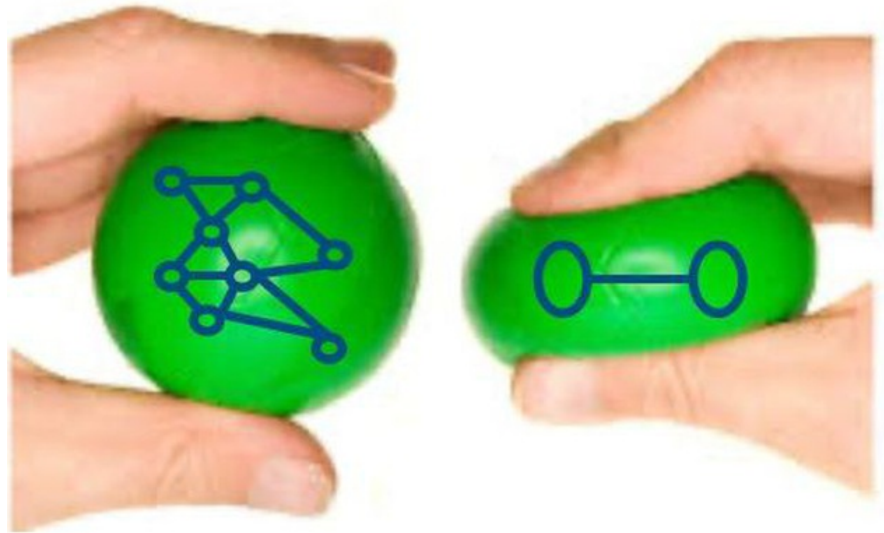
Partition and Code:

Learning how to Compress Graphs

Giorgos Bouritsas, Andreas Loukas, Nikolaos Karalias, Michael Bronstein

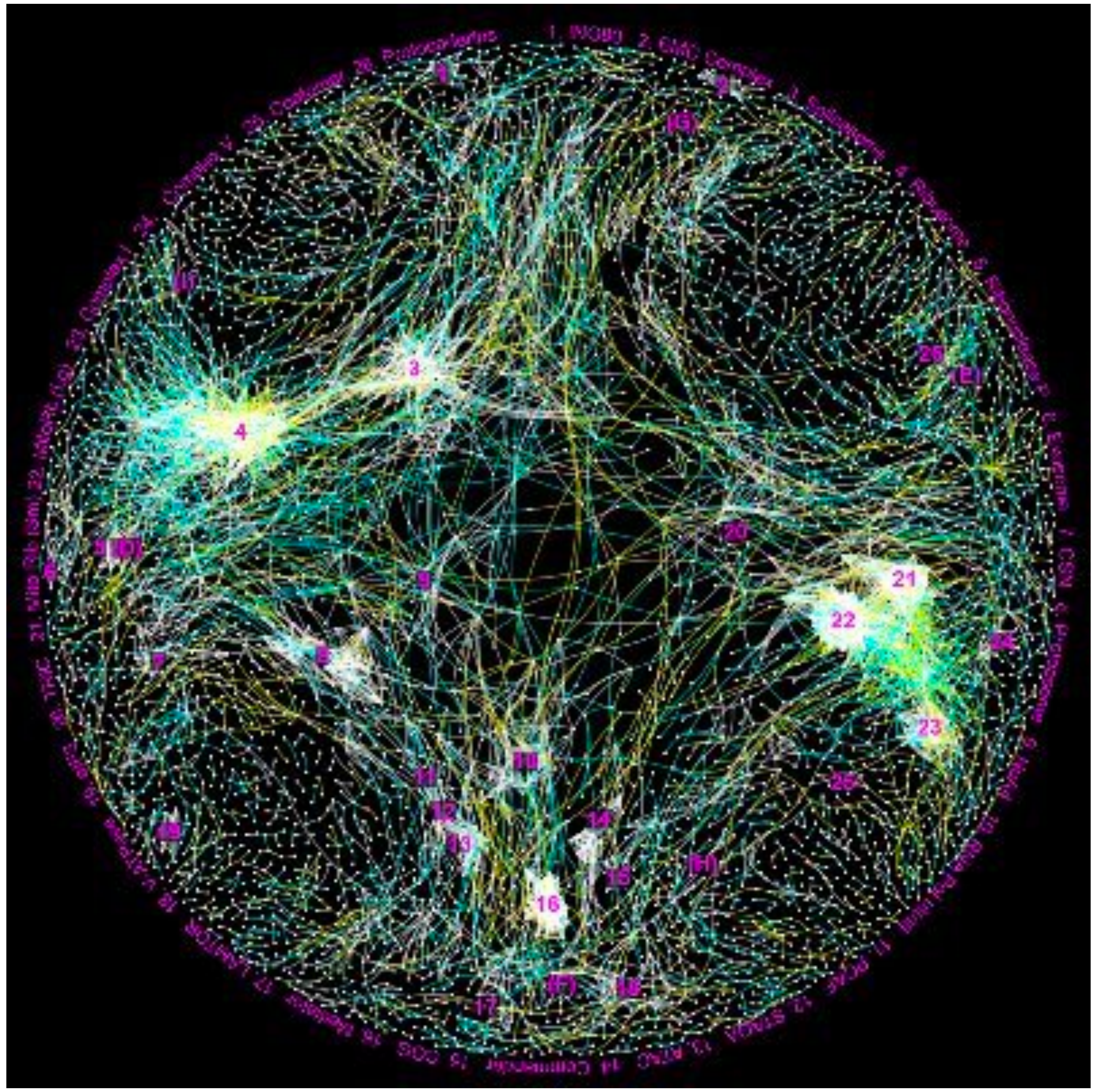
Imperial College London

EPFL

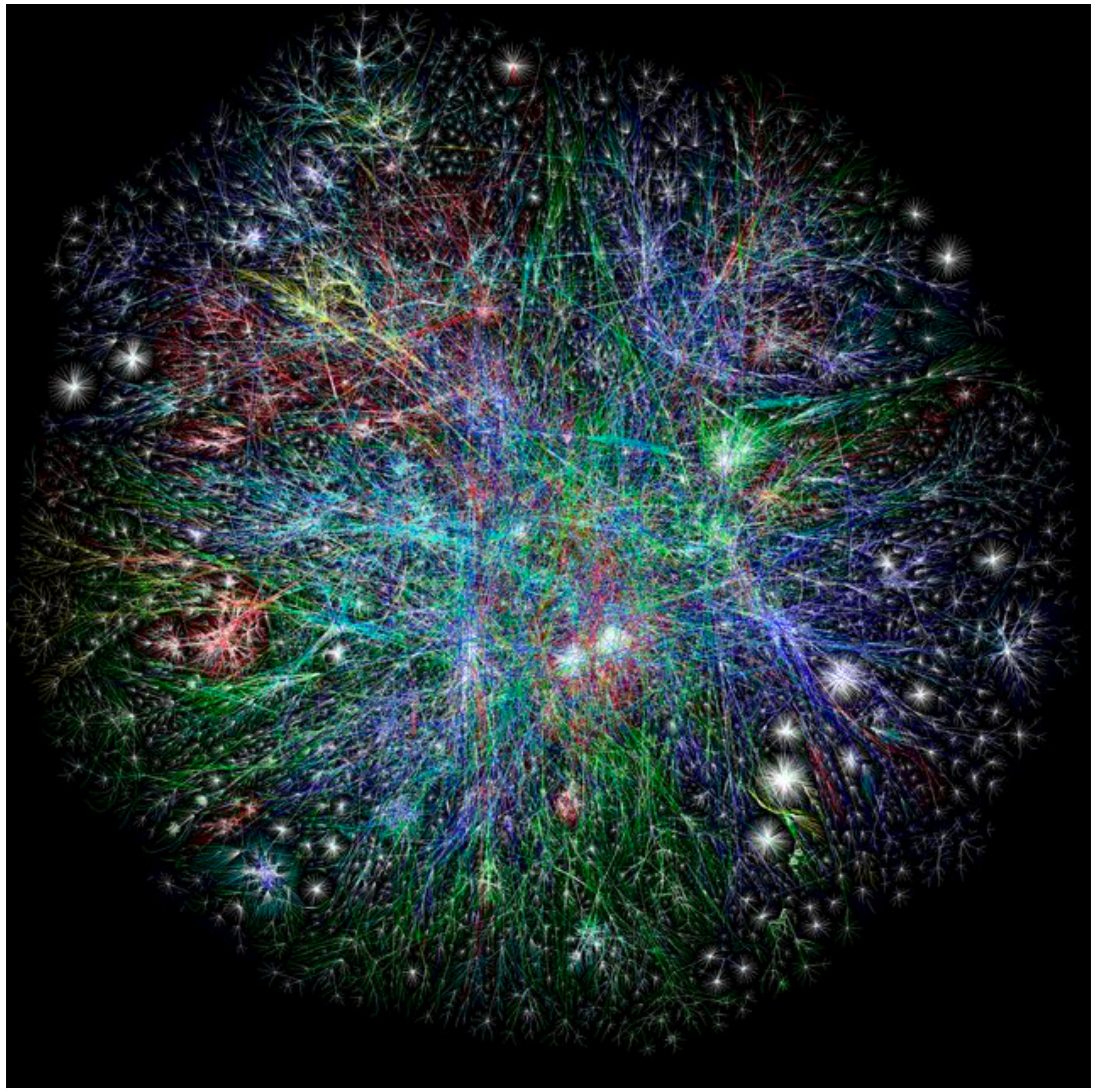


Why do we need to compress graphs?

Huttlin et al., Cell'21



Human Interactome



Internet

https://www.opte.org/

How to transmit/store/process all this information?

From a theorist's point of view: fundamental problem in computer science

What is the role of machine learning in compression?

Lossless Graph Compression

$$G_1 \cong G_2$$

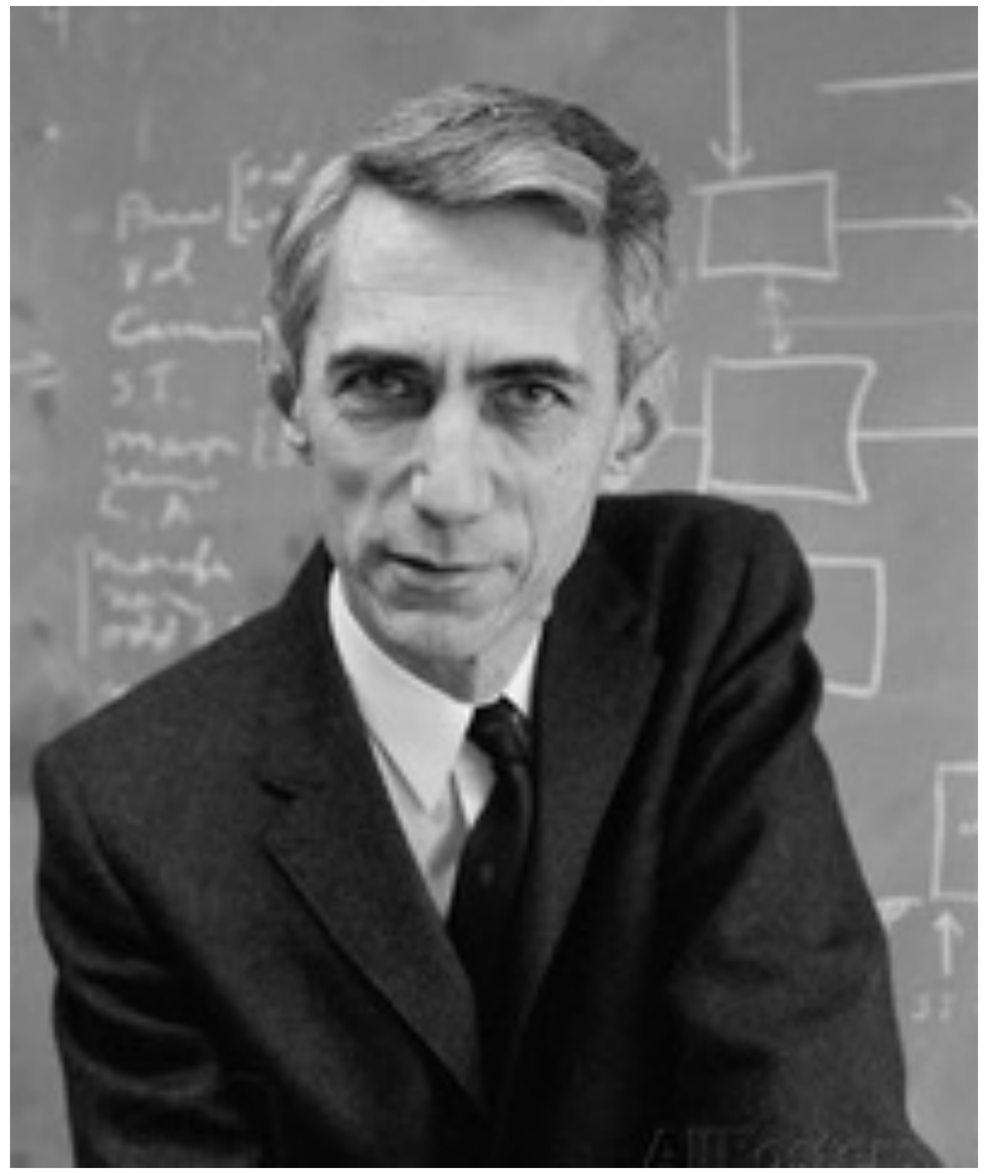
CODE needs to be **uniquely decodable**, i.e., each codeword should be mapped to a unique “source symbol”

Here: source symbol = isomorphism class, not labelled graph

Information theory basics

- Observation space $\mathfrak{G}$
- Probability distribution p
- **Objective:** find a description method that minimises the expected description length

$$\min_{\text{CODE}} \mathbb{E}_{G \sim p}[\mathrm{L}_{\text{CODE}}(G)]$$



***Shannon's source coding theorem (informal)**: For all description methods* CODE *it holds that:*

$$\mathbb{E}_{G \sim p}[\mathrm{L}_{\text{CODE}}(G)] \geq \mathrm{H}_{G \sim p}[G],$$

Where $\mathrm{H}_{G \sim p}[G] = -\sum_{G \in \mathfrak{G}} p(G) \log p(G)$ is the *Entropy* of the r.v. G

Information theory basics

$$\mathrm{L}_{\mathsf{CODE}}(G) = -\log p(G) \Leftrightarrow \text{ optimal CODE}$$

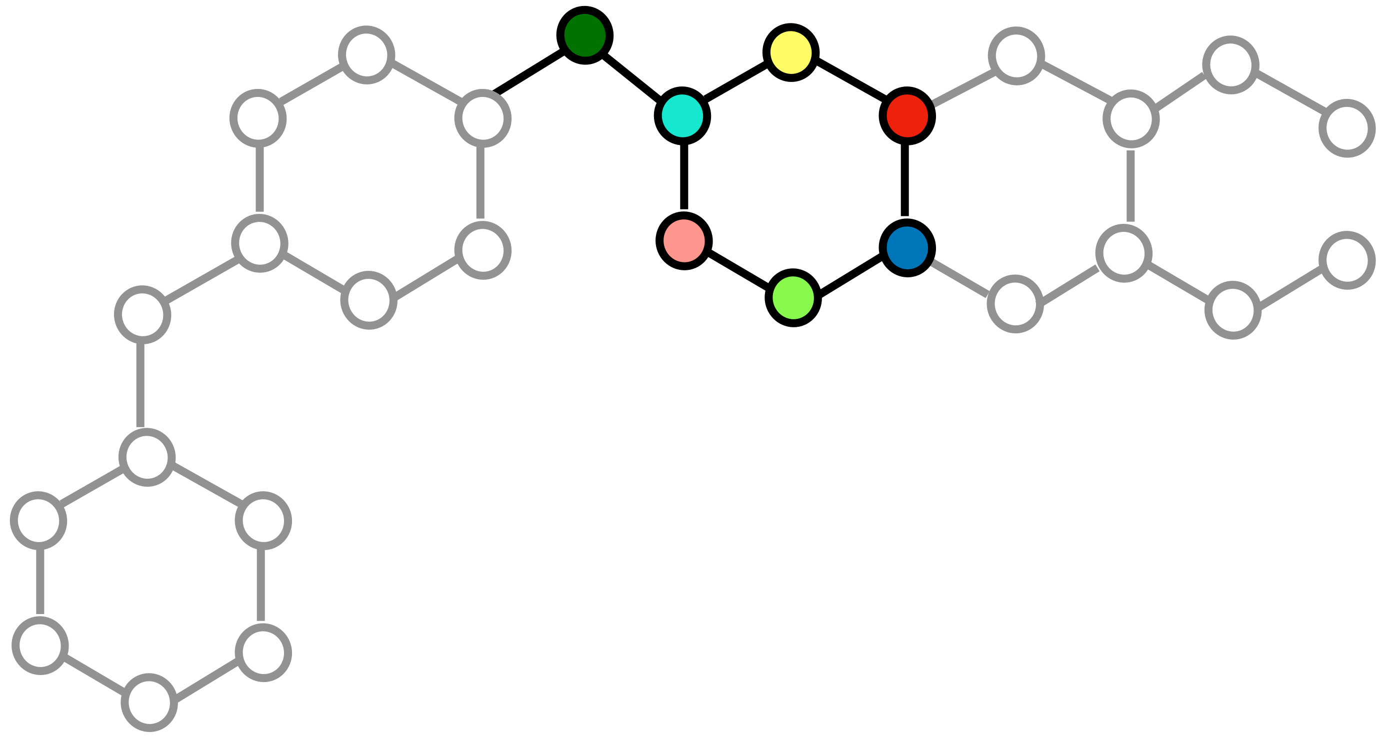
The compression problem amounts to estimating the probability distribution.

- Optimise for **probability distributions** instead of description methods

$$\min_q \mathbb{E}_{G \sim p}[-\log q(G)]$$

- Every distribution q can be converted to a uniquely decodable code (Kraft-McMillan inequality) using an entropy coder (Huffman coding, arithmetic coding, ANS,…).

Warmup: Simple uninformative encodings



Adjacency matrix $\xrightarrow{\textit{Encoder}}$

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

C = ...01000001010001...

$$\mathrm{L}_C(G) = n^2$$

$$q(G) = \frac{1}{2^{n^2}}$$

Erdos-Renyi $\xrightarrow{\textit{Encoder}}$

m = 14

$\mathscr{E}$ = {(1,2), (2,1), (2,3), (2,7), (3,2), (3,4), (4,3), (4,5), (5,4), (5,6), (6,5), (6,7), (7,2), (7,6)}

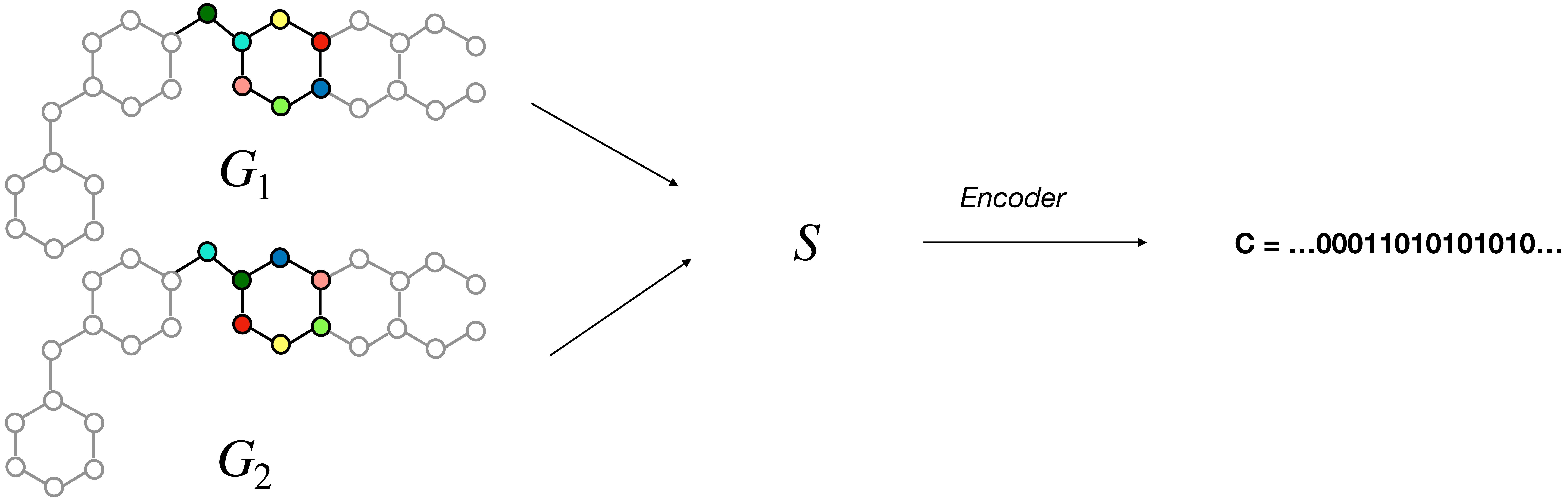
C = ...101000111000...

$$\mathrm{L}_C(G) = \log(n^2+1) + \log\binom{n^2}{m}$$

$$q(G) = \frac{1}{n^2+1}\frac{1}{\binom{n^2}{m}}$$

Challenge 1: The usual suspect - Isomorphism

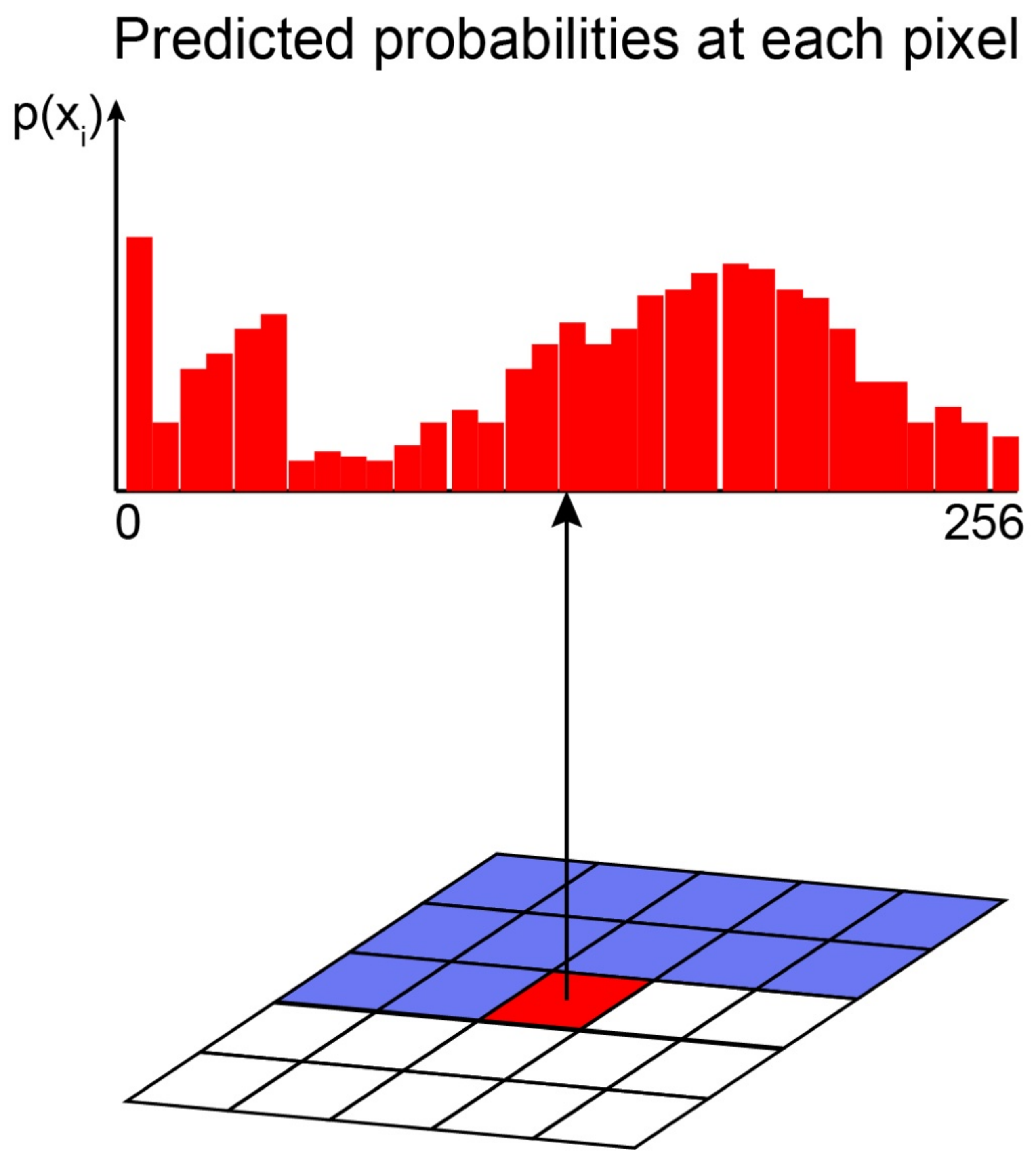
- Unique to graphs, makes the problem fundamentally different
- To achieve optimality, the distribution needs to be defined on isomorphism classes
- **Requires solving graph isomorphism**



Challenge 2: Evaluating & estimating the likelihood

- Evaluate the probability **everywhere.** Requires decomposing the probability distribution (e.g., autoregressive models for ordered data - images and text)

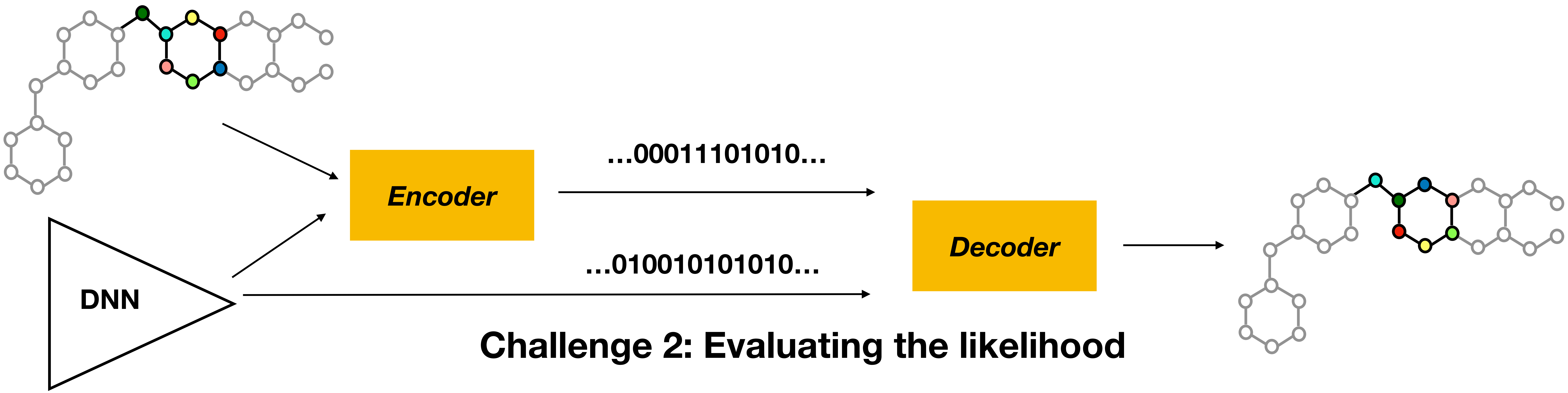
$$q(X) = q(x_1)\prod_{i=2}^{n} q(x_i \mid x_{i-1}, x_{i-2}, \ldots x_1)$$



https://towardsdatascience.com/autoregressive-models-pixelcnn-e30734ede0c1

- Observation space of graphs is huge - grows with $O\left(\frac{2^{n^2}}{n!}\right)$
- **How to decompose the distribution in the absence of ordering?**

Challenge 3: The description length of the model
Compression vs Generative models



Challenge 2: Evaluating the likelihood

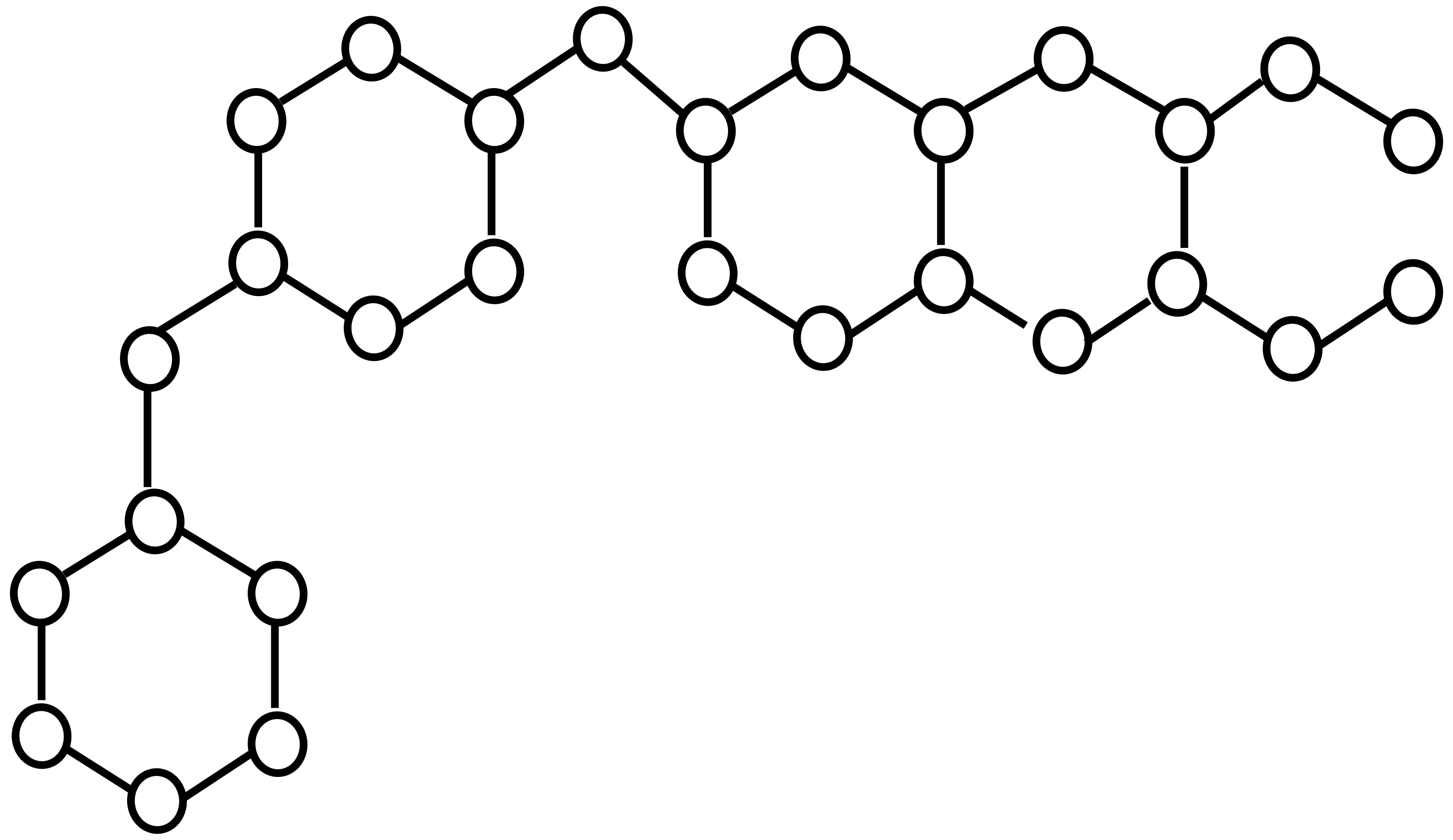
- In case of parametric model (e.g., a DNN), the encoder and the decoder need to both possess the model. Hence:

$$\min_{q,M} \mathbb{E}_{G\sim p}[-\log q(G\,|\,M)] + \frac{1}{N}\mathrm{L}(M)$$

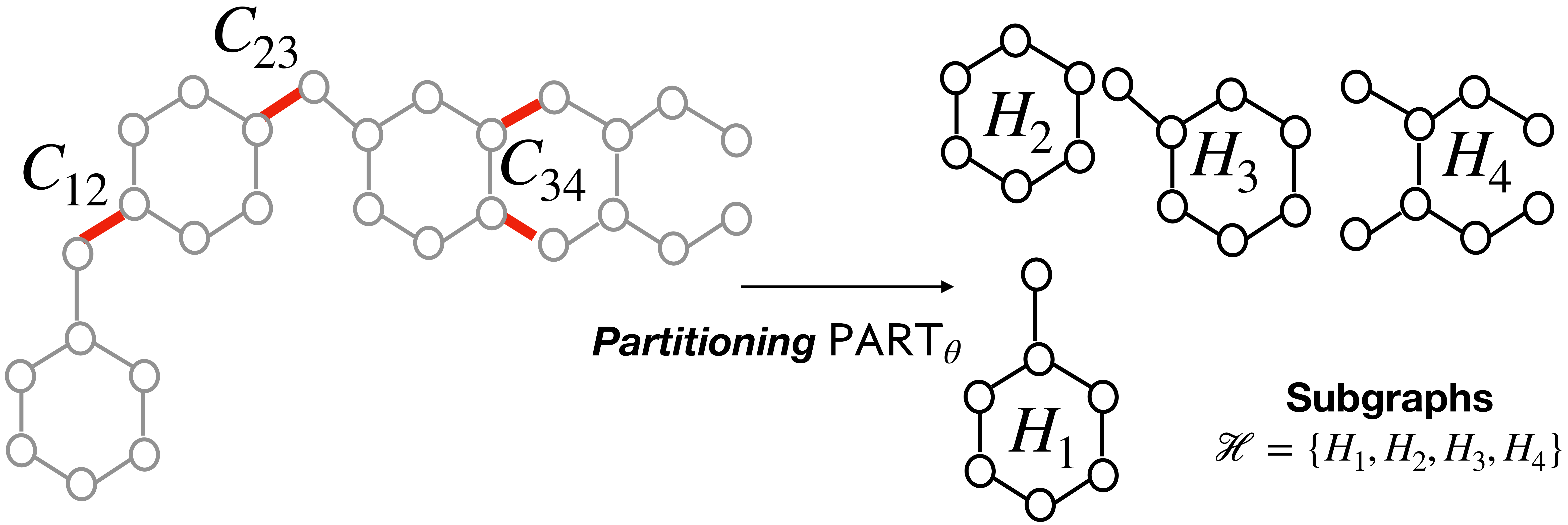
- Overparametrisation might be problematic (we will return to this later)

- **Typically NN training only optimises for** q

Partition and Code: Pipeline



Partition and Code: Pipeline



Cuts

$C = \{C_{12}, C_{13}, C_{14}, C_{23}, C_{24}, C_{34}\}$

Partition and Code: Pipeline

Partition and Code: Pipeline

Dictionary $D = \{a_1, a_2, \ldots, a_{|D|}\}$

Cut Encoding:		*Dictionary Encoding:*		*Subgraph Encoding:*
$-\log q(C \mid \mathscr{H})$	**+**	$-\log q(D)$	**+**	$-\log q(\mathscr{H} \mid D)$
....00010011......		**....00010011......**		**....1010001......**

How do we address the challenges?

- **C1 Isomorphism: Dictionary**
  - We efficiently solve it for small graphs of size $k = O(1)$
  - tradeoff between expressivity and complexity
- **C2 Evaluating the Likelihood: Partitioning**
  - Provides us with a **learnable decomposition** of the probability distribution (subgraphs + cuts)
- **C3 The DL of the model: End-to-end optimisation + Learnable Dictionary**

$$\min_{q,M} \mathbb{E}_{G\sim p}[-\log q(G\,|\,M)] + \frac{1}{N}\mathrm{L}(M) \Rightarrow \min_{\phi,D,\theta} \mathbb{E}_{G\sim p}[-\log q_{\phi}(\mathsf{PART}_{\theta}\,|\,D)] + \frac{1}{N}\mathrm{L}(D)$$

NB: PART_{θ} **does not need to be transmitted**

PART_{θ} **does all the heavy-lifting while** ϕ **is kept small!**

Distribution & dictionary parametrisation

1. Graph Likelihood: Subgraph Encoding + Cut encoding

$$
\begin{aligned}
q_\phi(G \mid D) &= q(\mathscr{H} \mid D)\, q(C \mid \mathscr{H}, D) \\
&= q_\phi(b_{dict}, b_{null})\, q_\phi(\mathscr{H}_{dict} \mid b_{dict}, D)\, q_{null}(\mathscr{H}_{null} \mid b_{null})\, q_{null}(C \mid \mathscr{H})
\end{aligned}
$$

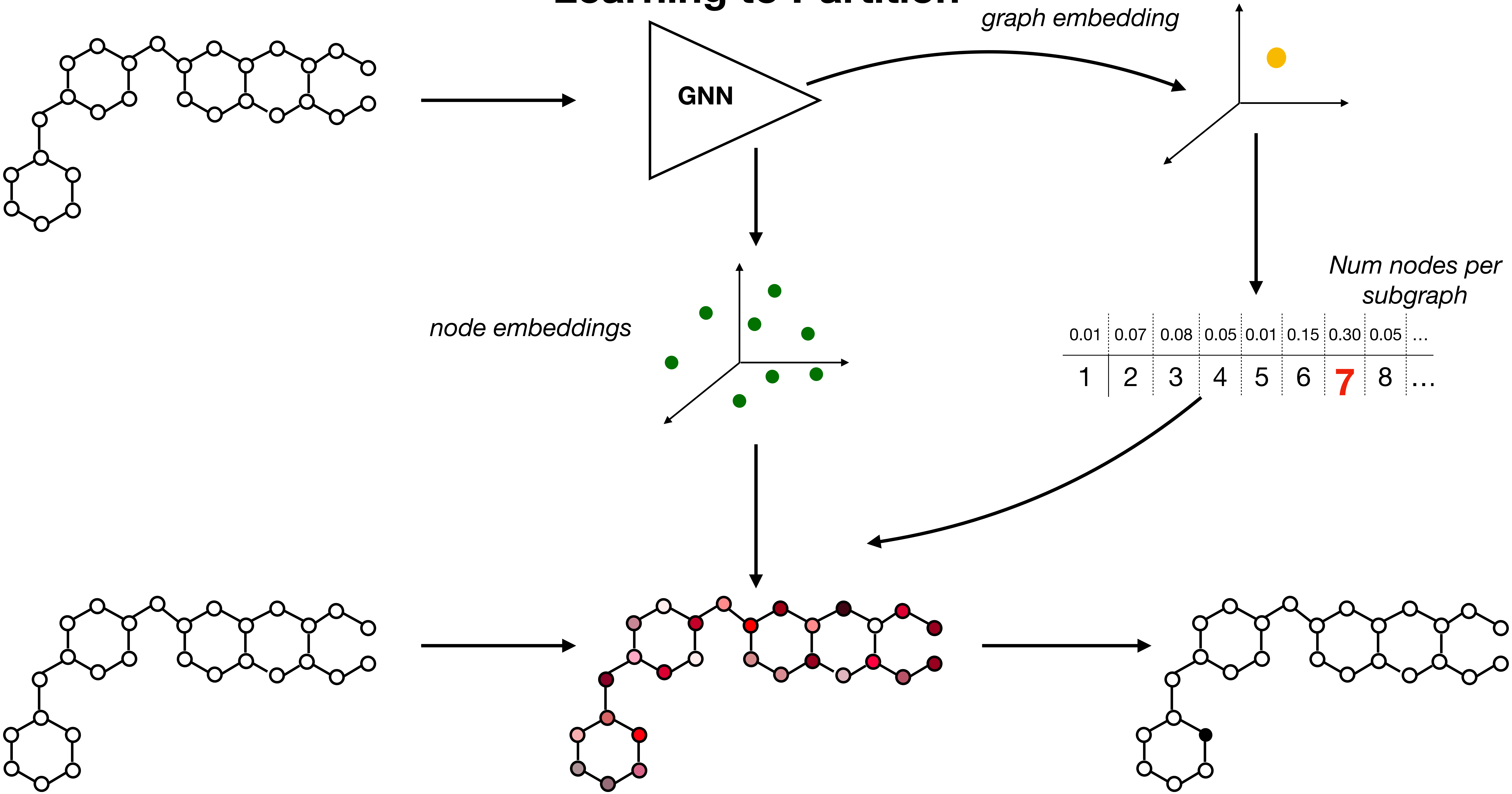
Number of subgraphs + Dictionary subgraphs + Non-dictionary subgraphs + Cuts

Parametric

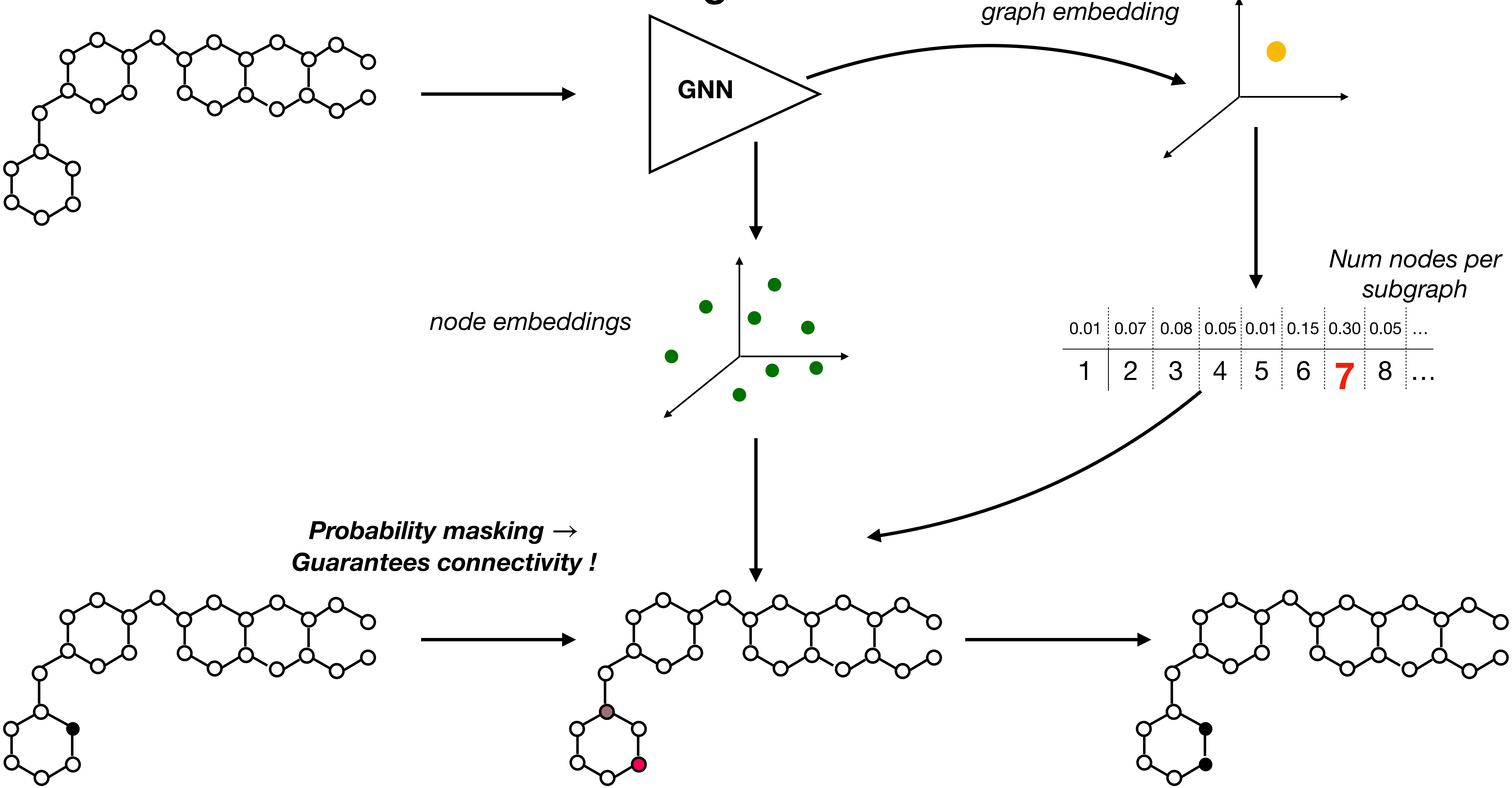
Non-parametric (null model)

2. Dictionary DL: $\mathrm{L}(D) = -\sum_{a_i \in \mathfrak{U}} x_i \log q_{null}(a_i), \qquad x_i = \begin{cases} 1 & \text{if } a_i \in D \\ 0 & \text{otherwise}\,. \end{cases},$ $\mathfrak{U}$: universe

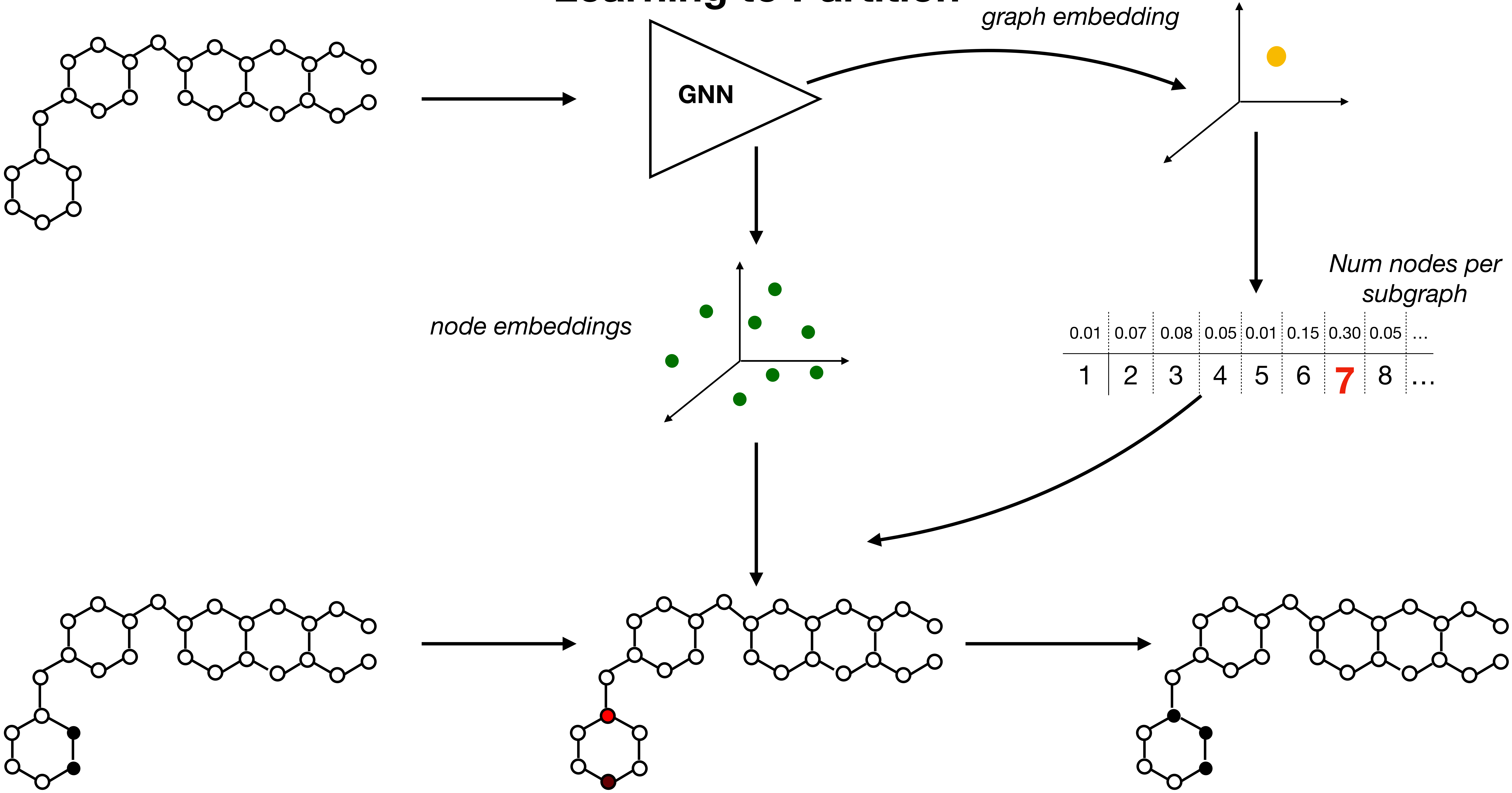
Learning to Partition



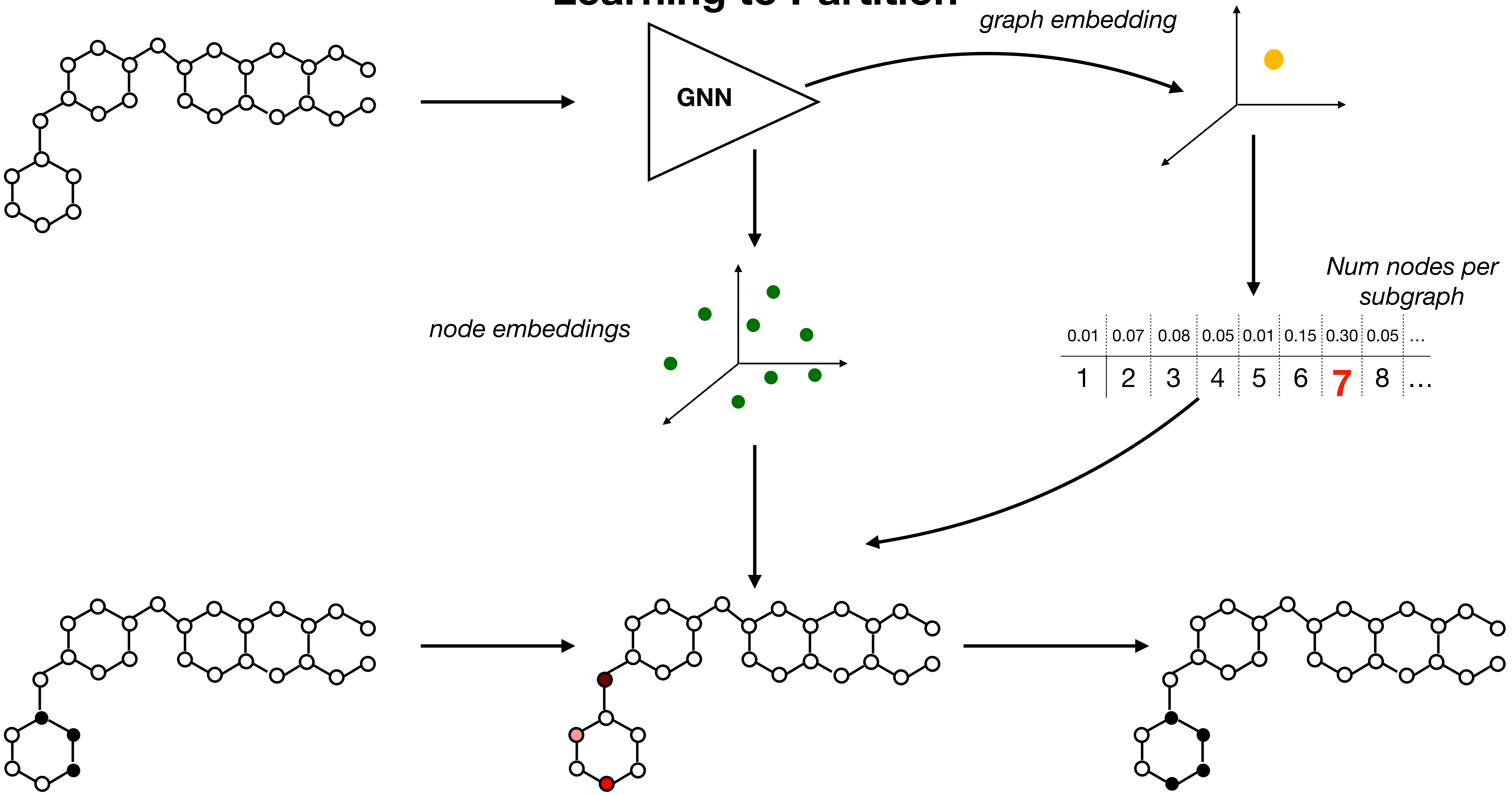
Learning to Partition



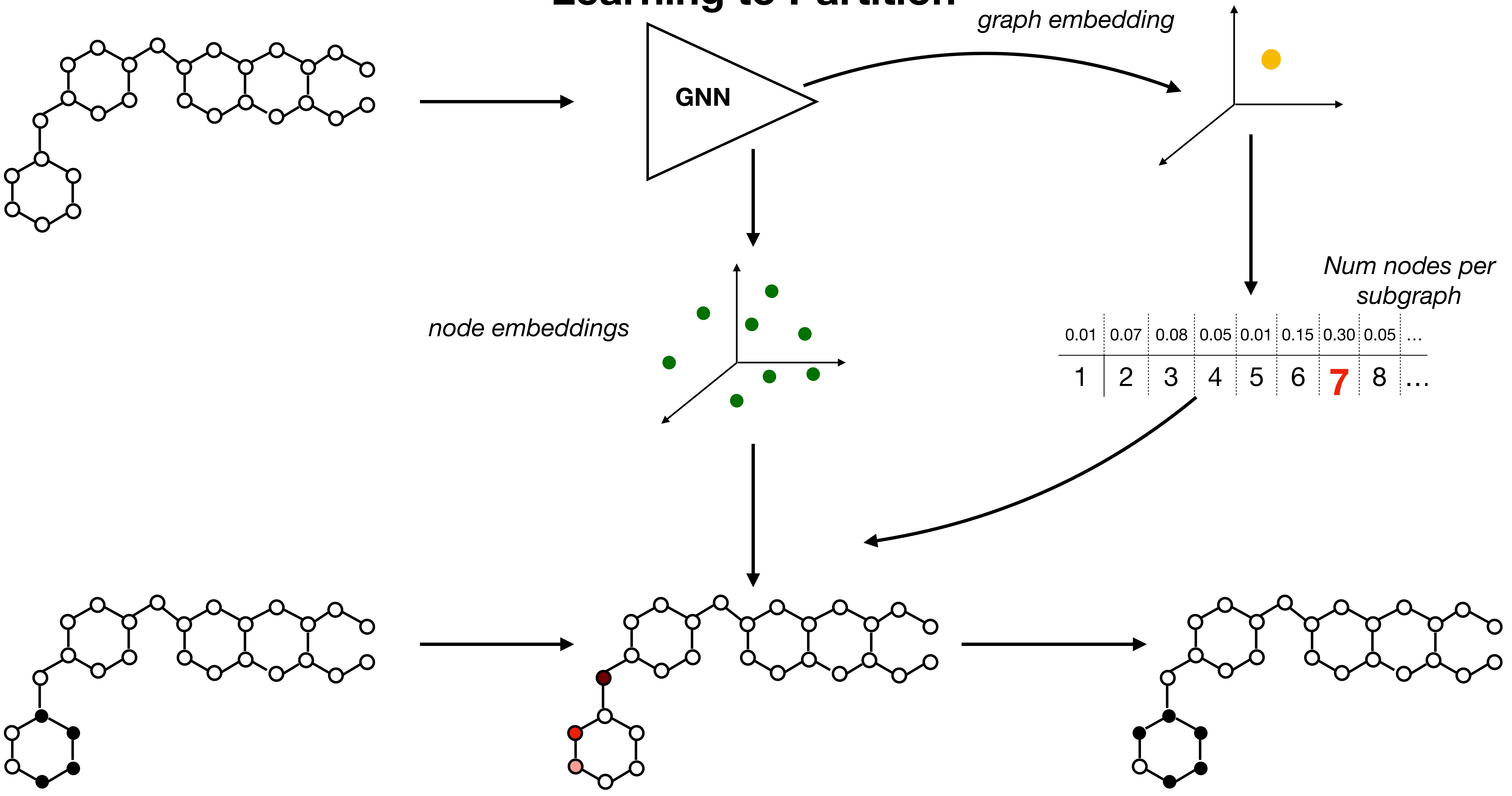
Learning to Partition



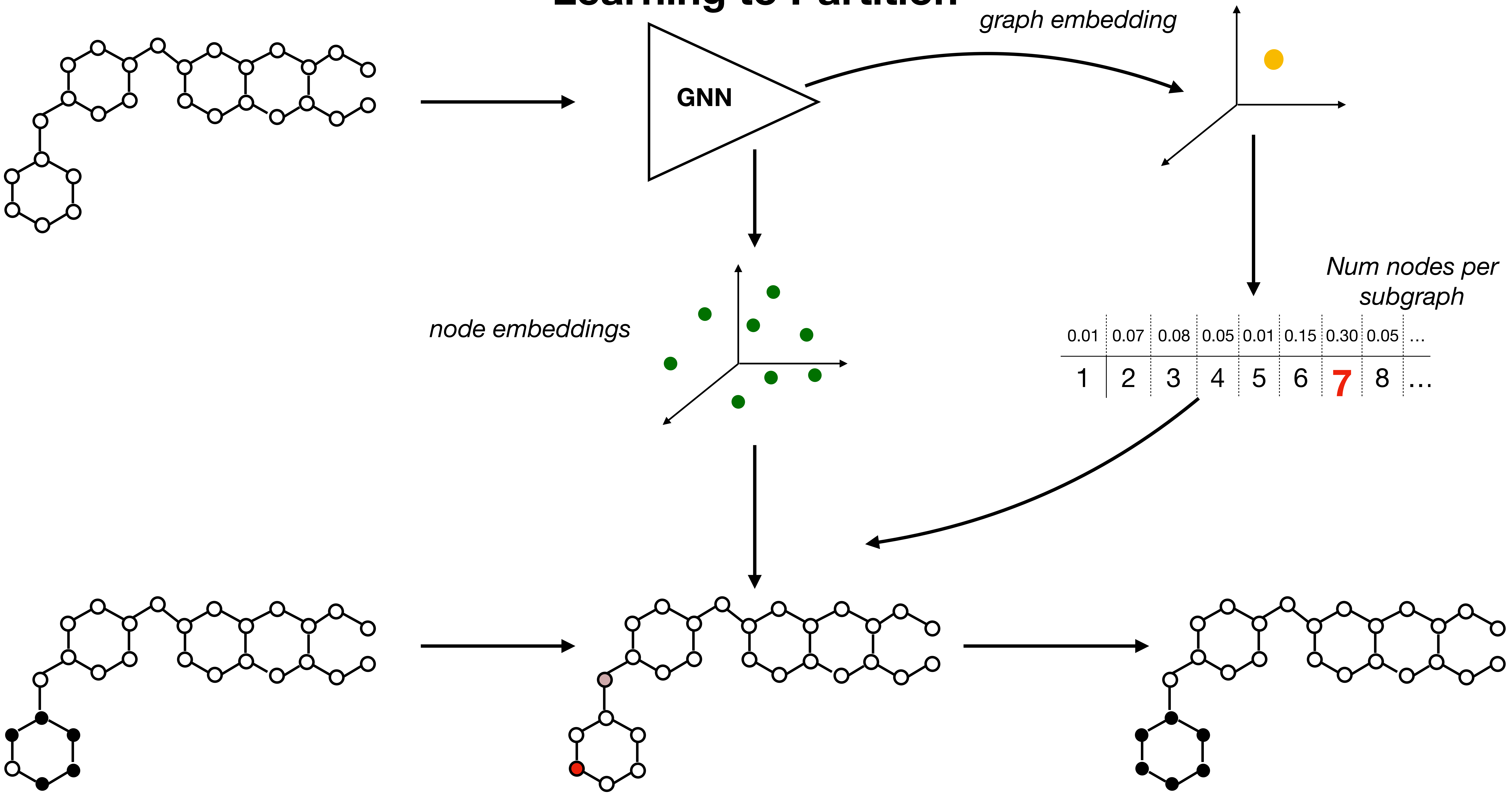
Learning to Partition



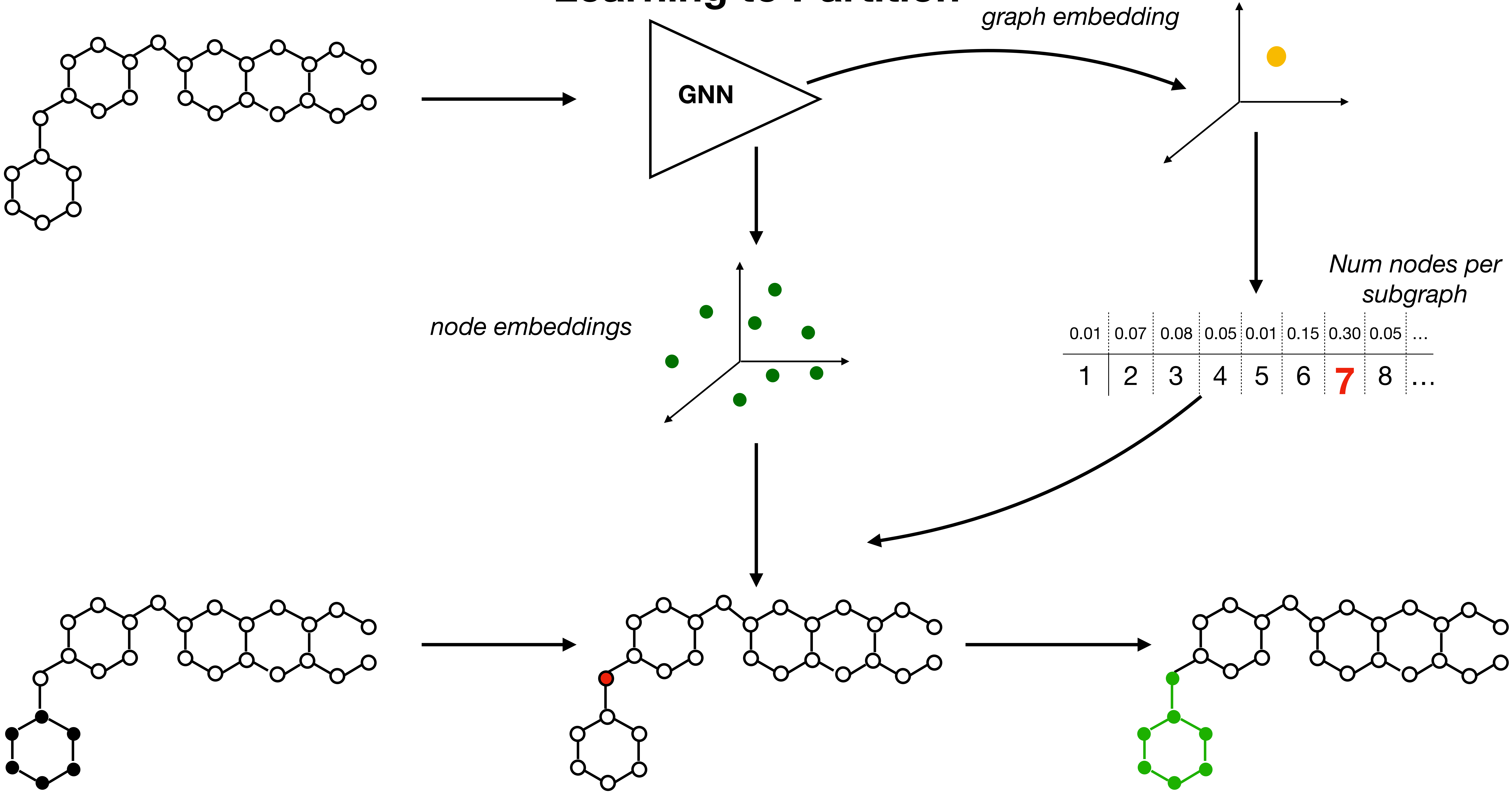
Learning to Partition



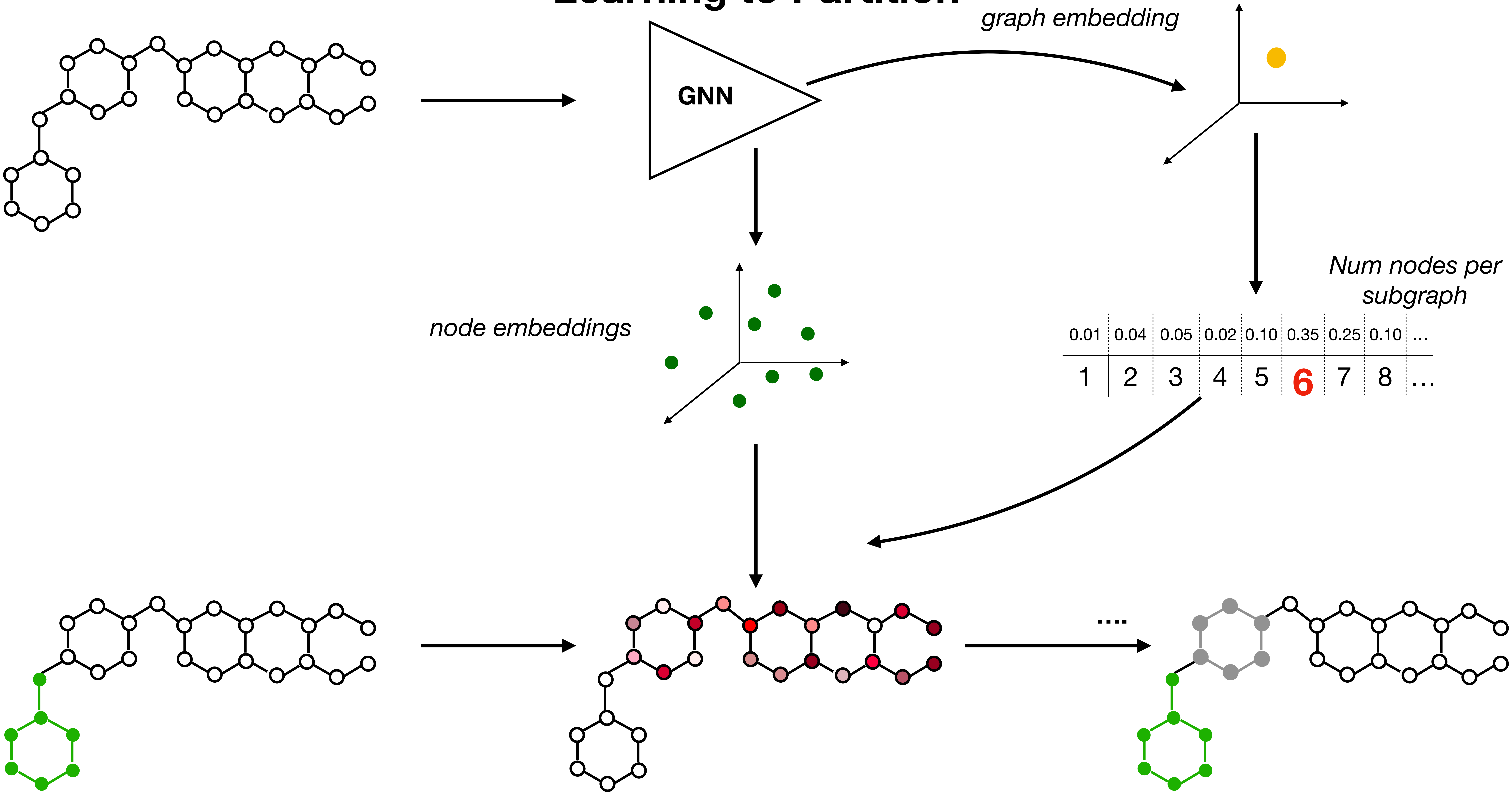
Learning to Partition



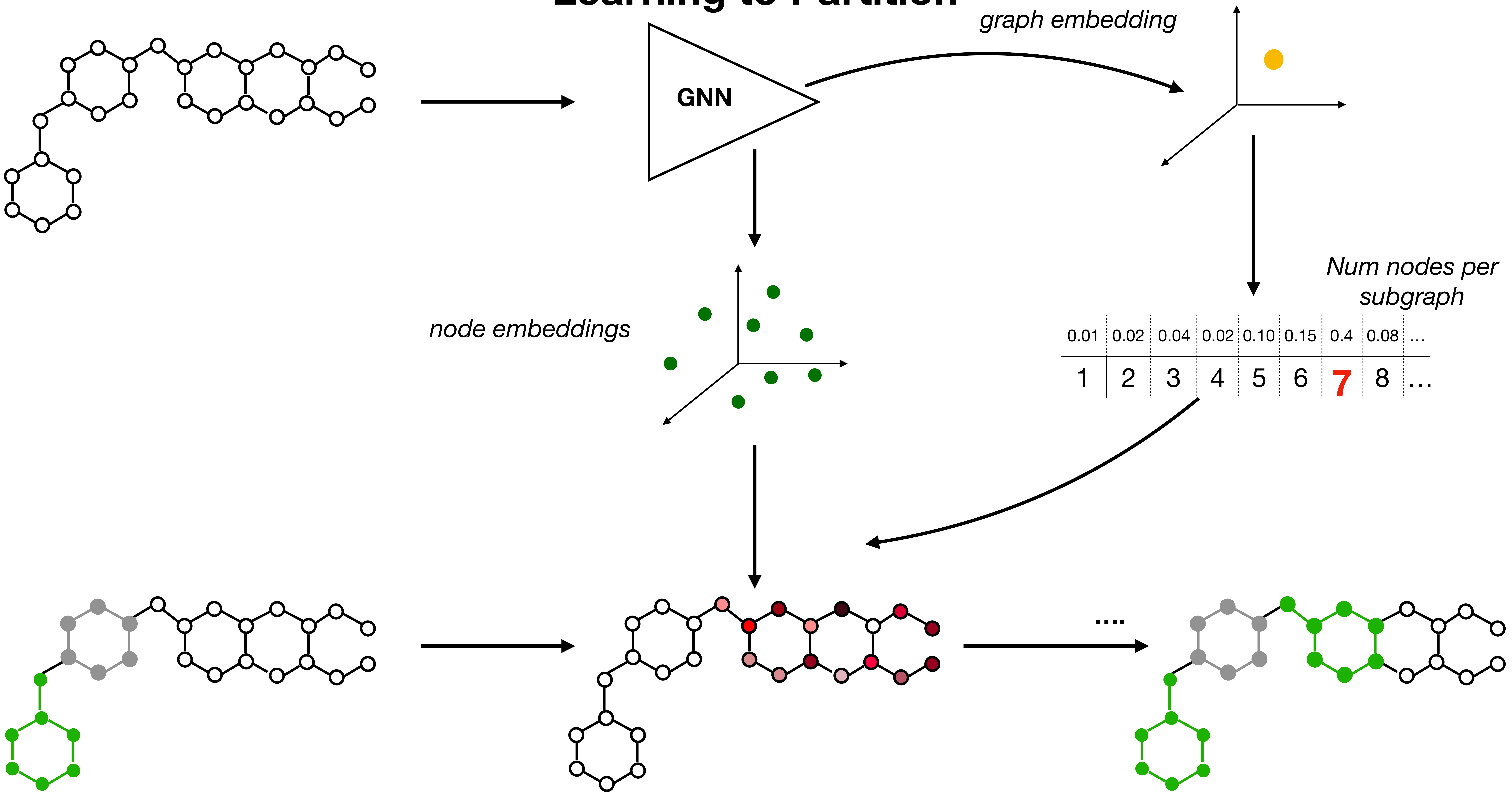
Learning to Partition



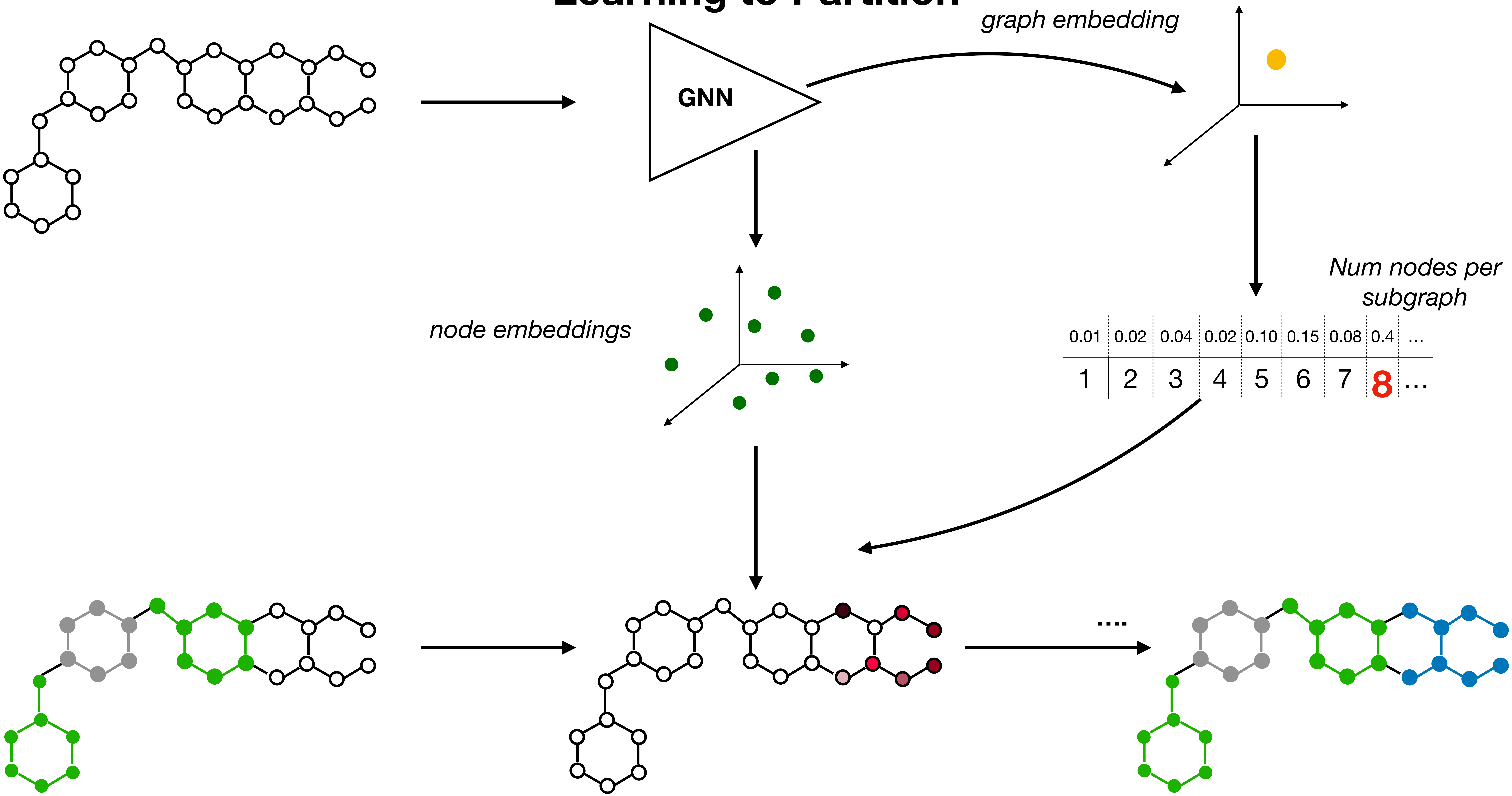
Learning to Partition



Learning to Partition



Learning to Partition



Optimisation

- End-to-end with gradient descent
- Differentiable w.r.t ϕ
- PART$_\theta$ **:** policy gradient
- D: continuous relaxation with fractional indicator variables $\hat{x} \in [0,1]$

Overall

$$\min_{\phi,\hat{\mathbf{x}},\theta} \sum_{G \in \mathcal{G}} \mathbb{E}_{p_\theta^{GNN}}[\mathrm{L}_{\phi,\hat{\mathbf{x}}}(\mathscr{H}, C \mid D)] + \mathrm{L}_{\hat{\mathbf{x}}}(D)$$

Theoretical gains

Quadratic gains against (1) Null models and **Linear** against (2) Pure non-parametric partitioning

Theorem 1 (informal). *Consider a partitioning algorithm that decomposes a graph of n vertices into blocks of $k = O(1)$ vertices. Under mild conditions, it holds that:*

$$\mathbb{E}_{G\sim p}[L_{PnC}(G)] \lesssim \mathbb{E}_{G\sim p}[L_{Part}(G)] \lesssim \mathbb{E}_{G\sim p}[L_{null}(G)]$$

The absolute compression gains are:

$$\mathbb{E}_{G\sim p}[L_{part}(G)] \lesssim \mathbb{E}_{G\sim p}[L_{null}(G)] - \Theta(n^2) \quad \textit{and} \quad \mathbb{E}_{G\sim p}[L_{PnC}(G)] \lesssim \mathbb{E}_{G\sim p}[L_{Part}(G)] - \Theta(n)$$

Linear gains against (3) PnC without isomorphism testing

Theorem 2 (informal). *Consider a PnC compressor that yields dictionary subgraphs with probability $1 - \delta$. Then:*

$$\mathbb{E}_{G\sim p}[\mathrm{L}_{PnC-S}(G)] \approx \mathbb{E}_{G\sim p}[\mathrm{L}_{PnC-G}(G)] - n(1-\delta)\log k$$

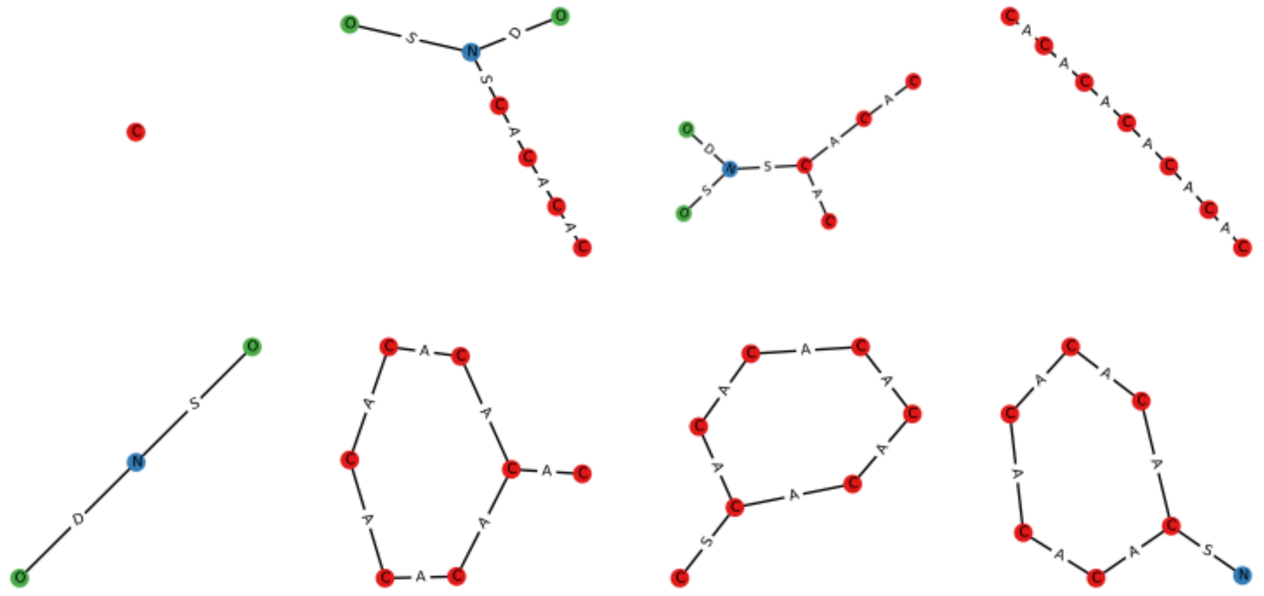
Results

- Biological and Social network distributions (TU datasets).
- Baselines:
  - (1) Null (uninformative),
  - (2) Pure partitioning,
  - (3) Deep generative models
- Description length is measured in **bits per edges.**

Results

Method type	Graph type	Small Molecules								
	Dataset name	MUTAG			PTC			ZINC		
		data	total	params	data	total	params	data	total	params
Null	Uniform (raw adjac.)	-	8.44	-	-	17.43	-	-	10.90	-
	Edge list	-	7.97	-	-	9.38	-	-	8.60	-
	Erdős-Renyi	-	4.78	-	-	5.67	-	-	5.15	-
Neural (likelihood)	GraphRNN	1.33	1669.77	388K	1.57	698.08	389K	1.62	22.39	388K
	GRAN	0.81	6279.28	1460K	2.18	2636	1470K	1.30	79.50	1461K
Partitioning (non-parametric)	SBM-Bayes	-	4.62	-	-	5.12	-	-	4.75	-
	Louvain	-	4.80	-	-	5.27	-	-	4.77	-
	PropClust	-	4.92		-	5.40		-	4.85	
PnC	PnC + SBM	3.81	4.11	49	4.38	4.79	155	3.34	3.45	594
	PnC + Louvain	2.20	**2.51**	47	2.65	**3.15**	166	1.96	**1.99**	196
	PnC + PropClust	2.42	**3.03**	63	3.38	**4.02**	178	2.20	**2.35**	726
	PnC + Neural Part.	2.17±0.02	**2.45±0.02**	46±1	2.63±0.26	**2.97±0.14**	143±31	2.01±0.02	**2.07±0.03**	384±105

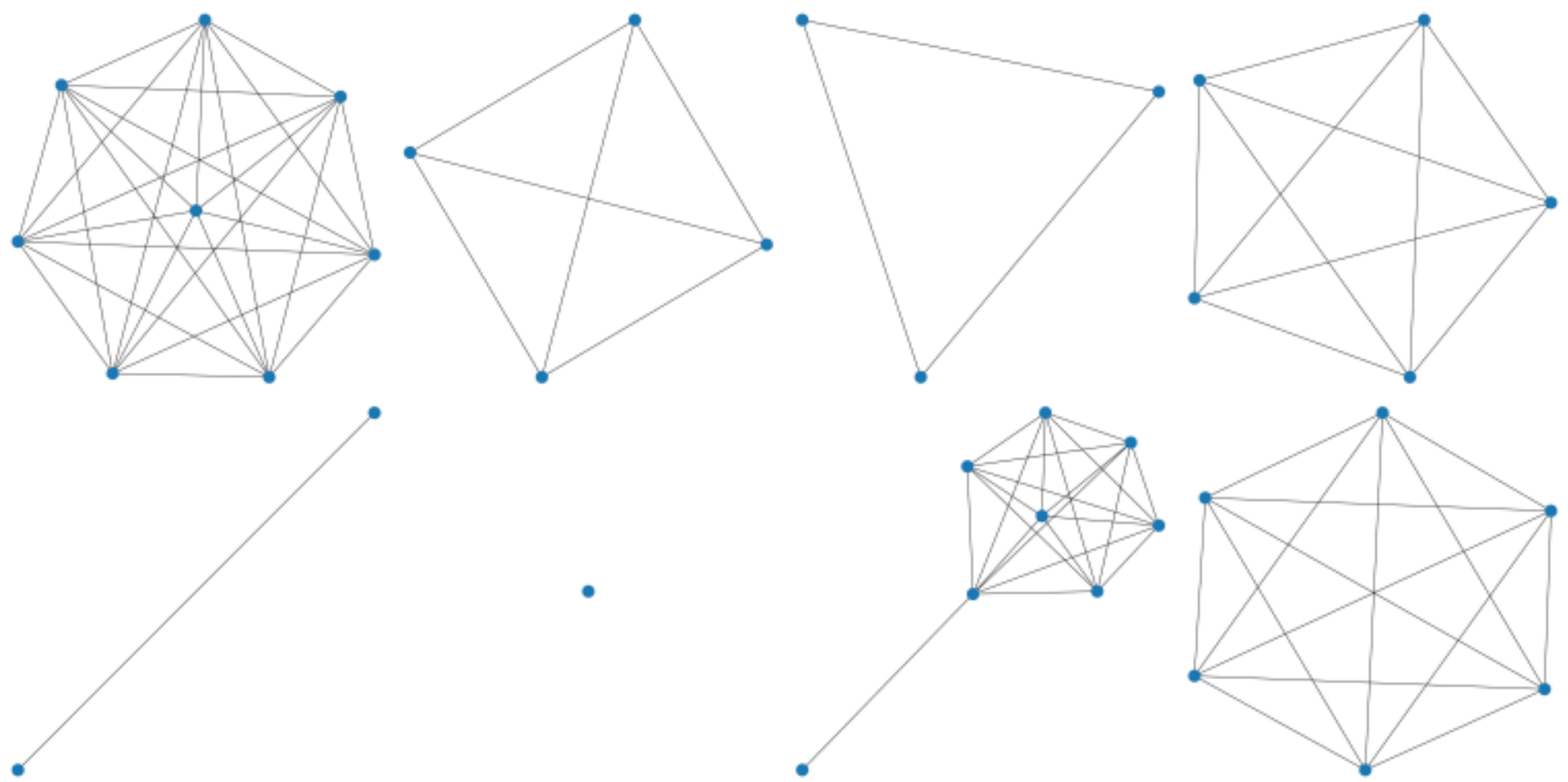
- **Learning to partition** helps in absence of clear community structure



Results

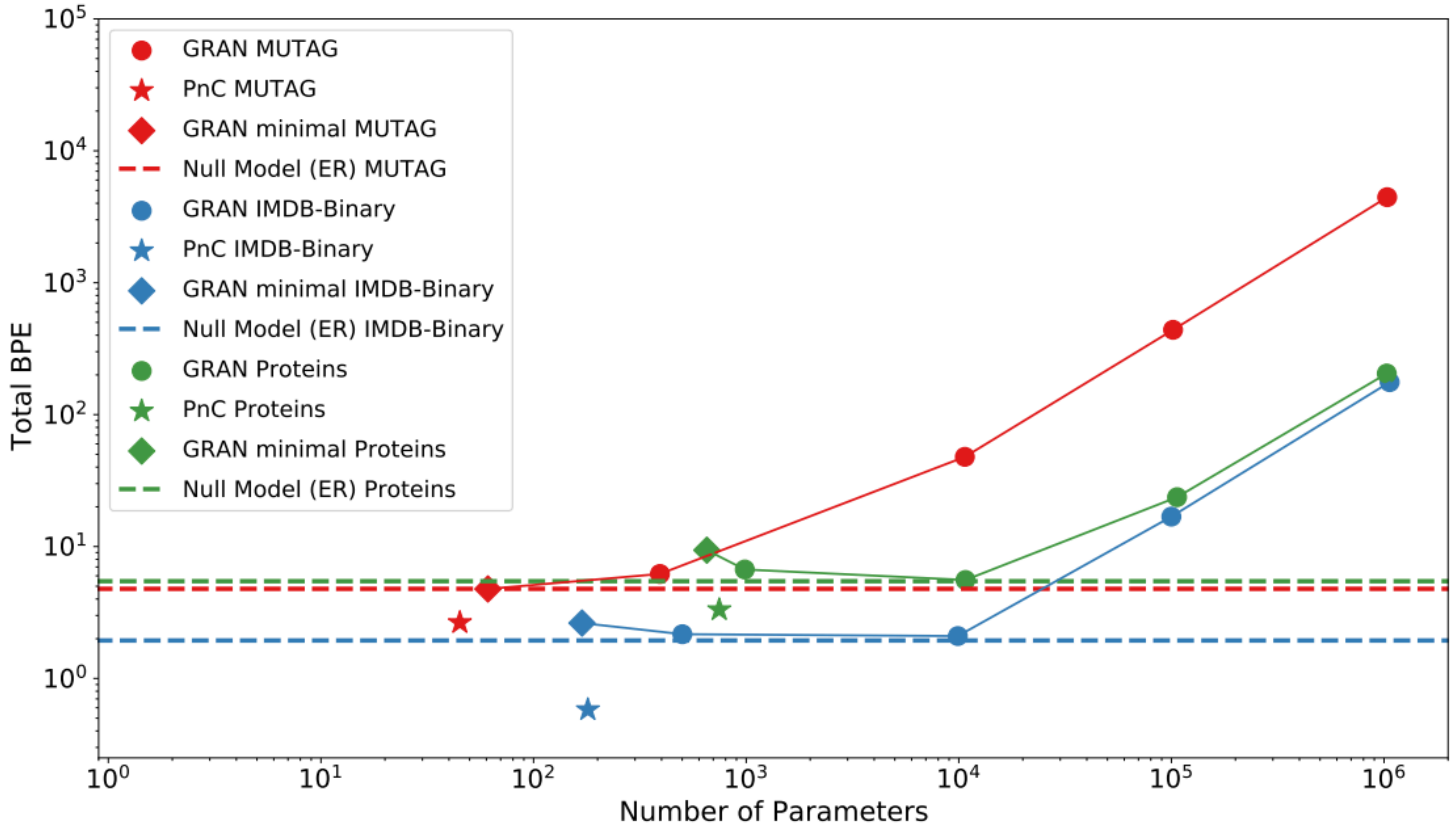
Method type	Graph type	Biology			Social Networks					
	Dataset name	PROTEINS			IMDB-B			IMDB-M		
		data	total	params	data	total	params	data	total	params
Null (non-parametric)	Uniform (raw adjac.)	-	24.71	-	-	2.52	-	-	1.83	-
	Edge list	-	10.92		-	8.29	-	-	7.74	-
	Erdős-Renyi	-	5.46	-	-	1.94	-	-	1.32	-
Neural (likelihood)	GraphRNN	2.03	79.51	392K	1.03	66.65	395K	0.72	64.28	392K
	GRAN	1.51	304.735	1545K	0.26	244.57	1473K	0.17	237.65	1467K
Partitioning (non-parametric)	SBM-Bayes	-	3.98	-	-	**0.80**	-	-	**0.60**	-
	Louvain	-	3.95	-	-	1.22	-	-	0.88	-
	PropClust	-	4.11		-	1.99	-	-	1.37	-
PnC	PnC + SBM	3.26	3.60	896	0.50	**0.54**	198	0.38	**0.38**	157
	PnC + Louvain	3.30	**3.58**	854	0.96	**1.02**	202	0.66	**0.70**	141
	PnC + PropClust	3.40	**3.70**	866	1.45	1.64	241	0.93	1.04	178
	PnC + Neural Part.	3.34±0.25	**3.51±0.23**	717±61	1.00±0.04	1.05±0.04	186±25	0.66±0.05	0.72±0.05	178±14

- PnC consistently improves compression
- Room for improvement in Neural Partitioning



PnC vs Overparametrised NNs

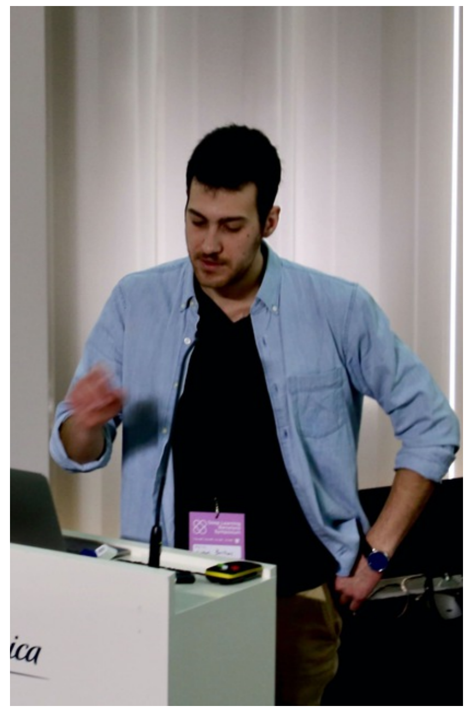
dataset	GraphRNN	GRAN
MUTAG	x1264	x3412
PTC	x484	x3173
ZINC	x38	x90
PROTEINS	x60	x168
IMDBB	infeasible	x763
IMDBM	infeasible	x1033

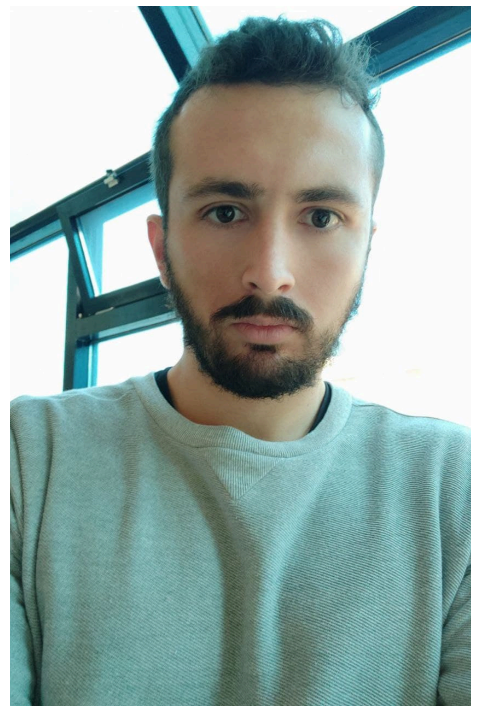


- Vanilla graph generators are suboptimal for compression **(heavily overparametrised!)**
- Unclear how to minimise total description length
- Posthoc model compression: tedious/often ineffective

Take home messages

- Challenging problem - very relevant to the machine learning community
- Potentially large impact
- Can we do better?
    1. Learnable Partitioning - problem at its own sake
    2. How to scale to large graphs?
    3. Deep generative models + accounting for the total DL during training *(general problem in neural compression)*
- Ideas? Let’s discuss!
- Interested in PhD/PostDoc positions? Get in touch with Andreas






@gbouritsas
@loukasa_tweet

arxiv.org/abs/2107.01952