End-to-End Learning of Probabilistic Hierarchies on Graphs

Daniel Zügner, Bertrand Charpentier, Morgane Ayle,

Sascha Geringer, Stephan Günnemann

ICLR 2022

Technical University of Munich

Real-World Graphs are Often Hierarchical

- Real-world **graphs** are often **hierarchically organized.**
- E.g., in citation networks, subgraphs are **more densely connected** the more specialized they become.
- **Hierarchical clustering** aims to discover the **latent hierarchy** in the data.

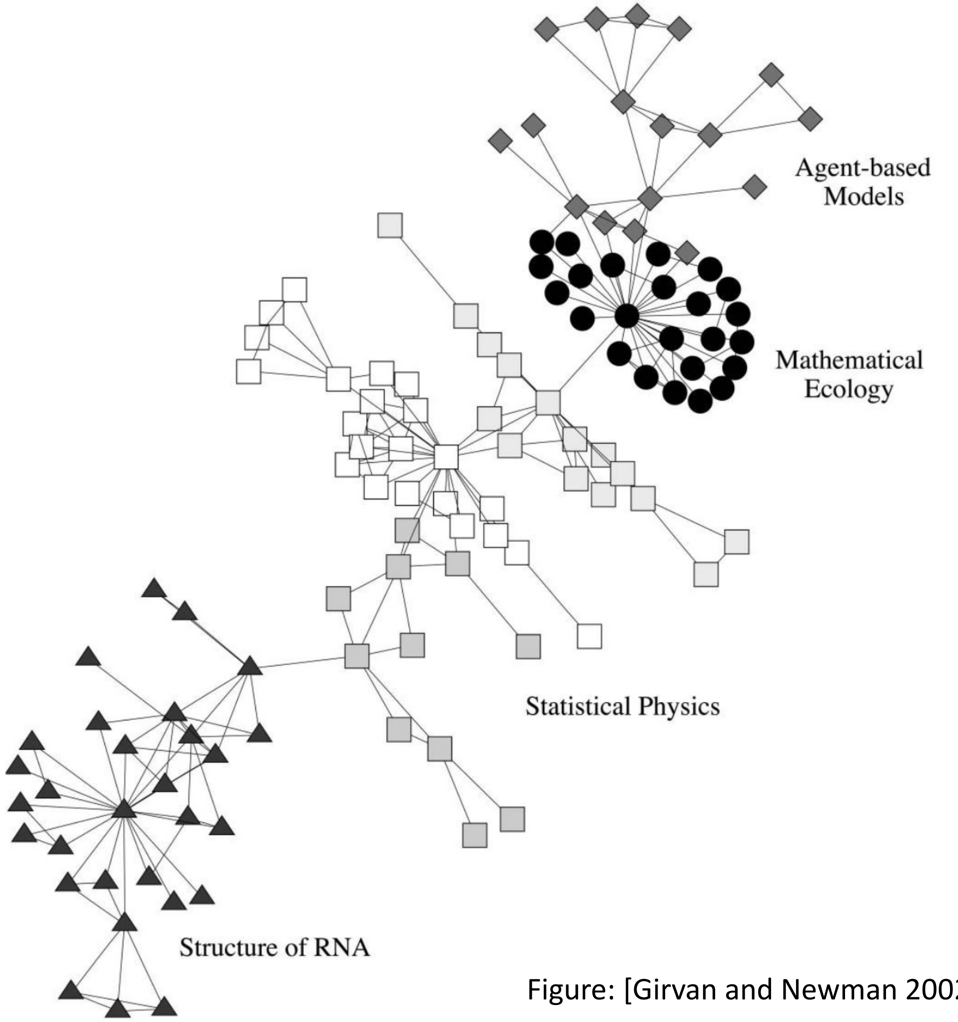


Figure: [Girvan and Newman 2002]

Problem Outline

- Since we typically do not have **ground-truth information** about the underlying hierarchy, we mostly rely on unsupervised learning with **internal metrics**.

- Existing quality metrics are discrete, i.e., **not differentiable**.

- Existing clustering approaches are **mostly heuristic** and not directly related to the **evaluation metrics**.

Contributions

1. We propose a **probabilistic model over hierarchies** via **continuous relaxation** of a tree's parent assignment matrices.

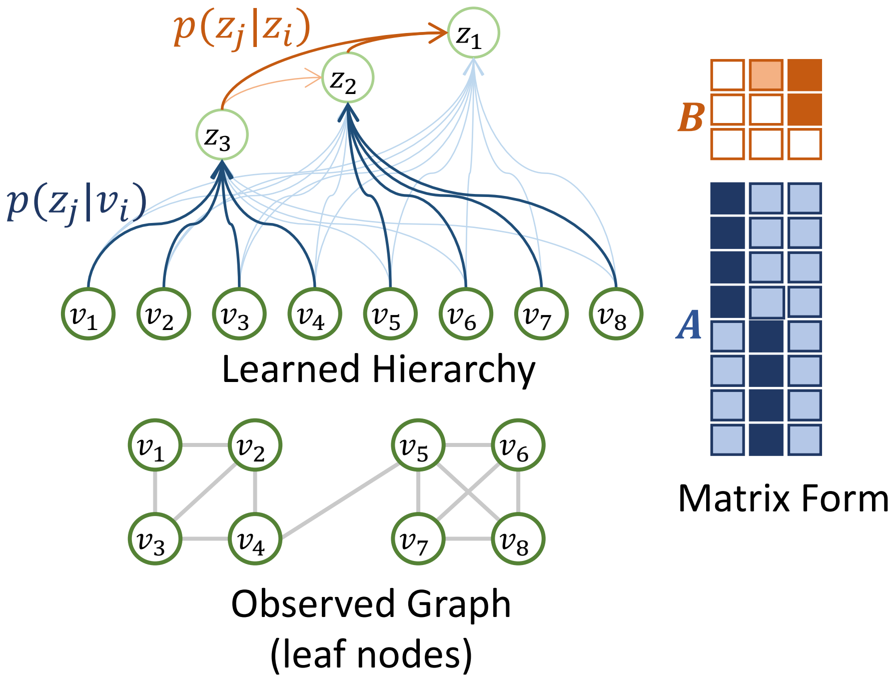
2. We **theoretically analyze** the model by drawing connections to **absorbing Markov chains**, which

3. allows **efficient and exact computation** of **lowest-common-ancestor** (LCA) probabilities, which enables us to

4. learn **hierarchies on graphs** by **end-to-end optimization** of relaxed versions of quality metrics such as **Dasgupta cost** and **TSD score**.

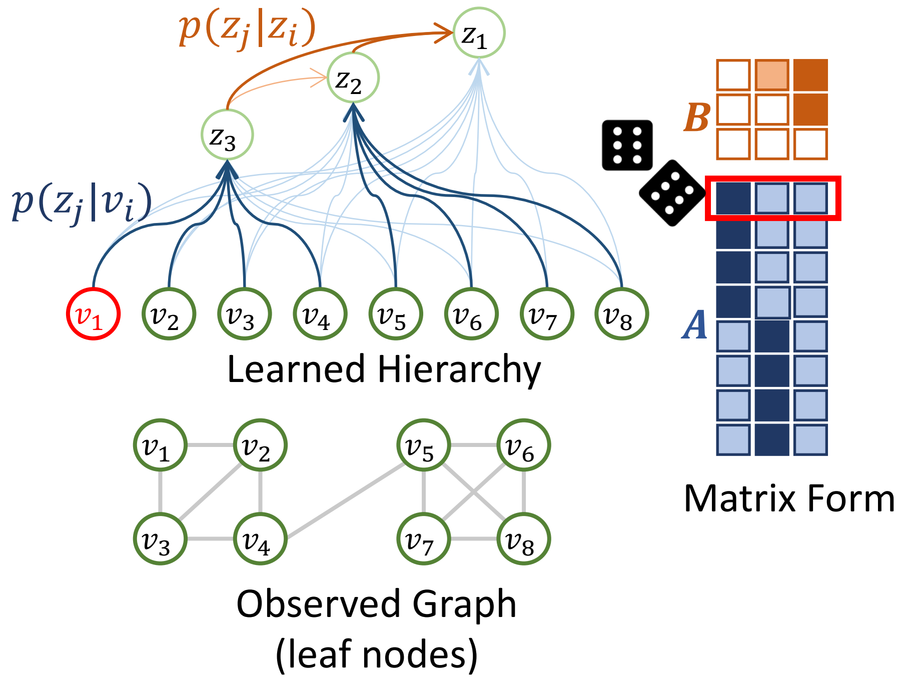
Probabilistic Hierarchy Model

- A **continuous relaxation** of discrete **tree hierarchies**.
- Learnable matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain **parent assignment probabilities** for leaves and internal nodes.
- We can easily obtain **valid tree hierarchies** via row-wise sampling from $\boldsymbol{A}$ and $\boldsymbol{B}$.



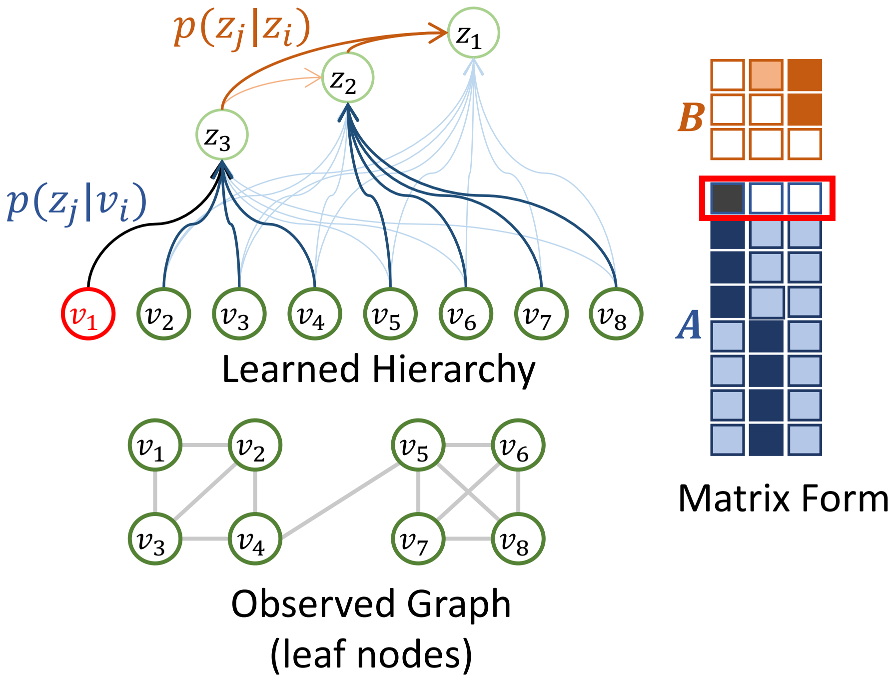
Probabilistic Hierarchy Model

- A **continuous relaxation** of discrete **tree hierarchies**.
- Learnable matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain **parent assignment probabilities** for leaves and internal nodes.
- We can easily obtain **valid tree hierarchies** via row-wise sampling from $\boldsymbol{A}$ and $\boldsymbol{B}$.



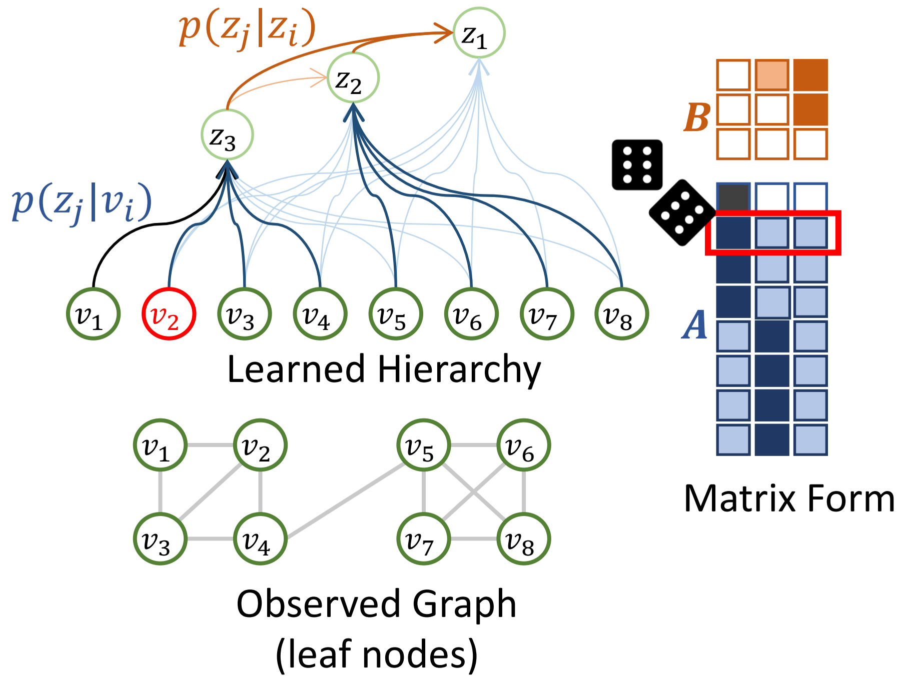
Probabilistic Hierarchy Model

- A **continuous relaxation** of discrete **tree hierarchies**.
- Learnable matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain **parent assignment probabilities** for leaves and internal nodes.
- We can easily obtain **valid tree hierarchies** via row-wise sampling from $\boldsymbol{A}$ and $\boldsymbol{B}$.



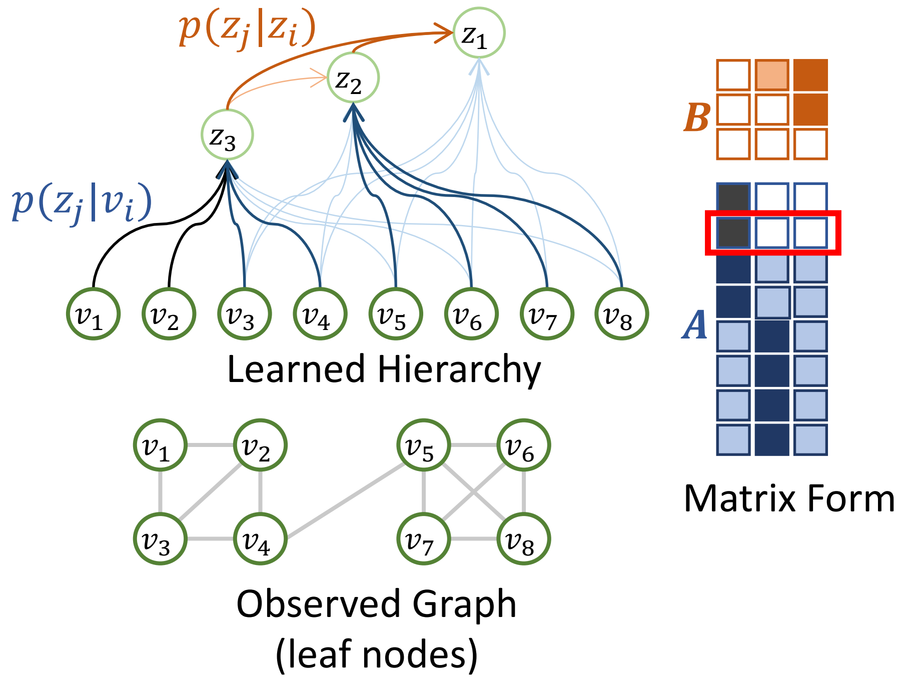
Probabilistic Hierarchy Model

- A **continuous relaxation** of discrete **tree hierarchies**.
- Learnable matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain **parent assignment probabilities** for leaves and internal nodes.
- We can easily obtain **valid tree hierarchies** via row-wise sampling from $\boldsymbol{A}$ and $\boldsymbol{B}$.



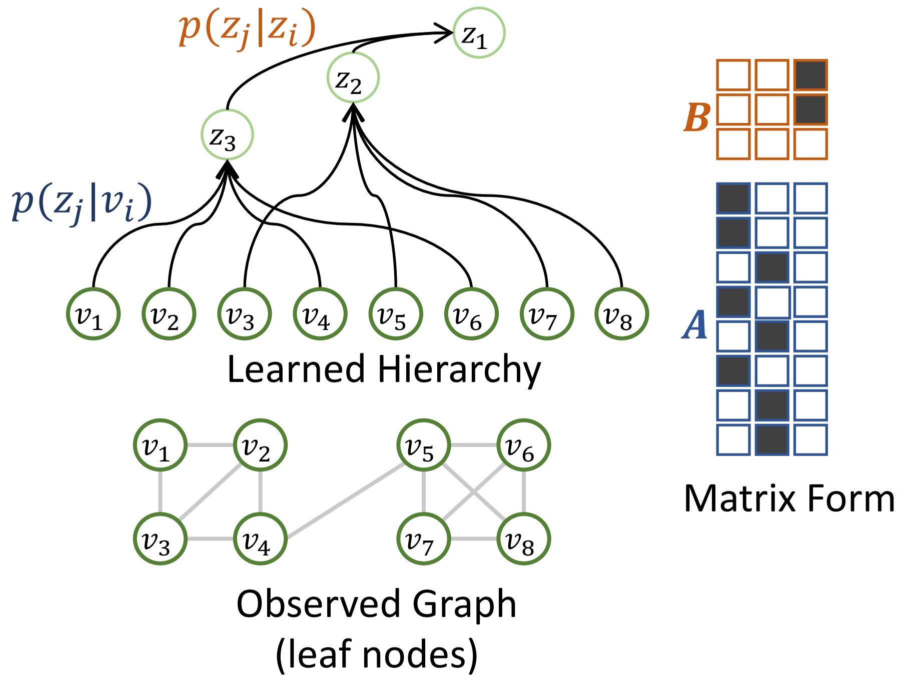
Probabilistic Hierarchy Model

- A **continuous relaxation** of discrete **tree hierarchies**.
- Learnable matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain **parent assignment probabilities** for leaves and internal nodes.
- We can easily obtain **valid tree hierarchies** via row-wise sampling from $\boldsymbol{A}$ and $\boldsymbol{B}$.



Probabilistic Hierarchy Model

- A **continuous relaxation** of discrete **tree hierarchies**.
- Learnable matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain **parent assignment probabilities** for leaves and internal nodes.
- We can easily obtain **valid tree hierarchies** via row-wise sampling from $\boldsymbol{A}$ and $\boldsymbol{B}$.



Hierarchical Clustering Metrics

Two **established internal metrics** for hierarchical clustering:

Dasgupta Cost [Dasgupta 2016]

$$\mathrm{Das}(\hat{\mathcal{T}}) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \cdot \sum_z \mathbb{I}_{[z = v_i \wedge v_j]} c(z)$$

Tree-Sampling Divergence (**TSD**) [Charpentier and Bonald 2019]

$$TSD(\hat{\mathcal{T}}) = \mathrm{KL}(p(z)||q(z)), \text{where}$$

$$p(z) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \mathbb{I}_{[z = v_i \wedge v_j]}$$

$$q(z) = \sum_{v_i, v_j \in V} P(v_i,) P(v_j) \mathbb{I}_{[z = v_i \wedge v_j]}$$

Hierarchical Clustering Metrics

Two **established internal metrics** for hierarchical clustering:

Dasgupta Cost [Dasgupta 2016]

$$\mathrm{Das}(\hat{\mathcal{T}}) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \cdot \sum_{z} \mathbb{I}_{[z = v_i \wedge v_j]} c(z)$$

Tree-Sampling Divergence (**TSD**) [Charpentier and Bonald 2019]

$$TSD(\hat{\mathcal{T}}) = \mathrm{KL}(p(z)||q(z)), \text{where}$$

$$p(z) = \sum_{v_i, v_j \in \mathcal{E}} P(v_i, v_j) \mathbb{I}_{[z = v_i \wedge v_j]}$$

$$q(z) = \sum_{v_i, v_j \in V} P(v_i,) P(v_j) \mathbb{I}_{[z = v_i \wedge v_j]}$$

Lowest common ancestors (LCAs) occur in both metrics.

Hierarchical Clustering Metrics: Relaxation

Relaxed Dasgupta Cost

$$\text{Soft–Das}(\hat{\mathcal{T}}) = \sum\nolimits_{v_i,v_j \in \mathcal{E}} P(v_i, v_j) \cdot \sum\nolimits_z P(z = v_i \land v_j) c(z)$$

Relaxed Tree-Sampling Divergence (**TSD**)

$$\text{Soft–}TSD(\hat{\mathcal{T}}) = \mathrm{KL}(p(z)||q(z)), \text{where}$$

$$p(z) = \sum\nolimits_{v_i,v_j \in \mathcal{E}} P(v_i, v_j) P(z = v_i \land v_j)$$

$$q(z) = \sum\nolimits_{v_i,v_j \in V} P(v_i,) P(v_j) P(z = v_i \land v_j)$$

Indicators replaced by expectations (probabilities) $\mathbb{I}_{[z=v_i \land v_j]}$ to $P(z = v_i \land v_j)$

Efficient Computation of LCA Probabilities

- Computation of **LCA probabilities** given matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is nontrivial.

- **Key result**: we draw connections to **absorbing Markov chains** to derive **efficient closed-form matrix equations** to compute **LCA probabilities**:

Efficient Computation of LCA Probabilities

- Computation of **LCA probabilities** given matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is nontrivial.
- **Key result**: we draw connections to **absorbing Markov chains** to derive **efficient closed-form matrix equations** to compute **LCA probabilities**:

Theorem 6. *The vector of LCA probabilities of all internal nodes w.r.t. leaf nodes* v_i *and* v_j *can be computed in a vectorized way via*

$$\boldsymbol{P}^{LCA}_{\mathrm{v}_i \neq \mathrm{v}_j} \in \mathbb{R}^{n'} = (\boldsymbol{P}^{anc}_{\mathrm{v}_i} \odot \boldsymbol{P}^{anc}_{\mathrm{v}_j})^T \cdot (\boldsymbol{I} + \tilde{\boldsymbol{P}}^{anc} \odot \tilde{\boldsymbol{P}}^{anc})^{-1} \qquad \boldsymbol{P}^{LCA}_{\mathrm{v}_i, \mathrm{v}_i} = \boldsymbol{A}_{\mathrm{v}i}, \tag{9}$$

where $\odot$ *denotes the element-wise (Hadamard) product. (See proof in App. A.9)*

Complexity for the whole graph: $O(\#internal\_nodes^2 \cdot \#edges)$

Results

Alg.	Dasgupta cost (lower is better) Ward	Louv.	UF	HypHC	HGHC	RGHC	Avg. lk.	FPH	Normalized TSD (higher is better) Ward	Louv.	UF	HypHC	HGHC	RGHC	Avg. lk.	FPH
Brain	618.81	777.14	712.33	571.64	749.40	556.57	556.64	**503.67**	31.72	29.28	28.61	17.48	24.18	22.05	28.88	**32.34**
OpenFlight	382.45	633.66	393.58	463.43	487.96	488.90	363.40	**355.61**	55.48	51.51	53.89	39.08	49.50	39.56	52.05	**57.72**
Genes	202.17	247.26	251.01	495.26	366.53	247.07	196.51	**183.63**	66.80	67.47	62.95	20.66	53.33	51.81	66.68	**67.69**
Citeseer	92.27	178.23	98.61	215.62	150.26	131.89	84.04	**77.16**	69.43	68.45	67.40	37.22	57.61	50.61	67.99	**69.57**
Cora-ML	281.82	336.86	342.86	442.09	411.49	350.00	297.03	**254.78**	56.47	57.51	53.06	30.73	46.76	42.68	55.41	**58.02**
PolBlogs	377.63	443.48	350.74	330.58	354.86	433.77	364.14	**262.48**	27.54	25.93	25.23	22.21	23.94	19.41	25.29	**31.41**
WikiPhysics	736.11	986.32	753.81	759.07	840.15	740.87	658.04	**537.95**	45.28	46.03	43.40	32.02	39.70	38.39	43.23	**49.97**
ogbn-arxiv	22,870	31,655	52,666	OOM	22,076	24,077	20,760	**14,354**	36.77	37.75	24.75	OOM	26.05	25.21	33.70	**39.66**
ogbl-collab	13,835	20,664	91,807	OOM	34,934	21,057	15,714	**13,493**	45.33	46.12	27.90	OOM	24.80	34.07	45.40	**48.36**
DBLP	**31,138**	40,744	148,439	OOM	94,384	44,424	36,463	31,686	38.26	40.92	20.21	OOM	15.96	27.82	38.97	**41.66**

Table 1: Hierarchical clustering results ($n' = 512$). Bold/underline indicate best/second best scores.

Dataset	FPH	DCSBM	DW	Ad./Ad.	VGAE
Cora-ML	95.7	95.5	94.3	86.5	**95.9**
Citeseer	**96.2**	93.6	96.0	76.8	94.8
PolBlogs	94.3	**94.9**	84.8	92.6	92.8
WikiPhysics	**97.2**	96.9	92.9	96.6	97.0
Brain	94.1	**95.2**	83.8	90.7	93.2
OpenFlight	**99.3**	99.0	94.3	98.4	99.0
Genes	69.8	66.9	**70.3**	53.0	66.6

Table 4: AUC-PR score (%) for link prediction.

Paper, Code & More:
https://www.daml.in.tum.de/fph