Dynamic Edge-Conditioned Filters in CNNs on Graphs

Martin Simonovsky, Nikos Komodakis

École des Ponts ParisTech
Université Paris-Est

www.github.com/mys007/ecc
martin.simonovsky@enpc.fr

Fork me on GitHub

Motivation

CNNs are massively popular in tasks where the underlying data representation has a grid structure. But in many other tasks, graph-structured data is common:

meshes & point clouds, knowledge bases, social graphs, chemical compounds, …

How to generalize CNNs to graph domains
How to organize weight-sharing in convolutions ?

Observation: Graphs may carry a lot of additional information in edge attributes:

scalar weights, relation types, mutual offset or overlap information, …

We propose a **convolution-like operation on graphs** which

- can exploit edge attributes in any form processable by a neural network
- can be used on datasets with varying graph structures (spatial formulation)
- can be shown to generalize the regular convolution on grids

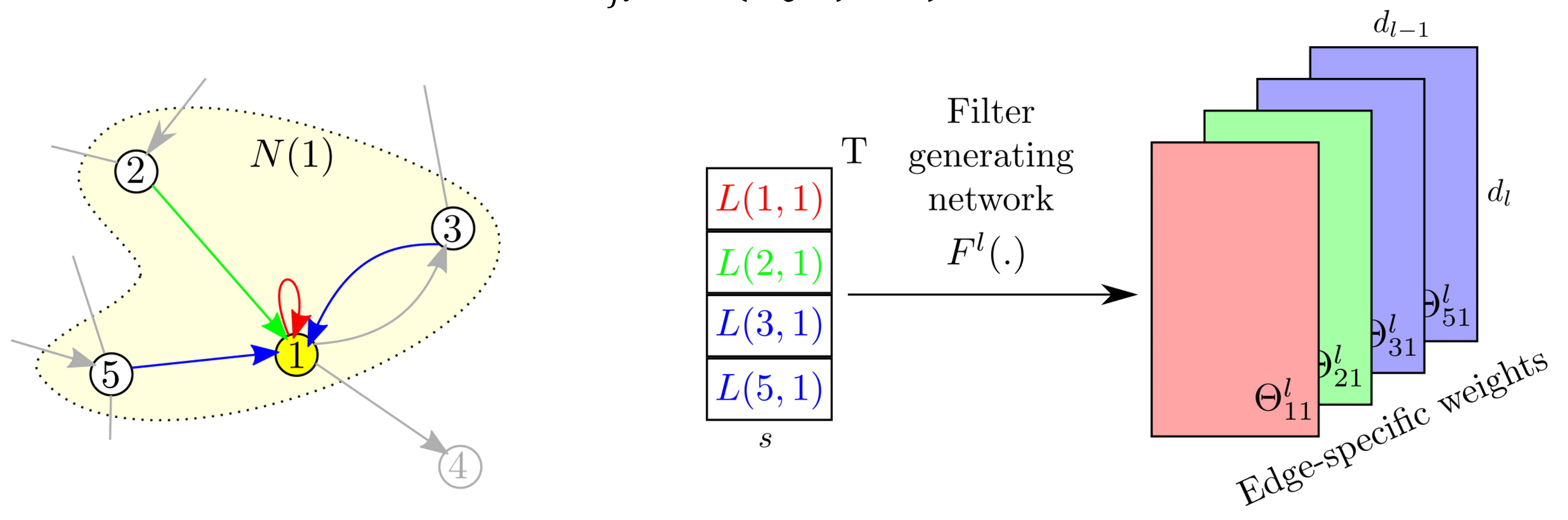
ECC - Edge-Conditioned Convolution

- **Notation**: Directed graph $G = (V, E)$ with edge attributes (labels) $L\colon E \rightarrow \mathbb{R}^s$ and vertex signals (features) $X^l\colon V \rightarrow \mathbb{R}^{d_l}$ on network layer l.
- **Convolution**: Weighted sum of signals over a neighbourhood $N(i) = \{j; (j,i) \in E\} \cup \{i\}$:

$$X^l(i) = \frac{1}{|N(i)|} \sum_{j \in N(i)} \Theta^l_{ji} X^{l-1}(j) + b^l$$

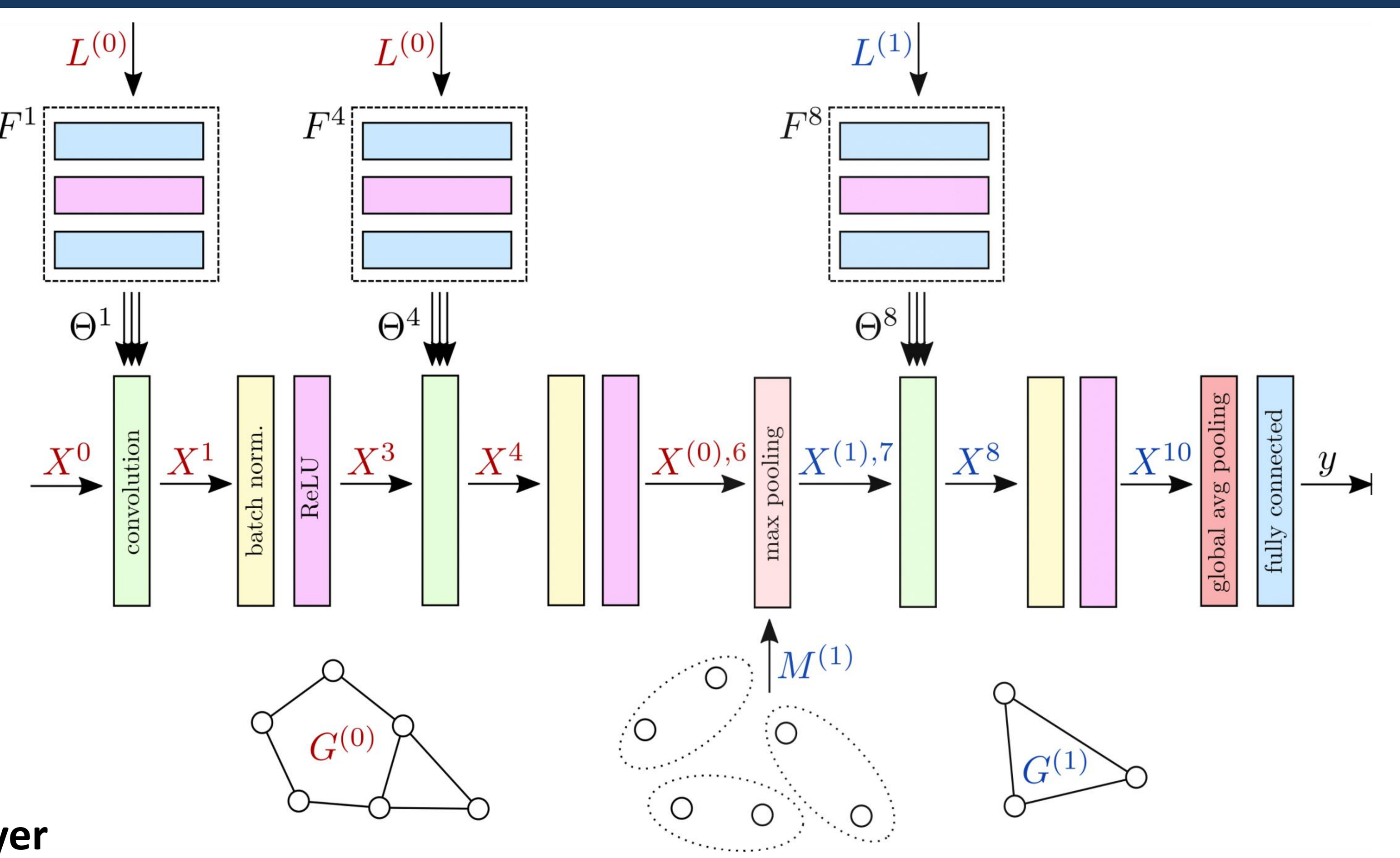
- **Key idea**: Filters Θ^l conditioned on the respective edge labels L and computed dynamically using filter-generating network [1] $F^l\colon \mathbb{R}^s \rightarrow \mathbb{R}^{d_l \times d_{l-1}}$:

$$\Theta^l_{ji} = F^l(L(j,i);\ w^l)$$



- Learned parameters b^l and w^l vs. generated convolution filters Θ^l
- **Complexity**: $\leq |E|$ evaluations of F, $|V| + |E|$ matrix-vector multiplications

Network Architecture



Pooling layer

- Signal on $G = G^{(0)}$ aggregated onto the vertices of a new, coarsened graph $G^{(1)}$.
- Coarsening is problem-specific: merging/subsampling of vertices, mapping $M^{(1)}\colon V^{(0)} \rightarrow V^{(1)}$, creating $E^{(1)}$ and $L^{(1)}$ (reduction).

Point Cloud Classification

- Directed graph construction from point cloud P: edges connect nearby points
- $L(j,i) = (\delta_x, \delta_y, \delta_z, ||\delta||, \arccos\delta_z/||\delta||, \arctan\delta_y/\delta_x)$ for $\delta = p_j - p_i$

- **Coarsening**: graphs built from point cloud pyramid created with VoxelGrid [2] downsampling, M assigns points to the nearest centroid.
- **Augmentation**: rotation about up-axis, jitter scale, mirroring, point dropout
- **Architecture** (Sydney): C(16)-C(32)-MP(0.25,0.5)-C(32)-C(32)-MP(0.75,1.5)-C(64)-MP(1.5,1.5)-GAP-FC(64)-D(0.2)-FC(14) with F^l as FC(16)-FC(32)-FC($d_l d_{l-1}$)

		Sydney Urban (mean F1)	ModelNet10 (mean class acc.)	ModelNet40 (mean class acc.)
3DShapeNets [5]	volume	–	83.5	77.3
VoxNet [6]	volume	73.0	92	83
ORION [7]	volume	77.8	93.8	–
MVCNN [8]	images	–	–	90.1
PointNet [9]	set	–	–	86.2
ECC		78.4	89.3	82.4
ECC (12 votes)		–	90.0	83.2
ECC ($\|\|\delta\|\|$)		60.7	–	–
ECC ($\|\|\delta\|\|$, $\arccos\delta_z/\|\|\delta\|\|$)		78.7	–	–

Graph Classification

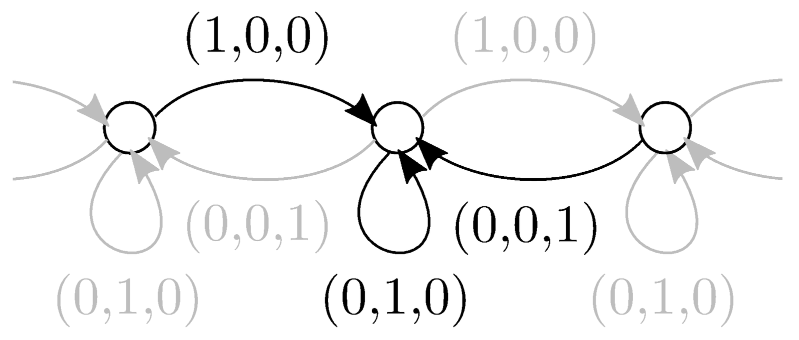
- **Coarsening** [3]: vertices halved by the sign of the largest eigenvector of Laplacian, followed by Kron reduction with spectral sparsification of edges.
- **Augmentation**: randomized sparsification (fractional pooling [4] analogy)
- **Configuration** (NCI1): C(48)-C(48)-C(48)-MP-C(48)-C(64)-MP-C(64)-GAP-FC(64)-D(0.1)-FC(2) with F^l as FC(64)-FC($d_l d_{l-1}$)

		NCI1	NCI109	MUTAG	ENZYMES	D&D
DCNN [10]	convolution	62.61	62.86	66.98	18.10	–
PSCN [11]	convolution	78.59	–	92.63	–	77.12
Deep WL [12]	kernels	80.31	80.32	87.44	53.43	–
Struct2vec [13]	rnd fields	83.72	82.16	88.28	61.10	82.22
WL [14]	kernels	84.55	84.49	83.78	59.05	79.78
ECC		83.80	81.87	89.44	50.00	73.65
ECC (5 scores)		83.80	82.14	88.33	52.67	74.10
ECC (no edge labels)		76.82	75.03	76.11	–	–
ECC (w/ $\sqrt{\deg(i)}$ edge labels)		83.58	82.28	86.67	55.00	75.79

Generalization of Convolution on Grids

Demonstration in 1D

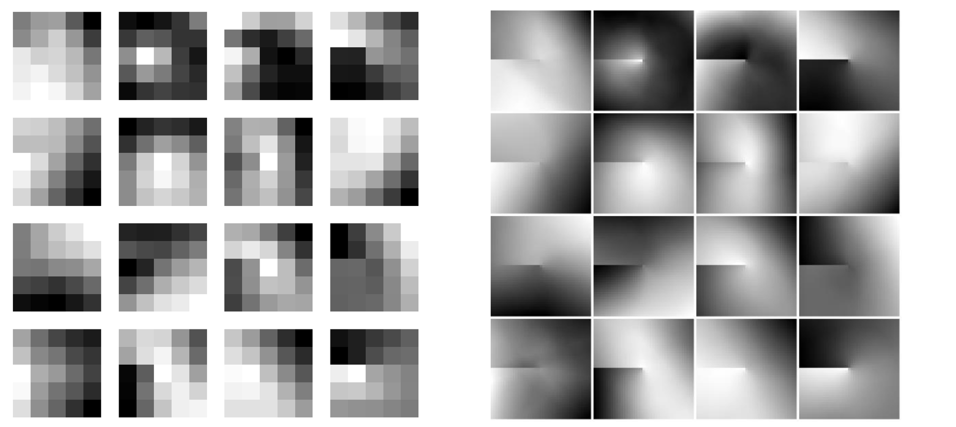
- Graph representation of convolution with a centered filter of size $s = 3$:



- Linear filter-generating network: $F^l(e_k; W) = We_k = W_{*k} \coloneqq W(k)$
- $X^l(i) \sim \sum_{j \in N(i)} F^l(L(j,i); W) X^{l-1}(j) = \sum_{k \in \{-1..1\}} W(k+2) X^{l-1}(i-k)$

MNIST as point cloud

	Train acc.	Test acc.
Full point cloud	99.12	99.14
Sparse point cloud	99.36	99.14
Baseline/one-hot	99.53	99.37



Sampled first layer filters

[1] Brabandere et al., Dynamic filter networks. In NIPS, 2016 [2] Rusu and Cousins. 3d is here: Point cloud library (pcl). In ICRA, 2011 [3] Shuman et al. A multiscale pyramid transform for graph signals. In TSP, 2016 [4] Graham. Fractional max-pooling. 2014 [5] Wu et al. [6] Maturana and Scherer. [7] Alvar et al. [8] Su el al. [9] Qi et al. [10] Atwood and Towsley. [11] Niepert et al. [12] Yanardag and Vishwanathan. [13] Dai et al. [14] Shervashidze et al.