Supplementary Material: NODIS: Neural Ordinary Differential Scene Understanding

Yuren Cong[1], Hanno Ackermann[1], Wentong Liao[1],
Michael Ying Yang[2], and Bodo Rosenhahn[1]

[1] Institute of Information Processing, Leibniz University Hannover, Germany
{cong, ackermann, liao, rosenhahn}@tnt.uni-hanover.de
[2] Scene Understanding Group, University of Twente, The Netherlands
michael.yang@utwente.nl

1 Architecture Details

Feature Extraction The images are normalized and resized to (592·592), then forwarded to the pre-trained Faster-RCNN [2,1]. The output of the Faster R-CNN for each image includes a number of proposal bounding boxes, corresponding $(512 \cdot 7 \cdot 7)$ feature maps and 151-d primary object distributions.

Object Detection For each proposal box, a 200-dimensional semantic embedding vector is obtained by multiplying the 151-d object distribution and the $(151 \cdot 200)$ embedding matrix. The 4-d annotation of the box position is forwarded into a batch normalization layer and a linear layer with ReLU activation to obtain the 128-dimensional position embedding vector. These two vectors are concatenated with the feature vector resulting from compressing the $512 \cdot 7 \cdot 7$ feature maps to size 4096. The concatenated vectors of the proposal boxes are compressed by a linear layer to size 1024, then organized as sequence with random order and forwarded into the O-ODE which uses a single layer bidirectional LSTM as the approximate function in the ODE solver. The output dimension of the O-ODE is the same as the input dimension. Object classification scores are computed by a fully-connected layer.

Predicate Prediction For each object pair, the $512 \cdot 7 \cdot 7$ feature maps of the union box and the $2 \cdot 27 \cdot 27$ spatial masks with value in $[-0.5, 0.5]$ which present the positions of subject and object are computed at first. The spatial masks are forwarded into a convolution layer with 256 kernels of size $7 \cdot 7$, a max pooling layer (kernel size 3, stride 2) and another convolution layer with 512 kernels of size $3 \cdot 3$ so that the output has the same size $512 \cdot 7 \cdot 7$ and can be element-wisely added to the feature map of the union box. To avoid a large number of parameters, the $512 \cdot 7 \cdot 7$ feature maps are compressed with global average pooling to obtain outputs of size 512. Then the features of the subject, object, and union boxes are concatenated as a $(512 \cdot 3)$-dimensional visual vector. Two 200-dimensional semantic embedding vectors are generated by a word2vec

module with the subject and object classes which is provided by the ground-truth in the training or predicted by the Object Classifier in the evaluation and concatenated as a 400-dimensional semantic vector.

The 1536-dimensional visual vectors and the 400-dimensional semantic vectors of the object pairs can be pre-processed in three ways before the P-ODE: *FC-Layer (Fig. 1):* The $(3 \cdot 512)$-dimensional the visual vectors and 400-dimensional semantic vectors are forwarded into two independent fully connection layers that both have 512 neurons. Then, the outputs are concatenated together as 1024-dimensional representation vectors for the P-ODE. *GCNN (Fig. 2):* The visual vectors and semantic vectors are first concatenated. Then, we use a two-layer graph convolutional neural network (GCNN) to infer information about context. The hidden size and output size are both 1024. Since the number of object pairs in each image is variable, we set each element on the diagonal of the adjacency matrix to 0.8. The weight of 0.2 is uniformly distributed among the remaining entries of each row. The output vectors of the GCNN are passed into the P-ODE. *LSTM (Fig. 3):* Similar as for the first variant, the $(3 \cdot 512)$-dimensional visual vectors and the 400-dimensional semantic vectors are fed into two single layer LSTMs, respectively. Both of them have the output dimension 512. We concatenate the two outputs for the P-ODE.
The P-ODE also uses a single layer bidirectional LSTM with 512 hidden size as the approximate function in the ODE solver. The final predicate distributions are computed by two linear layers with ReLU activation.

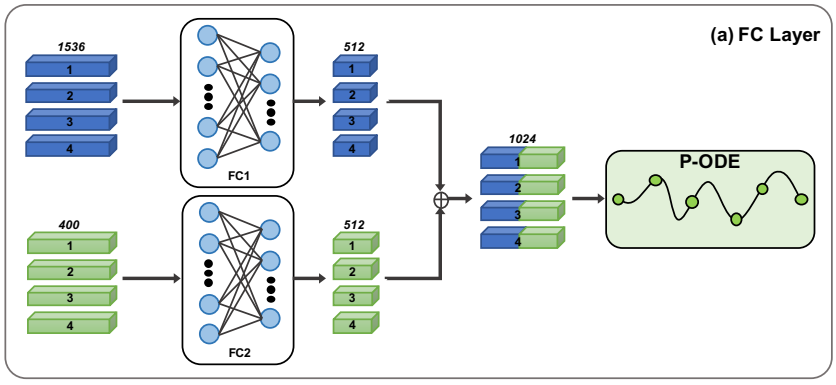

Fig. 1. Pre-Processing with fully connected layer. Blue boxes represent the visual vectors, green ones the semantic vectors and the numbers above the lengths of the input vectors.

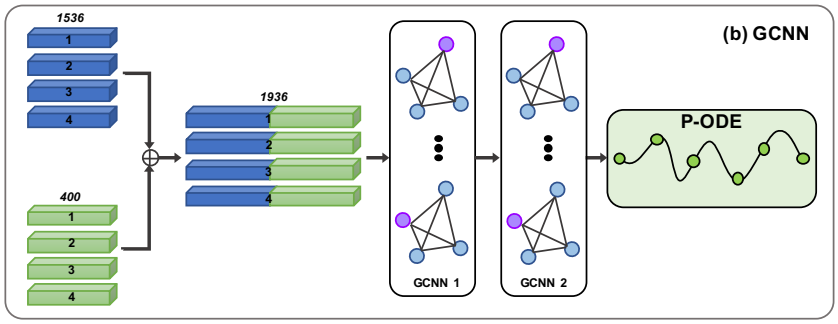

Fig. 2. Pre-Processing with a graph convolutional neural network (GCNN). Blue boxes represent the visual vectors, green ones the semantic vectors and the numbers above the input lengths.

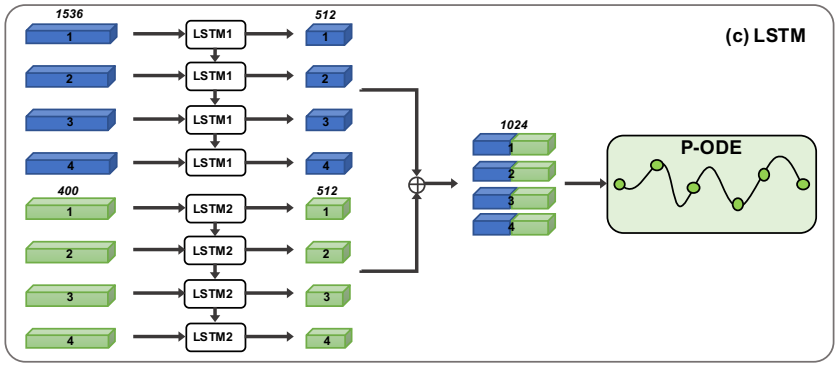

Fig. 3. Pre-Processing with an LSTM. Blue boxes represent the visual vectors, green ones the semantic vectors and the numbers above the input lengths.

2 Object Classification by Linear Program

We are interesting in assessing the performance of the object classifier proposed in Eq. (1). For faster calculations, we simplify Eq. (1) to

$$\max \sum_{u \in U_{obj}} \sum_{l \in \mathcal{L}_{obj}} \alpha_{u,l} x_{u,l} \tag{9a}$$

$$s.t. \quad x_{u,l} \in [0,1], \quad \sum_{l \in \mathcal{L}_{obj}} x_{u,l} \leq 1. \tag{9b}$$

We thus relax the Integer Linear Problem to a Linear Program (LP) which is much easier to solve than the ILP.

Regarding the scores $\alpha_{u,l}$, we select the first 20 occurrences of each object class in a training epoch (with random selection of the mini-batches). For each object class, we then created dictionaries $D_l \in \mathbb{R}^{1024 \times 20}$, $l \in \mathcal{L}_{obj}$, of the fea-

ture representations that are used as input for the following ODE layer. During validation, we compute the scores for a given sample s_u by

$$\alpha_{u,l} = \sum_{k=1}^{20} D_{l,k}^{\top} \cdot s_u. \tag{10}$$

Motivated by the recall-at-50, we further add the constraint that

$$\sum_{u \in U_{obj}} \sum_{l \in \mathcal{L}_{obj}} x_{u,l} \leq 50, \tag{11}$$

i.e. at maximum 50 objects can be detected in an image.

Using a feature generating network pre-trained by the proposed ODE-layers, we noticed that the continuous variables $x_{u,l}$ usually obtain values that are very close to either zero or one. That indicates that the assumption of using dictionaries is reasonable. That also justifies the hypothesis that the gradient generated by the ODE layer forces the feature network to embed samples of the same class coherently.

The computational complexity of solving even Eqs. 9 is very large. Using an interior-point solver with the maximal number of iterations set to 20, a single solution requires between several seconds and 30 minutes. In comparison, a single forward pass through the ODE layer is negligibly small (< 0.1 seconds). Surprisingly, the average recall-at-50 after 100 images (about 0.3% of the validation data) was comparable to the one obtained by using the proposed architecture. ODE layer followed by linear layers to compute the class scores. We stopped the computation after 4 hours. Using a non-trained network, we could not obtain a

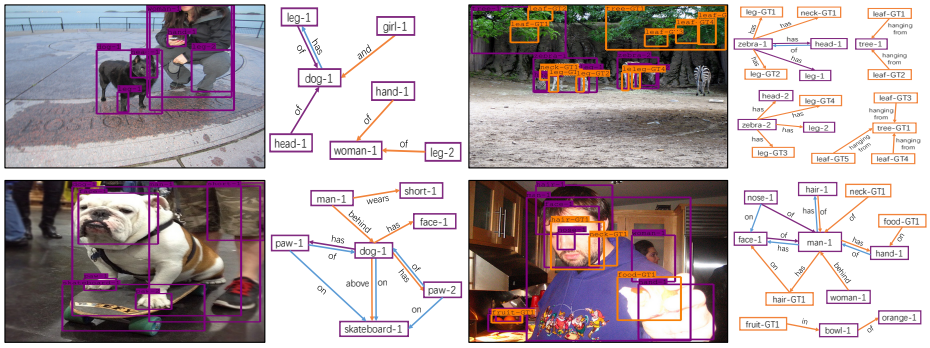

Fig. 4. Qualitative results from our model in the scene graph generation setting. Purple boxes denote correctly detected objects while orange boxes denote ground truth objects that are missing. Purple edges correspond to correctly classified relationships at the Re@20 while orange edges denote ground truth relationships that are not detected. Blue edges denote detected relationships that do not exist in ground truth annotations (false positives). Note that sometimes blue edges are also semantically correct.

good score using the model defined by Eqs. (9). This clearly demonstrates that end-to-end training is essential for this task. In other words, it justifies the use of the ODE layer for scene graph estimation.

3 Additional Qualitative Results

Fig. 4 shows the additional qualitative results for scene graph generation from Visual Genome dataset. Purple boxes denote correctly detected objects while orange boxes denote ground truth objects that are not detected. Purple edges correspond to correctly classified relationships at the R@20 setting while orange edges denote ground truth relationships that are not detected. Blue edges denote detected relationships that do not exist in ground truth annotations (false positives).

References

1. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Neural Information Processing Systems (NeurIPS). pp. 91–99 (2015) 1
2. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5831–5840 (2018) 1