Sign and Basis Invariant Networks

For Spectral Graph Representation Learning







Derek Lim* — *MIT CSAIL*

Josh Robinson* — *MIT CSAIL*

Lingxiao Zhao — *CMU*

Tess Smidt — *MIT RLE*

Suvrit Sra — *MIT LIDS*

Haggai Maron — *NVIDIA Research*

Stefanie Jegelka — *MIT CSAIL*

Motivation: Learning with Eigenvectors

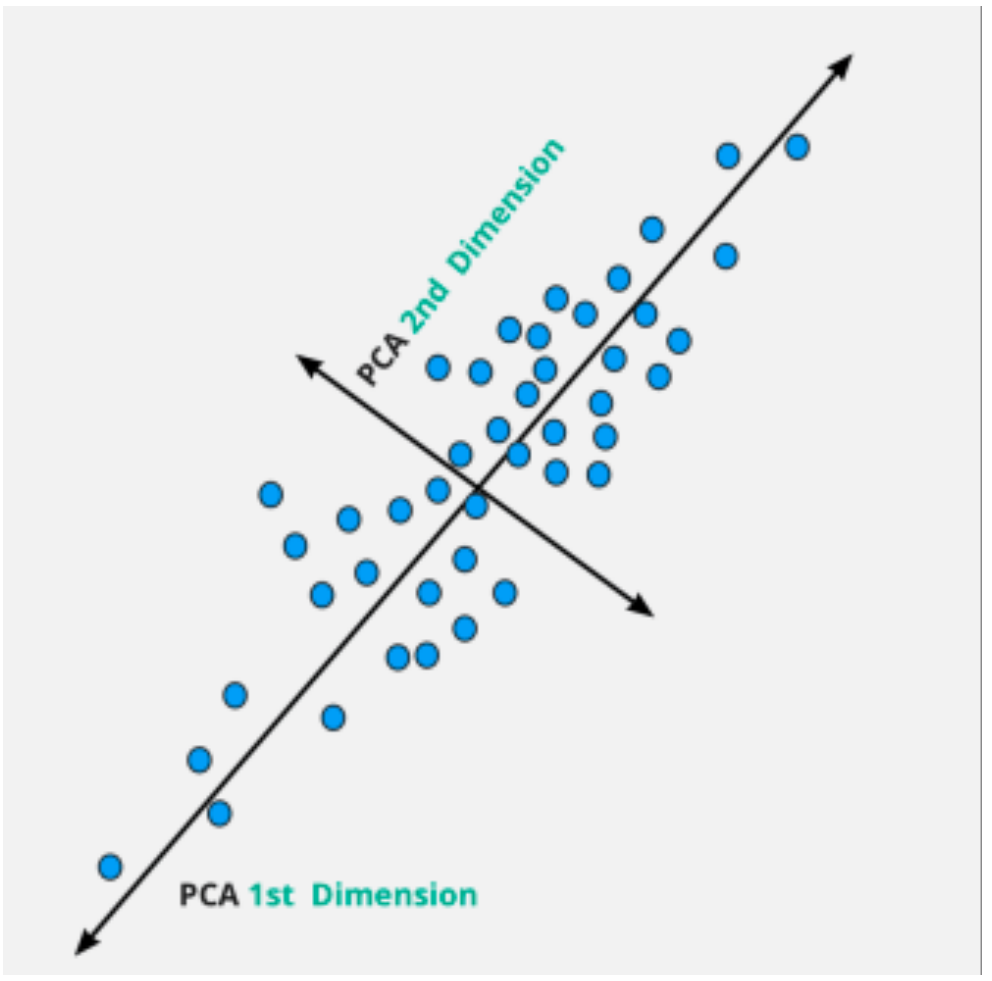
Learning with Eigenvectors is Common

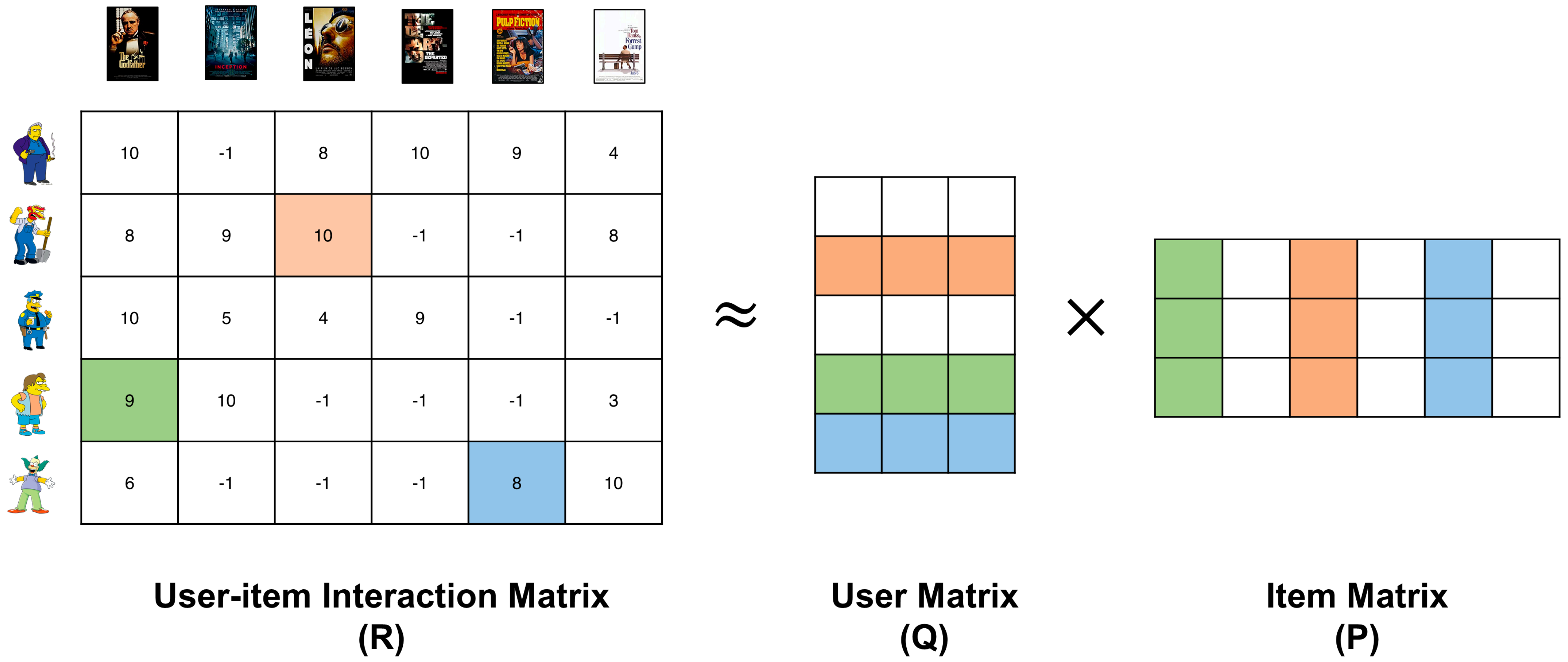
- Singular vectors from SVD, PCA

- Matrix factorization (e.g., Node2Vec, DeepWalk)

  (Grover & Leskovec 2016, Perozzi et al 2014, Qiu et al 2018)

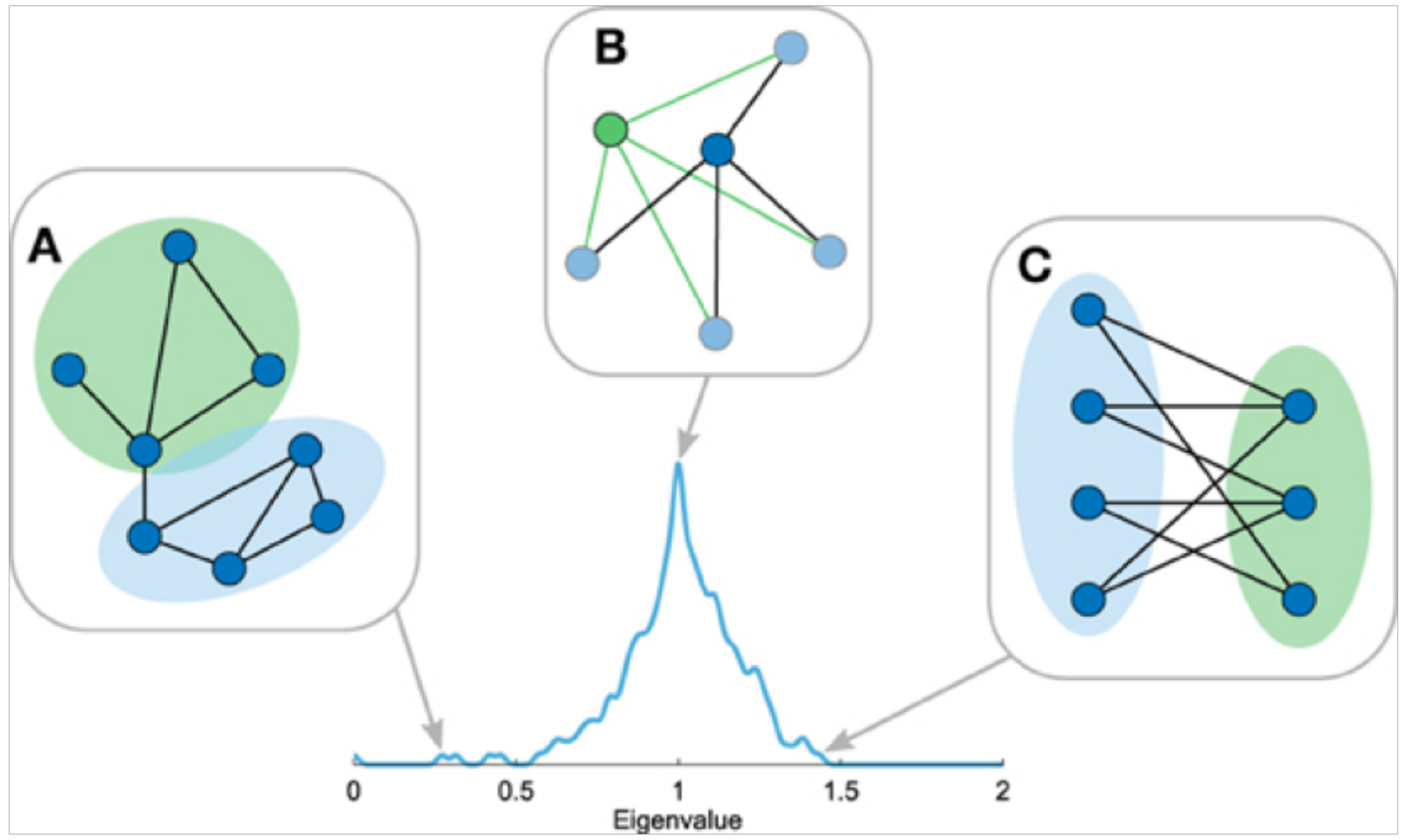
- Eigenvectors of graph adjacency/Laplacian matrix ***(more next)***



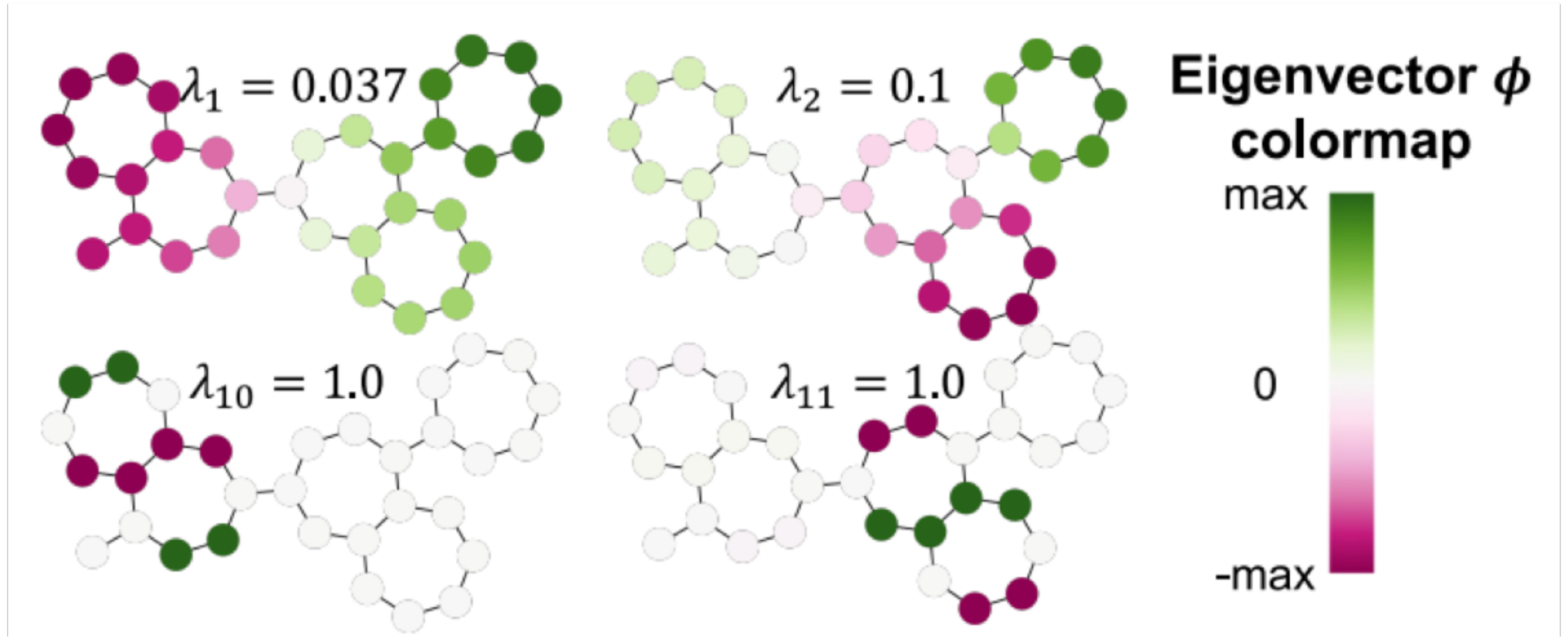


Laplacian Eigenvectors

- Graph Laplacian $L = I - D^{-1/2}AD^{-1/2}$
- Eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$, eigenvectors $v_1, \ldots, v_n$
- Captures distances, subgraphs, local structures, etc.



(de Lange, de Reus, van den Heuvel 2014)



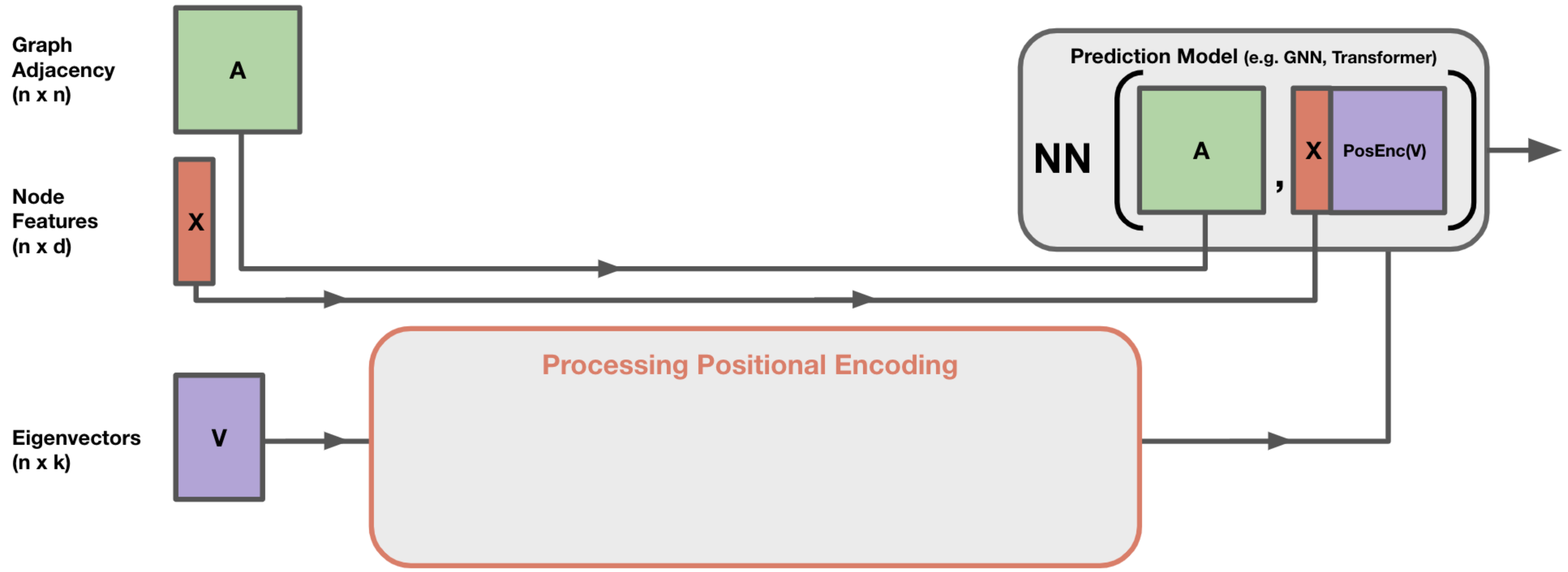
(Kreuzer, Beaini*, Hamilton, Létourneau, Tossou 2021)*

Laplacian Eigenvectors in graph ML

- **Spectral convolution:** *(Bruna et al 2014, Defferard et al 2016)*: replace DFT with learned functions

$$p(L)X = Vp(\Lambda)V^\top X$$
$$= \sum_{i=1}^{n} p(\lambda_i) v_i v_i^\top X$$

- **Graph positional encoding:** replace sin and cos positional encodings in sequences with eigenvectors in graphs
(Feldman et al 2022, Dwivedi et al 2022, Kreuzer & Beaini et al 2021, Dwivedi & Bresson 2021, Mialon et al 2021)



Problem: Handling Invariances in Eigenvectors

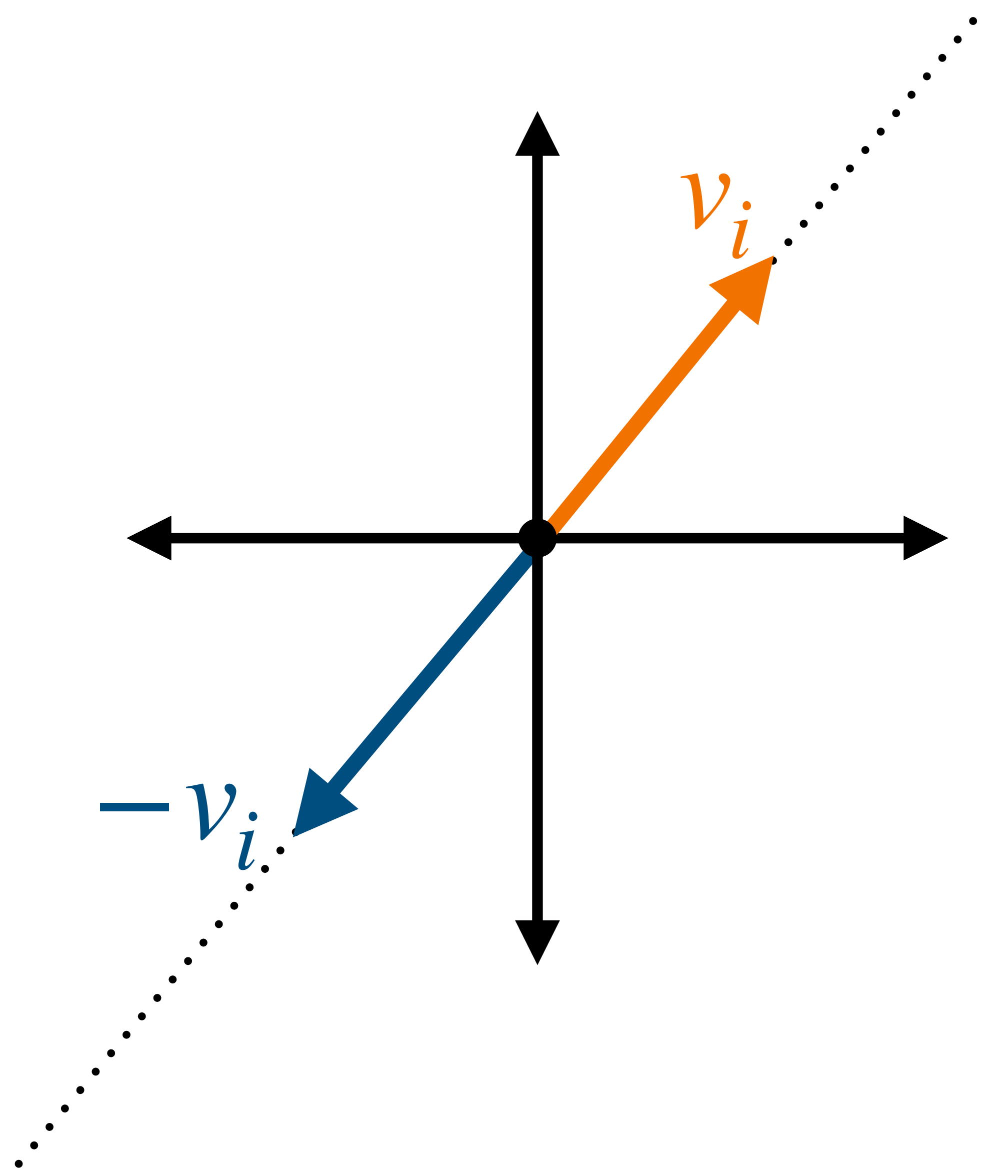
Functions on Eigenvectors

$$f(v_1, \ldots, v_n)?$$

- Can we learn an **arbitrary function** on a set of eigenvectors (and eigenvalues)?
- What invariances must f have?
- How to parameterize an architecture with the required invariances that can approximate **any** f ?

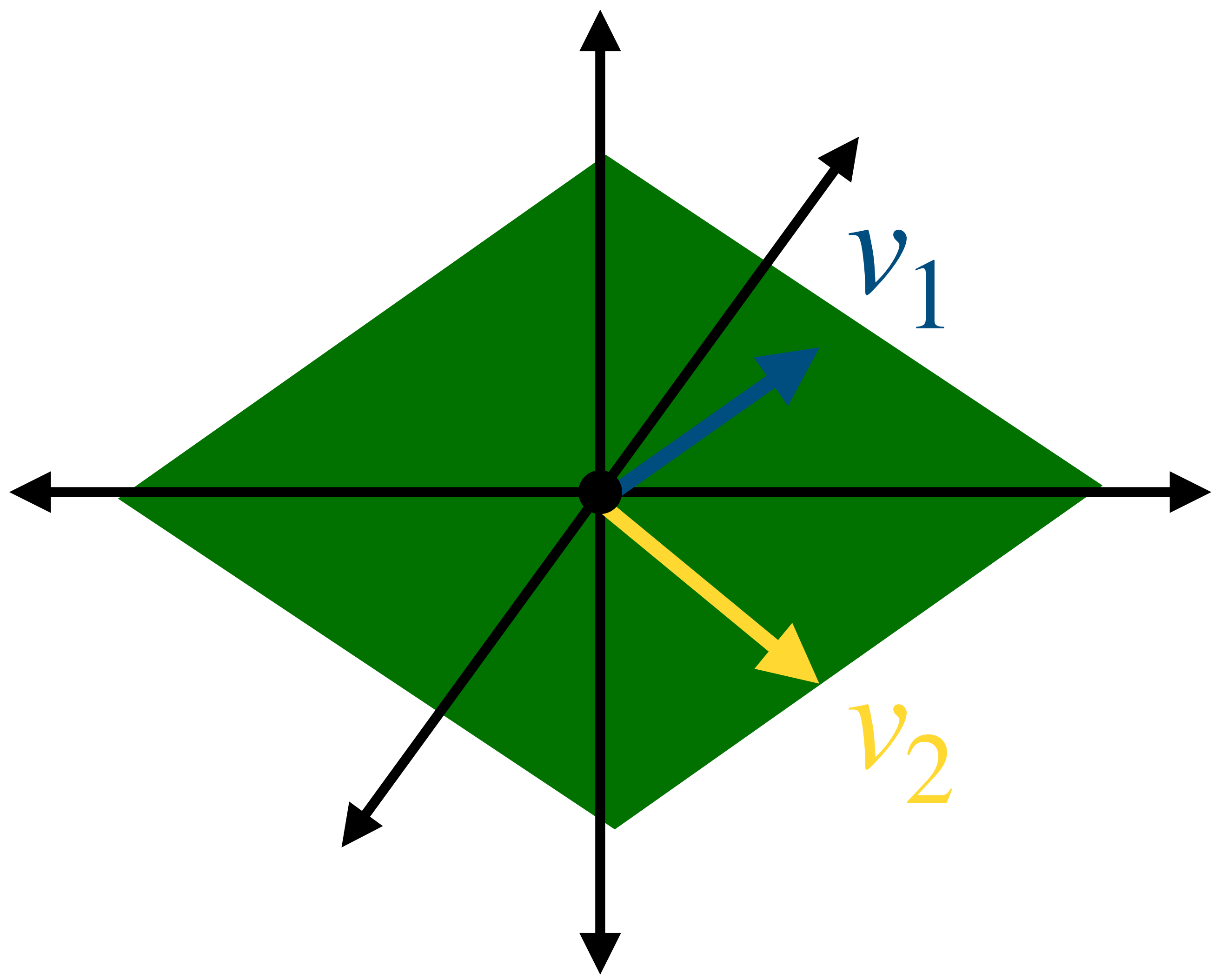
Eigenvector Invariances

- **Sign invariance:** If v_i an eigenvector, then so is $-v_i$. An eigensolver may return either



- **Basis invariance:** when there are multiplicities $\lambda_{i_1} = \ldots = \lambda_{i_d}$

Many bases describe the same d-dim eigenspace



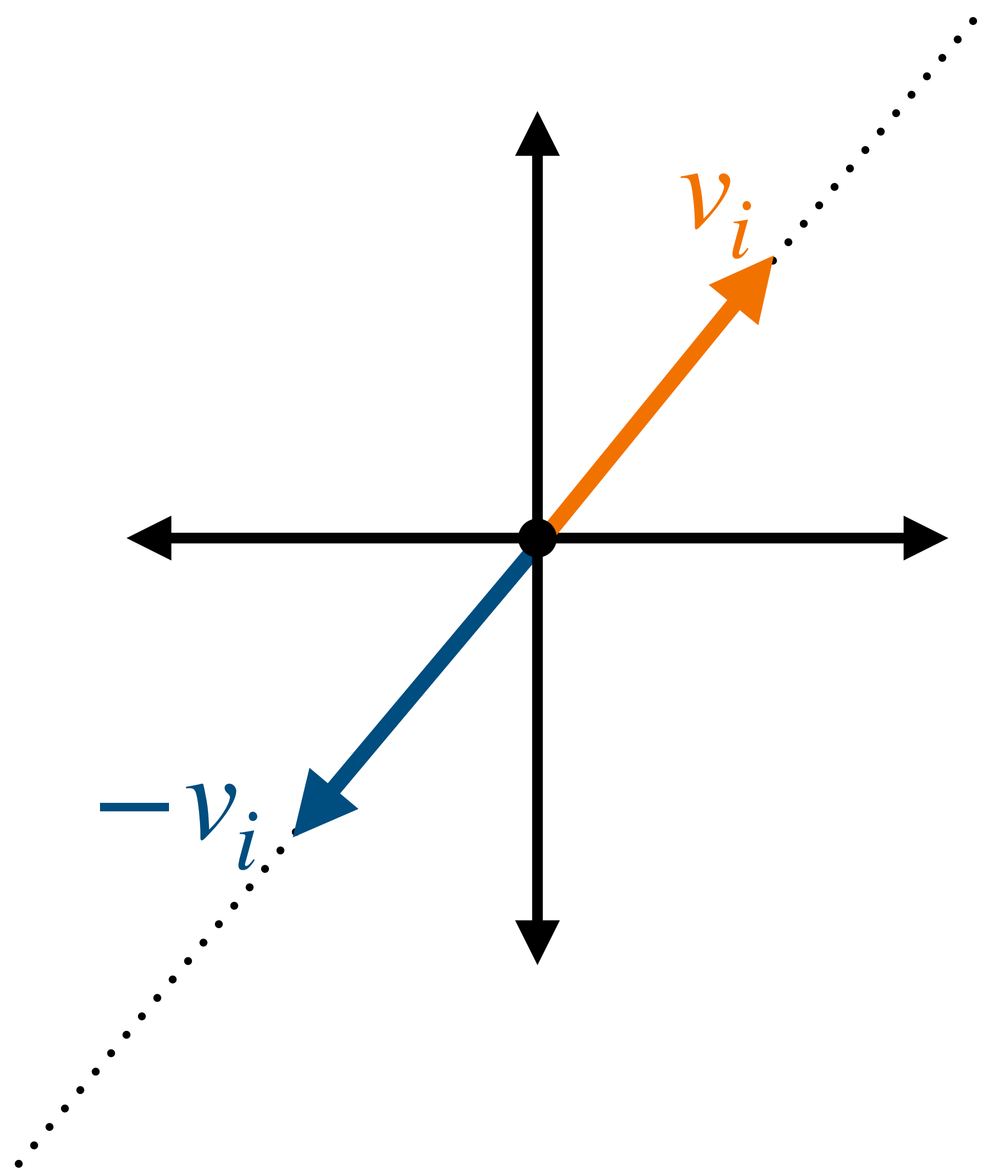
Multiplicities are frequent in real data!

Capturing invariances is critical for generalization

Eigenvector Invariances

- **Sign invariance:** If v_i an eigenvector, then so is $-v_i$. An eigensolver may return either

$$f(v_i) = f(-v_i)$$

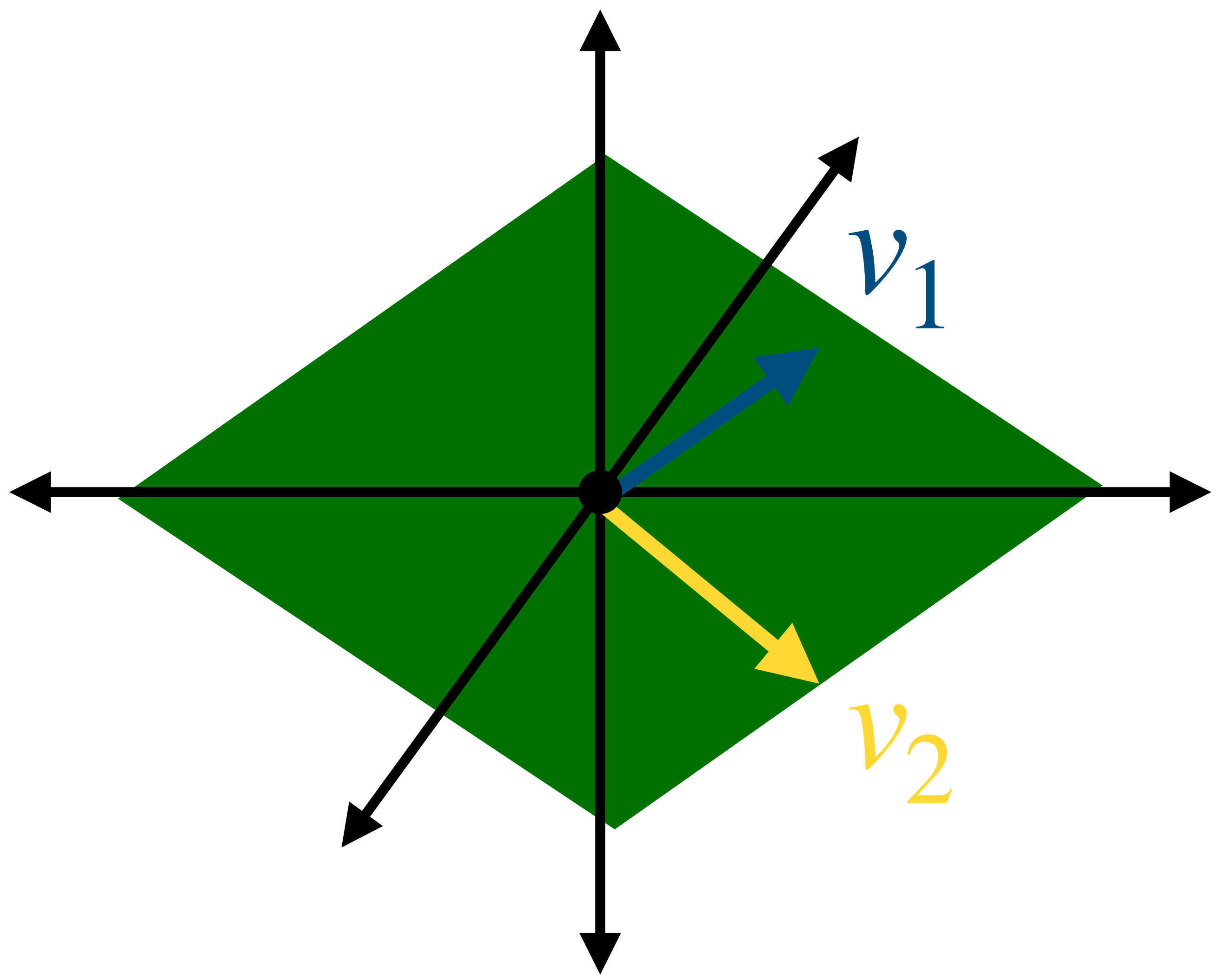


- **Basis invariance:** when there are multiplicities $\lambda_{i_1} = \ldots = \lambda_{i_d}$

Many bases describe the same d-dim eigenspace

$$f(V) = f(VQ) \quad \text{for all } Q \text{ in orthogonal group } O(d)$$

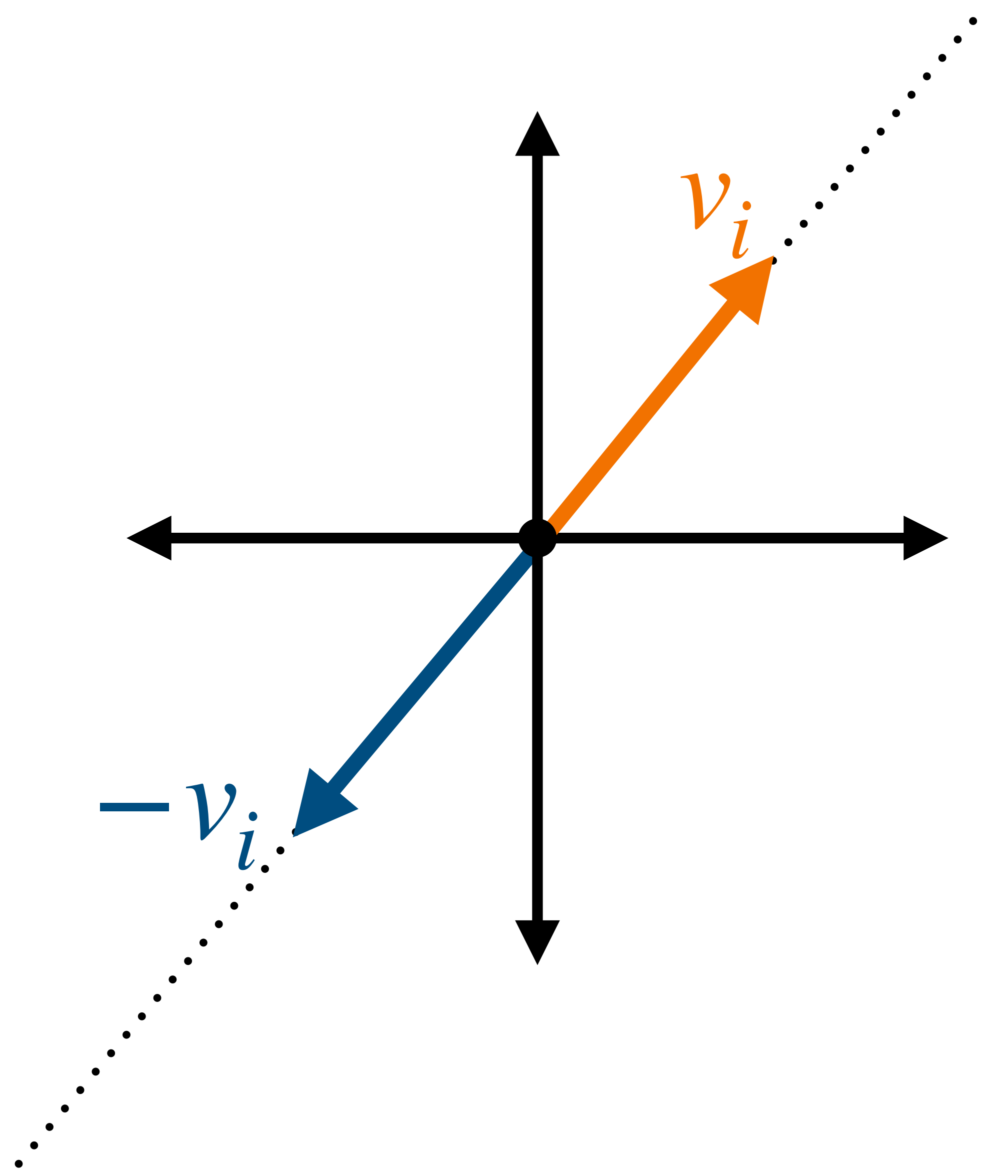
$$V = [v_{i_1}, \ldots, v_{i_d}]$$



Eigenvector Invariances

- **Sign invariance:** If v_i an eigenvector, then so is $-v_i$. An eigensolver may return either

$$f(v_1, \ldots, v_n) = f(\pm v_1, \ldots, \pm v_n)$$

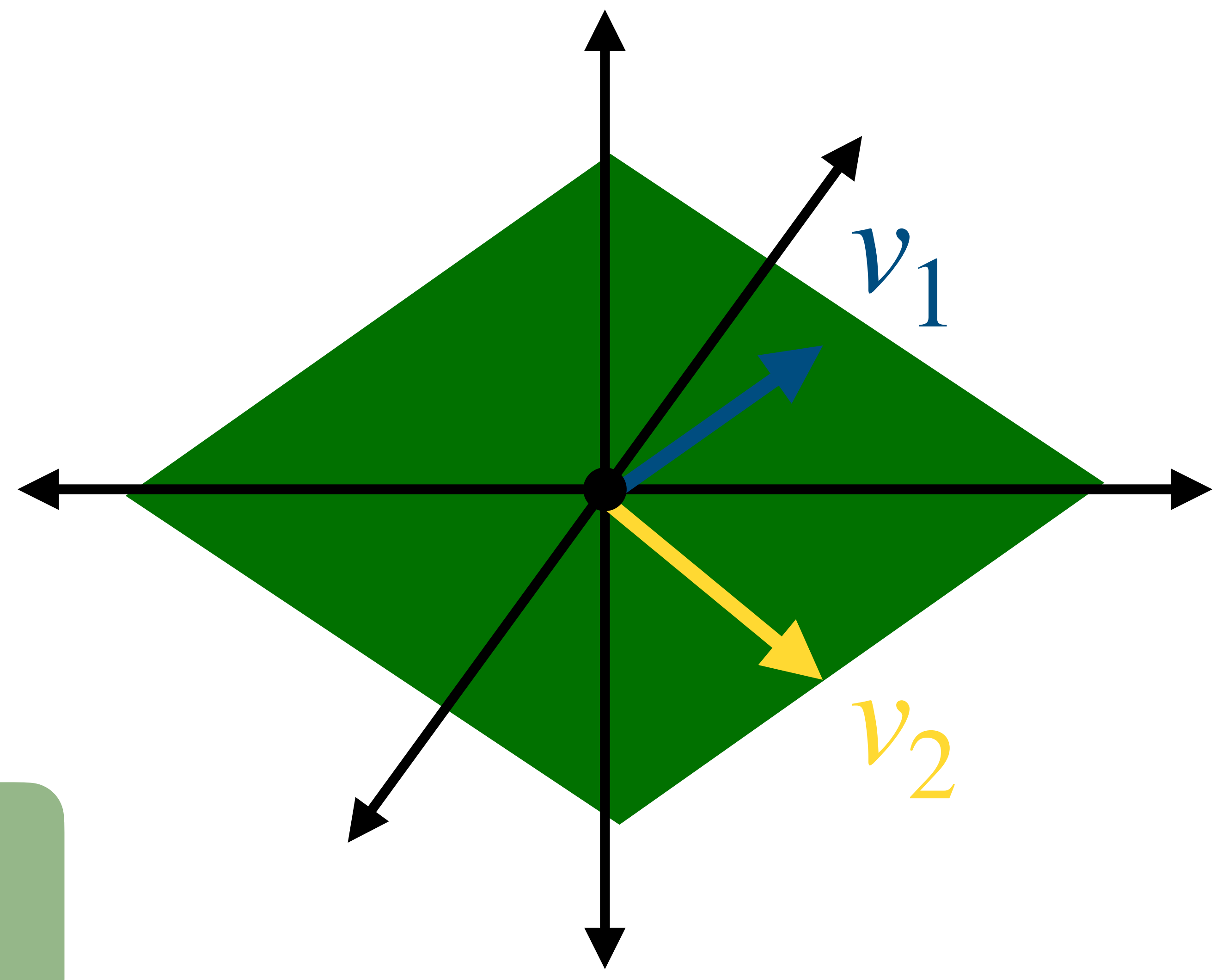


- **Basis invariance:** when there are multiplicities $\lambda_{i_1} = \ldots = \lambda_{i_d}$

Many bases describe the same d-dim eigenspace

$$f(V_1, \ldots, V_l) = f(V_1 Q_1, \ldots, V_l Q_l) \quad \text{for all } Q_j \text{ orthogonal}$$

V_j eigenbasis for jth **distinct** eigenvalue



+ multiple subspaces

Eigenvector Invariances

- **Sign invariance:** If v_i an eigenvector, then so is $-v_i$. An eigensolver may return either

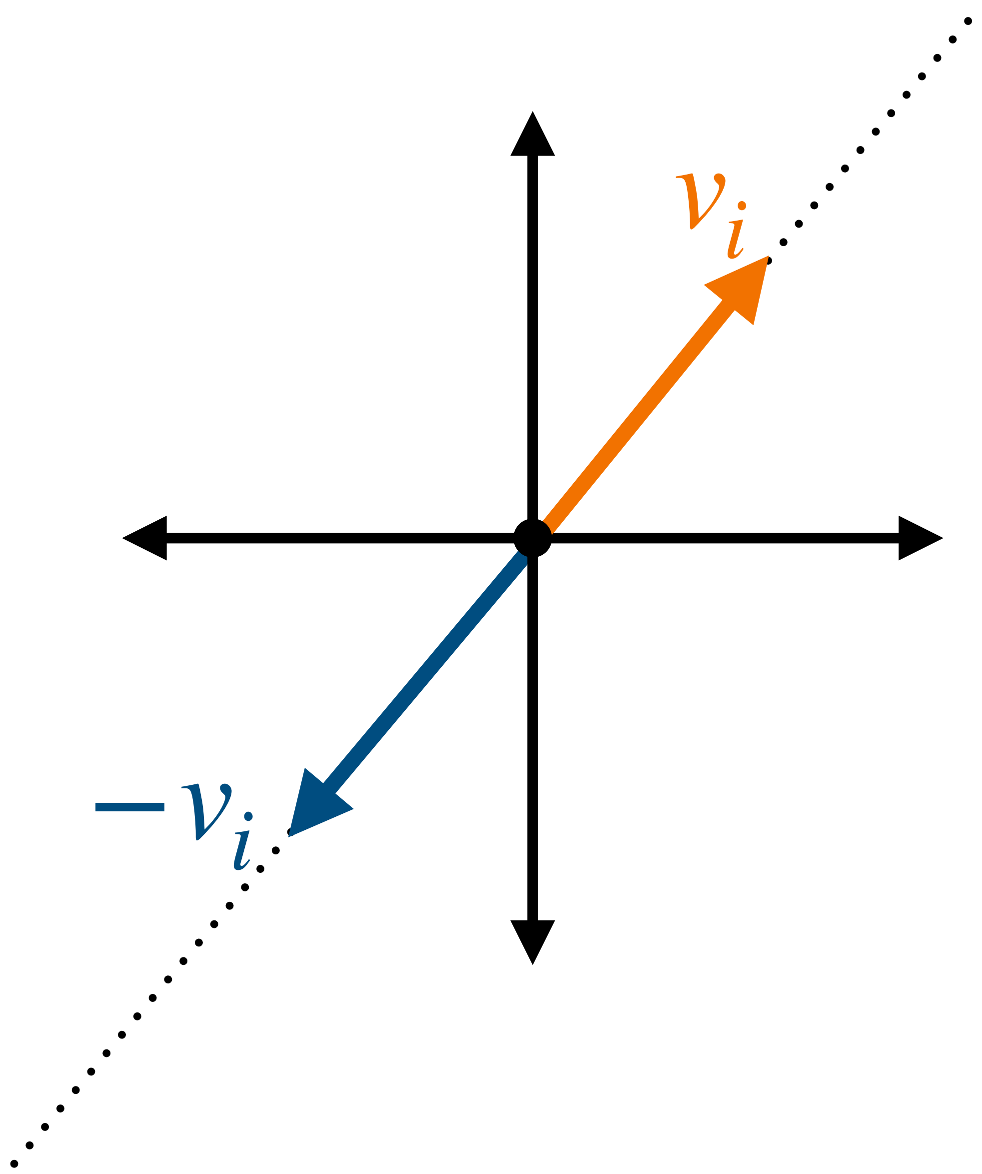
$$f(v_1, \ldots, v_n) = f(\pm v_1, \ldots, \pm v_n)$$

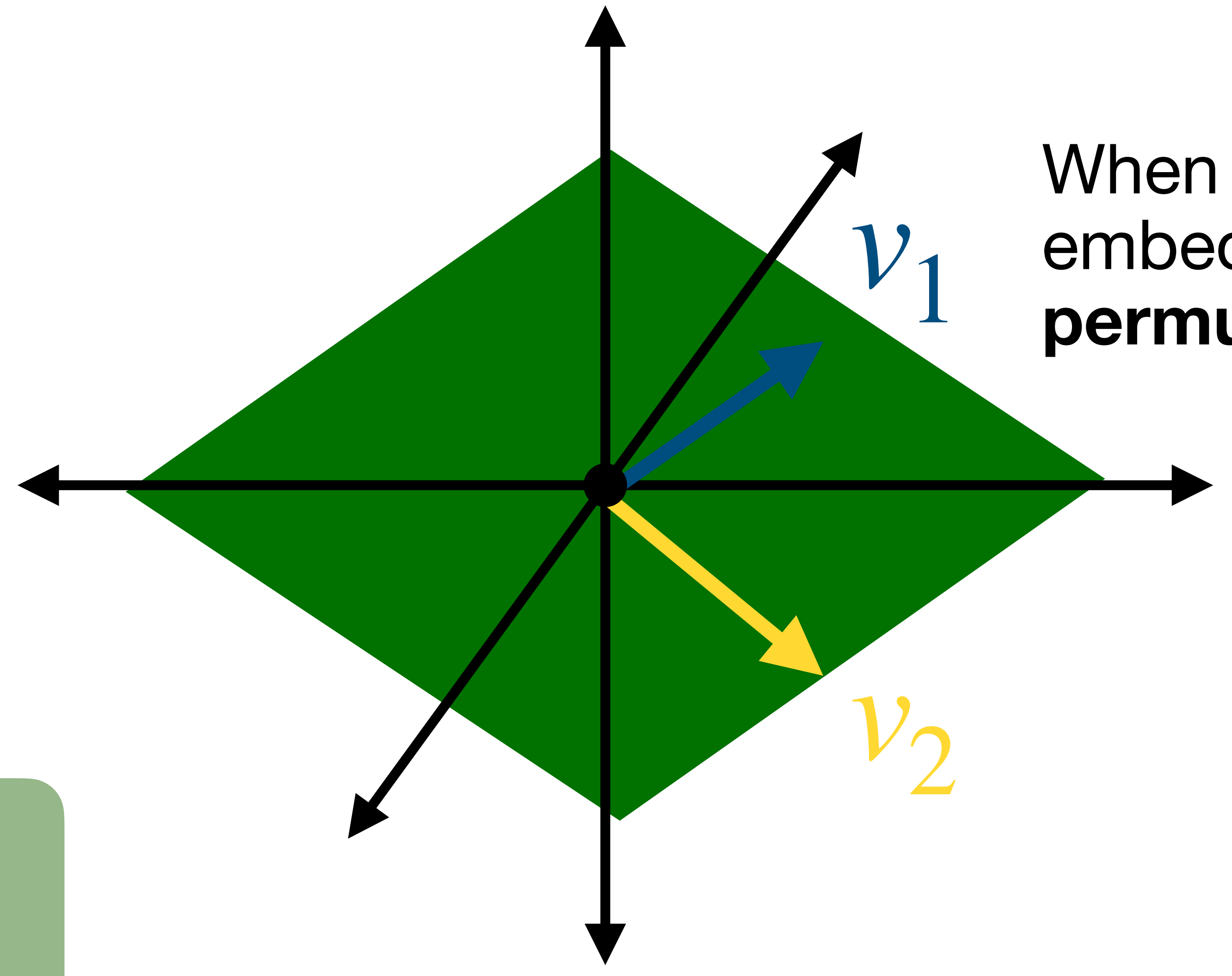
- **Basis invariance:** when there are multiplicities $\lambda_{i_1} = \ldots = \lambda_{i_d}$

Many bases describe the same d-dim eigenspace

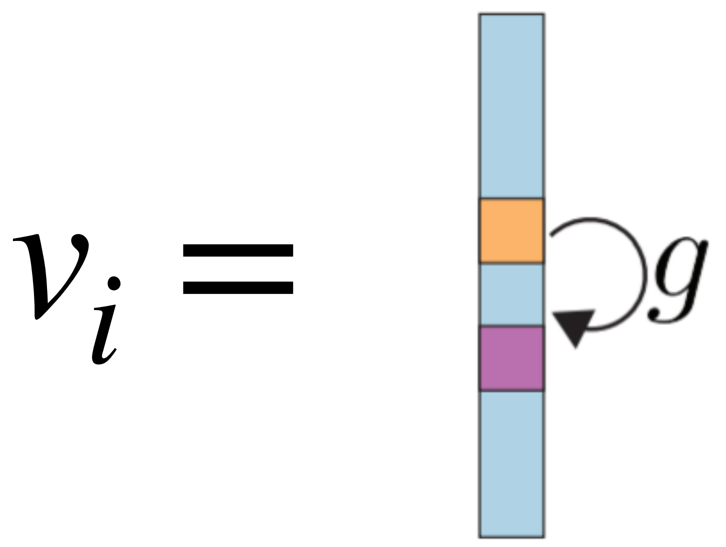
$$f(V_1, \ldots, V_l) = f(V_1 Q_1, \ldots, V_l Q_l) \quad \text{for all } Q_j \text{ orthogonal}$$

V_j eigenbasis for jth **distinct** eigenvalue





When computing node embeddings also want **permutation equivariance:**



+ multiple subspaces
+ permutation equivariance

Parameterizing Invariant Functions on One Eigenspace

Sign invariance on one subspace

Proposition: *(characterizing sign invariance)*

- $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and sign invariant if and only if

  $f(v) = \phi(v) + \phi(-v)$ for some continuous ϕ.

- If f is also permutation equivariant, then ϕ is as well.

Universal Architecture: $f(v) = \phi(v) + \phi(-v)$

General f: $\phi = \text{MLP}$

Permutation equivariant f: ϕ is DeepSet or Set Transformer

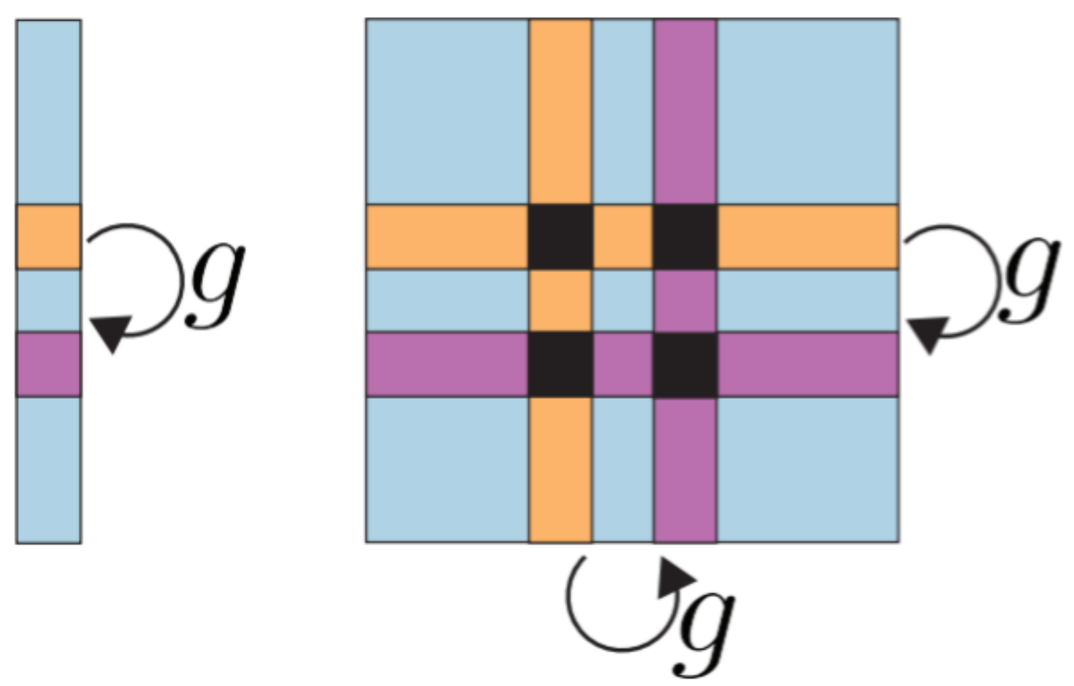
(Zaheer et al 2017, Lee et al 2019)

Basis invariance on one subspace

Proposition: *(characterizing basis invariance)*

- $f : \mathbb{R}^{n \times d} \to \mathbb{R}^n$ is continuous and $f(VQ) = f(V)$ for all orthogonal Q if and only if
    - $f(V) = \phi(VV^\top)$ for some continuous ϕ
- If f is also permutation equivariant, then $\phi : \mathbb{R}^{n \times n} \to \mathbb{R}^n$ permutation equivariant from matrices to vectors

$$(VQ)(VQ)^\top = V(QQ^\top)V^\top = VV^\top$$



Basis invariance on one subspace

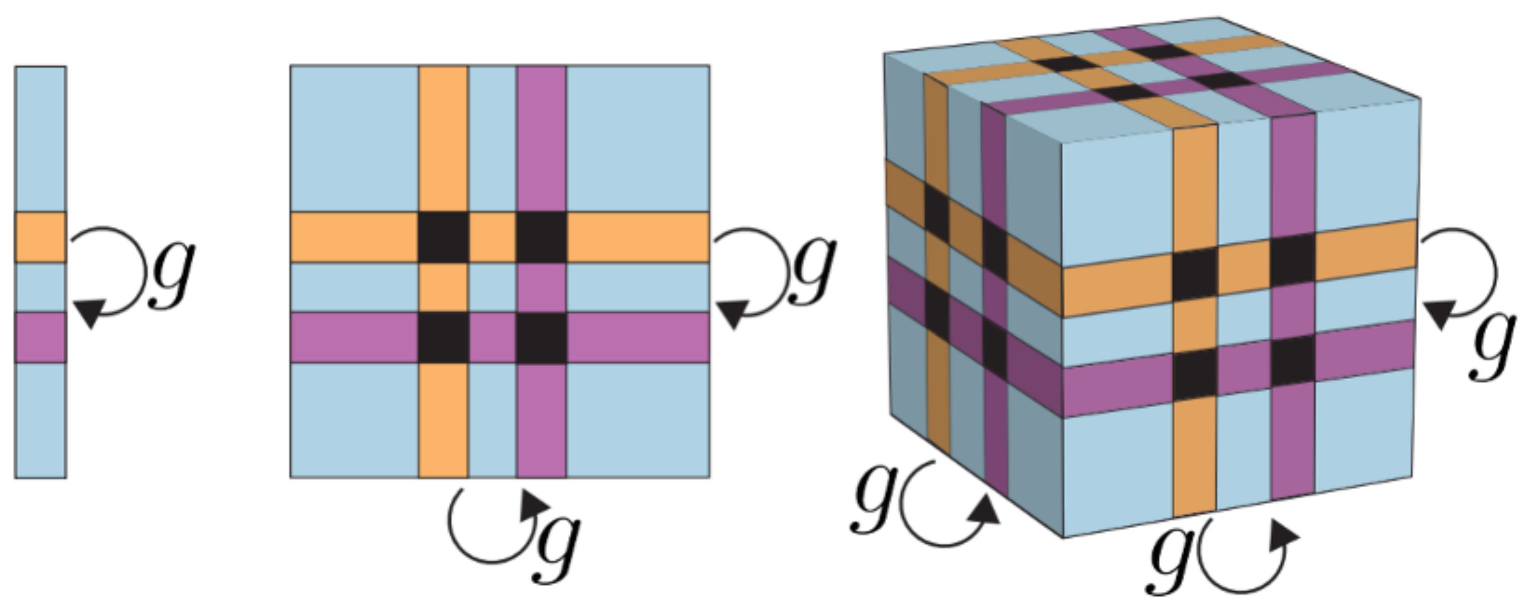
Universal representation of basis invariant functions:

$$f(V) = \phi(VV^\top)$$

General f: $\phi =$ MLP

Permutation equivariant f: $\phi =$ IGN *Invariant graph network* *(Maron et al 2018)*

- IGN: permutation equivariant (tensor) layers
- Universality needs large (order n) tensors, complexity $\Omega(n^n)$, in practice restrict to order 2 tensors



Parameterizing Invariant Functions on Multiple Eigenspaces

Decomposing Into Single Subspaces

Goal: $f(V_1Q_1, \ldots, V_lQ_l) = f(V_1, \ldots, V_l), \qquad (Q_1, \ldots, Q_l) \in O(d_1) \times \ldots \times O(d_l)$

Our Decomposition Theorem

Under mild topological conditions, every continuous f invariant to $G = G_1 \times \ldots \times G_l$ can be represented

$$f(x_1, \ldots, x_l) = \rho\left(\phi_1(x_1), \ldots, \phi_l(x_l)\right), \text{ and}$$

1. ϕ_i is G_i invariant
2. If $G_i = G_j$, can take $\phi_i = \phi_j$

- Takeaway: only need to do G_i invariance for $G_1 \times \ldots \times G_l$ invariance!!
- For sign group, $|G_1 \times \ldots \times G_l| = 2^l$, but $|G_i| = 2$

SignNet: Sign Invariant Neural Net

Our Decomposition Theorem

Under mild topological conditions, every continuous f invariant to $G = G_1 \times \ldots \times G_l$ can be represented

$$f(x_1, \ldots, x_l) = \rho\left(\phi_1(x_1), \ldots, \phi_l(x_l)\right), \text{ and}$$

1. ϕ_i is G_i invariant
2. If $G_i = G_j$, can take $\phi_i = \phi_j$

- **SignNet**: $f(v_1, \ldots, v_l) = \rho\left(\phi(v_1) + \phi(-v_1), \ldots, \phi(v_l) + \phi(-v_l)\right)$

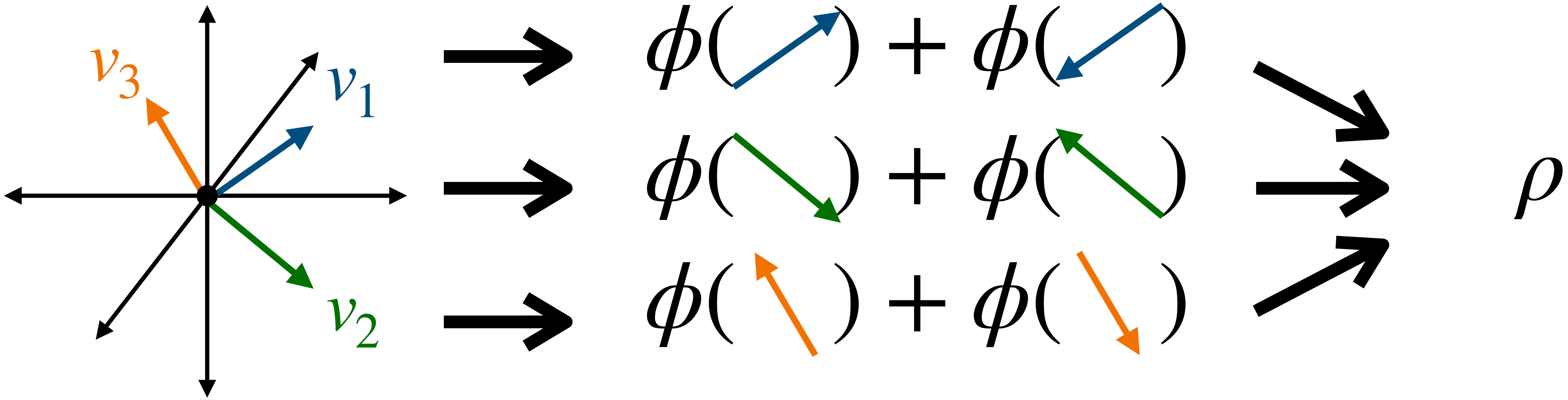
- ρ, ϕ: MLP, DeepSets, Transformer, or GNN
- Can add eigenvalues as arguments

SignNet: Sign Invariant Neural Net

- **SignNet**: $f(v_1, \ldots, v_l) = \rho\left(\phi(v_1) + \phi(-v_1), \ldots, \phi(v_l) + \phi(-v_l)\right)$

ϕ processes each eigenspace

ρ aggregates all eigenspaces



BasisNet: Basis Invariant Neural Net

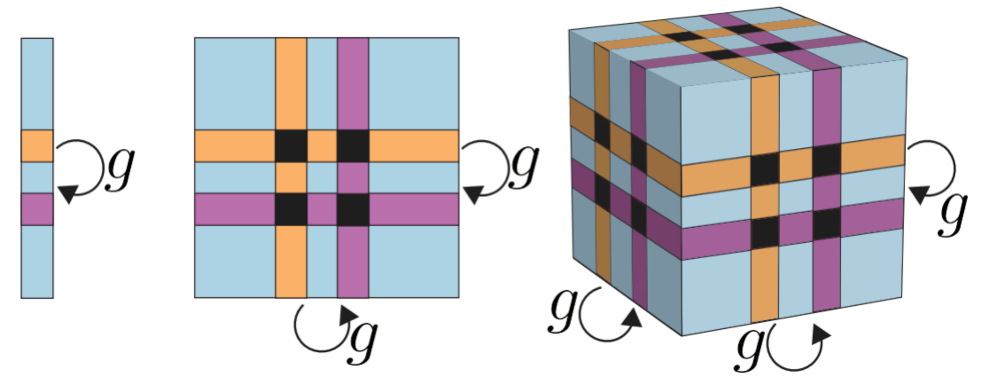
Our Decomposition Theorem

Under mild topological conditions, every continuous f invariant to $G = G_1 \times \ldots \times G_l$ can be represented

$$f(x_1, \ldots, x_l) = \rho\left(\phi_1(x_1), \ldots, \phi_l(x_l)\right), \text{ and}$$

1. ϕ_i is G_i invariant
2. If $G_i = G_j$, can take $\phi_i = \phi_j$

- **BasisNet**: $$f(V_1, \ldots, V_l) = \rho\left(\left[\phi_{d_i}(V_i V_i^\top)\right]_{i=1,\ldots,l}\right)$$

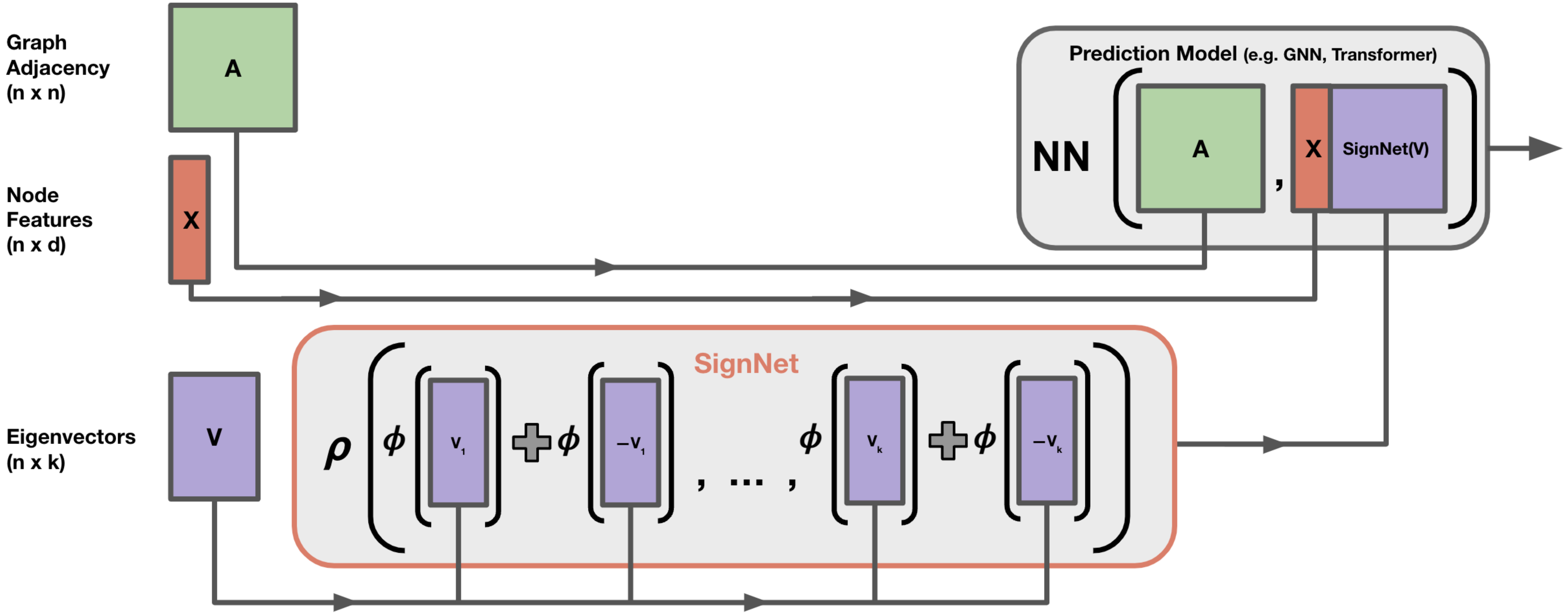


$\phi_d = \mathrm{IGN}_d$ order 2 (efficiency) or higher-order (universality, theoretical construct)

$\rho =$ MLP, DeepSets, Transformer

Theoretical and Empirical Strength for Graph Learning

SignNet for Node Positional Encodings



Experiments: ZINC dataset

Table 1: Results on the ZINC dataset with 500k parameter budget. All models use edge features. Numbers are the mean and standard deviation over 4 runs, each with different seeds.

Base model	Positional encoding	k	#param	Test MAE (↓)
GatedGCN	No PE	N/A	492k	$0.252_{\pm 0.007}$
	LapPE (flip)	8	492k	$0.198_{\pm 0.011}$
	LapPE (abs.)	8	492k	$0.204_{\pm 0.009}$
	LapPE (can.)	8	505k	$0.298_{\pm 0.019}$
	SignNet ($\phi(v)$ only)	8	495k	$0.148_{\pm 0.007}$
	SignNet	8	495k	$0.121_{\pm 0.005}$
	SignNet	All	505k	$\mathbf{0.102}_{\pm \mathbf{0.001}}$
Sparse Transformer	No PE	N/A	473k	$0.283_{\pm 0.030}$
	LapPE (flip)	16	487k	$0.223_{\pm 0.007}$
	SignNet	16	479k	$0.115_{\pm 0.008}$
	SignNet	All	486k	$\mathbf{0.102}_{\pm \mathbf{0.005}}$
GINE	No PE	N/A	470k	$0.170_{\pm 0.002}$
	LapPE (flip)	16	470k	$0.178_{\pm 0.004}$
	SignNet	16	470k	$0.147_{\pm 0.005}$
	SignNet	All	417k	$\mathbf{0.102}_{\pm \mathbf{0.002}}$
PNA	No PE	N/A	474k	$0.133_{\pm 0.011}$
	LapPE (flip)	8	474k	$0.132_{\pm 0.010}$
	SignNet	8	476k	$0.105_{\pm 0.007}$
	SignNet	All	487k	$\mathbf{0.084}_{\pm \mathbf{0.006}}$

- Beat all eigenvector baselines
- Best positional encoding
- PNA + SignNet SOTA on ZINC 500k param
  - CIN gets .079 *(Bodnar et al. 2022)*

- **SignNet**: $$f(v_1, \ldots, v_l) = \rho\left(\phi(v_1) + \phi(-v_1), \ldots, \phi(v_l) + \phi(-v_l)\right)$$

  ρ=MLP, ϕ=GIN

Expressive power: existing positional encodings

Can universally approximate **graph positional encodings** like:

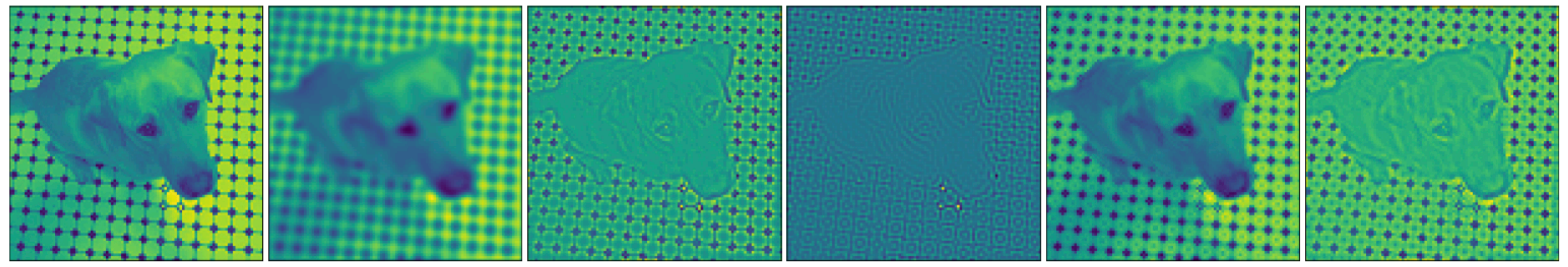
• Heat kernel PE *(Feldman et al. 2022)*	*SignNet, BasisNet*
• Random walk PE *(Dwivedi et al. 2022)*	*SignNet, BasisNet*
• Diffusion, p-step random walk relative PE *(Mialon et al. 2021)*	*BasisNet*
• Generalized PageRank, landing probability distance encodings *(Li et al. 2019)*	*BasisNet*

Expressive power: spectral graph convolutions

SignNet and BasisNet universally approximate **spectral graph convolutions**

$$h(V, \Lambda, X) = Vp(\Lambda)V^\top X = \sum_{i=1}^{n} p(\lambda_i) v_i v_i^\top X$$

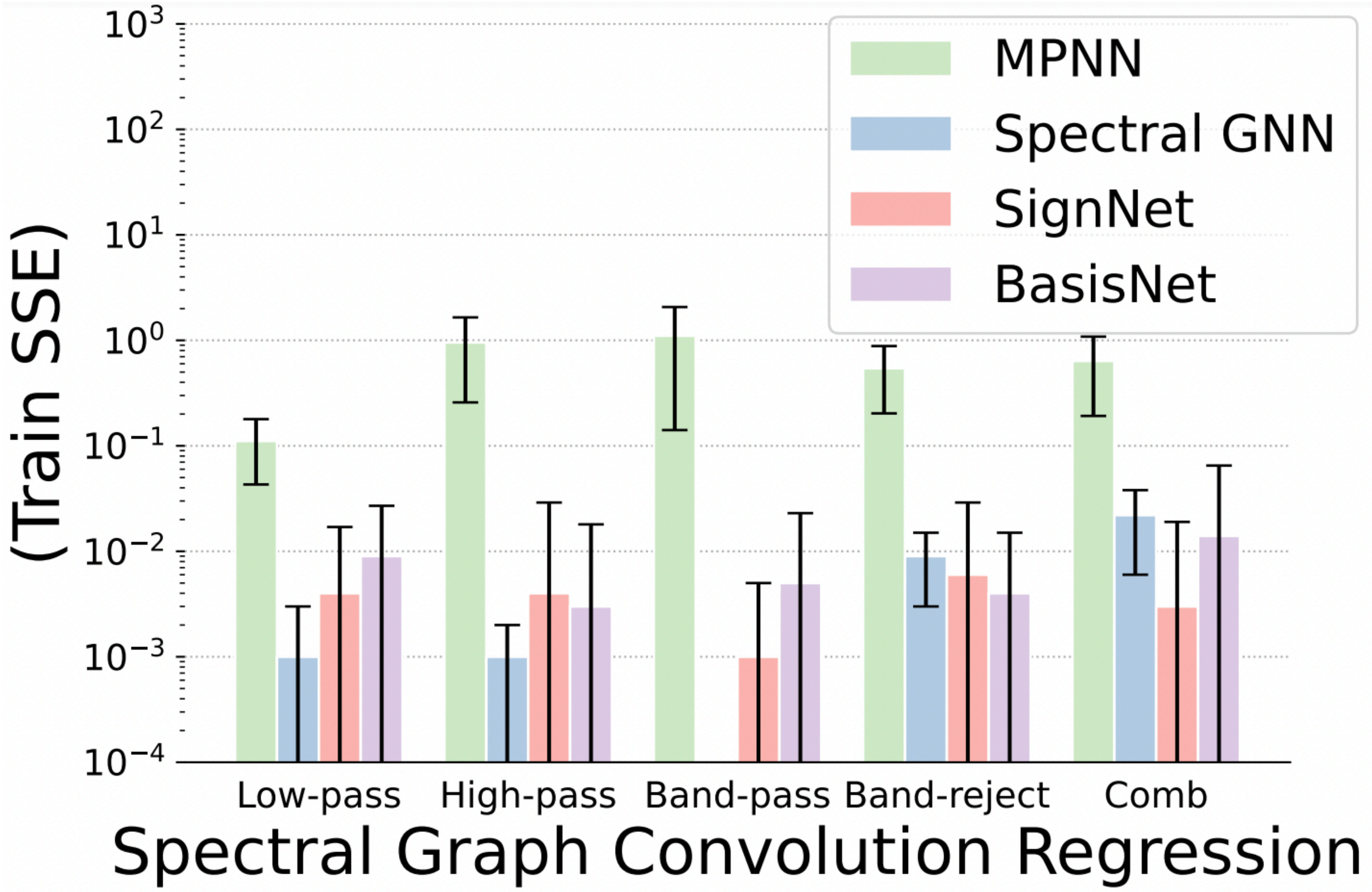
SignNet with: $\phi(v, \lambda, X) = \frac{1}{2} p(\lambda) v v^\top X, \qquad \rho = \sum_{i=1}^{n}$



(a) Original (b) Low-pass (c) High-pass (d) Band-pass (e) Band-rejection (f) Comb

Experiments: Spectral Graph Conv Regression

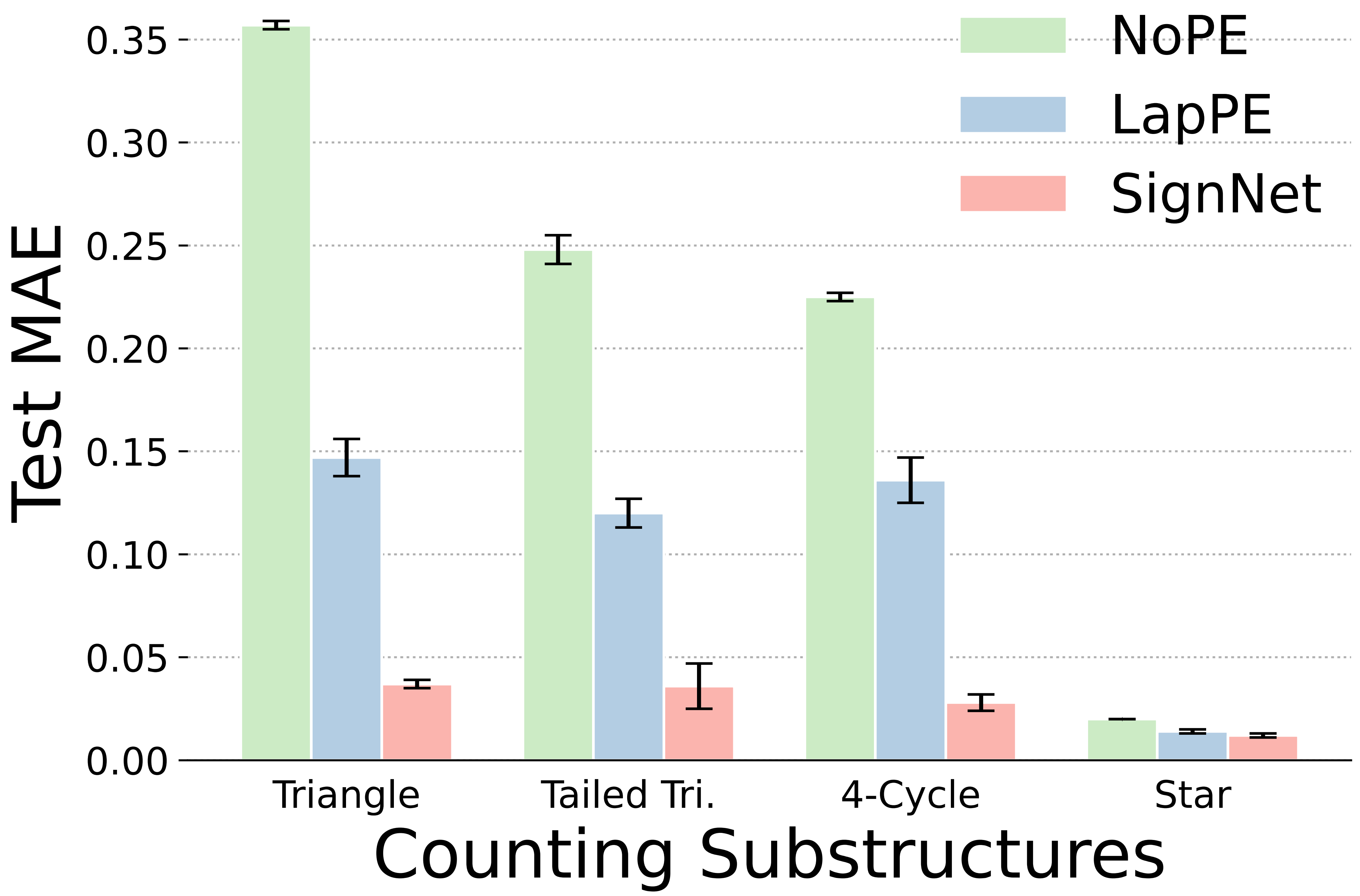
- $\rho, \phi \in \{\text{DeepSets}, \text{Transformer}, \text{IGN}\}$ (no GNN!!) *(Zaheer 2017, Vaswani et al 2017, Maron 2018)*
- Beat sign flip, absolute value eigvec baselines
- Beat message passing, often beat spectral GNNs



Expressive power: spectral graph invariants

- BasisNet can approximate **spectral graph invariants** *(Cvetković 1991)*
  - e.g. graph angles $||V_i V_i^\top e_j||$.
- Thus, can learn **number of 3-, 4-, 5-cycles, connectivity, length-k closed walks**, message passing GNNs cannot!!

- **Experiments**: SignNet boosts performance of message passing for counting substructures
- $\phi = \text{GIN}$ *(Xu et al 2019)* $\rho = \text{Transformer}$ *(Vaswani et al 2017)*



Visualization and Additional Results

Visualization of SignNet on Molecules

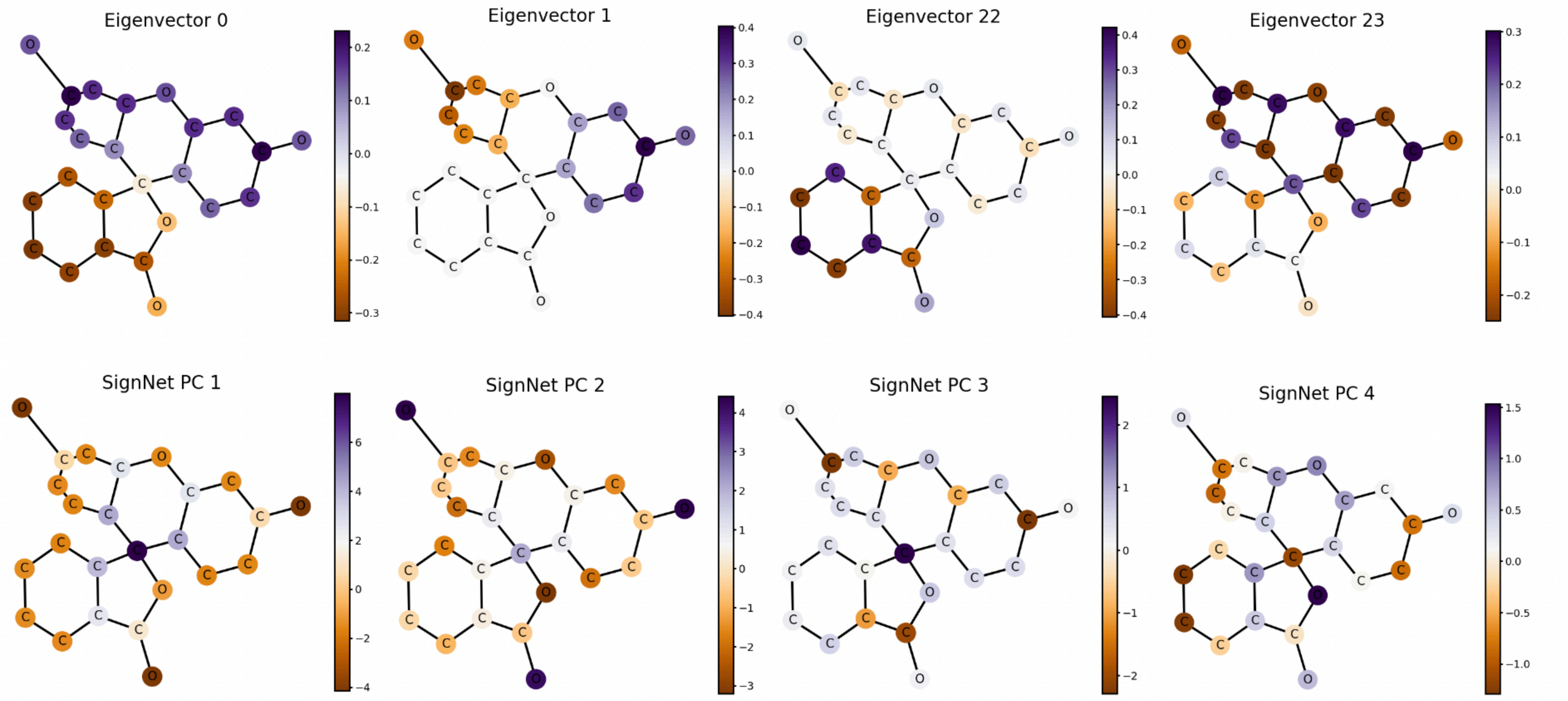
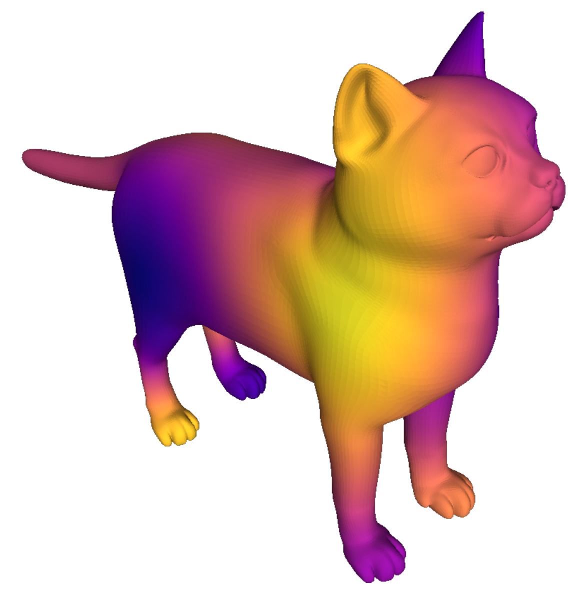
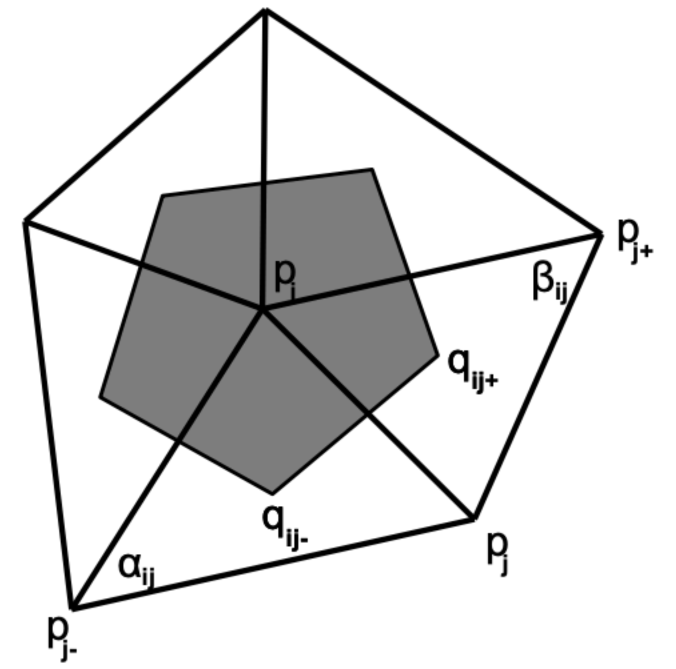


Figure 4: Normalized Laplacian eigenvectors and learned positional encodings for the graph of Fluorescein. (Top row) From left to right: smallest and second smallest nontrivial eigenvectors, then second largest and largest eigenvectors. (Bottom row) From left to right: first four principal components of the output $\rho([\phi(v_i) + \phi(-v_i)]_{i=1,\ldots,n})$ of SignNet.

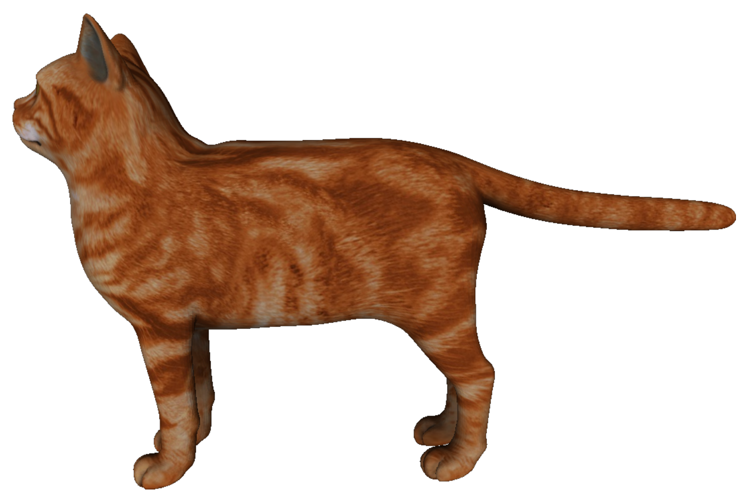
SignNet (trained on ZINC) learns structural information not found in original Laplacian eigenvectors!

Laplacians on Surfaces: Texture Reconstruction





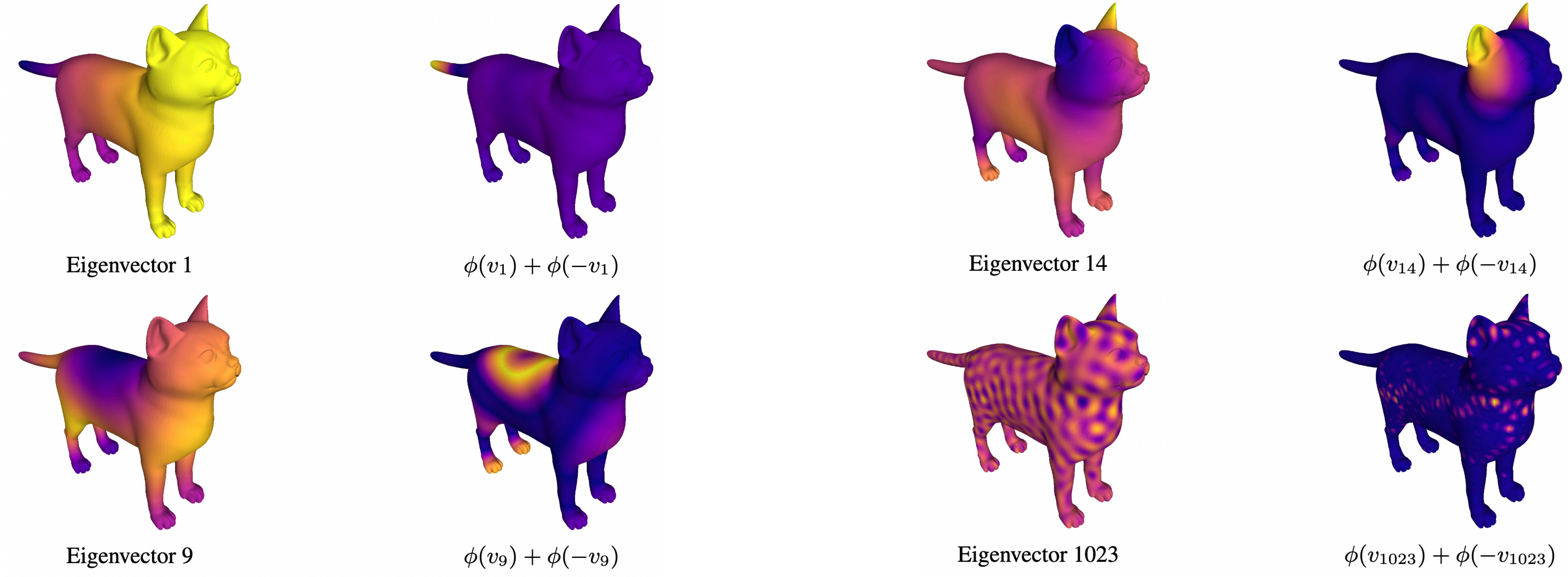
- Cotangent Laplacian eigenvectors of triangle mesh
- Texture reconstruction:
  - Input: Cotan Laplacian eigvecs at point p on surface
  - Output: RGB value of surface at p



- Intrinsic Neural Field *(Koestler et al. 2022):* $f(v_1, \ldots, v_l) = \mathrm{MLP}(v_1, \ldots, v_l)$

- With SignNet: $f(v_1, \ldots, v_l) = \mathrm{MLP}(\mathrm{SignNet}(v_1, \ldots, v_l))$

Cat Visualization



SignNet ϕ learns bilateral symmetry (texture has bilateral symmetry)

Conclusion

Summary: SignNet and BasisNet

- **SignNet and BasisNet**: Invariant neural nets for eigenvector data
- **Universal approximation**: Decomposition Theorem into single subspaces
- **Expressive power**: Spectral graph convolutions, spectral invariants
- **Experiments**: Powerful positional encoding, SOTA on ZINC 500k param

@dereklim_lzh

@Josh_d_robinson

github.com/cptq/SignNet-BasisNet

arxiv.org/abs/2202.13013

Appendix

Complexity

- SignNet: $f(v_1 \ldots, v_l) = \rho\left(\phi(v_1) + \phi(-v_1), \ldots, \phi(v_l) + \phi(-v_l)\right)$
  - Memory: $\mathcal{O}(nlh)$
- BasisNet: $f(V_1 \ldots, V_l) = \rho\left(\mathrm{IGN}_{d_1}(V_1 V_1^\top), \ldots, \mathrm{IGN}_{d_l}(V_l V_l^\top)\right)$
  - Memory: $\mathcal{O}(n^2 lh)$

Real world multiplicities

Table 4: Eigenspace statistics for datasets of multiple graphs. From left to right, the columns are: dataset name, number of graphs, range of number of nodes per graph, largest multiplicity, and percent of graphs with an eigenspace of dimension > 1.

Dataset	Graphs	# Nodes	Max. Mult	% Graphs mult. > 1
ZINC	12,000	9-37	9	64.1
ogbg-molhiv	41,127	2 - 222	42	68.0
IMDB-M	1,500	7 - 89	37	99.9
COLLAB	5,000	32 - 492	238	99.1
PROTEINS	1,113	4 - 620	20	77.3
COIL-DEL	3,900	3 - 77	4	4.00

Table 5: Eigenspace statistics for single graphs. From left to right, the columns are: dataset name, number of nodes, distinct eigenvalues (i.e. distinct eigenspaces), number of unique multiplicities, largest multiplicity, and percent of eigenvectors belonging to an eigenspace of dimension > 1.

Dataset	Nodes	Distinct λ	# Mult.	Max Mult.	% Vecs mult. > 1
32×32 image	1,024	513	3	32	96.9
Cora	2,708	2,187	11	300	19.7
Citeseer	3,327	1,861	12	491	44.8
Amazon Photo	7,650	7,416	8	136	3.71

Real-world graphs: many multiplicities, high dimensional eigenspaces.