# Sampling Large Graphs for Anticipatory Analysis

**Lauren Edwards*, Luke Johnson, Maja Milosavljevic, Vijay Gadepally, Benjamin A. Miller**

**IEEE High Performance Extreme Computing Conference**

**September 16, 2015**

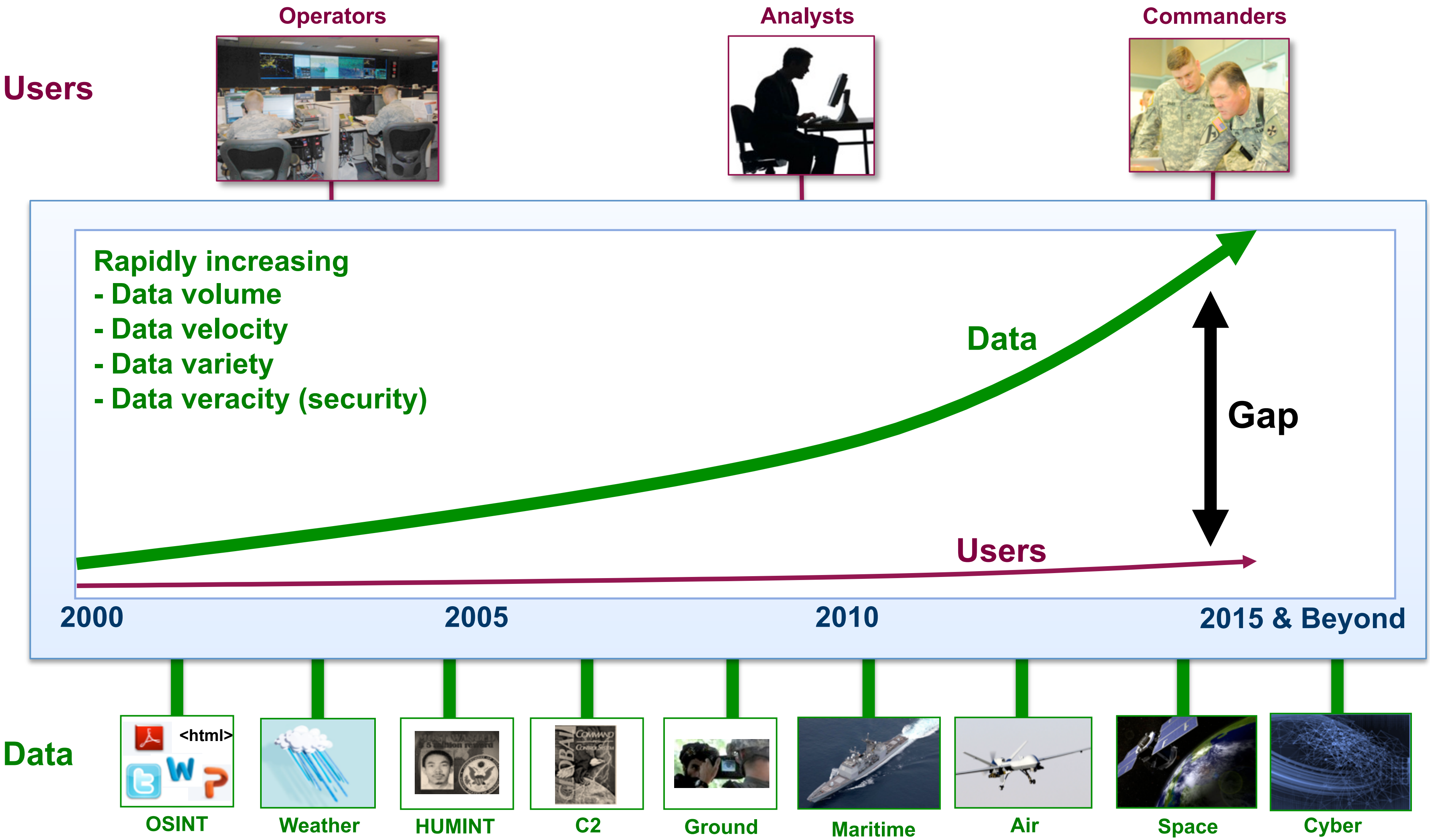
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

This work is sponsored by the Intelligence Advanced Research Projects Activity (IARPA) under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

# Outline

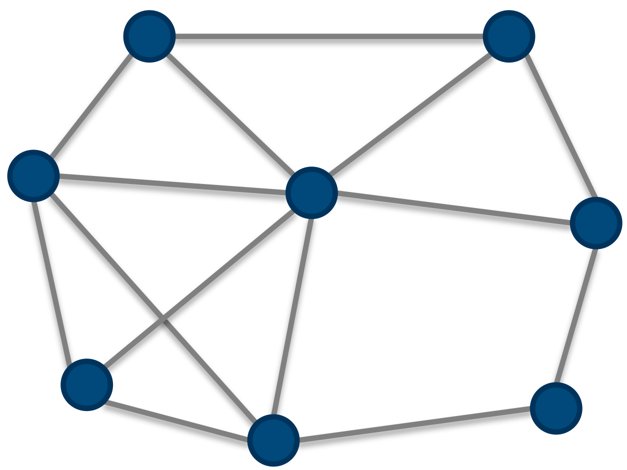
- Introduction
- Sampling Techniques
- Link Prediction
- Experimental Setup
- Results
- Summary

# Common Big Data Challenge



# What is a Graph?

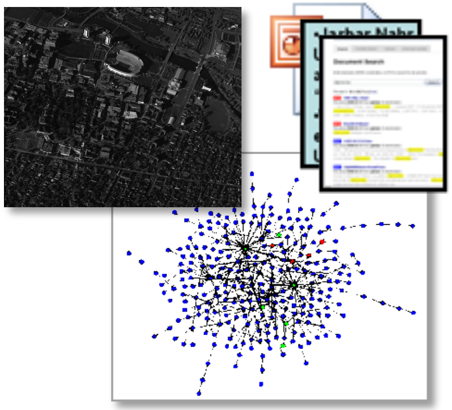
**Graphs are mathematical representations of physical or logical relationships between sets of objects**



**A graph is a pair of sets:**
- **A set of vertices/ nodes (entities)**
- **A set of edges/links (connections, transactions, and relationships)**

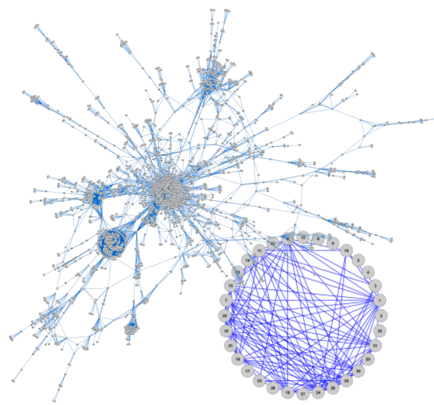
# Applications of Graph Analytics

## ISR



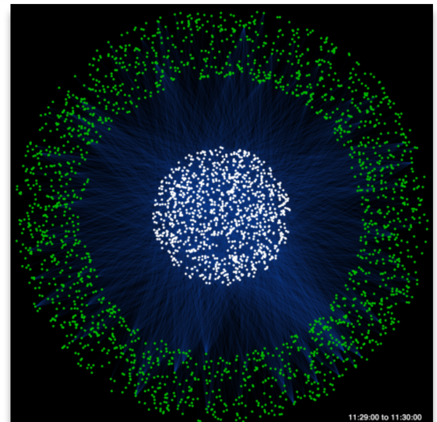
- Graphs represent entities and relationships detected through multiple data
- ~1K – 1M entities and connections
- GOAL: Identify anomalous patterns

## Social



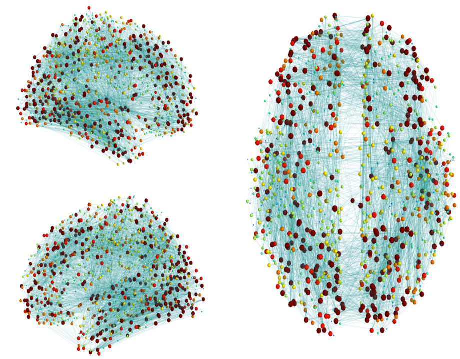
- Graphs represent relationships between individuals or documents
- ~10K – 10M individuals and interactions
- GOAL: Identify hidden social networks

## Cyber



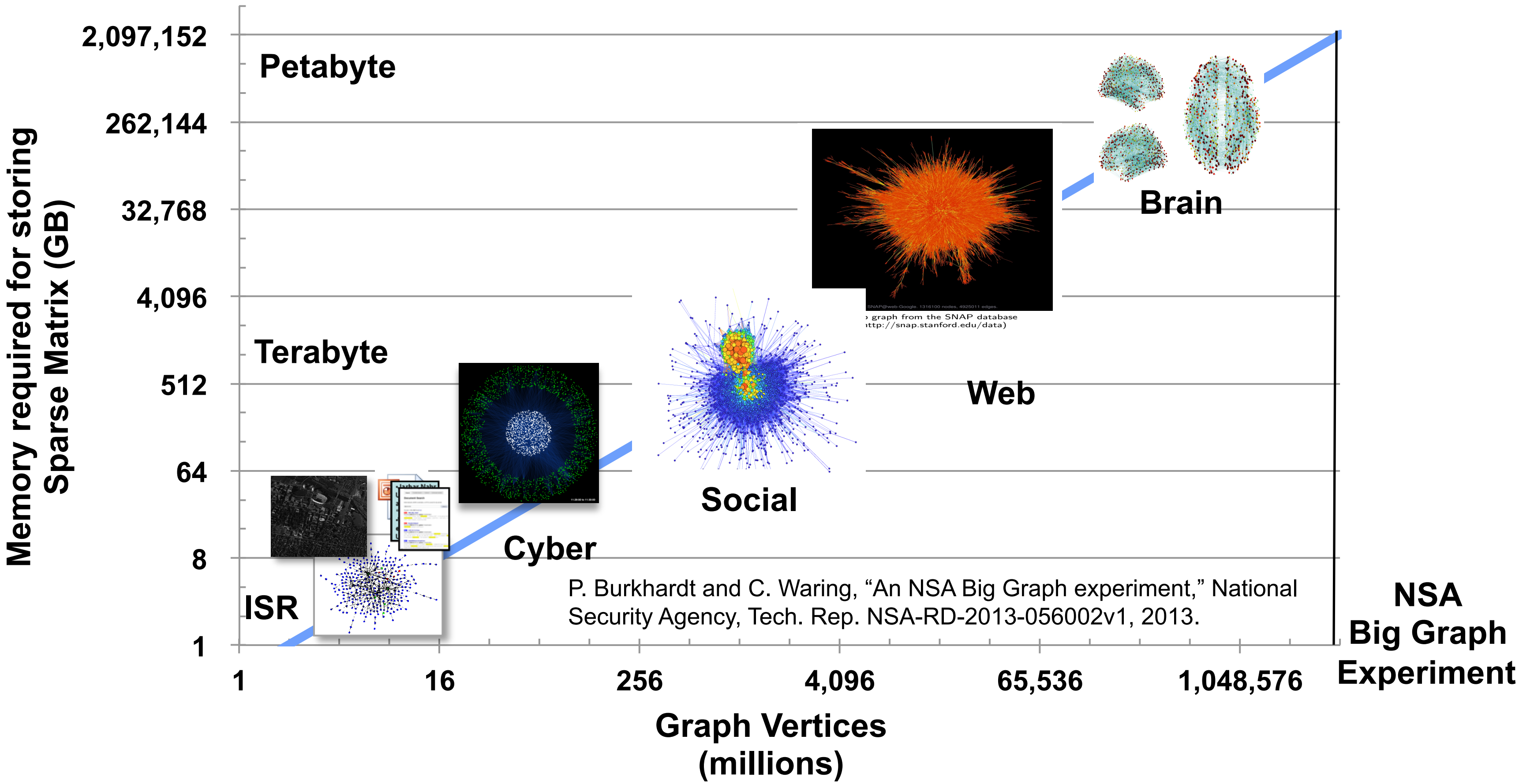

- Graphs represent communication patterns of computers on a network
- ~1M – 1B network events
- GOAL: Detect attack or malicious software

## Bio



- Graphs represent connectivity between brain regions
- ~1B – 1T regions and connections
- GOAL: Detect regions affected by a neurological condition

# Big Data Challenge



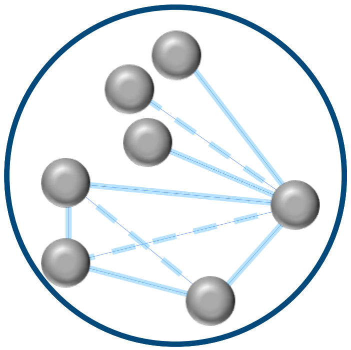

P. Burkhardt and C. Waring, "An NSA Big Graph experiment," National Security Agency, Tech. Rep. NSA-RD-2013-056002v1, 2013.

**How can these massive graphs be leveraged to provide accurate anticipatory intelligence?**

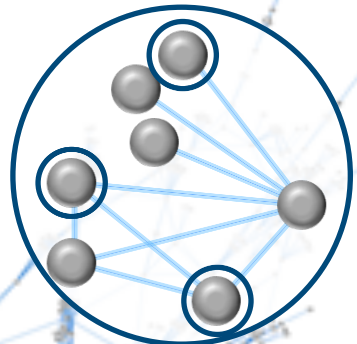
# Outline

- Introduction
- Sampling Techniques
- Link Prediction
- Experimental Setup
- Results
- Summary

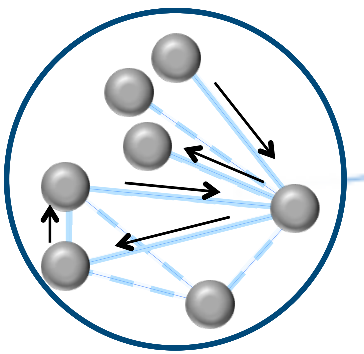
# Methods for Sampling Graphs



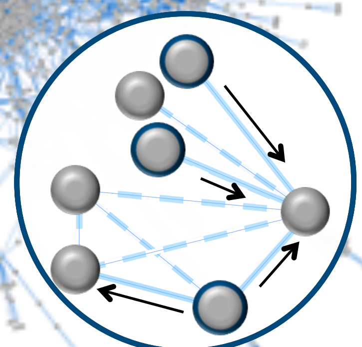
**Edge sampling methods**
- Consider each edges individually, keep it in the sample based on some criterion



**Vertex sampling methods**
- Sample individual vertices or local substructures



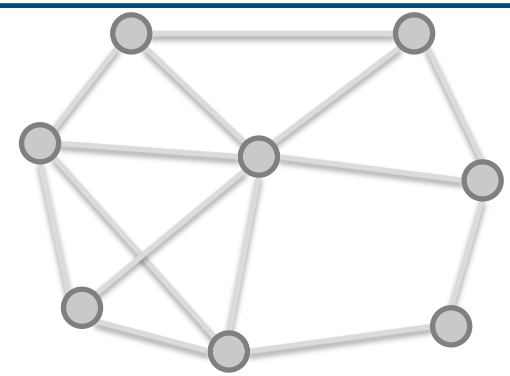
**Random walk methods**
- Walk from vertex to vertex by some criterion, keep edges along the path



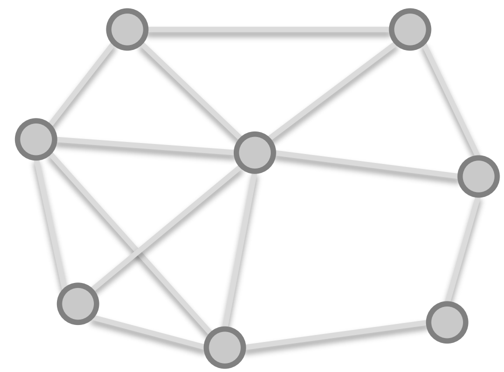
**“Snowball” sampling methods**
- Start with a set of “seed” vertices, grow the sample from those

# Methods Used

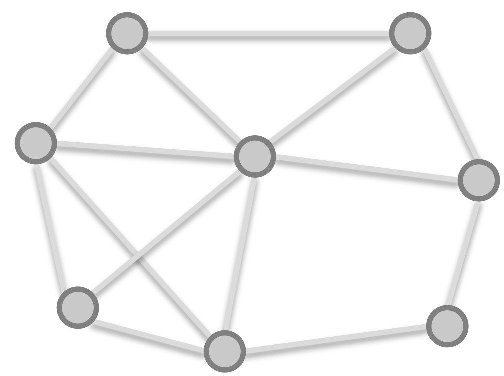
- **Random Edge Sampling**
  - **Each edge in the graph has equal probability of being included in the sample**



- **Popularity-Based Sampling**
  - **Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees**

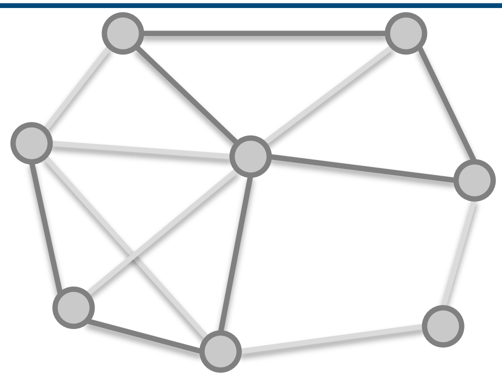


- **Wedge Sampling**
  - **Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample**

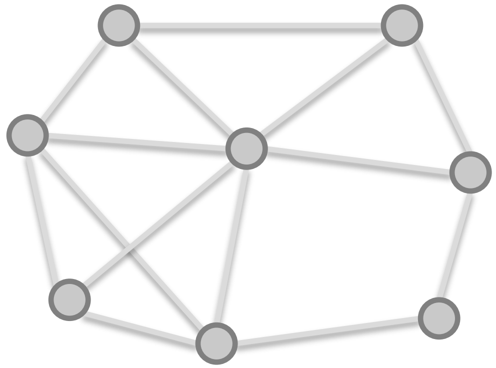


# Methods Used

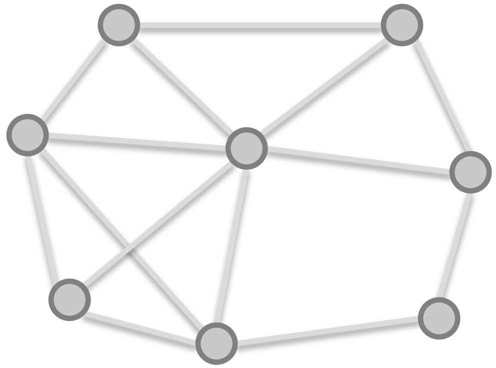
- **Random Edge Sampling**
  - Each edge in the graph has equal probability of being included in the sample



- **Popularity-Based Sampling**
  - Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees

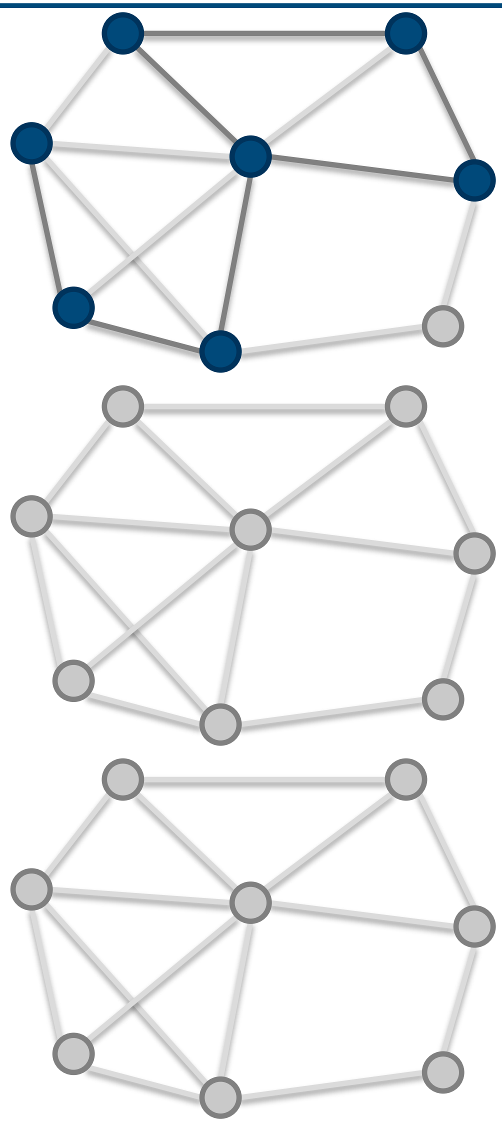


- **Wedge Sampling**
  - Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample



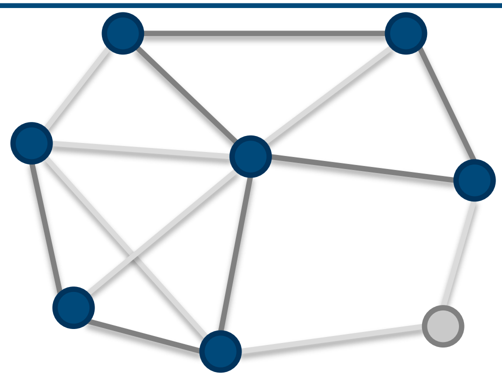
# Methods Used

- **Random Edge Sampling**
  - **Each edge in the graph has equal probability of being included in the sample**
- **Popularity-Based Sampling**
  - **Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees**
- **Wedge Sampling**
  - **Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample**

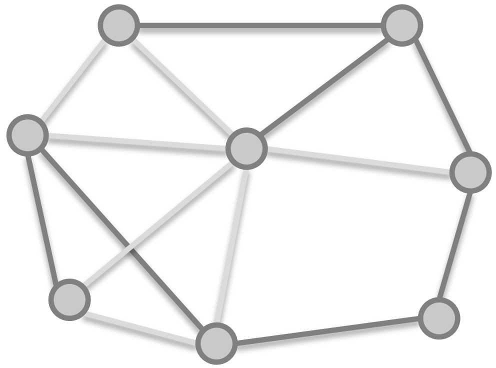


# Methods Used

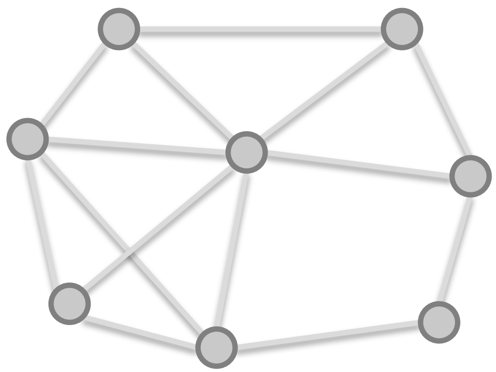
- **Random Edge Sampling**
  - **Each edge in the graph has equal probability of being included in the sample**



- **Popularity-Based Sampling**
  - **Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees**

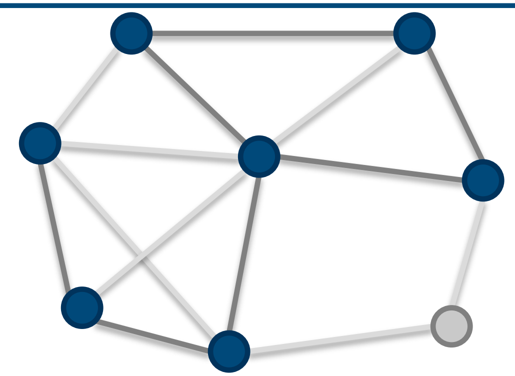


- **Wedge Sampling**
  - **Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample**

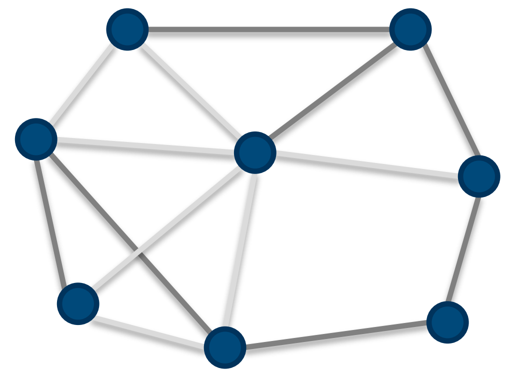


# Methods Used

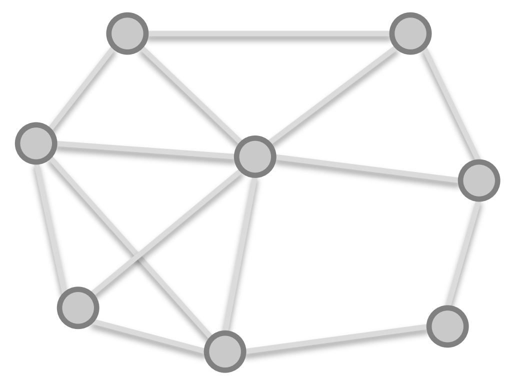
- **Random Edge Sampling**
  - Each edge in the graph has equal probability of being included in the sample



- **Popularity-Based Sampling**
  - Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees

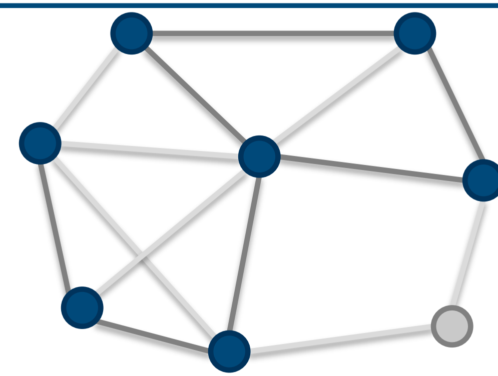


- **Wedge Sampling**
  - Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample

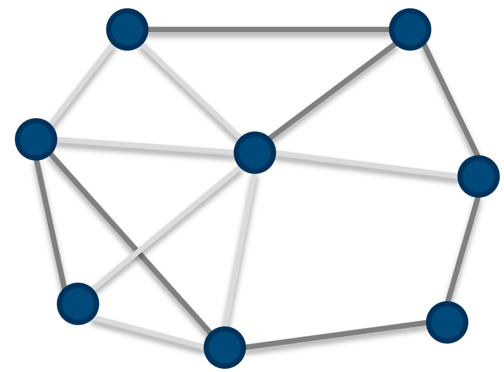


# Methods Used

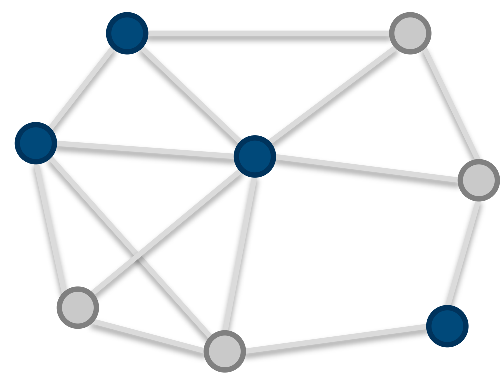
- **Random Edge Sampling**
  - Each edge in the graph has equal probability of being included in the sample



- **Popularity-Based Sampling**
  - Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees

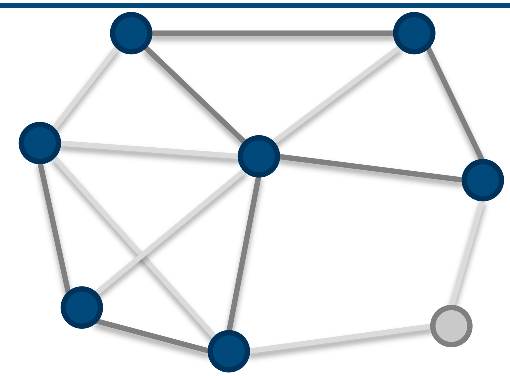


- **Wedge Sampling**
  - Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample

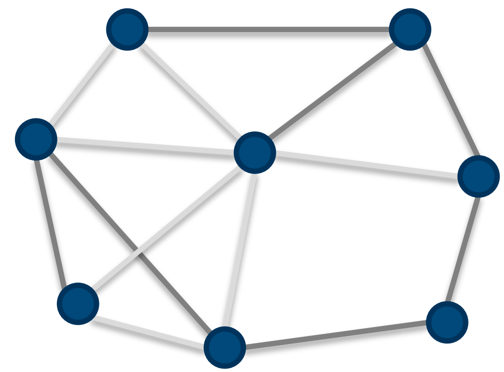


# Methods Used

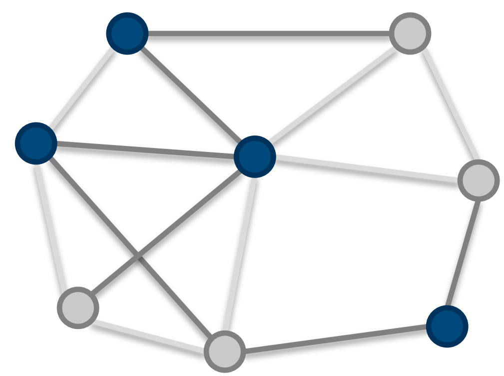
- **Random Edge Sampling**
  - Each edge in the graph has equal probability of being included in the sample



- **Popularity-Based Sampling**
  - Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees

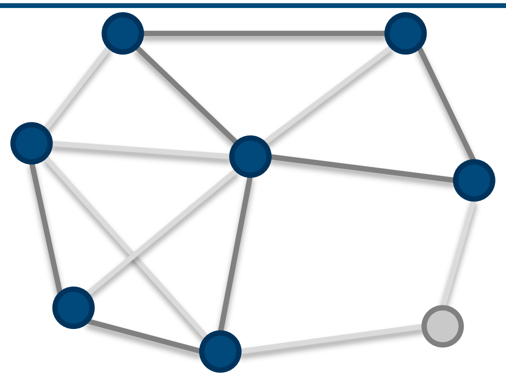


- **Wedge Sampling**
  - Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample

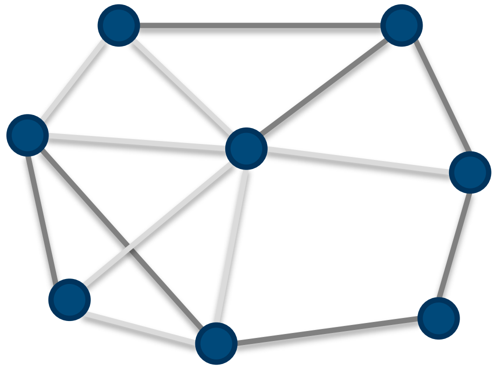


# Methods Used

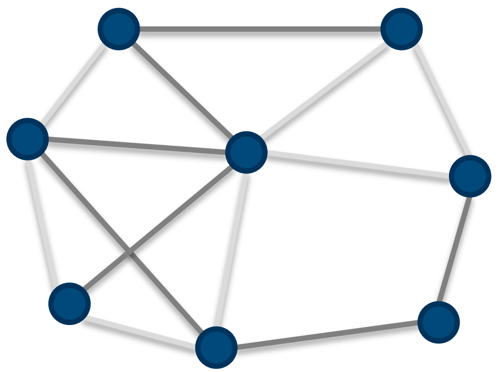
- **Random Edge Sampling**
  - Each edge in the graph has equal probability of being included in the sample



- **Popularity-Based Sampling**
  - Probability of sampling a given edge is proportional to a decreasing function of the vertex degrees

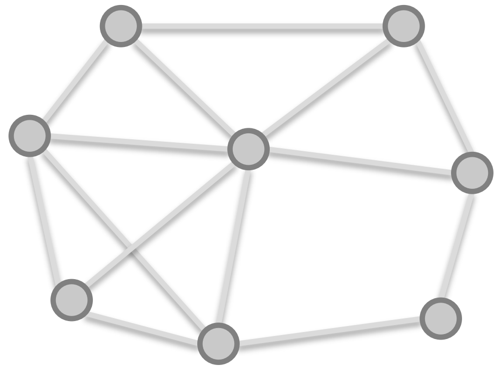


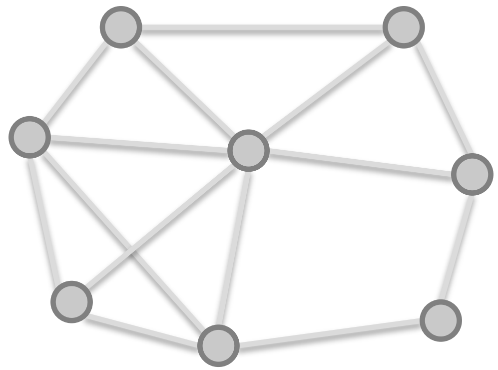
- **Wedge Sampling**
  - Choose a vertex at random, and randomly select two of its adjacent edges to be included in the sample



# Methods Used, Cont'd

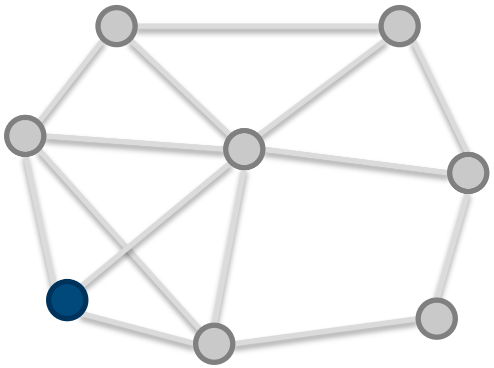
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

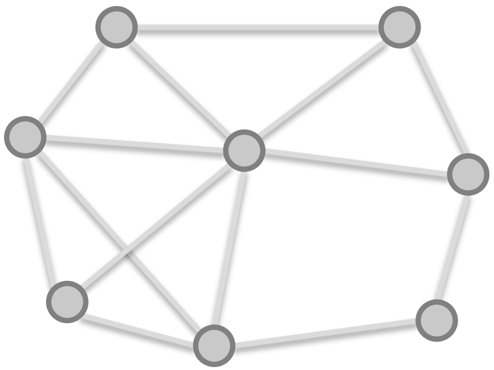




# Methods Used, Cont'd

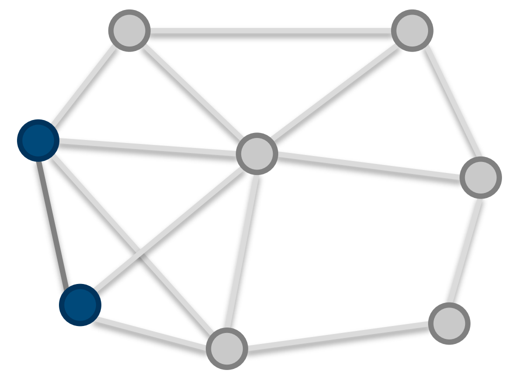
- Random Walk Sampling
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- Random Area Sampling
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

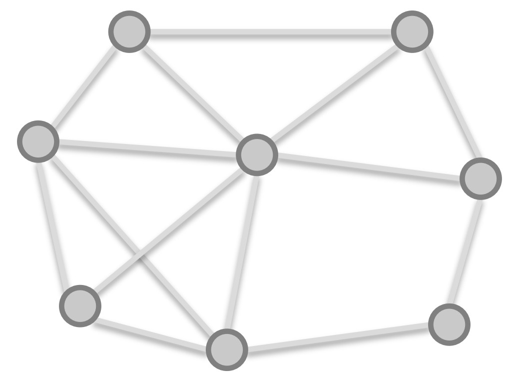




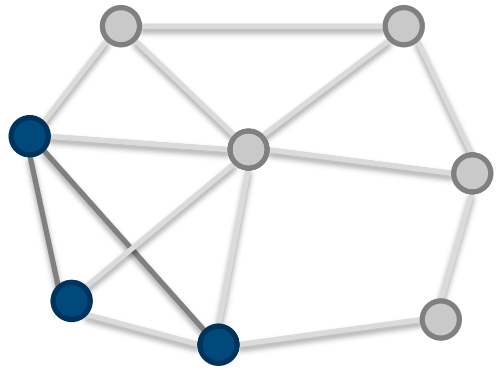
# Methods Used, Cont'd

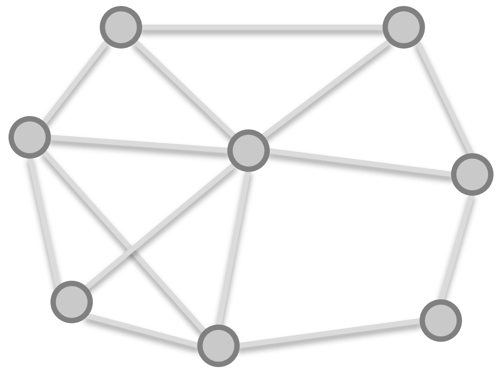
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample





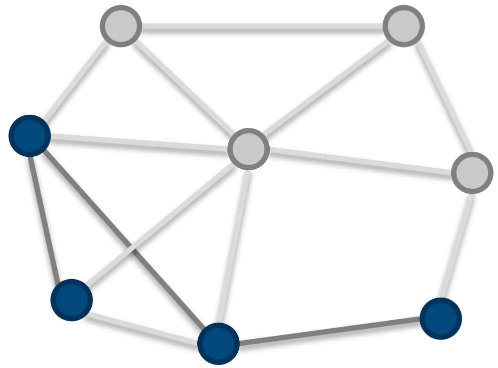
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

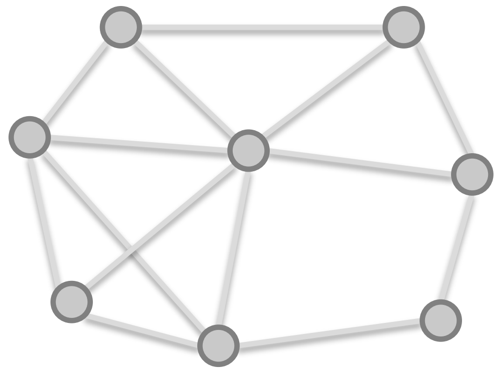




# Methods Used, Cont'd

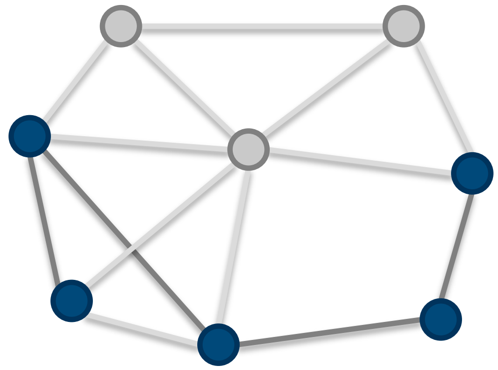
- **Random Walk Sampling**
  - **Move along edges from vertex to vertex, adding edges to the sample along the way**
  - **At each step, with probability 0.15, jump to any vertex in the graph**
- **Random Area Sampling**
  - **Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample**

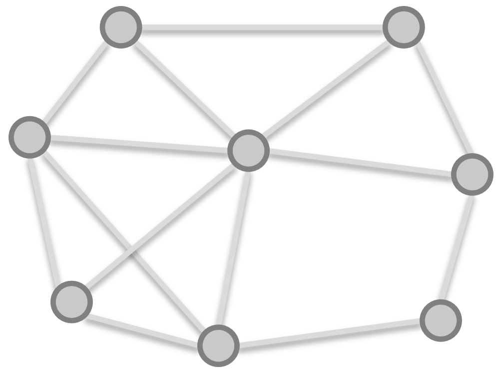




# Methods Used, Cont'd

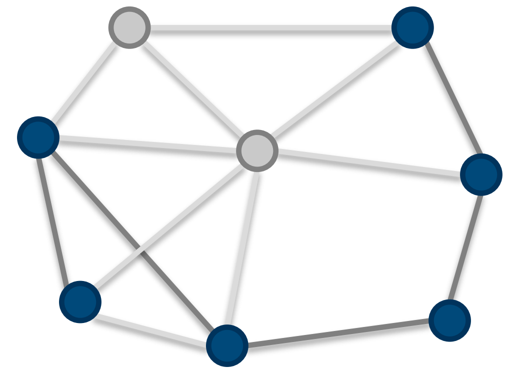
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

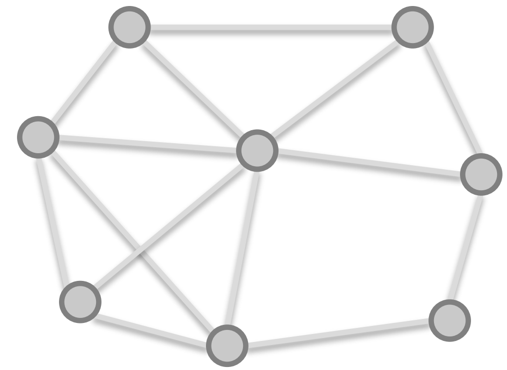




# Methods Used, Cont'd

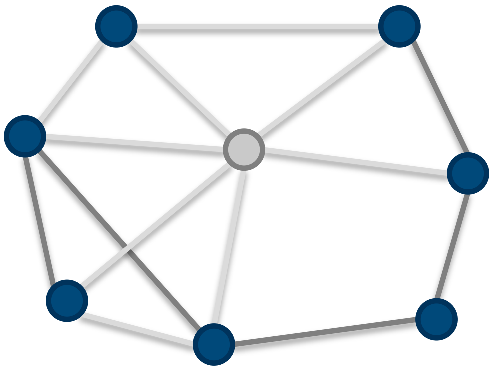
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

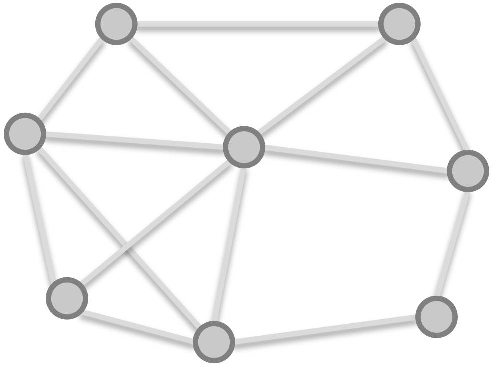




# Methods Used, Cont'd

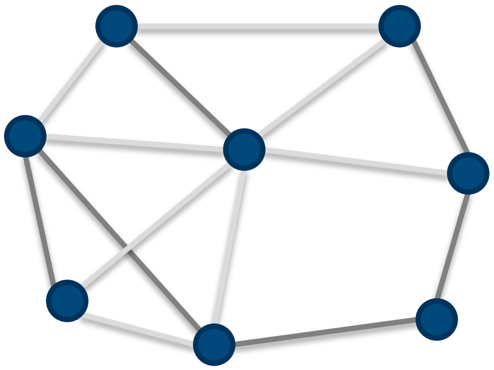
- Random Walk Sampling
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- Random Area Sampling
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

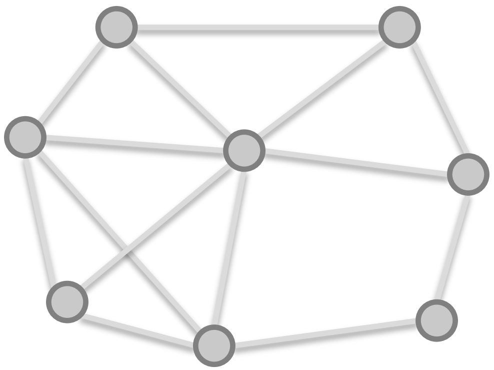




# Methods Used, Cont'd

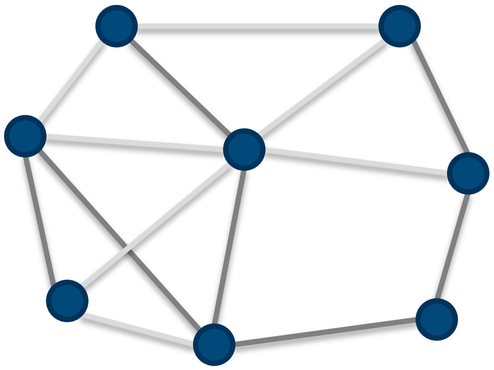
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

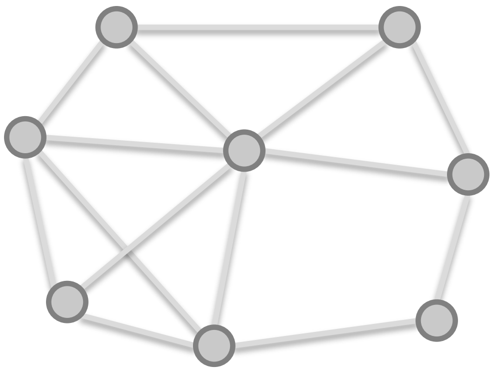




# Methods Used, Cont'd

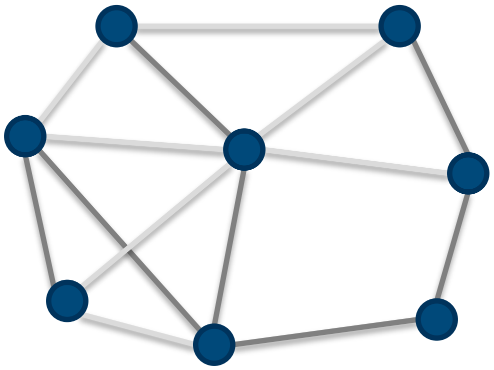
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

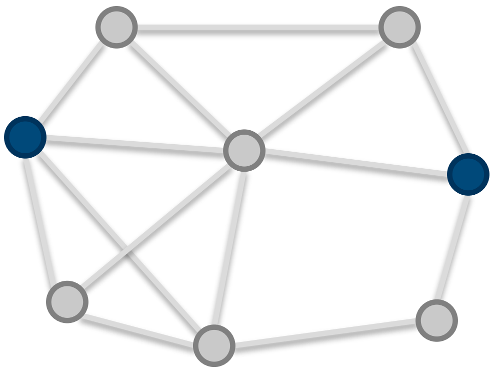




# Methods Used, Cont'd

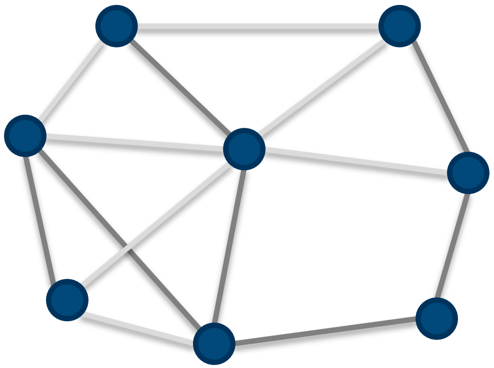
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

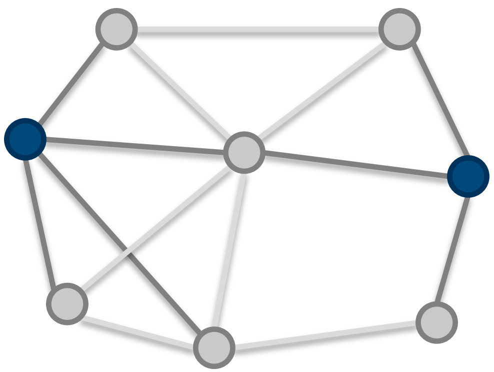




# Methods Used, Cont'd

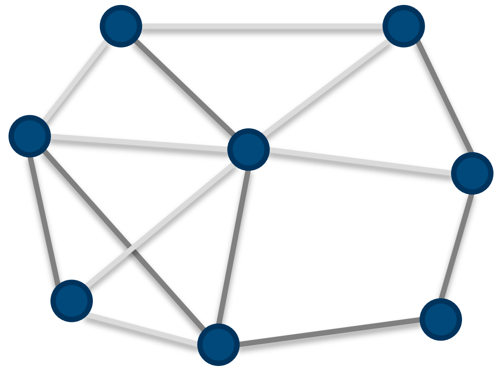
- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample

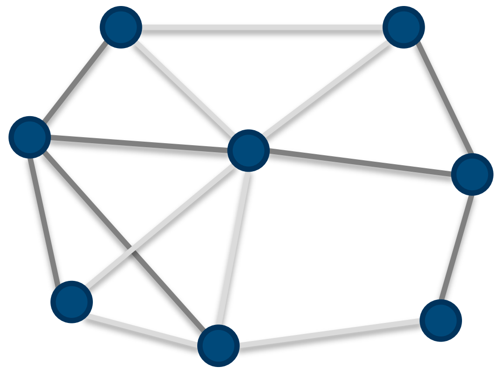




# Methods Used, Cont'd

- **Random Walk Sampling**
  - Move along edges from vertex to vertex, adding edges to the sample along the way
  - At each step, with probability 0.15, jump to any vertex in the graph
- **Random Area Sampling**
  - Choose a set of "seed" vertices, and add all edges adjacent to those vertices to the sample





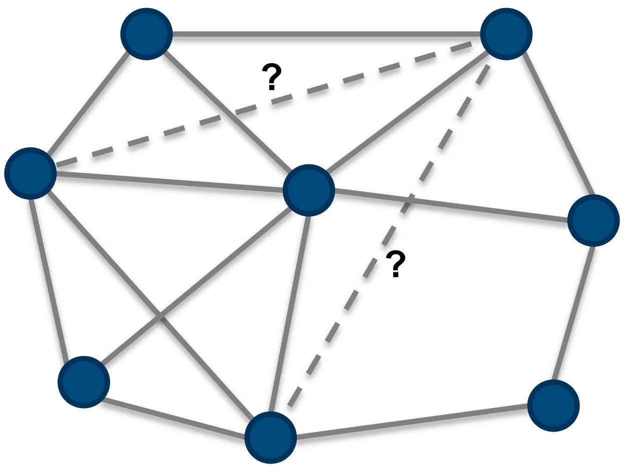
**Selected methods comprise a broad array of network sampling techniques**

# Outline

- Introduction
- Sampling Techniques
- Link Prediction
- Experimental Setup
- Results
- Summary

# Link Prediction

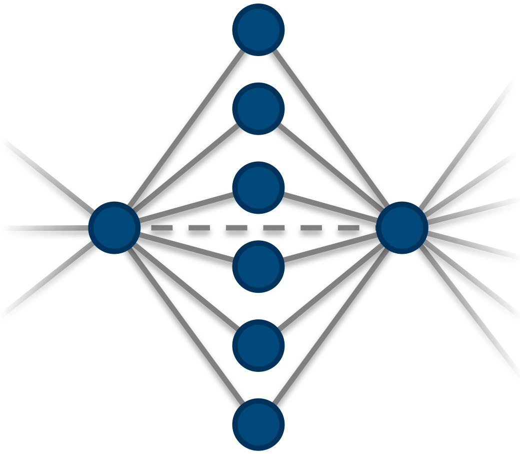
- Scenario: Two vertices currently do not share an edge
- Question: How likely are these vertices to form a connection in the near future?
- Examples:
  - How likely is a connection to be formed between two people in a social network?
  - How likely are two political entities to engage in diplomatic or military conflict?

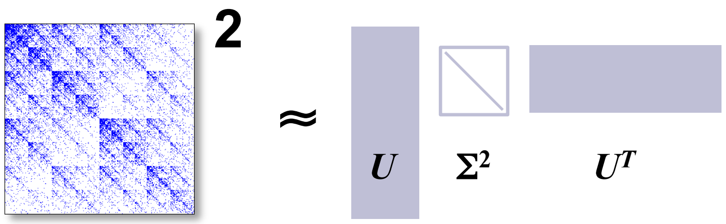


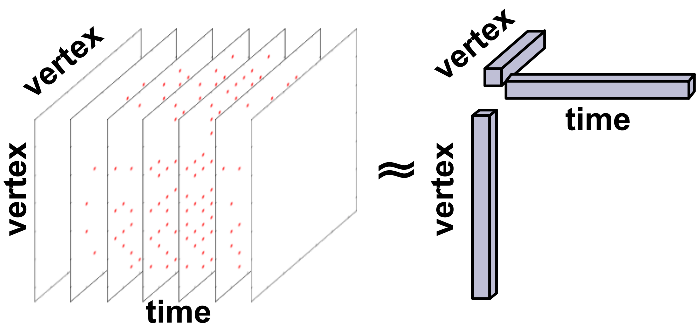

**Link prediction: Anticipation of new connections that currently do not exist**

# Link Prediction Statistics

- **Common neighbors**
  - Vertices with more neighbors in common are more likely to connect
- **Jaccard's coefficient**
  - Normalize common neighbors by total number of neighbors
- **Low-rank approximation**
  - Use a low-rank approximation for the square of the adjacency matrix
- **Tensor decomposition**
  - Break data into vertex-vertex-time factors, extrapolate along temporal axis

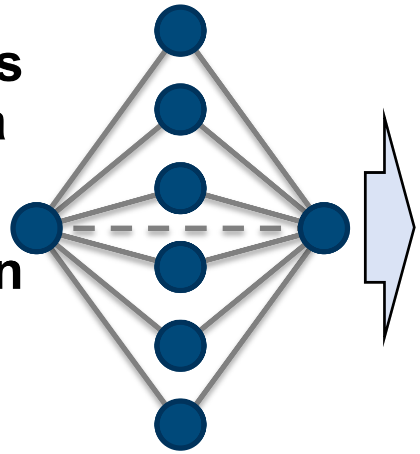


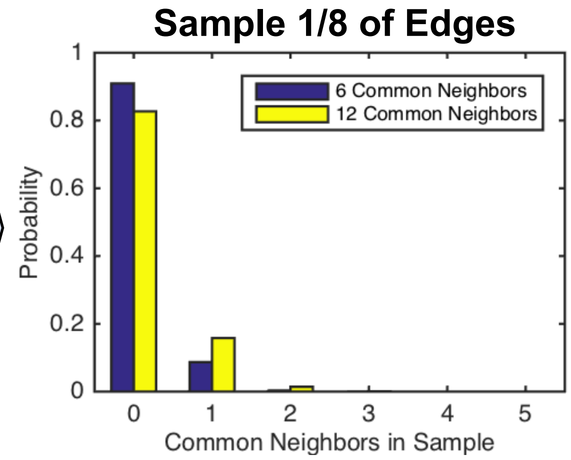


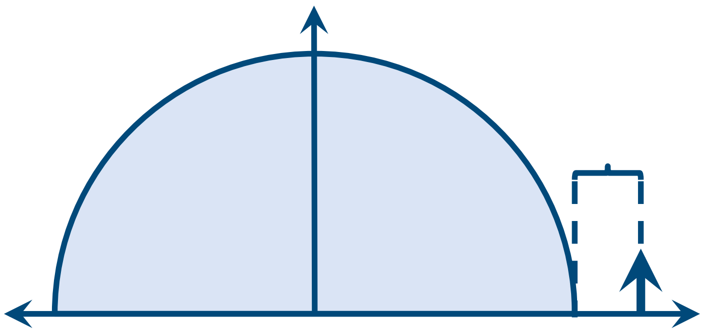
# How Does Sampling Hurt?

- If edge sampling probability is $p$, probability of maintaining a common neighbor is $p^2$
- This "smears" the distribution of common neighbors, inhibiting detection





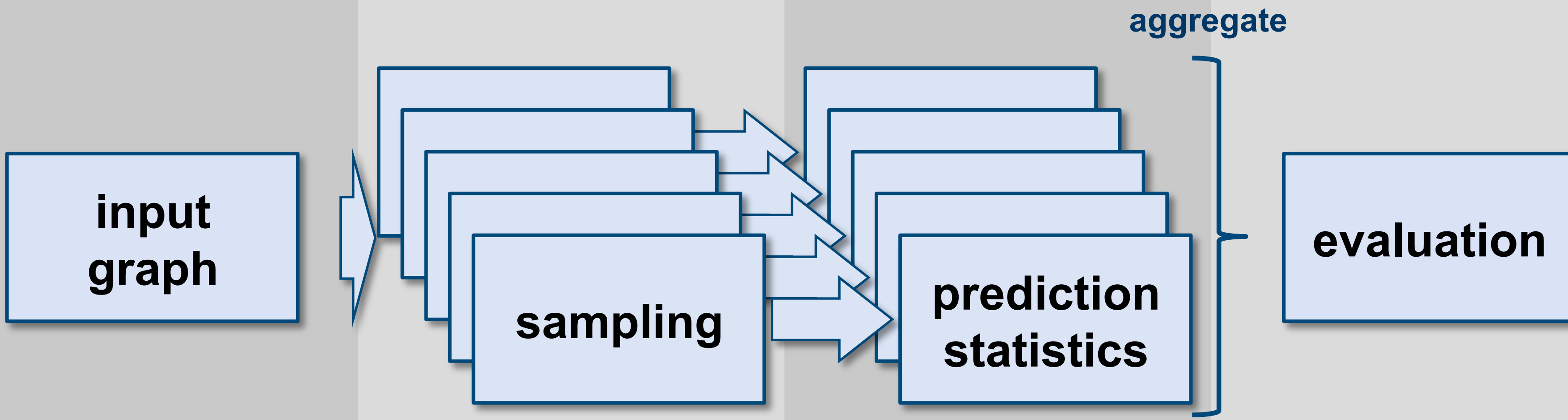

- For eigenvector-based techniques, removing edges causes the distribution of eigenvalues to contract
- The outlier eigenvalues, which provide the most information, contract more quickly



# Outline

- Introduction
- Sampling Techniques
- Link Prediction
- Experimental Setup
- Results
- Summary

# Experiments




**Simulated graph**

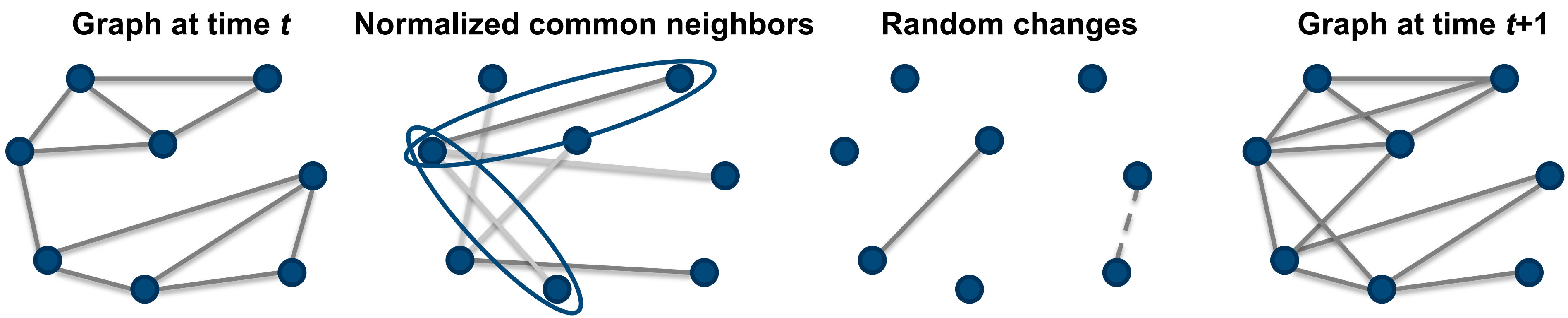
**Method**
- Random edge
- Popularity-based
- Random walk
- Random area
- Wedge

**Amount**
- Factor by which data size should be reduced

**Link Prediction**
- Common neighbors
- Jaccard's coefficient
- Low-rank approximation
- Tensor decomposition

- Prediction performance
- Running time

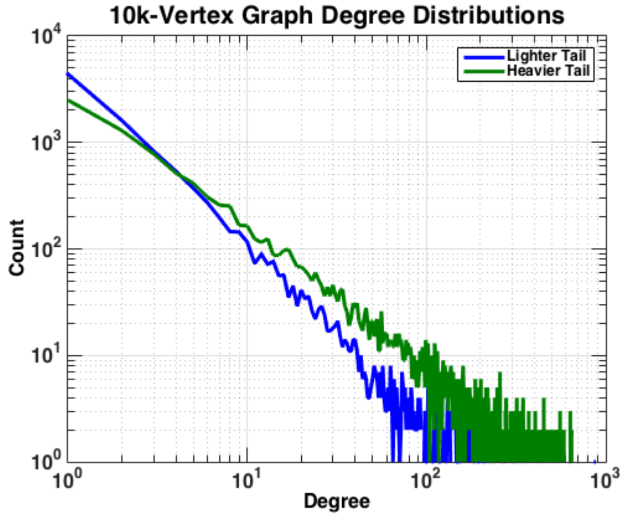
# Simulation Setup




- **BTER Model: Simulate graphs evolving over several time steps**
- **First time step: Create a graph from a generative model**
- **Future time steps:**
  - **Compute common neighbors**
    - **Normalize using cosine similarity**
    - **Randomly select based on scores**
  - **Create new random edges from generative model**
    - **Simulates "chance" encounters**
  - **Randomly remove edges**

# Simulated Graphs

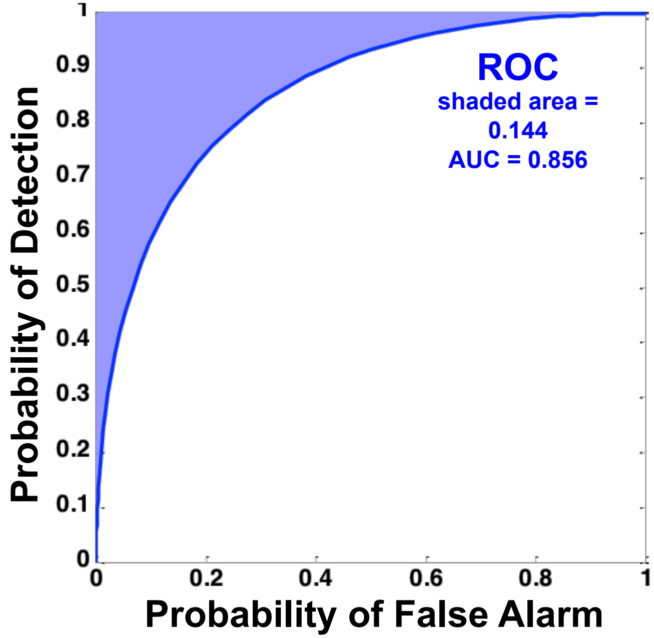
- Approximately 10,000 vertices, average degree of 17, 5% increase in edge count per time step
- Two degree distributions: one lighter tailed and one heavier tailed




**Variations designed to determine most robust sampling methods**

# Performance Metrics

- Area under the ROC curve (AUC)
  - Measure area under the receiver operating characteristic (ROC) curve, which maps false alarm rate to detection rate
- Running time
  - Time required (in Matlab implementation) to compute prediction statistics
- Memory Footprint
  - Storage required for prediction statistics

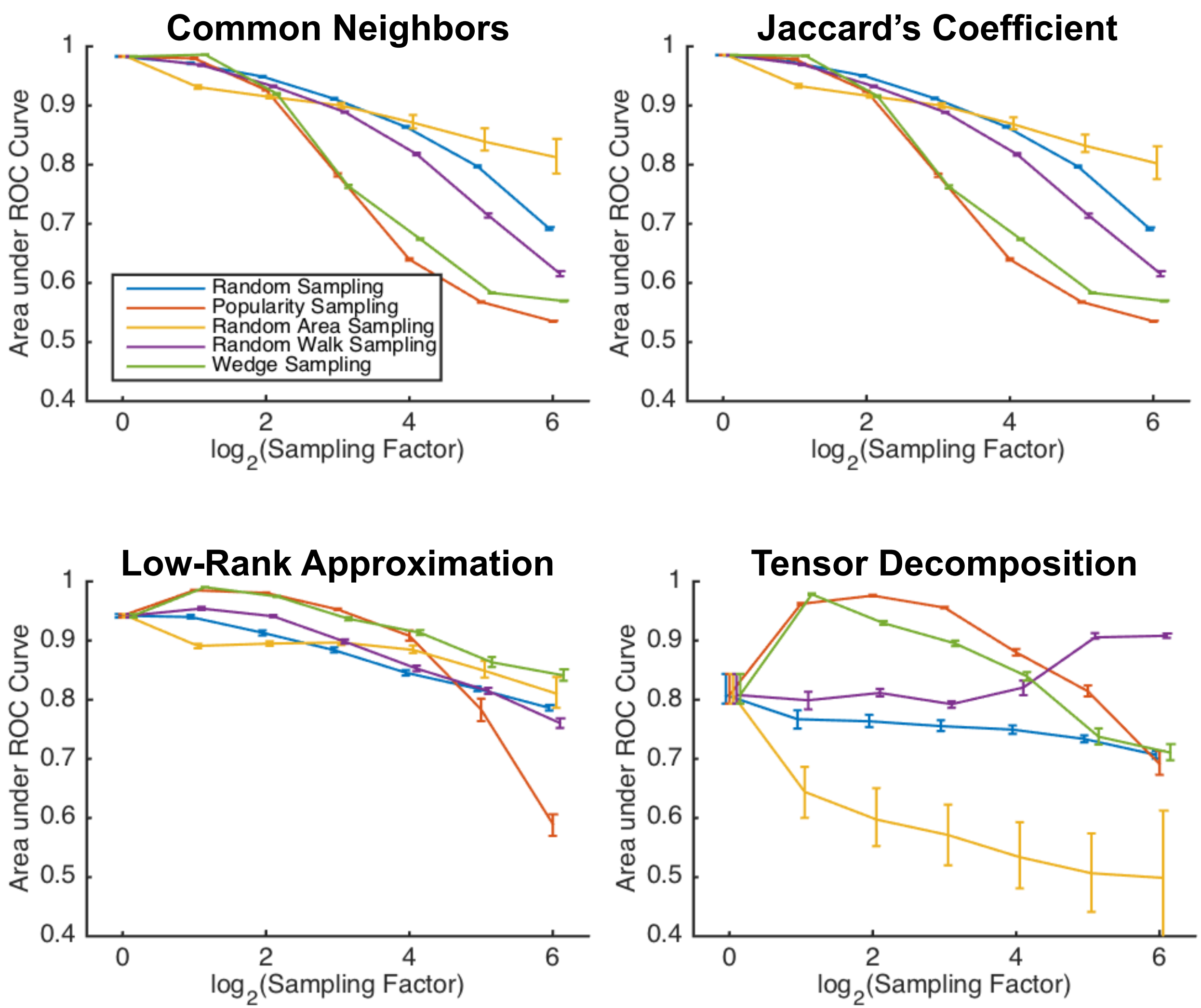



Fundamental metrics for anticipatory network analytics:
- Predictive performance
- Latency
- Computing resource use

# Outline

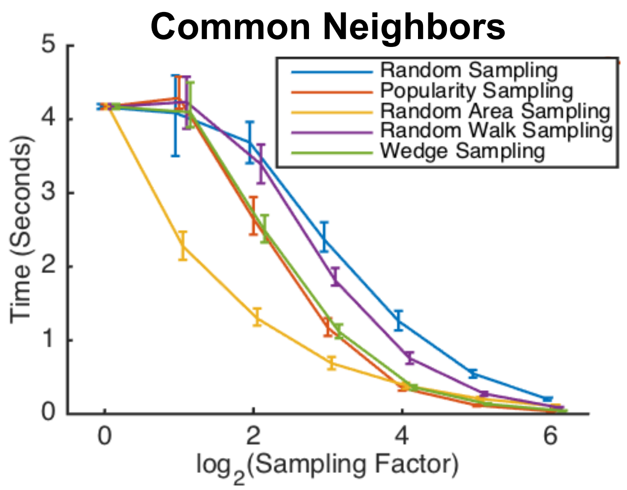
- Introduction
- Sampling Techniques
- Link Prediction
- Experimental Setup
- Results
- Summary

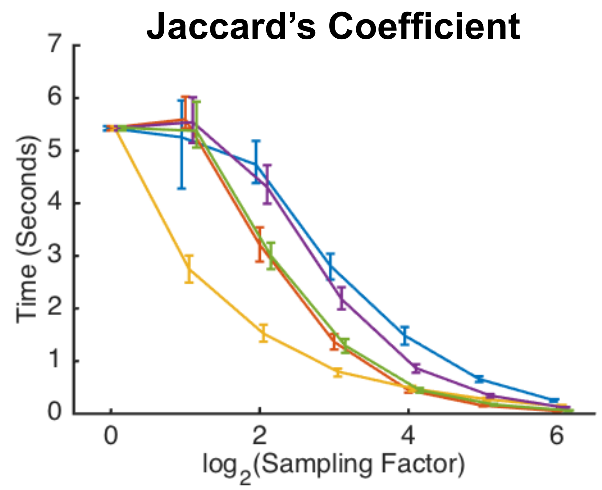
# Link Prediction in Baseline Simulation




- Common neighbors and Jaccard's coefficient are virtually identical
- Random area sampling has the most significant initial performance drop in all cases
- Random walk sampling improves AUC at a higher sampling factor with tensor factorization

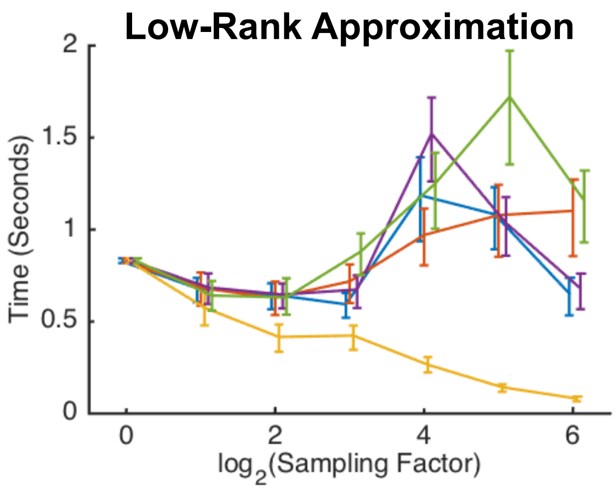
# Time for Link Prediction

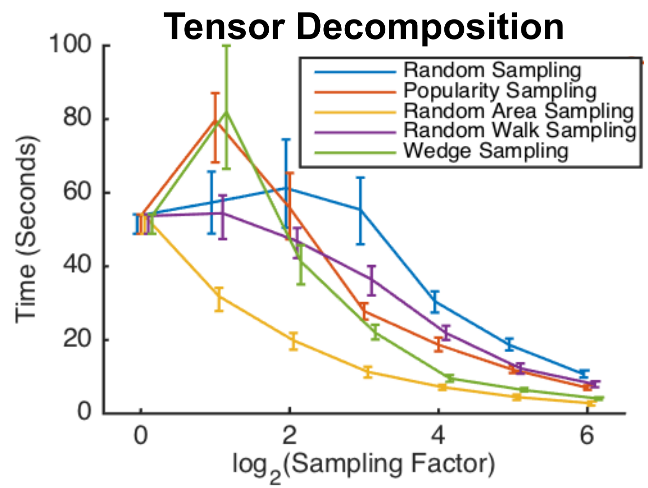



- Common neighbors and Jaccard: time is substantially reduced as sampling becomes more aggressive
- Best initial decrease is with random area sampling
  - However, this comes with the biggest decrease in predictive performance

# Time for Link Prediction




- **Low Rank: time increases with higher sampling rates**
  - **Changes in the distribution of eigenvalues alters convergence rate**
- **Tensor: High sampling rates significantly reduces runtime**
- **Random area sampling produces high number of isolated vertices, reduces the dimensionality of the adjacency matrix**

# Outline

- Introduction
- Sampling Techniques
- Link Prediction
- Experimental Setup
- Results
- Summary

# Summary

- Network analytics are of tremendous importance for anticipatory analytics
  - Forecast future connections
- Networks of interest are often extremely large, and data sampling enables computation on a dataset of a reasonable size
- Study considered the effect of sampling on anticipatory network analytics
  - Random edge sampling is typically the most stable
  - Random area sampling loses performance the fastest
- Promising areas for future work: test on larger simulated and real graph data, investigate emergent community detection, aggregate samples for higher predictive capabilities, integration with high-performance graph analytics hardware

# Backup