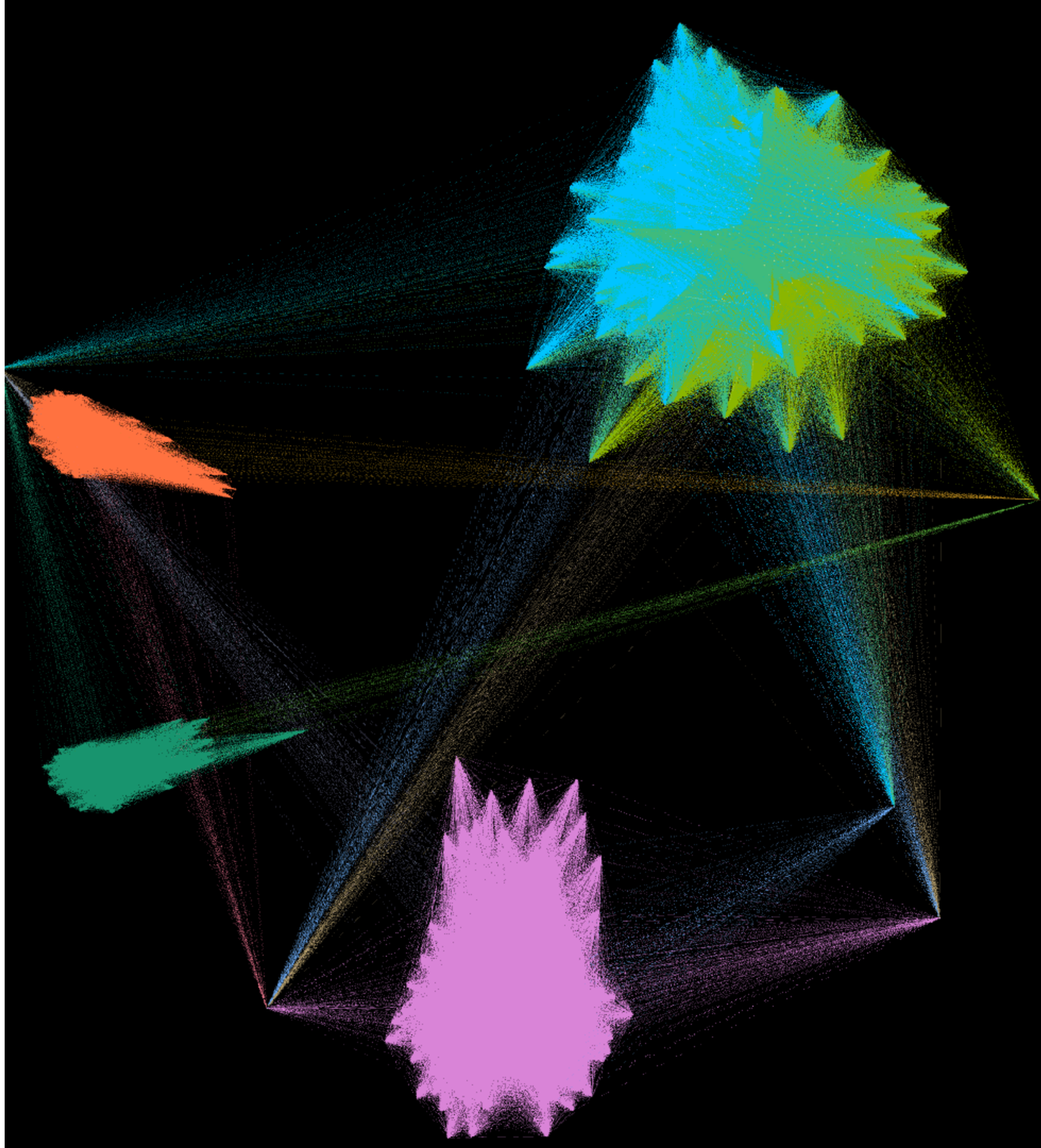
Graph Feature Representation

RC



Graphic File Format

Graph Format Table Comparison

	Edge List/Matrix Structure	XML Struture	Edge W[illegible]ht	Attrib[illegible]	Visual[illegible]n Attributes	Attrib[illegible]efault Value	Hierar[illegible]l Graphs	Dynamics
CSV	■		■					
DL Ucinet	■		■	■				
DOT Graphviz			■		■			
GDF			■	■	■	■		
GEXF		■	■	■	■	■	■	■
GML			■	■	■			
GraphML		■	■	■	■	■	■	
NET Pajek	■		■		■			
TLP Tulip								
VNA Netdraw			■	■				
Spreadsheet*			■	■				■

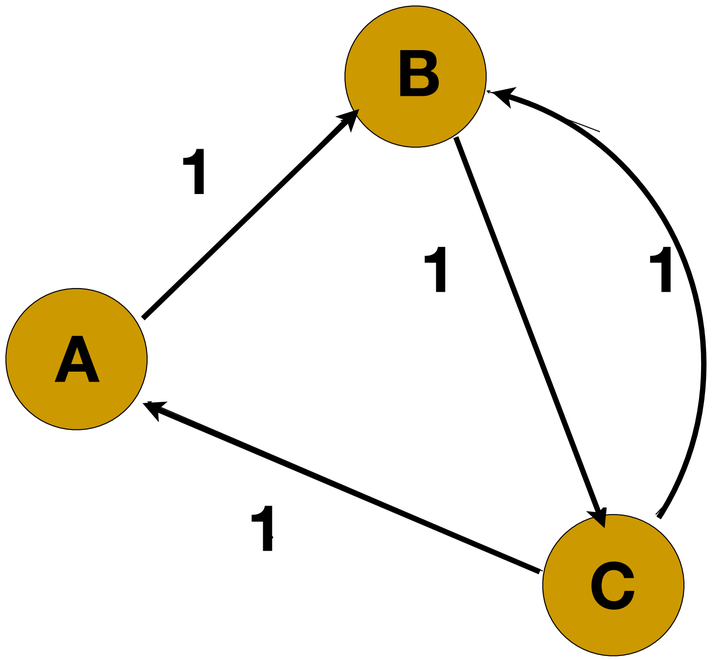
- GEXF
- Spreadsheet
- GraphML
- Guess GDF
- GML
- UCINet DL
- Netdraw VNA
- Graphviz DOT
- Pajek NET
- CSV
- Tulip TLP

Many features

Few features

File Type
- XML
- Tabular
- Text

Graphic Data Structure



Transition Matrix

	A	B	C
A	0	1	0
B	0	0	1
C	1/2	1/2	0

Adjacency Matrix

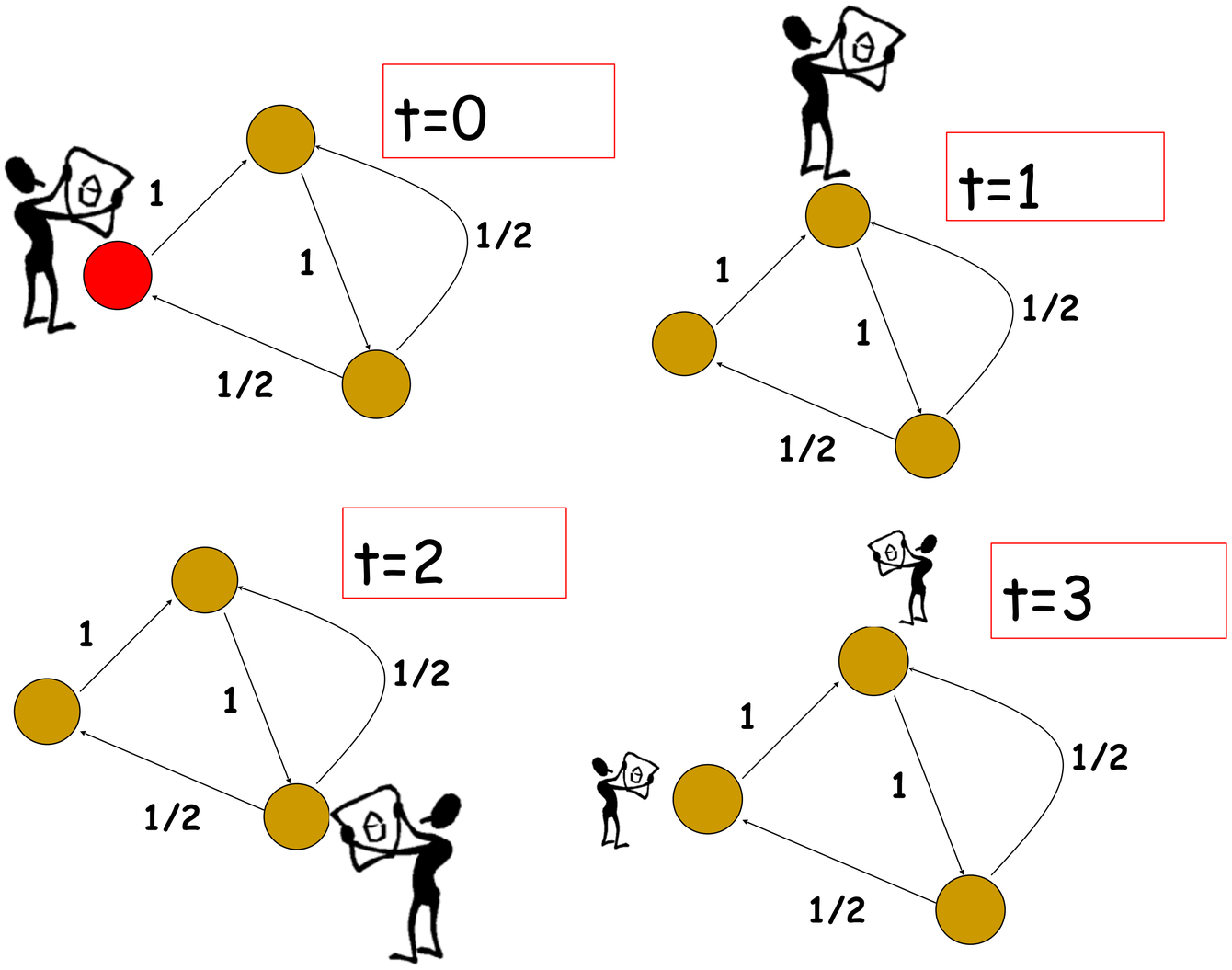
0	1	0
0	0	1
1	1	0

Degree **Relation**

Other methods to

“describe a Network”

Random Walk



Q.Does a **stationary distribution** always exist?

Yes, if the graph is **well-behaved**

Q.What is well-behaved?

- **Irreducible**
- **Aperiodic**

https://people.eecs.berkeley.edu/~sinclair/cs271/n24.pdf
http://www.csie.ntnu.edu.tw/~u91029/BipartiteGraph.html
https://en.wikipedia.org/wiki/Aperiodic_graph

夠多的 **Random Walk Path**

便可掌握所處的 **Network** 特性

Algorithm of
Random Walk

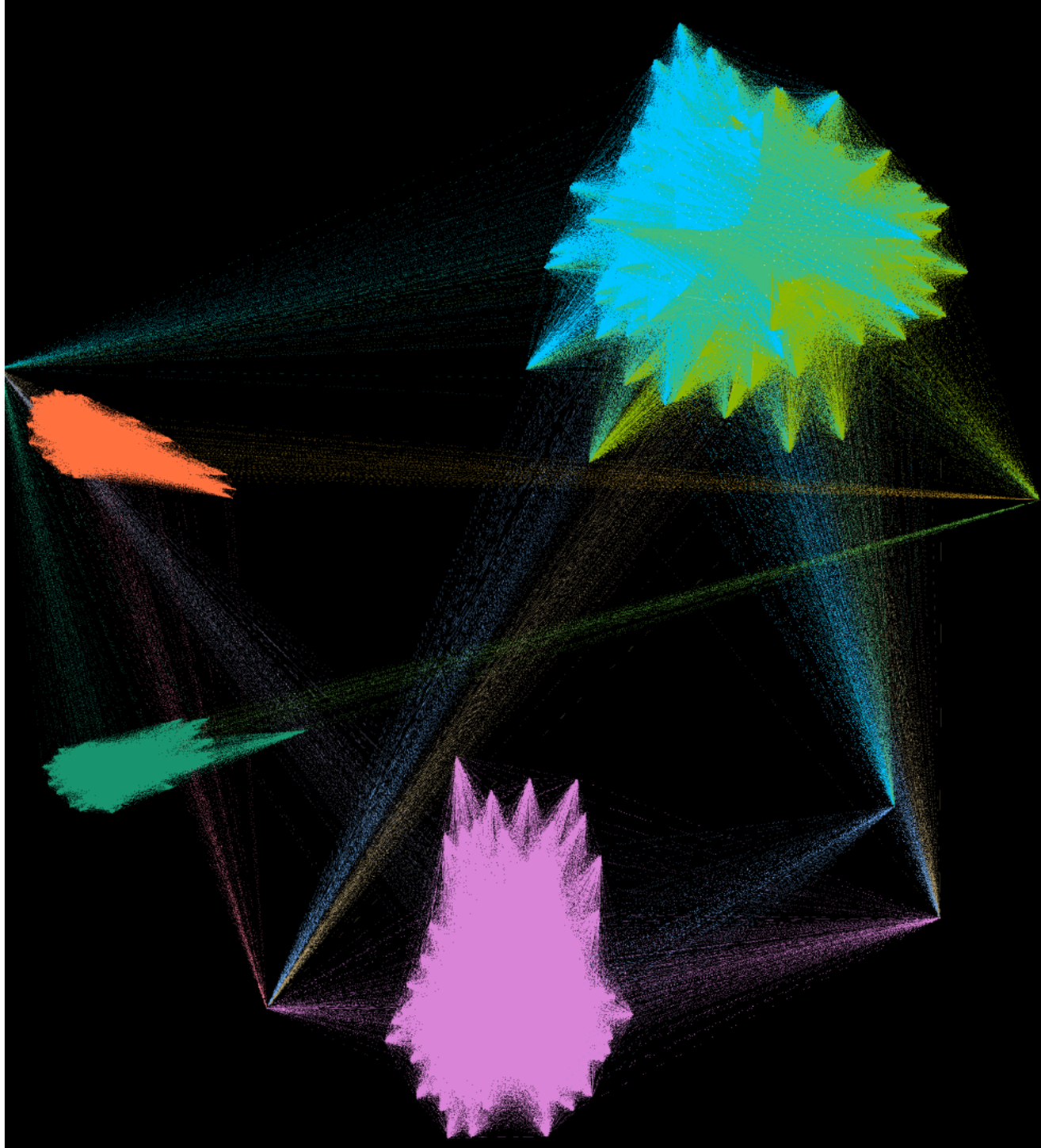
1. 網絡中每一個Node都當一次起始點

2.1 找出所在節點的所有相鄰節點

2.2 隨機挑選（或考慮權重）任一相鄰節點當成下一個時間點的位置

2.3 回到2.1，直至滿足拜訪長度

3.1 回到1.，直至收集夠多的隨機漫步路徑



Example of
Random Walk

people_061 >> people_160 >> people_050 >> people_919 >>
people_512 >> people_435 >> people_754 >> … >> people_223

people_160 >> people_919 >> people_111 >> people_782 >>
people_215 >> people_535 >> people_444 >> … >> people_411

people_223 >> people_753 >> people_745 >> people_160 >>
people_124 >> people_919 >> people_754 >> … >> people_555

people_627 >> people_002 >> people_777 >> people_558 >>
people_099 >> people_160 >> people_919 >> … >> people_388

Example of
Random Walk

people_061 >> **people_160** >> **people_050** >> people_919 >>
people_512 >> people_435 >> people_754 >> … >> people_223

people_160 >> **people_919** >> people_111 >> people_782 >>
people_215 >> people_535 >> people_444 >> … >> people_411

people_223 >> people_753 >> **people_745** >> **people_160** >>
people_124 >> people_919 >> people_754 >> … >> people_555

people_627 >> people_002 >> people_777 >> people_558 >>
people_099 >> **people_160** >> **people_919** >> … >> people_388

Example of

Random Walk

people_160 >> people_050
people_160 >> people_061

people_160 >> people_919

people_160 >> people_745
people_160 >> people_124

people_160 >> people_099
people_160 >> people_919

感覺到了嗎...？

詞彙向量化

與前後文

息息相關

Graph + Word2Vec

||

Graph Feature Representation

Example of

window_size = 2

Random Walk + Word2Vec

target >> context — SKIPGRAM

context >> target — CBOW

people_160 >> people_001
people_160 >> people_061
people_160 >> people_050
people_160 >> people_919

context context target context context

people_001 >> people_061 >> people_160 >> people_050 >> people_919 >> people_512
>> people_435 >> people_754 >> … people_223

people_001 >> people_061 >> people_160 >> people_050 >> people_919 >> people_512
>> people_435 >> people_754 >> … people_223

people_050 >> people_061
people_050 >> people_160
people_050 >> people_919
people_050 >> people_512

Hands-On: collections.deque

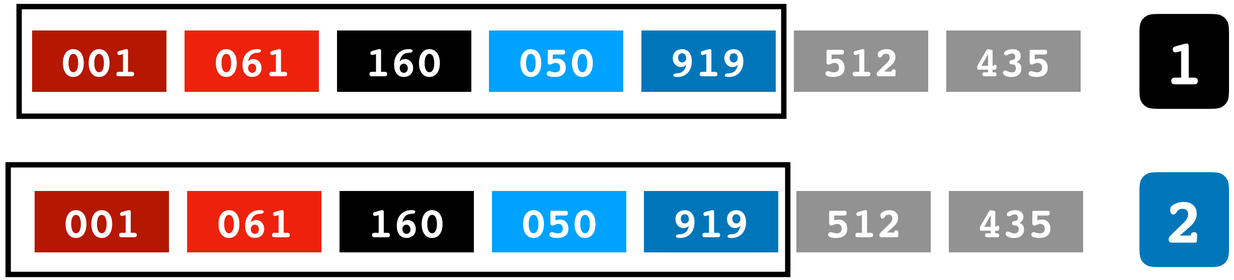
people_001 >> people_061 >> people_160 >> people_050 >> people_919 >>
people_512 >> people_435 >> people_754 >> … >> people_223

Code 1.

```
window_size = 2

path = [“001”, “061”, “160”, “050”, “919”, “512”,
“435”, “754”, “223”]

xs, ys = [], []
for center_idx in range(2, len(path)-2):
 for direction in [-1, 1]:
  for neighbor_idx in range(window_size):
   xs.append(path[center_idx])
   ys.append([path[center_idx+direction*number_idx]])
```



Code 2.

```
import collections

span = 1+window_size*2
q = collections.deque(maxlen=span)

for idx in range(span):
    q.append(path[idx])

xs, ys = [], []
for idx in range(2, len(path)-2):
  xx = q[window_size]*(span-1)
  yy = q[:window_size] + q[window_size+1:]
  for x, y in zip(xx, yy):
    xs.append(x)
    ys.append(y)

  q.append(path[idx])
```