Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs (SUPPLEMENTARY)

Martin Simonovsky
Université Paris Est, École des Ponts ParisTech
martin.simonovsky@enpc.fr

Nikos Komodakis
Université Paris Est, École des Ponts ParisTech
nikos.komodakis@enpc.fr

In this supplementary material, we provide details on the graph classification task (Section 1), on choice of edge labeling for point clouds (Section 2), and on robustness of point cloud classification to noise (Section 3).

1. Details on Graph Classification Benchmark

In this section we describe the differences in our network architecture to the one introduced for NCI1 in the main paper and discuss evaluation results for each dataset in detail.

NCI1. ECC (83.80%) performs distinctly better than convolution methods that are not able to use edge labels (DCNN [1] 62.61%, PSCN [3] 78.59%). Methods not approaching the problem as convolutions on graphs but rather combining deep learning with other techniques are stronger (Deep WL [5] 80.31%, structure2vec [2] 83.72%) but are still outperformed by ECC. While the Weisfeiler-Lehman graph kernel remains the strongest method (WL [4] 84.55%), it is fair to conclude that ECC, structure2vec, and WL perform at the same level.

NCI109. We use the same ECC-network configuration and training details as described in Section 4.3 for NCI1, since both datasets are similar. ECC (82.14%) performs distinctly better than DCNN [1] (62.86%), which is not able to use edge labels, and is on par with non-convolutional approaches (Deep WL [5] 80.32%, structure2vec [2] 82.16%, WL [4] 84.49%).

MUTAG. As MUTAG is a tiny dataset of small graphs, we trained a downsized ECC-network to combat overfitting. Using the notation from Section 4.3, its configuration is C(16)-C(32)-C(48)-MP-C(64)-MP-GAP-FC(64)-D(0.2)-FC(2), all other details are as with NCI1. While by numbers ECC (89.44%) outperforms all other approaches except of PSCN [3] (92.63%), we note that all four leading methods (Deep WL [5] 87.44%, structure2vec [2] 88.28%, ECC, PSCN) can be seen to perform equally well due to fluctuations caused by the dataset size. We account the tiny decrease in performance with test-time randomization (88.33%) to the same reason.

ENZYMES. Due to higher complexity of this task we use a wider ECC-network configured as C(64)-C(64)-C(96)-MP-C(96)-C(128)-MP-C(128)-C(160)-MP-C(160)-GAP-FC(192)-D(0.2)-FC(6) using the notation and other details in Section 4.3. As this dataset is not edge-labeled, we do not expect to obtain the best performance. Indeed, our method (53.50%) performs at the level of Deep WL [5] (53.43%) and is overperformed by WL [4] (59.05%) and structure2vec [2] (61.10%). Note that the gap to the other convolution-based method DCNN [1] (18.10%) is huge and there is an improvement of more than 4 percentage points due to edge labels in coarser graph resolutions from Kron reduction.

D&D. Due to large graphs in this dataset we designed a ECC-network with more pooling configured as C(48)-C(48)-C(48)-MP-C(48)-MP-C(64)-MP-C(64)-MP-C(64)-MP-C(64)-MP-GAP-FC(64)-D(0.2)-FC(2) using the notation and other details in Section 4.3. As this dataset is not edge-labeled, we do not expect to obtain the best performance. Our method (74.10%) is overperformed by the others who evaluated on this dataset (PSCN [3] 77.12%, WL [4] 79.78%, structure2vec [2] 82.22%), though the margin is not very large.

2. Edge Labels for Point Clouds

In Section 3.4 we defined edge labels $L(j,i)$ as the offset $\delta = p_j - p_i$ in Cartesian and spherical coordinates, $L(j,i) = (\delta_x, \delta_y, \delta_z, ||\delta||, \arccos \delta_z/||\delta||, \arctan \delta_y/\delta_x)$. Here, we explore the importance of individual elements in the proposed edge labeling and further evaluate labels invariant to rotation about objects' vertical axis z (IRz). Table 1 conveys that models with isotropic (60.7) or no labels (38.9) perform poorly as expected, while either of the coordinate systems is important. IRz labeling performs comparably or even slightly better than our proposed one. How-

Label $L(j,i)$	Mean F1
$(\delta_x, \delta_y, \delta_z, \|\|\delta\|\|, \arccos \delta_z/\|\|\delta\|\|, \arctan \delta_y/\delta_x)$	78.4
$(\delta_x, \delta_y, \delta_z)$	76.1
$(\|\|\delta\|\|, \arccos \delta_z/\|\|\delta\|\|, \arctan \delta_y/\delta_x)$	77.3
$(\|\|\delta_{xy}\|\|, \delta_z, \|\|\delta\|\|, \arccos \delta_z/\|\|\delta\|\|)$	75.8
$(\|\|\delta_{xy}\|\|, \delta_z)$	78.2
$(\|\|\delta\|\|, \arccos \delta_z/\|\|\delta\|\|)$	78.7
$(\|\|\delta\|\|)$	60.7
(0)	38.9

Table 1. ECC on Sydney with varied edge label definition.

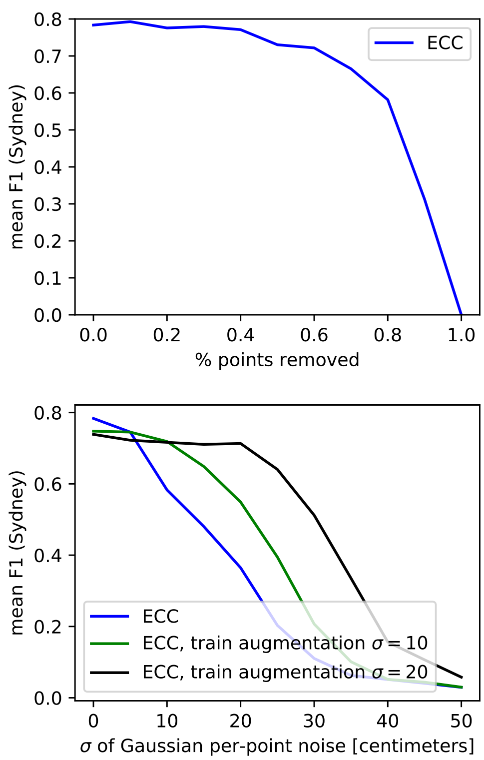

Figure 1. Robustness to point removal and Gaussian noise.

ever, we believe this a property of the specific dataset and may not necessarily generalize, an example being MNIST, where IRz is equivalent to full isotropy and decreases accuracy to 89.9%.

3. Robustness to Noise

Real-world point clouds contain several kinds of artifacts, such as holes due to occlusions and Gaussian noise due to measurement uncertainty. Figure 1 shows that ECC is highly robust to point removal and can be made robust to additive Gaussian noise by a proper training data augmentation.

References

[1] J. Atwood and D. Towsley. Diffusion-convolutional neural networks. In *NIPS*, 2016. 1

[2] H. Dai, B. Dai, and L. Song. Discriminative embeddings of latent variable models for structured data. In *ICML*, 2016. 1

[3] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, 2016. 1

[4] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011. 1

[5] P. Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *SIGKDD*, 2015. 1