Classifying Signals on Irregular Domains via Convolutional Cluster Pooling: Supplementary Material

Angelo Porrello **Davide Abati** **Simone Calderara** **Rita Cucchiara**

University of Modena and Reggio Emilia, Modena, Italy
name.surname@unimore.it

1 Pseudocode

This section introduces algorithms involved in the proposed CCP layer discussed in Section 4 of the paper. Specifically, Algorithm 1 illustrates the cluster step (Figure 2, left in the main paper), whereas Algorithm 2 details the filter step (Figure 2, center in the main paper)

2 Computational Complexity

Given an input affinity matrix $\mathcal{A}^{\mathcal{K}_m}$ and the desired number of output clusters $|\mathcal{K}_{m+1}|$, the cluster step has complexity equivalent to $\mathcal{O}(|\mathcal{K}_m|^2|\mathcal{K}_{m+1}| + |\mathcal{K}_{m+1}|^2|\mathcal{K}_{m+1}|)$, while the filter step has complexity $\mathcal{O}(|\mathcal{K}_m|^2|\mathcal{K}_{m+1}|)$. This analysis has been done without taking into account various possible optimizations, regarding for example multiplications in presence of sparse matrices. Moreover, the fastest schema consists in constructing the cluster hierarchy before the training process begins, then caching all intermediate $\mathcal{A}^{\mathcal{K}_m}$ and $\overrightarrow{\mathcal{N}}^{(m+1)}$. This way, the computational complexity for both steps is reduced to $\mathcal{O}(|\mathcal{K}_{m+1}| \times L \times d_{IN} \times d_{OUT})$. Such complexity constitutes an improvement w.r.t. the one deriving from Chebyshev [1] $\mathcal{O}(\mathcal{D}_{AVG} \times |\mathcal{A}_m| \times L \times d_{IN} \times d_{OUT})$ (where we indicate with $\mathcal{D}_{AVG}$ the average degree in $\mathcal{A}_m$), since $\mathcal{D}_{AVG} \times |\mathcal{A}_m| \approx |\mathcal{E}| > |\mathcal{A}_m| > |\mathcal{K}_{m+1}|$. Differently, the same conclusion cannot be easily achieved comparing our solution to GCN [2], since its cost is $\mathcal{O}(|\mathcal{A}_m| \times d_{IN} \times (d_{OUT} + \mathcal{D}_{AVG}))$.

3 Model Analysis

Impact of the input graph. Regarding our proposal, how important is the quality and the integrity of the underlying graph? In terms of capabilities generalisation, what happens if we keep the signals unchanged and train our architecture on a random graph? Aiming to answer such questions, we conduct experiments replacing the designed shared graph (e.g. for CIFAR-10 the graph representing the 8-connectivity between pixels) with a random one, guaranteed to be connected and characterised by the same number of nodes and edges. As shown in Tab. 1, we experience a considerable drop in performance when employing random connections between nodes (especially in the euclidean domain), consistently with what observed in [1]. Trivially, for graph convolutional neural networks, the compliance of the input affinity matrix with the domain-specific intrinsic bonds constitutes a crucial term for extracting meaningful features and, consequently, obtaining good level of accuracy on new and unseen data. In this respect, two ways may be investigated to improve such kind of approaches. On the one hand, drawing from the design principles underpinning kernel methods, many efforts may be put in designing better affinity measures between nodes. On the other hand, future works should move in a different direction, in which the affinity matrix is learned directly from the data, in a semi-supervised or completely unsupervised manner.

Adaptation to heterogeneous graphs. Despite our model being originally conceived to assess a slightly different problem, we conjecture the possibility to extend it for heterogeneous graph classification, in which each example features a different affinity matrix. Indeed, a potential solution would be replacing the U matrix in Eq. 7 (in the main paper) with the output of an auxiliary graph convolutional module, responsible for cluster memberships computation (given node features and the affinity matrix). As a consequence,

Table 1: Impact of different graph's definitions on CIFAR-10 and Cross Subject NTU RGB+D, in terms of test set accuracy.

Graph	CIFAR	NTU-CS
Random	69.0	76.6
Hand-crafted	84.4	80.1

Algorithm 1 Cluster Step

```
Input: affinity matrix $\mathcal{A}^{\mathcal{K}_m}$, number of output clusters $|\mathcal{K}_{m+1}|$
Output: affinity matrix $\mathcal{A}^{\mathcal{K}_{m+1}}$
if init then
  $U^{(m+1)} \leftarrow \text{random}(|\mathcal{K}_m|, |\mathcal{K}_{m+1}|)$
$K^{(m+1)} \leftarrow \text{rowsoftmax}(U^{(m+1)})$
$\mathcal{A}^{\mathcal{K}_{m+1}} \leftarrow K^{(m+1)\mathrm{T}}(\mathcal{A}^{\mathcal{K}_m} - I_{\mathcal{N}} \odot \mathcal{A}^{\mathcal{K}_m})K^{(m+1)}$
$D \leftarrow \mathcal{A}^{\mathcal{K}_{m+1}} \mathbf{1}_{|\mathcal{K}_{m+1}|}$
$\overline{\mathcal{A}}^{\mathcal{K}_{m+1}} \leftarrow D^{-\frac{1}{2}} \mathcal{A}^{\mathcal{K}_{m+1}} D^{-\frac{1}{2}}$
Return: $\overline{\mathcal{A}}^{\mathcal{K}_{m+1}}$
```

Algorithm 2 Filter Step

```
Input: normalized affinity matrix $\overline{\mathcal{A}}^{\mathcal{K}_m}$, input feature maps $\mathcal{F}^{(m)} \in \mathbb{R}^{|\mathcal{K}_m| \times d_{IN}}$, normalized and reduced affinity matrix $\overline{\mathcal{A}}^{\mathcal{K}_{m+1}}$, number of output channels $d_{OUT}$, filter size $L$
Output:, output feature maps $\mathcal{F}^{(m+1)} \in \mathbb{R}^{|\mathcal{K}_{m+1}| \times d_{OUT}}$

if init then
  $W, b \leftarrow random(L, d_{IN}, d_{OUT}),\ random(C_{OUT})$
  $\alpha, \beta \leftarrow \alpha \sim \mathcal{N}(\mu = 1,\ \sigma_1^2),\ \beta \sim \mathcal{N}(\mu = 0,\ \sigma_2^2)$

for $k = 1, 2, \ldots, |\mathcal{K}_{m+1}|$ do
  Given $\overline{\mathcal{A}}^{\mathcal{K}_m}$, select top $L$ score points for cluster $\mathcal{K}_k^{(m+1)}$
    ▷ $\phi \leftarrow (\phi(1), \phi(2), \ldots, \phi(L)) \in \mathcal{P}_L^{\{1,2\ldots,|\mathcal{K}_m|\}}$
    ▷ $Rank\ (\mathcal{V}_{\phi(1)}^{(m)} \to \mathcal{K}_k^{(m+1)}) \geqslant \cdots \geqslant Rank\ (\mathcal{V}_{\phi(L)}^{(m)} \to \mathcal{K}_k^{(m+1)})$
    ▷ $\overrightarrow{\mathcal{N}}(l, i) \leftarrow \mathcal{F}_{\phi(l),i}^{(m)} \quad l = 1, 2, \ldots, L \quad i = 1, 2, \ldots, d_{IN}$ ▷ where $\overrightarrow{\mathcal{N}} := \overrightarrow{\mathcal{N}}_k^{(m+1)}$
  Compute gates' activations $\sigma : \mathbb{R} \to (0, 1)$ on top scores
    ▷ $\sigma_{k,l} = \sigma(\alpha\ Rank\ (\mathcal{V}_{\phi(l)}^{(m)} \to \mathcal{K}_k^{(m+1)}) + \beta) \quad l = 1, 2, \ldots, L$
  for $j = 1, 2, \ldots, d_{OUT}$ do
    $\mathcal{F}_{k,j}^{(m+1)} = \sum_{i=1}^{d_{IN}} \sum_{l=1}^{L} W_{l,i,j}\ (\sigma_{k,l} \cdot \overrightarrow{\mathcal{N}}(l, i)) + b_j$
Return: $\mathcal{F}^{(m+1)}$
```

affinities would completely depend on learned vertex representations.

4 Limitations

Since the computational complexity is approximately quadratic with the number of nodes, it is difficult to scale our method to very large graphs (e.g. $> 10^5$ nodes), in terms of both time complexity and memory footprint. For this reason, future works should investigate strategies that avoid expensive computations implied by manipulations of the affinity matrix. Further, several experiments did not show a considerable benefit (in terms of accuracy improvements) from learning the cluster hierarchy through the use of information coming from the task supervision. This aspect, which may be due to some sort of lack in the current implementation, lead the authors to two observations. Firstly, since labels seem not to encourage preferable directions in the clusters' loss landscape, we observed slight improvements in a fully end-to-end training, with respect to a two-step optimization of the cluster hierarchy ($\mathcal{L}_K$) and the feature extractors ($\mathcal{L}_0$). Secondly, future studies should look into this matter in greater depth, trying to understand for which kind of problems a learnable routing on the underlying graph could provide considerable improvements against traditional architectures.

References

[1] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Information Processing Systems*, 2016.

[2] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.