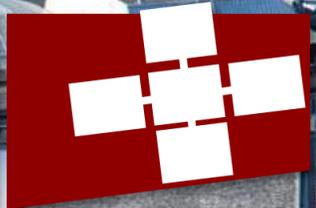
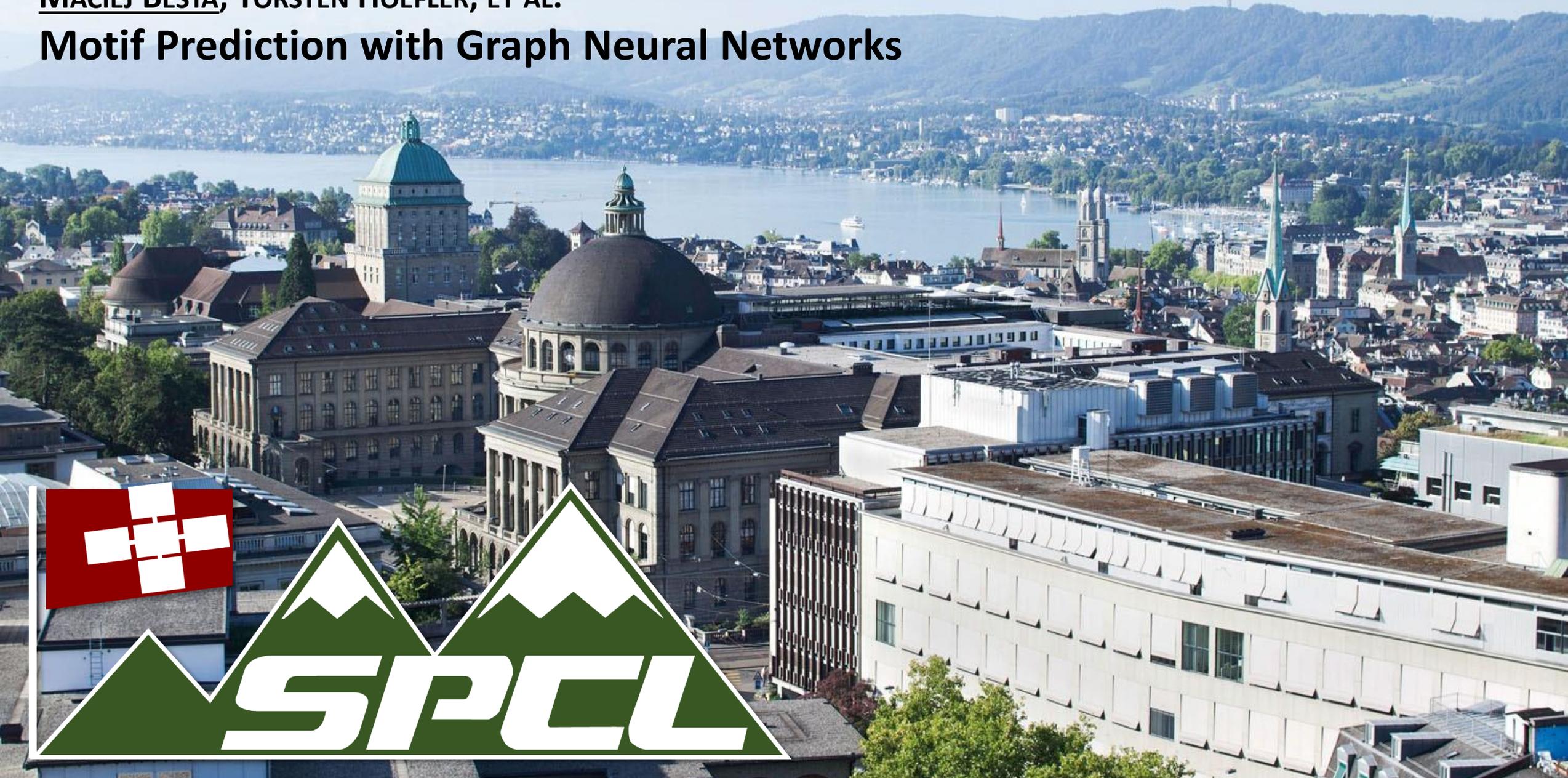
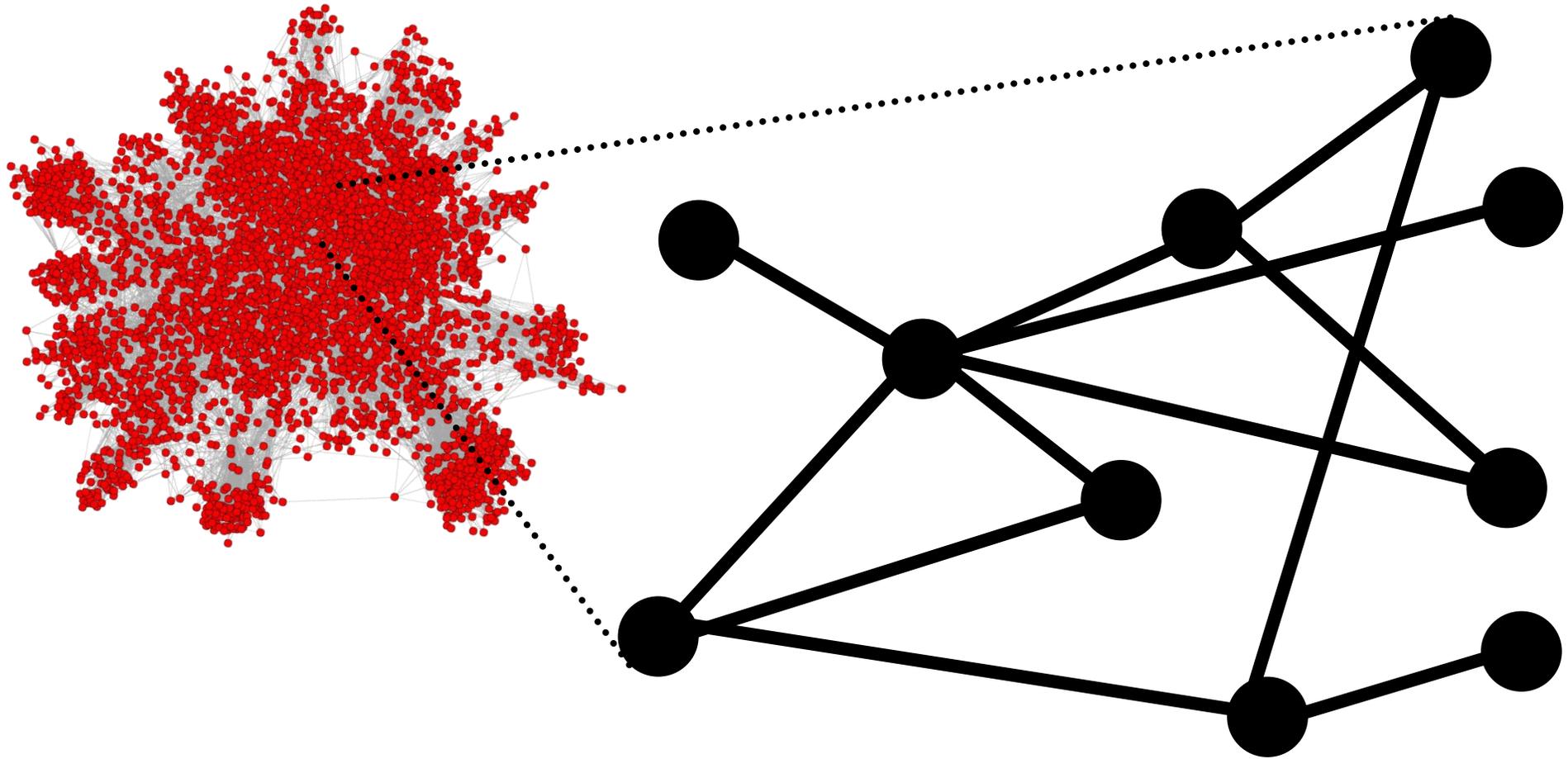


MACIEJ BESTA, TORSTEN HOEFLER, ET AL.

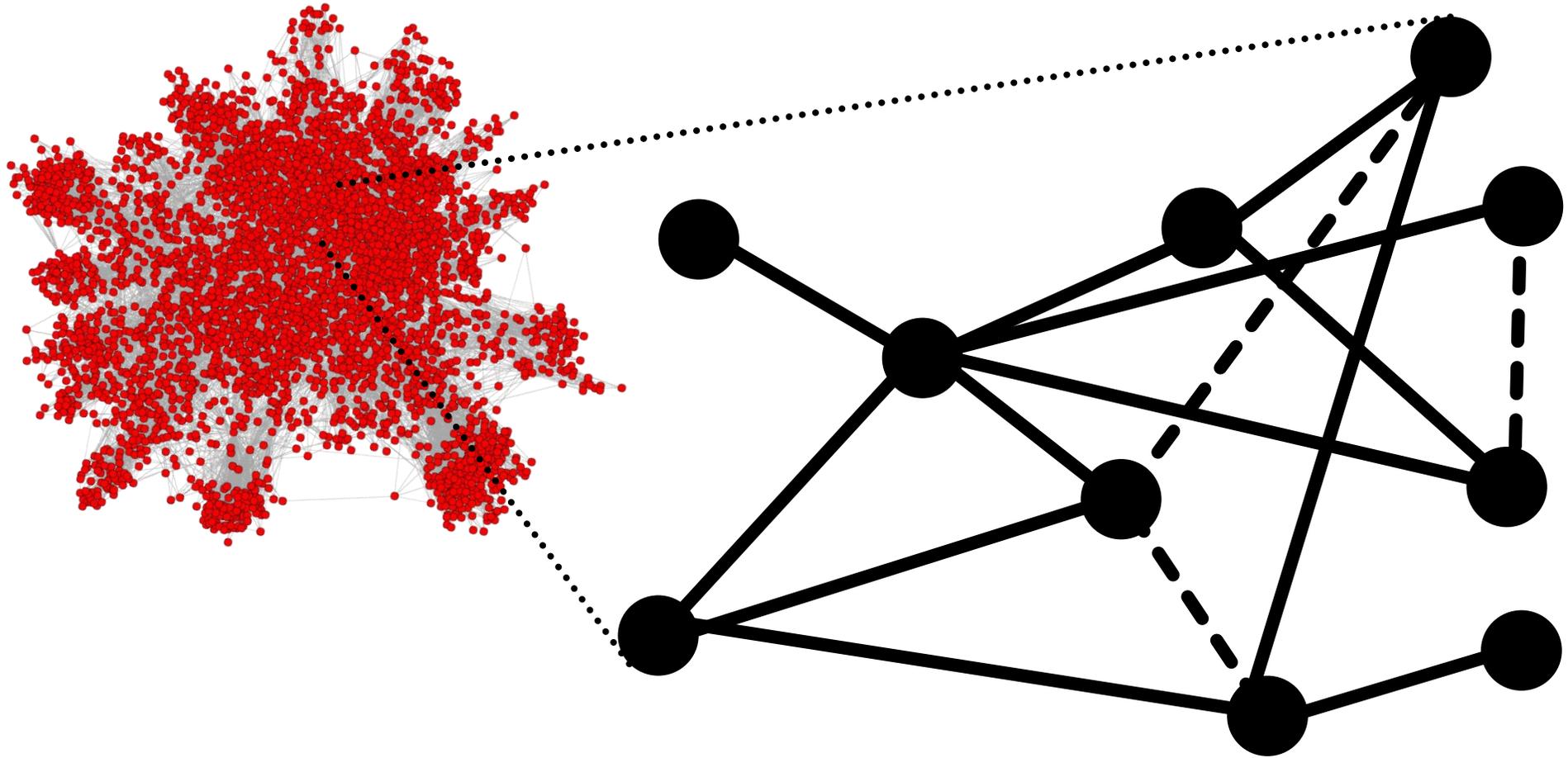
Motif Prediction with Graph Neural Networks



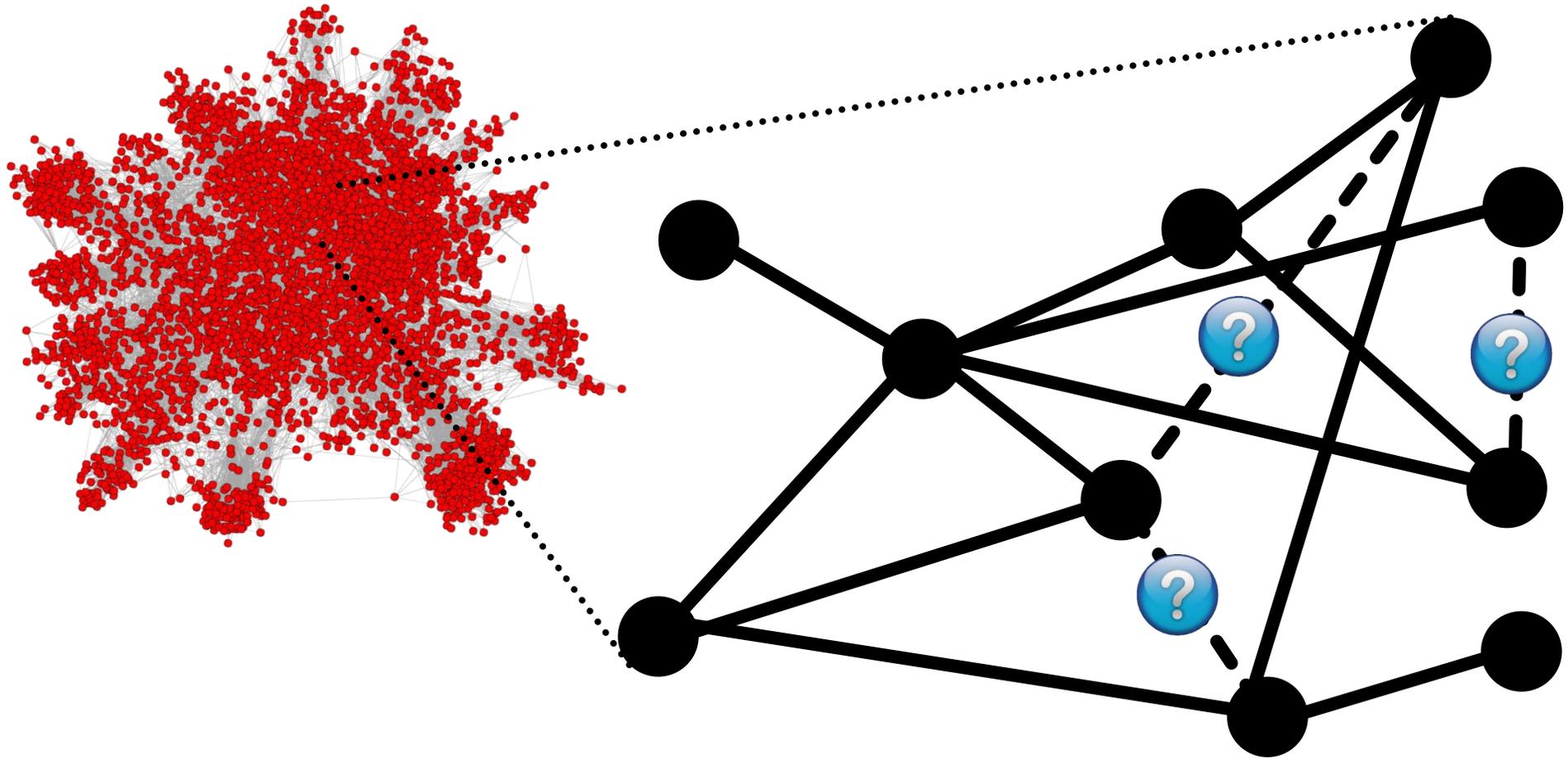
Starting Point: Link Prediction



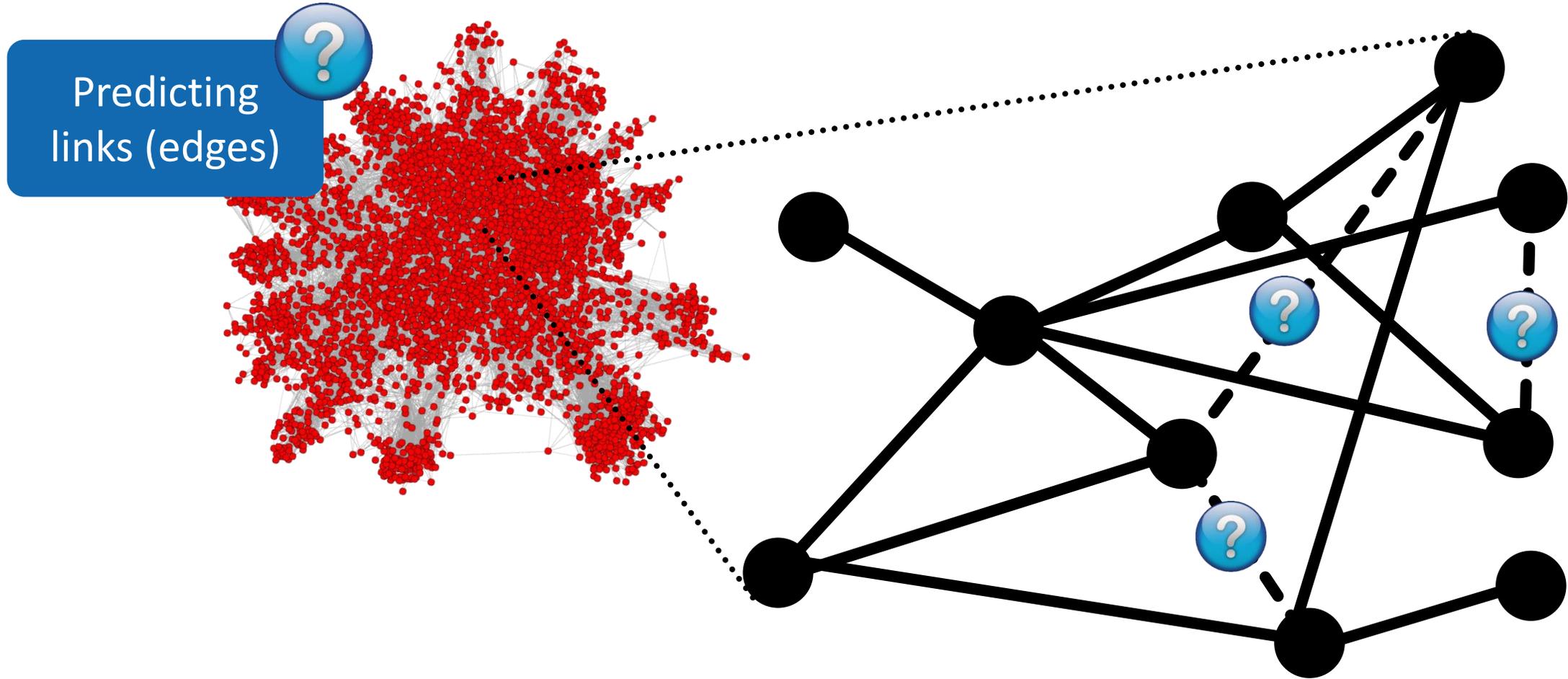
Starting Point: Link Prediction



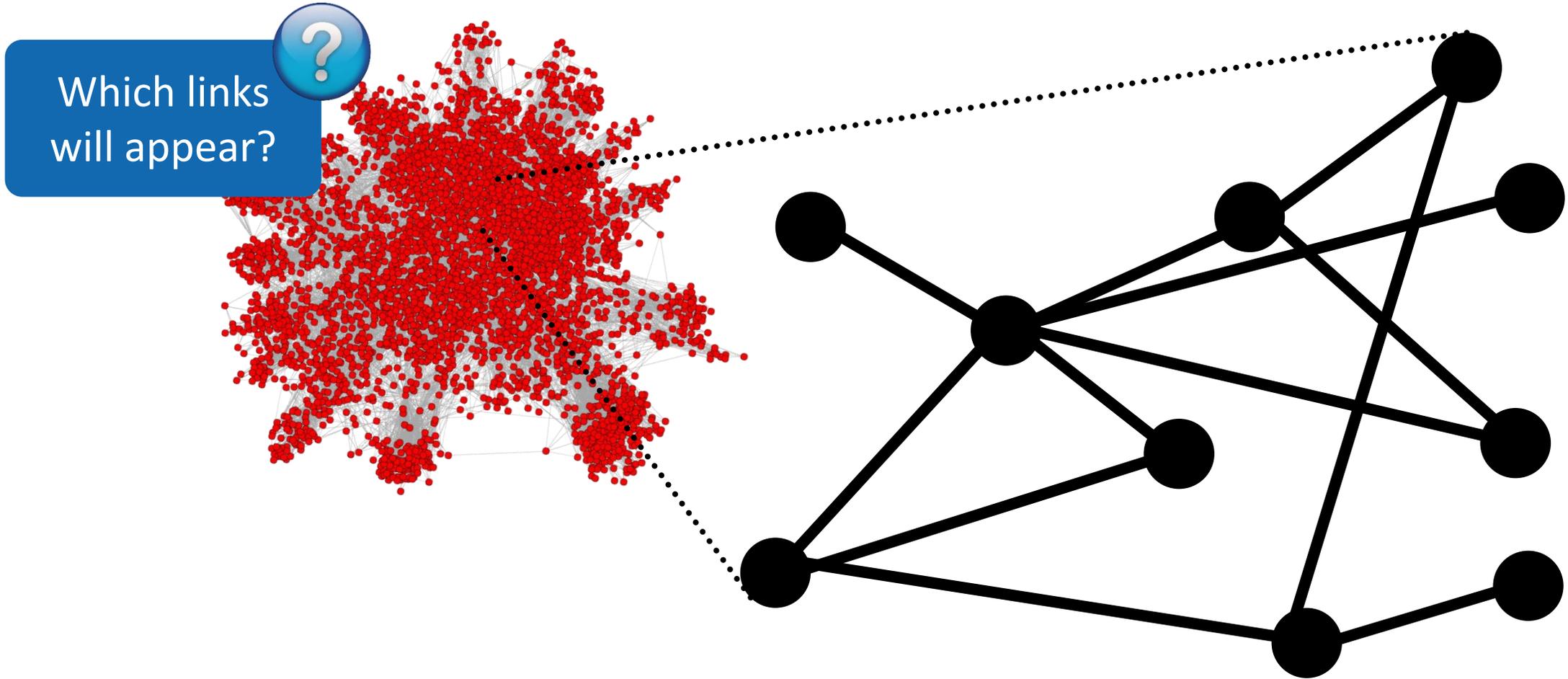
Starting Point: Link Prediction



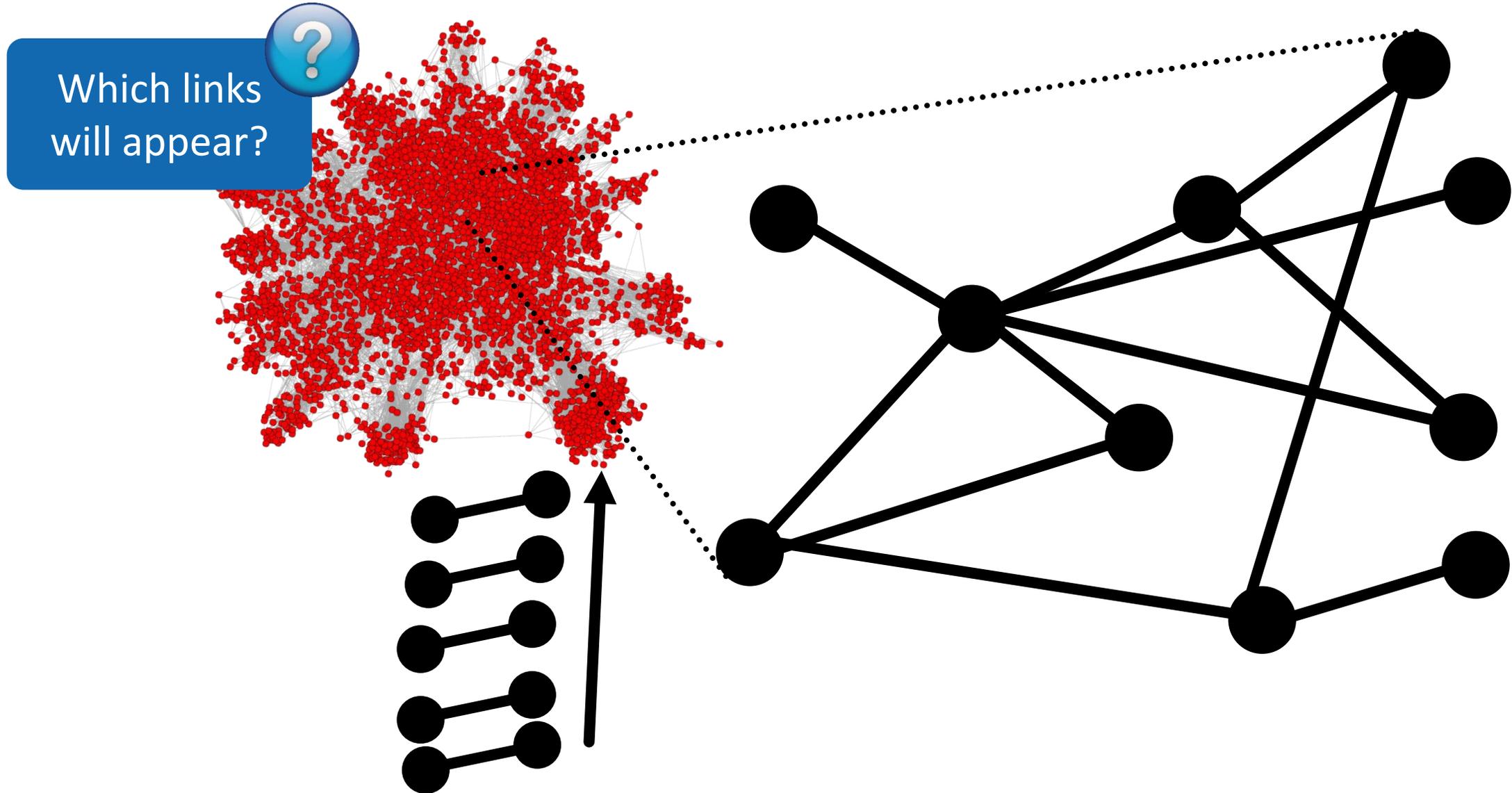
Starting Point: Link Prediction



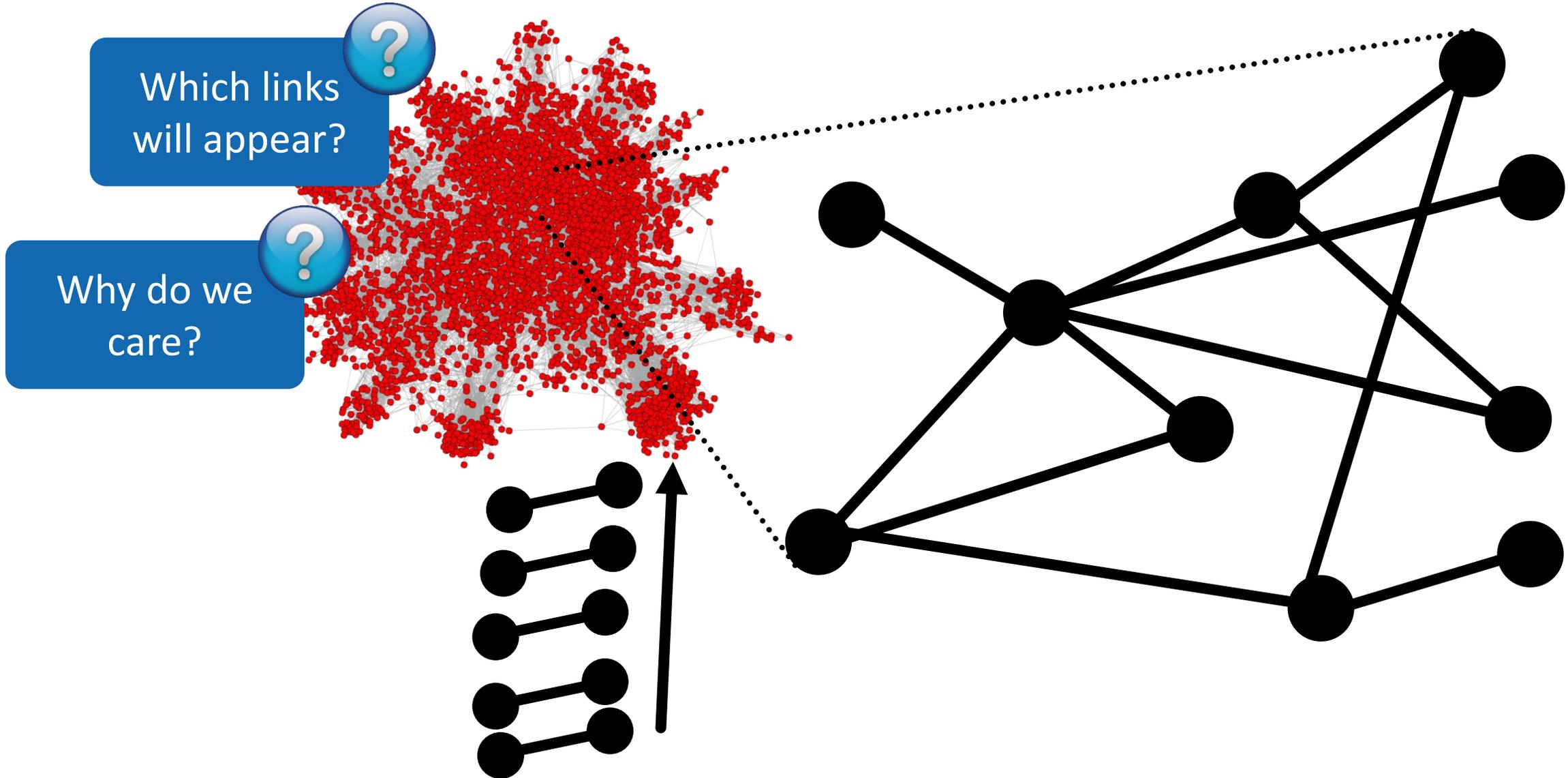
Case 1: New Edges Appear (Dynamic Graphs)



Case 1: New Edges Appear (Dynamic Graphs)



Case 1: New Edges Appear (Dynamic Graphs)

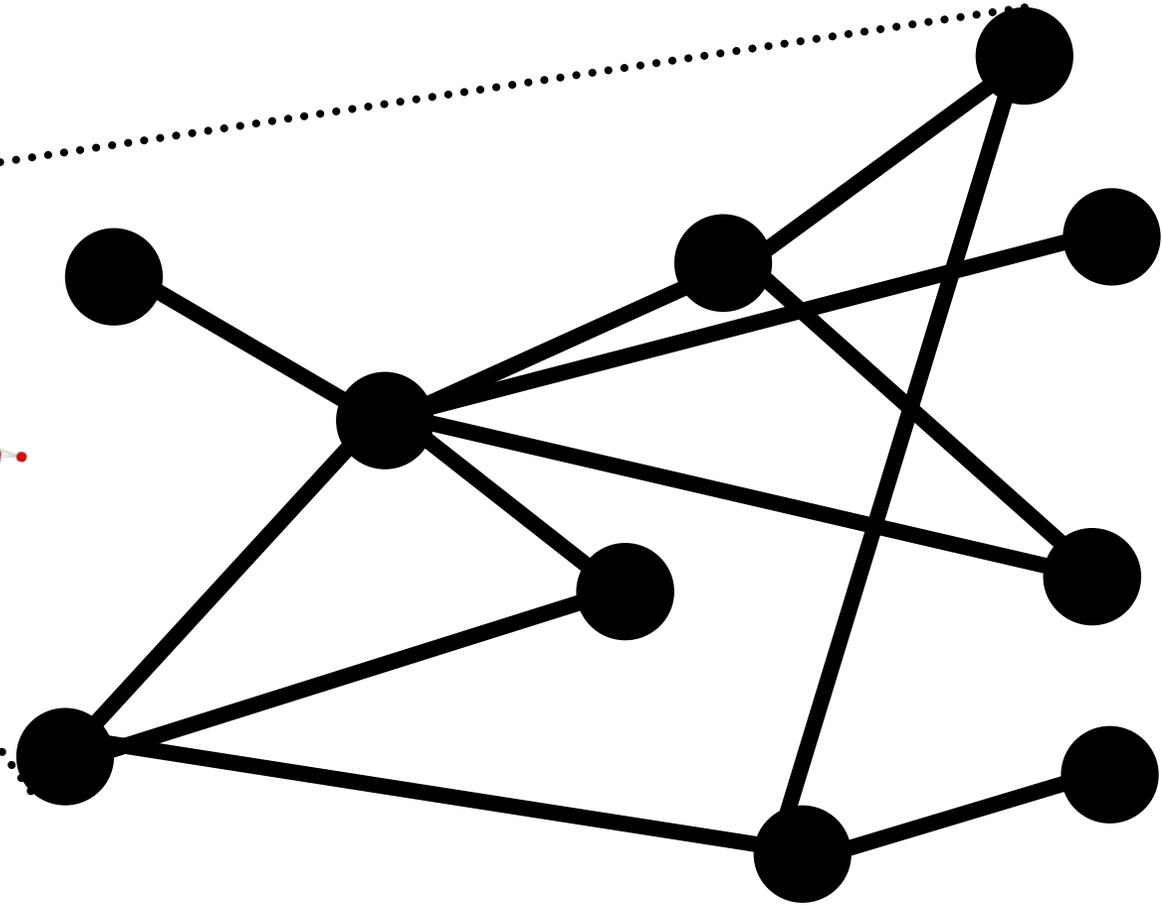
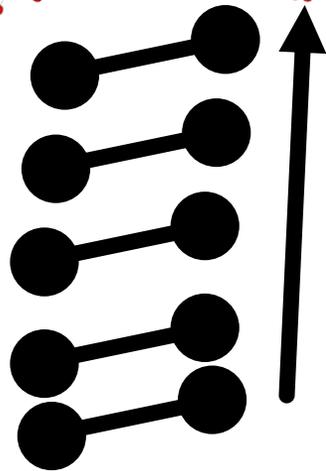
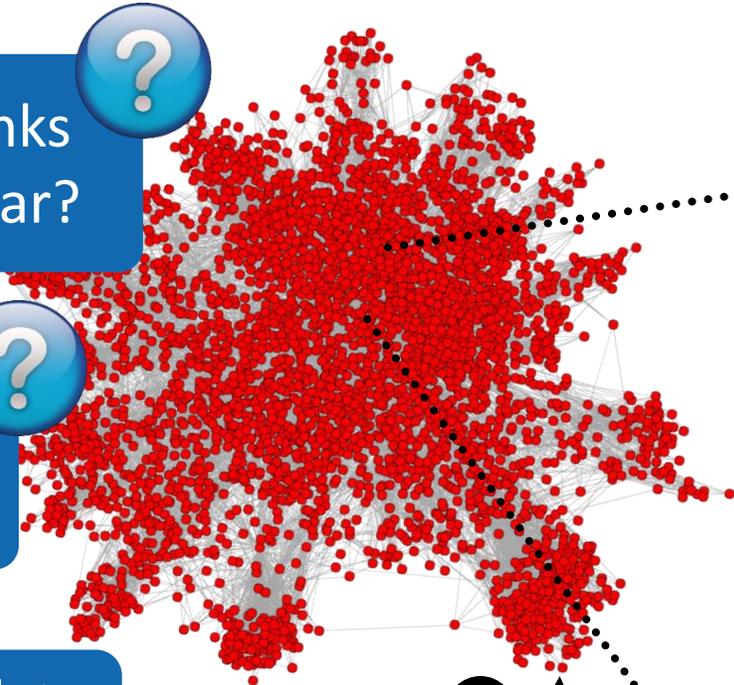


Case 1: New Edges Appear (Dynamic Graphs)

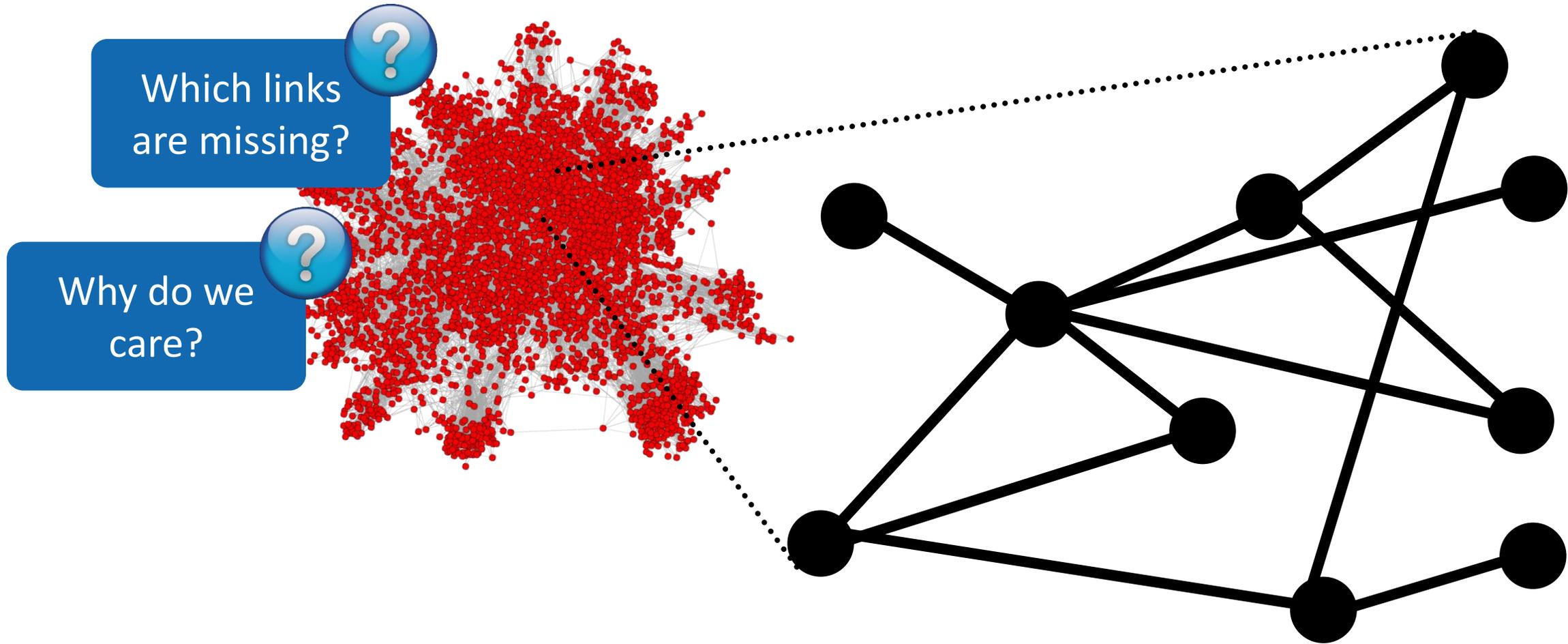
Which links will appear?

Why do we care?

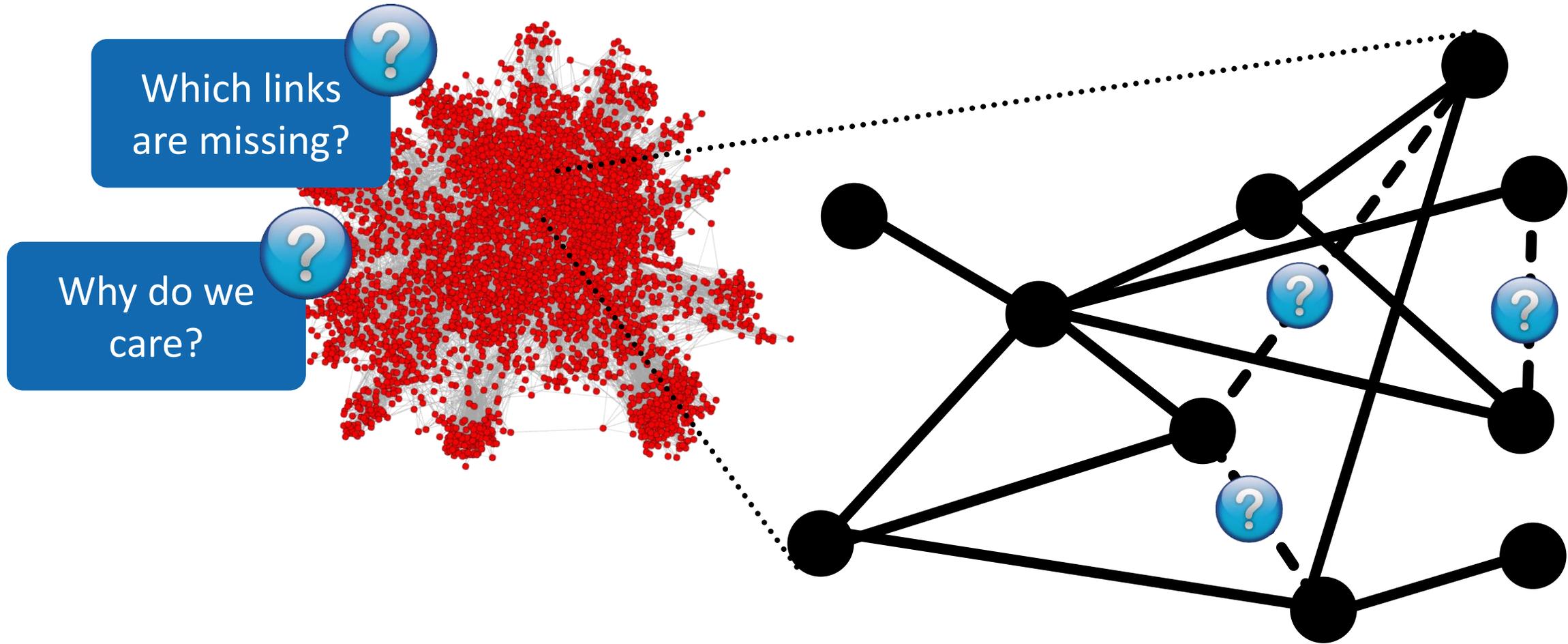
Predict future data



Case 2: Find Missing Data



Case 2: Find Missing Data



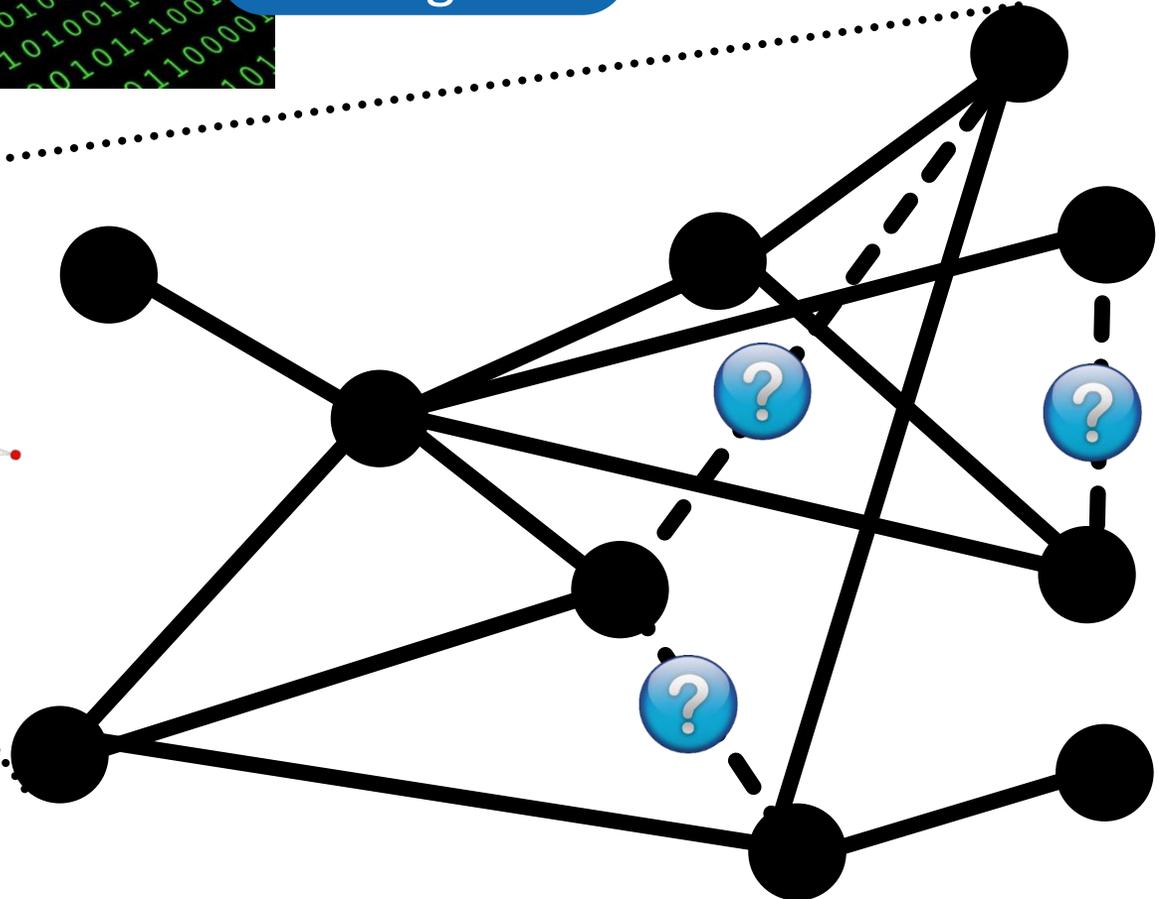
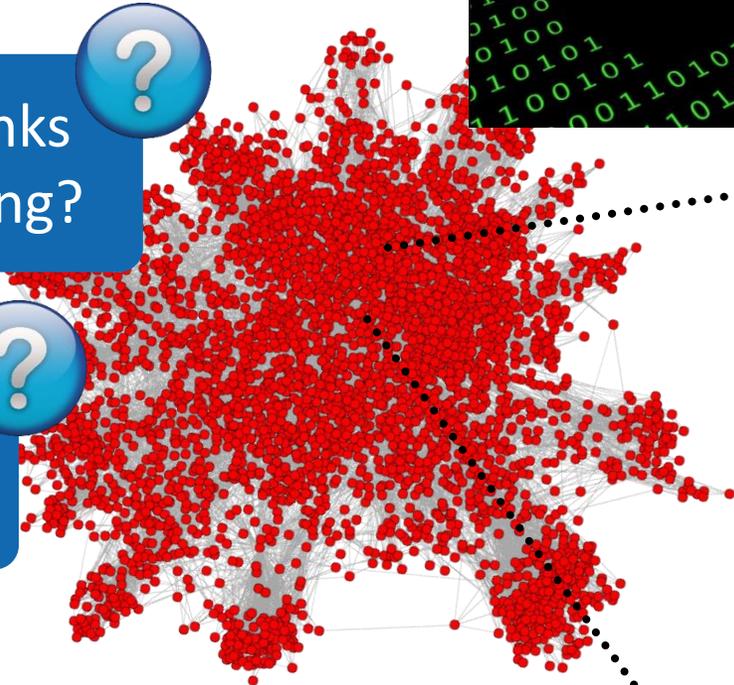
Case 2: Find Missing Data



Fixing missing data

Which links are missing?

Why do we care?



Case 2: Find Missing Data

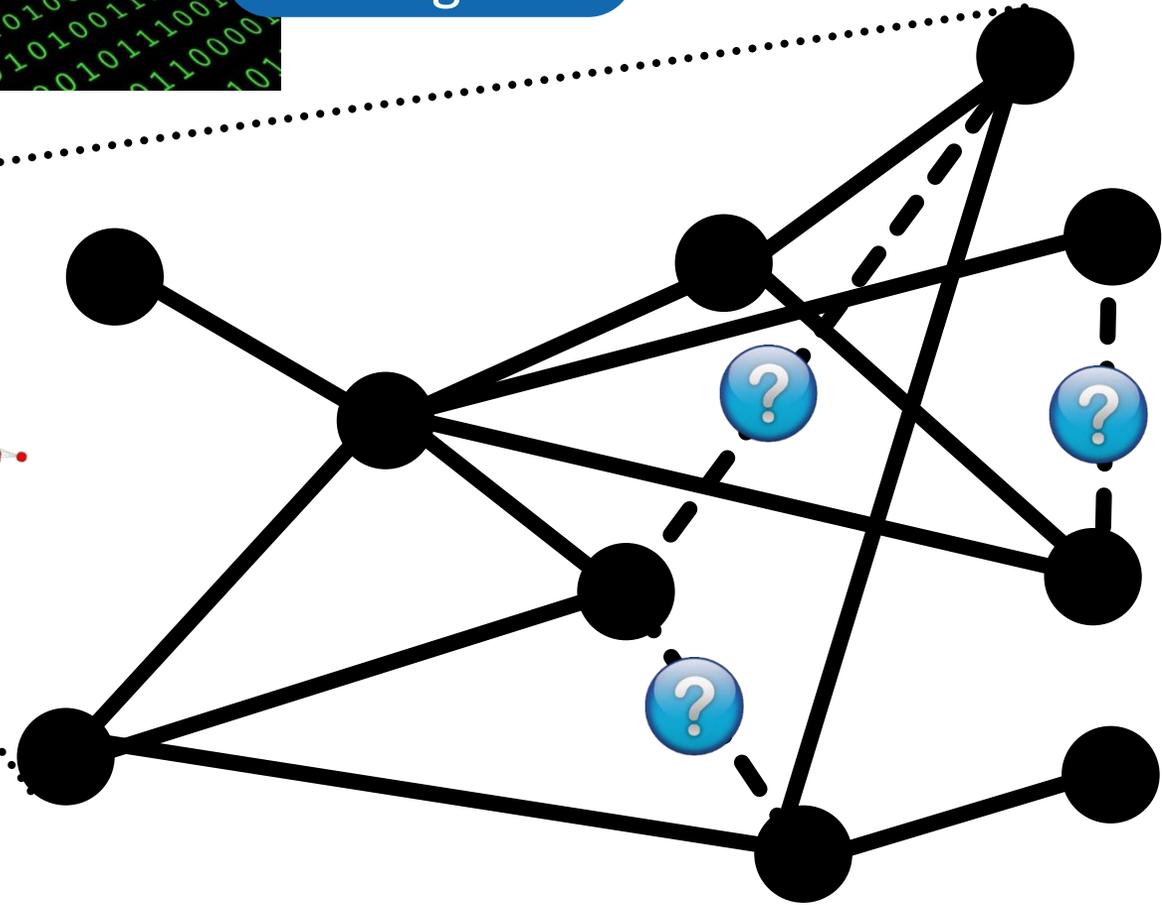
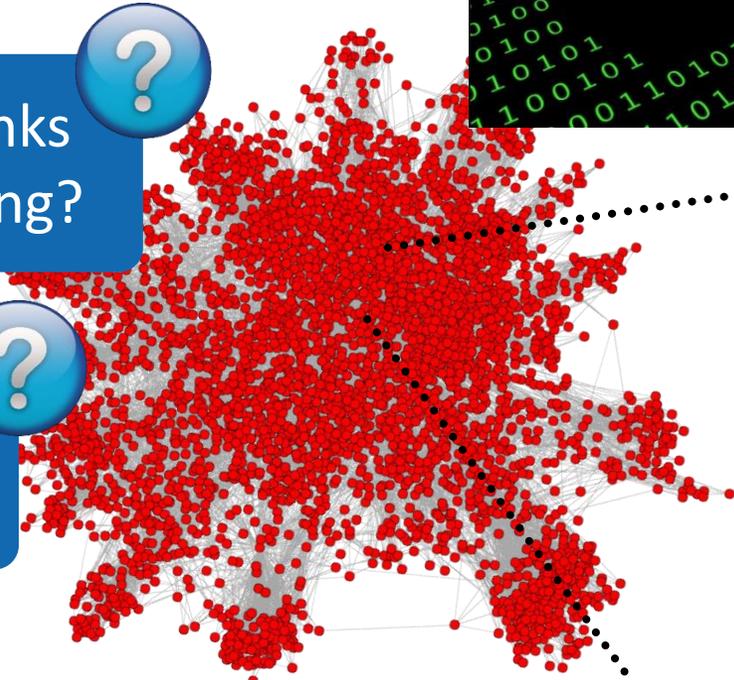


Fixing missing data

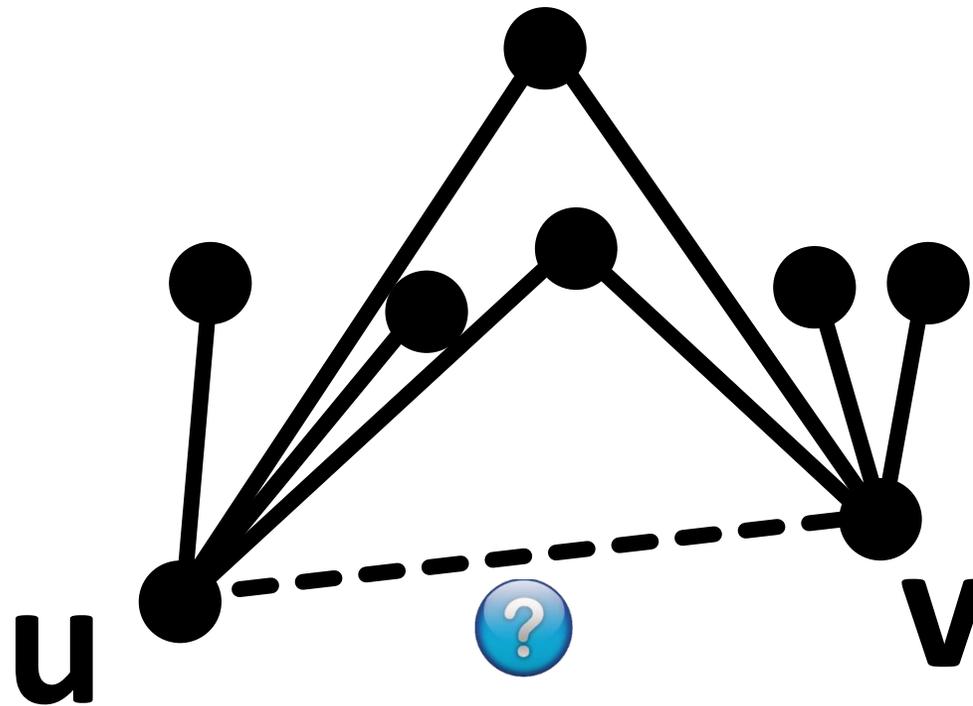
Which links are missing?

Why do we care?

Reduce experiment costs

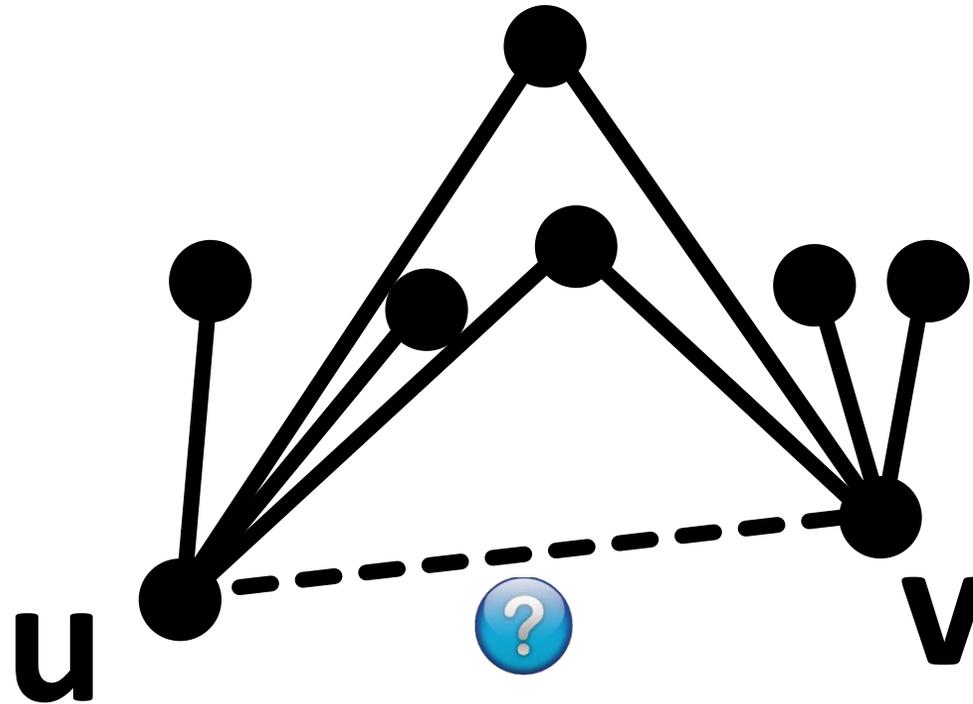


Warmup: Link Prediction



Warmup: Link Prediction

How to
assess?

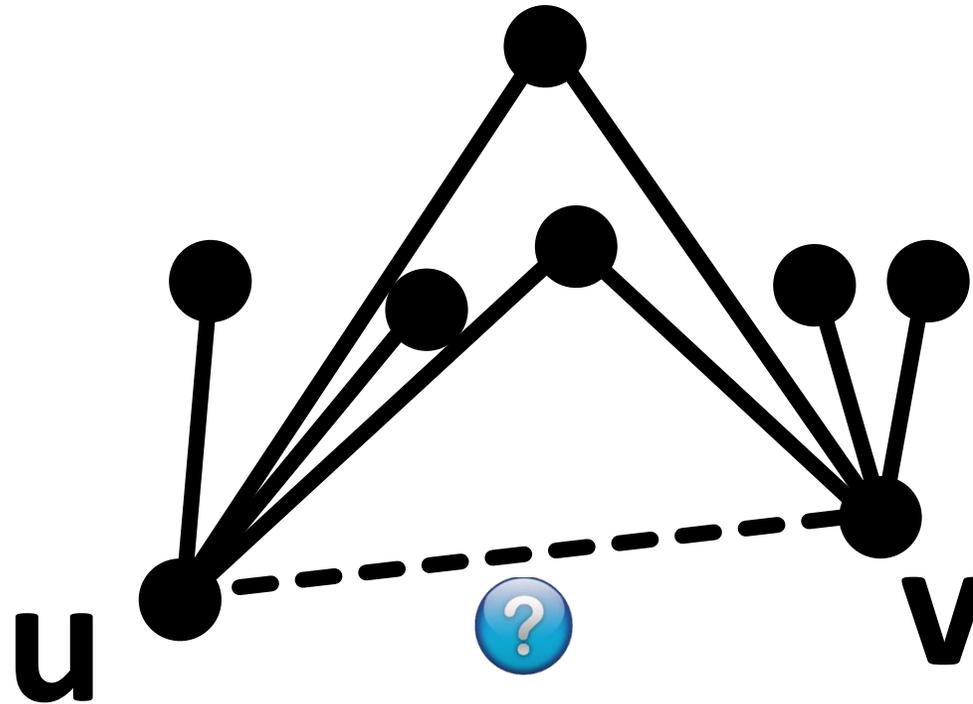


Warmup: Link Prediction

How to
assess?



$$s_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

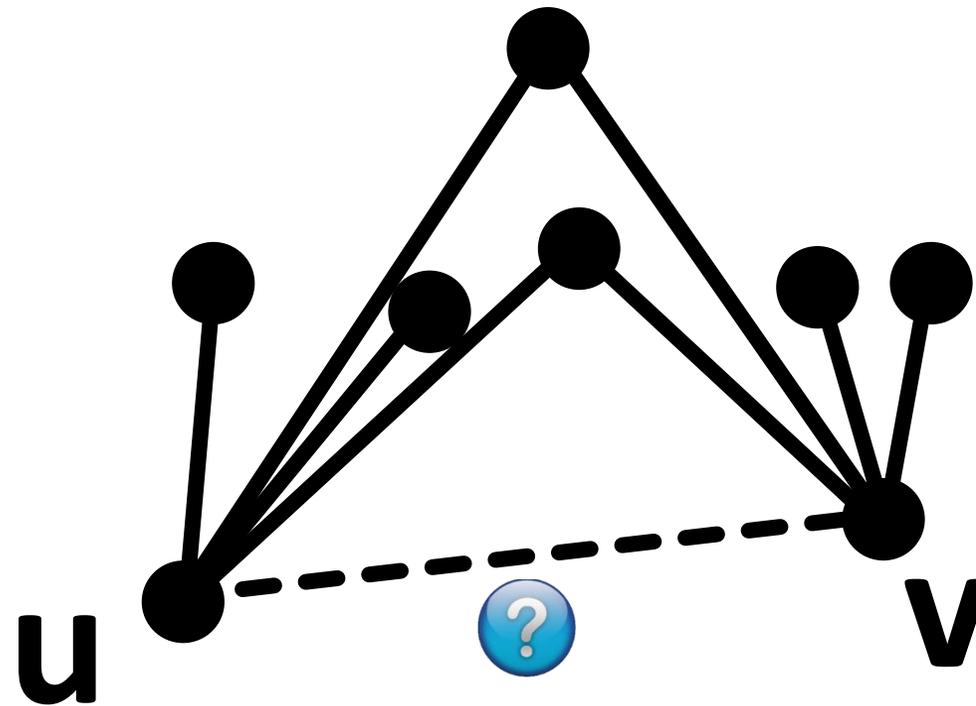


Warmup: Link Prediction

How to assess?

$$s_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

$$s_{u,v}^{Jaccard} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

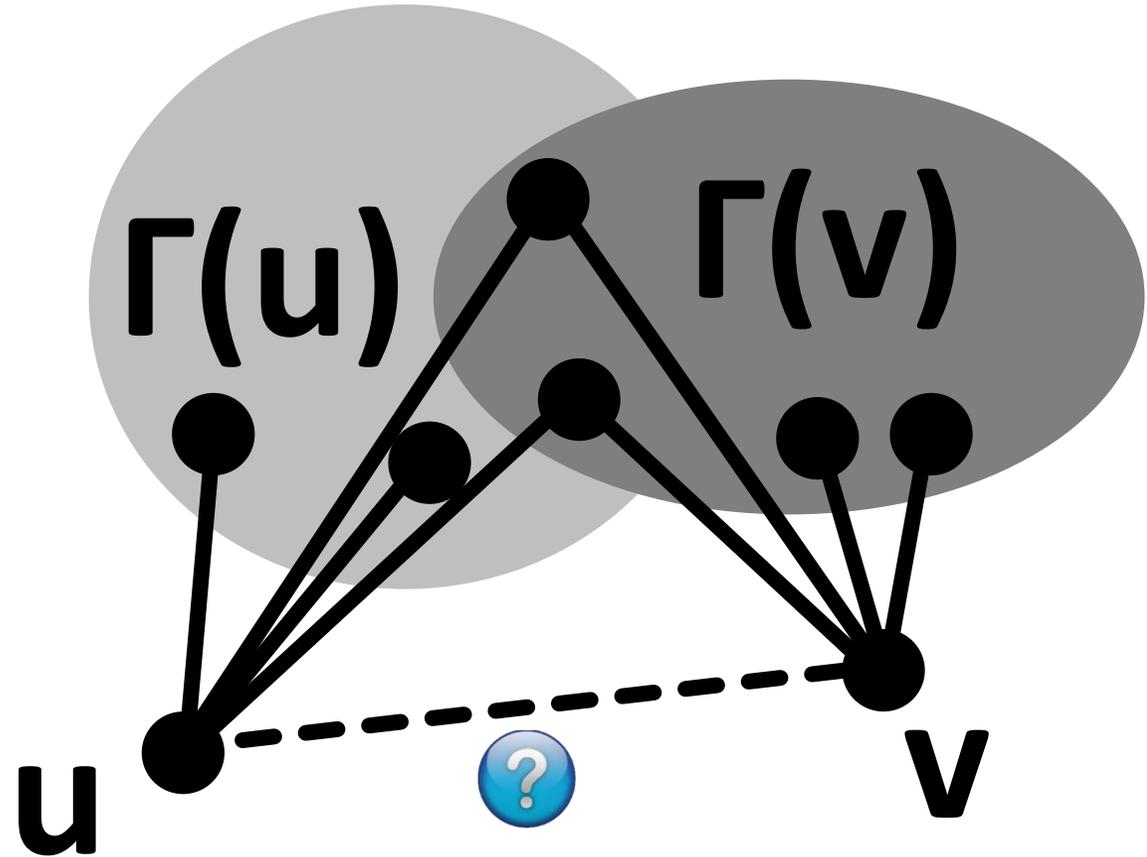


Warmup: Link Prediction

How to assess?

$$S_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

$$S_{u,v}^{Jaccard} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$



Warmup: Link Prediction

How to
assess?



$$s_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

$$s_{u,v}^{Jaccard} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

Warmup: Link Prediction

How to
assess?



$$s_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

$$s_{u,v}^{Jaccard} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

$$s_{u,v}^{Salton} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{d_u d_v}}$$

Warmup: Link Prediction

How to
assess?



$$S_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

$$S_{u,v}^{HPI} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\min\{d_u, d_v\}}$$

$$S_{u,v}^{HDI} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\max\{d_u, d_v\}}$$

$$S_{u,v}^{RA} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{d_z}$$

$$S_{u,v}^{Jaccard} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

$$S_{u,v}^{Salton} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{d_u d_v}}$$

$$S_{u,v}^{Sorensen} = \frac{2|\Gamma(u) \cap \Gamma(v)|}{d_u + d_v}$$

$$S_{u,v}^{AA} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log d_z}$$

$$S_{u,v}^{PAI} = |\Gamma(u)| |\Gamma(v)| = d_u d_v$$

Warmup: Link Prediction

How to assess?



One obtains a „score“
 $s(e) = s(u,v)$ for each
 (missing) link in a graph

$$s_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

$$s_{u,v}^{HPI} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\min\{d_u, d_v\}}$$

$$s_{u,v}^{HDI} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\max\{d_u, d_v\}}$$

$$s_{u,v}^{RA} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{d_z}$$

$$s_{u,v}^{Jaccard} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

$$s_{u,v}^{Salton} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{d_u d_v}}$$

$$s_{u,v}^{Sorensen} = \frac{2|\Gamma(u) \cap \Gamma(v)|}{d_u + d_v}$$

$$s_{u,v}^{AA} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log d_z}$$

$$s_{u,v}^{PAI} = |\Gamma(u)| |\Gamma(v)| = d_u d_v$$

Warmup: Link Prediction

How to assess?



One obtains a „score“
 $s(e) = s(u,v)$ for each
 (missing) link in a graph

The higher the score, the
 more probable a given link
 e is to appear in the graph

$$s_{u,v}^{CN} = |\Gamma(u) \cap \Gamma(v)|$$

$$s_{u,v}^{HPI} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\min\{d_u, d_v\}}$$

$$s_{u,v}^{HDI} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\max\{d_u, d_v\}}$$

$$s_{u,v}^{RA} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{d_z}$$

$$s_{u,v}^{Jaccard} = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$

$$s_{u,v}^{Salton} = \frac{|\Gamma(u) \cap \Gamma(v)|}{\sqrt{d_u d_v}}$$

$$s_{u,v}^{Sorensen} = \frac{2|\Gamma(u) \cap \Gamma(v)|}{d_u + d_v}$$

$$s_{u,v}^{AA} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log d_z}$$

$$s_{u,v}^{PAI} = |\Gamma(u)| |\Gamma(v)| = d_u d_v$$

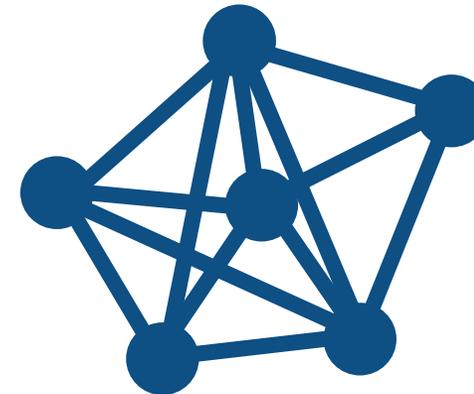
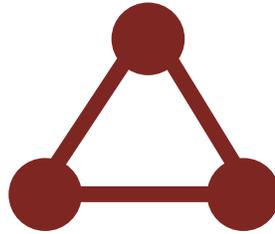
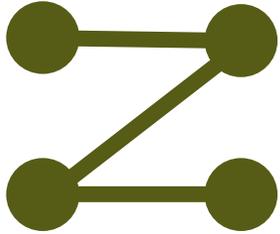
Idea: Generalize Link Prediction to Motifs

Idea: Generalize Link Prediction to Motifs

Motifs: small, recurring subgraphs (e.g. modelling molecules, gene interactions, groups of people)

Idea: Generalize Link Prediction to Motifs

Motifs: small, recurring subgraphs (e.g. modelling molecules, gene interactions, groups of people)



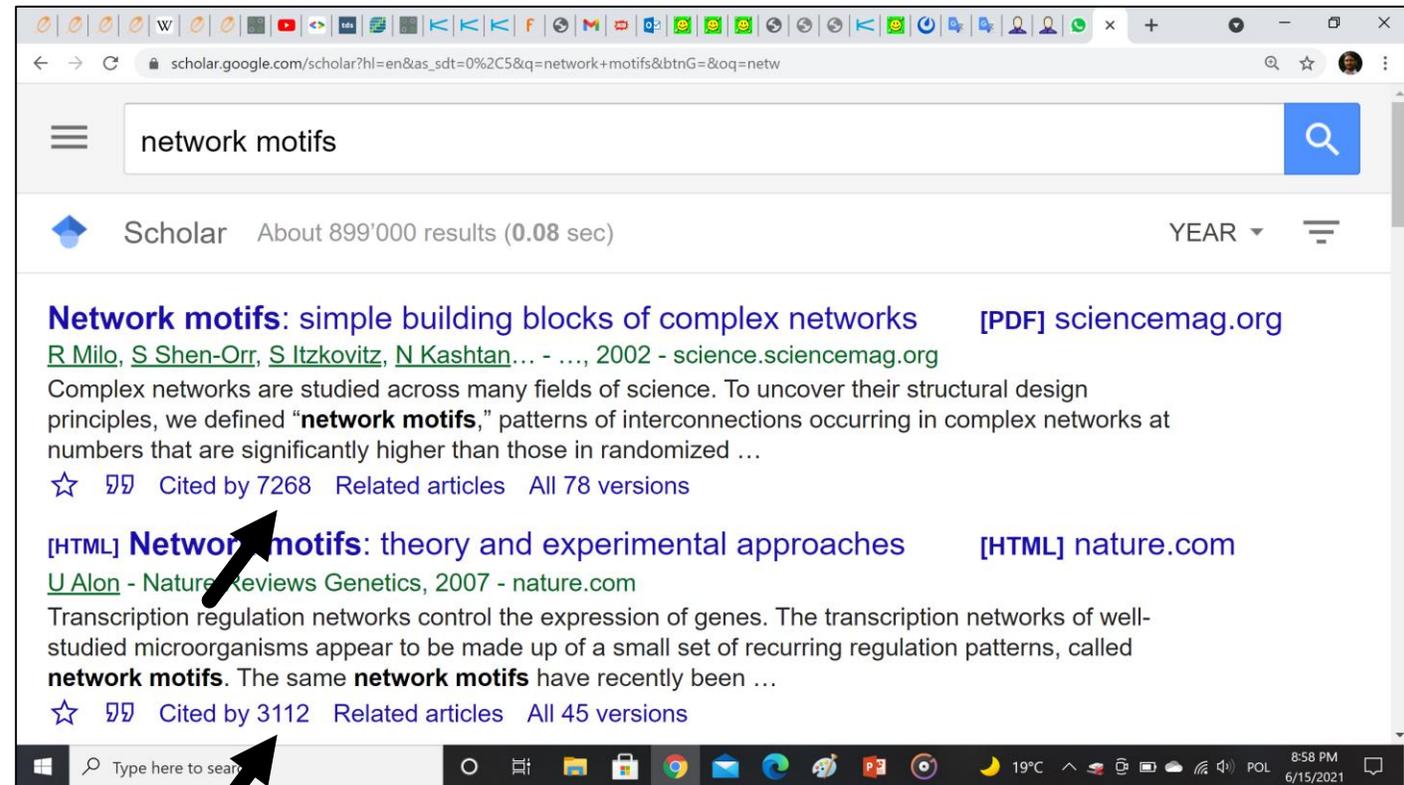
Motifs



Is this important?

Motifs

Is this important?



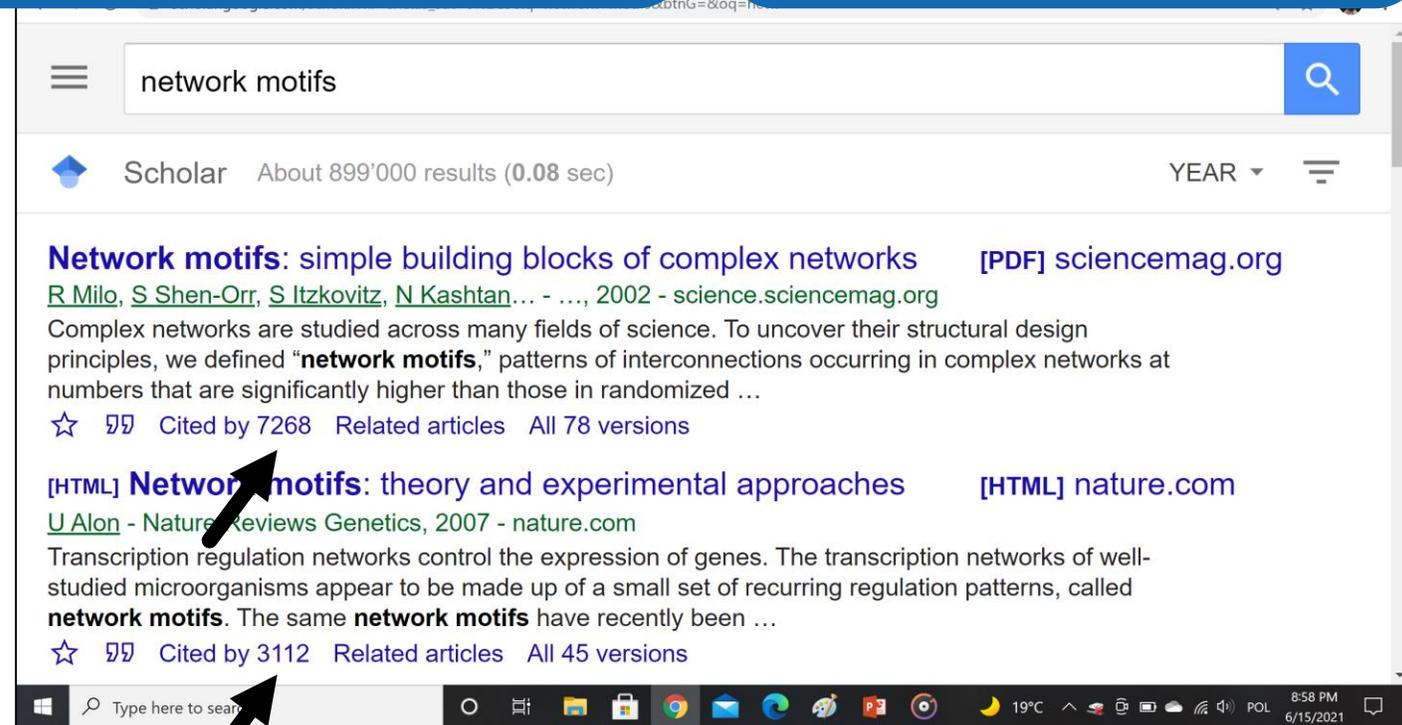
Motifs



Is this important?

A lot of work into motifs exists, recent renewed interest under the theme „Higher-order network organization“

Seeing a graph through the perspective of motifs instead of edges (“higher order”)



network motifs

Scholar About 899'000 results (0.08 sec) YEAR

Network motifs: simple building blocks of complex networks [PDF] sciencemag.org
[R Milo](#), [S Shen-Orr](#), [S Itzkovitz](#), [N Kashtan](#)... - ..., 2002 - science.sciencemag.org
 Complex networks are studied across many fields of science. To uncover their structural design principles, we defined “**network motifs**,” patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized ...
 ☆ Cited by 7268 Related articles All 78 versions

[HTML] **Network motifs: theory and experimental approaches** [HTML] nature.com
[U Alon](#) - Nature Reviews Genetics, 2007 - nature.com
 Transcription regulation networks control the expression of genes. The transcription networks of well-studied microorganisms appear to be made up of a small set of recurring regulation patterns, called **network motifs**. The same **network motifs** have recently been ...
 ☆ Cited by 3112 Related articles All 45 versions

Type here to search

Motifs



Is this important?

A lot of work into motifs exists, recent interest under the theme „Higher-order network organization“

Seeing a graph through the perspective of motifs instead of edges (“higher order”)

Google Scholar search results for "higher order organization".

Articles About 4'660'000 results (0.11 sec)

Any time
Since 2021
Since 2020
Since 2017
Custom range...

Sort by relevance
Sort by date

include patents
 include citations

Create alert

Higher-order organization of complex networks
AR Benson, DF Gleich, J Leskovec - Science, 2016 - science.sciencemag.org
Networks are a fundamental tool for understanding and modeling complex systems in physics, biology, neuroscience, engineering, and social science. Many networks are known to exhibit rich, lower-order connectivity patterns that can be captured at the level of individual nodes and edges. However, recent work has shown that higher-order connectivity patterns can be captured at the level of individual motifs. This work introduces a new framework for understanding and modeling complex networks based on motifs. This framework is based on the idea that motifs are the building blocks of complex networks. By understanding motifs, we can understand the structure and function of complex networks. This work introduces a new framework for understanding and modeling complex networks based on motifs. This framework is based on the idea that motifs are the building blocks of complex networks. By understanding motifs, we can understand the structure and function of complex networks.

Higher-order organization by mesoscale self-assembly and transfer hybrid nanostructures
H Cölfen, S Mann - Angewandte Chemie International Edition, 2003 - Wiley Online Library
The organization of nanostructures across extended length scales is a key challenge in the design of integrated materials with advanced functions. Current approaches tend to be based on physical methods, such as patterning, rather than the spontaneous chemical self-assembly of building blocks. This work introduces a new framework for understanding and modeling complex networks based on motifs. This framework is based on the idea that motifs are the building blocks of complex networks. By understanding motifs, we can understand the structure and function of complex networks.

Higher-order genome organization in human disease
T Misteli - Cold Spring Harbor perspectives in biology, 2010 - cshperspectives.cshlp.org
Genomes are organized into complex higher-order structures by folding of the DNA into chromatin fibers, chromosome domains, and ultimately chromosomes. The higher-order organization of genomes is functionally important for gene regulation and control of gene expression. This work introduces a new framework for understanding and modeling complex networks based on motifs. This framework is based on the idea that motifs are the building blocks of complex networks. By understanding motifs, we can understand the structure and function of complex networks.

[HTML] In vivo mapping of eukaryotic RNA interactomes reveals principles of higher-order organization and regulation
JGA Aw, Y Shen, A Wilm, M Sun, XN Lim, KL Boon... - Molecular cell, 2016 - Elsevier
Identifying pairwise RNA-RNA interactions is key to understanding how RNAs fold and interact with other RNAs inside the cell. We present a high-throughput approach, sequencing of psoralen crosslinked, ligated, and selected hybrids (SPLASH), that maps higher-order RNA-RNA interactions. This work introduces a new framework for understanding and modeling complex networks based on motifs. This framework is based on the idea that motifs are the building blocks of complex networks. By understanding motifs, we can understand the structure and function of complex networks.

Google Scholar search results for "network motifs".

network motifs

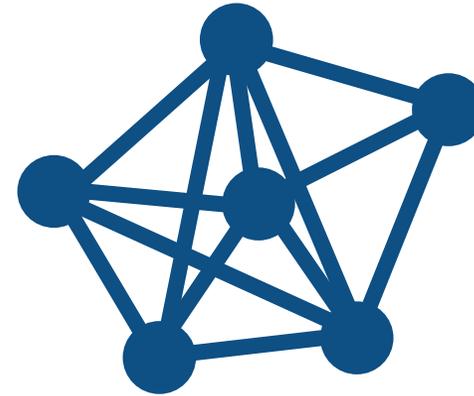
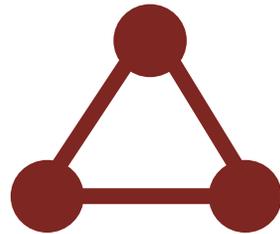
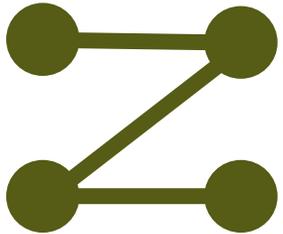
Scholar About 899'000 results (0.08 sec) YEAR

Network motifs: simple building blocks of complex networks [PDF] sciencemag.org
R Milo, S Shen-Orr, S Itzkovitz, N Kashtan... - ..., 2002 - science.sciencemag.org
Complex networks are studied across many fields of science. To uncover their structural design principles, we defined "network motifs," patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized networks. This work introduces a new framework for understanding and modeling complex networks based on motifs. This framework is based on the idea that motifs are the building blocks of complex networks. By understanding motifs, we can understand the structure and function of complex networks.

[HTML] Network motifs: theory and experimental approaches [HTML] nature.com
U Alon - Nature Reviews Genetics, 2007 - nature.com
Transcription regulation networks control the expression of genes. The transcription networks of well-studied microorganisms appear to be made up of a small set of recurring regulation patterns, called network motifs. The same network motifs have recently been identified in a wide range of other organisms. This work introduces a new framework for understanding and modeling complex networks based on motifs. This framework is based on the idea that motifs are the building blocks of complex networks. By understanding motifs, we can understand the structure and function of complex networks.

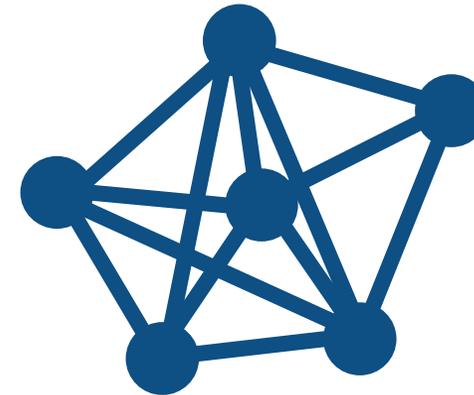
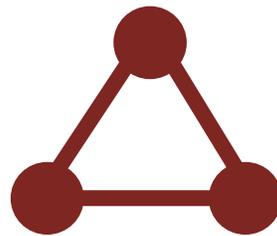
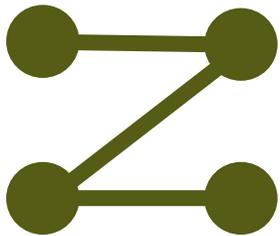
Idea: Generalize Link Prediction to Motifs

Motifs: small, recurring subgraphs (e.g. modelling molecules, gene interactions, groups of people)



Idea: Generalize Link Prediction to Motifs

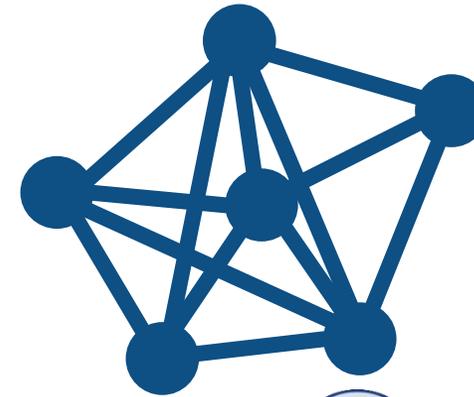
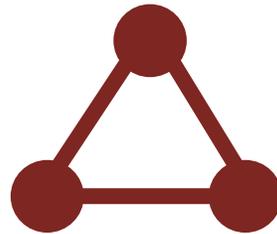
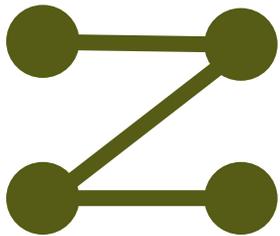
Motifs: small, recurring subgraphs (e.g. modelling molecules, gene interactions, groups of people)



Link prediction is well understood for, well, links (scores assigned to links, similarity based methods, etc.)

Idea: Generalize Link Prediction to Motifs

Motifs: small, recurring subgraphs (e.g. modelling molecules, gene interactions, groups of people)



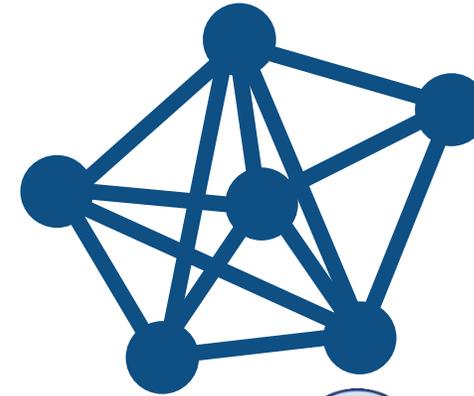
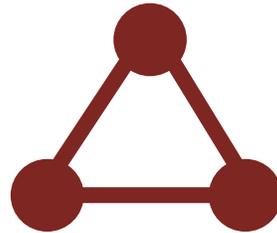
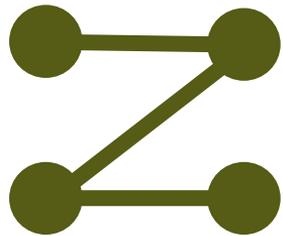
Link prediction is well understood for, well, links (scores assigned to links, similarity based methods, etc.)

How to generalize to motifs?



Idea: Generalize Link Prediction to Motifs

Motifs: small, recurring subgraphs (e.g. modelling molecules, gene interactions, groups of people)



Link prediction is well understood for, well, links (scores assigned to links, similarity based methods, etc.)

How to generalize to motifs?

General vision: assign some score to motifs (make them comparable)

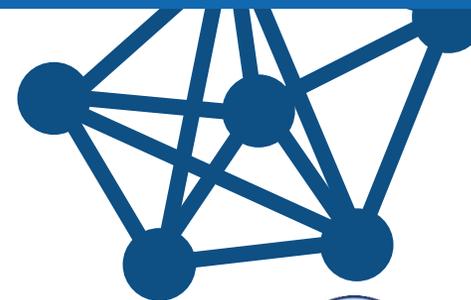
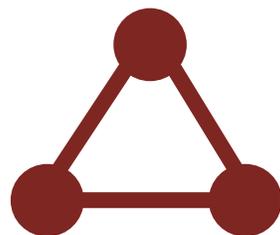
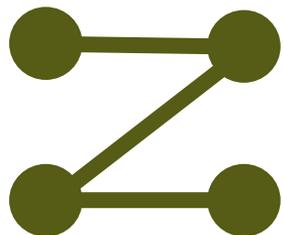
Motifs with higher scores are more probable

Idea: Generalize Link Prediction



...but there are so many differences to link prediction!

Motifs: small, recurring subgraphs
interactions, groups of people



Link prediction is well understood for, well, links (scores assigned to links, similarity based methods, etc.)

How to generalize to motifs?



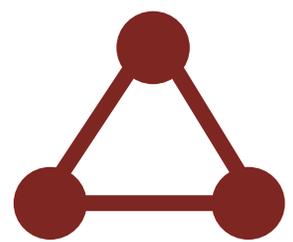
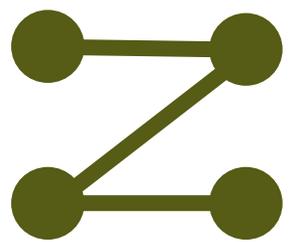
General vision: assign some score to motifs (make them comparable)

Motifs with higher scores are more probable

Idea: Generalize Link Prediction

...but there are so many differences to link prediction!

Motifs: small, recurring subgraphs
 interactions, groups of people



Let's go over them [1] ...

Link prediction is well understood for, well, links (scores assigned to links, similarity based methods, etc.)

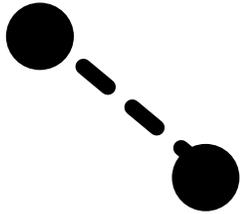
How to generalize to motifs?

General vision: assign some score to motifs (make them comparable)

Motifs with higher scores are more probable

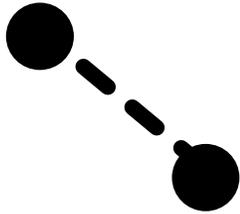
[1] M. Besta et al.: "Motif prediction with graph neural networks", KDD'22

Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set



Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

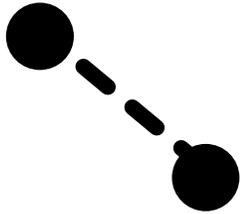
Link prediction:



A link is either there or not there

Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:

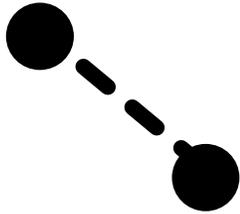


A link is either there or not there

Motif prediction:

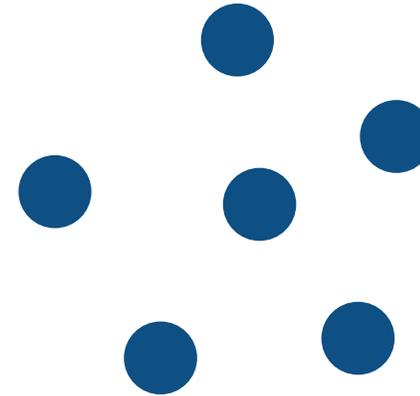
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



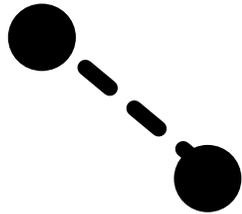
A link is either there or not there

Motif prediction:



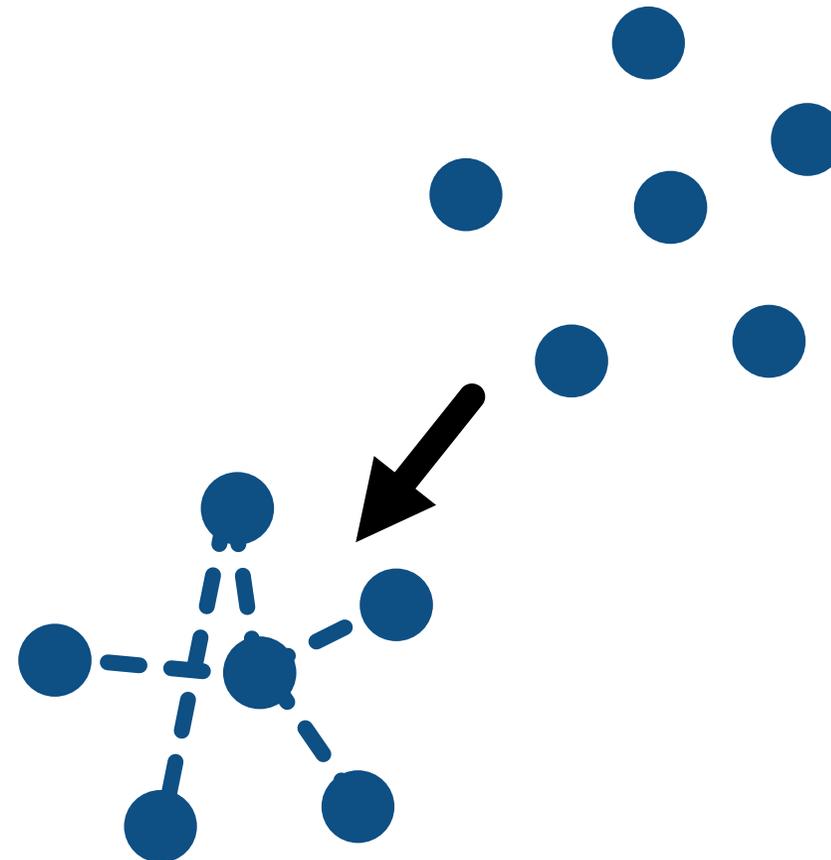
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



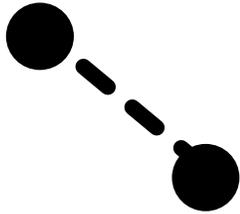
A link is either there or not there

Motif prediction:



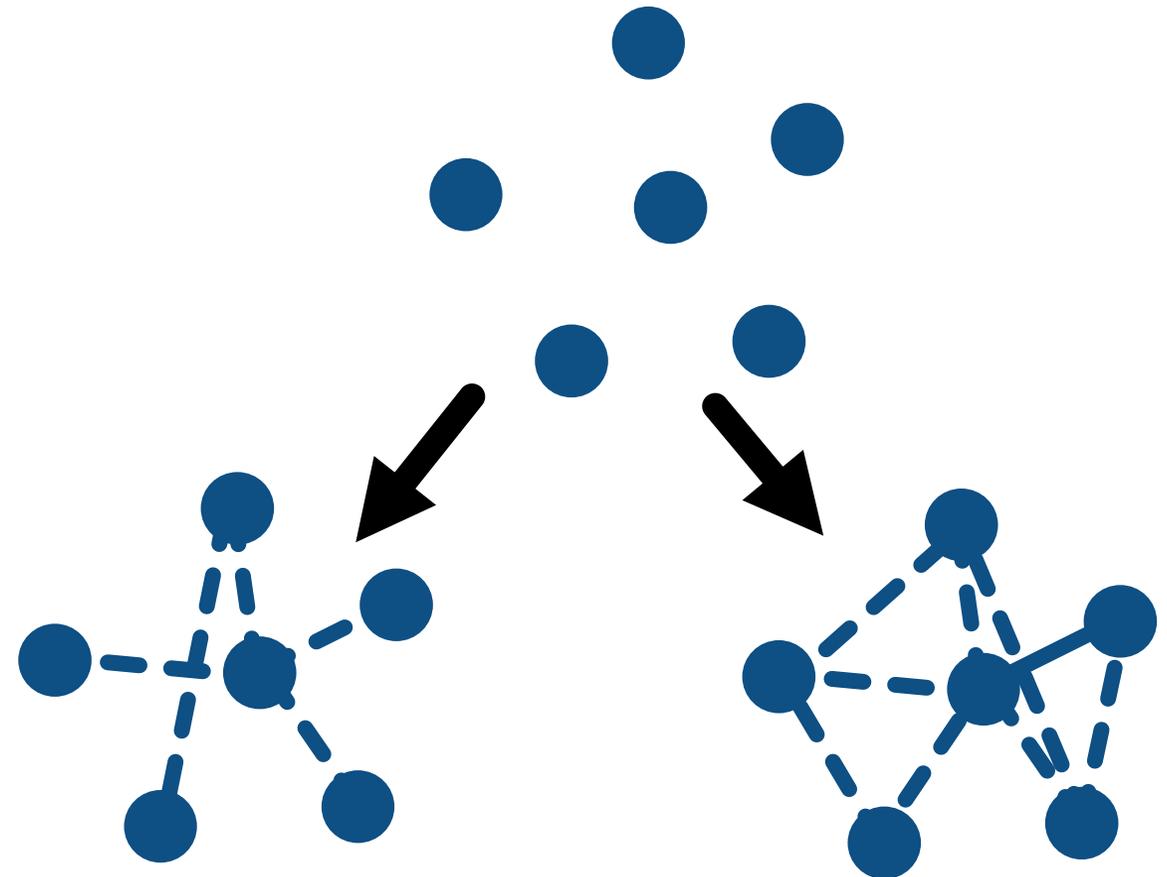
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



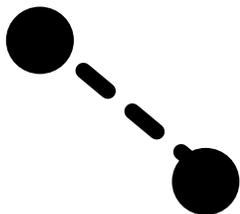
A link is either there or not there

Motif prediction:



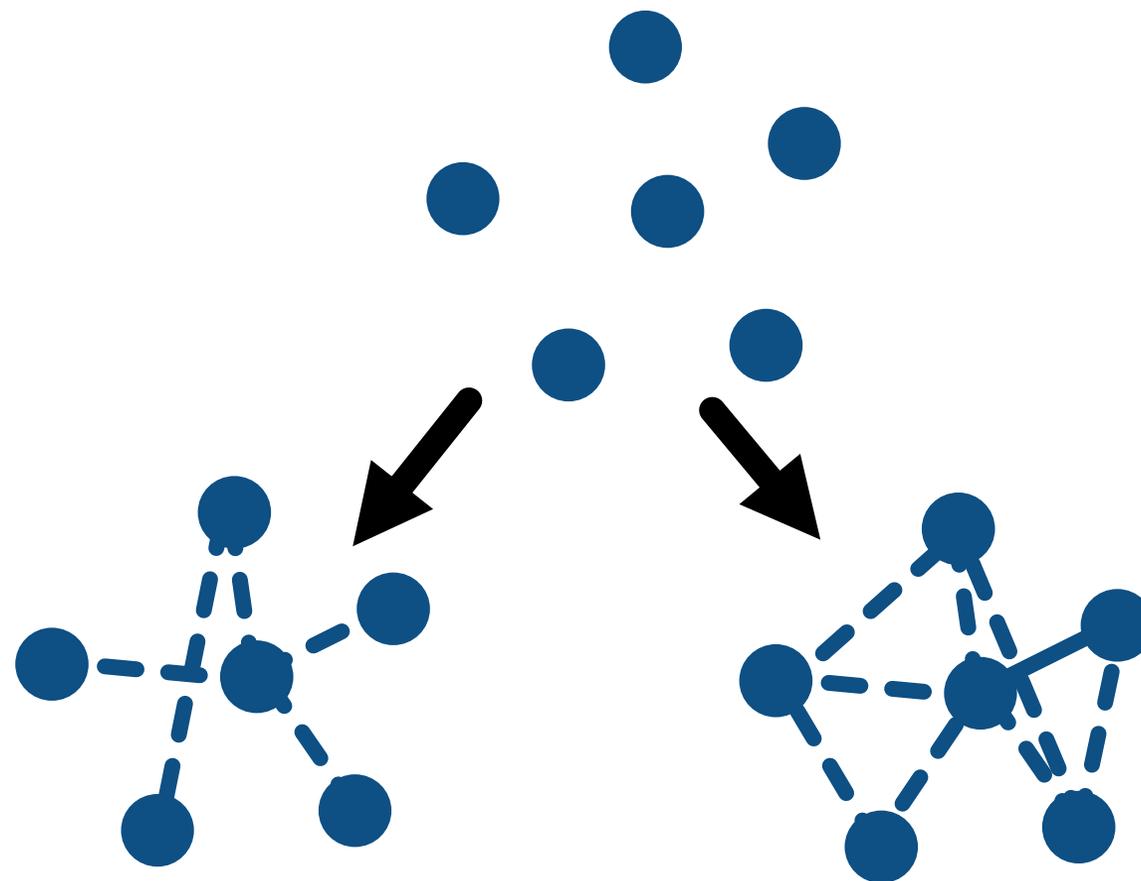
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



A link is either there or not there

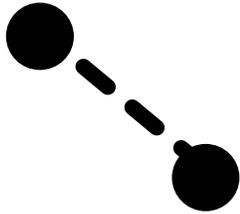
Motif prediction:



How to consider such diversity of possible patterns in score functions?

Difference 2: Incoming Motifs May Have Existing Edge

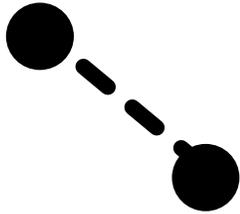
Link prediction:



Motif prediction:

Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:

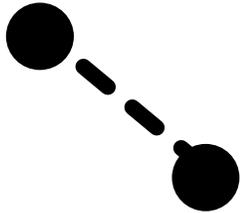


A link to be predicted
does not exist

Motif prediction:

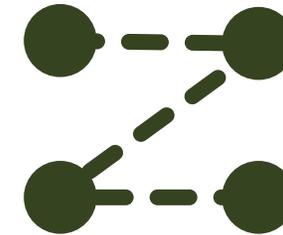
Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



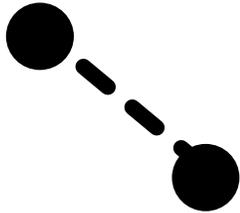
A link to be predicted
does not exist

Motif prediction:



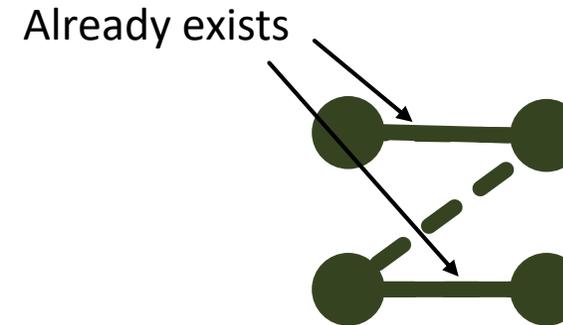
Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



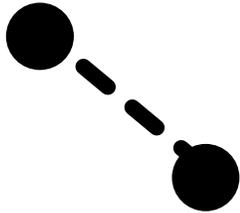
A link to be predicted
does not exist

Motif prediction:



Difference 2: Incoming Motifs May Have Existing Edge

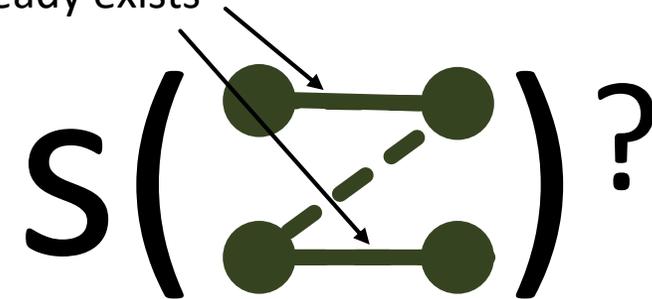
Link prediction:



A link to be predicted
does not exist

Motif prediction:

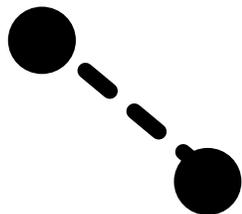
Already exists



How to consider
such edges in the
score functions?

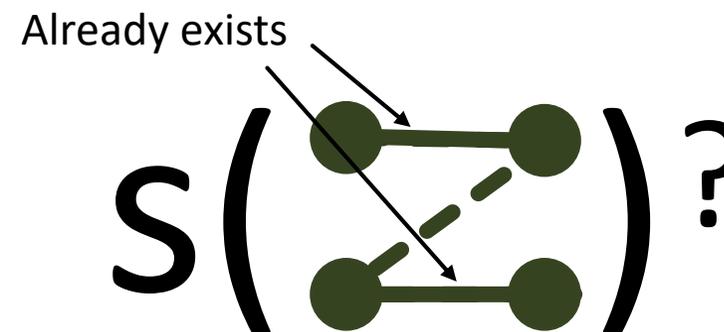
Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



A link to be predicted does not exist

Motif prediction:

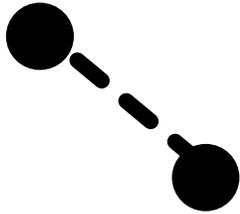


How to consider such edges in the score functions?

Example: some existing relationships in a group of people

Difference 3: There May Be “Deal-Breaker” Edges

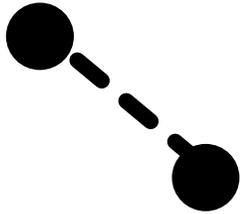
Link prediction:



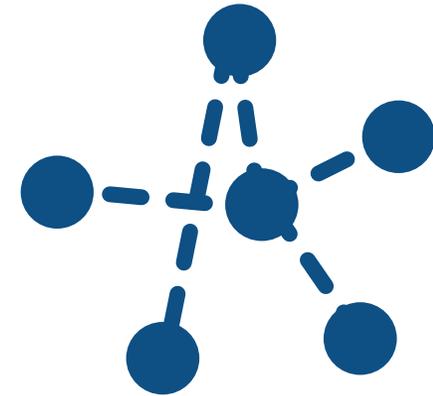
Motif prediction:

Difference 3: There May Be “Deal-Breaker” Edges

Link prediction:

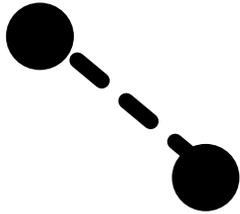


Motif prediction:



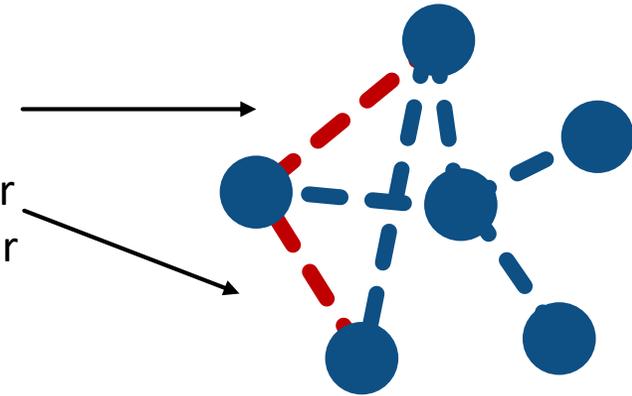
Difference 3: There May Be “Deal-Breaker” Edges

Link prediction:



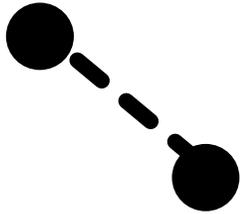
Motif prediction:

We don't want these links!
 (i.e., these links appearing
 would make it impossible for
 a motif in question to appear



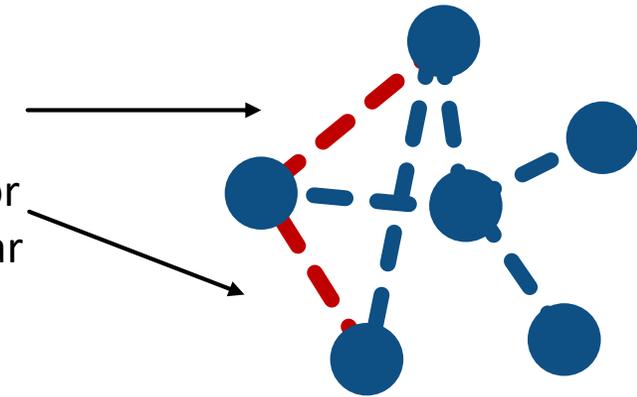
Difference 3: There May Be “Deal-Breaker” Edges

Link prediction:



Motif prediction:

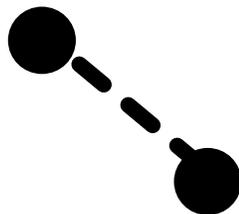
We don't want these links!
(i.e., these links appearing
would make it impossible for
a motif in question to appear



Example:
chemical bonds

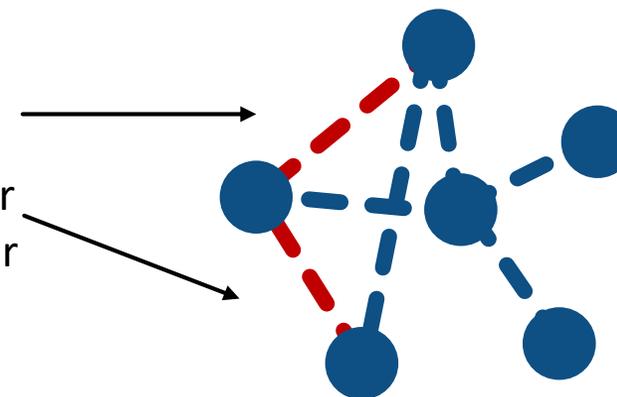
Difference 3: There May Be “Deal-Breaker” Edges

Link prediction:



Motif prediction:

We don't want these links!
(i.e., these links appearing
would make it impossible for
a motif in question to appear

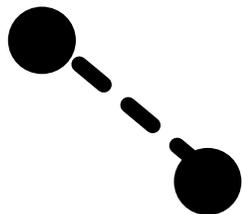


How to consider
such edges in the
score functions?

Example:
chemical bonds

Difference 3: There May Be “Deal-Breaker” Edges

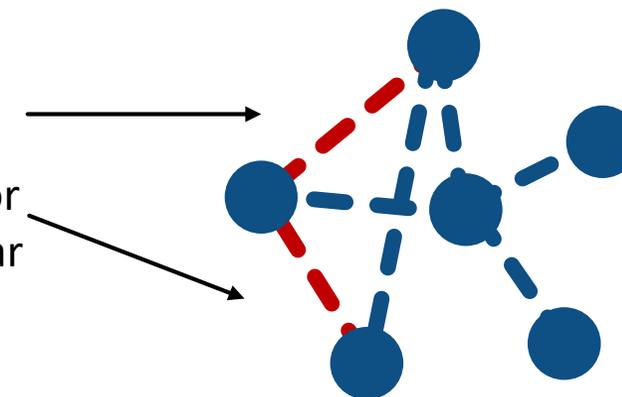
Link prediction:



No such effect (a link to be predicted is never a „deal breaker”)

Motif prediction:

We don't want these links!
(i.e., these links appearing would make it impossible for a motif in question to appear)

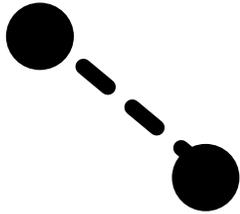


How to consider such edges in the score functions?

Example:
chemical bonds

Difference 4: Motif Prediction Query May Depend on Vertex Labeling

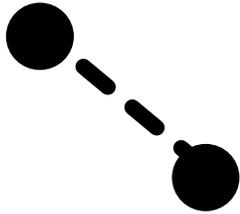
Link prediction:



Motif prediction:

Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:

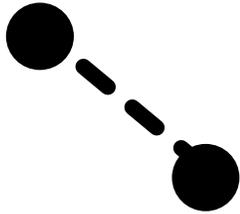


Motif prediction:

We want **this**:

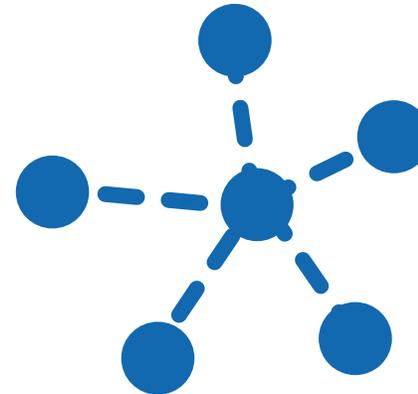
Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:



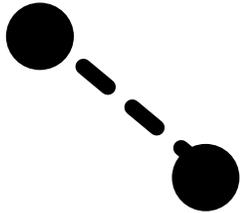
Motif prediction:

We want **this**:



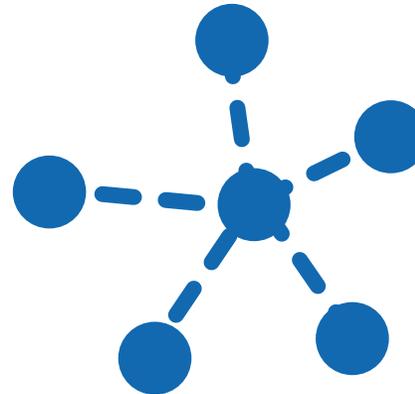
Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:

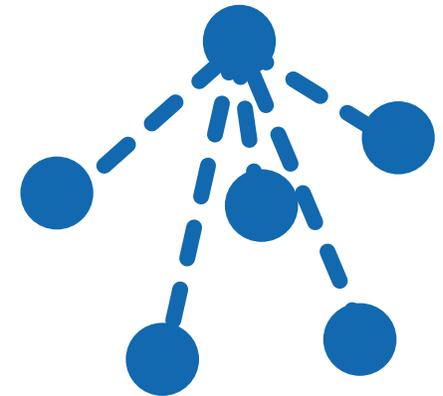


Motif prediction:

We want **this**:

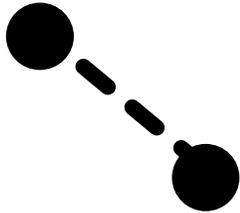


...but not this:



Difference 4: Motif Prediction Query May Depend on Vertex Labeling

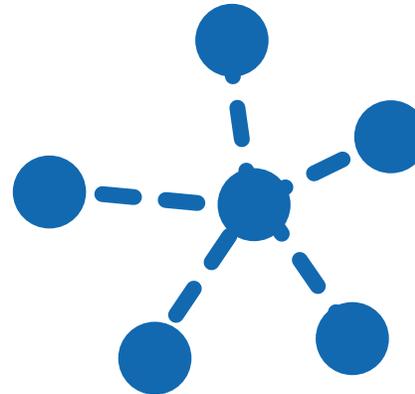
Link prediction:



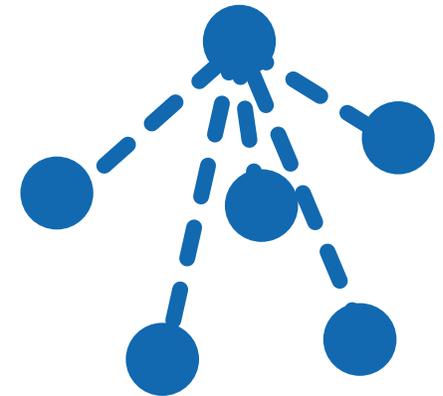
No such effect (not enough room with a single link)

Motif prediction:

We want **this**:

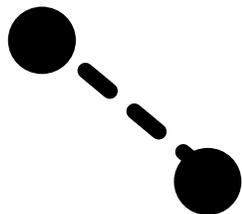


...but **not** this:



Difference 4: Motif Prediction Query May Depend on Vertex Labeling

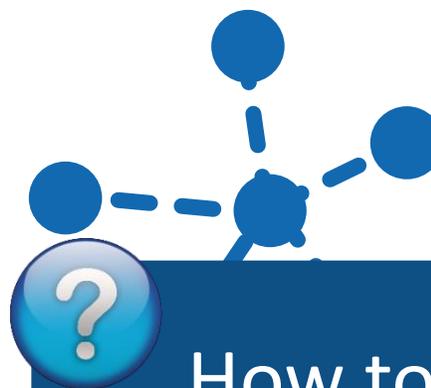
Link prediction:



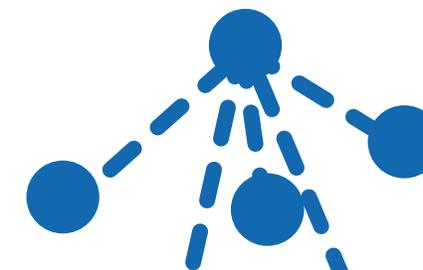
No such effect (not enough room with a single link)

Motif prediction:

We want **this**:



...but **not** this:



How to formulate motif prediction, considering all these differences?

Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

Motif Prediction Score Functions

A motif: $M = (V_M, E_M)$

Starting Simple: Motif Scores Based on Independent Links

Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\varepsilon}$$

Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\varepsilon}$$

Motif edges that
do **not** yet exist



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that
do **not** yet exist

Motif edges that
already exist

Motif Prediction Score Functions

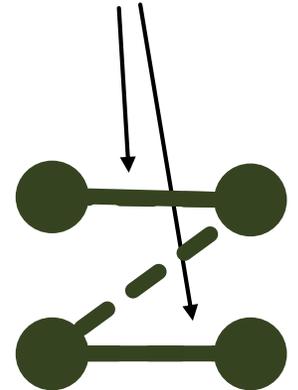
Starting Simple: Motif Scores Based on Independent Links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that
do **not** yet exist

Motif edges that
already exist



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

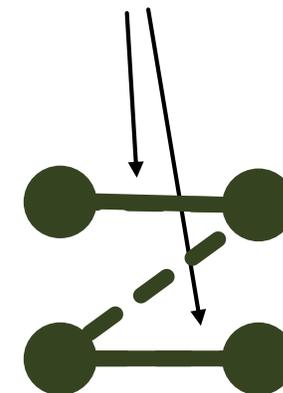
$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

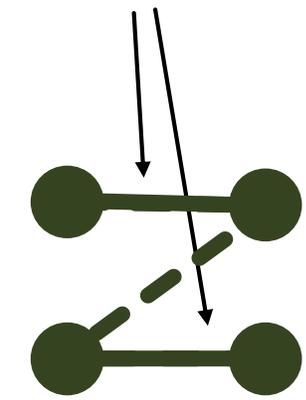
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist

$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

A motif: $M = (V_M, E_M)$

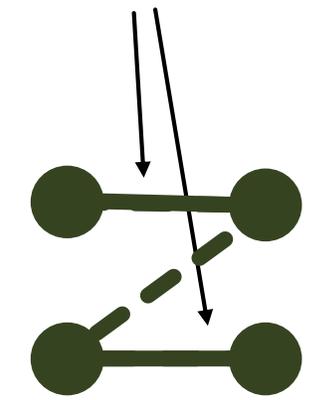
$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist

$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

$s(e)$ is any link prediction score which outputs into $[0, 1]$ (e.g., Jaccard)



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

A motif: $M = (V_M, E_M)$

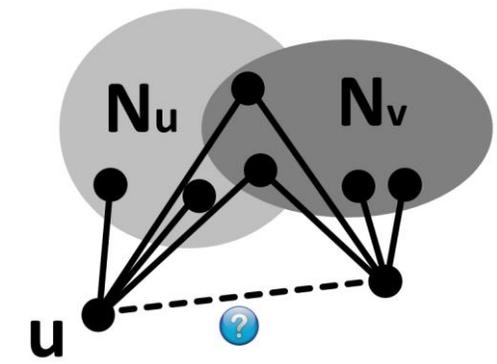
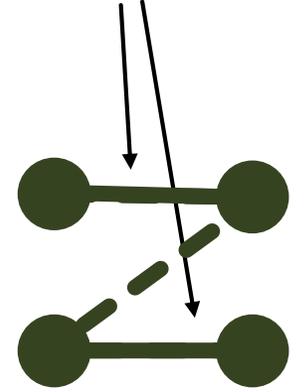
$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist

$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

$s(e)$ is any link prediction score which outputs into $[0, 1]$ (e.g., Jaccard)



[1] M. Besta et al.: "Motif prediction with graph neural networks", KDD'22

Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

For edges that already exist, we set $s(e) = 1$ (i.e., we assume that a motif is more likely to appear if the edges that participate in that motif are also more likely)

$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

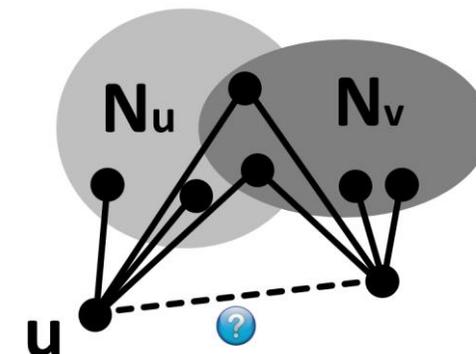
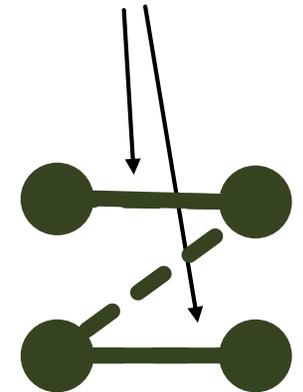
$s(e)$ is any link prediction score which outputs into $[0, 1]$ (e.g., Jaccard)

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

For edges that already exist, we set $s(e) = 1$ (i.e., we assume that a motif is more likely to appear if the edges that participate in that motif are also more likely)

$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

$\in [0, 1]$ by construction

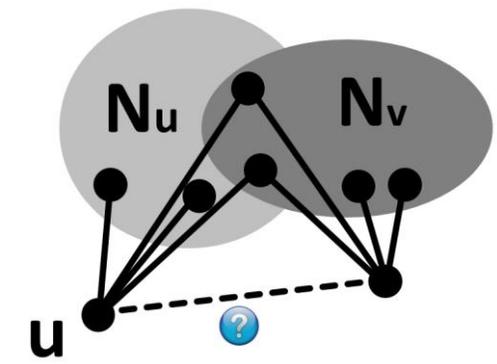
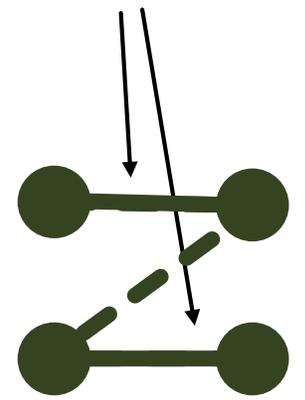
$s(e)$ is any link prediction score which outputs into $[0, 1]$ (e.g., Jaccard)

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

For edges that already exist, we set $s(e) = 1$ (i.e., we assume that a motif is more likely to appear if the edges that participate in that motif are also more likely)

$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

$\in [0, 1]$ by construction

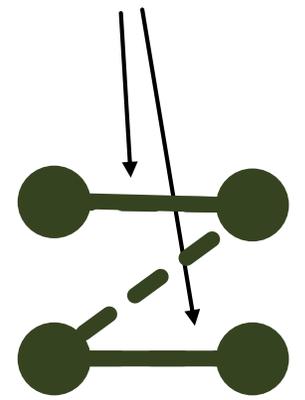
$s(e)$ is any link prediction score which outputs into $[0, 1]$ (e.g., Jaccard)

A motif: $M = (V_M, E_M)$

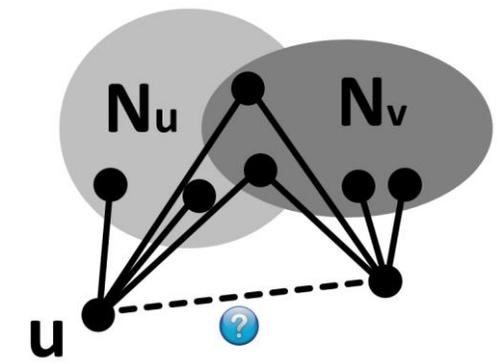
$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist



Example heuristic for the Jaccard score $s(e)$ for links:



Motif Prediction Score Functions

Starting Simple: Motif Scores Based on Independent Links

For edges that already exist, we set $s(e) = 1$ (i.e., we assume that a motif is more likely to appear if the edges that participate in that motif are also more likely)

$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

$\in [0, 1]$ by construction

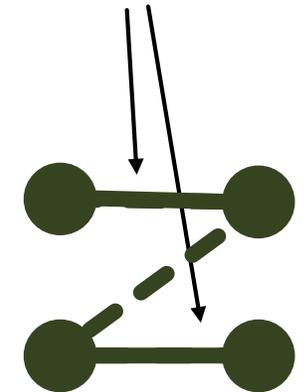
$s(e)$ is any link prediction score which outputs into $[0, 1]$ (e.g., Jaccard)

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

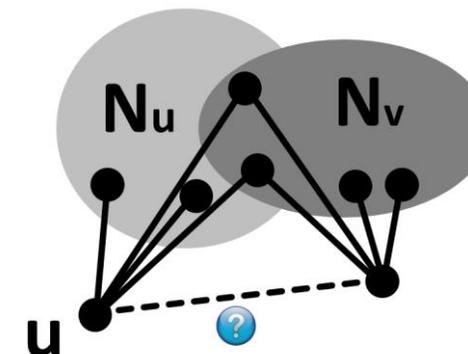
Motif edges that do **not** yet exist

Motif edges that **already** exist



Example heuristic for the Jaccard score $s(e)$ for links:

$$s_{\perp}(M)^J = \prod_{e_{u,v} \in E_{M,\mathcal{N}}} \frac{|N_u \cap N_v|}{|N_u \cup N_v|}$$



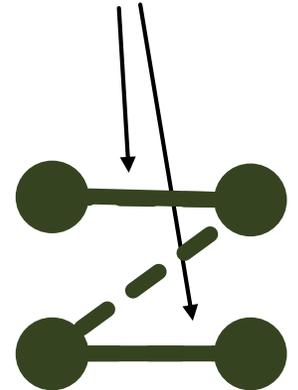
Motif Prediction Score Functions

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist



Example heuristic for the Jaccard score $s(e)$ for links:

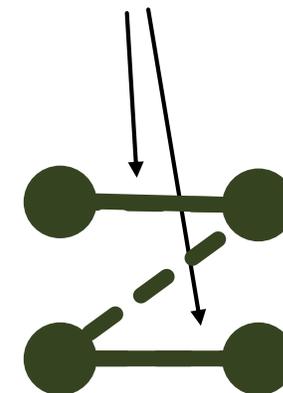
Motif Prediction Score Functions

What if the arrival of some (motif) links impacts the chances for other motif links to appear?

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

\nearrow Motif edges that do **not** yet exist
 \nwarrow Motif edges that **already** exist



Example heuristic for the Jaccard score $s(e)$ for links:

Motif Prediction Score Functions

What if the arrival of some (motif) links impacts the chances for other motif links to appear?

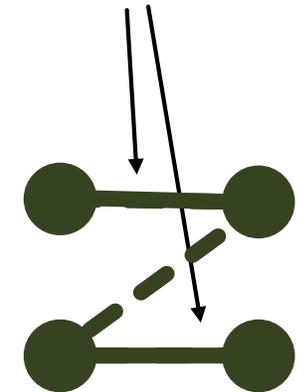
Adding correlation between links

Example heuristic for the Jaccard score $s(e)$ for links:

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist Motif edges that **already** exist



Motif Prediction Score Functions

What if the arrival of some (motif) links impacts the chances for other motif links to appear?

Adding correlation between links

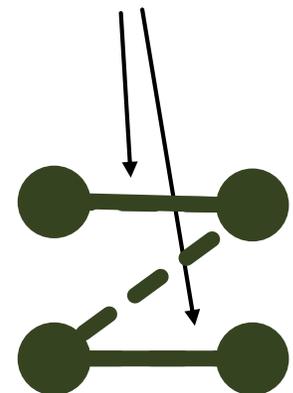
$$s(M) = f(\mathbf{s}(e)) = \langle \mathbf{w}, \mathbf{s}(e) \rangle$$

Example heuristic for the Jaccard score $s(e)$ for links:

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist Motif edges that **already** exist



Motif Prediction Score Functions

What if the arrival of some (motif) links impacts the chances for other motif links to appear?

Adding correlation between links

$$s(M) = f(s(e)) = \langle \mathbf{w}, \mathbf{s}(e) \rangle$$

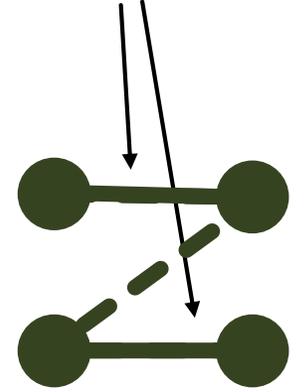
$s(M)$ is a convex combination of the vector of link prediction scores $s(e)$

Example heuristic for the Jaccard score $s(e)$ for links:

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist Motif edges that **already** exist



Motif Prediction Score Functions

What if the arrival of some (motif) links impacts the chances for other motif links to appear?

Adding correlation between links

The weight vector (incorporates user's domain knowledge)

$$s(M) = f(\mathbf{s}(e)) = \langle \mathbf{w}, \mathbf{s}(e) \rangle$$

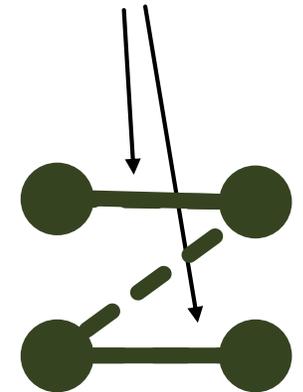
$s(M)$ is a convex combination of the vector of link prediction scores $s(e)$

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist



Example heuristic for the Jaccard score $s(e)$ for links:

Motif Prediction Score Functions

What if the arrival of some (motif) links impacts the chances for other motif links to appear?

Adding correlation between links

The weight vector (incorporates user's domain knowledge)

$$s(M) = f(s(e)) = \langle \mathbf{w}, \mathbf{s}(e) \rangle$$

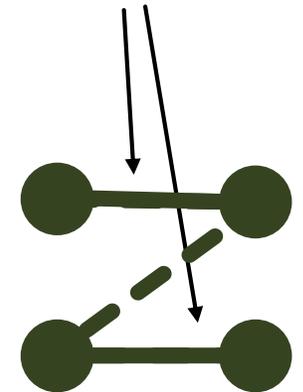
$s(M)$ is a convex combination of the vector of link prediction scores $s(e)$

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

Motif edges that do **not** yet exist

Motif edges that **already** exist



Example heuristic for the Jaccard score $s(e)$ for links:

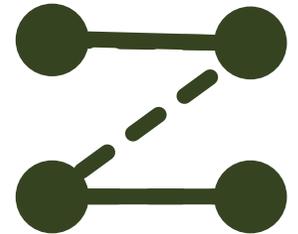
$$s(M)^J = \frac{1}{|E_M|} \left(\sum_{e_{u,v} \in E_{M,\mathcal{N}}} \frac{|N_u \cap N_v|}{|N_u \cup N_v|} + |E_{M,\mathcal{E}}| \right)$$

Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\varepsilon}$$



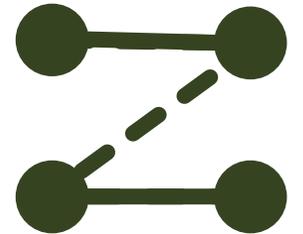
Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\varepsilon}$$



Motif Prediction Score Functions

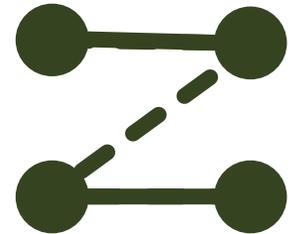
What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\varepsilon}$$

$$E_M^* = E_M$$



Motif Prediction Score Functions

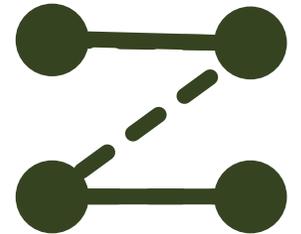
What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

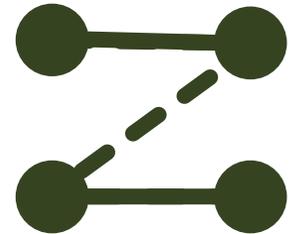
Incorporating deal-breaker links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif
edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

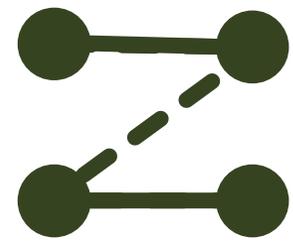
Incorporating deal-breaker links

A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges Deal-breaker edges



[1] M. Besta et al.: "Motif prediction with graph neural networks", KDD'22

Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

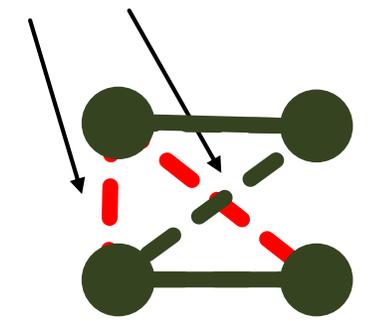
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

$$s(M) = f(\mathbf{s}(\mathbf{e})) = \langle \mathbf{w}, \mathbf{s}(\mathbf{e}) \rangle$$

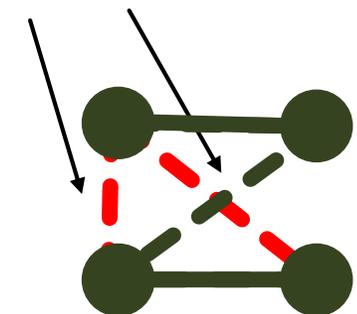
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

$$s^*(M) = f(s^*(e)) = \max(0, \langle \mathbf{w}, s^*(e) \rangle)$$

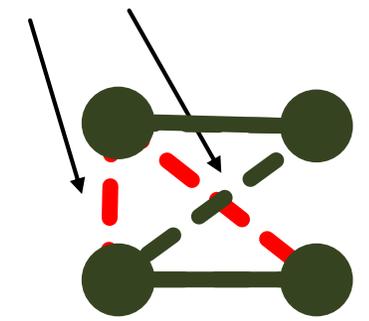
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

The weight vector (incorporates user's domain knowledge)

$$s^*(M) = f(s^*(e)) = \max(0, \langle \mathbf{w}, s^*(e) \rangle)$$

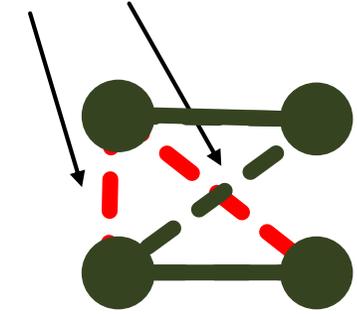
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

The weight vector (incorporates user's domain knowledge)

$$s^*(M) = f(s^*(e)) = \max(0, \langle \mathbf{w}, s^*(e) \rangle)$$

A deal-breaker edge that **does not** exist (i.e., it might arrive later):

$$s_i^*(e) = -s_i(e)$$

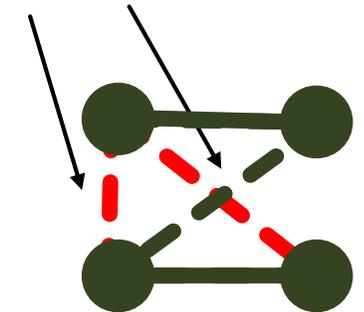
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

The weight vector (incorporates user's domain knowledge)

$$s^*(M) = f(\mathbf{s}^*(\mathbf{e})) = \max(0, \langle \mathbf{w}, \mathbf{s}^*(\mathbf{e}) \rangle)$$

A deal-breaker edge that **does not** exist (i.e., it might arrive later):

$$s_i^*(e) = -s_i(e)$$

A deal-breaker edge that **does already** exist:

$$s^*(e) = 0$$

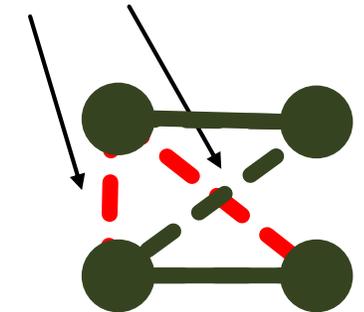
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

The weight vector (incorporates user's domain knowledge)

$$s^*(M) = f(\mathbf{s}^*(\mathbf{e})) = \max(0, \langle \mathbf{w}, \mathbf{s}^*(\mathbf{e}) \rangle)$$

A deal-breaker edge that **does not** exist (i.e., it might arrive later):

$$s_i^*(e) = -s_i(e)$$

A deal-breaker edge that **does already** exist:

$$s^*(e) = 0$$

Another formulation (perhaps more intuitive):

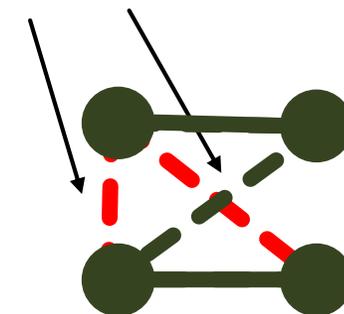
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



Motif Prediction Score Functions

What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

Incorporating deal-breaker links

The weight vector (incorporates user's domain knowledge)

$$s^*(M) = f(\mathbf{s}^*(\mathbf{e})) = \max(0, \langle \mathbf{w}, \mathbf{s}^*(\mathbf{e}) \rangle)$$

A deal-breaker edge that **does not** exist (i.e., it might arrive later):

$$s_i^*(e) = -s_i(e)$$

A deal-breaker edge that **does already** exist:

$$s^*(e) = 0$$

Another formulation (perhaps more intuitive):

$$s_{\perp}^*(M) = \prod_{e \in E_M} s(e) \cdot \prod_{e \in \bar{E}_{M,\mathcal{D}}} (1 - s(e))$$

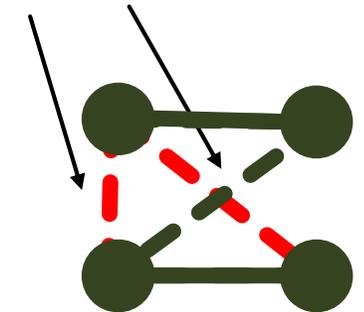
A motif: $M = (V_M, E_M)$

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



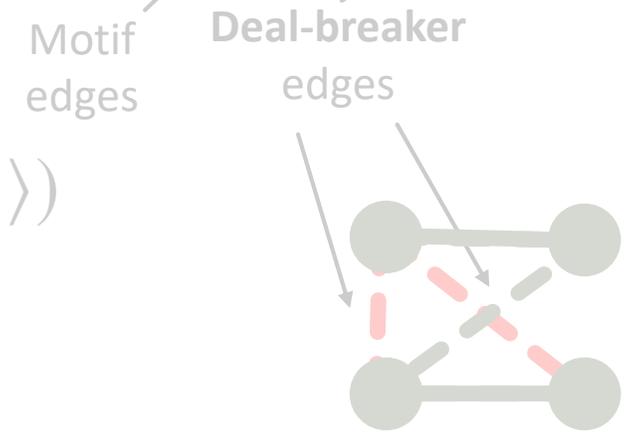
Motif Prediction Score Functions

A motif: $M = (V_M, E_M)$

Check the paper [1] for details about heuristics (based on pairwise Jaccard, Common Neighbors, and Adamic-Adar scores)

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$



$$s^*(M) = f(s^*(e)) = \max(0, \langle \bar{w}, s^*(e) \rangle)$$

A deal-breaker edge that **does not** exist (i.e., it might arrive later):

$$s_i^*(e) = -s_i(e)$$

A deal-breaker edge that **does already** exist:

$$s^*(e) = 0$$

Another formulation (perhaps more intuitive):

$$s_{\perp}^*(M) = \prod_{e \in E_M} s(e) \cdot \prod_{e \in \bar{E}_{M,\mathcal{D}}} (1 - s(e))$$

[1] M. Besta et al.: "Motif prediction with graph neural networks", KDD'22

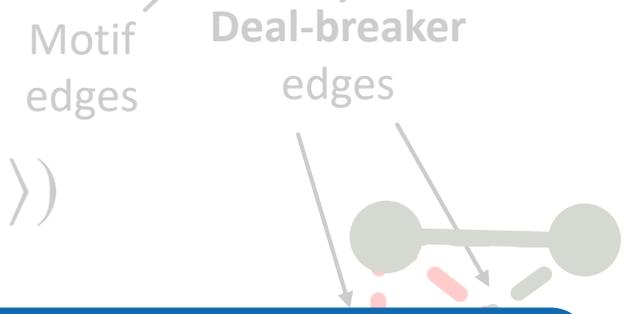
Motif Prediction Score Functions

A motif: $M = (V_M, E_M)$

Check the paper [1] for details about heuristics (based on pairwise Jaccard, Common Neighbors, and Adamic-Adar scores)

$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

$$E_M^* = E_M \cup \bar{E}_{M,\mathcal{D}}$$



$$s^*(M) = f(s^*(e)) = \max(0, \langle \mathbf{w}, s^*(e) \rangle)$$

A deal-breaker edge that does not exist (i.e., it might not exist)

$$s_i^*(e) =$$

A deal-breaker edge

These are all heuristics... but recent results for learning-enhanced link prediction [2] post a question: **can we use learning for motif prediction as well?**

Another formula

$$s_{\perp}^*(M) = \prod_{e \in E_M} s(e) \cdot \prod_{e \in \bar{E}_{M,\mathcal{D}}} (1 - s(e))$$

[1] M. Besta et al.: "Motif prediction with graph neural networks", KDD'22

[2] M. Zhang et al.: "Link Prediction Based on Graph Neural Networks", NeurIPS'18

The animation borrowed
from T. Hoefler

The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

The animation borrowed
from T. Hoefler

The graph structure may be arbitrary, maybe
one could arrive at better heuristics by learning?

How does deep learning work?



The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

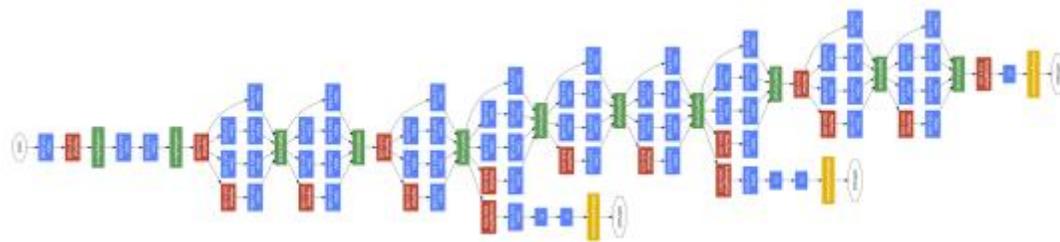
Samples



The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

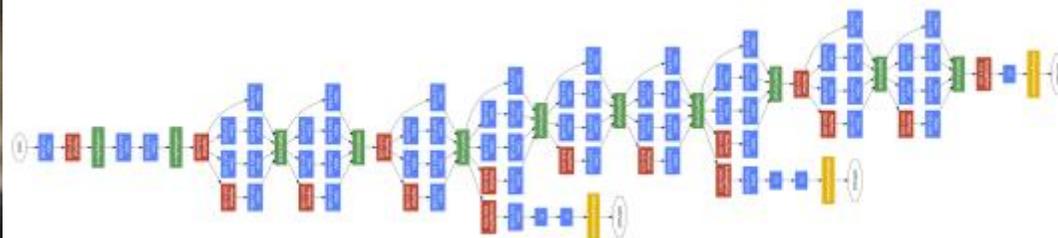
Samples



The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

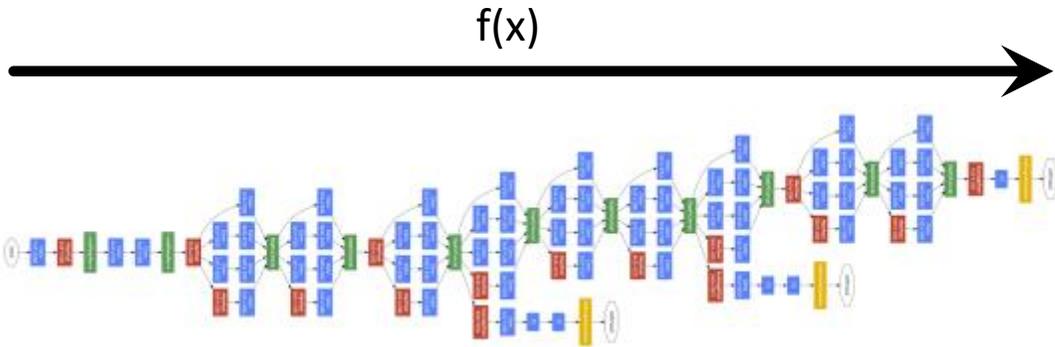
Samples



The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

Samples

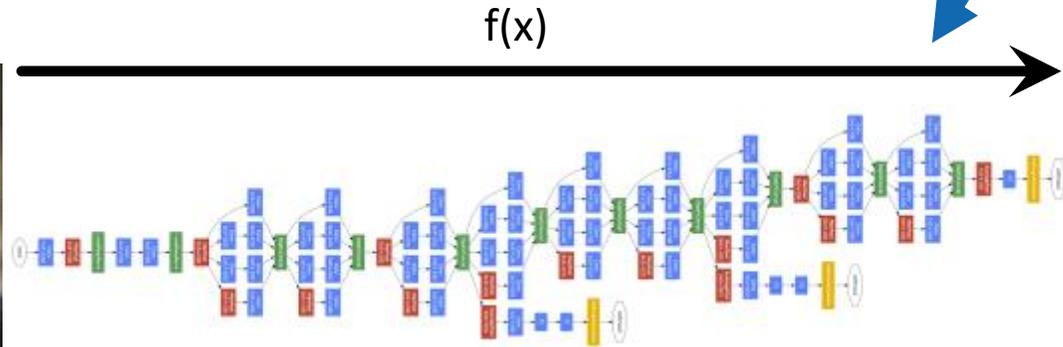


The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

Samples

Learning a heuristic



The animation borrowed from T. Hoefler

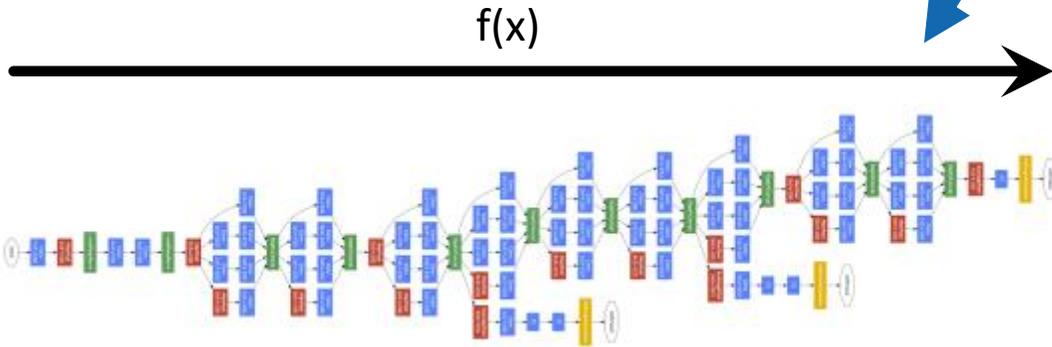
The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

Samples



Learning a heuristic



Cat	0.54
Dog	0.28
Airplane	0.07
Horse	0.04
Bicycle	0.02
Truck	0.02

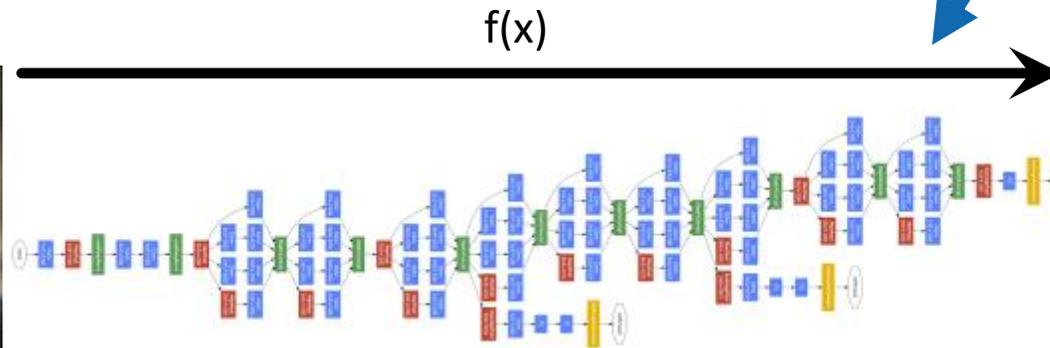
The animation borrowed from T. Hoefler

The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

Samples

Learning a heuristic



Cat	0.54
Dog	0.28
Airplane	0.07
Horse	0.04
Bicycle	0.02
Truck	0.02

Cat	1.00
Dog	0.00
Airplane	0.00
Horse	0.00
Bicycle	0.00
Truck	0.00

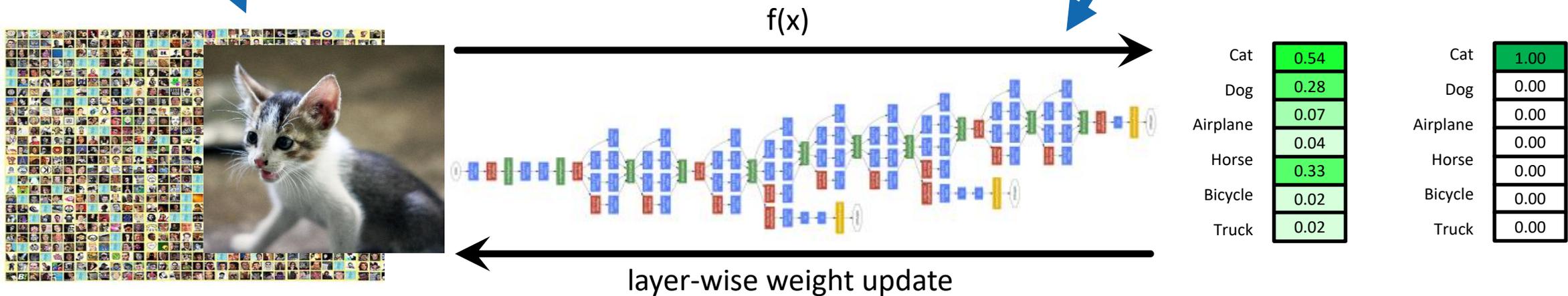
The animation borrowed from T. Hoefler

The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

Samples

Learning a heuristic



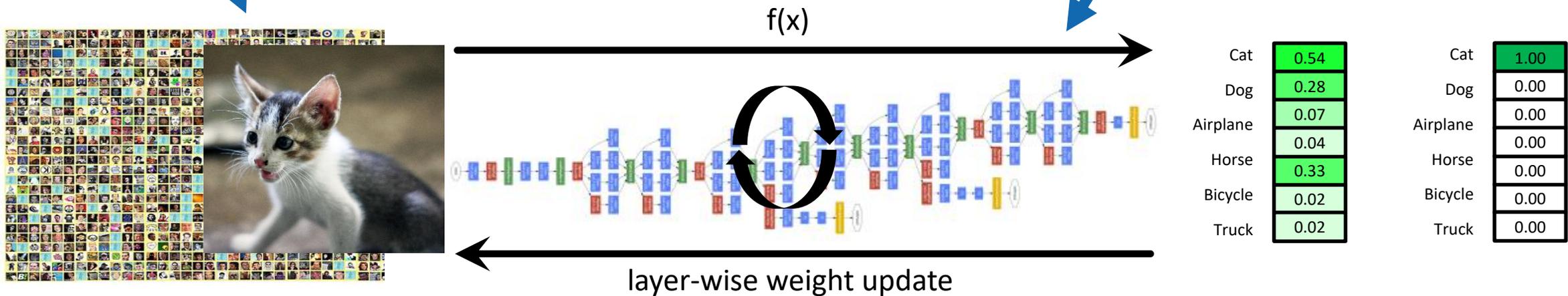
The animation borrowed from T. Hoefler

The graph structure may be arbitrary, maybe one could arrive at better heuristics by learning?

How does deep learning work?

Samples

Learning a heuristic



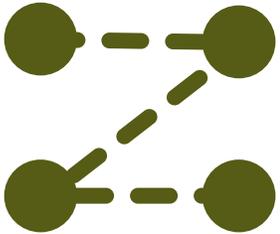
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

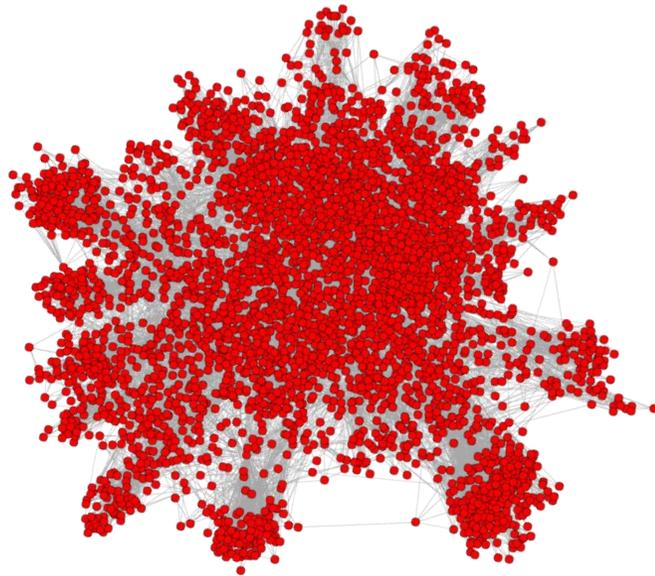
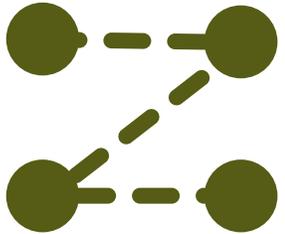
Goal: predict
this motif M:



Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

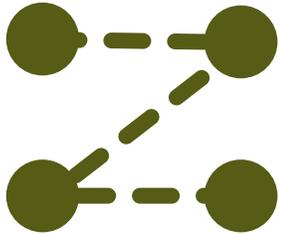
Goal: predict
this motif M :



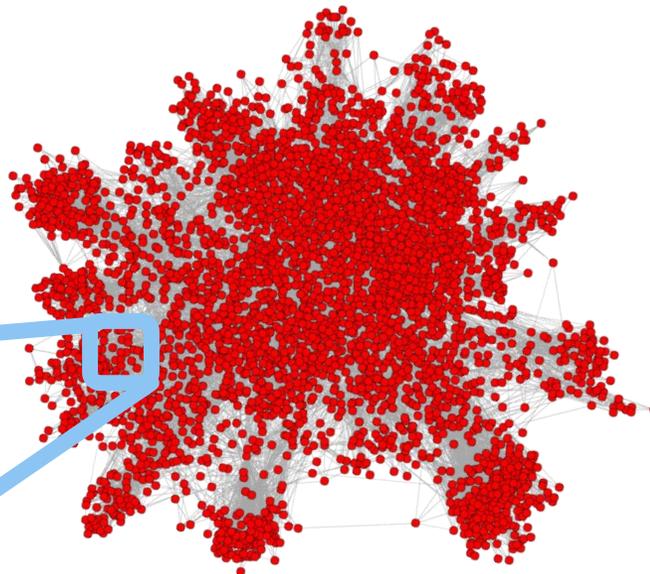
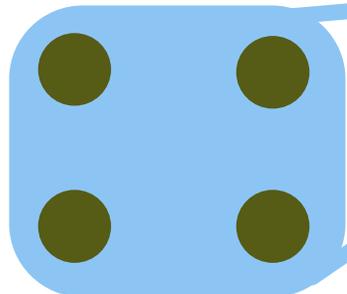
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

Goal: predict this motif M :



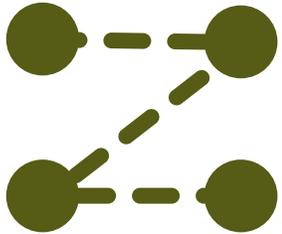
A given fixed set of vertices:



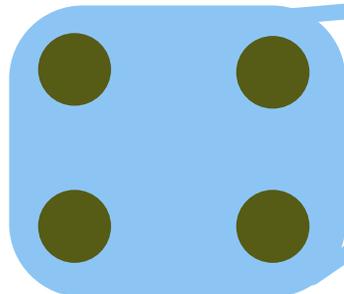
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

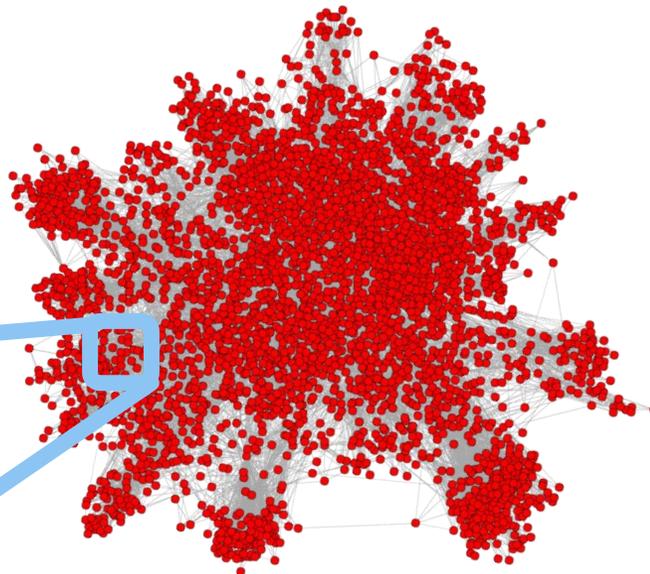
Goal: predict this motif M :



A given fixed set of vertices:



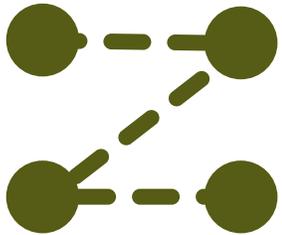
Positive samples



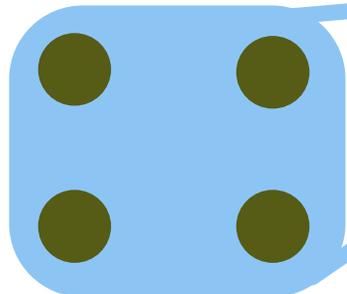
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

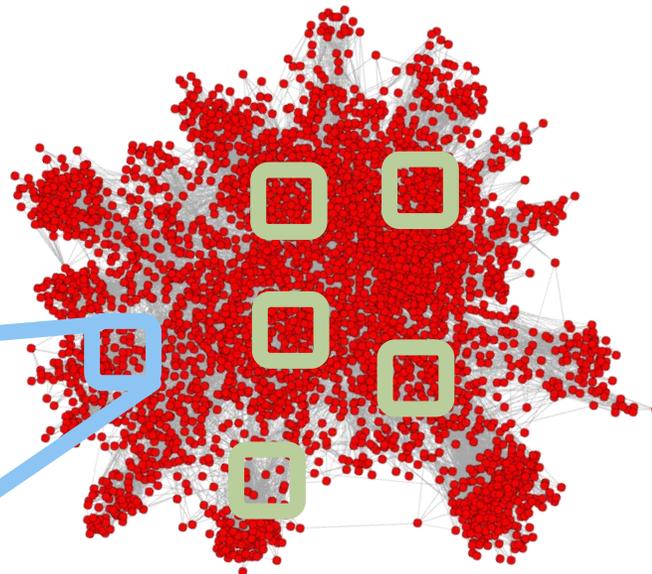
Goal: predict this motif M:



A given fixed set of vertices:



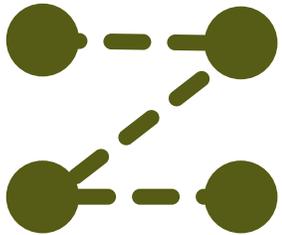
Positive samples



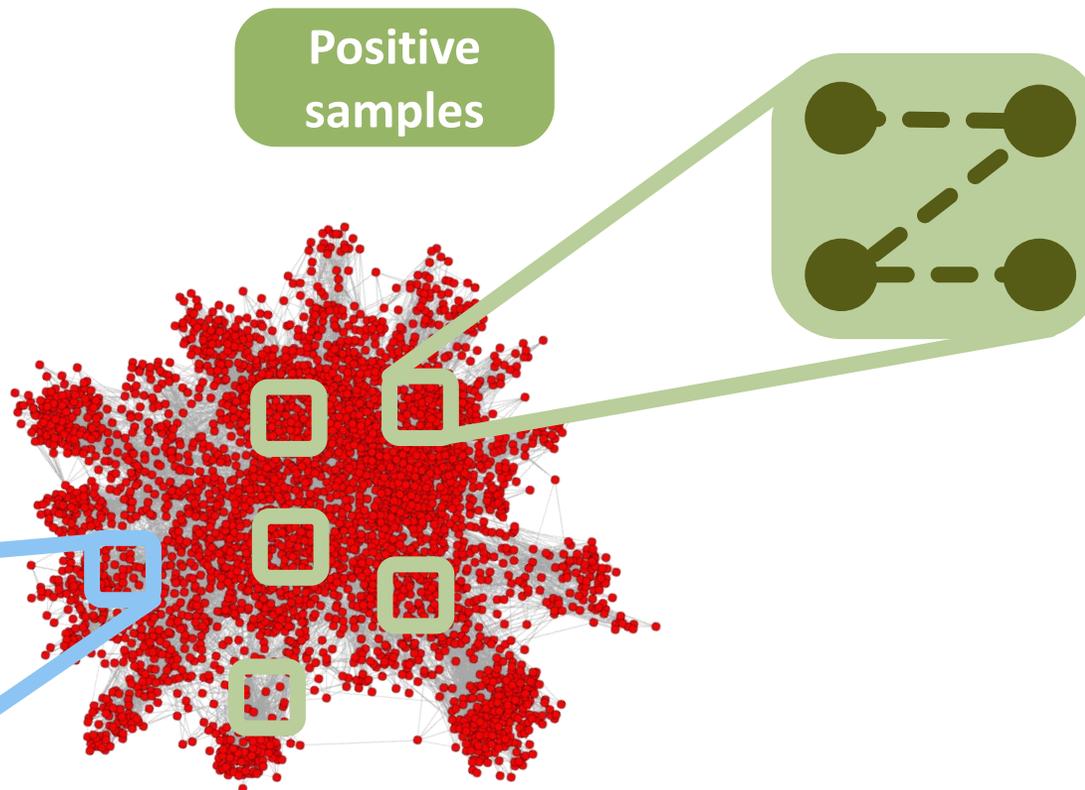
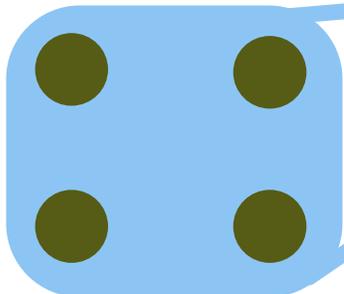
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

Goal: predict this motif M:



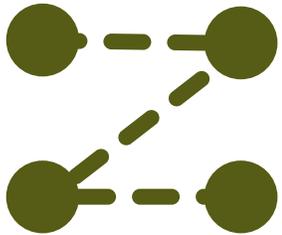
A given fixed set of vertices:



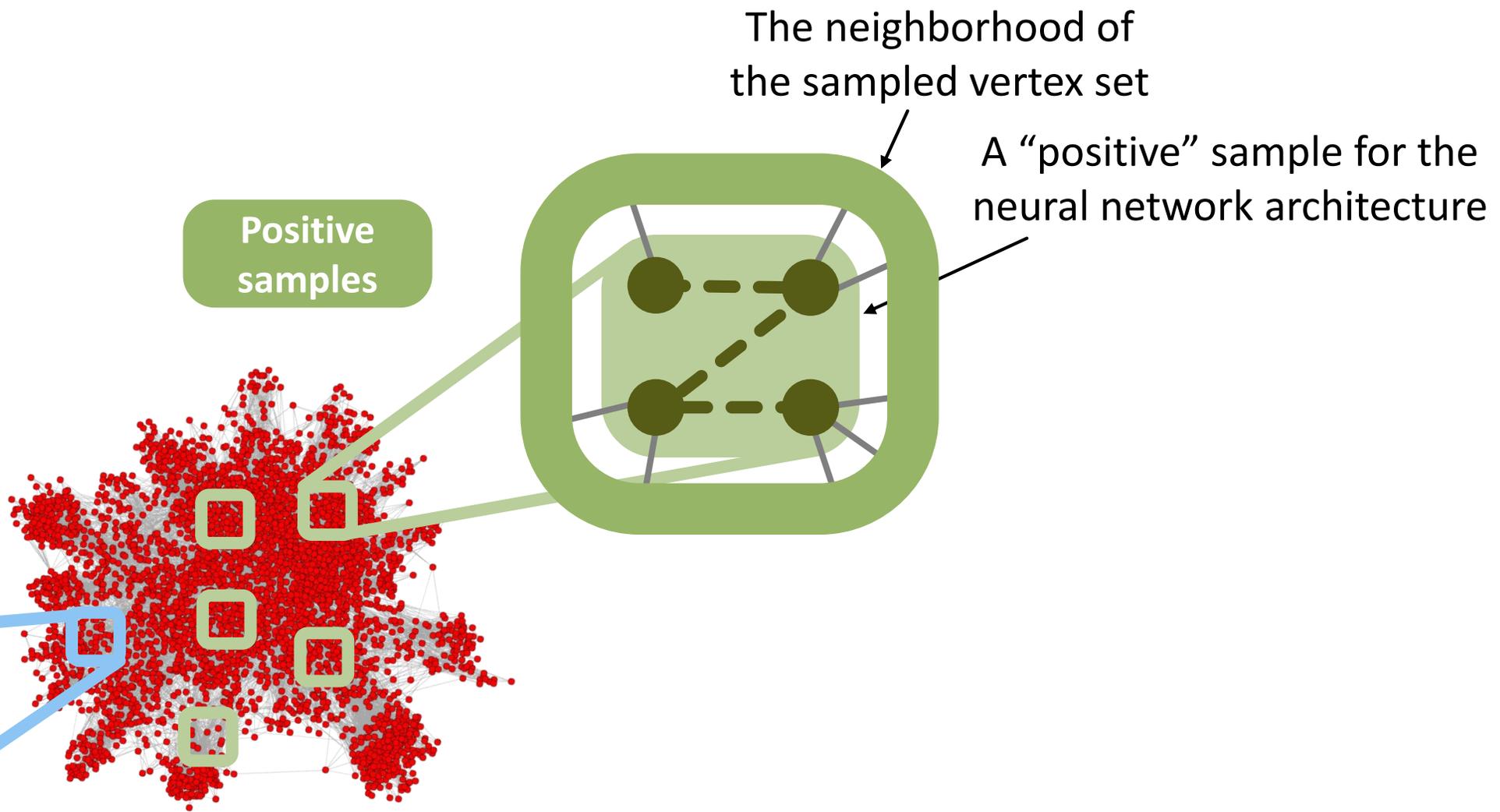
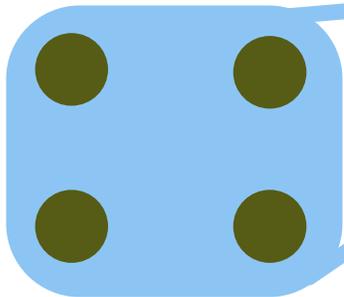
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

Goal: predict this motif M:



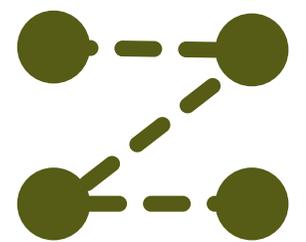
A given fixed set of vertices:



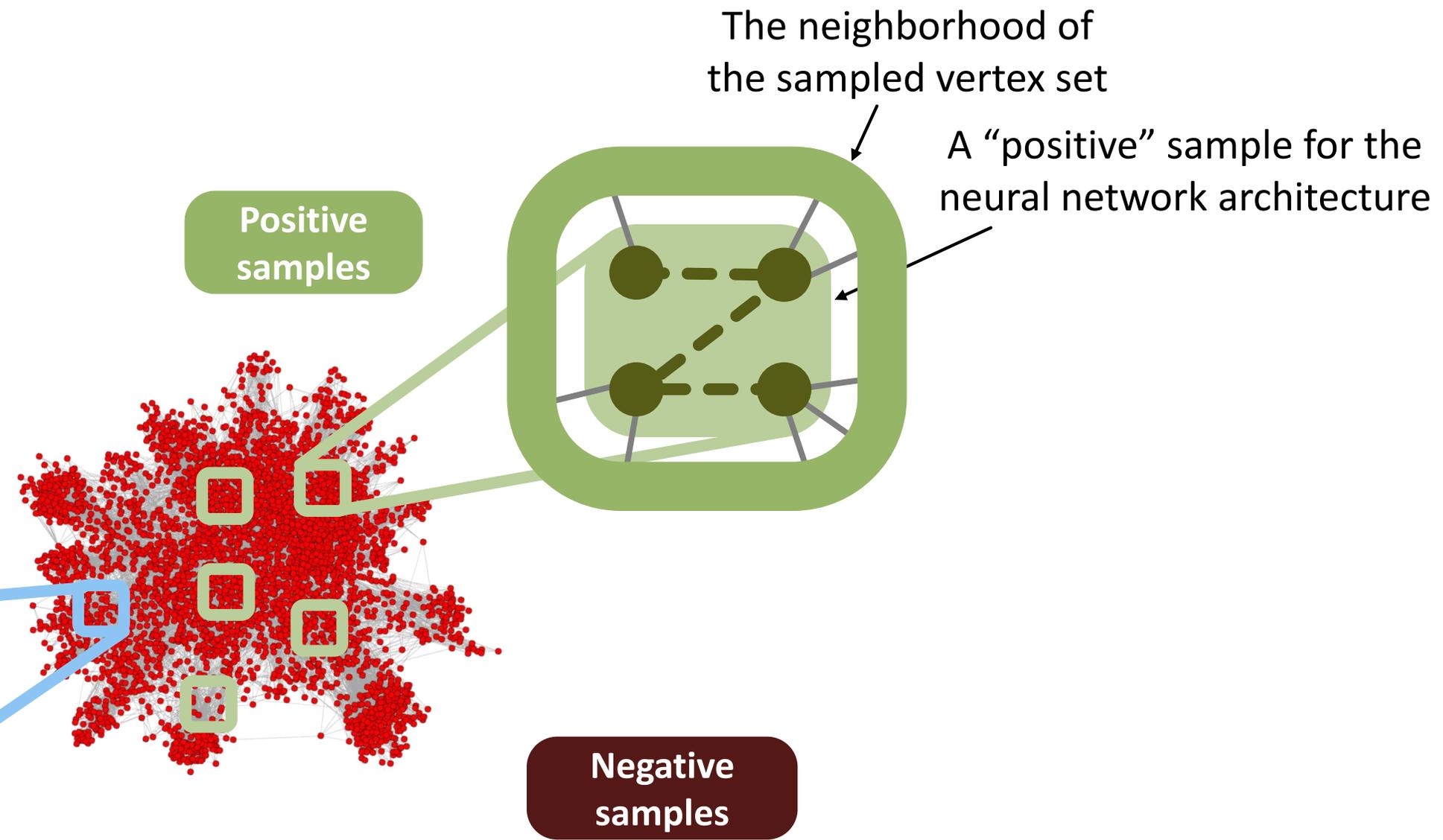
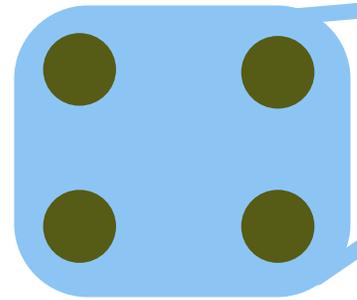
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

Goal: predict this motif M:



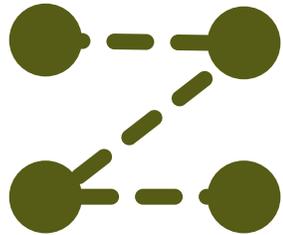
A given fixed set of vertices:



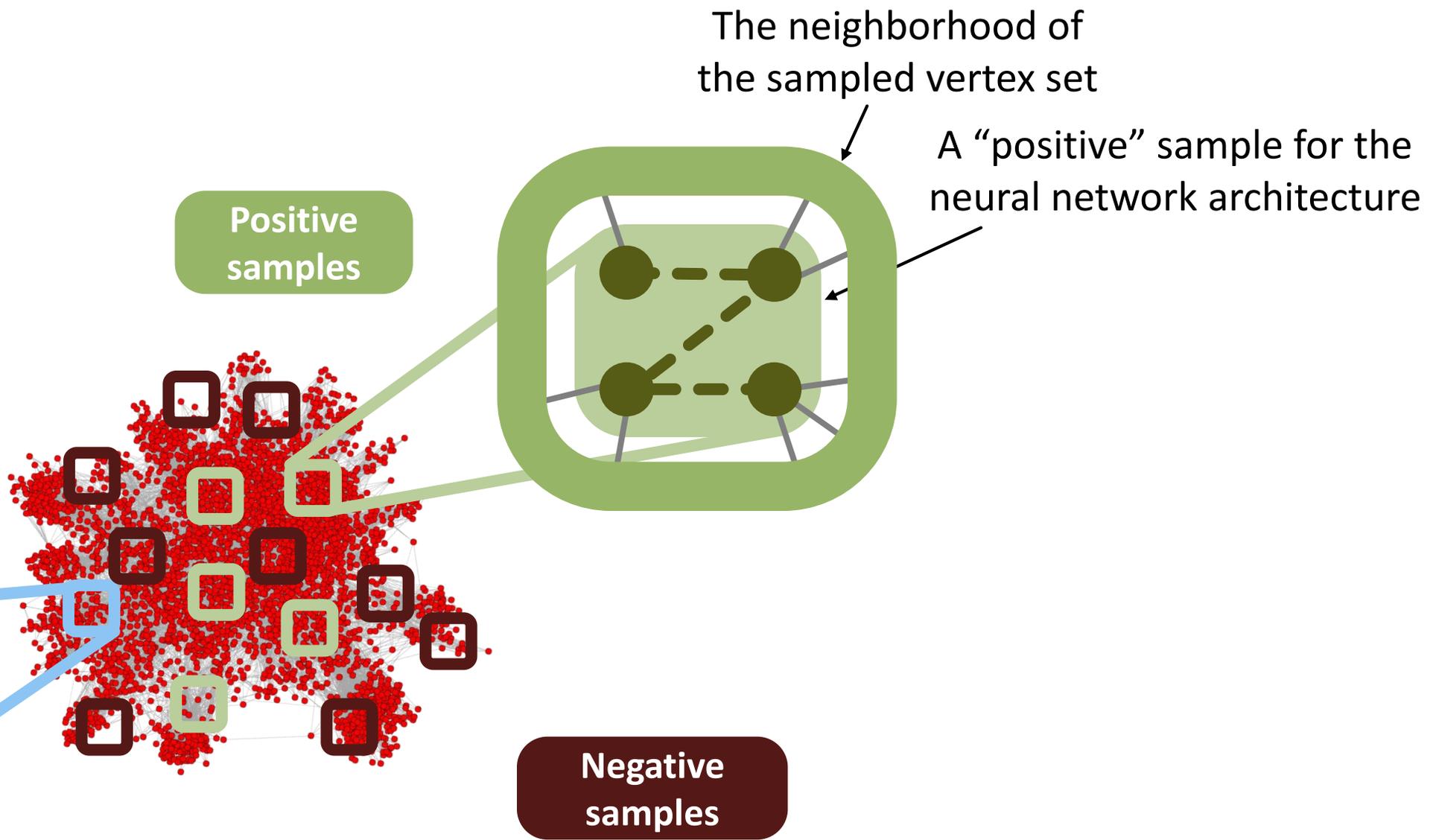
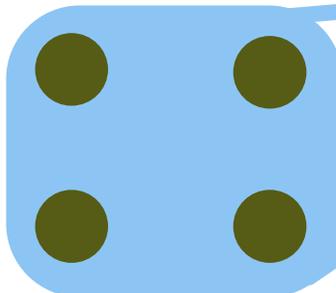
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

Goal: predict this motif M:



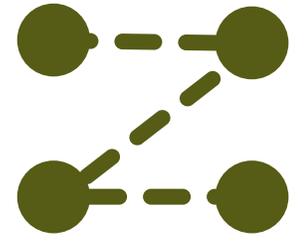
A given fixed set of vertices:



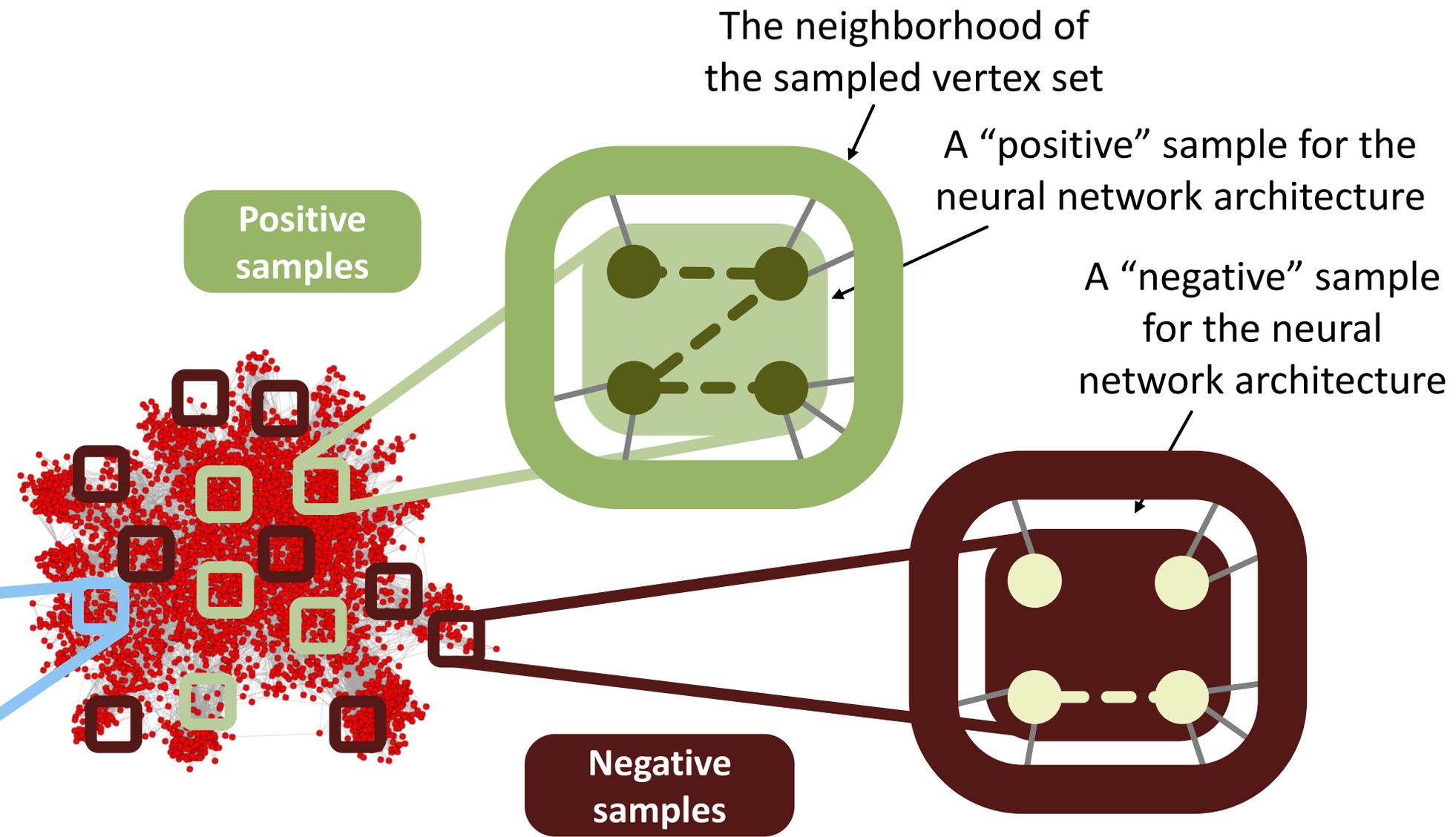
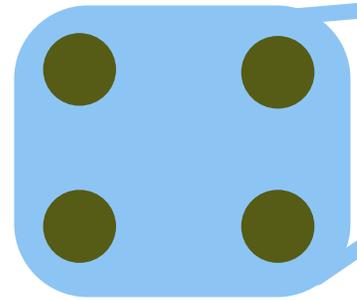
Motif Prediction: Deep Learning Formulation

A motif: $M = (V_M, E_M)$

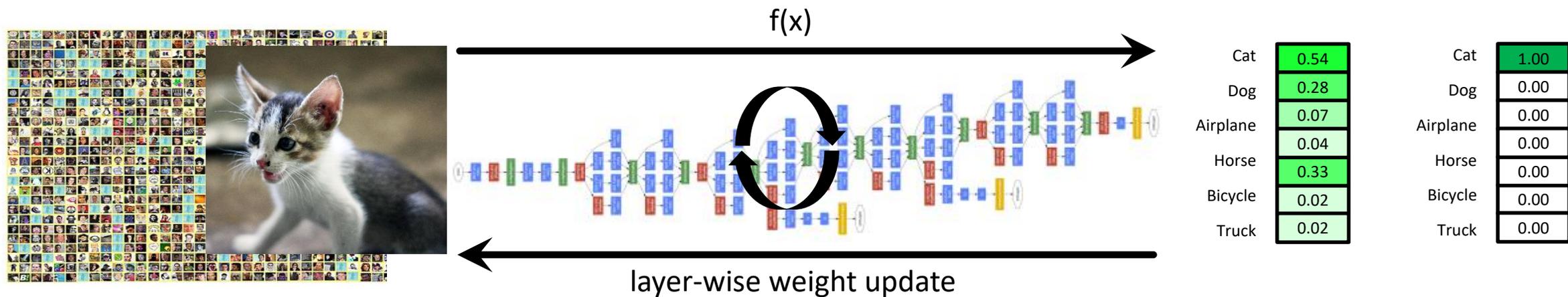
Goal: predict this motif M:



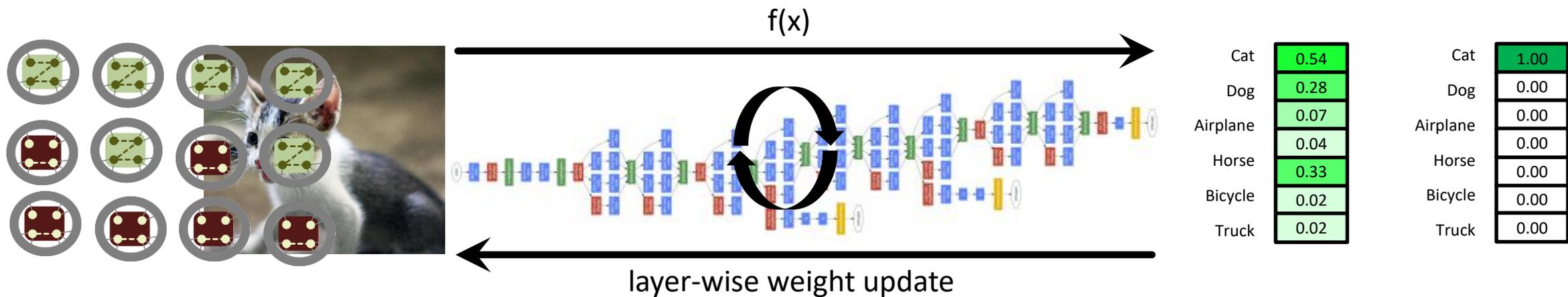
A given fixed set of vertices:



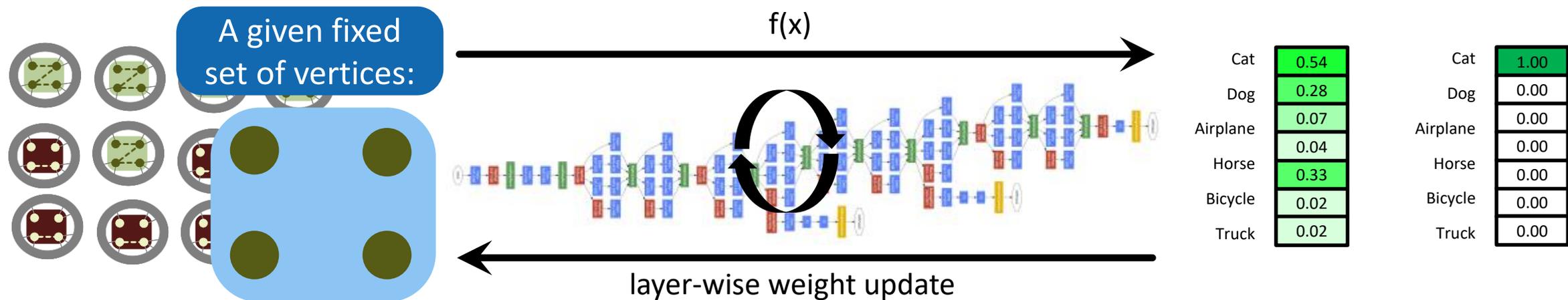
Motif Prediction: Deep Learning Formulation



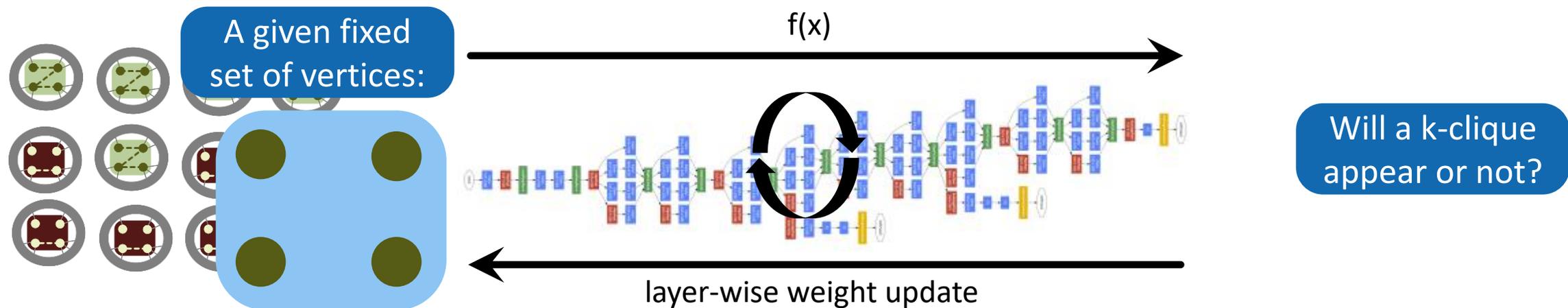
Motif Prediction: Deep Learning Formulation



Motif Prediction: Deep Learning Formulation

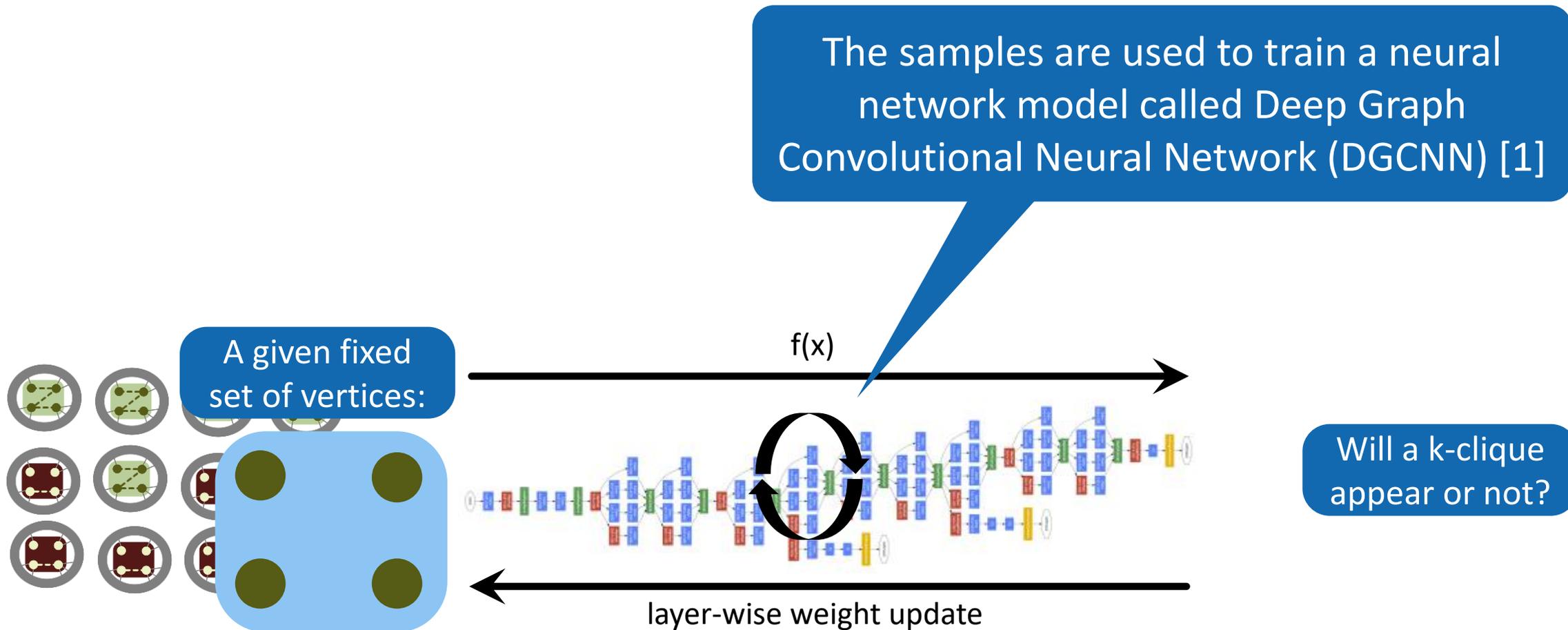


Motif Prediction: Deep Learning Formulation



Motif Prediction: Deep Learning Formulation

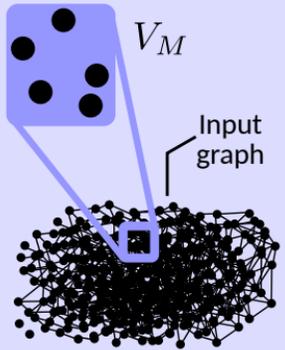
[1] M. Zhan et al. 2018. An end-to-end deep learning architecture for graph classification. AAAI Conference on Artificial Intelligence



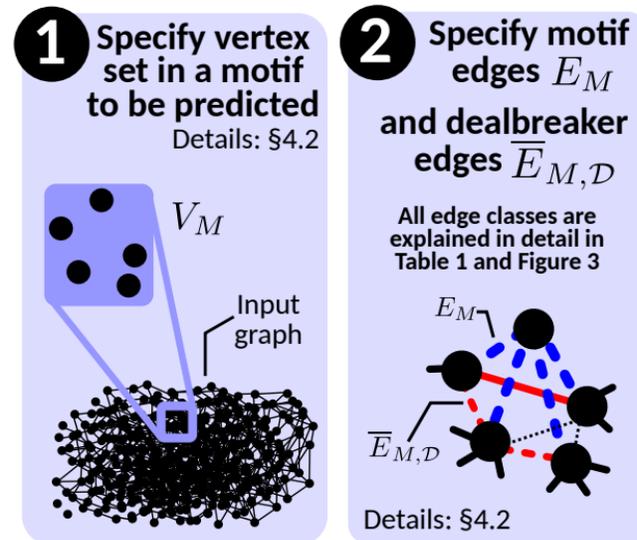
SEAM Architecture (learning from Subgraphs, Embeddings and Atttributes for Motif prediction)

SEAM Architecture (learning from Subgraphs, EMBEDdings and ATTributes for Motif prediction)

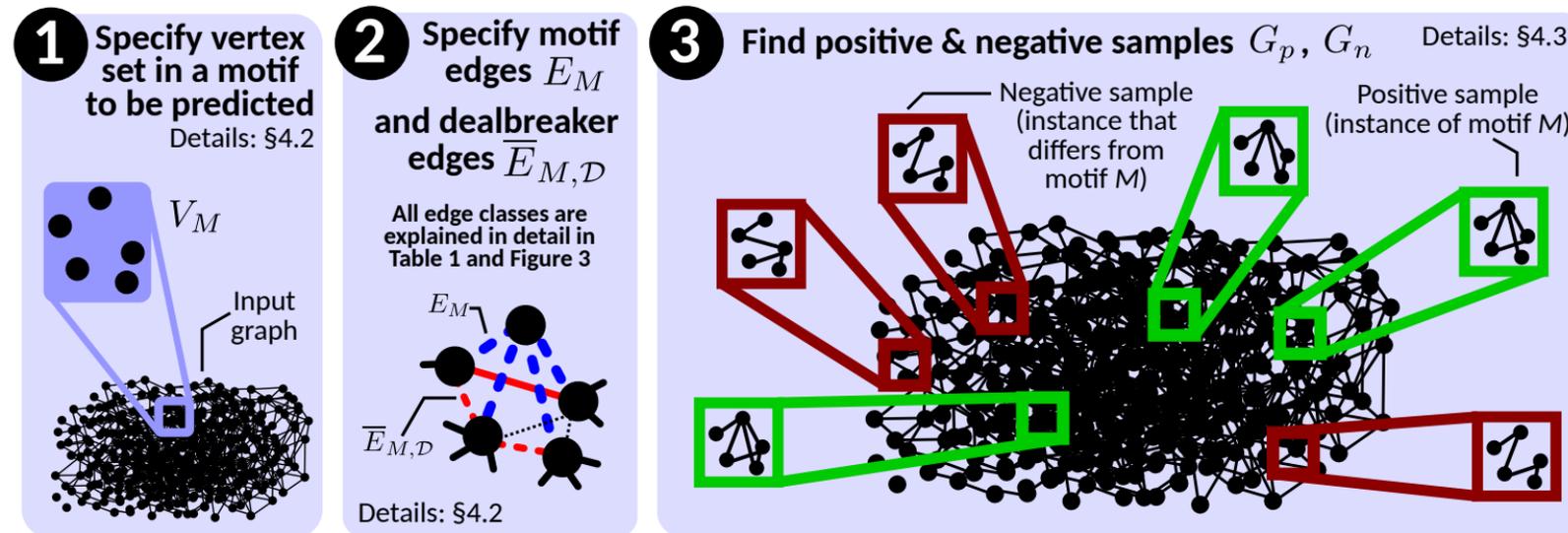
1 Specify vertex set in a motif to be predicted
Details: §4.2



SEAM Architecture (learning from Subgraphs, Embeddings and Atributes for Motif prediction)



SEAM Architecture (learning from Subgraphs, Embeddings and Atributes for Motif prediction)



SEAM Architecture (learning from Subgraphs, Embeddings and Atributes for Motif prediction)

1 Specify vertex set in a motif to be predicted
Details: §4.2

V_M
Input graph

2 Specify motif edges E_M and dealbreaker edges $\bar{E}_{M,D}$
All edge classes are explained in detail in Table 1 and Figure 3
Details: §4.2

E_M
 $\bar{E}_{M,D}$

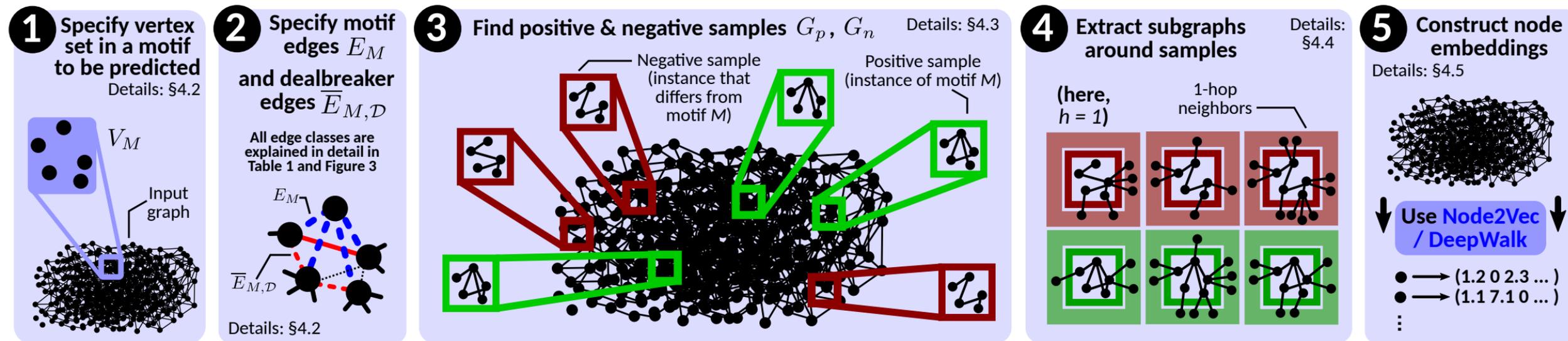
3 Find positive & negative samples G_p, G_n Details: §4.3

Negative sample (instance that differs from motif M)
Positive sample (instance of motif M)

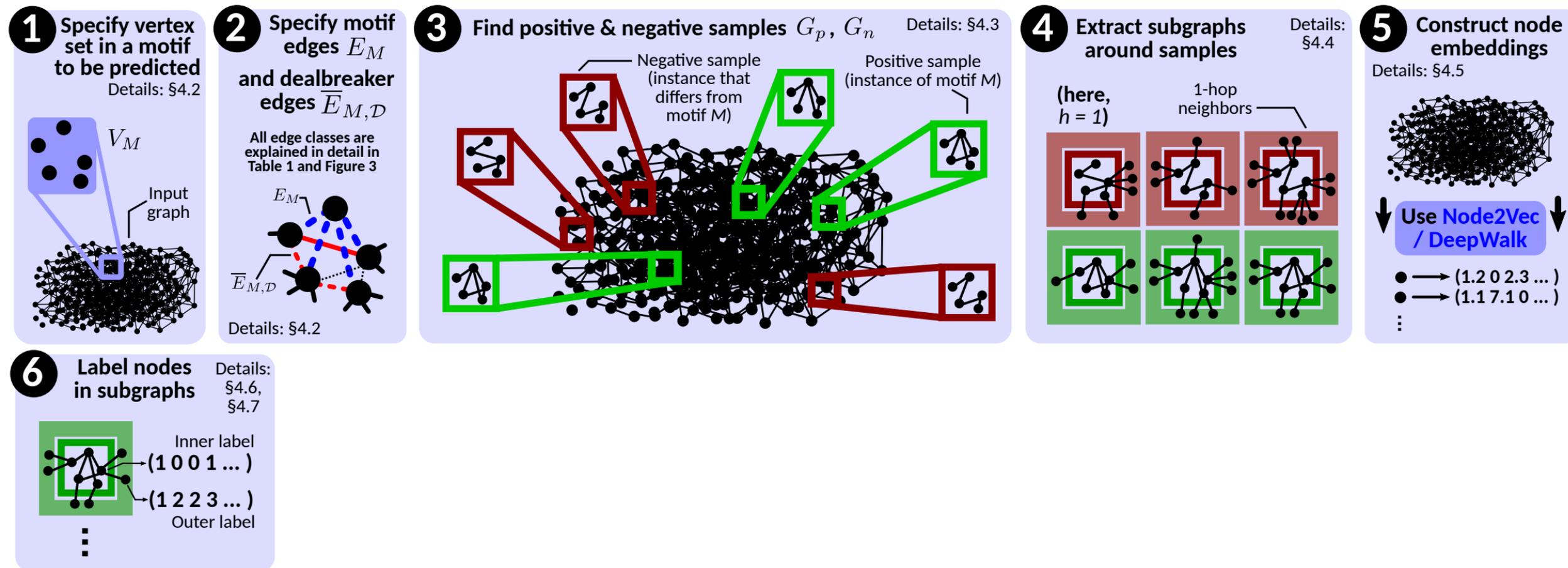
4 Extract subgraphs around samples Details: §4.4

(here, $h = 1$)
1-hop neighbors

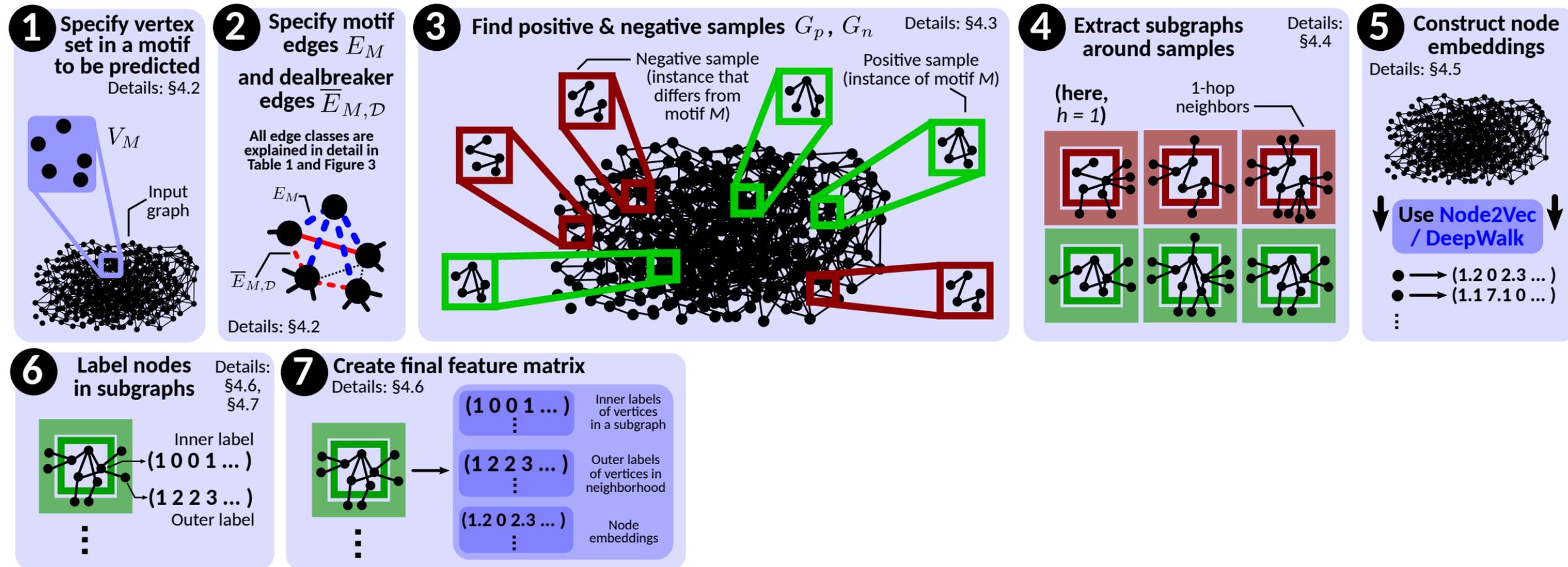
SEAM Architecture (learning from Subgraphs, EMBEDdings and ATTributes for Motif prediction)



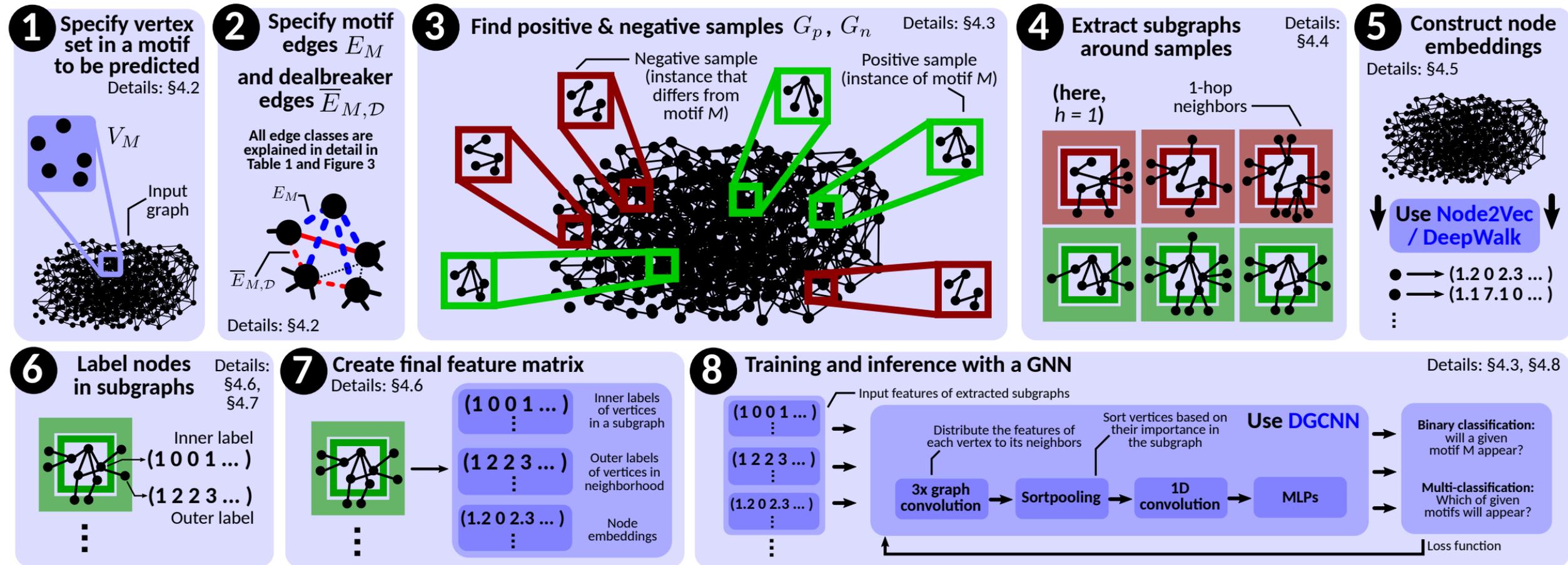
SEAM Architecture (learning from Subgraphs, Embeddings and Atributes for Motif prediction)



SEAM Architecture (learning from Subgraphs, Embeddings and Atributes for Motif prediction)



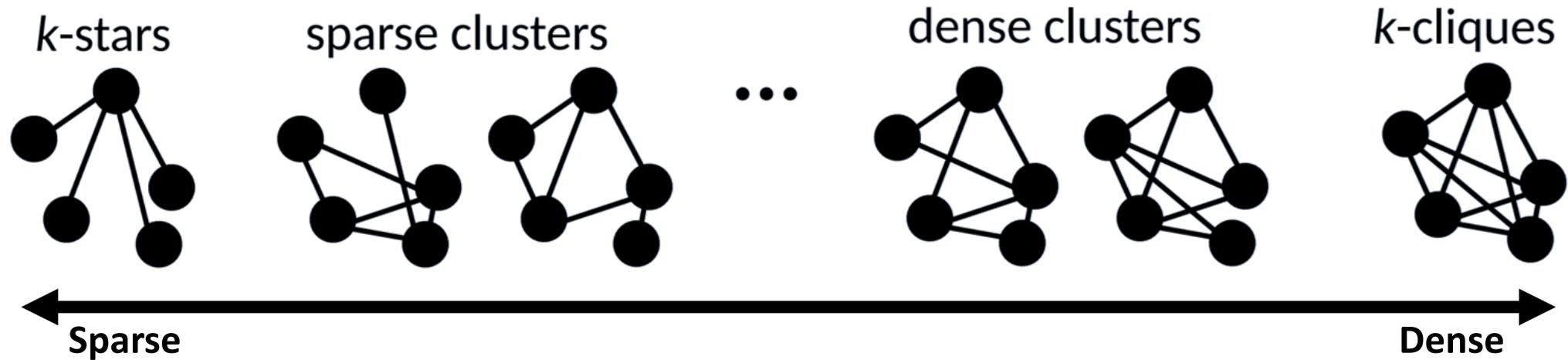
SEAM Architecture (learning from Subgraphs, Embeddings and Atributes for Motif prediction)



Evaluation: considered motifs & scenarios



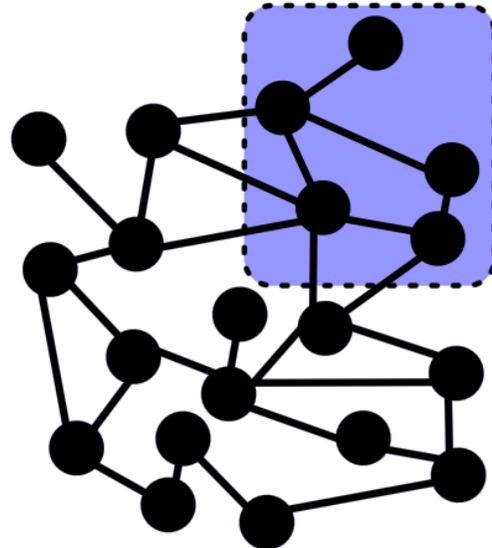
Evaluation: considered motifs & scenarios



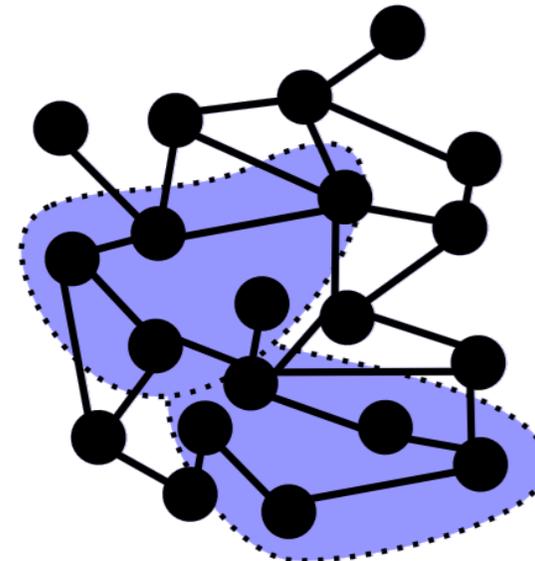
Evaluation: considered motifs & scenarios (transductive, inductive)

Evaluation: considered motifs & scenarios (transductive, inductive)

Pick motif X in graph A

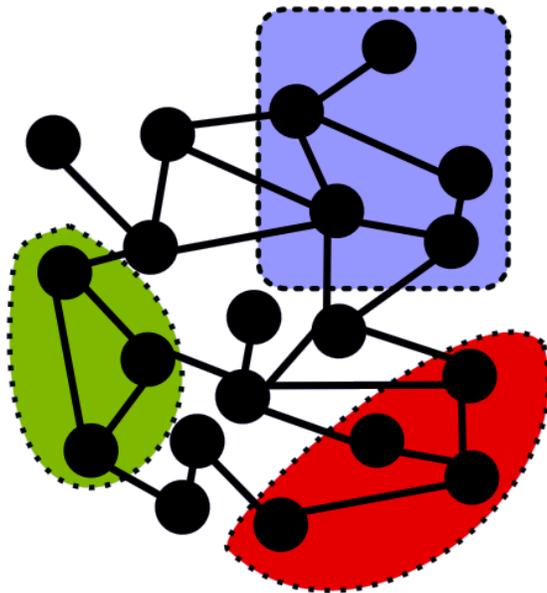


Train for predicting new instances of motif X in graph A

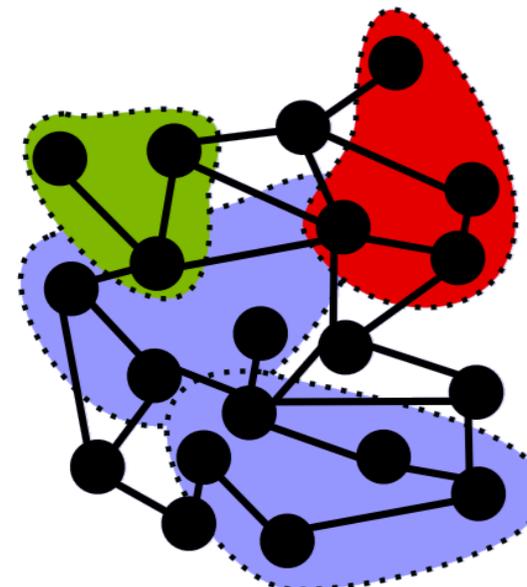


Evaluation: considered motifs & scenarios (transductive, inductive)

Pick motifs
 X, Y, \dots
in graph A

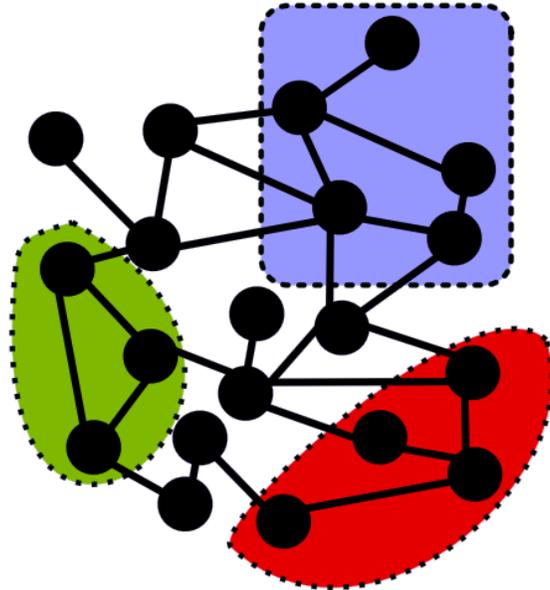


Train for
predicting
new instances
of motifs X, Y, \dots
in graph A

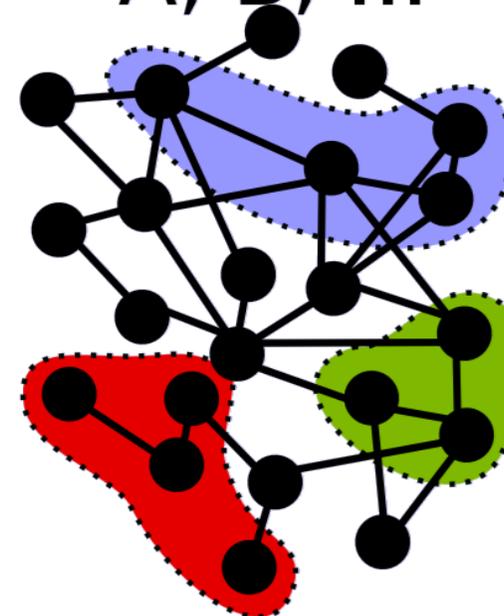


Evaluation: considered motifs & scenarios (transductive, inductive)

Pick motifs
X, Y, ...
in graphs
A, B, ...



Train for
predicting
new instances
of motifs X, Y, ...
in graphs
A, B, ...



Evaluation

CN (Mul)	49.99 ± 0.45	49.78 ± 0.39	50.19 ± 0.47	50.13 ± 0.65	50.37 ± 0.79	51.55 ± 0.32	52.93 ± 0.61	55.42 ± 0.58	54.81 ± 0.58
CN (Min)	49.98 ± 0.33	49.72 ± 0.49	50.26 ± 0.59	50.35 ± 0.27	50.48 ± 0.32	51.77 ± 0.40	52.99 ± 0.75	54.75 ± 0.48	54.60 ± 0.85
CN (Avg)	49.76 ± 0.32	49.50 ± 0.64	50.18 ± 0.64	50.28 ± 0.51	50.91 ± 0.35	51.70 ± 0.59	53.32 ± 0.35	54.93 ± 0.77	54.20 ± 0.68
AA (Mul)	63.05 ± 0.71	62.09 ± 0.57	60.67 ± 0.95	54.95 ± 0.92	51.25 ± 0.63	51.40 ± 0.68	53.92 ± 0.52	55.15 ± 0.77	54.93 ± 0.61
AA (Min)	63.34 ± 0.68	62.81 ± 0.80	61.59 ± 0.94	54.81 ± 0.77	51.26 ± 0.38	51.60 ± 0.68	54.15 ± 0.89	54.59 ± 0.65	54.94 ± 0.27
AA (Avg)	63.96 ± 0.68	63.66 ± 0.48	62.71 ± 0.52	55.78 ± 0.74	51.28 ± 0.55	51.79 ± 0.62	54.52 ± 0.57	55.20 ± 0.53	54.55 ± 0.39
Jaccard (Mul)	67.17 ± 0.92	62.01 ± 0.72	59.71 ± 0.93	69.62 ± 1.09	57.60 ± 0.73	52.75 ± 0.97	51.75 ± 1.10	51.68 ± 0.85	50.93 ± 0.64
Jaccard (Min)	69.20 ± 0.80	67.11 ± 0.46	65.24 ± 0.80	73.88 ± 0.88	63.36 ± 1.17	56.50 ± 0.94	52.30 ± 0.54	51.86 ± 0.77	50.87 ± 0.53
Jaccard (Avg)	70.12 ± 0.78	68.59 ± 0.71	68.69 ± 0.77	75.35 ± 0.60	67.93 ± 0.87	61.22 ± 1.11	51.76 ± 0.91	49.74 ± 0.75	47.66 ± 0.58
SEAL (Mul)	76.68 ± 0.61	74.00 ± 0.50	71.80 ± 0.95	76.25 ± 1.90	63.66 ± 4.01	59.48 ± 4.87	68.53 ± 0.88	67.49 ± 1.27	67.88 ± 1.43
SEAL (Min)	77.15 ± 0.43	74.62 ± 0.55	73.11 ± 0.99	78.00 ± 1.49	69.70 ± 3.56	64.49 ± 5.47	66.40 ± 1.44	62.94 ± 1.98	62.88 ± 3.57
SEAL (Avg)	77.91 ± 0.91	75.98 ± 0.99	75.71 ± 0.66	77.50 ± 2.35	72.68 ± 3.21	66.95 ± 6.79	66.05 ± 0.78	65.14 ± 0.89	66.99 ± 1.32
SEAM, no embedding	86.24 ± 0.99	85.57 ± 0.94	88.61 ± 0.71	91.20 ± 1.03	96.16 ± 0.55	98.40 ± 0.22	83.39 ± 0.94	86.12 ± 0.66	87.86 ± 1.06
SEAM	90.78 ± 1.30	90.00 ± 1.84	91.53 ± 1.53	93.06 ± 0.61	97.26 ± 0.23	98.90 ± 0.18	83.81 ± 0.53	87.56 ± 0.79	88.59 ± 1.51
	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star

SEAM: learning from Subgraphs, Embeddings and Attributes for Motif prediction

Evaluation

CN (Mul)	49.99 ± 0.45	49.78 ± 0.39	50.19 ± 0.47	50.13 ± 0.65	50.37 ± 0.79	51.55 ± 0.32	52.93 ± 0.61	55.42 ± 0.58	54.81 ± 0.58
CN (Min)	49.98 ± 0.33	49.72 ± 0.49	50.26 ± 0.59	50.35 ± 0.27	50.48 ± 0.32	51.77 ± 0.40	52.99 ± 0.75	54.75 ± 0.48	54.60 ± 0.85
CN (Avg)	49.76 ± 0.32	49.50 ± 0.64	50.18 ± 0.64	50.28 ± 0.51	50.91 ± 0.35	51.70 ± 0.59	53.32 ± 0.35	54.93 ± 0.77	54.20 ± 0.68
AA (Mul)	63.05 ± 0.71	62.09 ± 0.57	60.67 ± 0.95	54.95 ± 0.92	51.25 ± 0.63	51.40 ± 0.68	53.92 ± 0.52	55.15 ± 0.77	54.93 ± 0.61
AA (Min)	63.34 ± 0.68	62.81 ± 0.80	61.59 ± 0.94	54.81 ± 0.77	51.26 ± 0.38	51.60 ± 0.68	54.15 ± 0.89	54.59 ± 0.65	54.94 ± 0.27
AA (Avg)	63.96 ± 0.68	63.66 ± 0.48	62.71 ± 0.52	55.78 ± 0.74	51.28 ± 0.55	51.79 ± 0.62	54.52 ± 0.57	55.20 ± 0.53	54.55 ± 0.39
Jaccard (Mul)	67.17 ± 0.92	62.01 ± 0.72	59.71 ± 0.93	69.62 ± 1.09	57.60 ± 0.73	52.75 ± 0.97	51.75 ± 1.10	51.68 ± 0.85	50.93 ± 0.64
Jaccard (Min)	69.20 ± 0.80	67.11 ± 0.46	65.24 ± 0.80	73.88 ± 0.88	63.36 ± 1.17	56.50 ± 0.94	52.30 ± 0.54	51.86 ± 0.77	50.87 ± 0.53
Jaccard (Avg)	70.12 ± 0.78	68.59 ± 0.71	68.69 ± 0.77	75.35 ± 0.60	67.93 ± 0.87	61.22 ± 1.11	51.76 ± 0.91	49.74 ± 0.75	47.66 ± 0.58
SEAL (Mul)	76.68 ± 0.61	74.00 ± 0.50	71.80 ± 0.95	76.25 ± 1.90	63.66 ± 4.01	59.48 ± 4.87	68.53 ± 0.88	67.49 ± 1.27	67.88 ± 1.43
SEAL (Min)	77.15 ± 0.43	74.62 ± 0.55	73.11 ± 0.99	78.00 ± 1.49	69.70 ± 3.56	64.49 ± 5.47	66.40 ± 1.44	62.94 ± 1.98	62.88 ± 3.57
SEAL (Avg)	77.91 ± 0.91	75.98 ± 0.99	75.71 ± 0.66	77.50 ± 2.35	72.68 ± 3.21	66.95 ± 6.79	66.05 ± 0.78	65.14 ± 0.89	66.99 ± 1.32
SEAM, no embedding	86.24 ± 0.99	85.57 ± 0.94	88.61 ± 0.71	91.20 ± 1.03	96.16 ± 0.55	98.40 ± 0.22	83.39 ± 0.94	86.12 ± 0.66	87.86 ± 1.06
SEAM	90.78 ± 1.30	90.00 ± 1.84	91.53 ± 1.53	93.06 ± 0.61	97.26 ± 0.23	98.90 ± 0.18	83.81 ± 0.53	87.56 ± 0.79	88.59 ± 1.51
	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star

SEAM: learning from Subgraphs, Embeddings and Attributes for Motif prediction

„db” – with deal-breaker edges

Evaluation

CN: Common Neighbors, **AA**: Adamic-Adar, **SEAL [2]**: Link Prediction using GNNs
Mul, Min, Avg: different variants (i.e., how link scores are aggregated to form motif scores)

[2] M. Zhang and Y. Chen. 2018. Link prediction based on graph neural networks. NeurIPS'18

CN (Mul)	49.99 ± 0.45	49.78 ± 0.39	50.19 ± 0.47	50.13 ± 0.65	50.37 ± 0.79	51.55 ± 0.32	52.93 ± 0.61	55.42 ± 0.58	54.81 ± 0.58
CN (Min)	49.98 ± 0.33	49.72 ± 0.49	50.26 ± 0.59	50.35 ± 0.27	50.48 ± 0.32	51.77 ± 0.40	52.99 ± 0.75	54.75 ± 0.48	54.60 ± 0.85
CN (Avg)	49.76 ± 0.32	49.50 ± 0.64	50.18 ± 0.64	50.28 ± 0.51	50.91 ± 0.35	51.70 ± 0.59	53.32 ± 0.35	54.93 ± 0.77	54.20 ± 0.68
AA (Mul)	63.05 ± 0.71	62.09 ± 0.57	60.67 ± 0.95	54.95 ± 0.92	51.25 ± 0.63	51.40 ± 0.68	53.92 ± 0.52	55.15 ± 0.77	54.93 ± 0.61
AA (Min)	63.34 ± 0.68	62.81 ± 0.80	61.59 ± 0.94	54.81 ± 0.77	51.26 ± 0.38	51.60 ± 0.68	54.15 ± 0.89	54.59 ± 0.65	54.94 ± 0.27
AA (Avg)	63.96 ± 0.68	63.66 ± 0.48	62.71 ± 0.52	55.78 ± 0.74	51.28 ± 0.55	51.79 ± 0.62	54.52 ± 0.57	55.20 ± 0.53	54.55 ± 0.39
Jaccard (Mul)	67.17 ± 0.92	62.01 ± 0.72	59.71 ± 0.93	69.62 ± 1.09	57.60 ± 0.73	52.75 ± 0.97	51.75 ± 1.10	51.68 ± 0.85	50.93 ± 0.64
Jaccard (Min)	69.20 ± 0.80	67.11 ± 0.46	65.24 ± 0.80	73.88 ± 0.88	63.36 ± 1.17	56.50 ± 0.94	52.30 ± 0.54	51.86 ± 0.77	50.87 ± 0.53
Jaccard (Avg)	70.12 ± 0.78	68.59 ± 0.71	68.69 ± 0.77	75.35 ± 0.60	67.93 ± 0.87	61.22 ± 1.11	51.76 ± 0.91	49.74 ± 0.75	47.66 ± 0.58
SEAL (Mul)	76.68 ± 0.61	74.00 ± 0.50	71.80 ± 0.95	76.25 ± 1.90	63.66 ± 4.01	59.48 ± 4.87	68.53 ± 0.88	67.49 ± 1.27	67.88 ± 1.43
SEAL (Min)	77.15 ± 0.43	74.62 ± 0.55	73.11 ± 0.99	78.00 ± 1.49	69.70 ± 3.56	64.49 ± 5.47	66.40 ± 1.44	62.94 ± 1.98	62.88 ± 3.57
SEAL (Avg)	77.91 ± 0.91	75.98 ± 0.99	75.71 ± 0.66	77.50 ± 2.35	72.68 ± 3.21	66.95 ± 6.79	66.05 ± 0.78	65.14 ± 0.89	66.99 ± 1.32
SEAM, no embedding	86.24 ± 0.99	85.57 ± 0.94	88.61 ± 0.71	91.20 ± 1.03	96.16 ± 0.55	98.40 ± 0.22	83.39 ± 0.94	86.12 ± 0.66	87.86 ± 1.06
SEAM	90.78 ± 1.30	90.00 ± 1.84	91.53 ± 1.53	93.06 ± 0.61	97.26 ± 0.23	98.90 ± 0.18	83.81 ± 0.53	87.56 ± 0.79	88.59 ± 1.51
	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star

SEAM: learning from Subgraphs, Embeddings and Attributes for Motif prediction

„db” – with deal-breaker edges

Evaluation

With correlation → **CN: Common Neighbors, AA: Adamic-Adar, SEAL [2]: Link Prediction using GNNs**
 No correlation ← **Mul, Min, Avg: different variants (i.e., how link scores are aggregated to form motif scores)**
 [2] M. Zhang and Y. Chen. 2018. Link prediction based on graph neural networks. NeurIPS'18

CN (Mul)	49.99 ± 0.45	49.78 ± 0.39	50.19 ± 0.47	50.13 ± 0.65	50.37 ± 0.79	51.55 ± 0.32	52.93 ± 0.61	55.42 ± 0.58	54.81 ± 0.58
CN (Min)	49.98 ± 0.33	49.72 ± 0.49	50.26 ± 0.59	50.35 ± 0.27	50.48 ± 0.32	51.77 ± 0.40	52.99 ± 0.75	54.75 ± 0.48	54.60 ± 0.85
CN (Avg)	49.76 ± 0.32	49.50 ± 0.64	50.18 ± 0.64	50.28 ± 0.51	50.91 ± 0.35	51.70 ± 0.59	53.32 ± 0.35	54.93 ± 0.77	54.20 ± 0.68
AA (Mul)	63.05 ± 0.71	62.09 ± 0.57	60.67 ± 0.95	54.95 ± 0.92	51.25 ± 0.63	51.40 ± 0.68	53.92 ± 0.52	55.15 ± 0.77	54.93 ± 0.61
AA (Min)	63.34 ± 0.68	62.81 ± 0.80	61.59 ± 0.94	54.81 ± 0.77	51.26 ± 0.38	51.60 ± 0.68	54.15 ± 0.89	54.59 ± 0.65	54.94 ± 0.27
AA (Avg)	63.96 ± 0.68	63.66 ± 0.48	62.71 ± 0.52	55.78 ± 0.74	51.28 ± 0.55	51.79 ± 0.62	54.52 ± 0.57	55.20 ± 0.53	54.55 ± 0.39
Jaccard (Mul)	67.17 ± 0.92	62.01 ± 0.72	59.71 ± 0.93	69.62 ± 1.09	57.60 ± 0.73	52.75 ± 0.97	51.75 ± 1.10	51.68 ± 0.85	50.93 ± 0.64
Jaccard (Min)	69.20 ± 0.80	67.11 ± 0.46	65.24 ± 0.80	73.88 ± 0.88	63.36 ± 1.17	56.50 ± 0.94	52.30 ± 0.54	51.86 ± 0.77	50.87 ± 0.53
Jaccard (Avg)	70.12 ± 0.78	68.59 ± 0.71	68.69 ± 0.77	75.35 ± 0.60	67.93 ± 0.87	61.22 ± 1.11	51.76 ± 0.91	49.74 ± 0.75	47.66 ± 0.58
SEAL (Mul)	76.68 ± 0.61	74.00 ± 0.50	71.80 ± 0.95	76.25 ± 1.90	63.66 ± 4.01	59.48 ± 4.87	68.53 ± 0.88	67.49 ± 1.27	67.88 ± 1.43
SEAL (Min)	77.15 ± 0.43	74.62 ± 0.55	73.11 ± 0.99	78.00 ± 1.49	69.70 ± 3.56	64.49 ± 5.47	66.40 ± 1.44	62.94 ± 1.98	62.88 ± 3.57
SEAL (Avg)	77.91 ± 0.91	75.98 ± 0.99	75.71 ± 0.66	77.50 ± 2.35	72.68 ± 3.21	66.95 ± 6.79	66.05 ± 0.78	65.14 ± 0.89	66.99 ± 1.32
SEAM, no embedding	86.24 ± 0.99	85.57 ± 0.94	88.61 ± 0.71	91.20 ± 1.03	96.16 ± 0.55	98.40 ± 0.22	83.39 ± 0.94	86.12 ± 0.66	87.86 ± 1.06
SEAM	90.78 ± 1.30	90.00 ± 1.84	91.53 ± 1.53	93.06 ± 0.61	97.26 ± 0.23	98.90 ± 0.18	83.81 ± 0.53	87.56 ± 0.79	88.59 ± 1.51
	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star

SEAM: learning from Subgraphs, Embeddings and Attributes for Motif prediction

„db” – with deal-breaker edges

Evaluation

With correlation → **CN: Common Neighbors, AA: Adamic-Adar, SEAL [2]: Link Prediction using GNNs**
 No correlation ← **Mul, Min, Avg: different variants (i.e., how link scores are aggregated to form motif scores)**
 [2] M. Zhang and Y. Chen. 2018. Link prediction based on graph neural networks. NeurIPS'18

CN (Mul)	49.99 ± 0.45	49.78 ± 0.39	50.19 ± 0.47	50.13 ± 0.65	50.37 ± 0.79	51.55 ± 0.32	52.93 ± 0.61	55.42 ± 0.58	54.81 ± 0.58
CN (Min)	49.98 ± 0.33	49.72 ± 0.49	50.26 ± 0.59	50.35 ± 0.27	50.48 ± 0.32	51.77 ± 0.40	52.99 ± 0.75	54.75 ± 0.48	54.60 ± 0.85
CN (Avg)	49.76 ± 0.32	49.50 ± 0.64	50.18 ± 0.64	50.28 ± 0.51	50.91 ± 0.35	51.70 ± 0.59	53.32 ± 0.35	54.93 ± 0.77	54.20 ± 0.68
AA (Mul)	63.05 ± 0.71	62.09 ± 0.57	60.67 ± 0.95	54.95 ± 0.92	51.25 ± 0.63	51.40 ± 0.68	53.92 ± 0.52	55.15 ± 0.77	54.93 ± 0.61
AA (Min)	63.34 ± 0.68	62.81 ± 0.80	61.59 ± 0.94	54.81 ± 0.77	51.26 ± 0.38	51.60 ± 0.68	54.15 ± 0.89	54.59 ± 0.65	54.94 ± 0.27
AA (Avg)	63.96 ± 0.68	63.66 ± 0.48	62.71 ± 0.52	55.78 ± 0.74	51.28 ± 0.55	51.79 ± 0.62	54.52 ± 0.57	55.20 ± 0.53	54.55 ± 0.39
Jaccard (Mul)	67.17 ± 0.92	62.01 ± 0.72	59.71 ± 0.93	69.62 ± 1.09	57.60 ± 0.73	52.75 ± 0.97	51.75 ± 1.10	51.68 ± 0.85	50.93 ± 0.64
Jaccard (Min)	69.20 ± 0.80	67.11 ± 0.46	65.24 ± 0.80	73.88 ± 0.88	63.36 ± 1.17	56.50 ± 0.94	52.30 ± 0.54	51.86 ± 0.77	50.87 ± 0.53
Jaccard (Avg)	70.12 ± 0.78	68.59 ± 0.71	68.69 ± 0.77	75.35 ± 0.60	67.93 ± 0.87	61.22 ± 1.11	51.76 ± 0.91	49.74 ± 0.75	47.66 ± 0.58
SEAL (Mul)	76.68 ± 0.61	74.00 ± 0.50	71.80 ± 0.95	76.25 ± 1.90	63.66 ± 4.01	59.48 ± 4.87	68.53 ± 0.88	67.49 ± 1.27	67.88 ± 1.43
SEAL (Min)	77.15 ± 0.43	74.62 ± 0.55	73.11 ± 0.99	78.00 ± 1.49	69.70 ± 3.56	64.49 ± 5.47	66.40 ± 1.44	62.94 ± 1.98	62.88 ± 3.57
SEAL (Avg)	77.91 ± 0.91	75.98 ± 0.99	75.71 ± 0.66	77.50 ± 2.35	72.68 ± 3.21	66.95 ± 6.79	66.05 ± 0.78	65.14 ± 0.89	66.99 ± 1.32
SEAM, no embedding	86.24 ± 0.99	85.57 ± 0.94	88.61 ± 0.71	91.20 ± 1.03	96.16 ± 0.55	98.40 ± 0.22	83.39 ± 0.94	86.12 ± 0.66	87.86 ± 1.06
SEAM	90.78 ± 1.30	90.00 ± 1.84	91.53 ± 1.53	93.06 ± 0.61	97.26 ± 0.23	98.90 ± 0.18	83.81 ± 0.53	87.56 ± 0.79	88.59 ± 1.51
	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star

SEAM: learning from Subgraphs, Embeddings and Attributes for Motif prediction

„db” – with deal-breaker edges

Evaluation

With correlation → **Mul, Min, Avg**: different variants (i.e., how link scores are aggregated to form motif scores)
 No correlation ↗

CN: Common Neighbors, **AA**: Adamic-Adar, **SEAL [2]**: Link Prediction using GNNs
 [2] M. Zhang and Y. Chen. 2018. Link prediction based on graph neural networks. NeurIPS'18

CN (Mul)	49.99 ± 0.45	49.78 ± 0.39	50.19 ± 0.47	50.13 ± 0.65	50.37 ± 0.79	51.55 ± 0.32	52.93 ± 0.61	55.42 ± 0.58	54.81 ± 0.58
CN (Min)	49.98 ± 0.33	49.72 ± 0.49	50.26 ± 0.59	50.35 ± 0.27	50.48 ± 0.32	51.77 ± 0.40	52.99 ± 0.75	54.75 ± 0.48	54.60 ± 0.85
CN (Avg)	49.76 ± 0.32	49.50 ± 0.64	50.18 ± 0.64	50.28 ± 0.51	50.91 ± 0.35	51.70 ± 0.59	53.32 ± 0.35	54.93 ± 0.77	54.20 ± 0.68
AA (Mul)	63.05 ± 0.71	62.09 ± 0.57	60.67 ± 0.95	54.95 ± 0.92	51.25 ± 0.63	51.40 ± 0.68	53.92 ± 0.52	55.15 ± 0.77	54.93 ± 0.61
AA (Min)	63.34 ± 0.68	62.81 ± 0.68	62.81 ± 0.68	62.81 ± 0.68	62.81 ± 0.68	62.81 ± 0.68	54.15 ± 0.89	54.59 ± 0.65	54.94 ± 0.27
AA (Avg)	63.96 ± 0.68	63.66 ± 0.68	63.66 ± 0.68	63.66 ± 0.68	63.66 ± 0.68	63.66 ± 0.68	54.52 ± 0.57	55.20 ± 0.53	54.55 ± 0.39
Jaccard (Mul)	67.17 ± 0.92	62.01 ± 0.92	62.01 ± 0.92	62.01 ± 0.92	62.01 ± 0.92	62.01 ± 0.92	51.75 ± 1.10	51.68 ± 0.85	50.93 ± 0.64
Jaccard (Min)	69.20 ± 0.80	67.11 ± 0.80	67.11 ± 0.80	67.11 ± 0.80	67.11 ± 0.80	67.11 ± 0.80	52.30 ± 0.54	51.86 ± 0.77	50.87 ± 0.53
Jaccard (Avg)	70.12 ± 0.78	68.59 ± 0.78	68.59 ± 0.78	68.59 ± 0.78	68.59 ± 0.78	68.59 ± 0.78	51.76 ± 0.91	49.74 ± 0.75	47.66 ± 0.58
SEAL (Mul)	76.68 ± 0.61	74.00 ± 0.61	74.00 ± 0.61	74.00 ± 0.61	74.00 ± 0.61	74.00 ± 0.61	68.53 ± 0.88	67.49 ± 1.27	67.88 ± 1.43
SEAL (Min)	77.15 ± 0.43	74.62 ± 0.55	73.11 ± 0.99	78.00 ± 1.49	69.70 ± 3.56	64.49 ± 5.47	66.40 ± 1.44	62.94 ± 1.98	62.88 ± 3.57
SEAL (Avg)	77.91 ± 0.91	75.98 ± 0.99	75.71 ± 0.66	77.50 ± 2.35	72.68 ± 3.21	66.95 ± 6.79	66.05 ± 0.78	65.14 ± 0.89	66.99 ± 1.32
SEAM, no embedding	86.24 ± 0.99	85.57 ± 0.94	88.61 ± 0.71	91.20 ± 1.03	96.16 ± 0.55	98.40 ± 0.22	83.39 ± 0.94	86.12 ± 0.66	87.86 ± 1.06
SEAM	90.78 ± 1.30	90.00 ± 1.84	91.53 ± 1.53	93.06 ± 0.61	97.26 ± 0.23	98.90 ± 0.18	83.81 ± 0.53	87.56 ± 0.79	88.59 ± 1.51
	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star

Much more results and sensitivity analyses in the paper 😊

SEAM: learning from Subgraphs, Embeddings and Attributes for Motif prediction

„db” – with deal-breaker edges

MACIEJ BESTA, TORSTEN HOEFLER, ET AL.

Motif Prediction with Graph Neural Networks

Thank you for your attention

