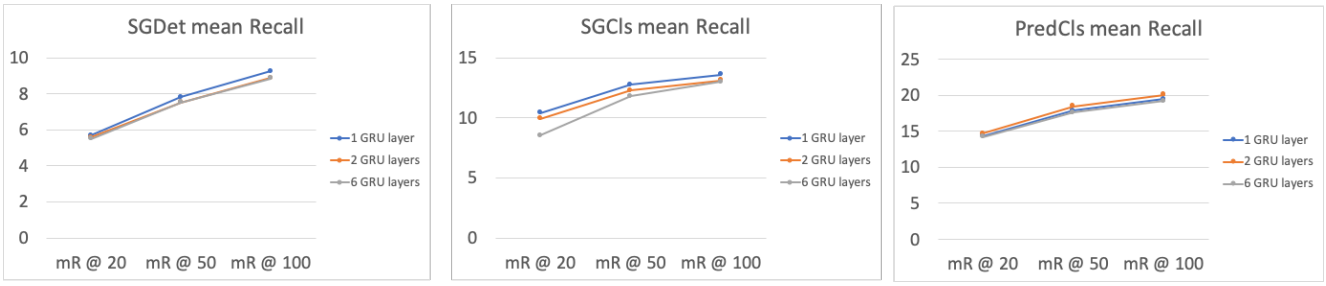

Figure 12: Graphical representation of mean Recall Results for SGDet (left), SGCls (middle), PredCls (right). Evaluation of performance change influenced by using 1, 2, or 6 layers of bidirectional GRUs in the BGT-Net model.

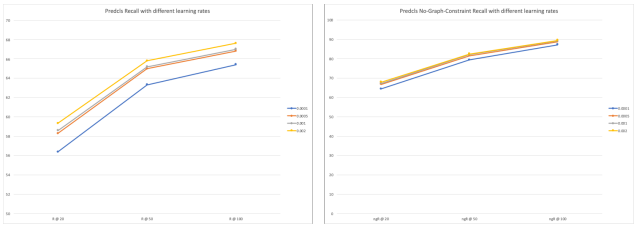

Figure 13: Recall@K (left) and no graph constraint Recall@K (right) for Predicate Classification using different learning rates

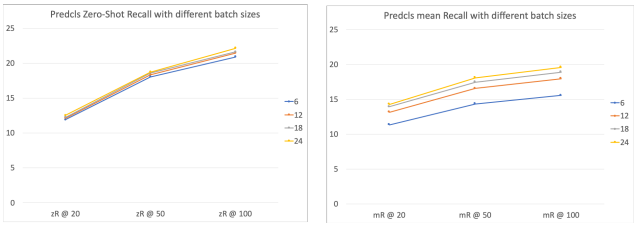

Figure 16: Zero shot Recall@K (left) and mean Recall@K (right) for Predicate Classification using different batch sizes

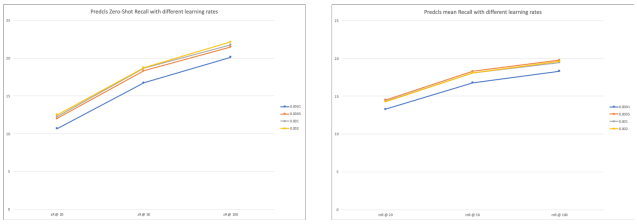

Figure 14: Zero shot Recall@K (left) and mean Recall@K (right) for Predicate Classification using different learning rates

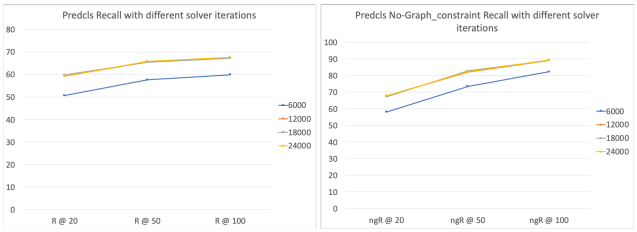

Figure 17: Recall@K (left) and no graph constraint Recall@K (right) for Predicate Classification compared to different solver iterations

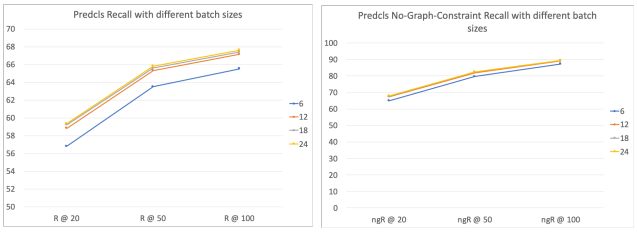

Figure 15: Recall@K (left) and no graph constraint Recall@K (right) for Predicate Classification using different batch sizes

Figure 18: Zero shot Recall@K (left) and mean Recall@K (right) for Predicate Classification compared to different solver iterations

a learning rate of 0.002 a batch size of 24 and a solver iteration of 24000. Possibly, the least important of these would be the solver iterations since only the smallest difference going from 18000 to 24000 was detected.

4. Qualitative Results: BGT-Net

The qualitative results in Figure 19 show generated scene graphs on images of the Visaul Genome dataset. The exam-

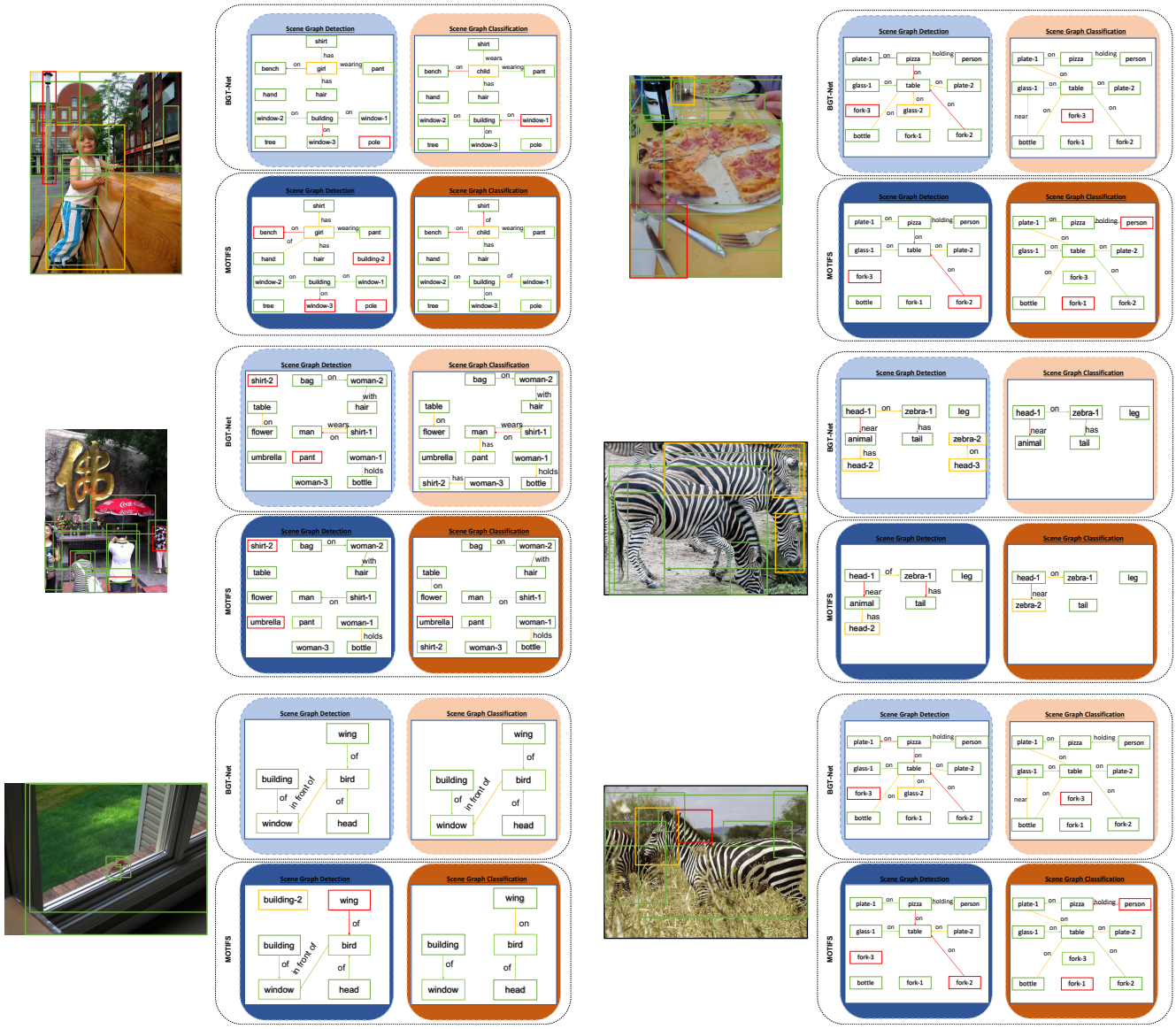

Figure 19: Qualitative results of BGT-Model generated scene graphs. Two protocols are shown. left: Scene Graph Detection (SGDet), right: Scene Graph Classification (SGCls). BGT-Net is qualitatively compared to the MOTIFS model [1]. Three colours are used to specify properties of detections. 'Green' show detections that also perfectly correspond with ground-truth. 'Red' is used for wrong detections and 'orange' for detections not available in ground-truth but when checking with visual scene still represent the situation correctly.

ples are compared to the scene graphs generated by the MOTIFS model [1]. For illustration purpose, the objects and relationships are coloured to represent properties of these detection. Object or relationship coloured "green" are detected properly and correspond to the ground-truth. 'Red' shows wrongly detected entities. For objects this can be either due to incorrect class prediction or due to not detection of the object during detection step with the Faster R-CNN. 'Orange' denotes detections which do not correspond with the ground-truth annotations but can be validated by human inspection which means that the prediction generally corresponds with the visual scene.

In Figure 19, the scene graphs generated with the Scene Graph Detection (SGDet) protocol are shown on the left and the ones generated with the Scene Graph Classification (SGCls) protocol are shown on the right. In SGDet, in many images, a large amount of objects are being detected. To increase readability, the object connected to the ground-truth objects are additionally inserted into the scene graph (with the colour 'orange'). In many images, lots of $<$ subject- object-relationship$>$ triplets get correctly predicted but these cannot be found in the ground-truth triplets.

This can effect the the performance of the model since these maybe correctly detected but not annotated in ground-truth and influence the prediction of other relationships.

Both of the compared models in Figure 19, show errors in their predictions. But while detecting an object wrongly happens quite rarely, the relationship prediction is still much more prone to errors. This can be just the problem of the model but it is much more likely that many of the predictions made are not essentially wrong. This can be illustrated with the example of the object pair 'person' and 'pants'. Most often a person 'wears' their pants. Relationships like 'has', 'in' and 'on' do not contradict the reality. Added difficulty lays in the fact that throughout the dataset, these mentioned relationships do recently also appear. But there cannot be an evidence in the image what the relationship in this case can be because the difference between <person-pants-wears> and <person-pants-has> will not be visible in the image. This leads to worse model performance directly induced by the characteristics of the dataset.

As shown in 19, errors in object prediction in SGCls do not happen very often. This may be the result of the highly improved performance of the BGT-Net in this protocol when measured in all evaluated metrics. Compared to the other protocols, the gained performance in SGCls is proportionally higher.

References

[1] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.