Fast Attributed Graph Embedding via Density of States

Saurabh Sawlani Lingxiao Zhao Leman Akoglu

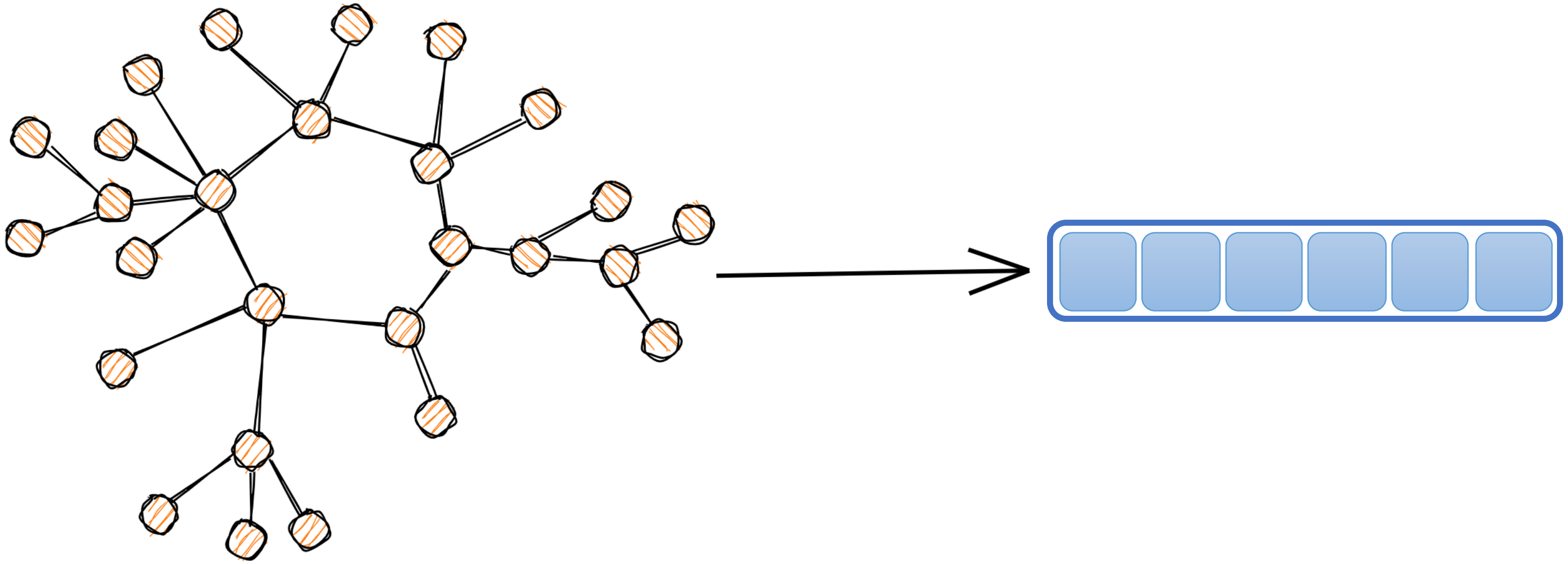


Carnegie Mellon University

Talk outline

- Graph embedding: desirable characteristics
- Prior and related work
- Proposed method: A-DOGE
- Applications

Graph embedding



Picture from https://towardsdatascience.com/graph-representation-learning-network-embeddings-d1162625c52b

Aim: Represent a graph using a **vector** of real numbers which can capture all its "information".

Desired properties:

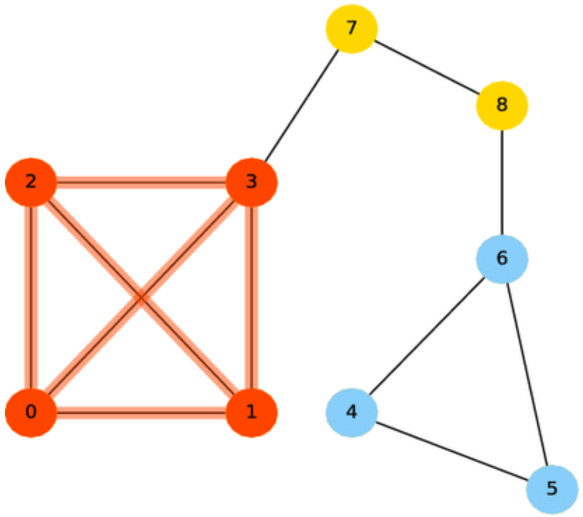
- Task-agnostic (unsupervised)
- Permutation and size invariant
- Independent
- Multi-scale
- Band-pass
- Attributed
- Scalable

Related work

	FGSD (NIPS '17) NetLSD (KDD '18)	DOS kernel (SDM '21)	Prop Kernel (ML '16)	GCN (ICLR '17) GIN (ICLR '19)	ChebNet (NIPS '16) CaleyNet (Sig Proc '19)	A-DOGE
Unsupervised	✓	✓	✓			✓
Independent	✓					✓
Multi-scale	✓	✓			✓	✓
Band-pass		✓			✓	✓
Node attributes			✓	✓	✓	✓
Edge Weights		✓	✓	✓	✓	✓
Scalable	✓			✓	✓	✓

Graph spectrum

- G(V,E,X): Node attributed graph



Vertex	a	b	c	d
0	0.6	1	0	1
1	1.5	1	0	0
2	2.2	1	0	1
3	1.4	1	1	1
4	2.5	0	0	0
5	4.7	0	0	0
6	4.3	0	0	0
7	3.6	1	1	1
8	3.8	1	1	0

Attributes

1	0	0
1	0	0
1	0	0
1	0	0
0	1	0
0	1	0
0	0	1
0	0	1
0	0	1

One-hot enc. label

- **S** = Symmetrically normalized adjacency matrix: $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$

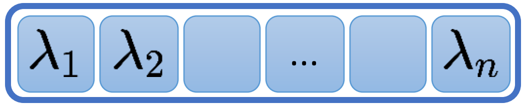
Adjacency Matrix

(Diagonal) Degree Matrix

- Eigendecomposition: $S = V \Lambda V^T$

$$\underbrace{\begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix}}_{\text{Eigen vectors}} \underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}}_{\text{Eigen values}}$$

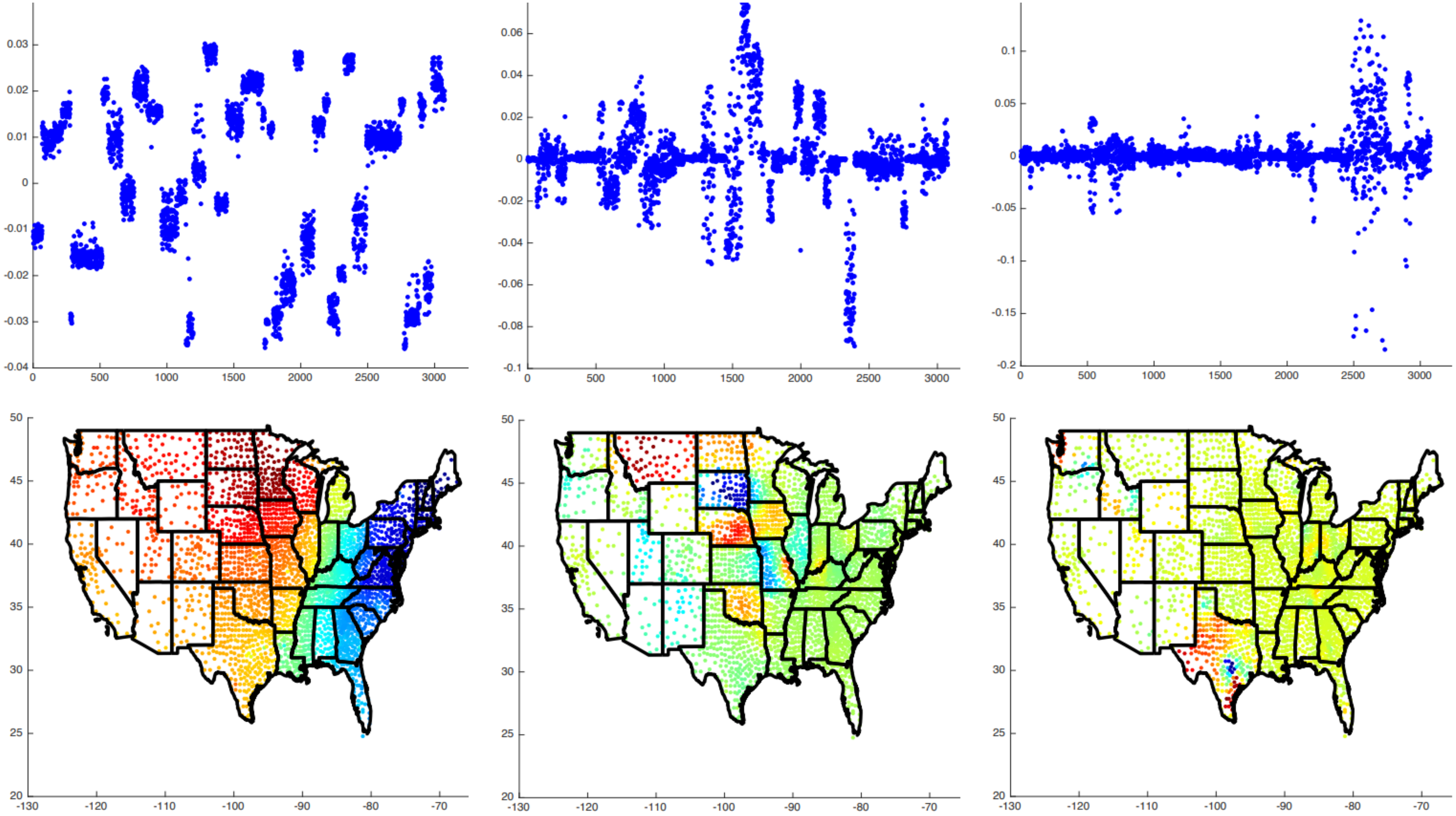
➢ Using full Graph Spectrum as embedding



Desired properties:

- Task-agnostic (unsupervised)
- Independent
- Multi-scale
- Permutation and size invariant
- Band-pass
- Attributed
- Scalable

Eigenspectrum is multi-scale



(a) 2nd, *macro* (b) 41st, *meso* (c) 128th, *micro*

- ➢ Using full Graph Spectrum as embedding

λ_1	λ_2		...		λ_n

Desired properties:

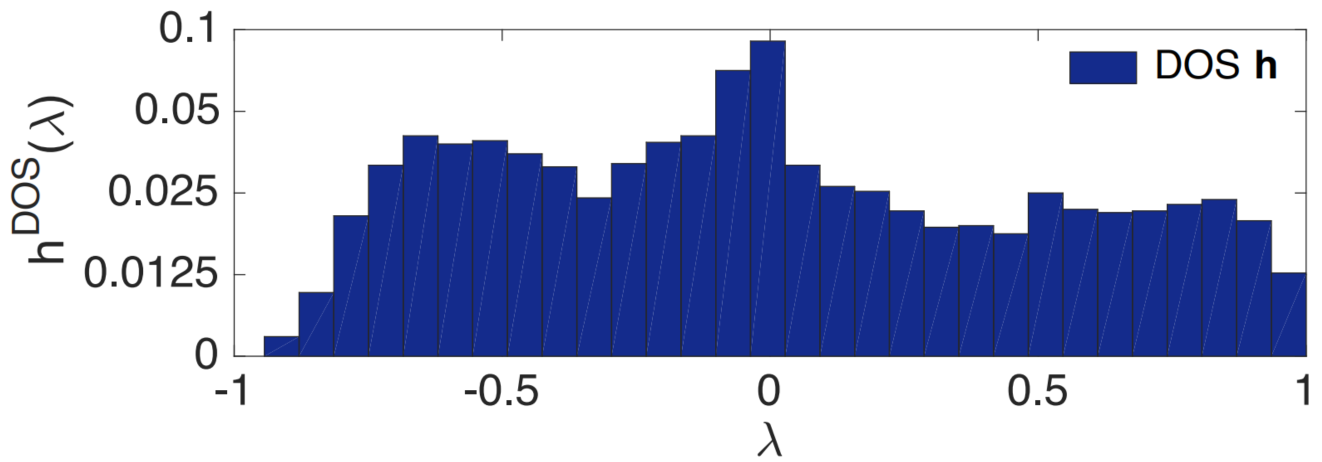
- Task-agnostic (unsupervised) ✓
- Independent ✓
- Multi-scale ✓
- Permutation and size invariant
- Band-pass
- Attributed
- Scalable

Density of States

- Why not use the spectrum? - the size depends on graph size!

- Solution: **Density of States Histogram**
  - i.e., fraction of eigenvalues in each bin
  - K. Dong, A. R. Benson, and D. Bindel, “Network density of states,” KDD '19
  - L. Huang, A. J. Graven, and D. Bindel, “Density of states graph kernels,” SDM '21

$$\mathrm{h}^{DOS}(\lambda) = \frac{\sum_{j=1}^{n} \mathbb{I}(\lambda_j \in \mathrm{bin}_\lambda)}{n}$$



- Using **DOS Histogram** as embedding

$$h^{DOS}(\lambda)$$

Desired properties:

- Task-agnostic (unsupervised)
- Independent
- Multi-scale
- Permutation and size invariant
- Band-pass
- Attributed
- Scalable

Capturing any "band" of eigenvalues

- The **DOS histogram** helps capture all parts of the eigenspectrum

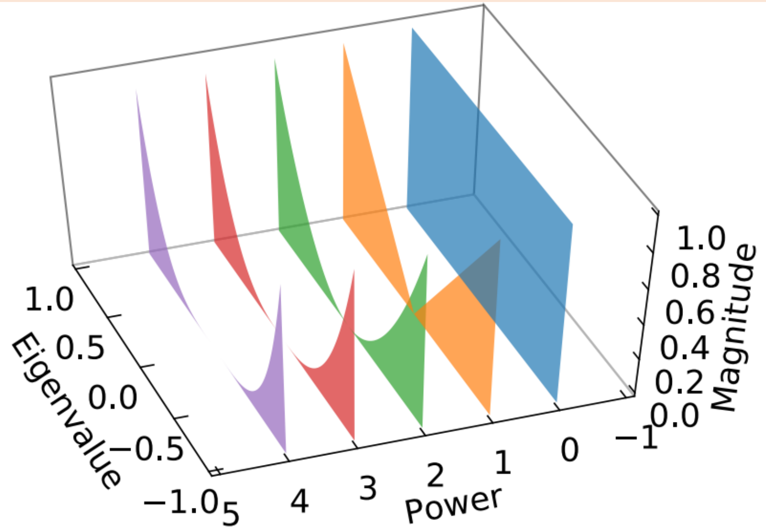
- **Frequency Response Filters**: useful to "select" a part of the spectrum
  - e.g.: low-pass/band-pass/high-pass filters
  - more generally, each eigenvalue λ_i is assigned a scalar $\phi(\lambda_i)$
  - filter output = $\mathrm{h}^{DOS}(\lambda) \cdot \phi(\lambda) \; = \; \sum_{i=1}^{\#\text{bins}} \mathrm{h}^{DOS}(\lambda_i)\phi(\lambda_i)$

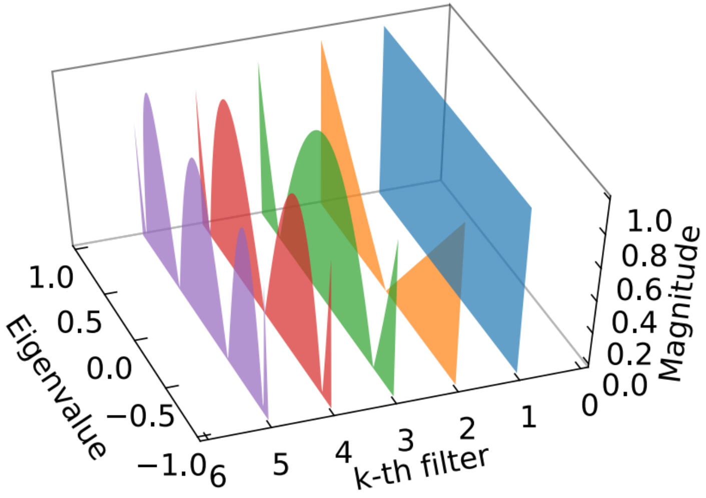
Aggregate functions/filters

- Can we supplement each histogram with **FRFs** to make a better embedding?

Exploratory analysis:
- High interpretability
- **Power functions** as filters

Classification tasks:
- High expressivity
- **Chebyshev polynomials** as filters

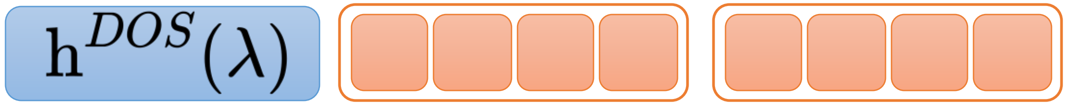




➢Embedding:

DOS Histogram + Filters

hist	agg. g_ϕ	
	Cheb.	Pow.



Each one is a separate filter output

Desired properties:

- Task-agnostic (unsupervised)
- Independent
- Multi-scale
- Permutation and size invariant
- Band-pass
- Attributed
- Scalable

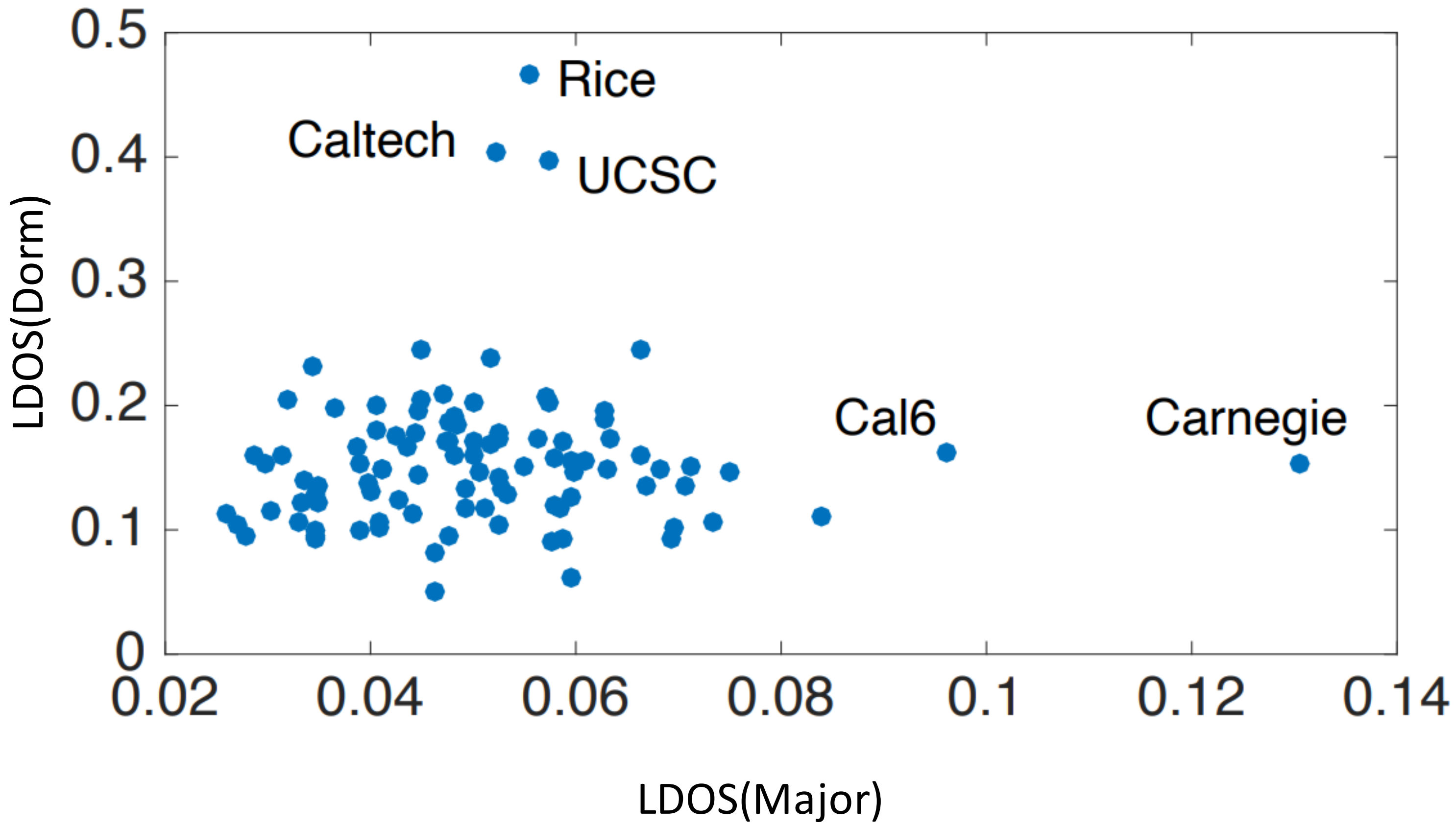
Incorporating Attributes

- DOS histogram: equal weight to each eigenvalue
- Local Density of States (**LDOS**): $$\mathrm{h}^{LDOS}(\lambda, x) = \frac{\sum_{i=1}^{n} (x \cdot v_i)^2 \mathbb{I}(\lambda_i \in \mathrm{bin}_\lambda)}{n}$$
  - Given attribute vector x: $(x \cdot v_i)^2$ represents the weight of λ_i
  - Models the **alignment** between attribute and the structure captured by v_i
  - Related notion – **PDOS** (x is only allowed to be indicator vector)
    - K. Dong, A. R. Benson, and D. Bindel, “Network density of states,” KDD '19
    - L. Huang, A. J. Graven, and D. Bindel, “Density of states graph kernels,” SDM '21

LDOS Aggregate functions: $$\mathrm{h}^{LDOS}(\lambda, x) \cdot \phi(\lambda) \approx x^T \phi(S) x$$

Application I: exploratory graph mining

- Alignment of Facebook friendships in colleges w.r.t. dorm and major
- Using LDOS aggregate features (power=1)



Attribute pairs

- Attribute pairs – Coupled LDOS (cLDOS):
  - Given **two** attribute vectors x and y: $(x \cdot v_i)(y \cdot v_i)$ is the weight of λ_i

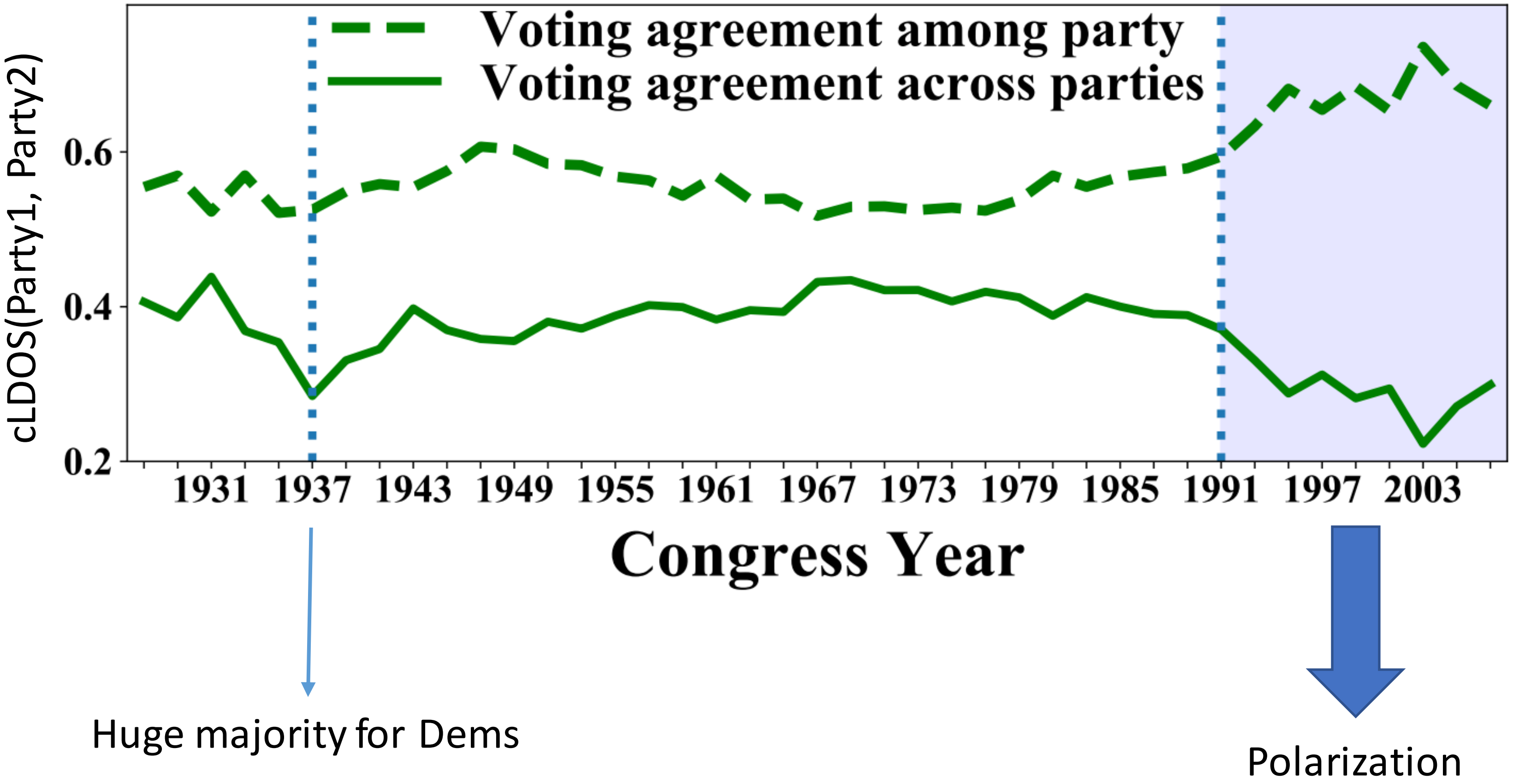
$$\mathrm{h}^{cLDOS}(\lambda, x, y) = \frac{\sum_{i=1}^{n}(x \cdot v_i)(y \cdot v_i)\mathbb{I}(\lambda_i \in \mathrm{bin}_\lambda)}{n}$$

cLDOS Aggregate functions: $\mathrm{h}^{cLDOS}(\lambda, x, y) \cdot \phi(\lambda) \approx x^T \phi(S) y$

Application I: exploratory graph mining

- Voting agreement between Rep and Dem senators

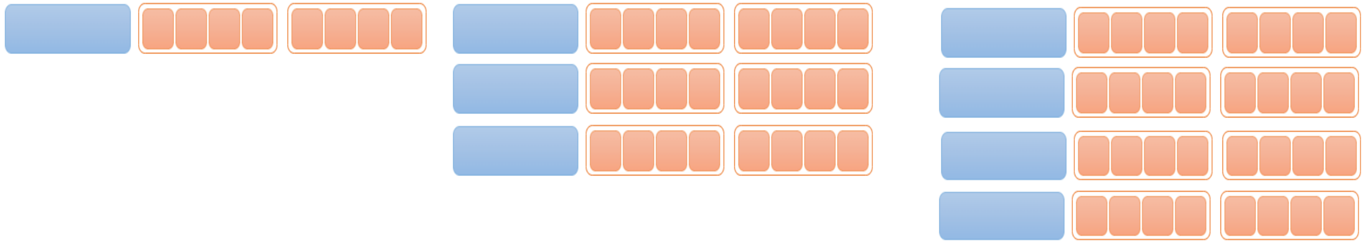
- Using cLDOS aggregate feature (power=1)



Attributed-DOS-based Graph Embedding

- Our graph embedding:

DOS			LDOS			cLDOS		
hist	agg. g_ϕ		hist	agg. g_ϕ		hist	agg. g_ϕ	
	Cheb.	Pow.		Cheb.	Pow.		Cheb.	Pow.
B	K	K	BD	KD	KD	$B\binom{D}{2}$	$K\binom{D}{2}$	$K\binom{D}{2}$



- D = number of node attributes
- B = number of bins in histogram
- K = number of aggregate functions (filters)

Desired properties:

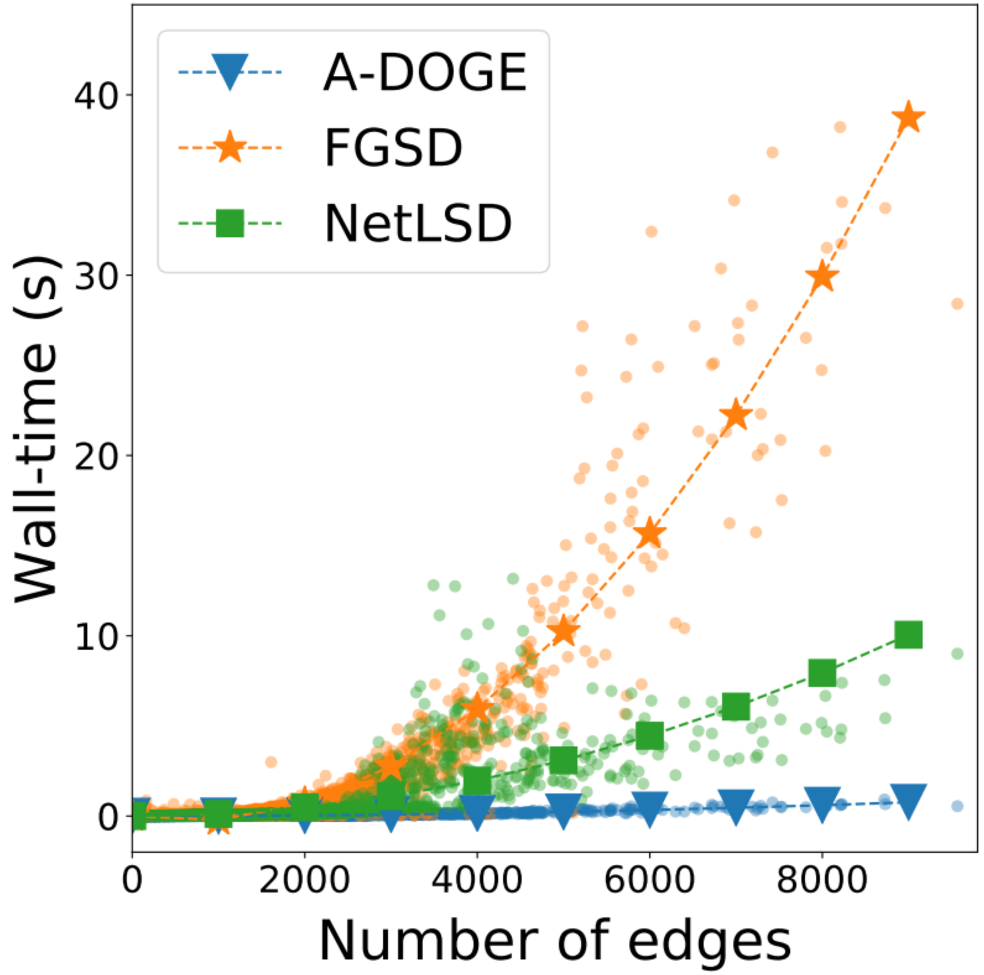
- Task-agnostic (unsupervised)
- Independent
- Multi-scale
- Permutation and size invariant
- Band-pass
- Attributed
- Scalable

Scalability

- Usually, eigendecomposition is **slow**!
- But we only need spectral density histograms

- (Dong, Benson, Bindel – Network Density of States KDD '19)
  - shows how to (approximately) compute DOS and LDOS fast!

Scalability Experiment



- Comparison with the next fastest competitors
- GNNs famously need a lot more resources, and training labels
- Graph Kernels cannot compute each embedding independently

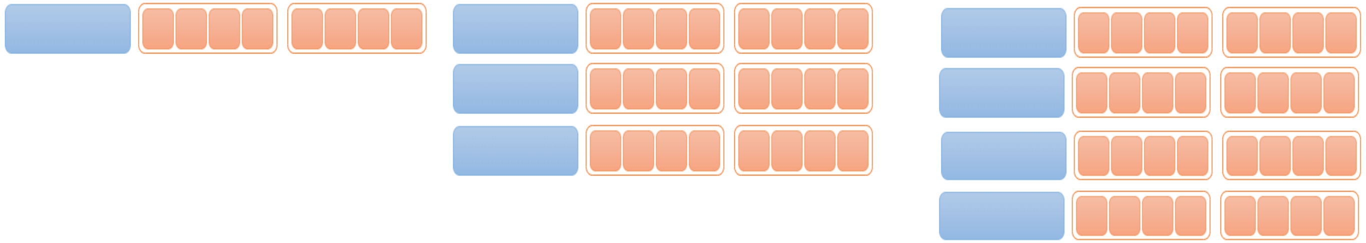
Application II: classification tasks

Benchmark	Graph Embedding (Unsupervised)					Graph Kernels (Unsupervised)				GNNs (Supervised)		
	A-DOGE	DOGE	FGSD	NETLSD	G2VEC	WL	WL-OA	PK	DOSGK	CHEBNET	GCN	GIN
RED-B	91.6 (1.5)	90.3 (1.8)	82.4 (2.6)	85.6 (2.2)	74.2 (2.7)	83.9 (0.5)‡	88.9 (0.1)‡	85.5 (0.3)‡	88.8 (0.3)*	90.2 (2.0)	89.9 (2.0)	**91.7** (1.6)
RED-5K	**55.6** (2.2)	53.8 (2.1)	47.0 (1.8)	45.9 (2.1)	41.5 (1.6)	51.2 (0.3)*	E	E	52.8 (0.2)*	55.0 (2.2)	54.2 (1.7)	54.7 (2.0)
COLLAB	72.2 (2.0)	72.2 (2.0)	70.2 (1.8)	68.4 (1.9)	57.9 (1.5)	74.8 (0.2)*	79.8 (1.6)	77.8 (1.7)	80.8 (0.2)*	**84.6** (1.1)	84.2 (1.2)	83.8 (1.6)
IMDB-B	72.6 (4.3)	71.6 (4.3)	70.6 (4.1)	69.7 (4.1)	56.0 (4.1)	71.3 (1.0)‡	73.5 (0.6)	71.2 (0.7)‡	72.8 (0.9)*	80.2 (3.9)	79.9 (3.7)	**80.8** (4.5)
IMDB-M	47.8 (3.5)	47.6 (3.7)	48.6 (3.4)	47.9 (3.7)	44.4 (3.8)	50.7 (0.6)‡	50.7 (0.5)‡	51.0 (0.7)‡	49.4 (0.5)*	55.6 (2.7)	55.2 (2.7)	**56.3** (3.1)
DD	80.1 (3.5)	76.2 (3.4)	76.5 (3.5)	76.6 (3.5)	76.2 (3.5)	80.9 (0.3)	79.9 (0.5)	**81.6** (0.5)	73.4 (3.7)	78.9 (1.9)	78.0 (1.8)	79.3 (1.9)
PROTN	74.9 (3.5)	74.9 (3.5)	74.2 (3.3)	74.5 (4.0)	72.1 (3.1)	73.9 (0.7)‡	75.9 (0.6)‡	74.6 (0.5)‡	72.1 (3.9)	78.3 (2.7)	76.7 (3.5)	**78.4** (3.9)
AIDS	**99.8** (0.3)	99.8 (0.3)	99.6 (0.4)	99.6 (0.5)	98.8 (0.7)	99.7 (0.0)‡	99.7 (0.0)‡	99.7 (0.0)‡	99.1 (0.7)	96.9 (1.6)	95.5 (1.3)	98.6 (0.6)
Cong	**99.5** (1.5)	54.7 (11.0)	95.1 (4.3)	99.5 (1.5)	86.8 (7.4)	84.8 (7.3)	81.1 (7.7)	68.6 (8.3)	60.0 (10)	50.0 (0.0)	50.0 (0.0)	57.0 (5.9)
Cong-l	**78.0** (8.6)	58.9 (10.0)	50.0 (0.0)	60.4 (9.7)	59.8 (11)	62.2 (10)	62.3 (10)	58.2 (10)	55.7 (9.7)	50.0 (0.0)	50.0 (0.0)	71.5 (9.4)
MIG	**100.0** (0)	62.3 (9.7)	99.5 (1.5)	99.9 (1.1)	50.0 (0.0)	99.8 (1.4)	99.8 (1.4)	100 (0.0)	53.5 (12)	100.0 (0.0)	78.5 (1.7)	100.0 (0.0)
BPass	90.8	51.9	47.9	51.4	50	50	51.6	70.4	48.5	**98.2**†	77.9†	87.6†
Avg.	82.5	69.1	74.1	75.8	66.0	75.6	76.6	76.3	68.6	78.4	74.2	80.5

Attributed-DOS-based Graph Embedding

- Our graph embedding:

DOS			LDOS			cLDOS		
hist	agg. g_ϕ		hist	agg. g_ϕ		hist	agg. g_ϕ	
	Cheb.	Pow.		Cheb.	Pow.		Cheb.	Pow.
B	K	K	BD	KD	KD	$B\binom{D}{2}$	$K\binom{D}{2}$	$K\binom{D}{2}$



- D = number of node attributes
- B = number of bins in histogram
- K = number of aggregate functions (filters)

Desired properties:

- Task-agnostic (unsupervised)
- Independent
- Multi-scale
- Permutation and size invariant
- Band-pass
- Attributed
- Scalable

Thank you!

- Paper is available at: arxiv.org/abs/2110.05228
- Code is available at: github.com/sawlani/A-DOGE
- E-mail me for questions: saurabh.sawlani@gmail.com