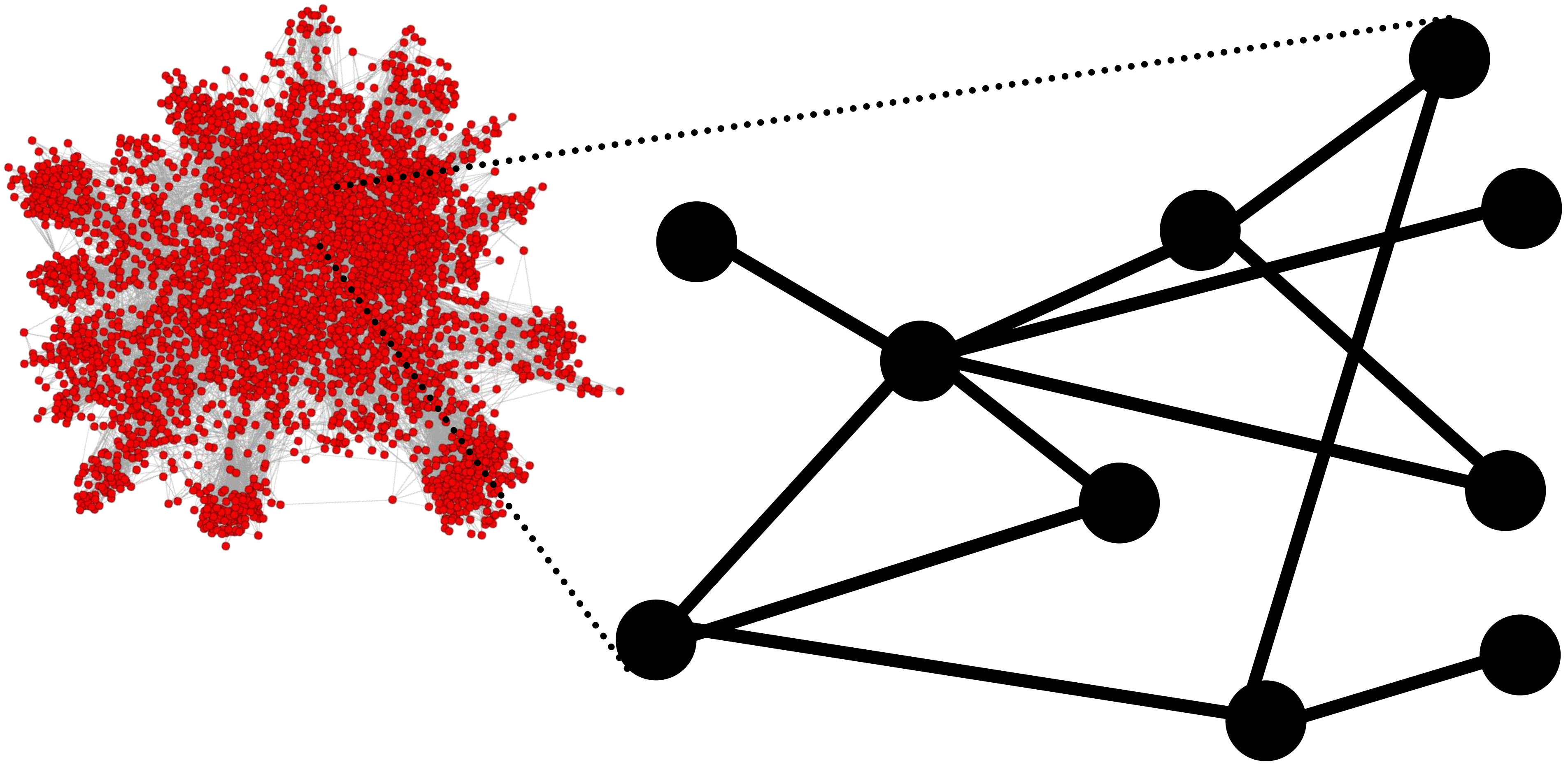
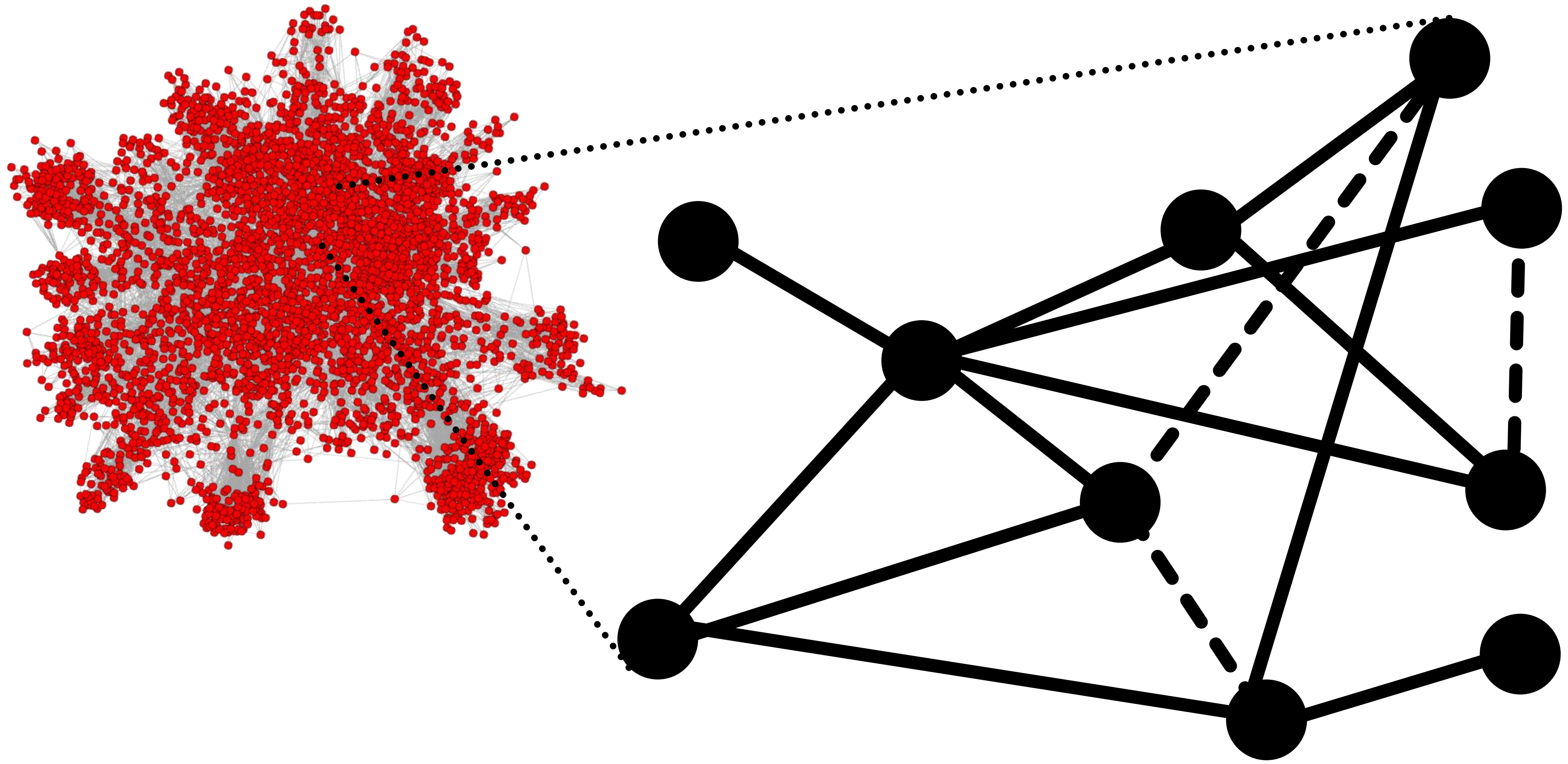
MACIEJ BESTA, TORSTEN HOEFLER, ET AL.

Motif Prediction with Graph Neural Networks

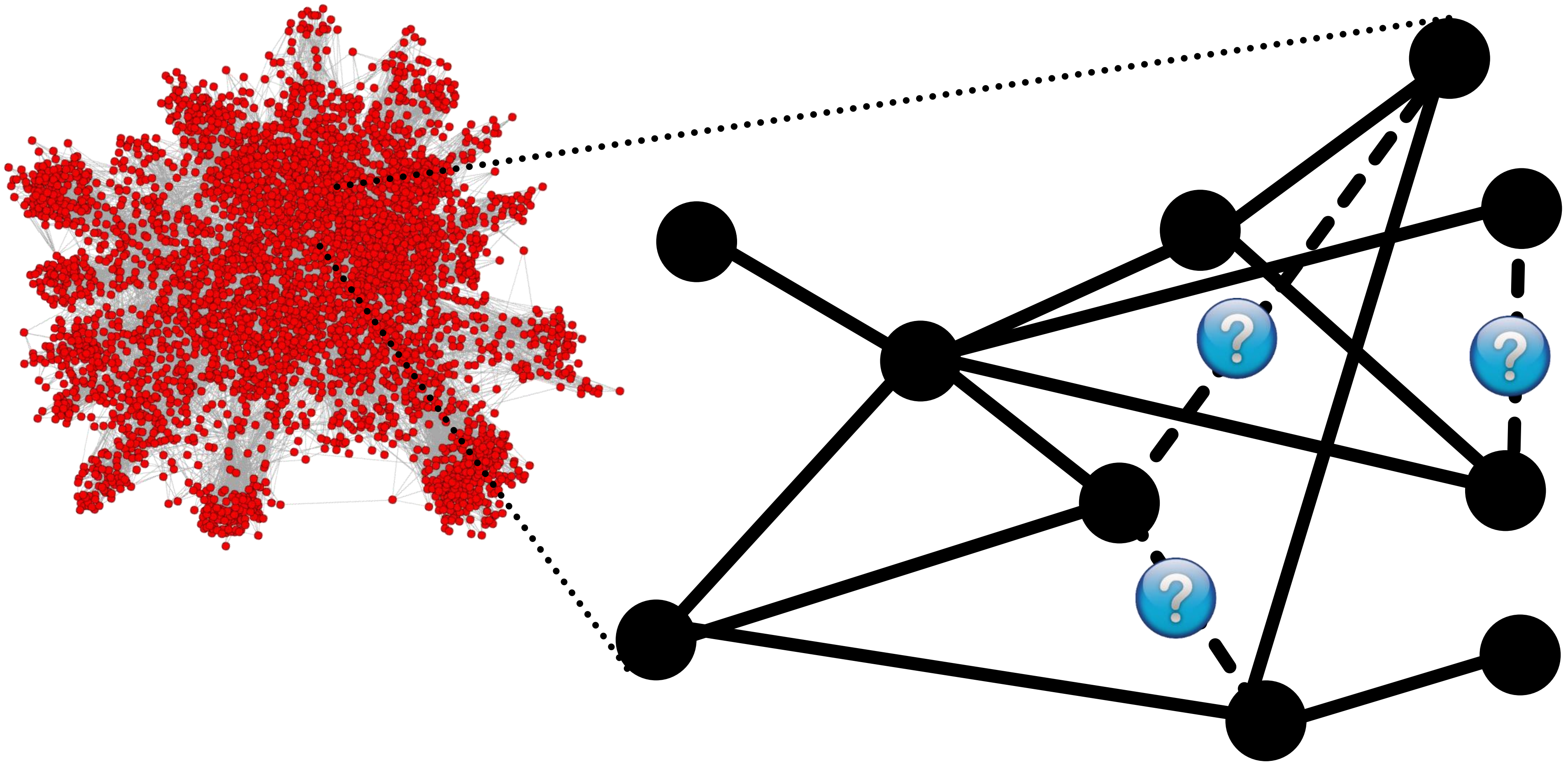
Starting Point: Link Prediction



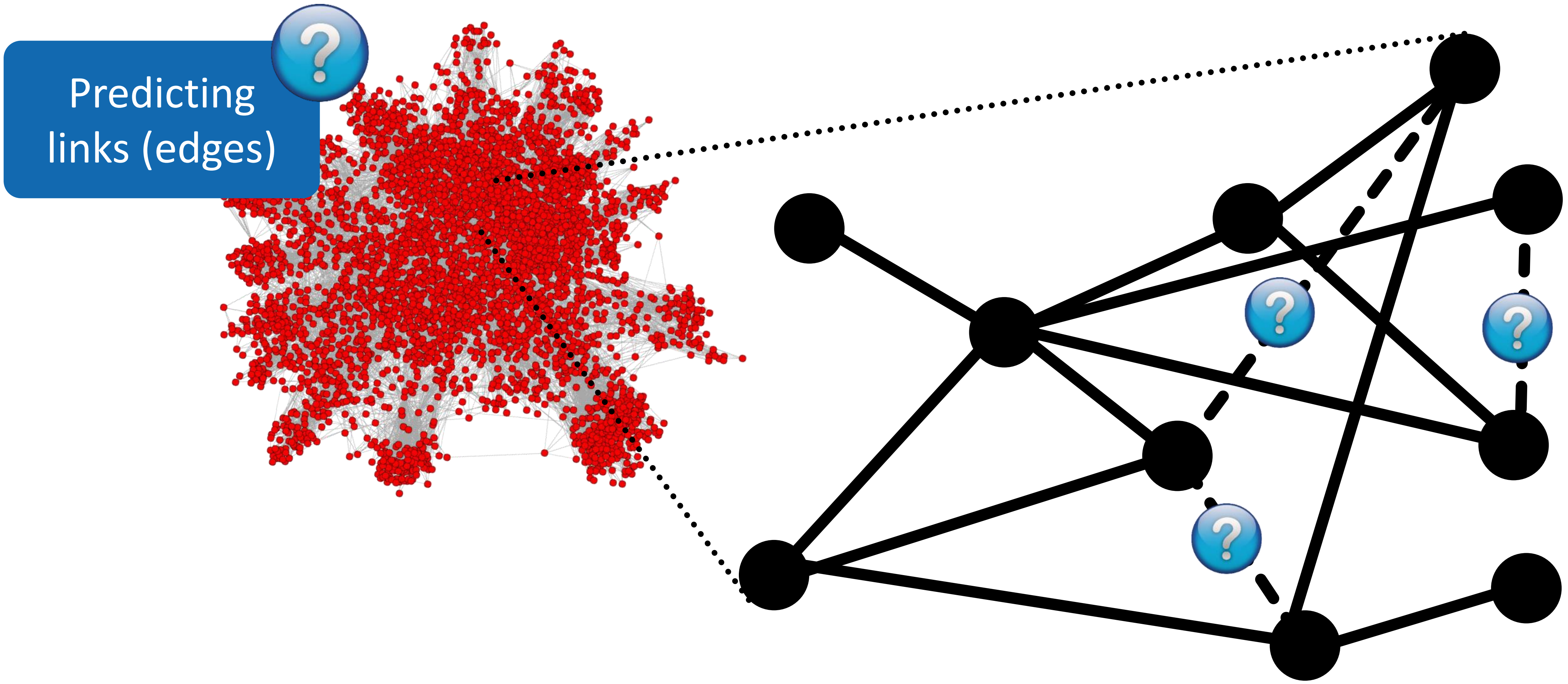
Starting Point: Link Prediction



Starting Point: Link Prediction

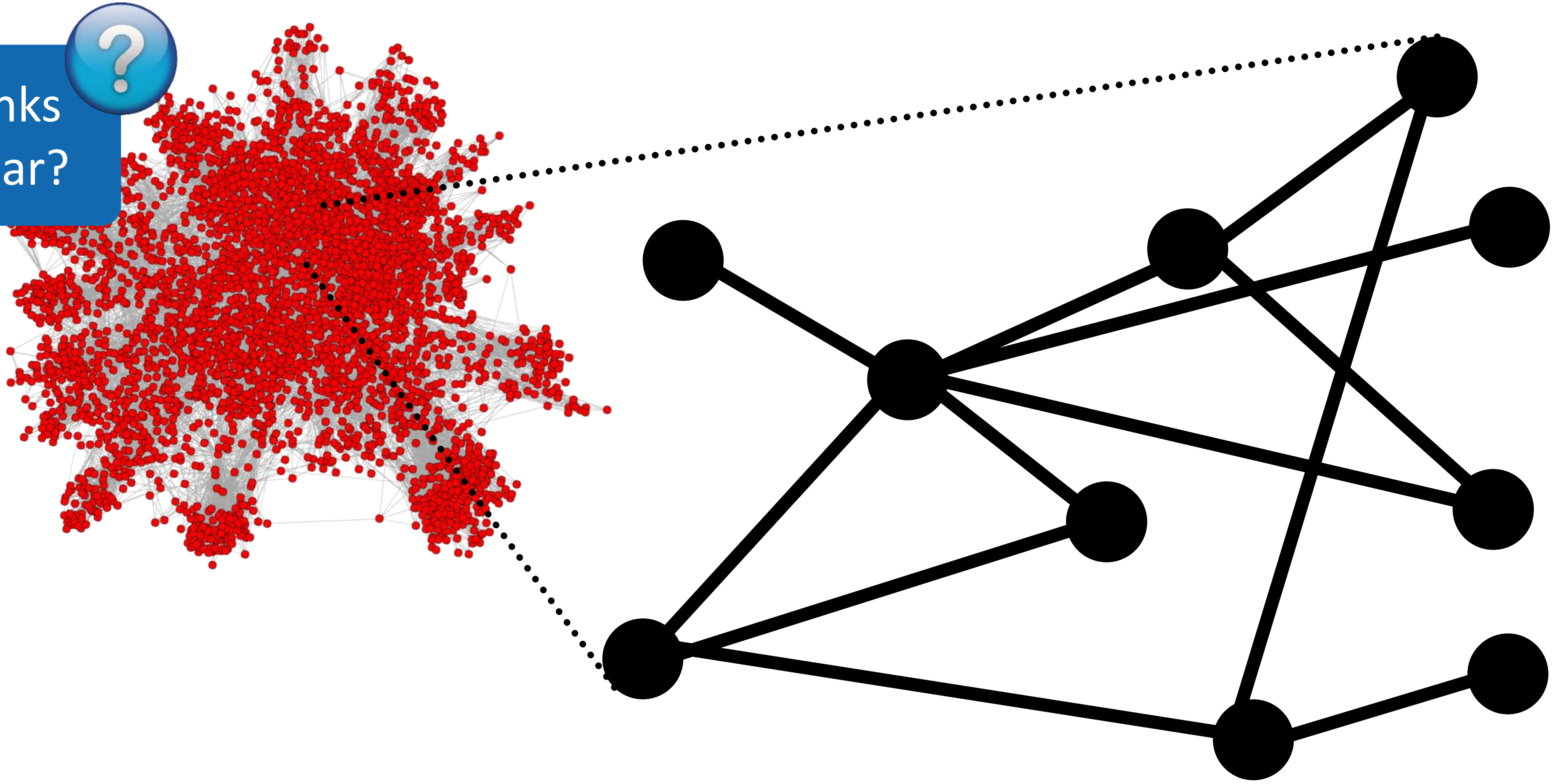


Starting Point: Link Prediction

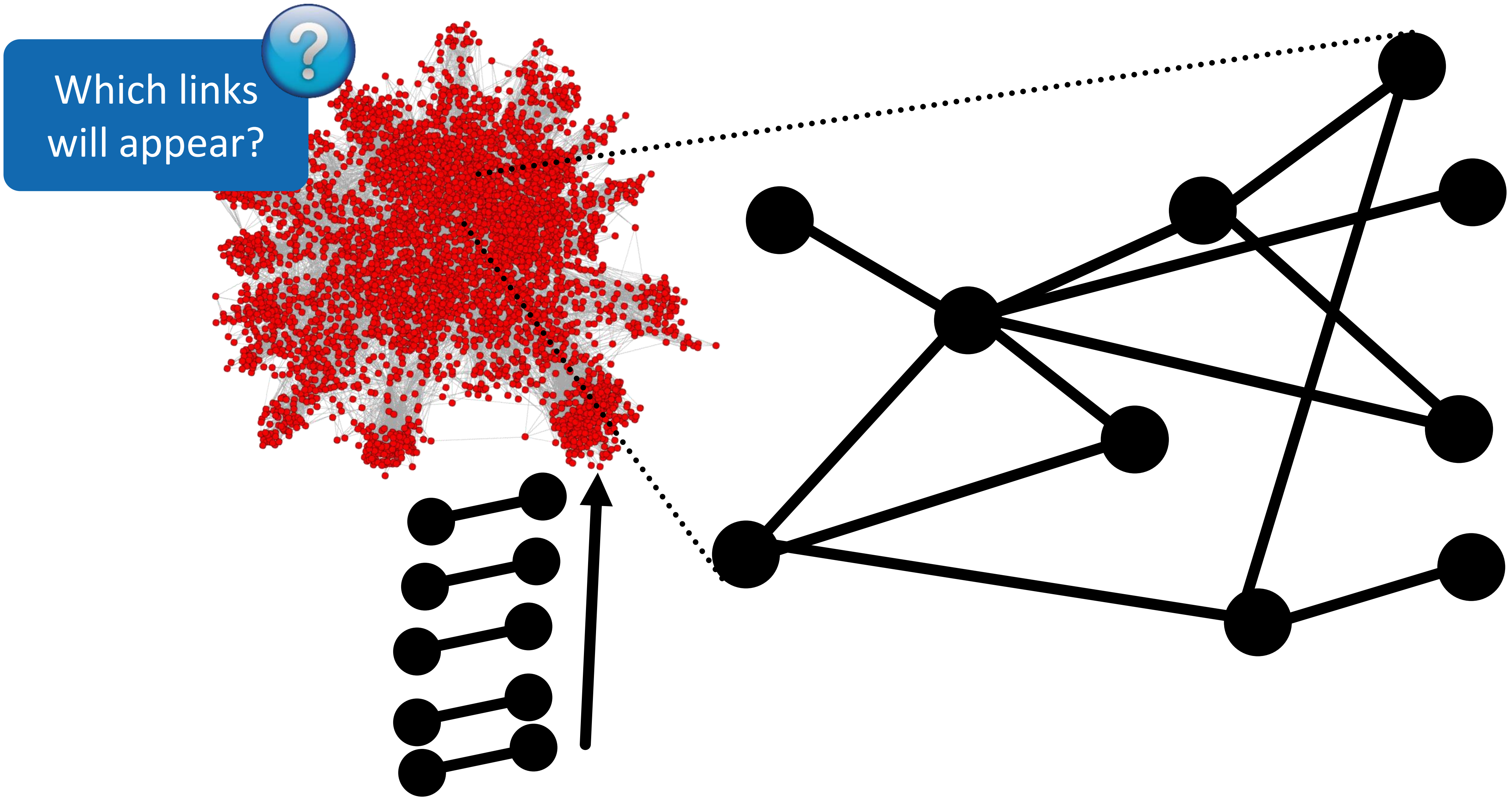


Case 1: New Edges Appear (Dynamic Graphs)

Which links will appear?



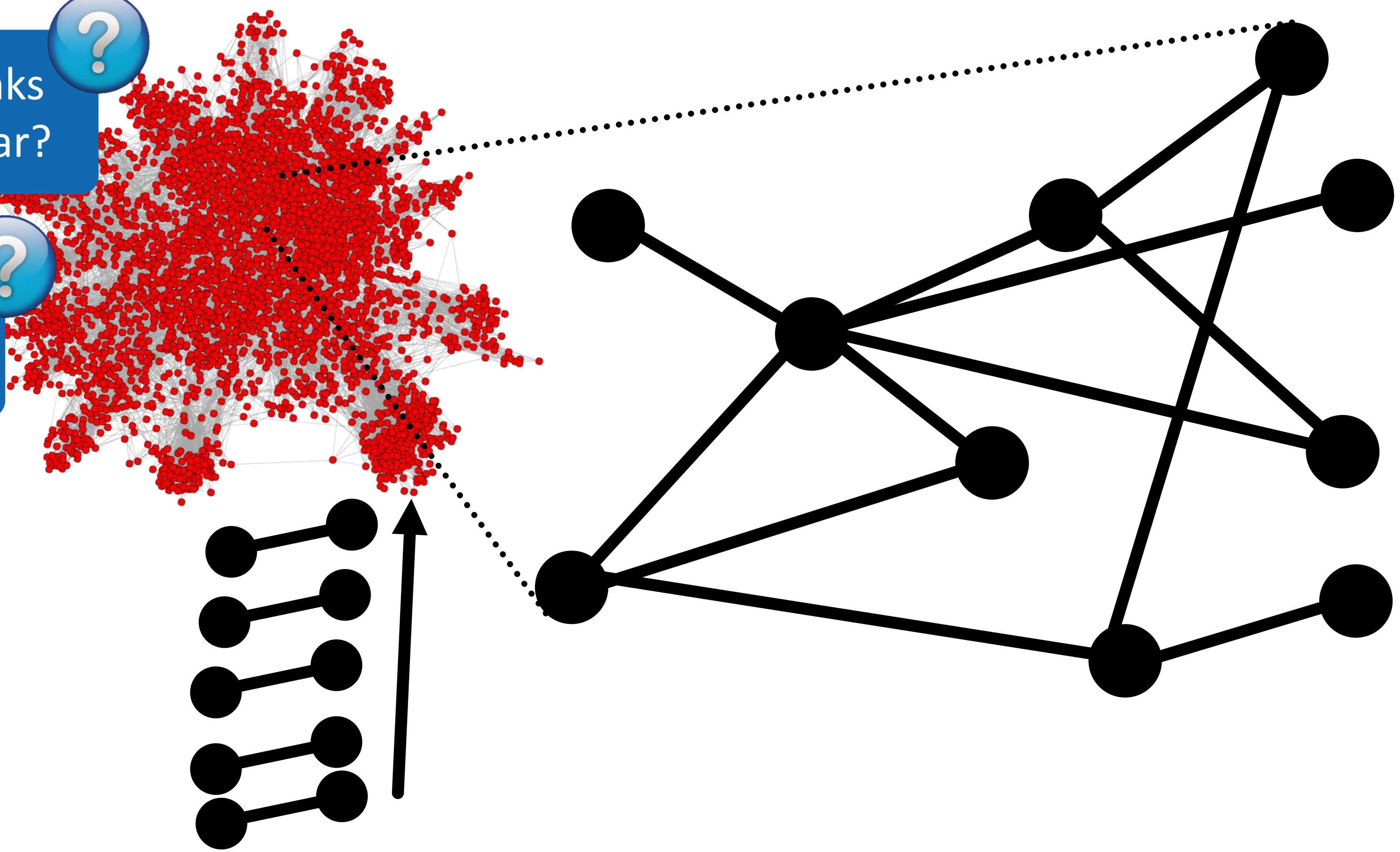
Case 1: New Edges Appear (Dynamic Graphs)



Case 1: New Edges Appear (Dynamic Graphs)

Which links will appear?

Why do we care?

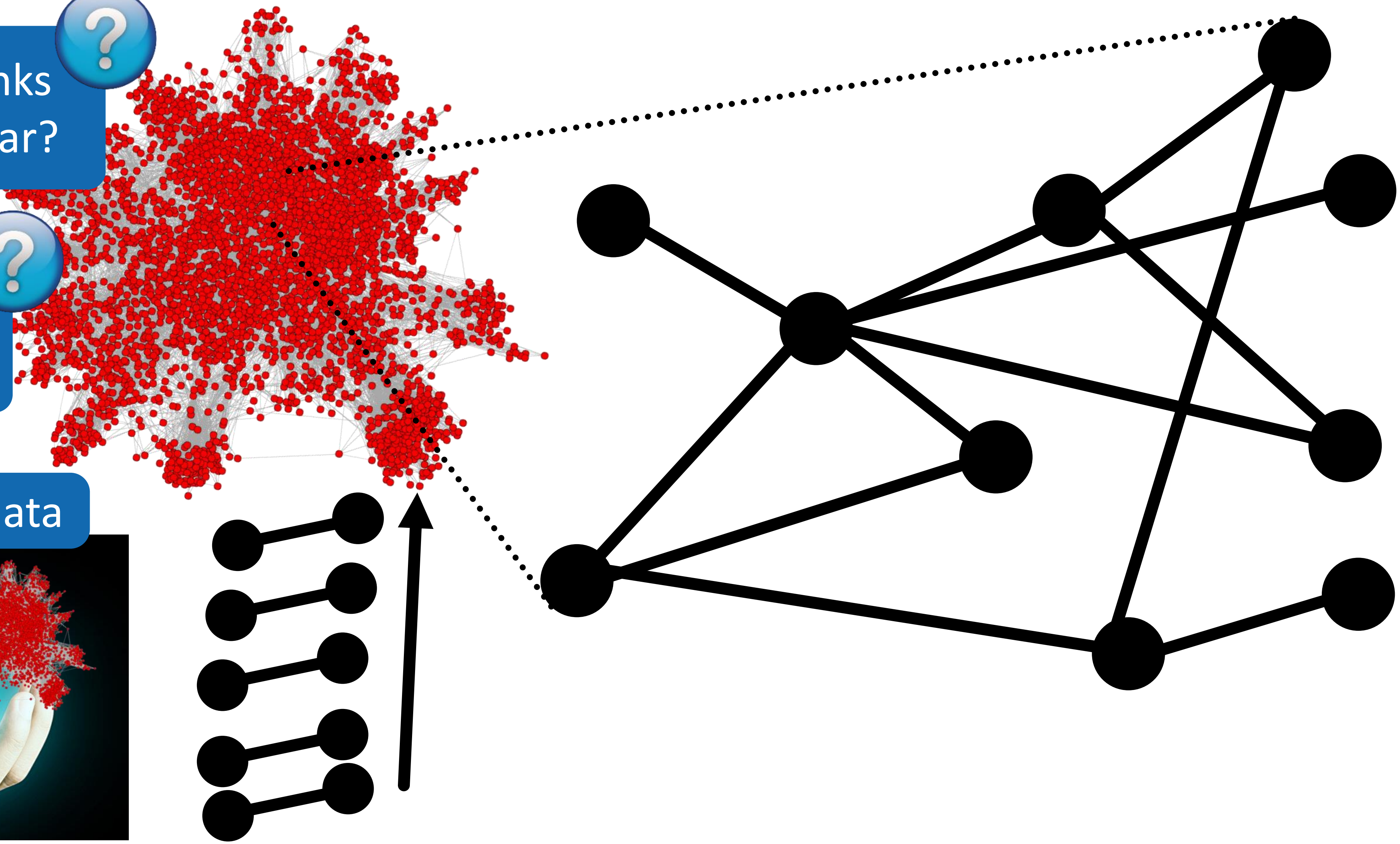


Case 1: New Edges Appear (Dynamic Graphs)

Which links will appear?

Why do we care?

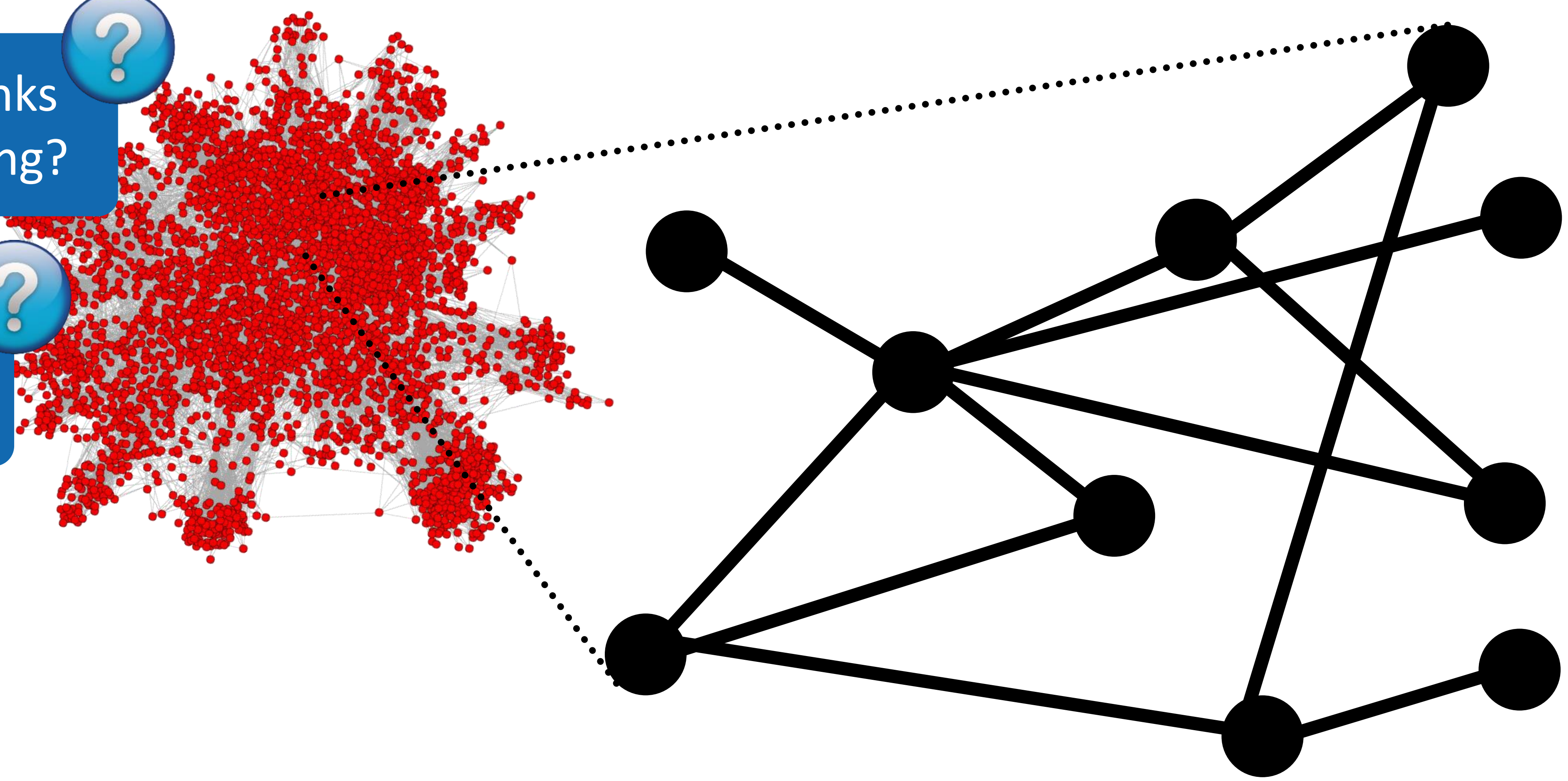
Predict future data



Case 2: Find Missing Data

Which links are missing?

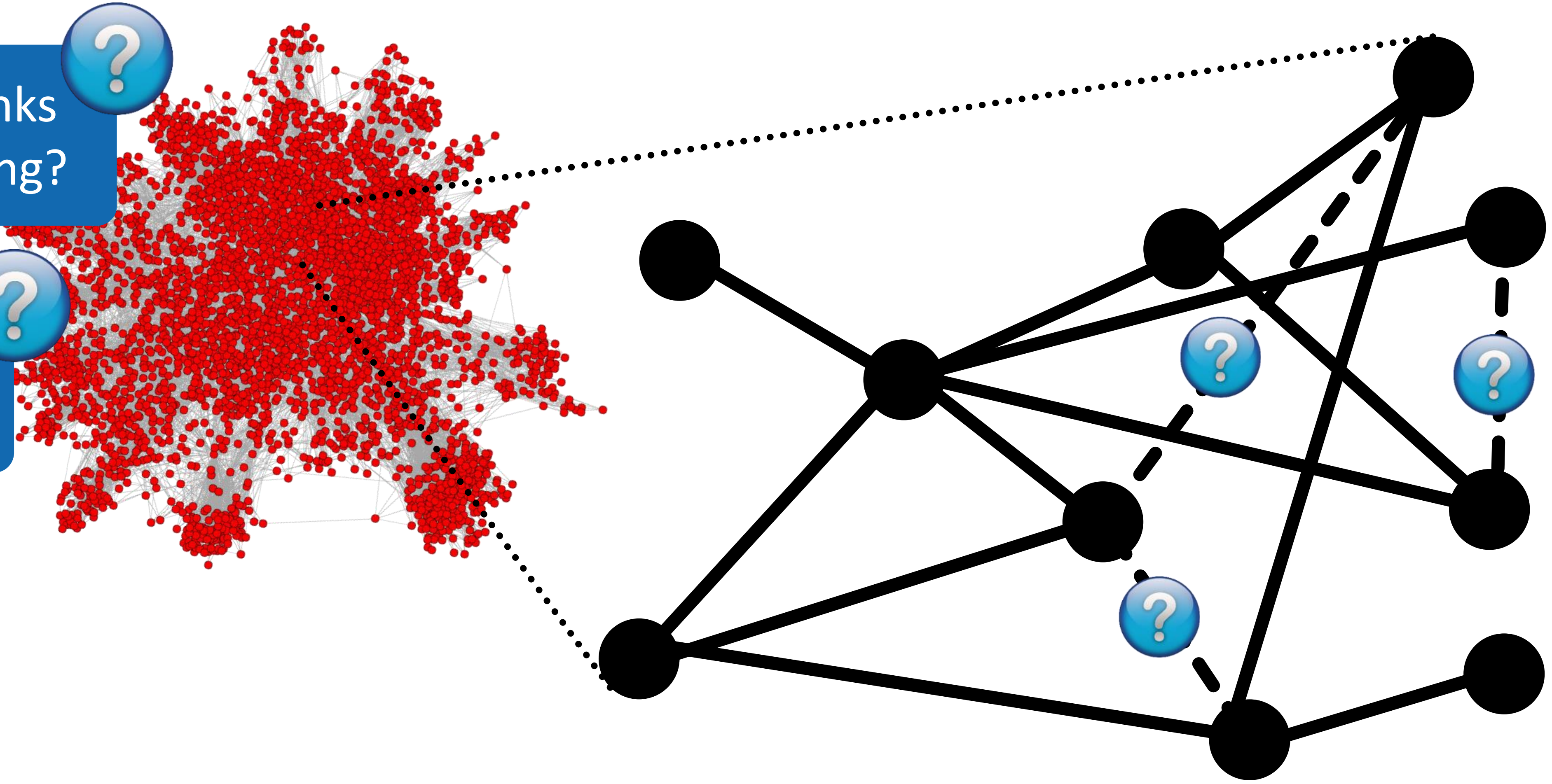
Why do we care?



Case 2: Find Missing Data

Which links are missing?

Why do we care?

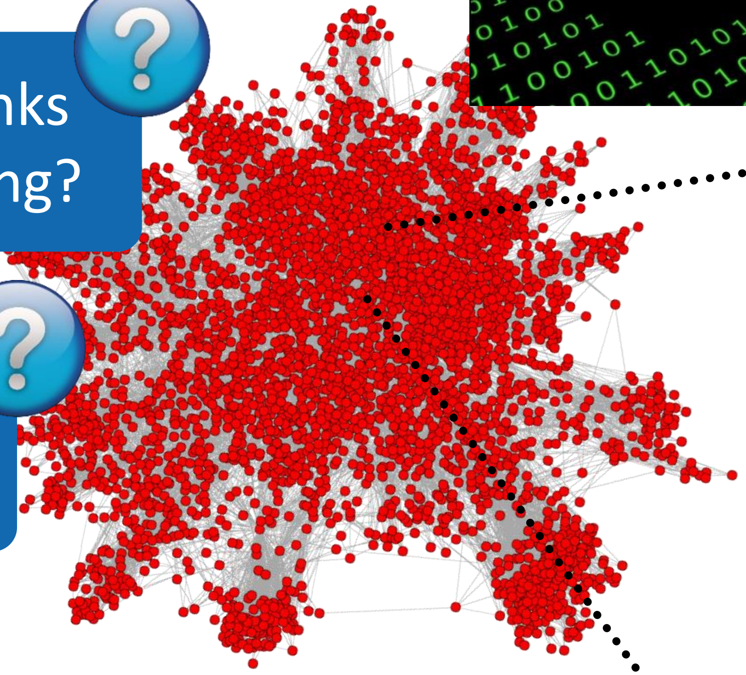


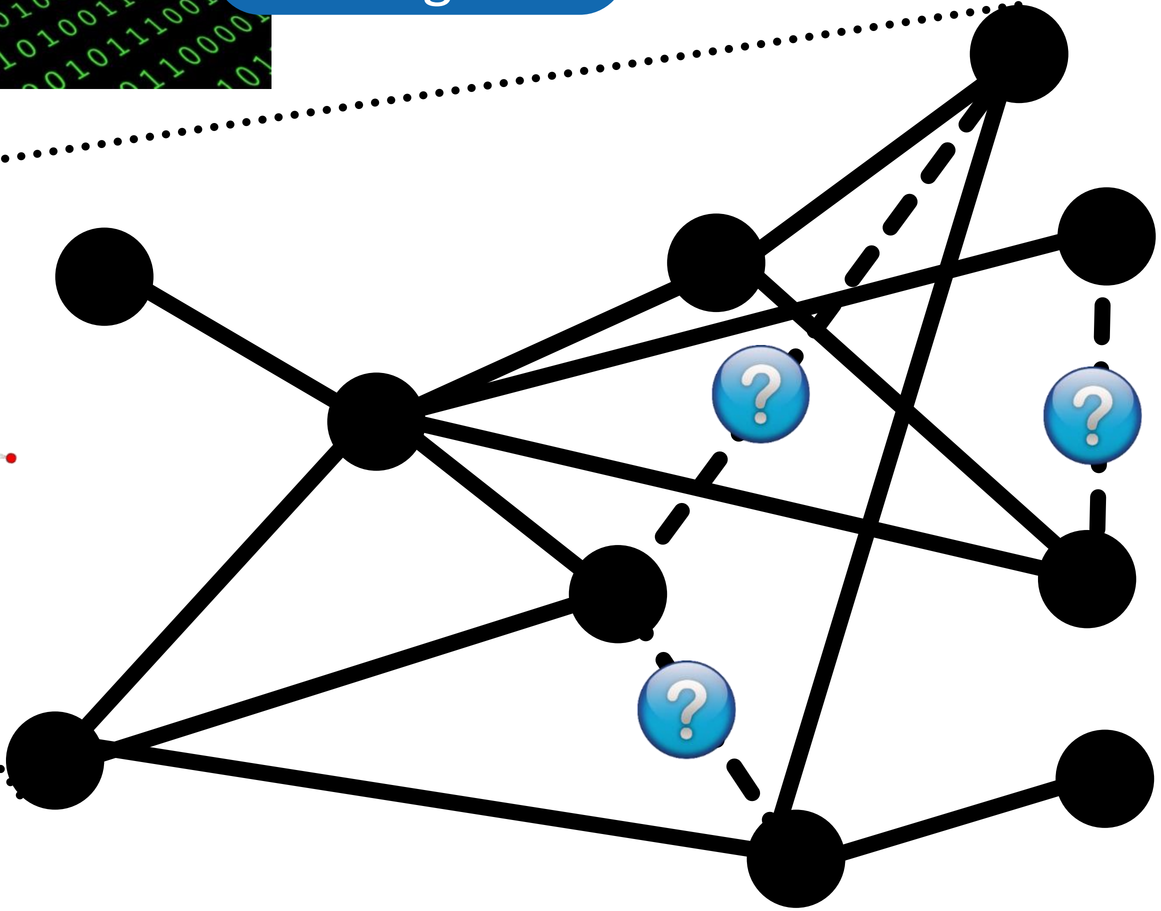
Case 2: Find Missing Data

Fixing missing data

Which links are missing?

Why do we care?



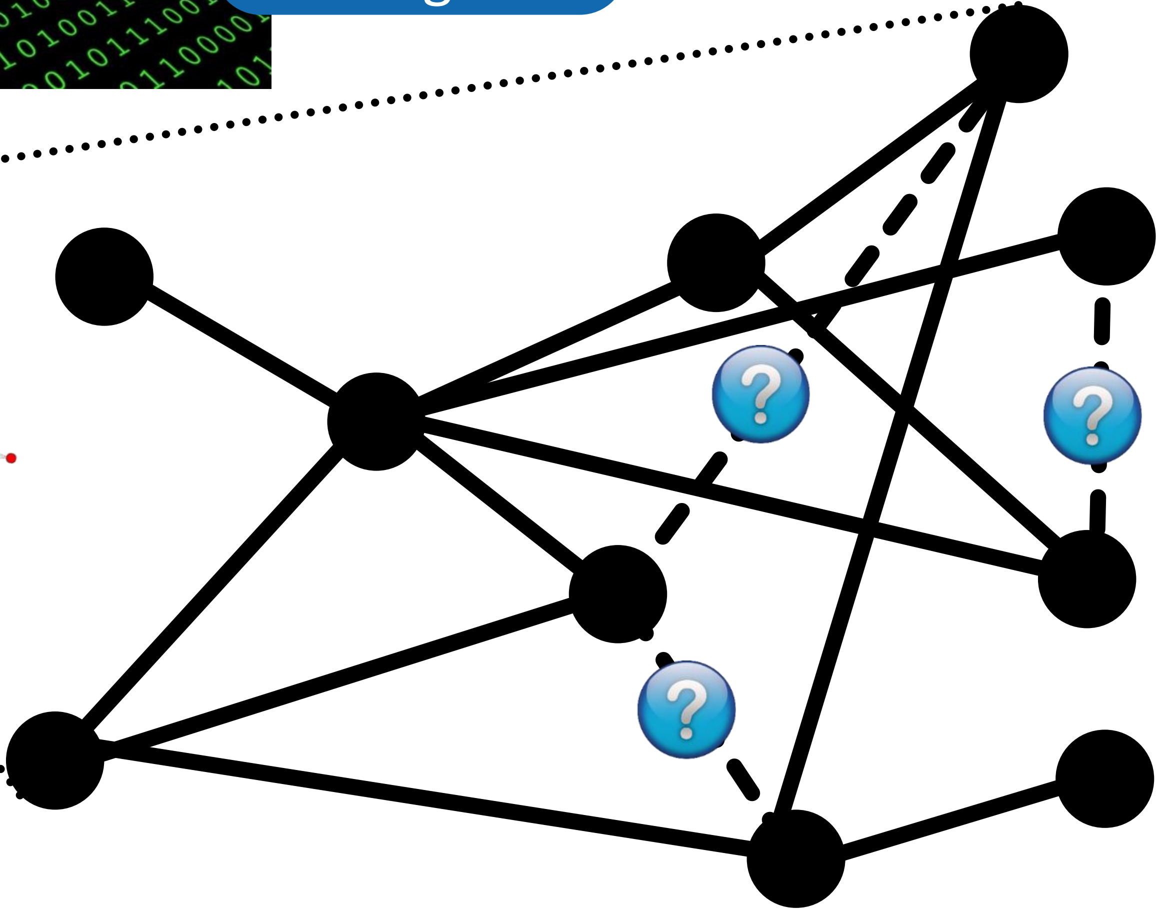


Case 2: Find Missing Data

Fixing missing data

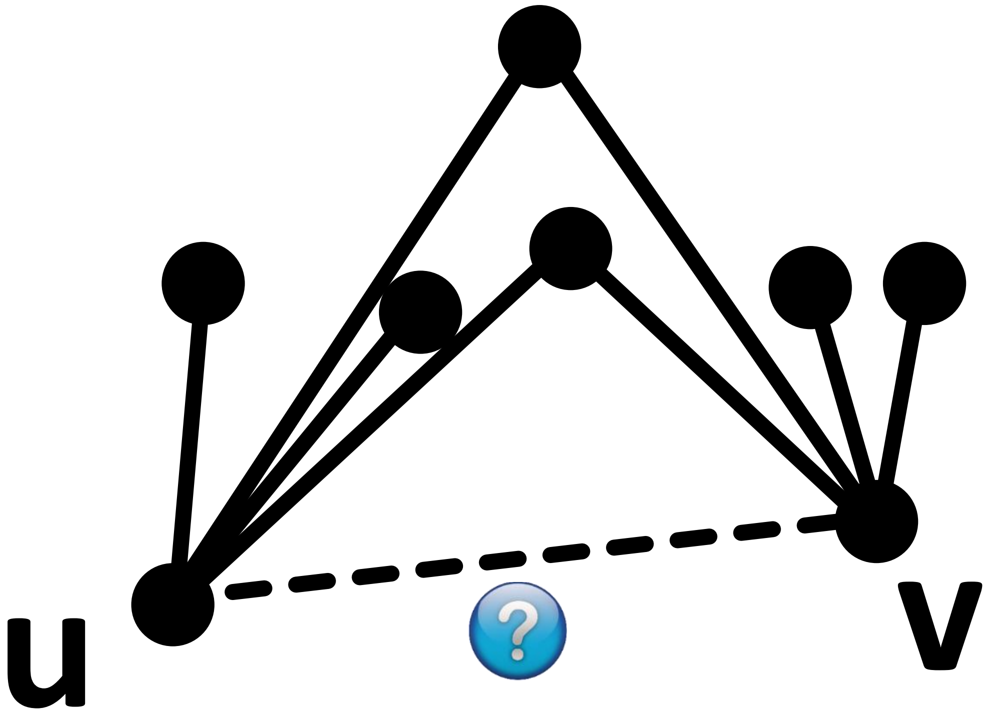
Which links are missing?

Why do we care?



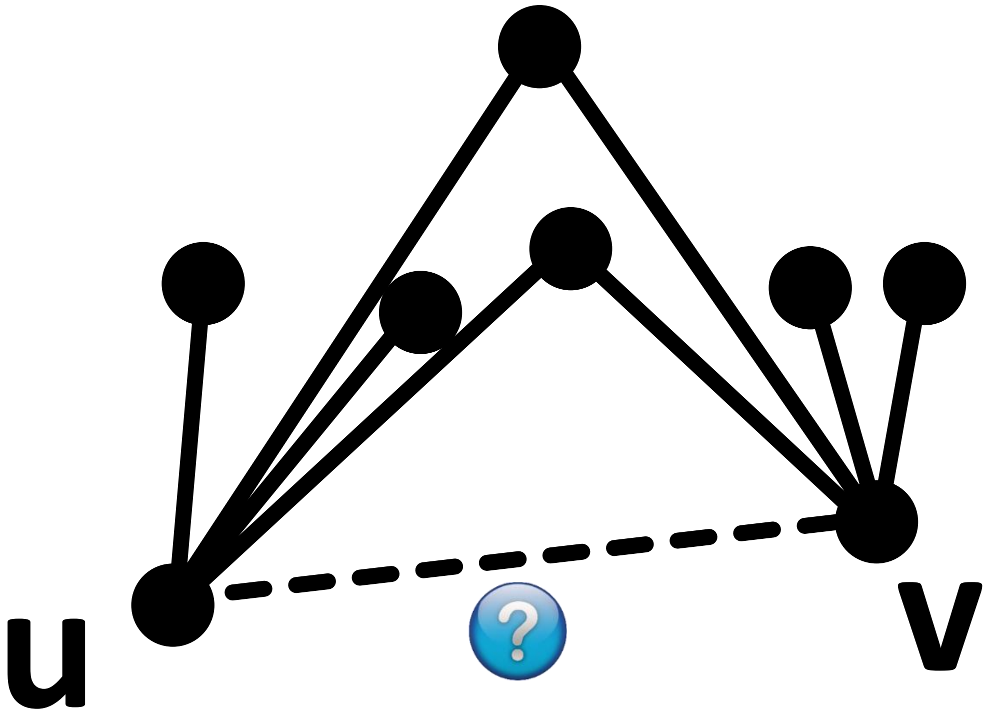
Reduce experiment costs

Warmup: Link Prediction



Warmup: Link Prediction

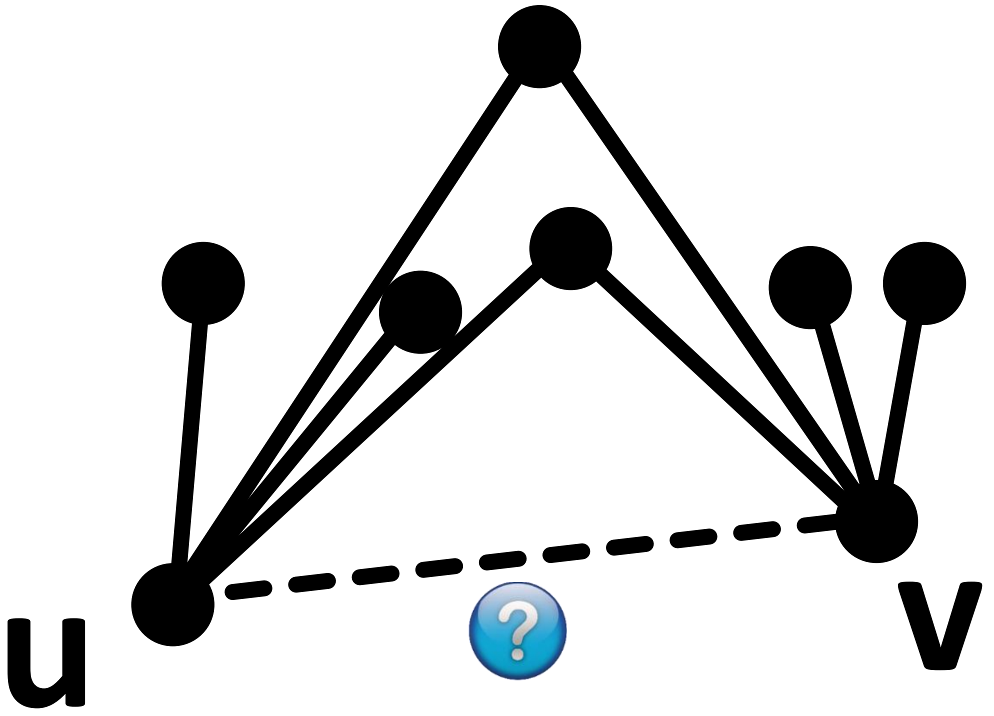
How to assess?



Warmup: Link Prediction

How to assess?

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

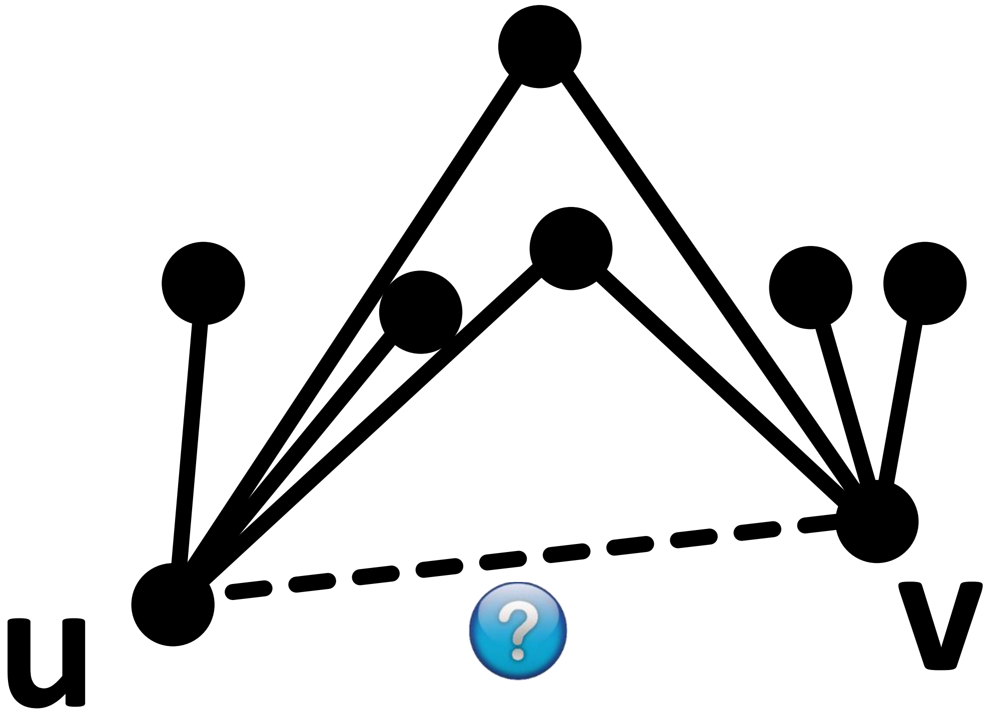


Warmup: Link Prediction

How to assess?

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

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

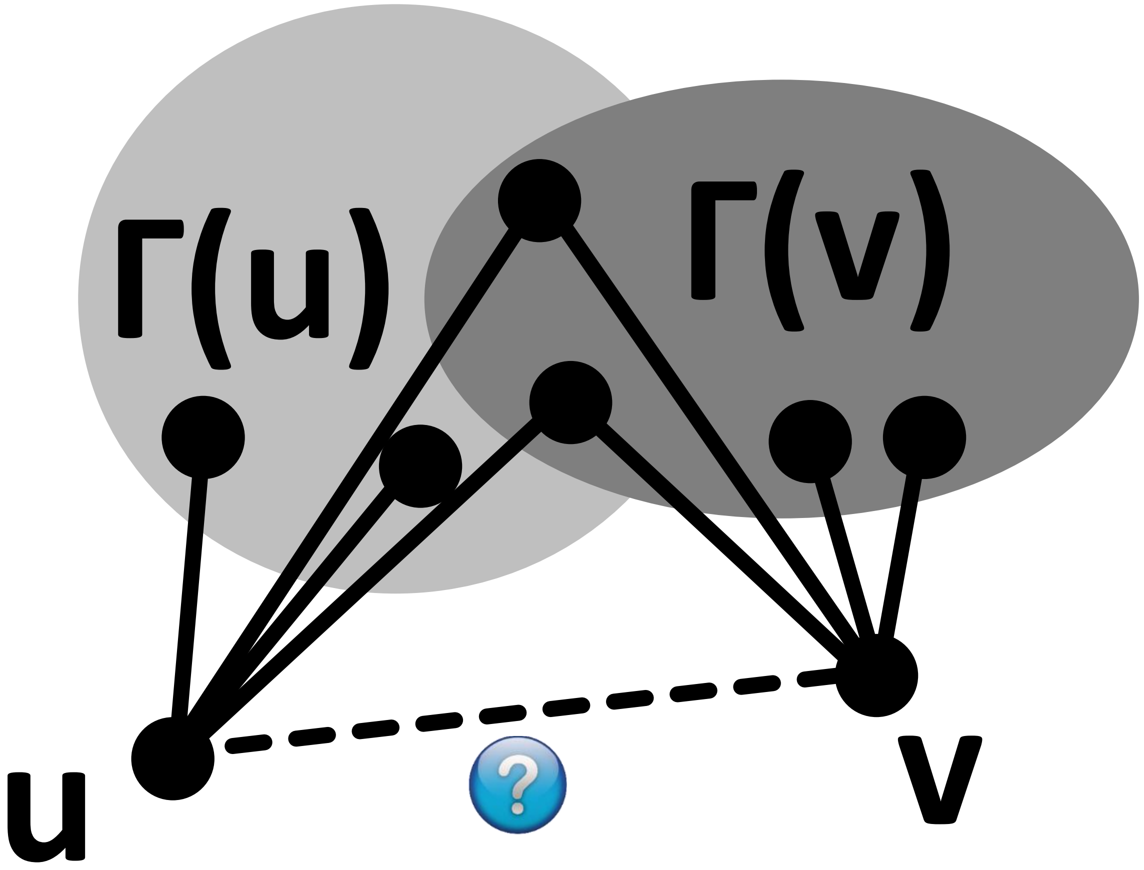


Warmup: Link Prediction

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

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

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**

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

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

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

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

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

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

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

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

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

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

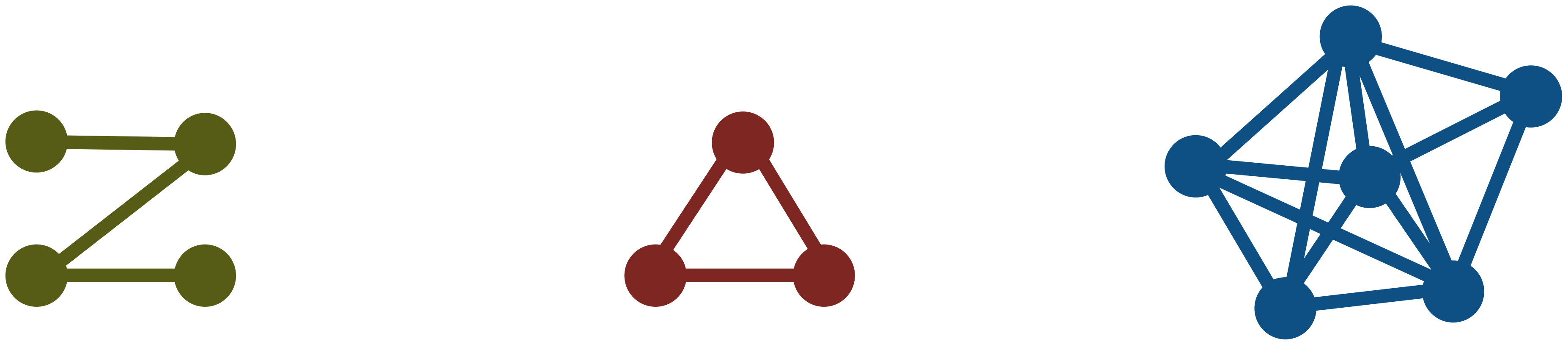
Idea: Generalize Link Prediction to Motifs

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)

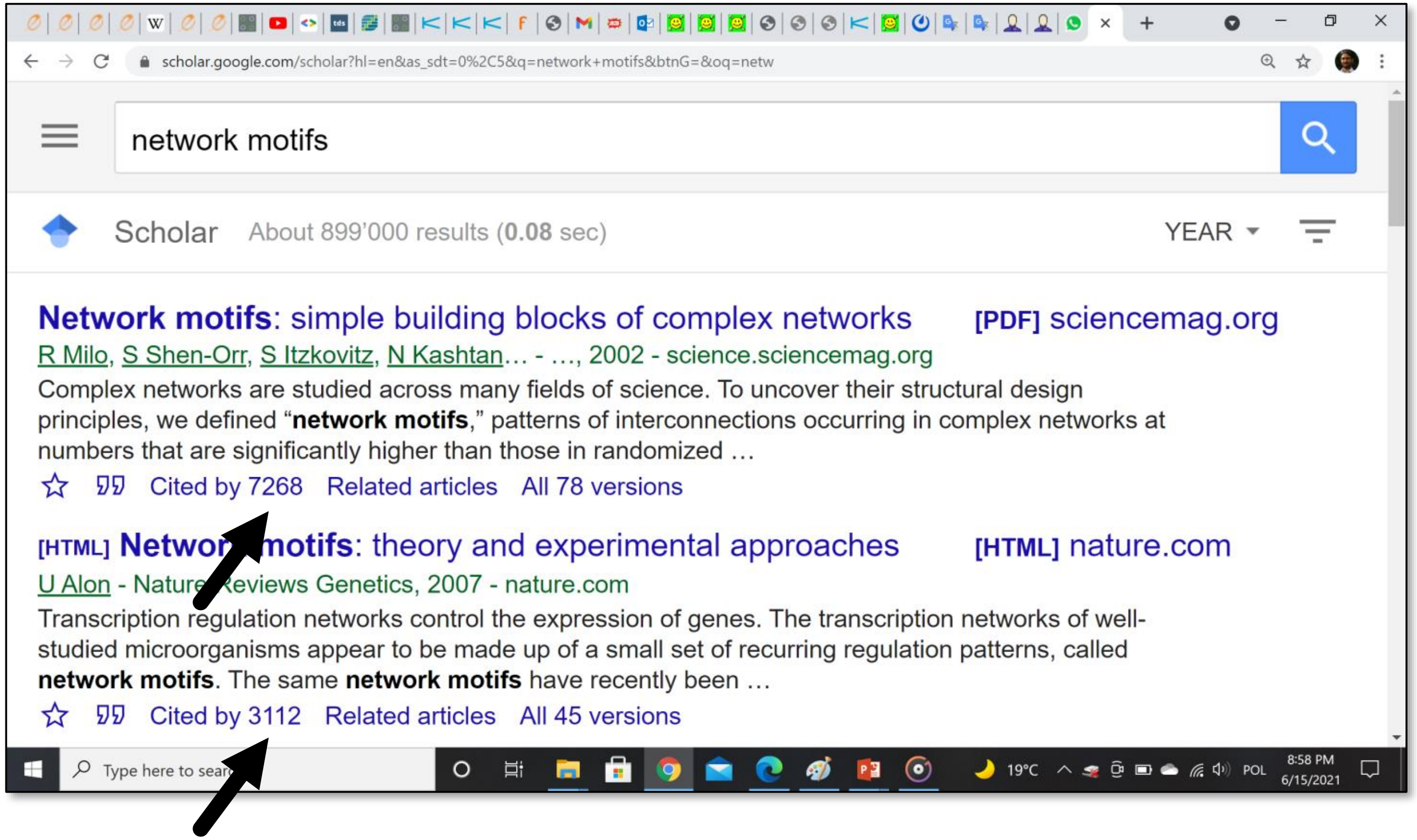


Motifs

Is this important?

Motifs

Is this important?

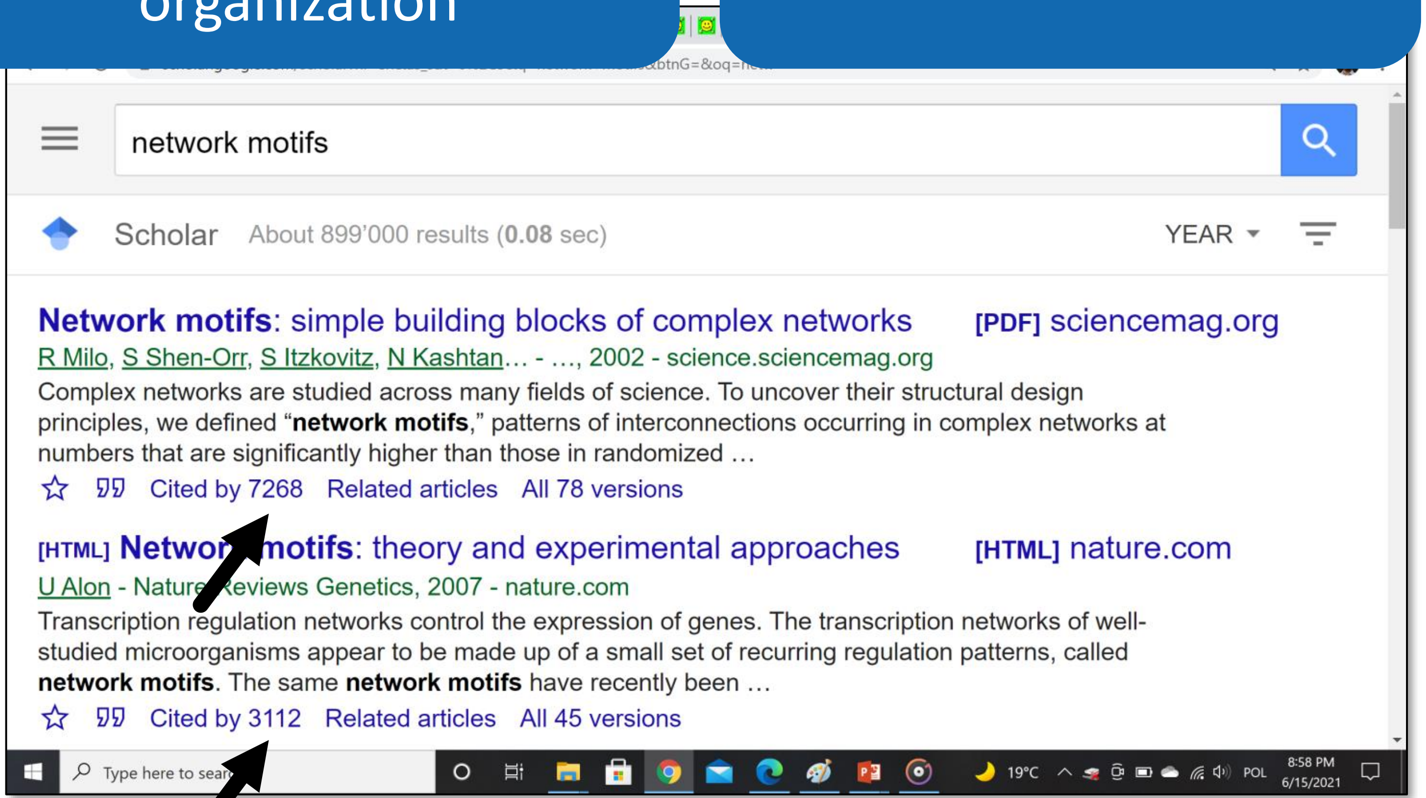


Motifs

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")

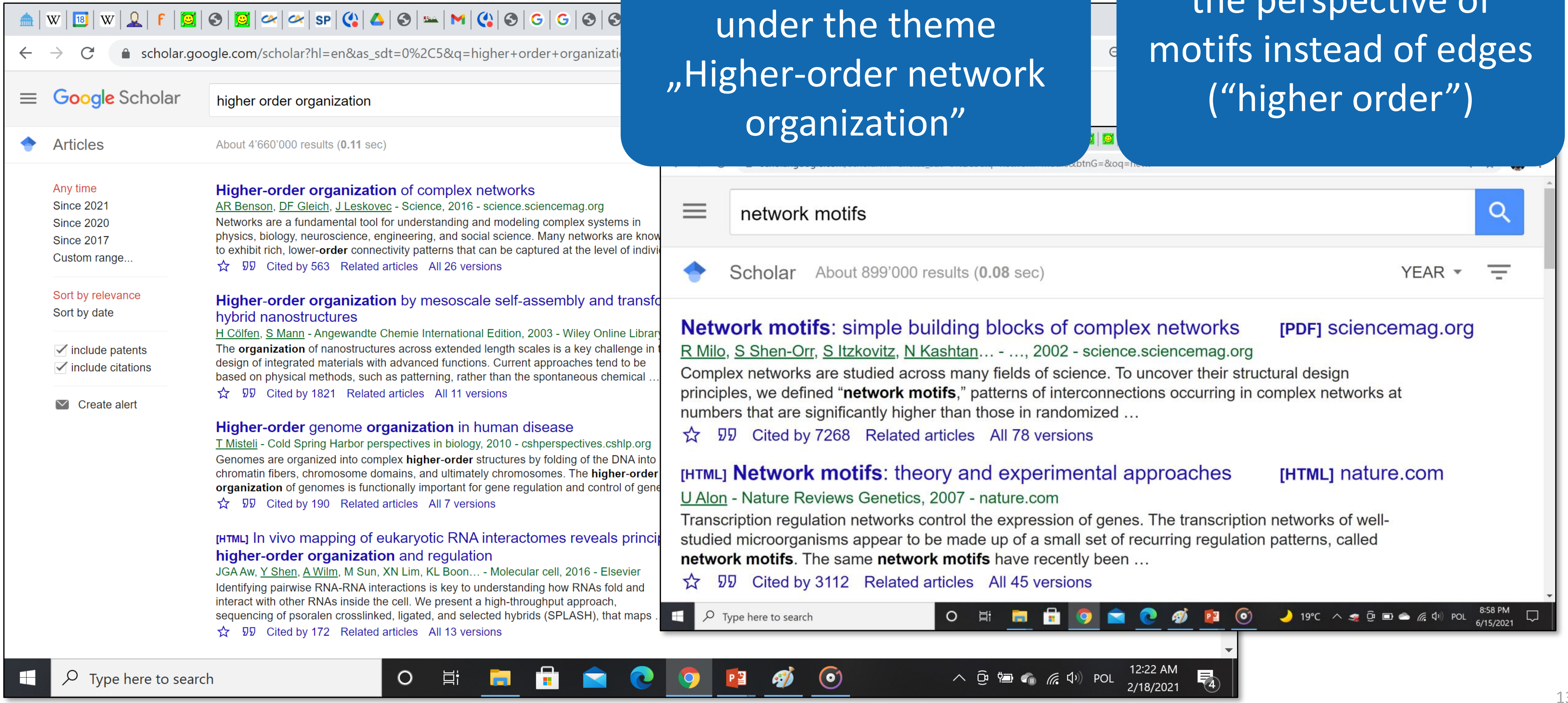


Motifs

Is this important?

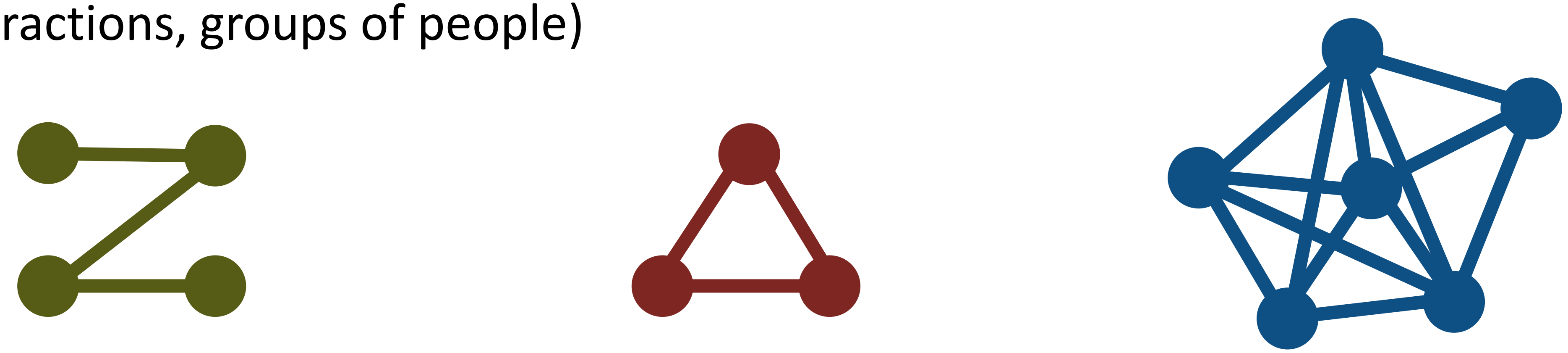
A lot of work into motifs exists, recent interest under the theme „Higher-order network organization”

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



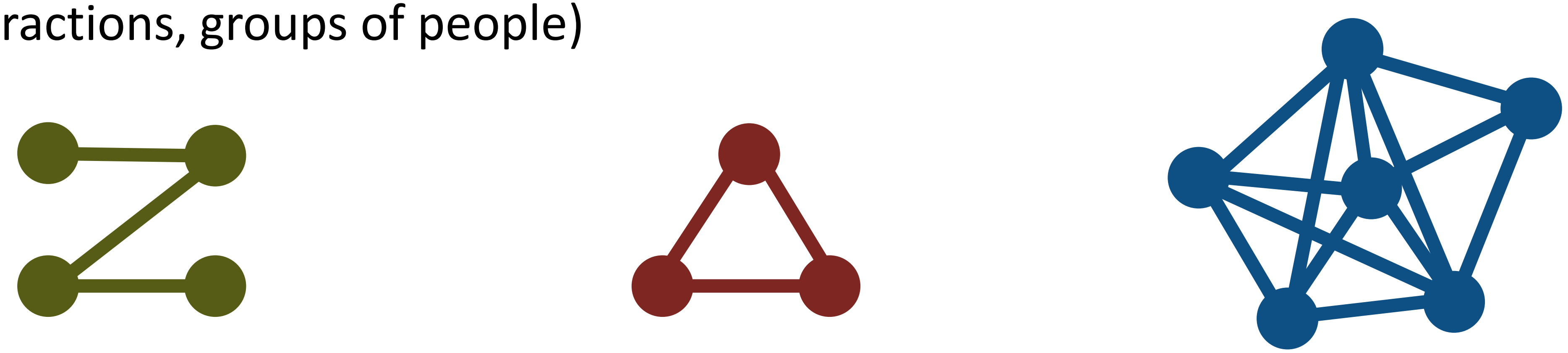
Idea: Generalize Link Prediction to Motifs

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



Idea: Generalize Link Prediction to Motifs

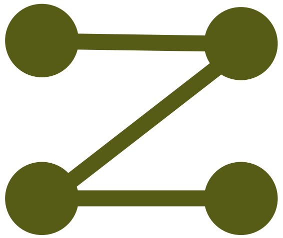
Motifs: small, recurring subgraphs (e.g. modelling molecules, gene interactions, groups of people)

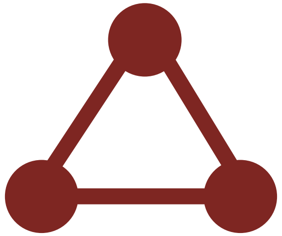


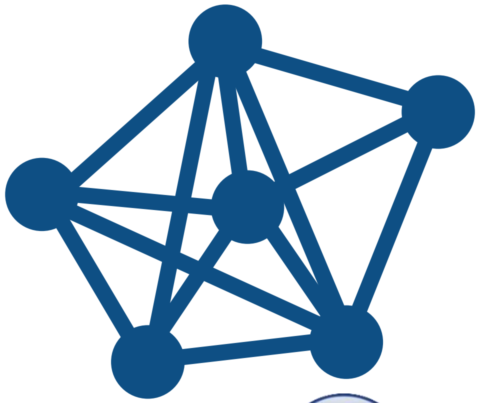
Link prediction is well understood for, well, links (scores assigned to links, similarity based methods, etc.)

Idea: Generalize Link Prediction to Motifs

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





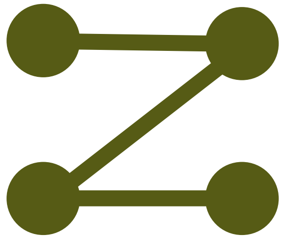


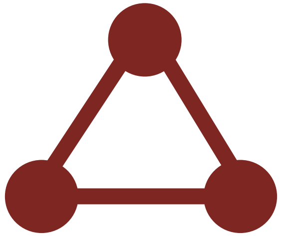
Link prediction is well understood for, well, links (scores assigned to links, similarity based methods, etc.)

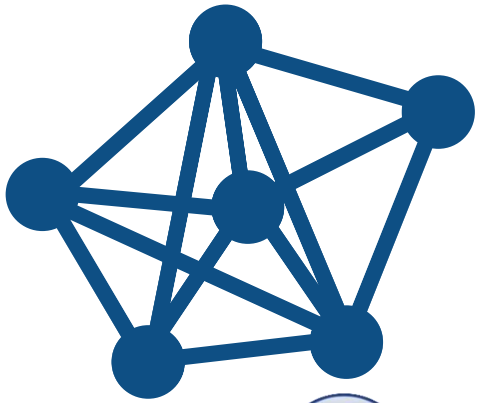
How to generalize to motifs?

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?

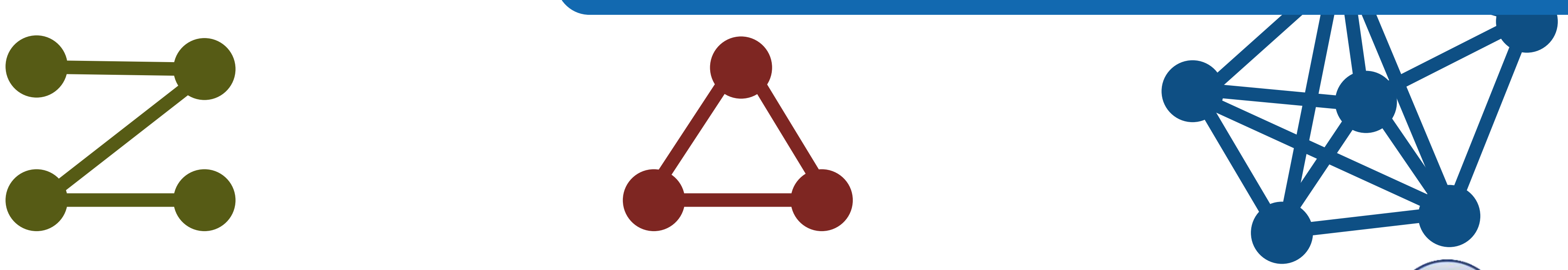
General vision: assign some score to motifs (make them comparable)

Motifs with higher scores are more probable

Idea: Generalize Link Pr

Motifs: small, recurring subgr
interactions, groups of people

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



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

How to generalize to motifs?

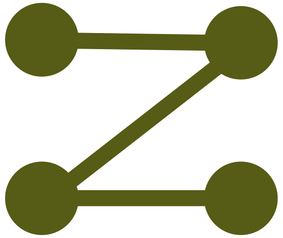
General vision: assign some score to motifs (make them comparable)

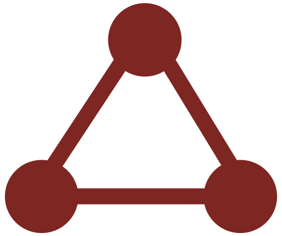
Motifs with higher scores are more probable

Idea: Generalize Link Pr

Motifs: small, recurring subgra
interactions, groups of people

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





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

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

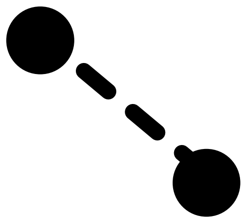
How to generalize to motifs?

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

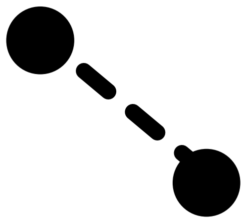
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set



[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

Link prediction:



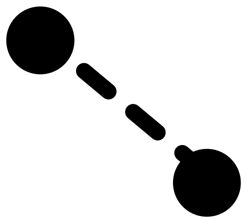
A link is either there or not there

[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

Link prediction:

Motif prediction:

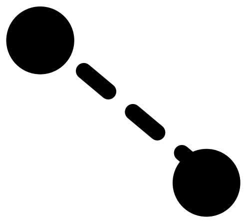


A link is either there or not there

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

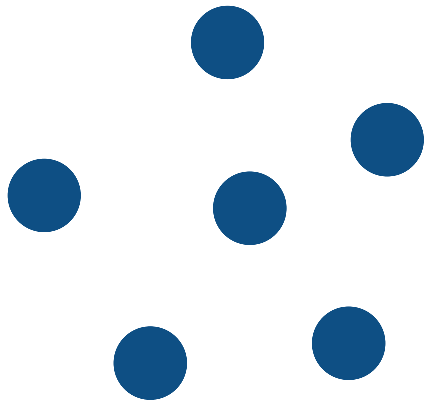
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



A link is either there or not there

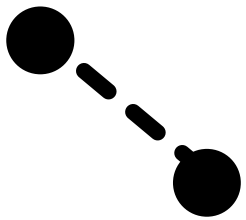
Motif prediction:



[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

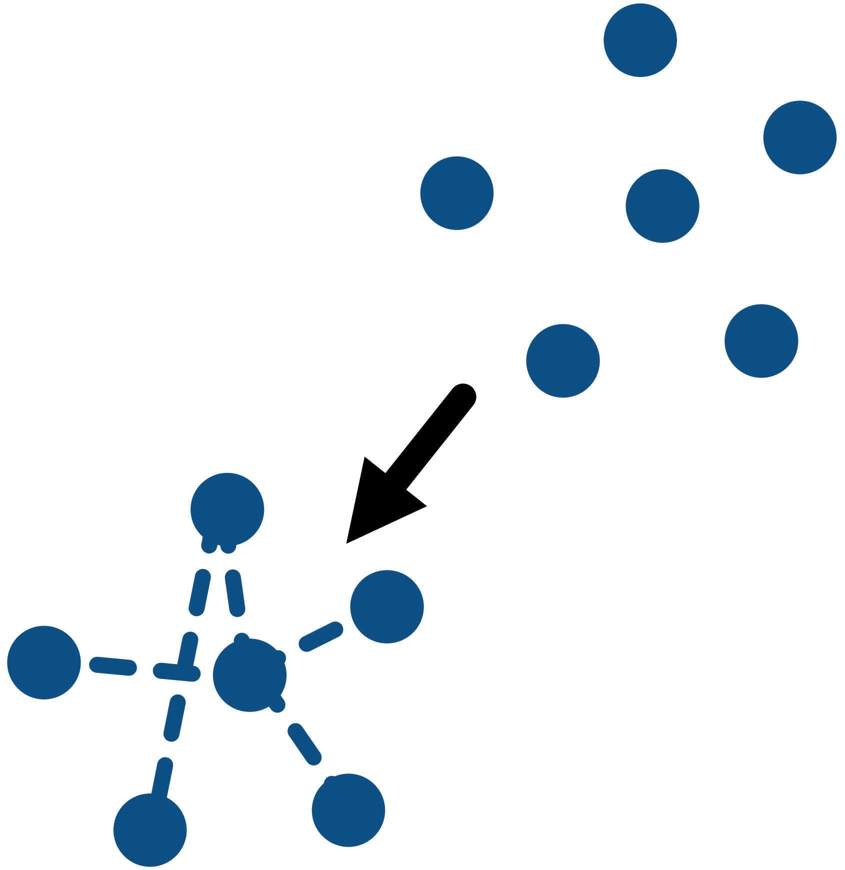
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



A link is either there or not there

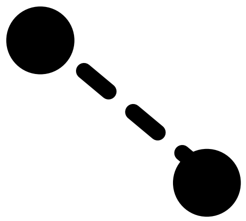
Motif prediction:



[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

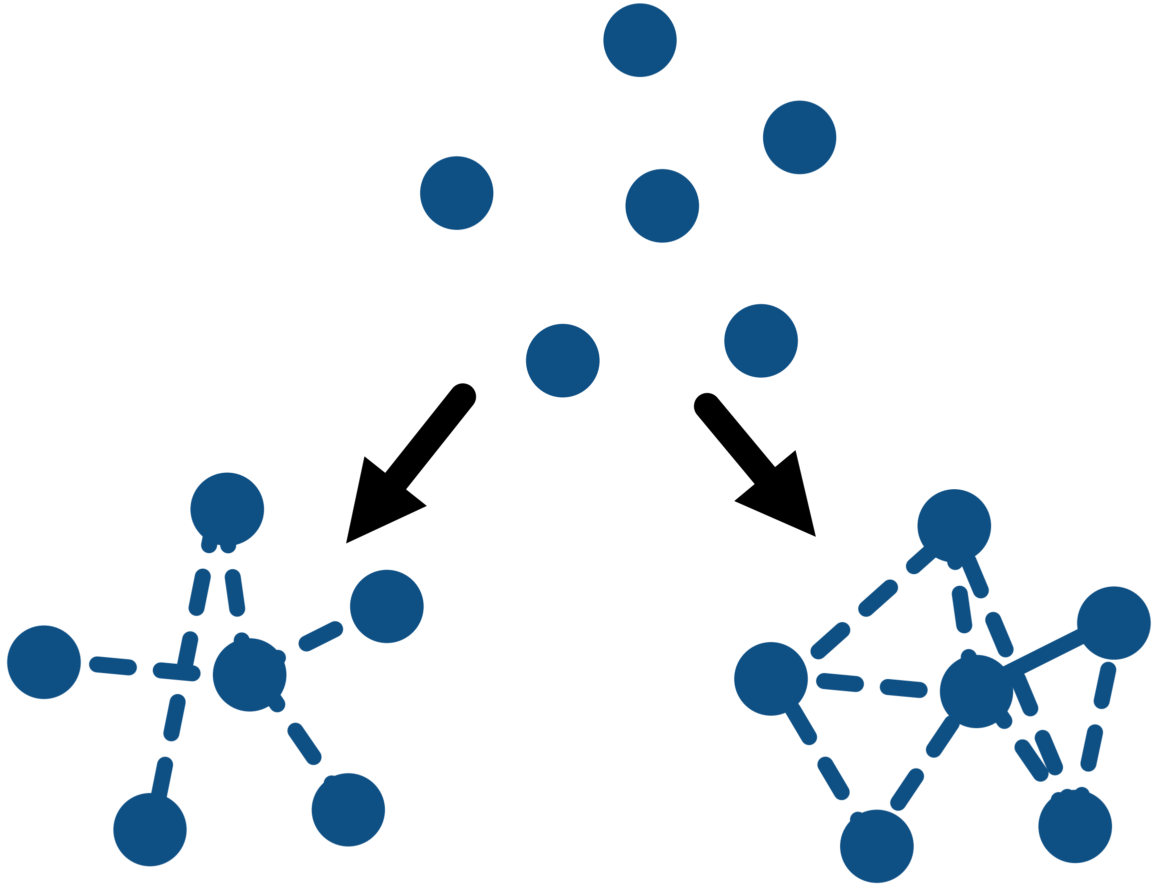
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



A link is either there or not there

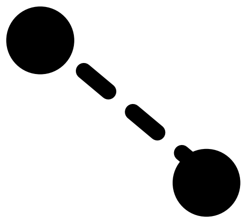
Motif prediction:



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

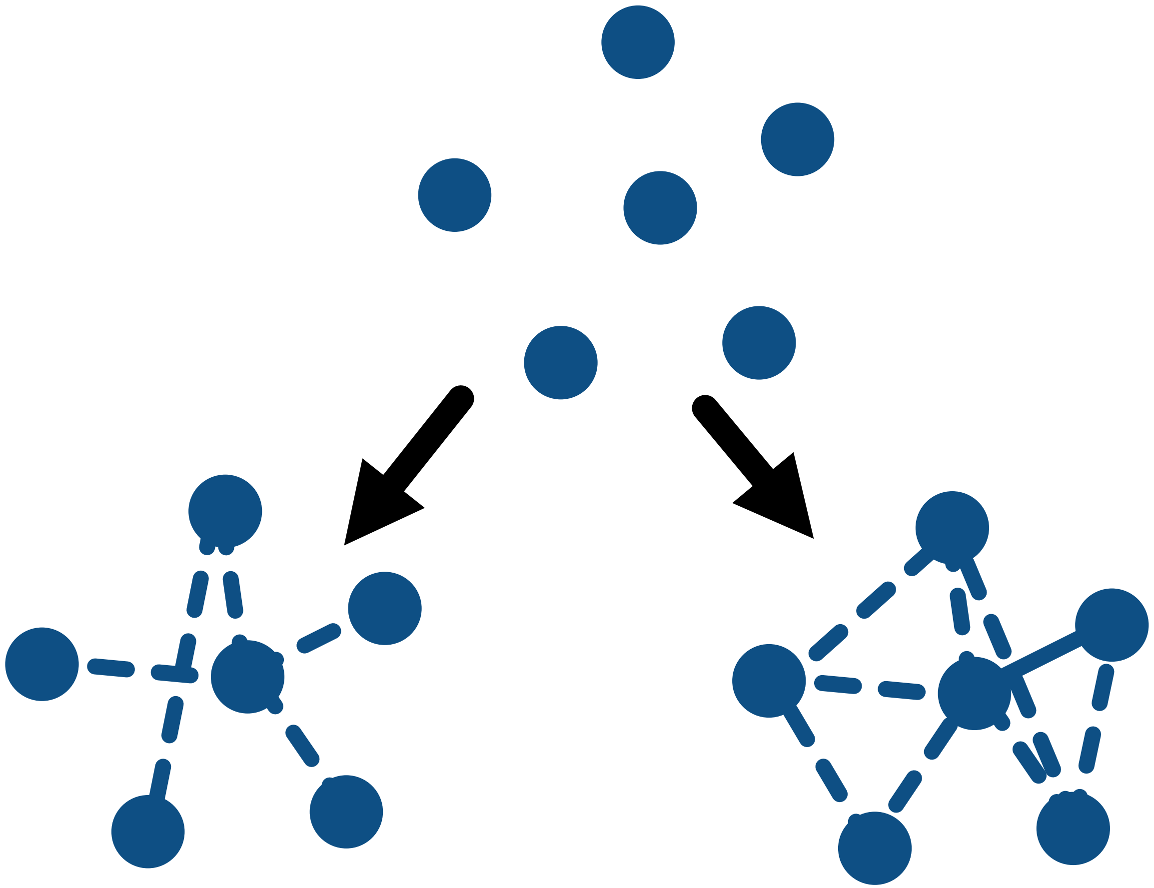
Difference 1: There May Be Many Potential New Motifs for a Fixed Vertex Set

Link prediction:



A link is either there or not there

Motif prediction:

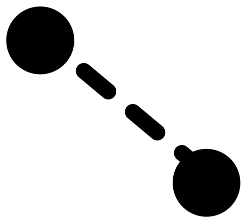


How to consider such diversity of possible patterns in score functions?

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:

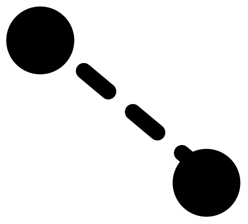


Motif prediction:

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



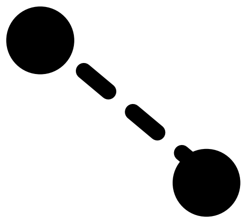
A link to be predicted does not exist

Motif prediction:

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

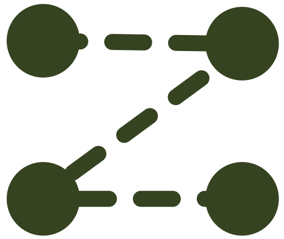
Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



A link to be predicted does not exist

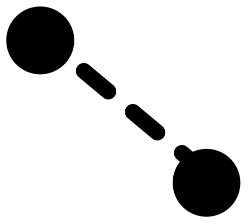
Motif prediction:



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

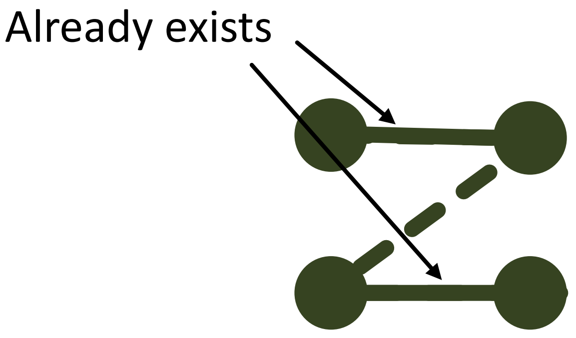
Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



A link to be predicted does not exist

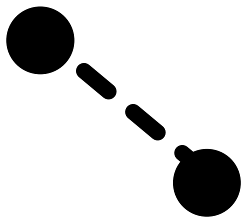
Motif prediction:



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

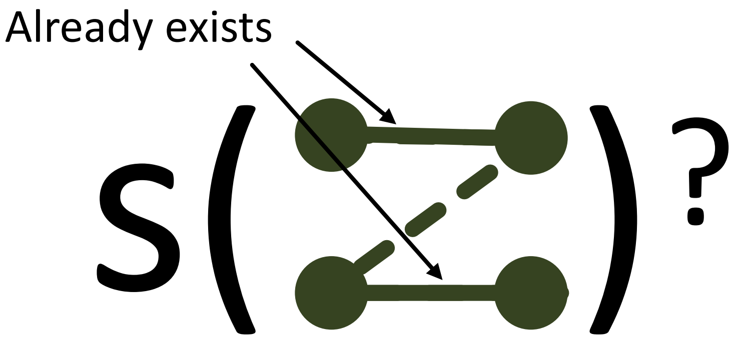
Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



A link to be predicted does not exist

Motif prediction:

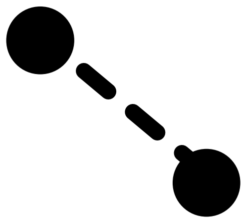


How to consider such edges in the score functions?

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

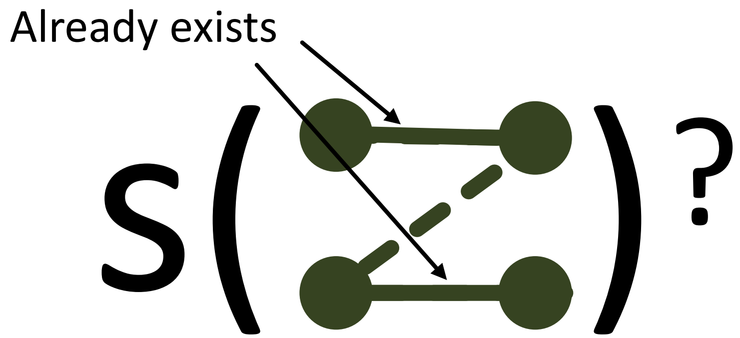
Difference 2: Incoming Motifs May Have Existing Edge

Link prediction:



A link to be predicted does not exist

Motif prediction:



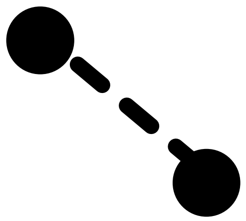
How to consider such edges in the score functions?

Example: some existing relationships in a group of people

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

Difference 3: There May Be "Deal-Breaker" Edges

Link prediction:

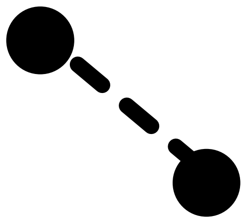


Motif prediction:

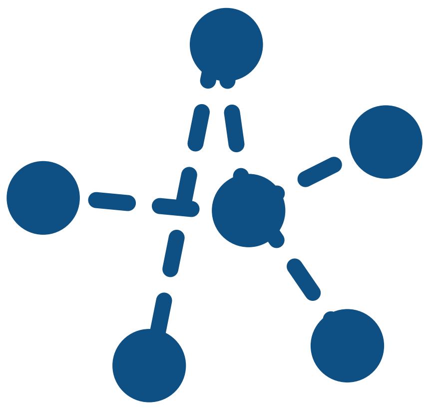
[1] M. Besta et al.: "Motif prediction with graph neural networks", KDD'22

Difference 3: There May Be "Deal-Breaker" Edges

Link prediction:



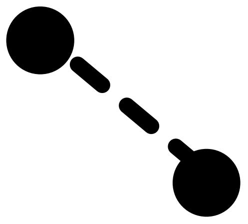
Motif prediction:



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

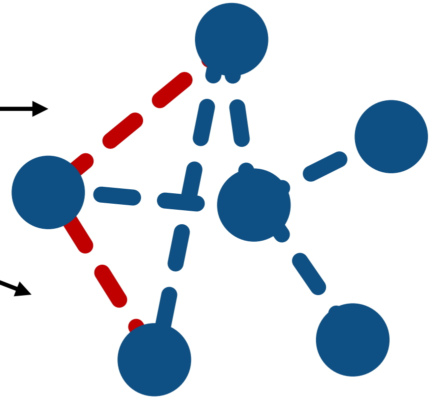
Difference 3: There May Be "Deal-Breaker" Edges

Link prediction:



Motif prediction:

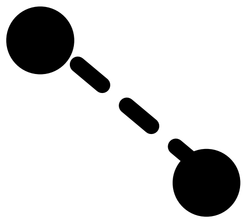
We don't want these links!
(i.e., these links appearing
would make it impossible for
a motif in question to appear



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

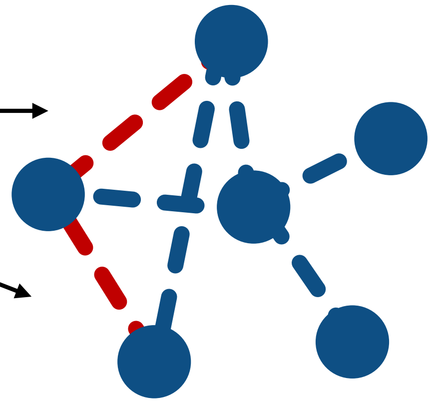
Difference 3: There May Be “Deal-Breaker” Edges

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

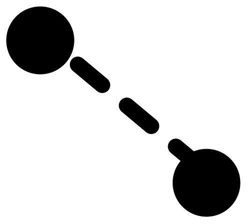


Example:
chemical bonds

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

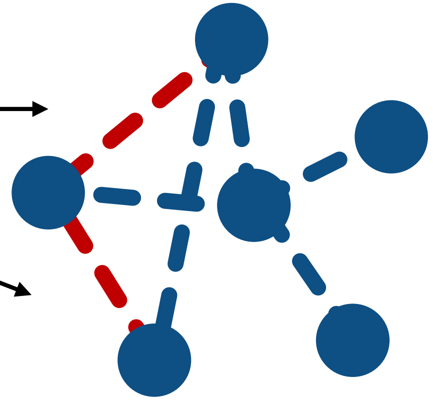
Difference 3: There May Be “Deal-Breaker” Edges

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



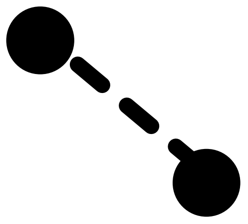
How to consider such edges in the score functions?

Example: chemical bonds

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Difference 3: There May Be "Deal-Breaker" Edges

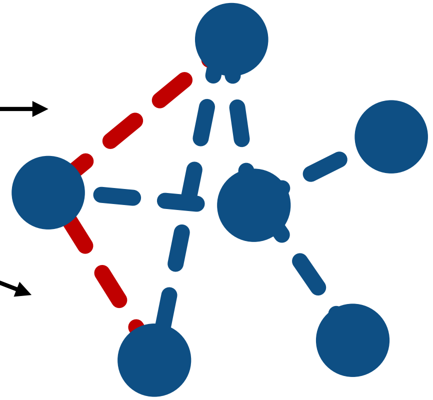
Link prediction:



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



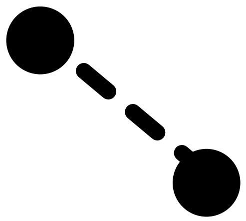
How to consider such edges in the score functions?

Example: chemical bonds

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

Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:

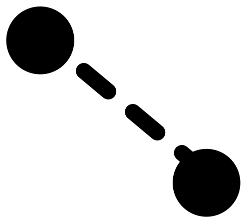


Motif prediction:

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

Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:



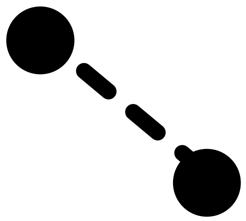
Motif prediction:

We want **this**:

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

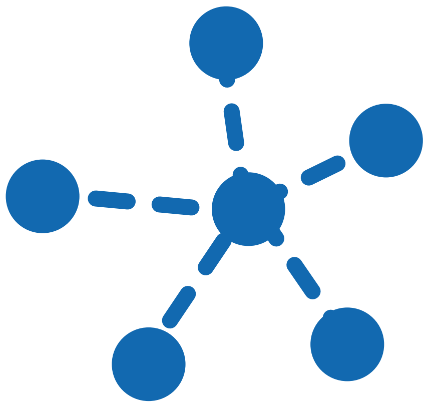
Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:



Motif prediction:

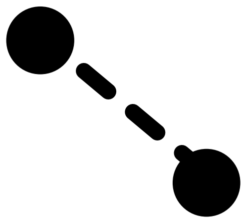
We want **this**:



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

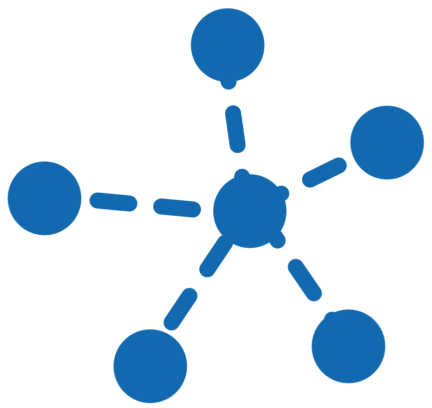
Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:

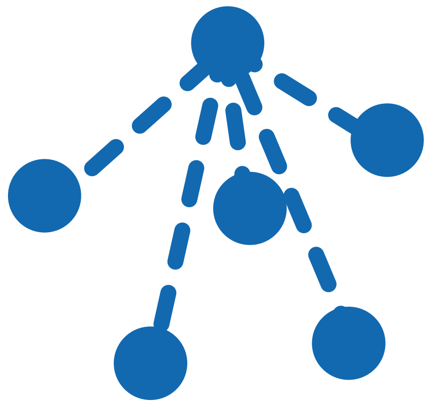


Motif prediction:

We want **this**:



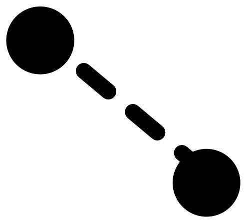
...but **not** this:



[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Difference 4: Motif Prediction Query May Depend on Vertex Labeling

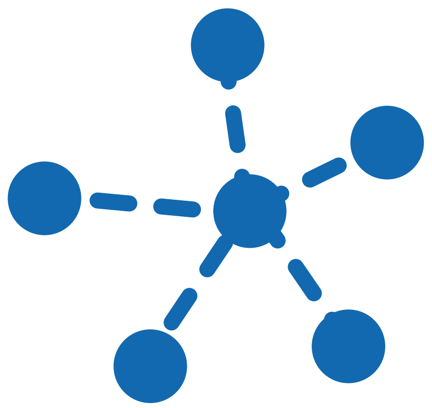
Link prediction:



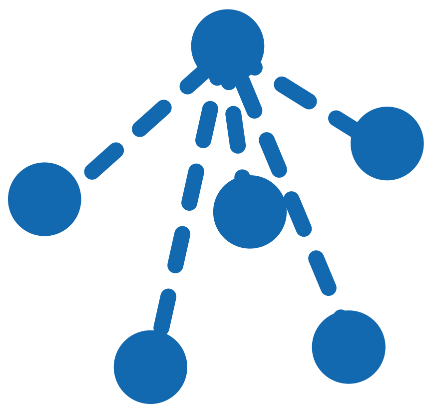
No such effect (not enough room with a single link)

Motif prediction:

We want **this**:



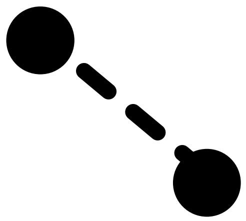
...but **not** this:



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

Difference 4: Motif Prediction Query May Depend on Vertex Labeling

Link prediction:

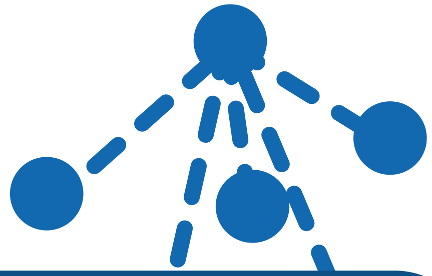


No such effect (not enough room with a single link)

Motif prediction:

We want **this**:

...but **not** this:



How to formulate motif prediction, considering all these differences?

[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

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Motif Prediction Score Functions

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

Starting Simple: Motif Scores Based on Independent Links

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Motif Prediction Score Functions

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

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

Starting Simple: Motif Scores Based on Independent Links

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Motif Prediction Score Functions

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

Starting Simple: Motif Scores Based on Independent Links

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

Motif edges that do **not** yet exist

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

Motif Prediction Score Functions

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

Starting Simple: Motif Scores Based on Independent Links

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

Motif edges that do **not** yet exist

Motif edges that **already** exist

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

Motif Prediction Score Functions

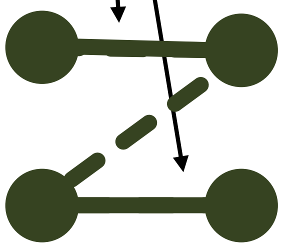
A motif: $M = (V_M, E_M)$

Starting Simple: Motif Scores Based on Independent Links

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

Motif edges that do **not** yet exist

Motif edges that **already** exist



[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Motif Prediction Score Functions

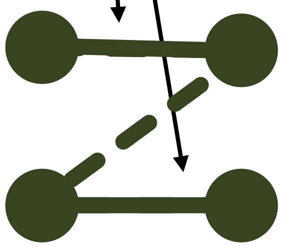
A motif: $M = (V_M, E_M)$

Starting Simple: Motif Scores Based on Independent Links

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

Motif edges that do **not** yet exist

Motif edges that **already** exist



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

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

Motif Prediction Score Functions

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

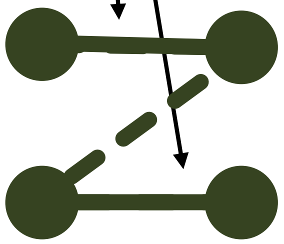
Starting Simple: Motif Scores Based on Independent Links

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

Motif edges that do **not** yet exist

Motif edges that **already** exist

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



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

Motif Prediction Score Functions

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

Starting Simple: Motif Scores Based on Independent Links

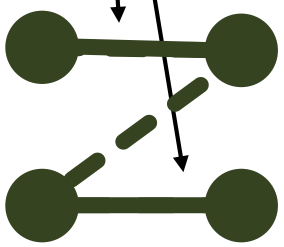
$$E_M = E_{M,\mathcal{N}} \cup E_{M,\mathcal{E}}$$

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

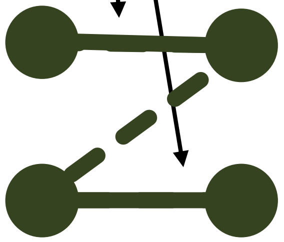
A motif: $M = (V_M, E_M)$

Starting Simple: Motif Scores Based on Independent Links

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

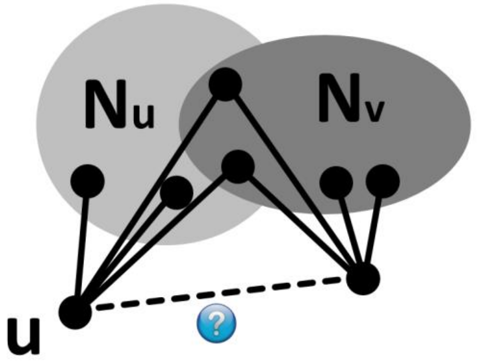
Motif edges that do **not** yet exist

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

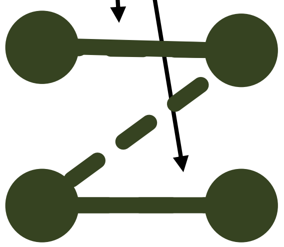
Motif Prediction Score Functions

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

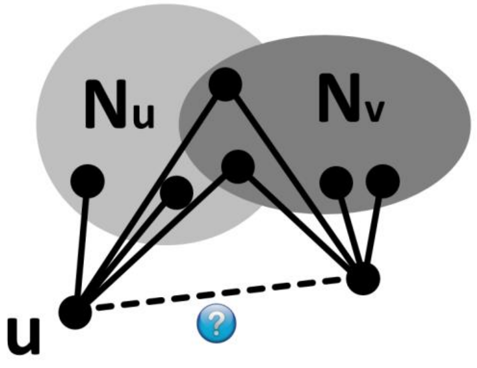


Starting Simple: Motif Scores Based on Independent Links

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)



[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

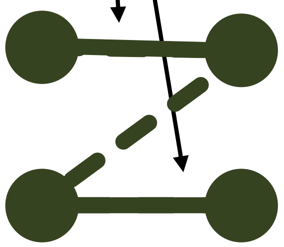
Motif Prediction Score Functions

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



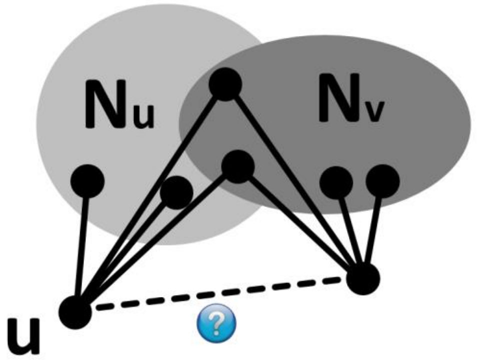
Starting Simple: Motif Scores Based on Independent Links

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)

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

∈ [0, 1] by construction

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

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

Starting Simple: Motif Scores Based on Independent Links

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

Motif edges that do **not** yet exist

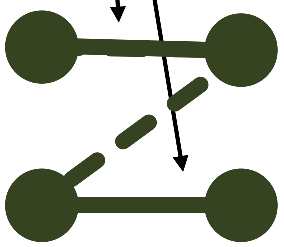
Motif edges that **already** exist

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)

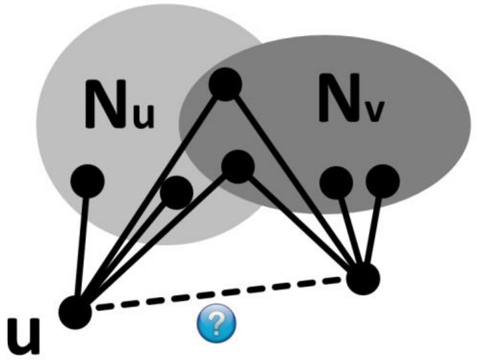
$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

∈ [0, 1] by construction

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



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



[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Motif Prediction Score Functions

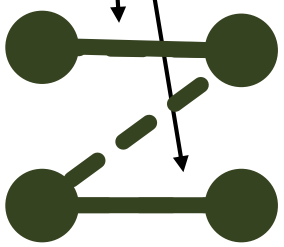
A motif: $M = (V_M, E_M)$

Starting Simple: Motif Scores Based on Independent Links

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

Motif edges that do **not** yet exist

Motif edges that **already** exist



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)

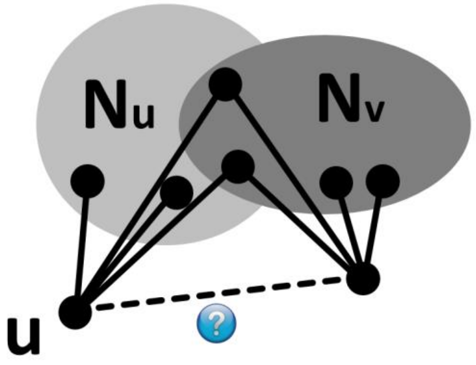
$$s_{\perp}(M) = \prod_{e \in E_{M,\mathcal{N}}} s(e)$$

∈ [0, 1] by construction

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

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

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



[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

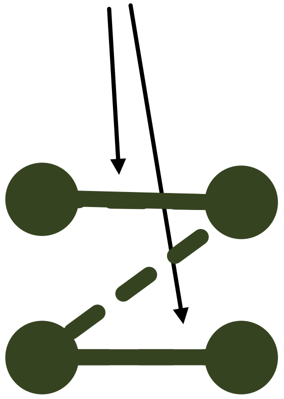
Motif Prediction Score Functions

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:

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

Motif Prediction Score Functions

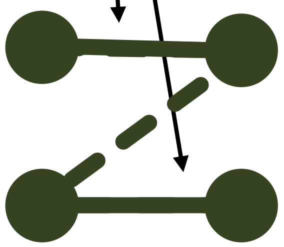
A motif: $M = (V_M, E_M)$

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

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

Motif edges that do **not** yet exist

Motif edges that **already** exist



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

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

Motif Prediction Score Functions

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

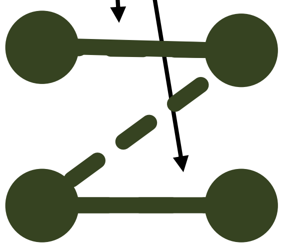
What if the arrival of some (motif) links impacts the chances for other motif links to appear?

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

Motif edges that do **not** yet exist

Motif edges that **already** exist

Adding correlation between links



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

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

Motif Prediction Score Functions

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

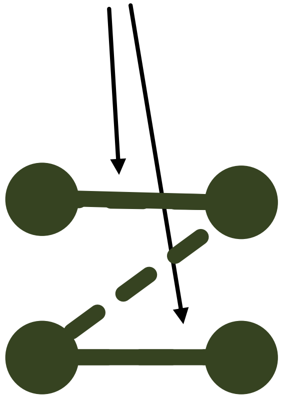
What if the arrival of some (motif) links impacts the chances for other motif links to appear?

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

Motif edges that do **not** yet exist

Motif edges that **already** exist

Adding correlation between links



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

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

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

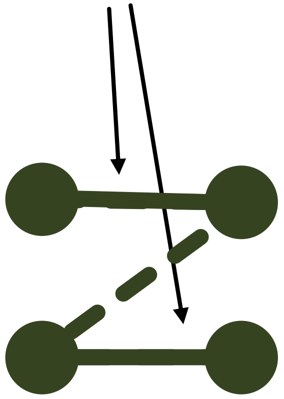
Motif Prediction Score Functions

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



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

Adding correlation between links

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

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:

[1] M. Besta et al.: “Motif prediction with graph neural networks”, KDD’22

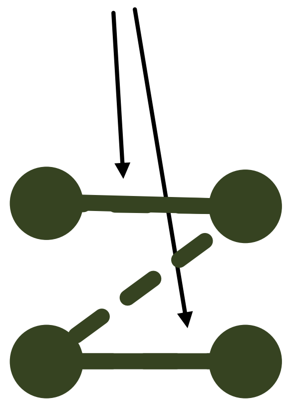
Motif Prediction Score Functions

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



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)

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

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:

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

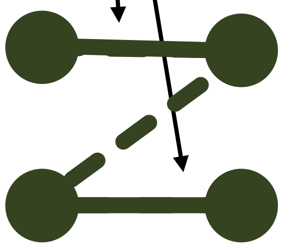
Motif Prediction Score Functions

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



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)

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

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:

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

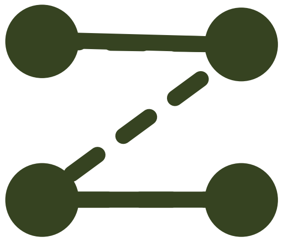
[1] M. Besta et al.: "Motif prediction with graph neural networks", KDD'22

Motif Prediction Score Functions

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

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

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



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

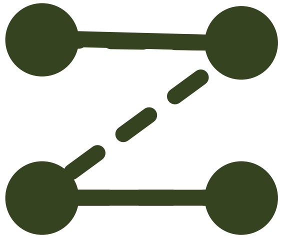
Motif Prediction Score Functions

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

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

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

Incorporating deal-breaker links



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

Motif Prediction Score Functions

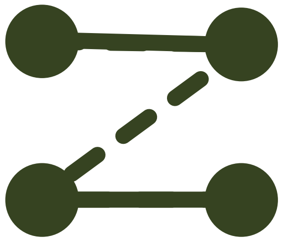
A motif: $M = (V_M, E_M)$

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

Incorporating deal-breaker links

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

$E_M^* = E_M$



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

Motif Prediction Score Functions

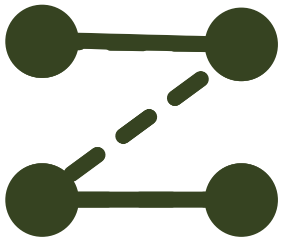
A motif: $M = (V_M, E_M)$

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

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

Incorporating deal-breaker links

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



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

Motif Prediction Score Functions

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

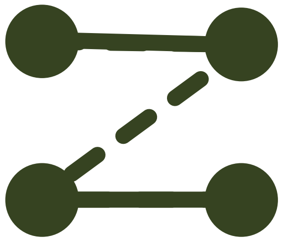
What if the arrival of some (motif) links reduces, or even prevents, motif's appearance?

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

Incorporating deal-breaker links

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

Motif edges



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

Motif Prediction Score Functions

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

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

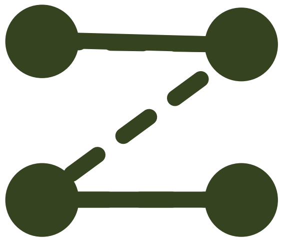
Incorporating deal-breaker links

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

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

Motif edges

Deal-breaker edges



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

Motif Prediction Score Functions

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

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

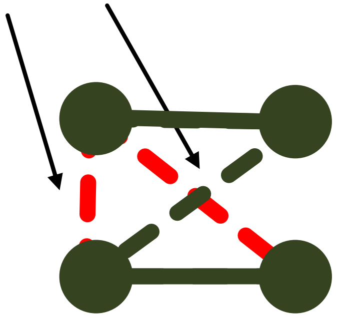
Incorporating deal-breaker links

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

$$E^*_M = \boxed{E_M} \cup \overline{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges



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

Motif Prediction Score Functions

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

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

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

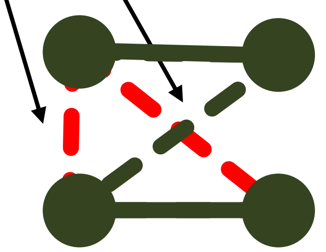
Incorporating deal-breaker links

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

Motif edges

Deal-breaker edges

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



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

Motif Prediction Score Functions

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

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

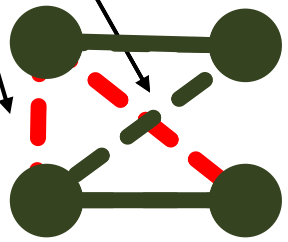
Incorporating deal-breaker links

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

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

Motif edges

Deal-breaker edges



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

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

Motif Prediction Score Functions

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

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

Incorporating deal-breaker links

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

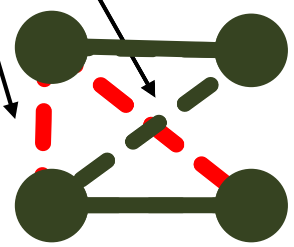
$$E^*_M = E_M \cup \overline{E}_{M,\mathcal{D}}$$

Motif edges

Deal-breaker edges

The weight vector (incorporates user's domain knowledge)

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



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

Motif Prediction Score Functions

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

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

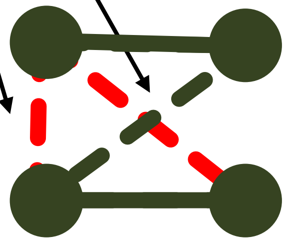
Incorporating deal-breaker links

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

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

Motif edges

Deal-breaker edges



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):

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

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

Motif Prediction Score Functions

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

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

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

Incorporating deal-breaker links

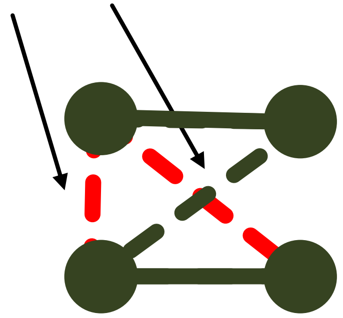
$E_M^* = \boxed{E_M} \cup \overline{E}_{M,\mathcal{D}}$

Motif edges

Deal-breaker edges

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:

$$\mathbf{s}^*(\mathbf{e}) = \mathbf{0}$$

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

Motif Prediction Score Functions

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

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

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

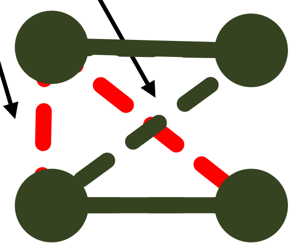
Incorporating deal-breaker links

$E^*_M = \boxed{E_M} \cup \overline{E}_{M,\mathcal{D}}$

Motif edges

Deal-breaker edges

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:

$$\mathbf{s}^*(\mathbf{e}) = \mathbf{0}$$

Another formulation (perhaps more intuitive):

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

Motif Prediction Score Functions

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

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

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

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

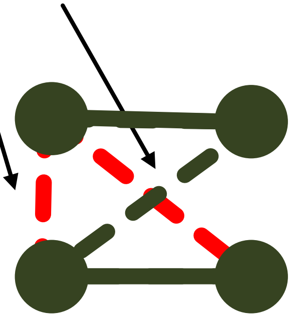
Motif edges

Deal-breaker edges

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:

$$\mathbf{s}^*(\mathbf{e}) = \mathbf{0}$$

Another formulation (perhaps more intuitive):

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

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

Motif Prediction Score Functions

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

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

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

Motif edges

Deal-breaker edges

re

In

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

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:

$$\mathbf{s}^*(\mathbf{e}) = 0$$

Another formulation (perhaps more intuitive):

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

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

Motif Prediction Score Functions

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

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

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

Motif edges

Deal-breaker edges

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

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

A deal-breaker edge

$s_i^*(e) =$

Another formul

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?**

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

[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 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?

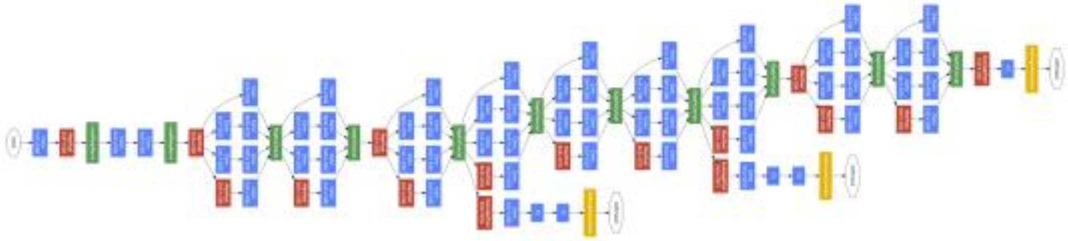
Samples

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

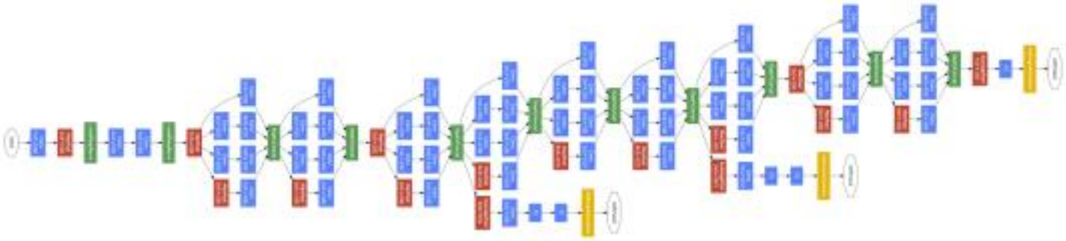


The animation borrowed from T. Hoefler

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

How does deep learning work?

Samples



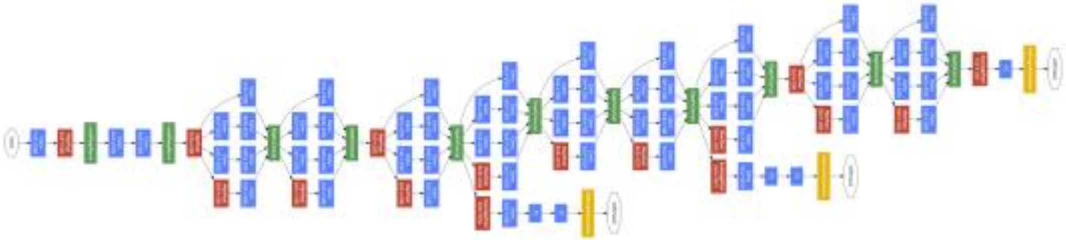
The animation borrowed from T. Hoefler

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

How does deep learning work?

Samples

f(x)



The animation borrowed from T. Hoefler

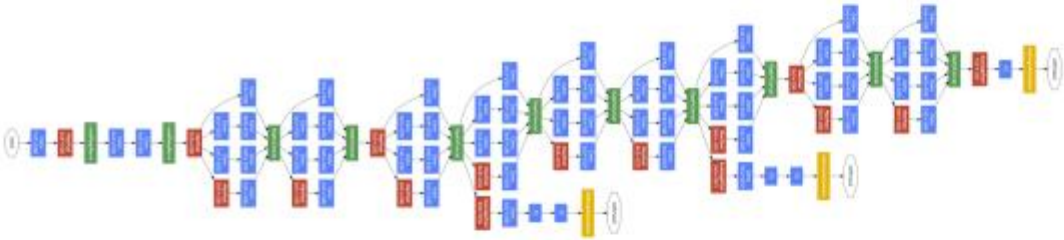
The graph structure may be arbitrary, maybe one could arrive at better heuristics by **learning**?

How does deep learning work?

Samples

Learning a heuristic

f(x)



The animation borrowed from T. Hoefler

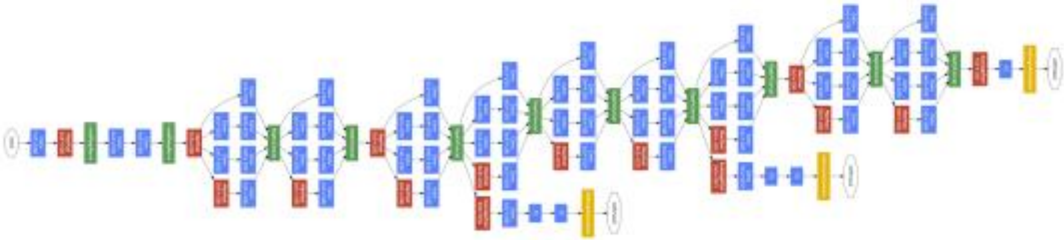
The graph structure may be arbitrary, maybe one could arrive at better heuristics by **learning**?

How does deep learning work?

Samples

Learning a heuristic

f(x)



Cat	0.54
Dog	0.28
Airplane	0.07
	0.04
Horse	0.33
Bicycle	0.02
Truck	0.02

The animation borrowed from T. Hoefler

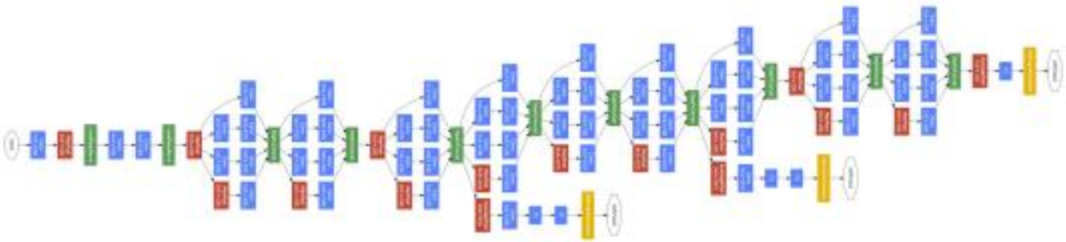
The graph structure may be arbitrary, maybe one could arrive at better heuristics by **learning**?

How does deep learning work?

Samples

Learning a heuristic

f(x)



Cat	0.54
Dog	0.28
Airplane	0.07
Horse	0.04
Bicycle	0.33
Truck	0.02
	0.02

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

The animation borrowed from T. Hoefler

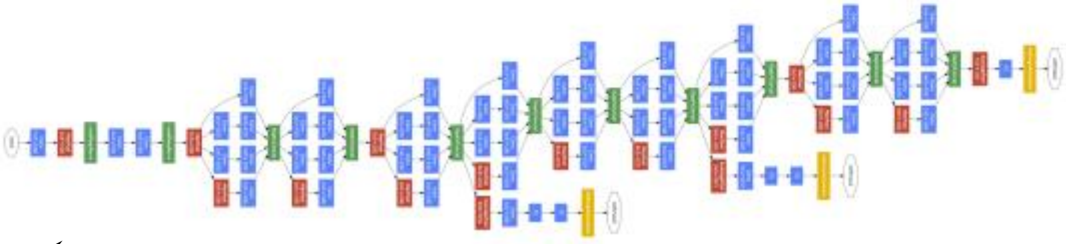
The graph structure may be arbitrary, maybe one could arrive at better heuristics by **learning**?

How does deep learning work?

Samples

Learning a heuristic

f(x)



layer-wise weight update

Cat	0.54
Dog	0.28
Airplane	0.07
Horse	0.04
Bicycle	0.33
Truck	0.02
	0.02

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

The animation borrowed from T. Hoefler

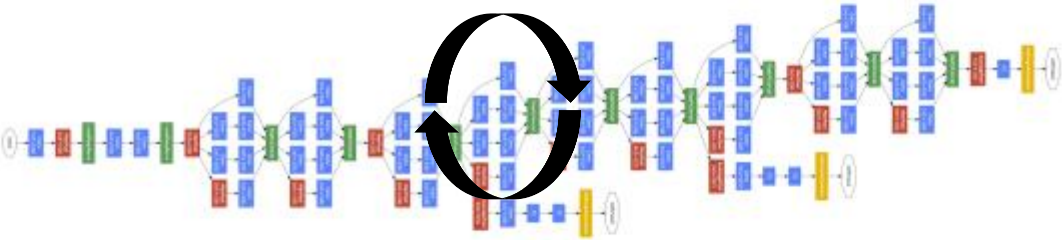
The graph structure may be arbitrary, maybe one could arrive at better heuristics by **learning**?

How does deep learning work?

Samples

Learning a heuristic

f(x)



layer-wise weight update

Cat	0.54
Dog	0.28
Airplane	0.07
Horse	0.04
Bicycle	0.33
Truck	0.02
	0.02

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

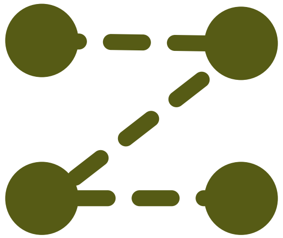
Motif Prediction: Deep Learning Formulation

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

Motif Prediction: Deep Learning Formulation

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

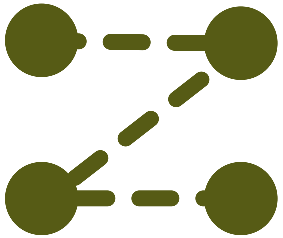
Goal: predict this motif M:

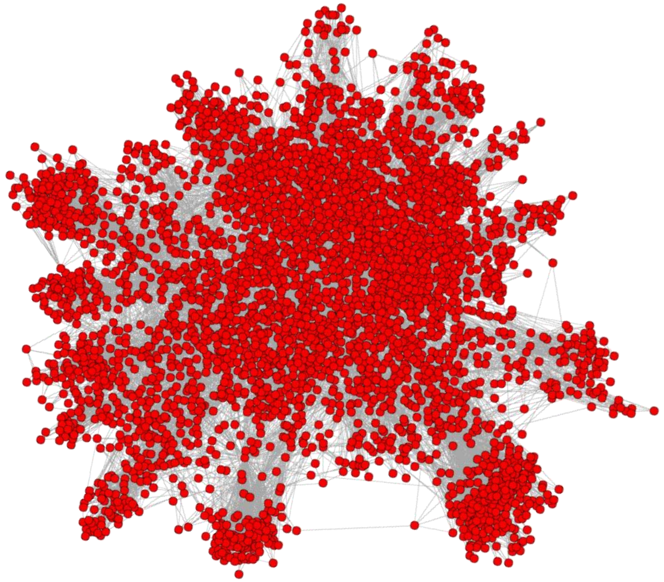


Motif Prediction: Deep Learning Formulation

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

Goal: predict this motif M:

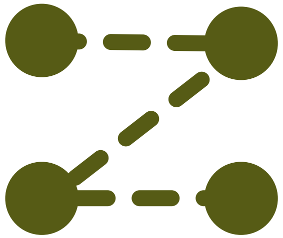




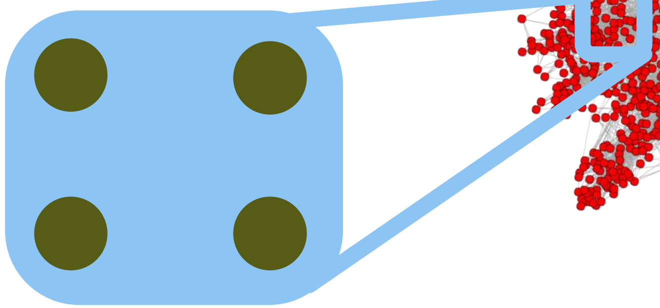
Motif Prediction: Deep Learning Formulation

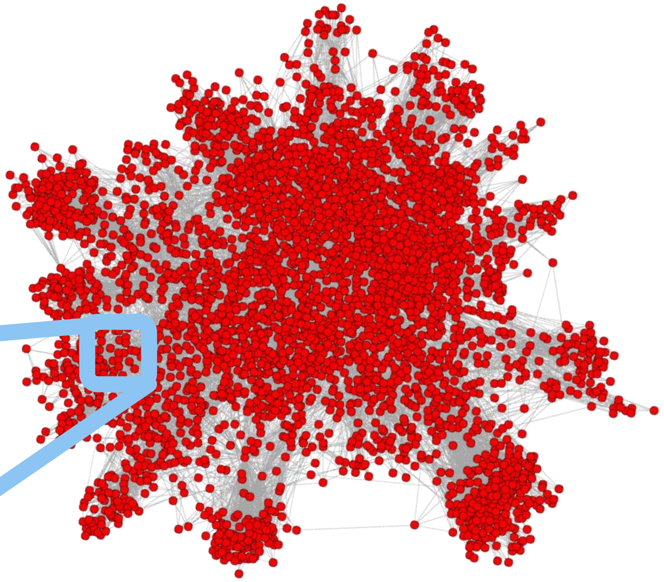
A motif: $M = (V_M, E_M)$

Goal: predict this motif M:



A given fixed set of vertices:

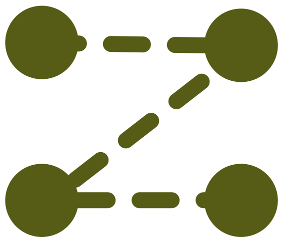




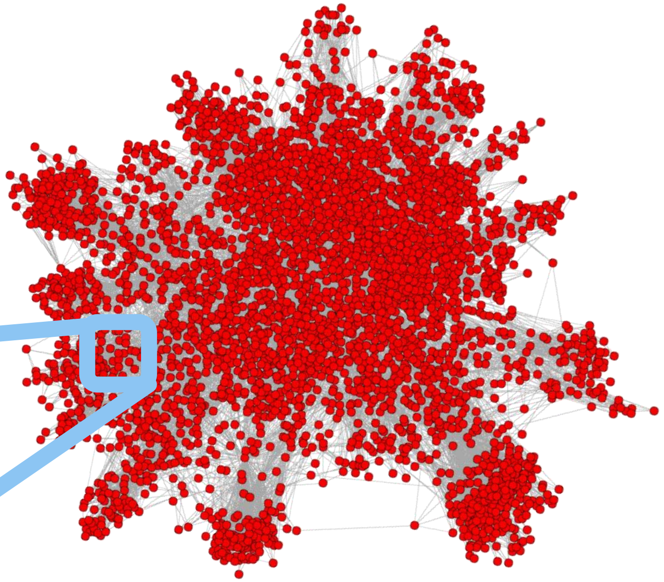
Motif Prediction: Deep Learning Formulation

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

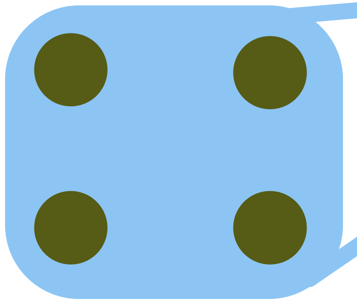
Goal: predict this motif M:



Positive samples

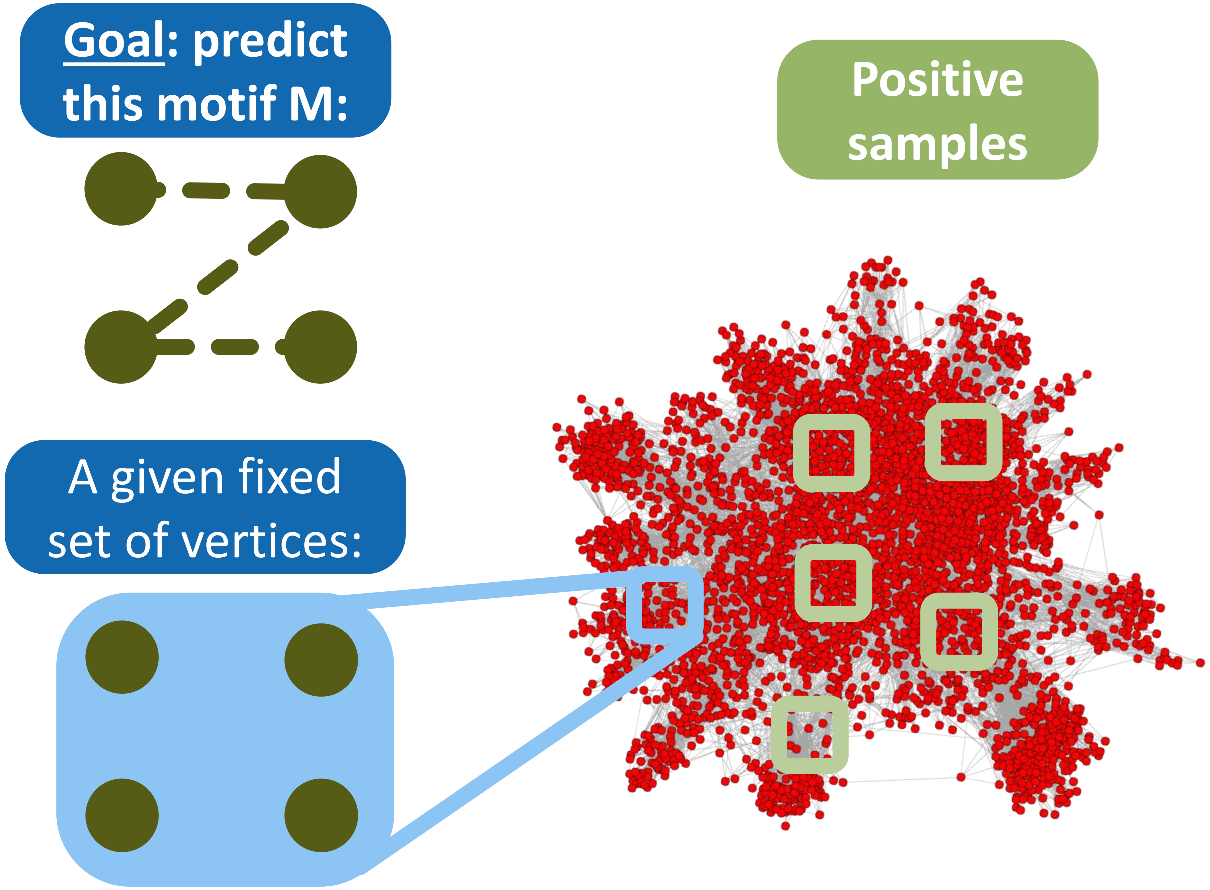


A given fixed set of vertices:



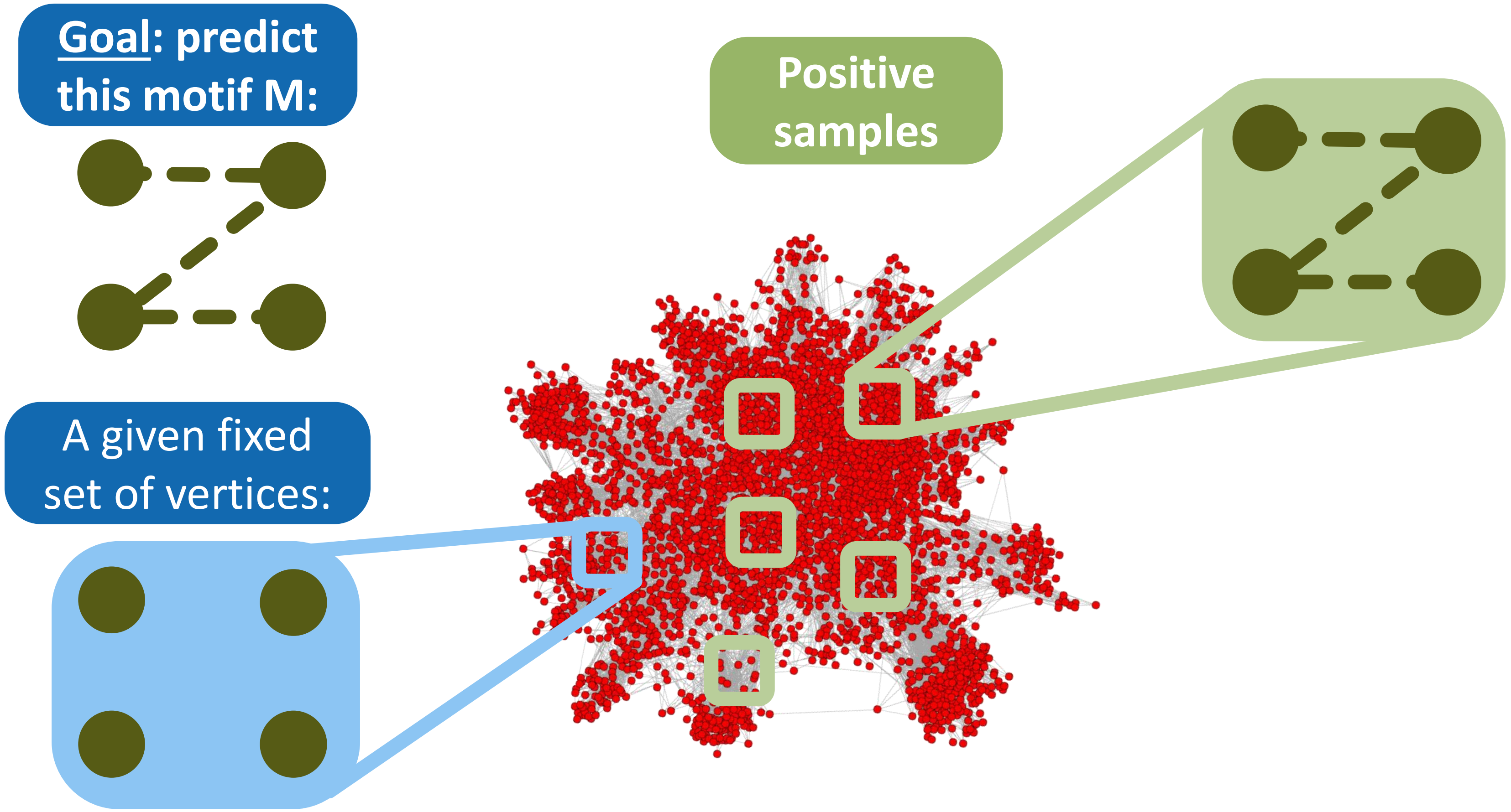
Motif Prediction: Deep Learning Formulation

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



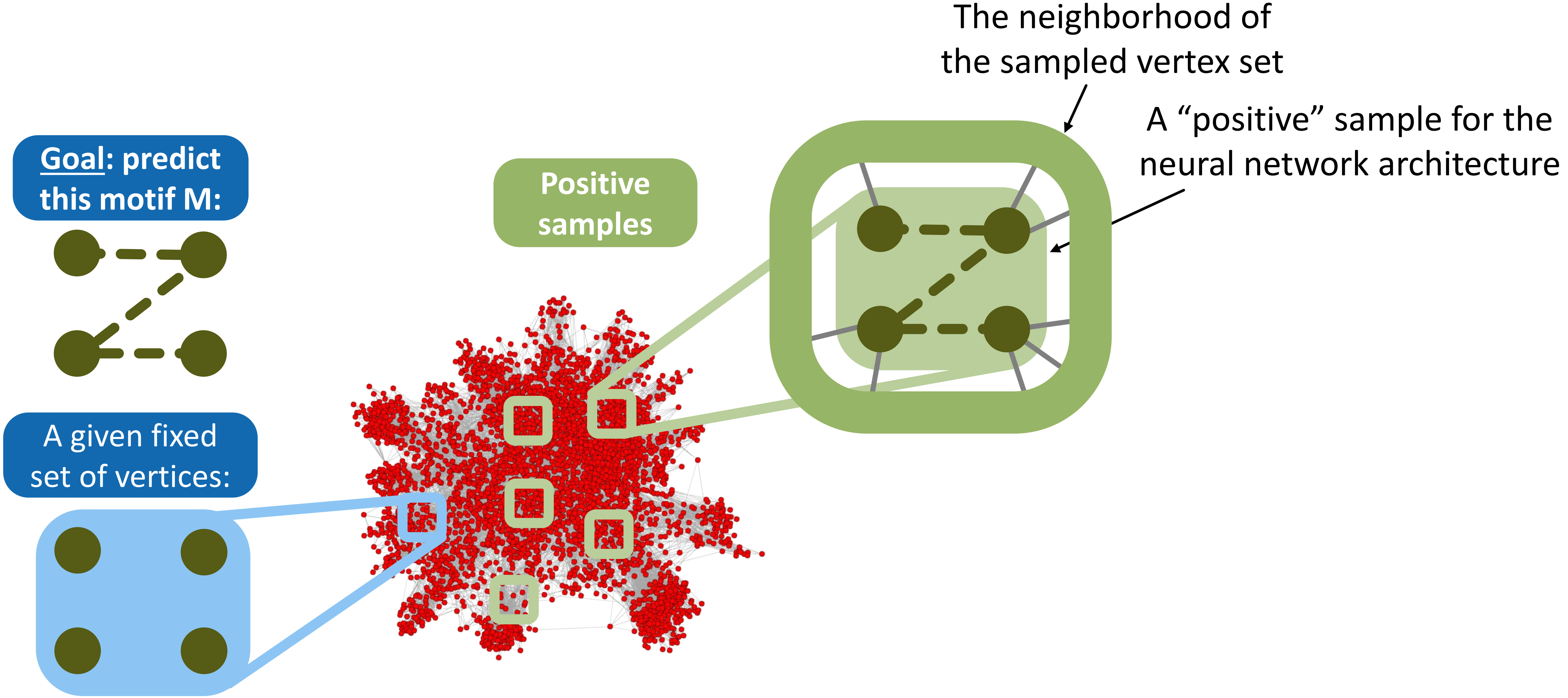
Motif Prediction: Deep Learning Formulation

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



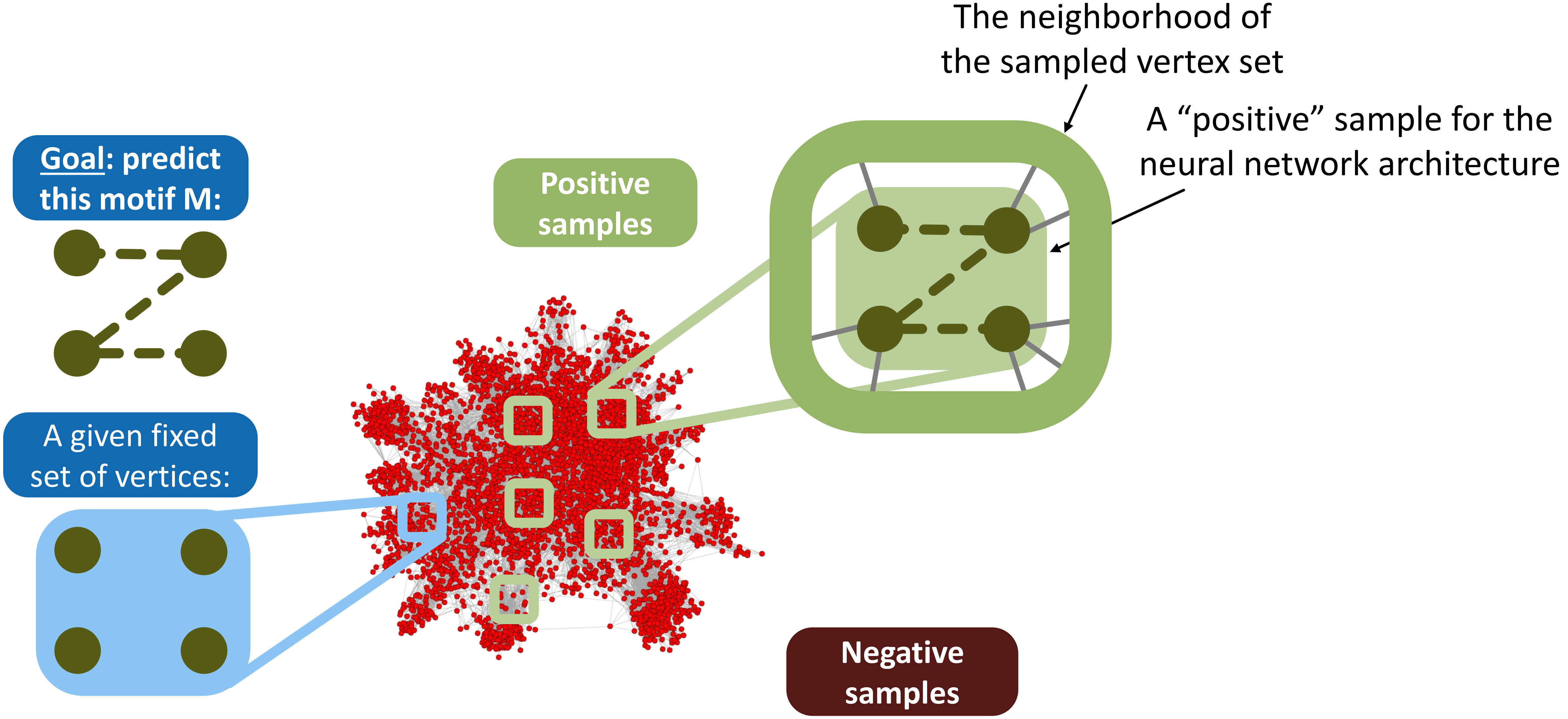
Motif Prediction: Deep Learning Formulation

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



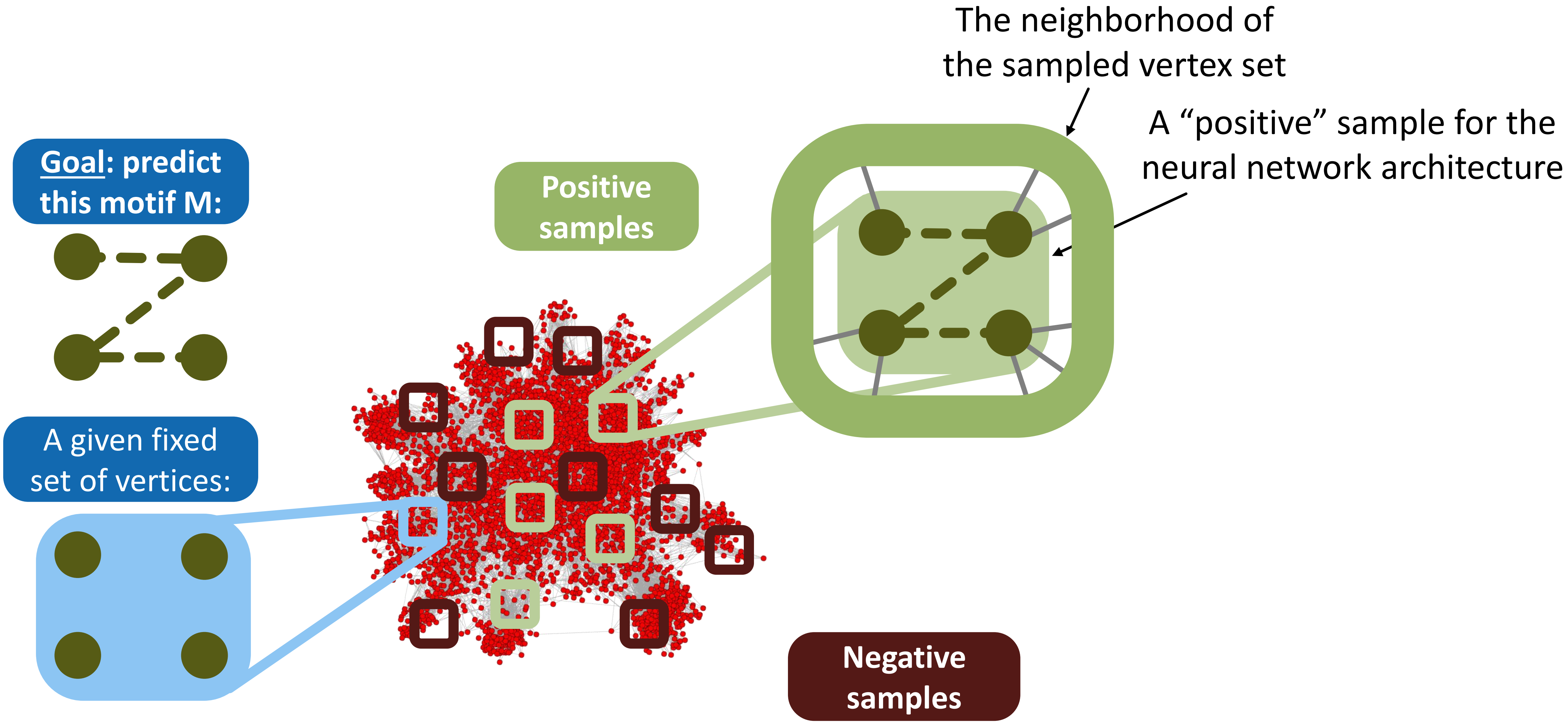
Motif Prediction: Deep Learning Formulation

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



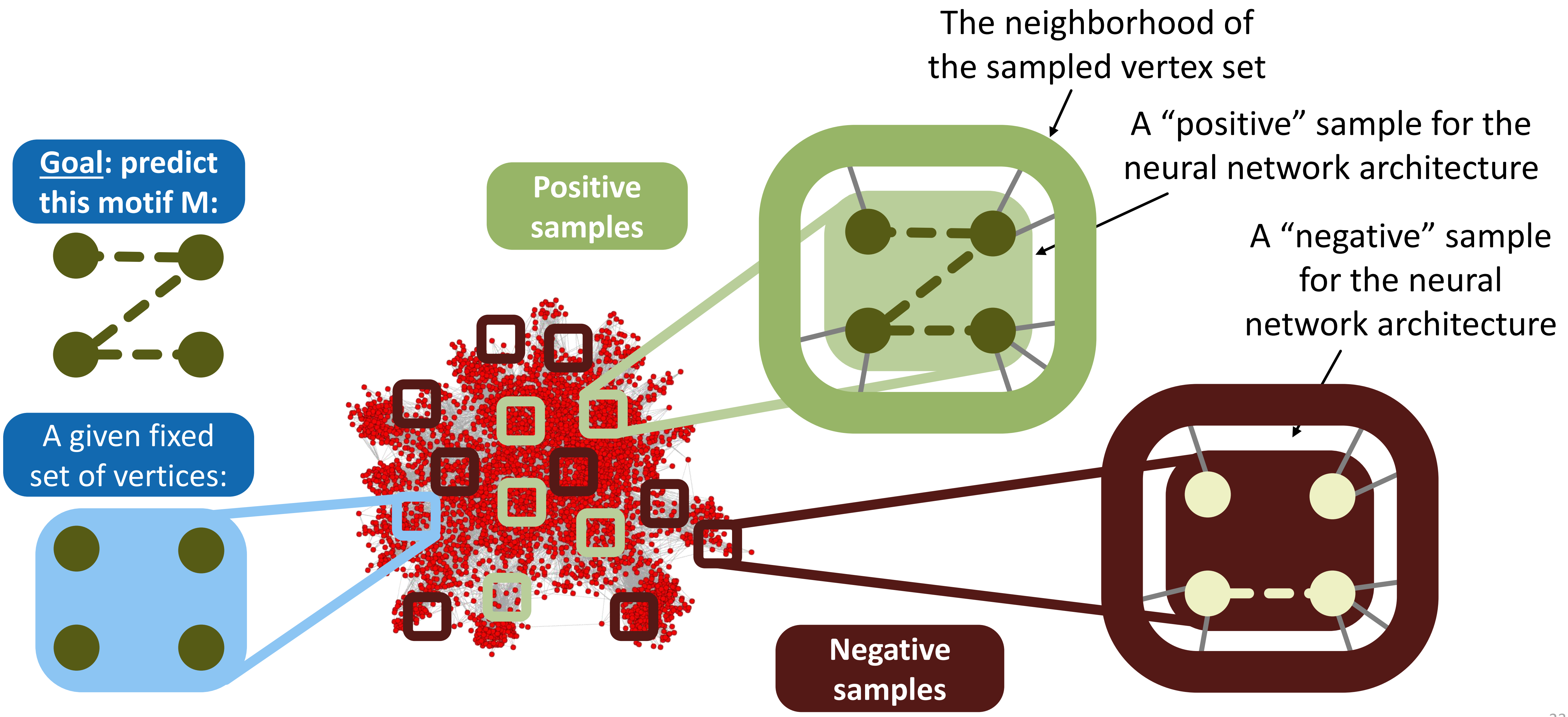
Motif Prediction: Deep Learning Formulation

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

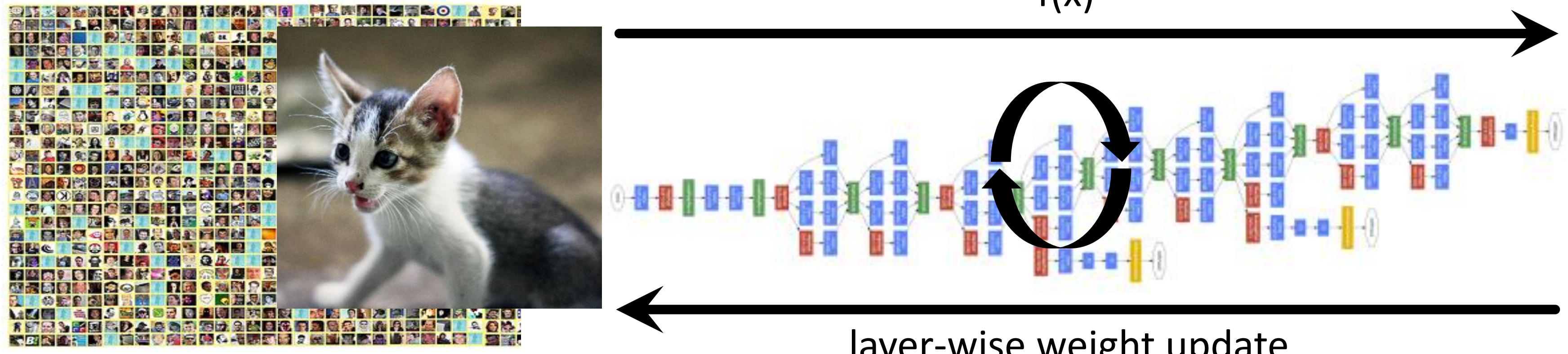


Motif Prediction: Deep Learning Formulation

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

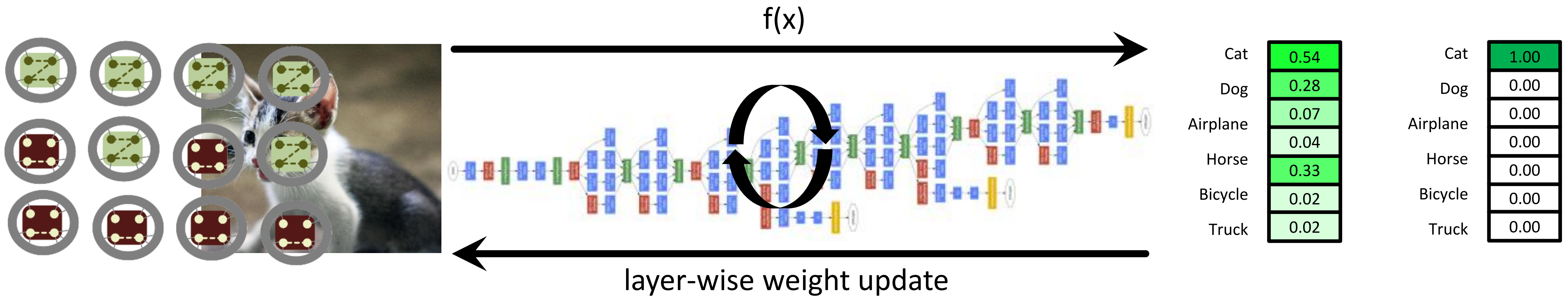


Motif Prediction: Deep Learning Formulation

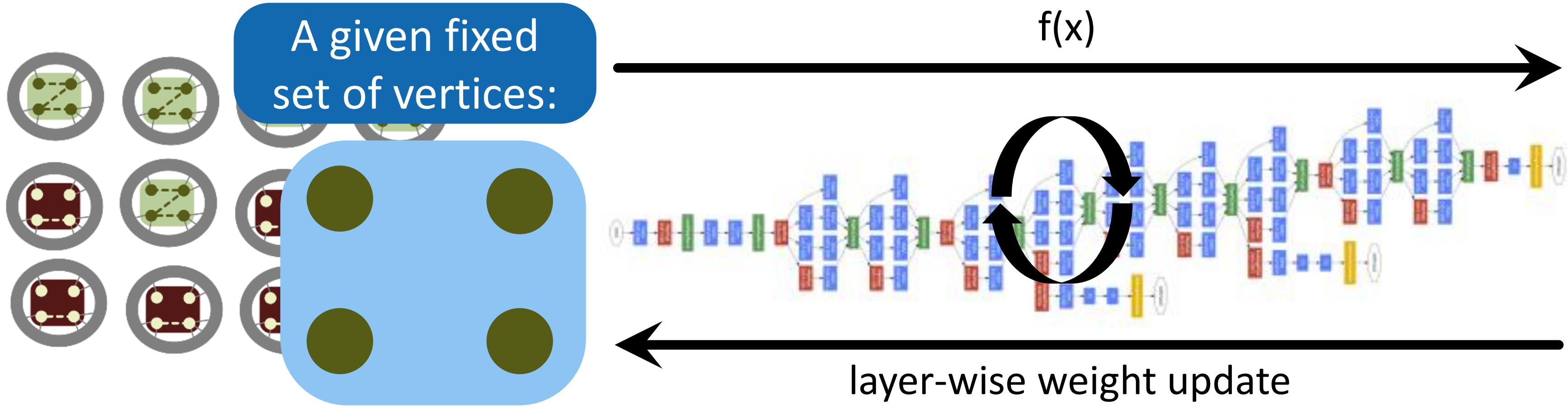


Cat	0.54	1.00
Dog	0.28	0.00
Airplane	0.07	0.00
	0.04	0.00
Horse	0.33	0.00
Bicycle	0.02	0.00
Truck	0.02	0.00

Motif Prediction: Deep Learning Formulation

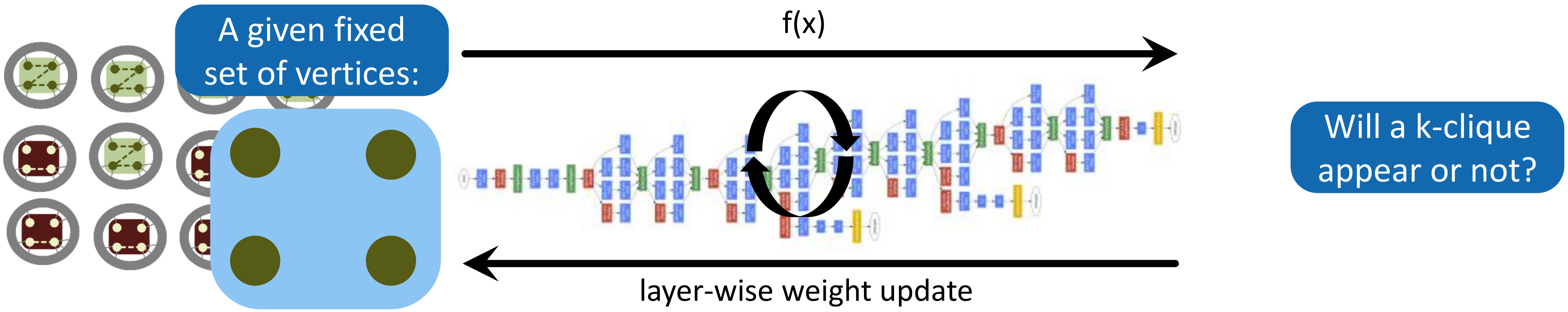


Motif Prediction: Deep Learning Formulation



Cat	0.54	1.00
Dog	0.28	0.00
Airplane	0.07	0.00
	0.04	0.00
Horse	0.33	0.00
Bicycle	0.02	0.00
Truck	0.02	0.00

Motif Prediction: Deep Learning Formulation

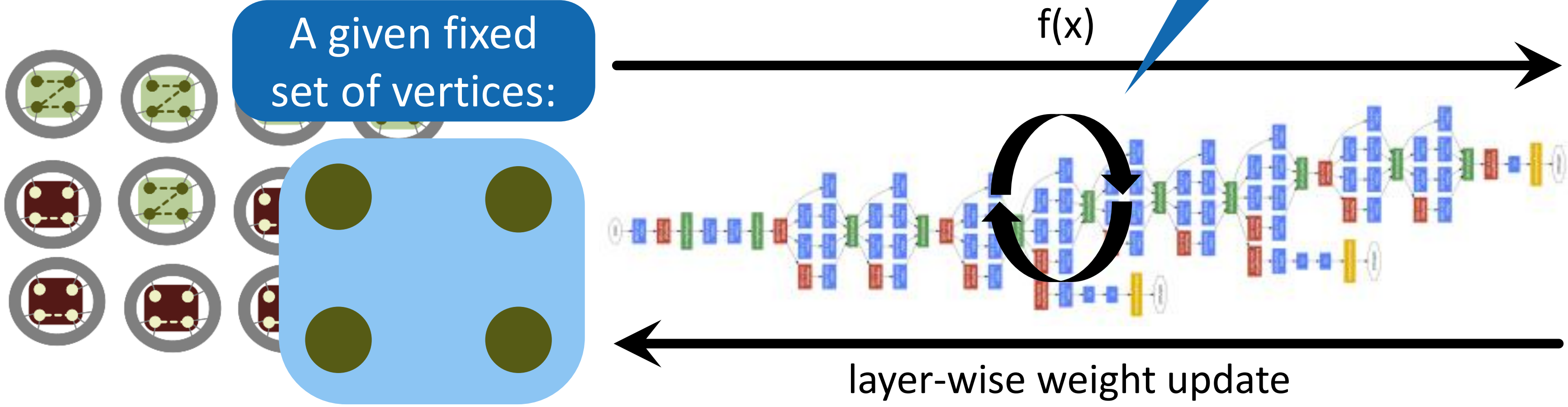


Motif Prediction: Deep Learning Formulation

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

The samples are used to train a neural network model called Deep Graph Convolutional Neural Network (DGCNN) [1]

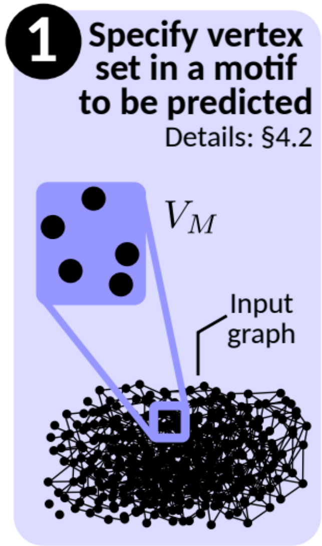
A given fixed set of vertices:



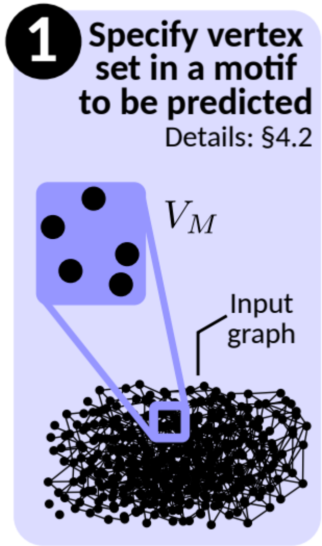
Will a k-clique appear or not?

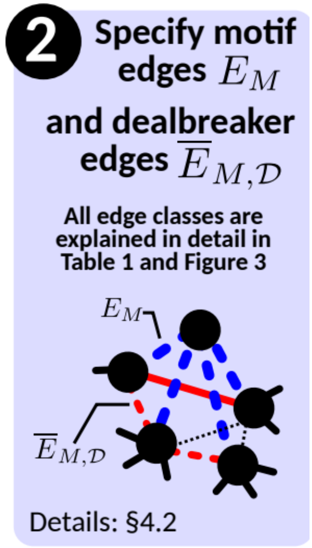
SEAM Architecture (learning from Subgraphs, Embeddings and Attributes for Motif prediction)

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

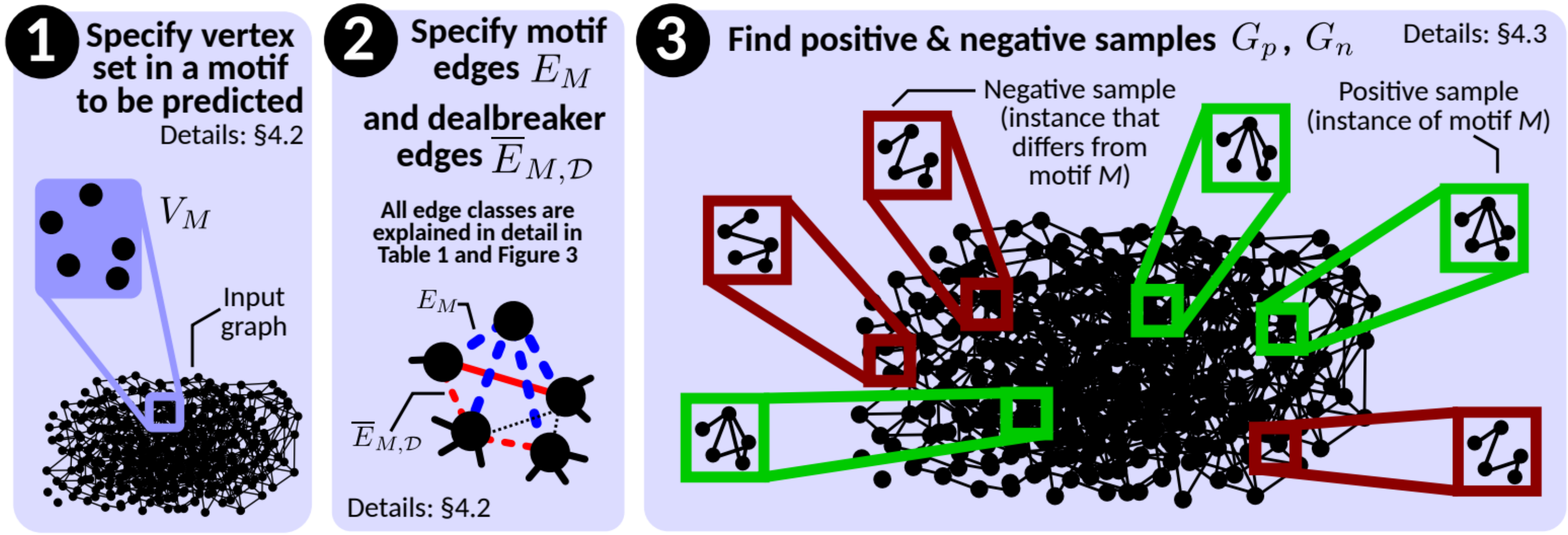


SEAM Architecture (learning from Subgraphs, Embeddings and Attributes for Motif prediction)

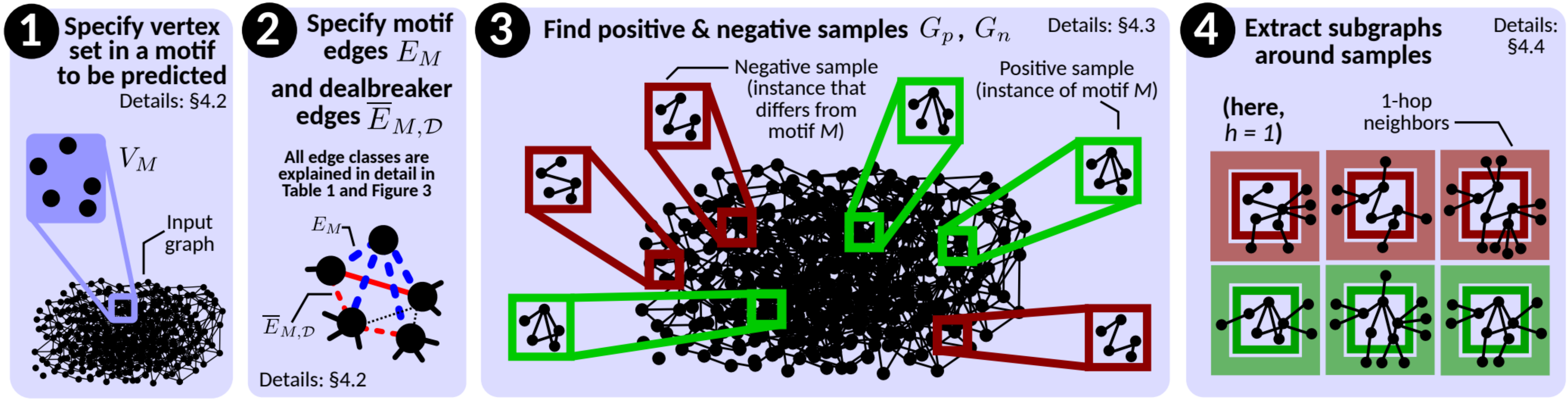




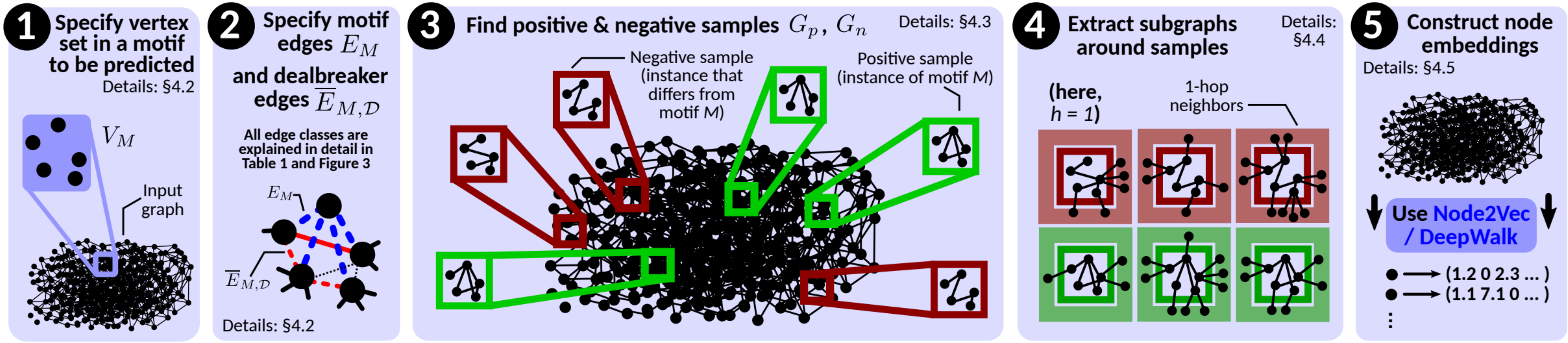
SEAM Architecture (learning from Subgraphs, Embeddings and Attributes for Motif prediction)



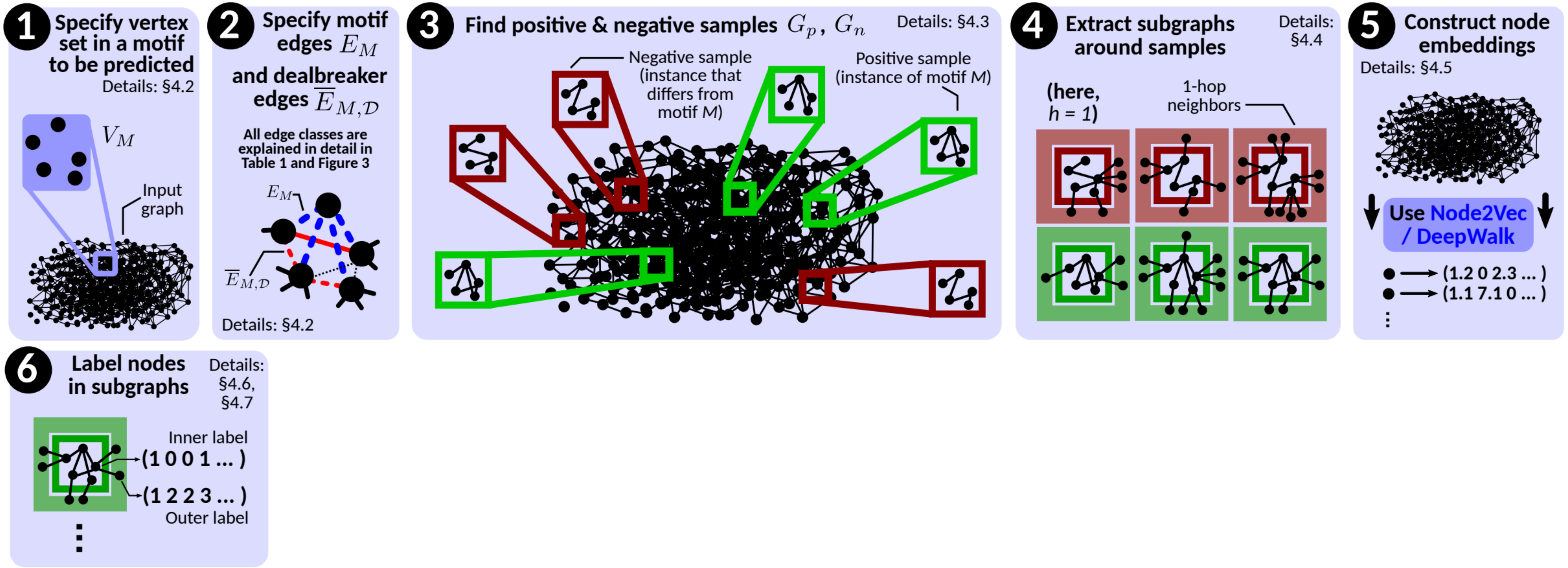
SEAM Architecture (learning from Subgraphs, Embeddings and Attributes for Motif prediction)



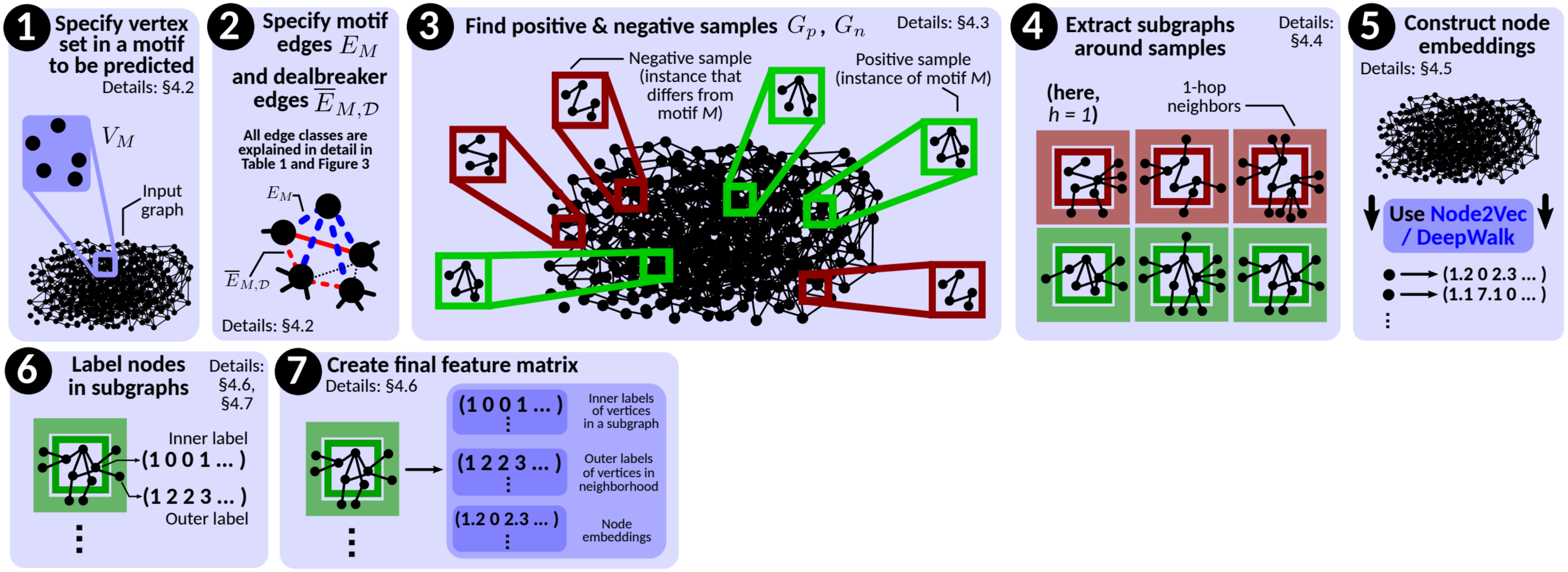
SEAM Architecture (learning from Subgraphs, Embeddings and Attributes for Motif prediction)



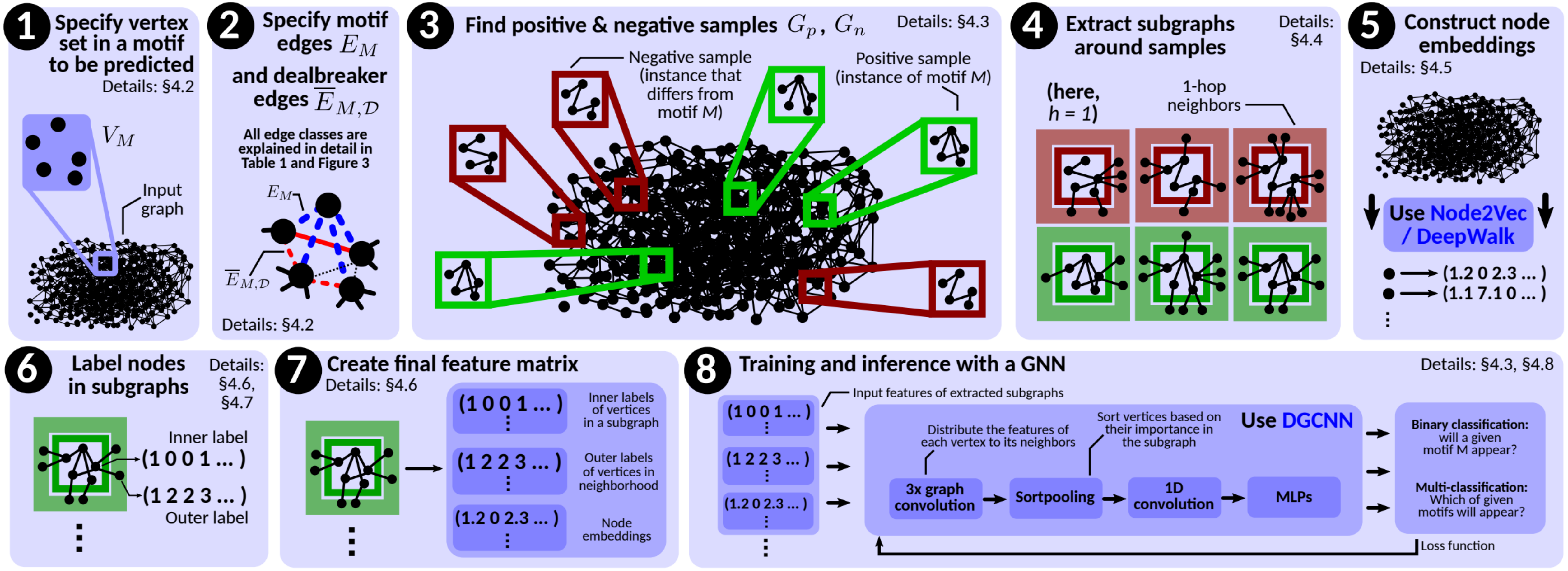
SEAM Architecture (learning from Subgraphs, Embeddings and Attributes for Motif prediction)



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

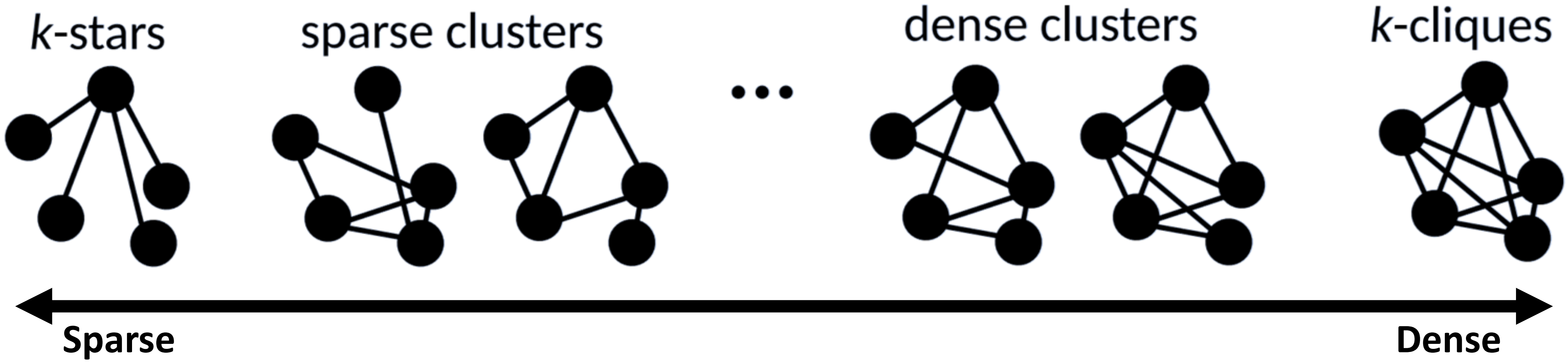


SEAM Architecture (learning from Subgraphs, Embeddings and Attributes for Motif prediction)



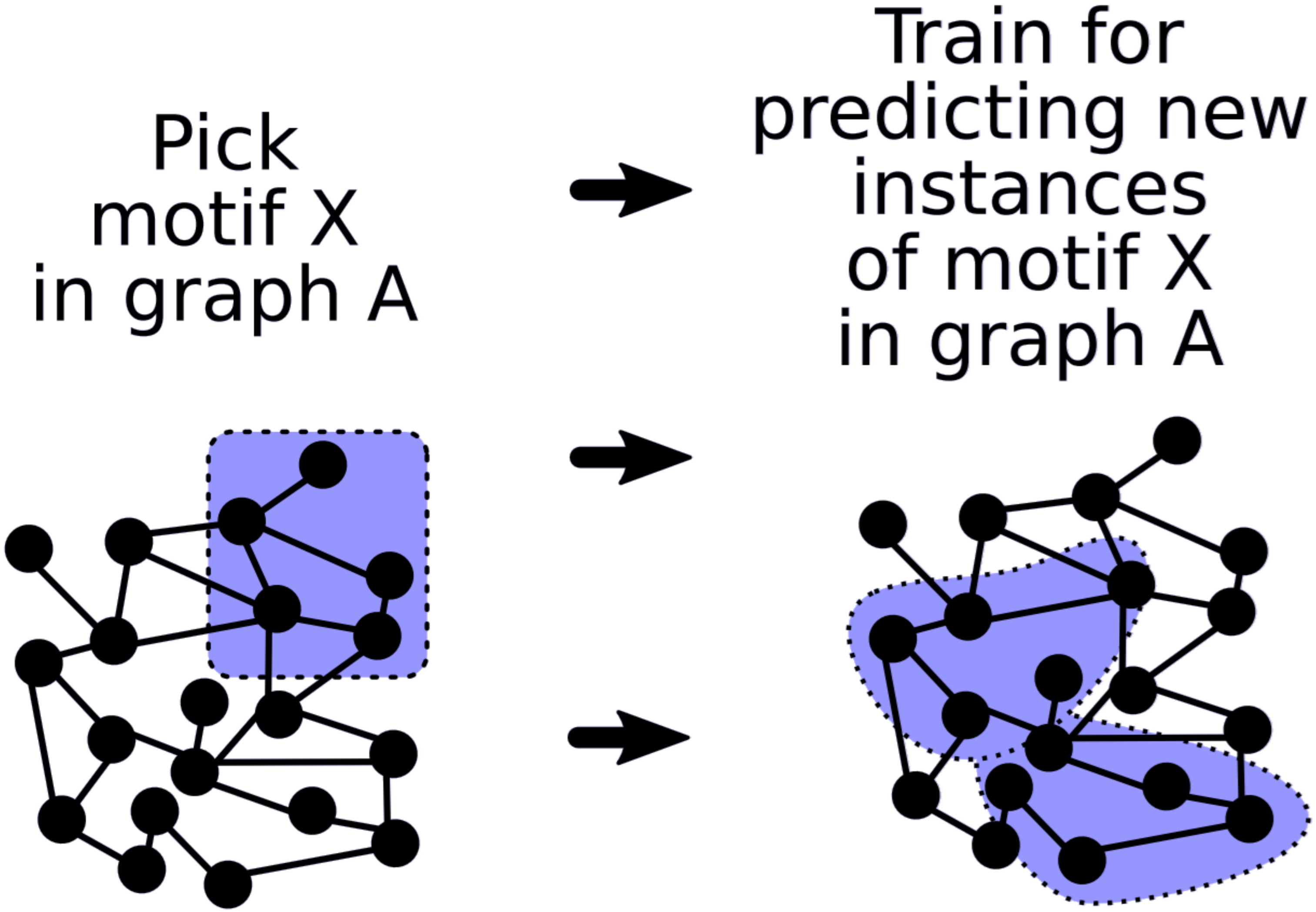
Evaluation: considered motifs & scenarios

Evaluation: considered motifs & scenarios



Evaluation: considered motifs & scenarios (transductive, inductive)

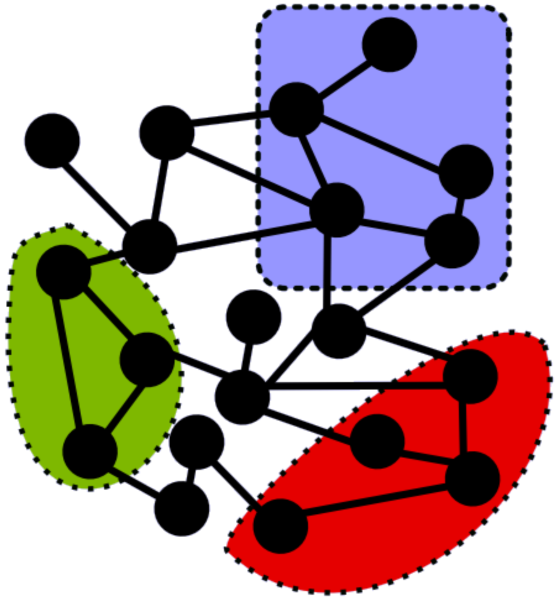
Evaluation: considered motifs & scenarios (transductive, inductive)

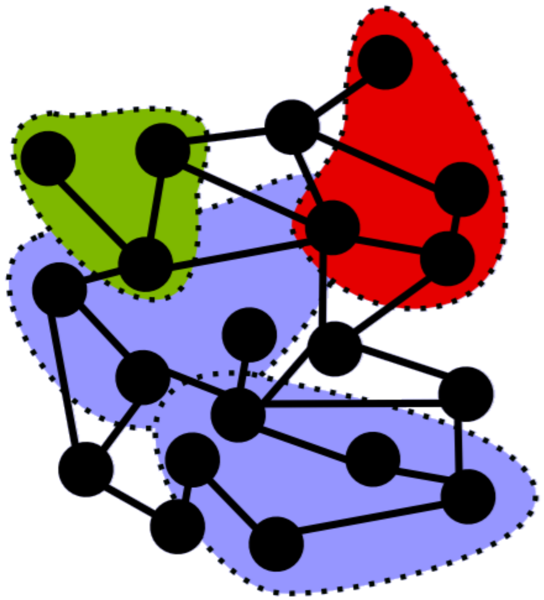


Evaluation: considered motifs & scenarios (transductive, inductive)

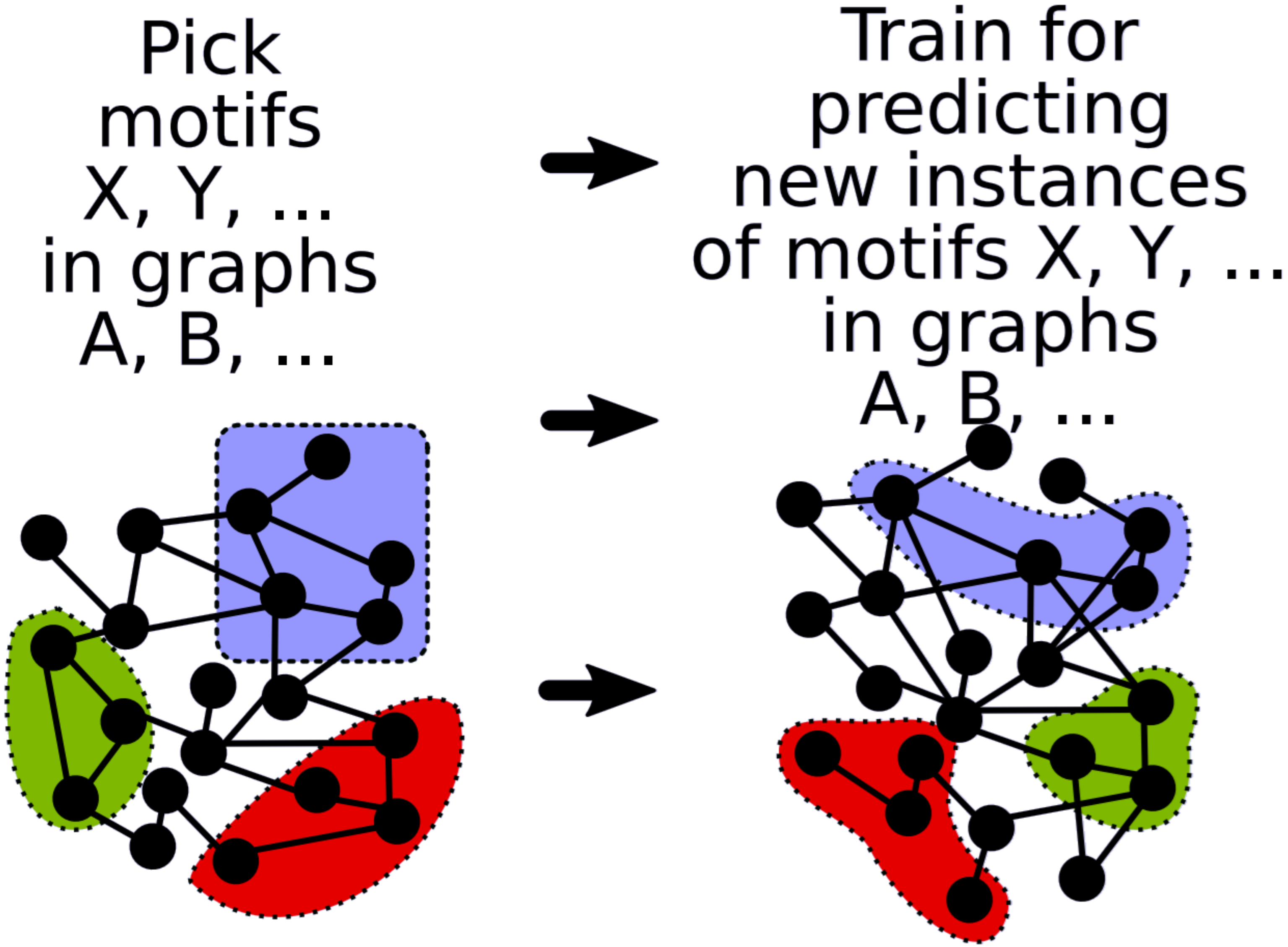
Pick motifs X, Y, ... in graph A

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





Evaluation: considered motifs & scenarios (transductive, inductive)



Evaluation

	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star
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

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

Evaluation

	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star
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

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

	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star
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

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

„db" – with deal-breaker edges

Evaluation

With correlation

No correlation

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

	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star
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

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

Mul, Min, Avg: different variants (i.e., how link scores are aggregated to form motif scores)

No correlation

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

	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star
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

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

„db" – with deal-breaker edges

Evaluation

With correlation

No correlation

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

	3-star	5-star	7-star	3-clique	5-clique	7-clique	3-db-star	5-db-star	7-db-star
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	[illegible]	[illegible]	[illegible]	[illegible]	4.15 ± 0.89	54.59 ± 0.65	54.94 ± 0.27
AA (Avg)	63.96 ± 0.68	63.66 ± 0	[illegible]	[illegible]	[illegible]	[illegible]	4.52 ± 0.57	55.20 ± 0.53	54.55 ± 0.39
Jaccard (Mul)	67.17 ± 0.92	62.01 ± 0	[illegible]	[illegible]	[illegible]	[illegible]	1.75 ± 1.10	51.68 ± 0.85	50.93 ± 0.64
Jaccard (Min)	69.20 ± 0.80	67.11 ± 0	[illegible]	[illegible]	[illegible]	[illegible]	2.30 ± 0.54	51.86 ± 0.77	50.87 ± 0.53
Jaccard (Avg)	70.12 ± 0.78	68.59 ± 0	[illegible]	[illegible]	[illegible]	[illegible]	1.76 ± 0.91	49.74 ± 0.75	47.66 ± 0.58
SEAL (Mul)	76.68 ± 0.61	74.00 ± 0	[illegible]	[illegible]	[illegible]	[illegible]	8.53 ± 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

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

SPCL