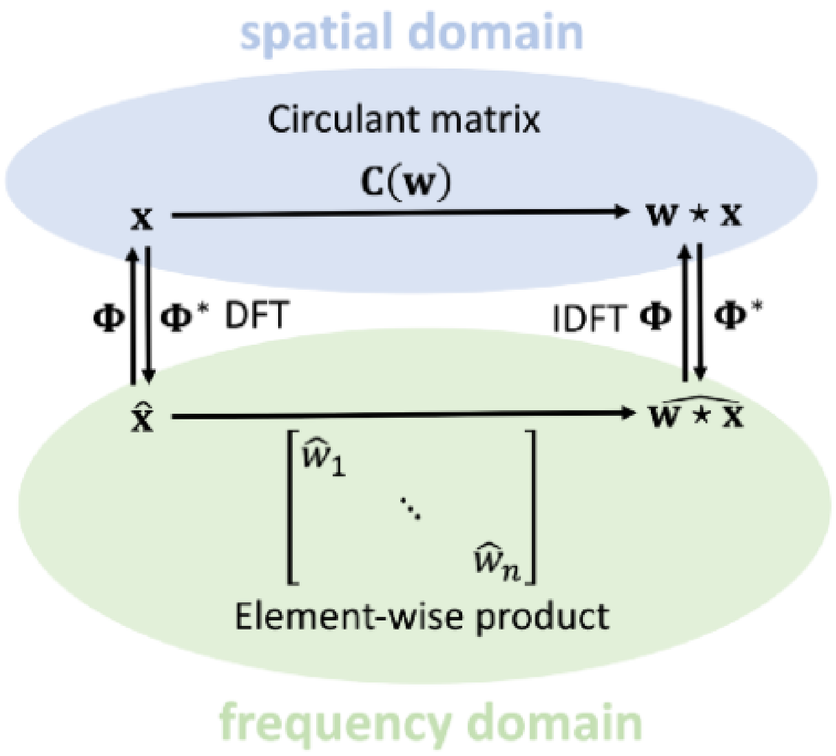
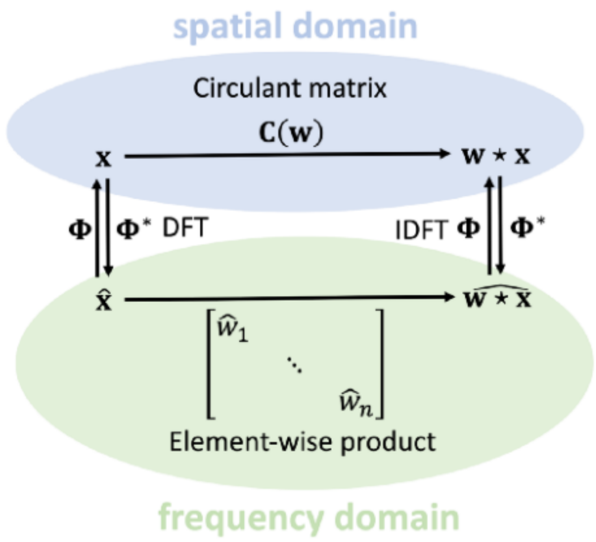
Spectral graph convolutions



https://towardsdatascience.com/deriving-convolution-from-first-principles-4ff124888028

From convolutions to spectral graph convolutions

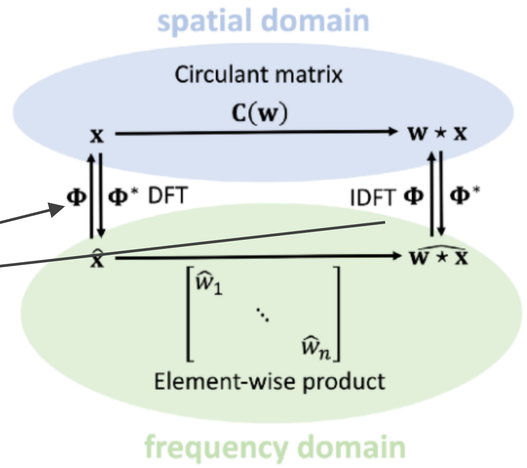
- Inspired by the Convolution Theorem
- We can assume our weights do not change (still a matrix)
- However, what is the 'shift' in graphs?
- What is the equivalent of Fourier basis for graphs?



Graph Laplacian to replace Fourier Transform

- Eigenvectors of Graph Laplacian as analogy to Fourier
  - Equivalent on grids, not on graphs
- For undirected graphs ⇒ symmetric adjacency matrix and graph Laplacian
- For directed graphs ⇒ generalized eigenvectors/Jordan decomposition
  - More elaborate

Replace $\mathbf{\Phi}$ with eigenvectors of graph Laplacian
Otherwise, same thing



Spectral graph convolutions

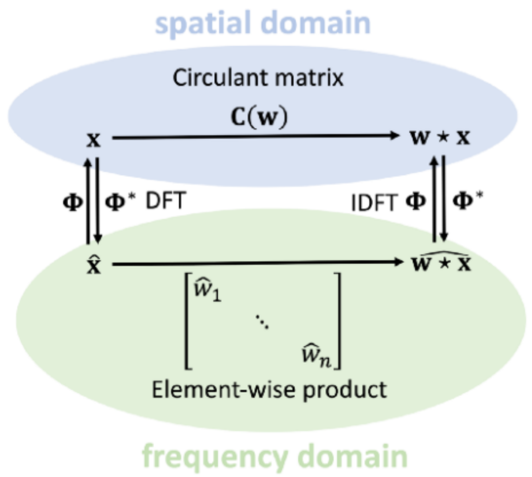
- Similar to regular convolutions
  - Compute (Graph) Fourier Transform

$$\hat{\boldsymbol{x}} = \boldsymbol{\Phi}^* \, \mathbf{x}$$

Where $\boldsymbol{\Phi}^*$ are the eigenvectors (conjugate transpose) of graph Laplacian

  - Apply filter in Fourier space

$$\hat{\boldsymbol{x}} \odot \hat{\boldsymbol{w}} = \begin{bmatrix} \hat{w}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{w}_n \end{bmatrix} \hat{\boldsymbol{x}}$$



  - Compute Inverse (Graph) Fourier Transform

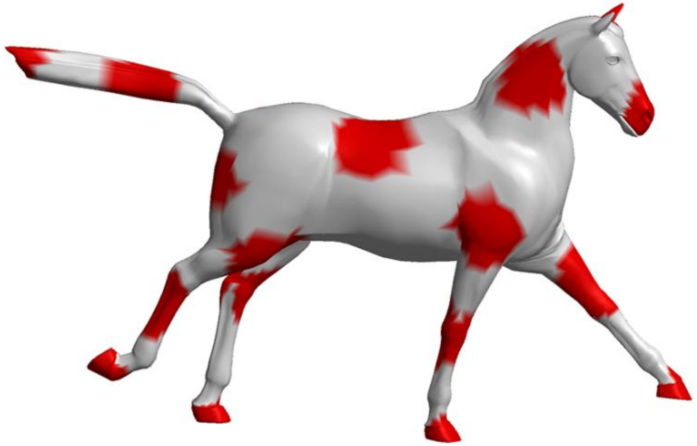
$$\boldsymbol{x} * \boldsymbol{y} = \boldsymbol{\Phi} \cdot (\hat{\boldsymbol{x}} \odot \hat{\boldsymbol{w}})$$

Some drawbacks

- Computational complexity of at least $O(n^2)$
- Parameter complexity of $O(n)$
- Isotropic filters
- Filters that depend on choice of basis and do not generalize across graphs

Basis dependence

<u>Original signal</u>

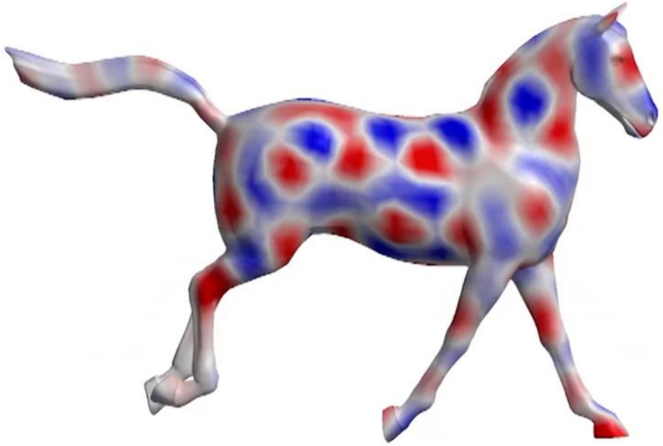


$$\boldsymbol{x}$$

<u>Applying filter with basis #1</u>

$$\boldsymbol{x} * \boldsymbol{y} = \boldsymbol{\Phi} \begin{bmatrix} \hat{y}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{y}_n \end{bmatrix} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{x}$$

<u>Applying filter with basis #2</u>



$$\boldsymbol{x} * \boldsymbol{y} = \boldsymbol{\Psi} \begin{bmatrix} \hat{y}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{y}_n \end{bmatrix} \boldsymbol{\Psi}^{\mathrm{T}} \boldsymbol{x}$$

Isotropic filters

- In a graph there is no sense of “up”, “down”, “left”, “right”
  - We cannot assign an order on edges, so all weights must be shared
  - We don’t even have same size neighborhoods per node
  - ⇒ Isotropic filters