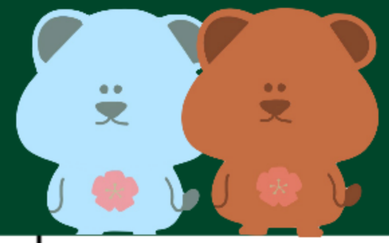
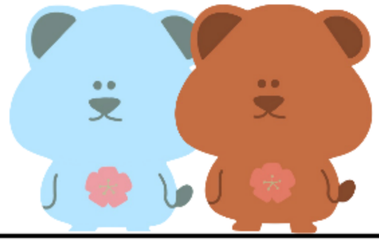


12. Frequent Subgraph Mining with GNNs

김나현, 이은빈

EWHA EURON

목차



EWHA EURON

Fast Neural Subgraph Matching & Counting

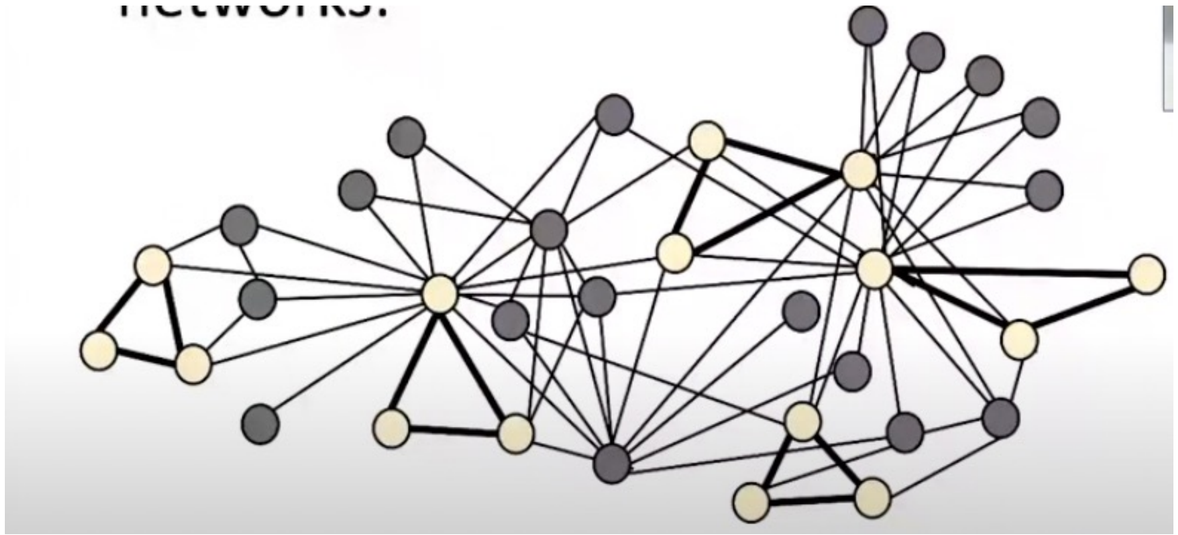
EWHA EURON

Subgraph

 # Subgraph

- the building blocks of networks (네트워크의 구성 요소)
- 네트워크를 characterize하고 discriminate 할 수 있다.
- Ex) Lego 조각 하나가 subgraph

  Lego 결과물이 network



Subgraph and motifs

Given grph G = (V, E) :

Def 1. Node-induced subgraph (= induced subgraph)

node를 중심으로 subset 구성

ex) chemistry

Def 2. Edge-induced subgraph (= non-induced subgraph / subgraph)

edge를 중심으로 subset 구성

ex) knowledge graph

Graph Isomorphism

Graph isomorphism problem

두 graph가 identical인지 확인하는 것은 중요한 문제이다

- $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic** if there exists a bijection $f: V_1 \rightarrow V_2$ such that $(u, v) \in E_1$ iff $\big(f(a), f(b)\big) \in E_2$
  - f is called the **isomorphism:**

* bijection : 일대일 대응

graph 크기 같더라도 edge의 방향/수에 따라 다양한 non-isomorphic graph 존재

Graph Isomorphism

Graph isomorphism problem

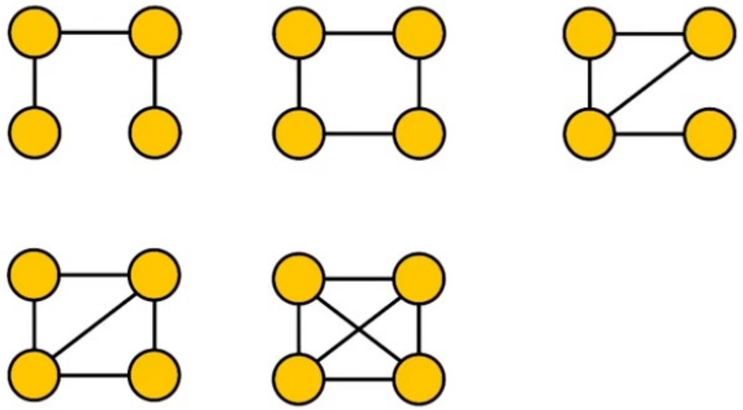
'They are the same graph, they're just drawn in the different way'

same edges, same direction을 가지고 있음

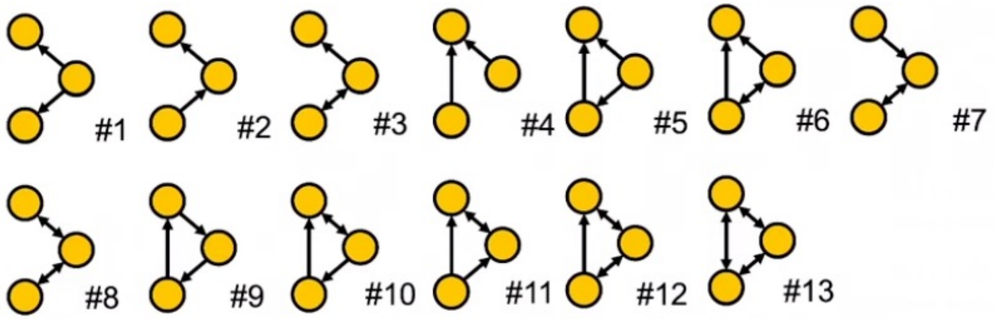
그래프의 구성이 본질적으로 다른 경우, **non-isomorphic**

Case examples of Subgraphs

All non-isomorphic, connected, undirected graphs of size 4



All non-isomorphic, connected, directed graphs of size 3



Network motifs

Network motifs

recurring, significant patterns of interconnections

How define a network motif:

1/ Pattern : Small (node-induced) subgraph

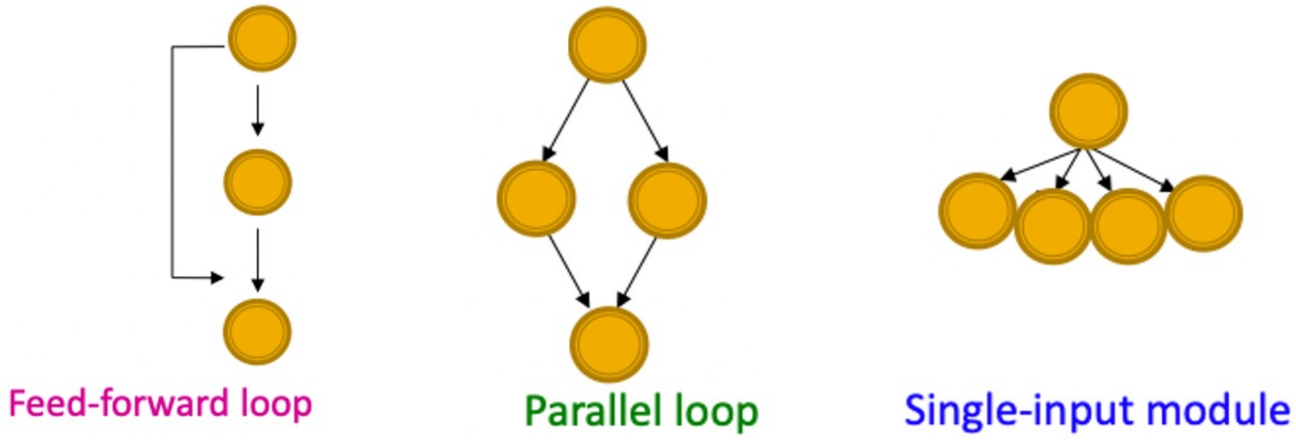
2/ Recurring : Pattern이 전체 graph에서 나타나는 빈도

3/ Significant : More frequent than expected in randomly generated graph?

Network motifs

Network motifs

- graph가 어떻게 작동하는지 이해할 수 있다
- graph의 presence에 따라 prediction을 할 수 있다
- Examples : Feed-forward loops, Parallel loops, single-input modules

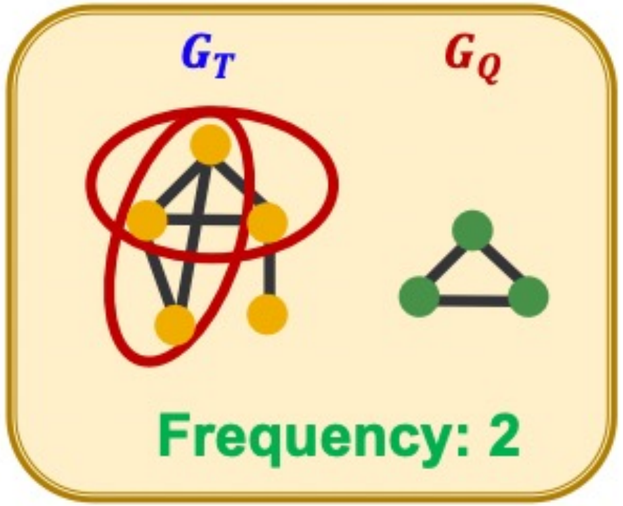


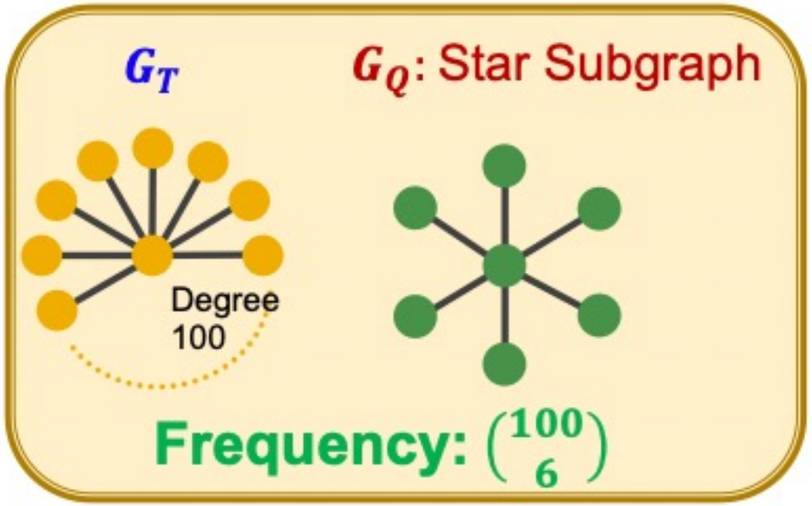
Subgraph Frequency

 # Subgraph Frequency

- G_T : Target graph dataset , G_Q : small graph
- Frequency of G_Q in G_T :

  Large graph G_T의 subgraph에 포함되는 small graph G_Q의 개수



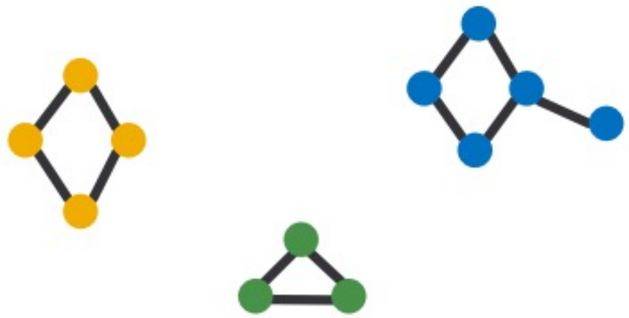


Subgraph Frequency

Subgraph Frequency

What if the dataset contains multiple graphs?

- 개별 graph에 해당하는 분리된 구성요소가 있는 Large graph dataset G_T 으로 간주하여 frequency 계산

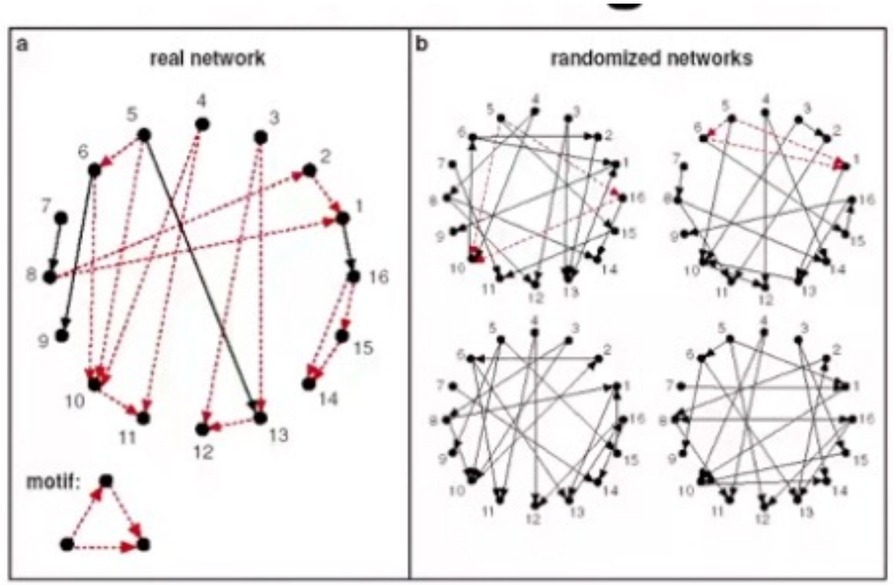


Defining Motif Significance

 # Key Idea

Random하게 생성된 Network의 motif frequency보다

Real network의 motif frequency가 더 많을 것이라고 가정



Milo *et. al.*, Science 2002

Jure Leskovec, Stanford CS224W: Machine Learning with Graphs, cs224w.stanford.edu

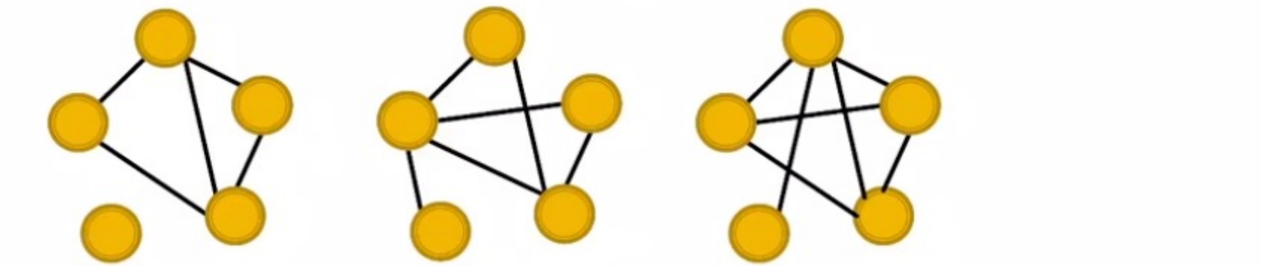
Defining Motif Significance

 # Random Graph Generation (1/3)

1/ Erdos-Renyi Random Graph

$G_{n,p}$는 n개의 node에서 확률 p에 의해 edge를 random하게 생성하는 undirected graph

- Generated graph is a result of a random process:



Three random graphs drawn from $G_{5,0.6}$

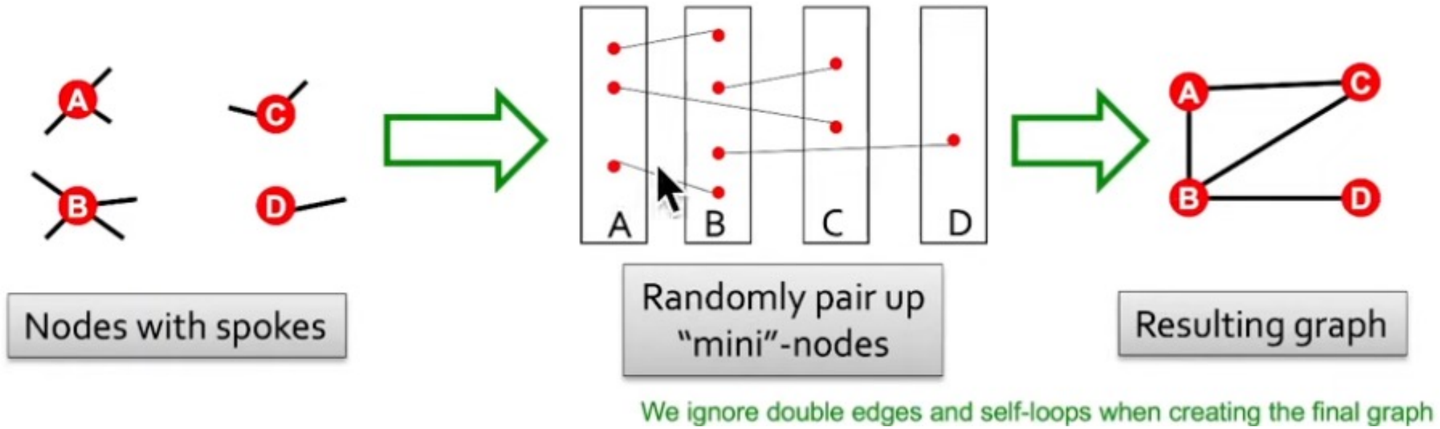
Defining Motif Significance

 # Random Graph Generation (2/3)

2/ Configuration model

같은 Degree sequence를 가진 G^{real}과 비교

각 node는 degree만큼 random하게 edge 연결하여 graph 생성



Defining Motif Significance

 # Random Graph Generation (3/3)

3/ Switching

Edge 쌍을 무작위로 선택하여 endpoint를 바꿔 새로운 graph 생성

비교 대상 graph와 node degree가 같다는 특징

Z-score for Statistical Significance

 # Z-score

통계학의 Z-score 개념 차용하여 network motif에서 중요 subgraph 선택

일반적으로 10,000개~100,000 개의 random subgraph 생성

- Z_i captures **statistical significance of motif i**:

$$Z_i = (N_i^{\text{real}} - \bar{N}_i^{\text{rand}}) / \text{std}(N_i^{\text{rand}})$$

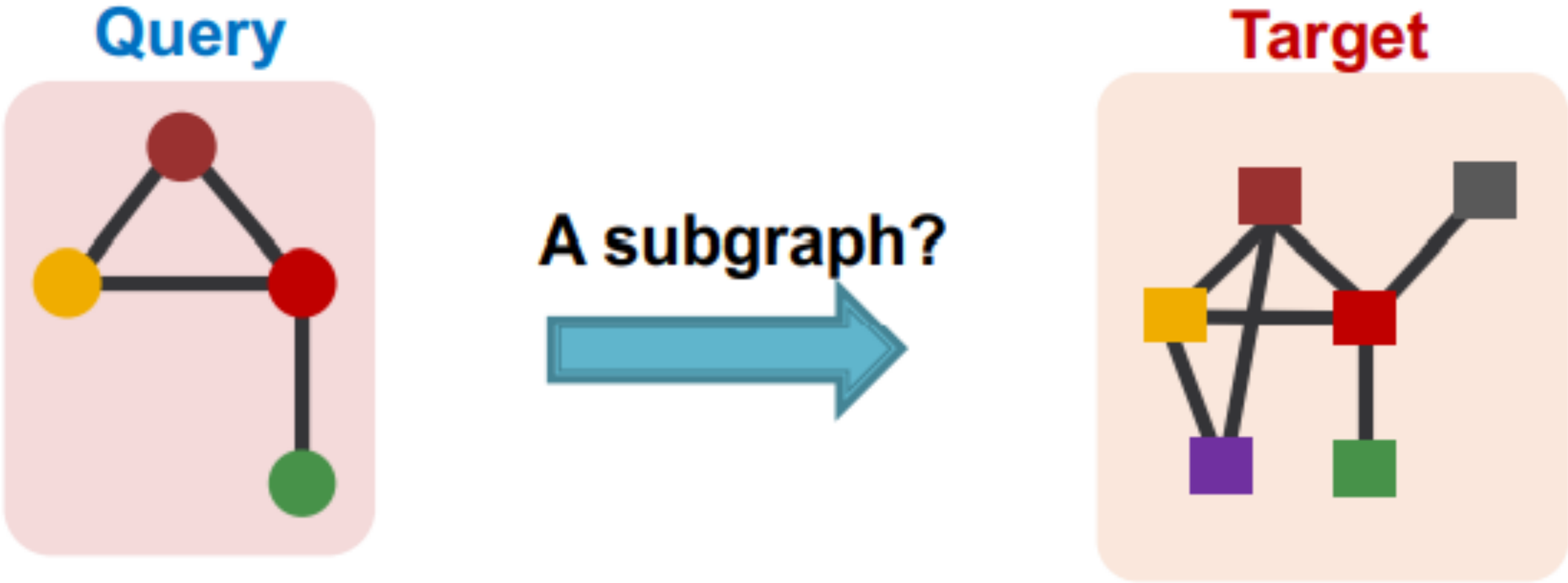
  - N_i^{real} is #(motif i) in graph G^{real}
  - $\bar{N}_i^{\text{rand}}$ is average #(motifs i) in random graph instances

Neural Subgraph Matching

EWHA EURON

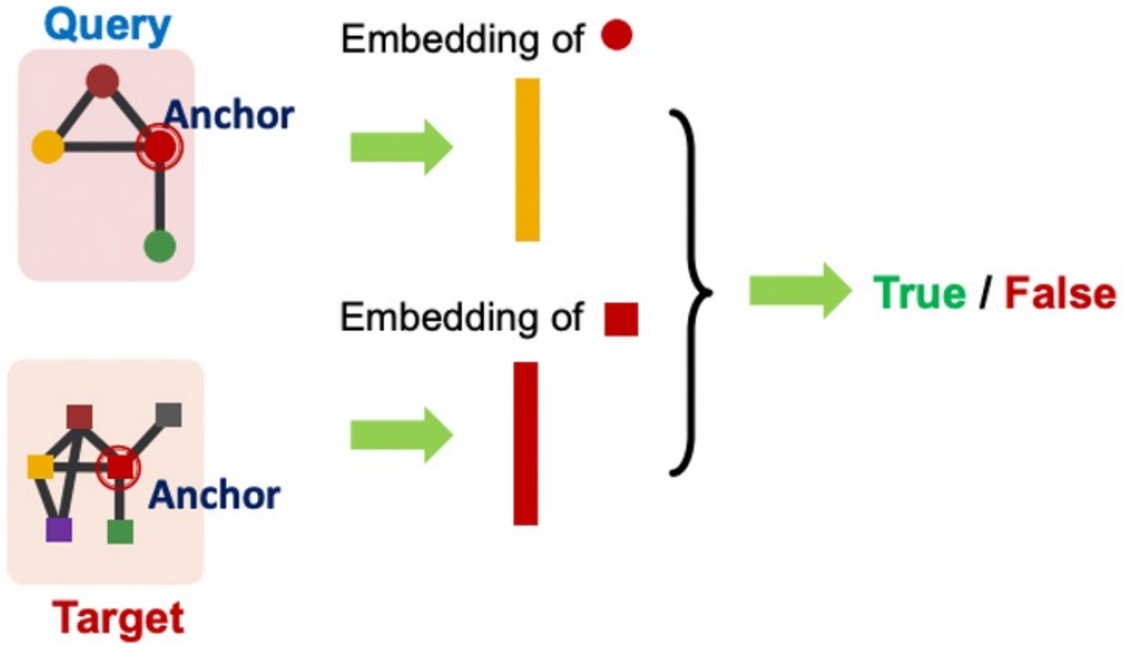
Subgraph Matching

: query 그래프가 target 그래프의 subgraph isomorphism인지 확인하는 task



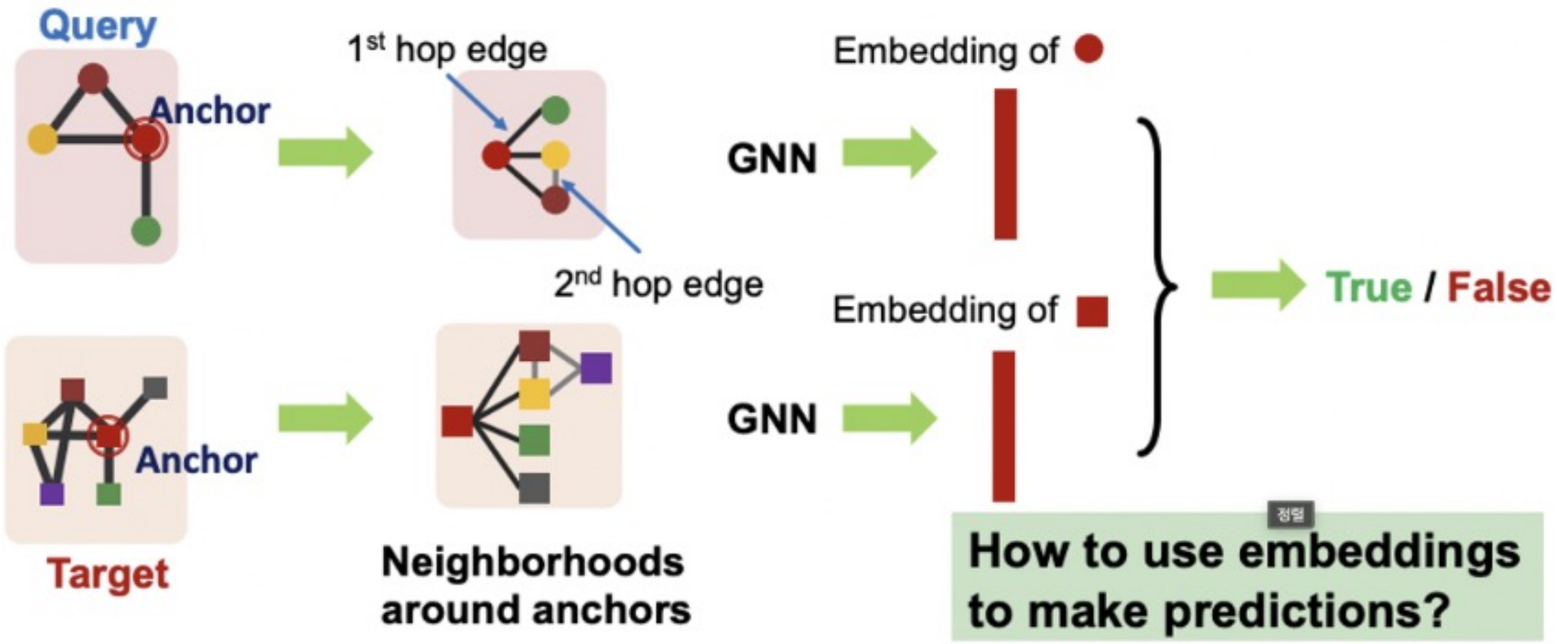
(노드 간 올바른 맵핑을 위해 노드별로 색을 달리 함.)

Neural Architecture for Subgraphs



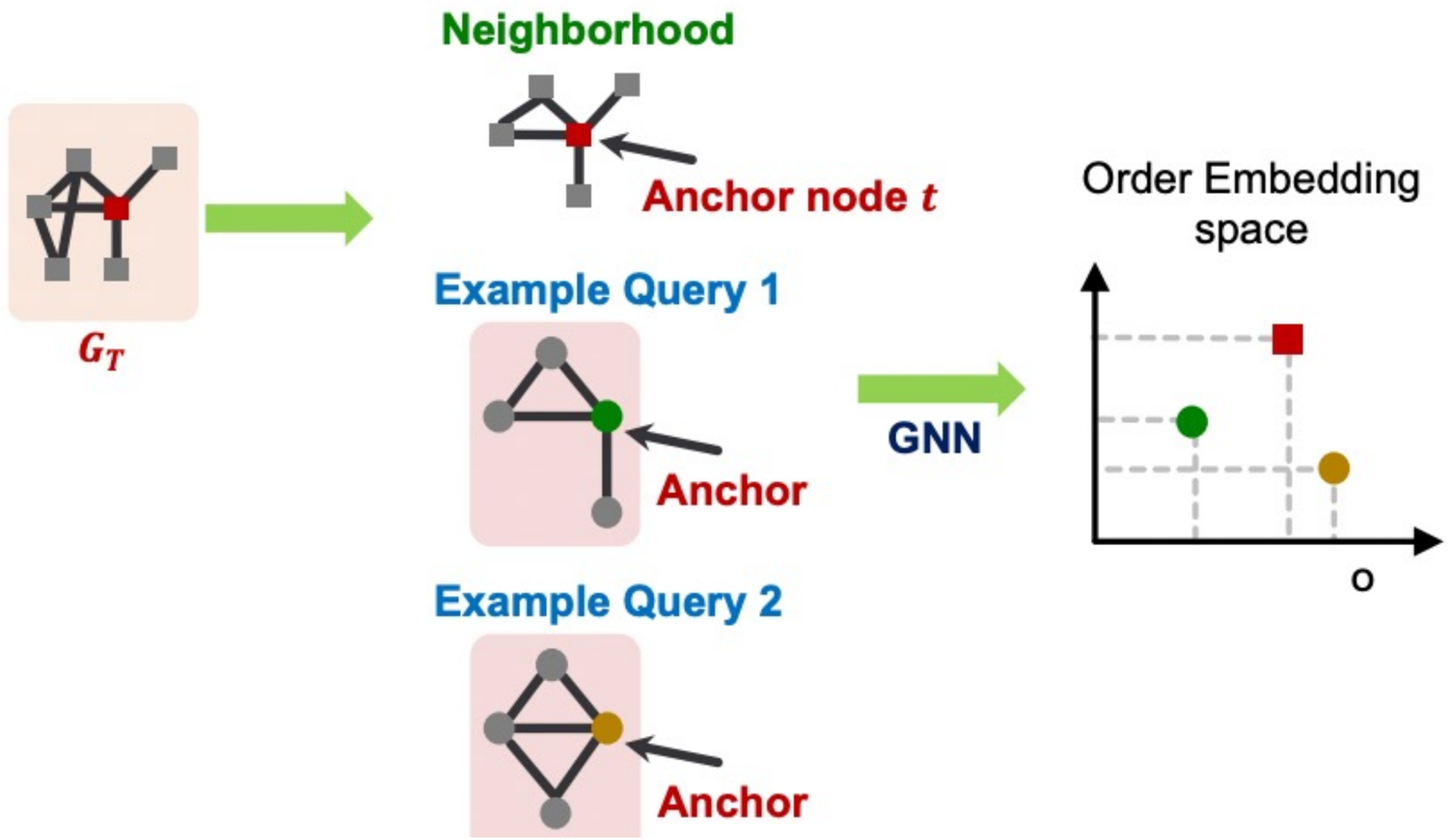
node anchor를 활용하여 query의 노드 v와 target의 노드 u의 임베딩이 동일한지 확인

Neural Architecture for Subgraphs



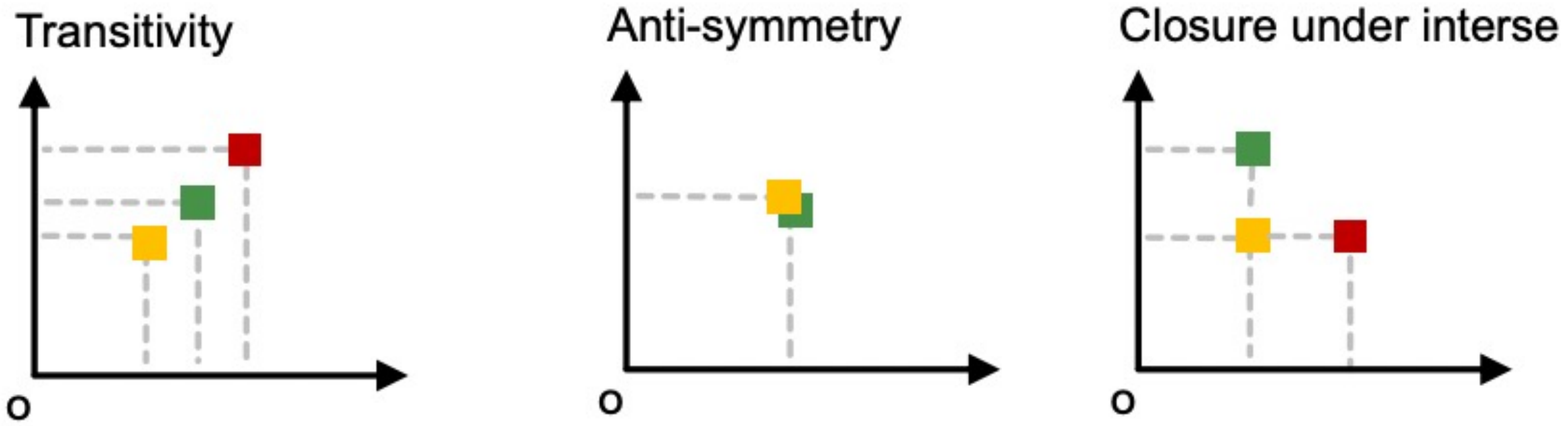
Query의 anchor node가 n-hop을 가질 때 n-hop 내에 있는 이웃노드들의 임베딩 비교.

Subgraph Order Embedding Space



(노드 간 올바른 맵핑을 위해 노드별로 색을 달리 함.)

Order Embedding Space



Transitivity: *G*1이 *G*2의 subgraph이고 *G*2가 *G*3의 subgraph라면 *G*1은 *G*3의 subgraph이다.

Anti-symmetry: *G*1이 *G*2의 subgraph이고 *G*2가 *G*1의 subgraph라면 두 그래프는 isomorphic하다.

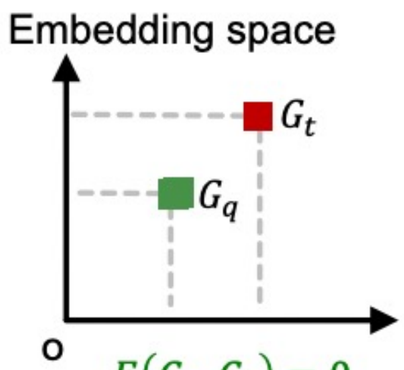
Closure under intersection: 노드가 하나인 그래프는 모든 그래프의 subgraph이다. 음수를 가지는 임베딩은 없으며 $a \leq b$, $a \leq c$라면 a는 유효한 값을 가진다.

Order Constraint

: GNN 사용 시, 어떤 Loss function을 사용해야 할까?

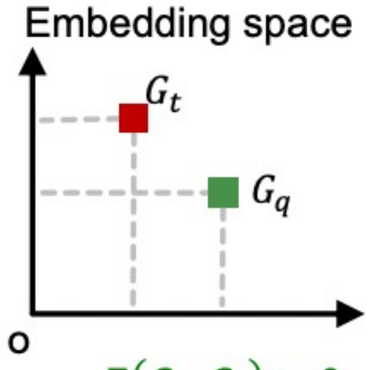
$$E(G_q, G_t) = \sum_{i=1}^{D} (\max(0, z_q[i] - z_t[i]))^2$$

Order constraint는 이상적인 embedding



$E(G_q, G_t) = 0$

According to the order embedding,
G_q is a subgraph of G_t!



$E(G_q, G_t) > 0$

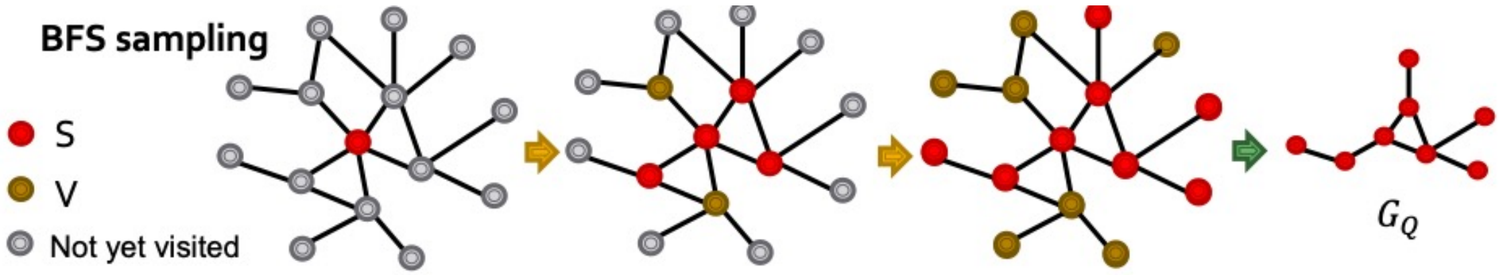
According to the order embedding,
G_q is **not** a subgraph of G_t!

E(*Gq*, *Gt*)=0 이면 *Gq*는 *Gt* 의 Subgraph가 이고 *E*(*Gq*, *Gt*)〉0 이면 *Gq*는 *Gt* 의 Subgraph가 아니다.

(E는 ‘order constraint violation의 양‘을 나타낸다.
이를 margin 이라고도 부른다.)

Training

- 학습 데이터셋 (G_q, G_t)는 G_t의 subgraph인 G_q가 반, 그렇지 않은 것이 반이 되도록 구성해야 한다.
- Positive sample에 대해서는 $E(G_q, G_t)$를 최소화하도록 negative smaple에 대해서는 $max(0, \alpha - E(G_q, G_t))$를 최소화하도록 학습하는데 이는 모델이 임베딩을 너무 멀리 이동시키는 것을 방지하기 위함이다.
- 데이터셋 G로부터 학습을 위한 G_T와 G_Q를 **샘플링**하는 과정이 필요하다.
- G_T는 **무작위로 anchor 노드 v를 뽑은 뒤 거리가 K인 모든 노드를 포함**시켜 만든다.
- **Positive example G_Q는 BFS 샘플링**을 거친다.

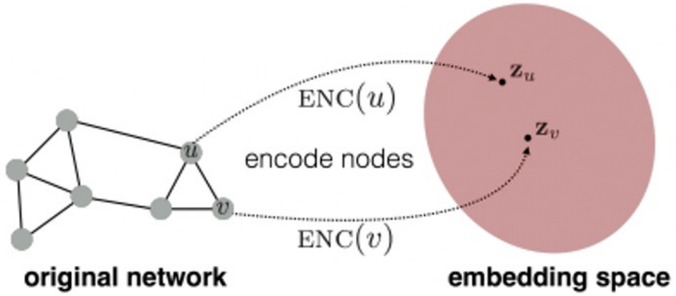


Finding Frequent Subgraphs

EWHA EURON

Finding Frequent Subgraphs

- **1) Enumerating** all size-k connected subgraphs
- **2) Counting** #(occurrences of each subgraph type)



❗위 방법들 : 모든 패턴들을 조합함으로써 Combinatorial explosion을 가져오기 때문에 높은 computation 비용을 수반한다.

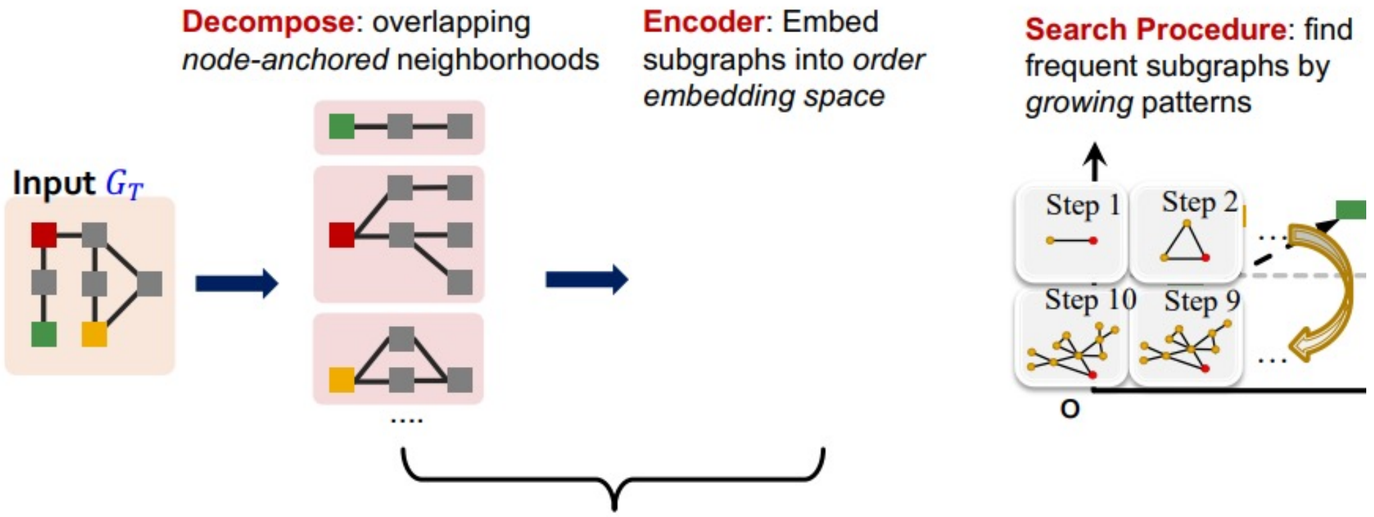
👉따라서 우리는 이러한 문제를 Representation learning을 통해서 해결

Solution with Representation Learning

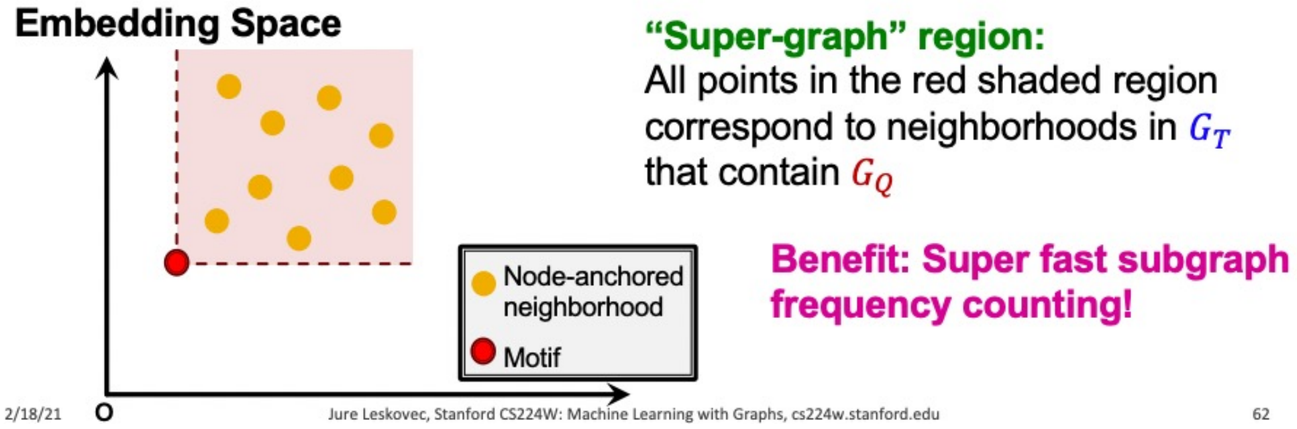
✅ big target graph를 embed
✅ small query graph의 빈도수를 예측 (직접 세기보다는, 빈도수 예측)

✅ all size scale에서의 빈도수 예측보다는,
small subgraph에서 시작해서 큰 size에 이르기까지
노드 단위로 grow

SPMiner

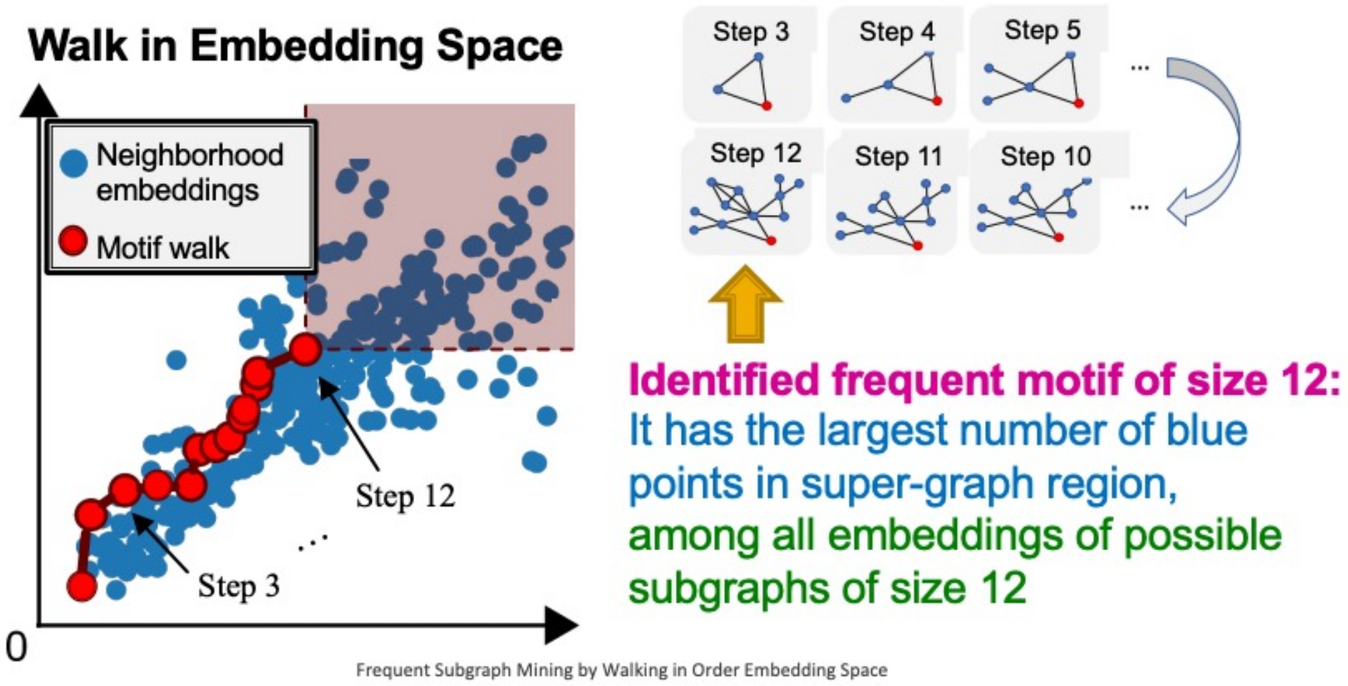


👉*GT* 그래프 & Subgraph *GQ*를 모두 비교하며 Subgraph 빈도 수 구하기



👉'super-graph' region 안의 **노란 점들은 *GQ*를 포함하는 모든 *GT*의 neighborhoods**가 된다.

SPMiner



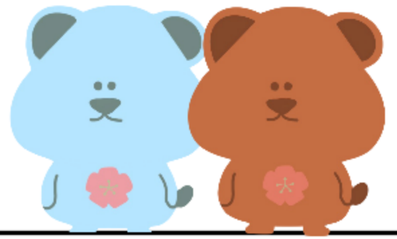
Frequent Subgraph Mining by Walking in Order Embedding Space

1️⃣시작 노드 u를 무작위로 선택

2️⃣node by node로 이웃노드를 덧붙여 motif를 키운다. (빈도수 높은 motif
👉*k* 스텝 후에 **붉은 영역에 속하는 neighborhoods의 수를 최대화하는 것이 목적**

3️⃣원했던 motif size에 도달하면 멈추고 *S*로부터 subgraph를 도출

THANK YOU



EWHA EURON