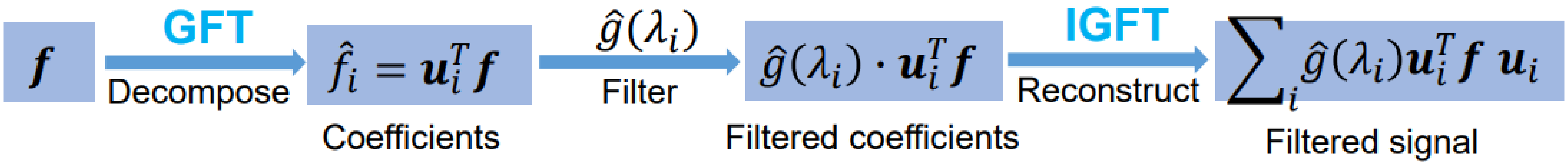
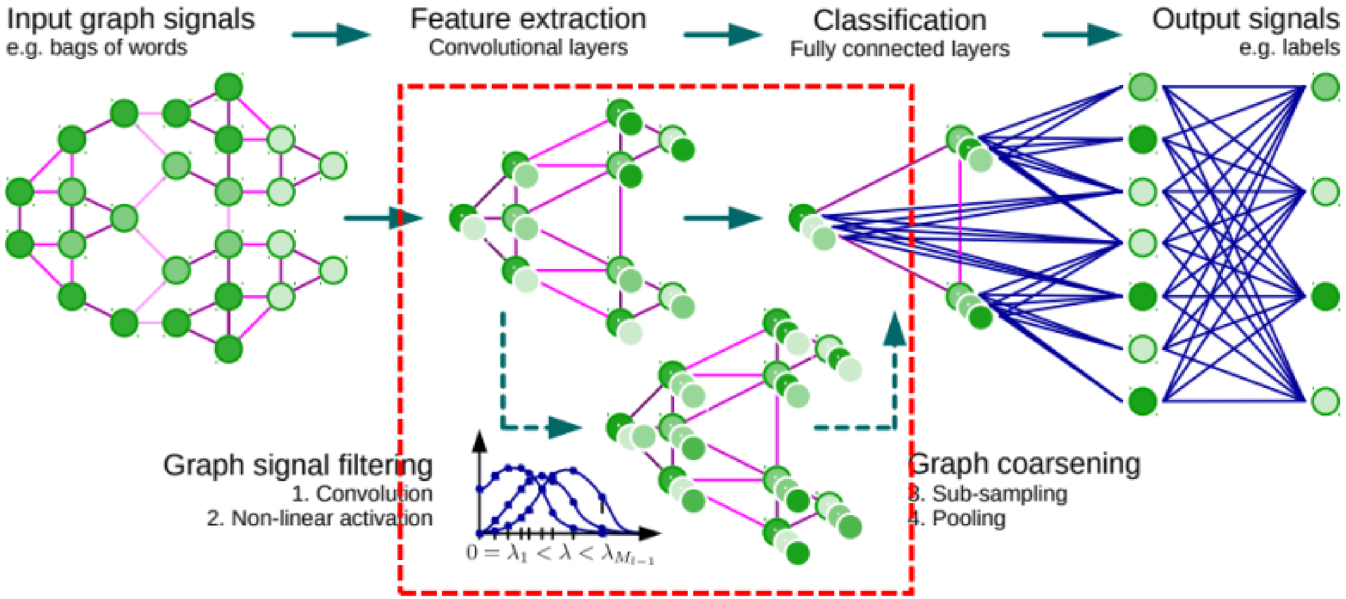
Summary Questions of the lecture

- Explain the spectral filtering of a graph signal
- Decompose the graph signal $\boldsymbol{f}$ with the Fourier Transform, and multiply the filter $\hat{g}$ to filter the transformed signal in the spectral domain. Then, reconstruct the filtered signal in the spatial/temporal domain with the inverse Fourier Transform.



Summary Questions of the lecture

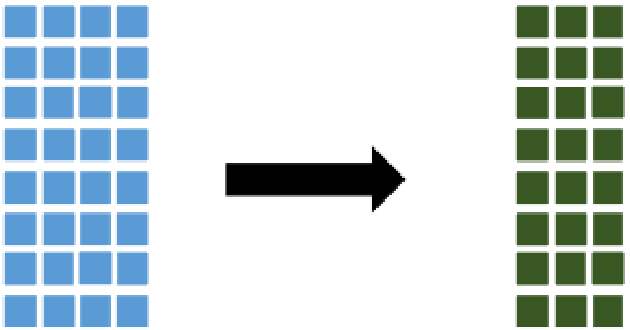
- How to design the spectral filter for GCN?
- The spectral filters are learned in a data-driven way. For end-to-end learning, such filters are parametrized.



Summary Questions of the lecture

- What is the channel in a graph signal?
- Letting N and d_1 denote the number of nodes and the feature dimension in each node respectively, the graph signal is represented by a $N \times d_1$ matrix. Then, each column of the matrix refers to 'channel'.

$$\boldsymbol{F}_{in} \in \mathbb{R}^{N \times d_1} \rightarrow \boldsymbol{F}_{out} \in \mathbb{R}^{N \times d_2}.$$



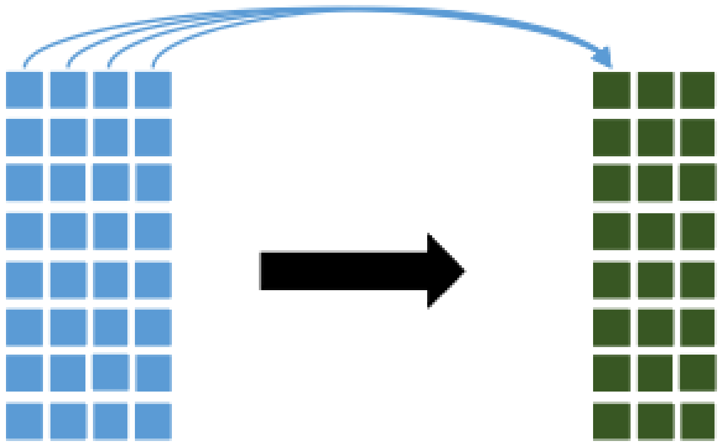
Summary Questions of the lecture

- How to deal with multi-channel signals in GCN?
- To create the n'th output channel, we transform each input channel with a different filter and add up the resulting signals. Thus, we need to learn $d_1 \times d_2$ many filters .

$$\boldsymbol{F}_O[:,n] = \sum_{m=1}^{d_1} \hat{g}_{nm}(\boldsymbol{L}) \boldsymbol{F}_I[:,m] \quad n = 1, \dots d_2$$

Filter for input channel

Learn $d_2 \times d_1$ filter



$GCN(\mathcal{G})$

Geometric deep learning via graph convolution … (continue)

$$\boldsymbol{h}_{\boldsymbol{i}}^{(\boldsymbol{l}+\mathbf{1})} = \sum_{\boldsymbol{v}_{\boldsymbol{j}} \in \boldsymbol{N}(\boldsymbol{v}_{\boldsymbol{i}})} \boldsymbol{f}\left(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{w}_{\boldsymbol{i}\boldsymbol{j}}, \boldsymbol{h}_{\boldsymbol{j}}^{(\boldsymbol{l})}, \boldsymbol{x}_{\boldsymbol{j}}\right)$$