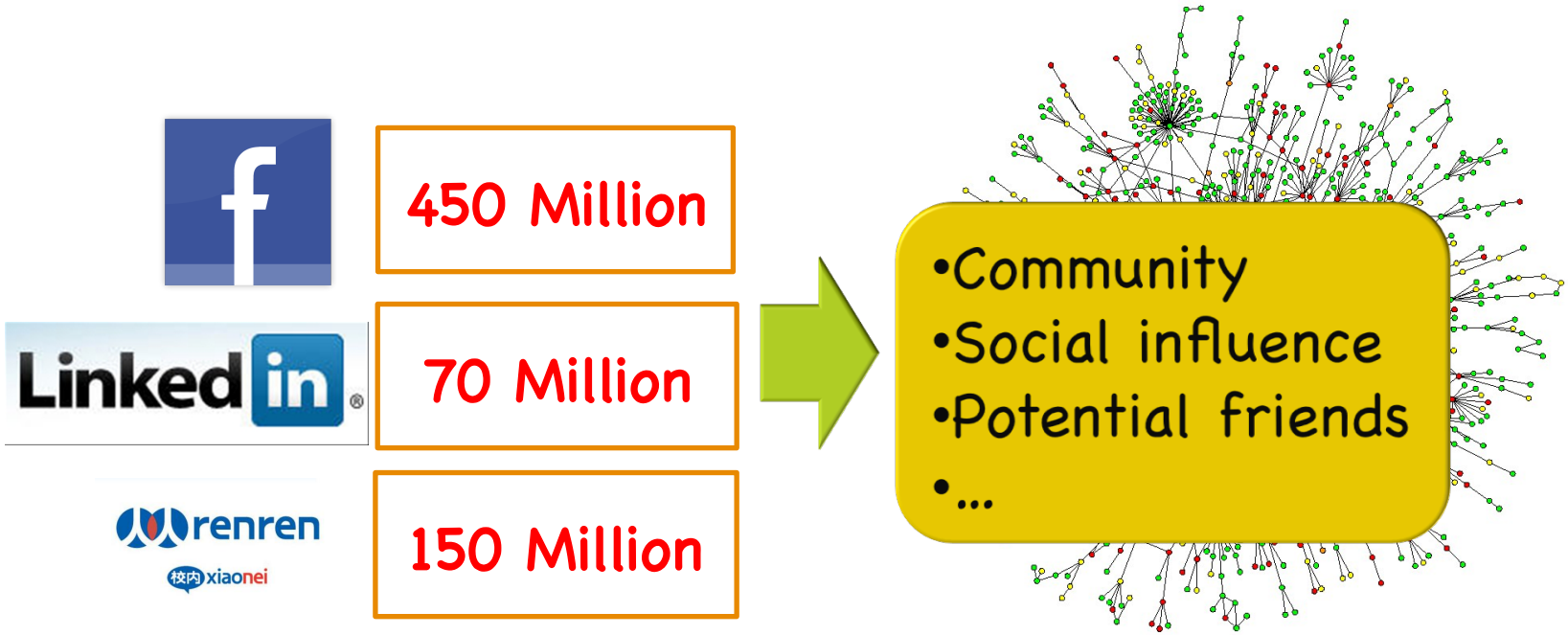
Orion: Shortest Path Estimation for Large Social Graphs

Xiaohan Zhao, Alessandra Sala, Christo Wilson,
Haitao Zheng and Ben Y. Zhao

Department of Computer Science, UC Santa Barbara, USA

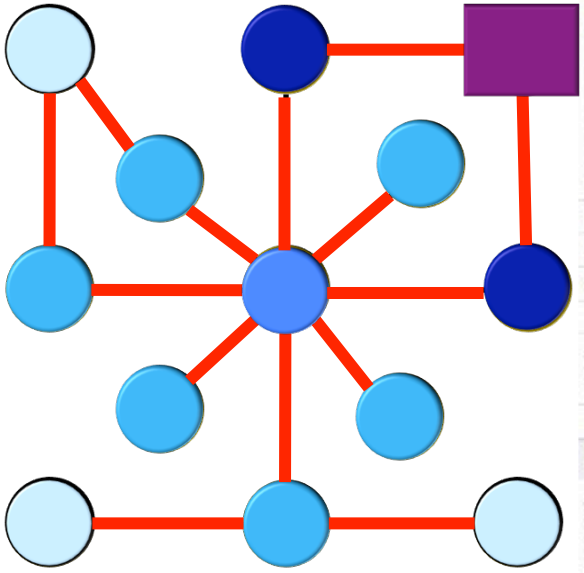
Super Large Social Graphs



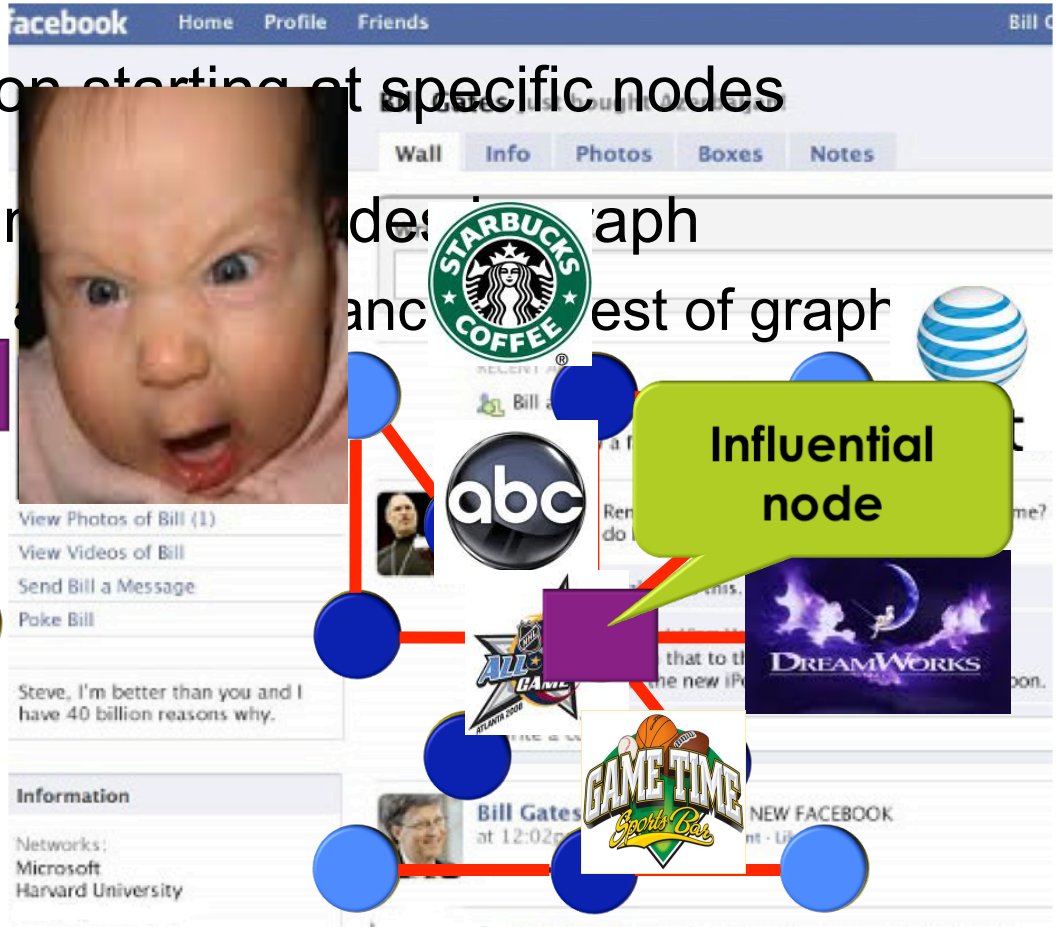
Maximizing Social Influence

- Product advertisement in OSN
  - Bill Gates "likes" Windows Mobile 7

- Propagate information starting at specific nodes
- Goal: find the most in… …des in …aph
  - Nodes with shorter … …anc… …est of graph



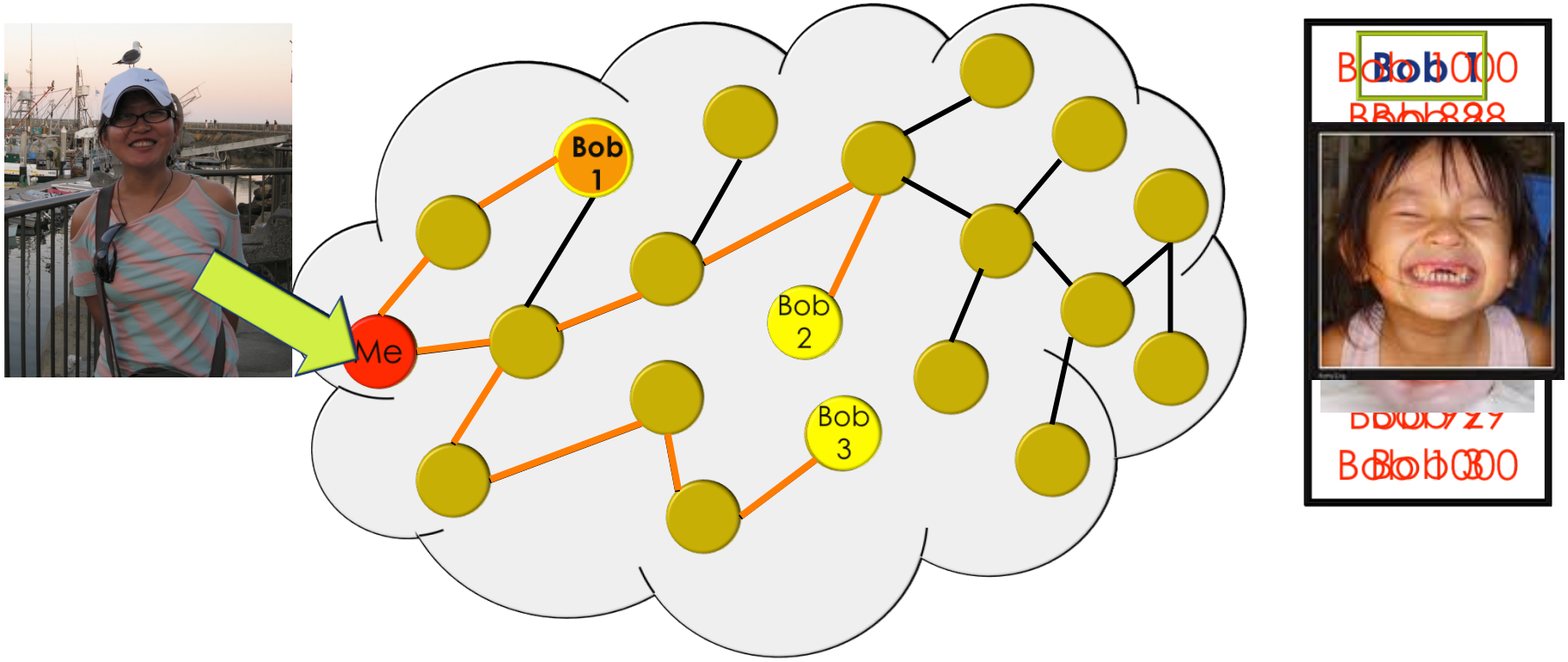
Start @ random node



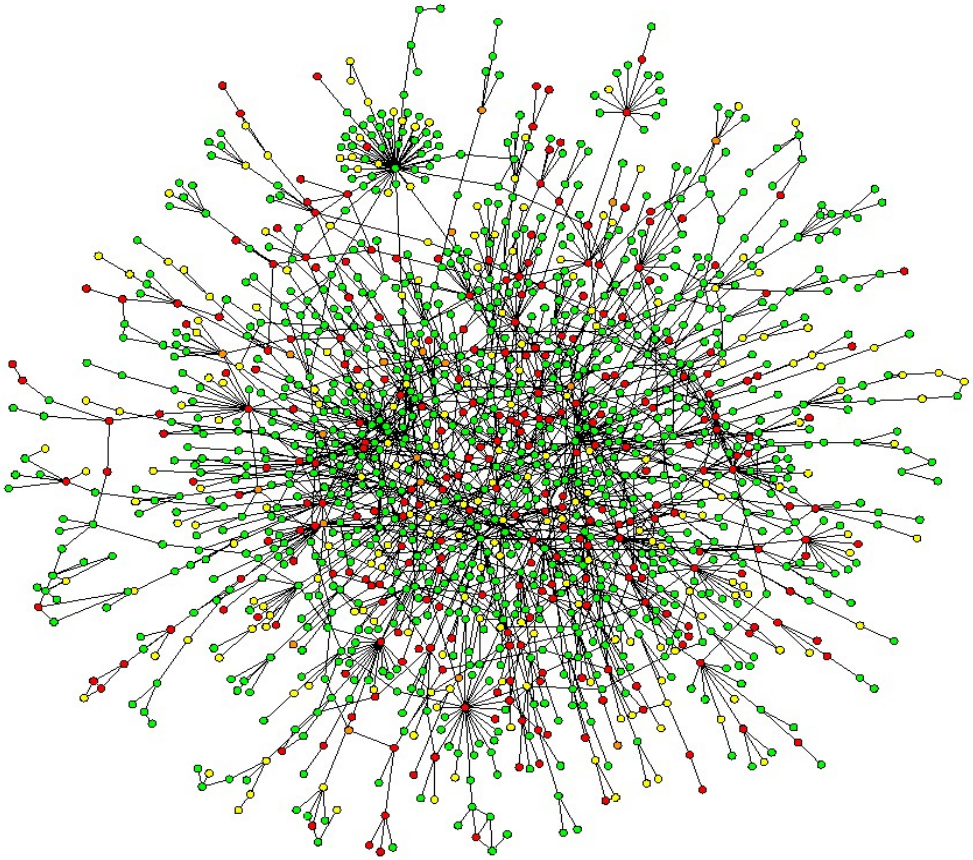
Start @ influential node

Ranked Social Search

- Search for specific friends in social network
  - Rank search results based on the social distances



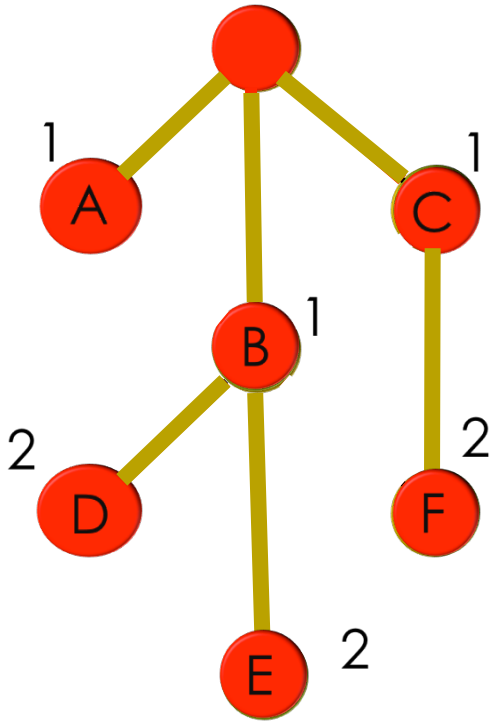
Node distance computation is hard!!



Node Distance Algorithms

For a graph with *n* nodes and *m* edges

Algorithm
Breadth-First Search (BFS)
Dijkstra
Floyd-Warshall



Problem of Node Distance Algorithms

Node distance algorithms
do not scale!

43 Million nodes

+

1 Billion edges

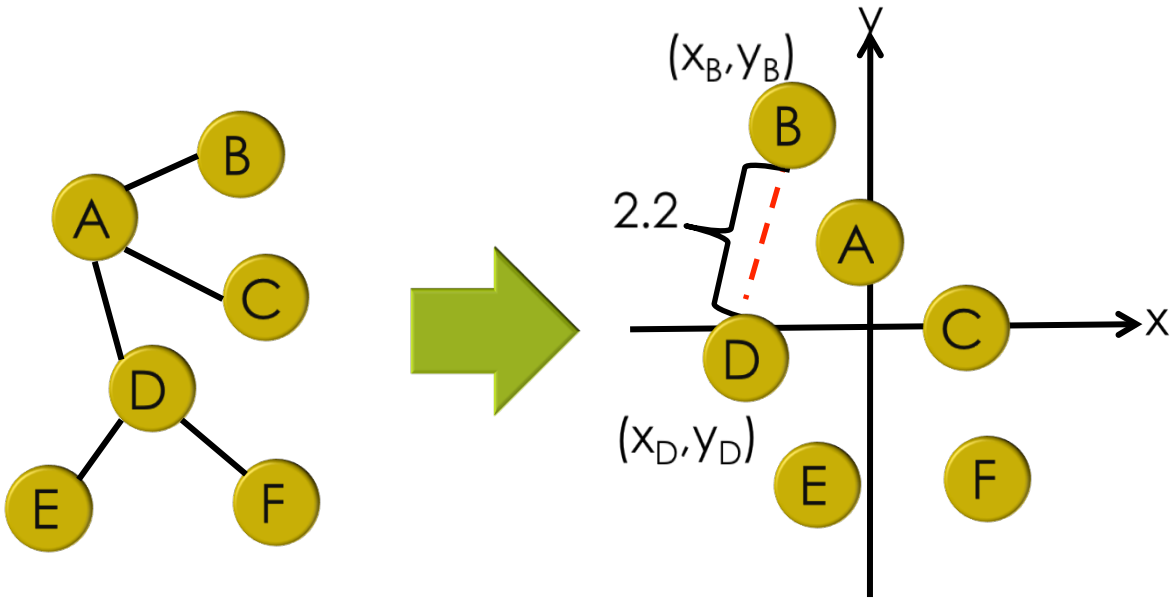
A More Scalable Solution?

- Design a scalable system for large graphs
  - Real-time queries are important
  - Desired query time: O(1)
  - Do preprocessing

- How to achieve O(1) query time?
  - Represent node distance in a graph as distance between two nodes in Euclidean Space
- Map all graph nodes into Euclidean Space
  - A Graph Coordinate System

Orion

- A Graph Coordinate System
  - **Embedding: "Capture"** node distances using Euclidean positions
  - Estimate node distances using coordinates in **constant time**



Outline

- Motivation
- Designing Orion
- Experimental Results
- Using Orion in Graph Applications
- Conclusion

Design Goals of Orion

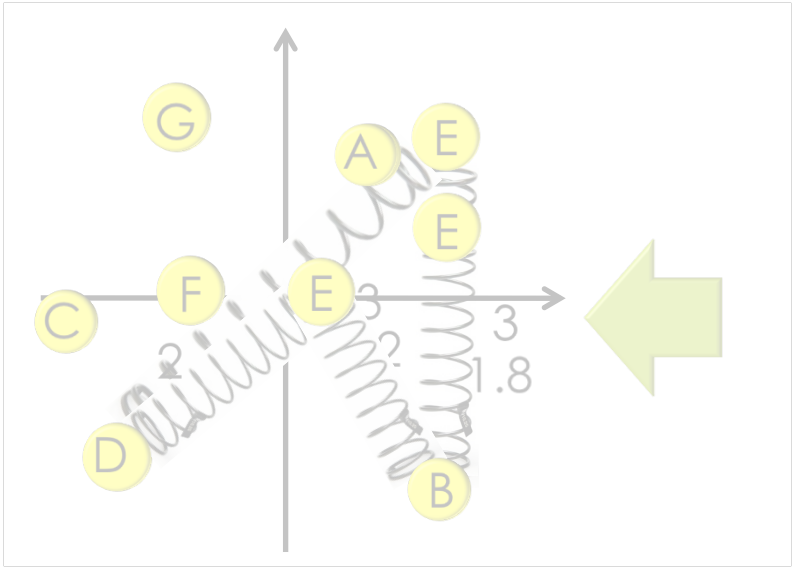
- **Scalability (preprocessing time)**
  - Preprocessing time scales linearly w/ graph size
  - Minimize number of BFS operations
- **Accuracy**
  - Distance estimates approximate ground truth
- **Fast convergence**
  - Individual node calibration should not oscillate

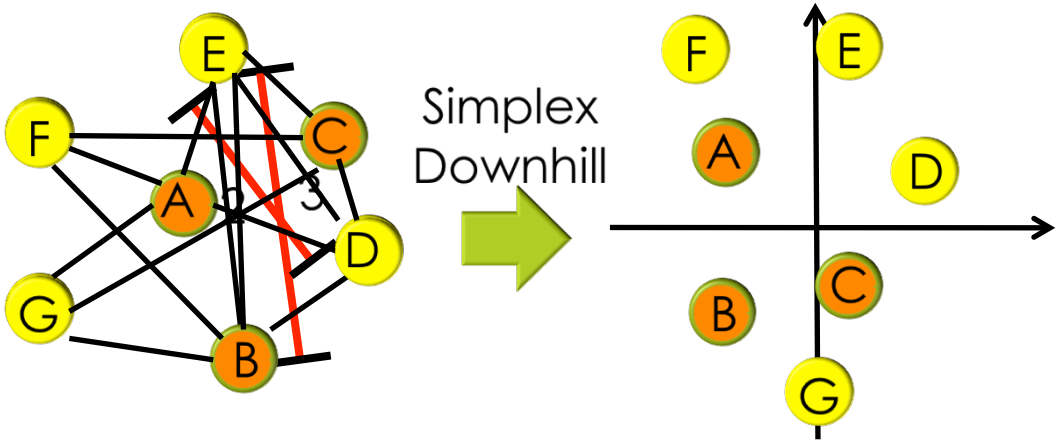
Approaches for Embedding

Our Choice

- Physical spring system
  - Each node needs to do BFS
  - Huge n
  - Multiple
  - Slow convergence
- Landmark-based approach
  - Distances to fixed number
  - onstant number of BFS
  - te once each node
  - Fast convergence

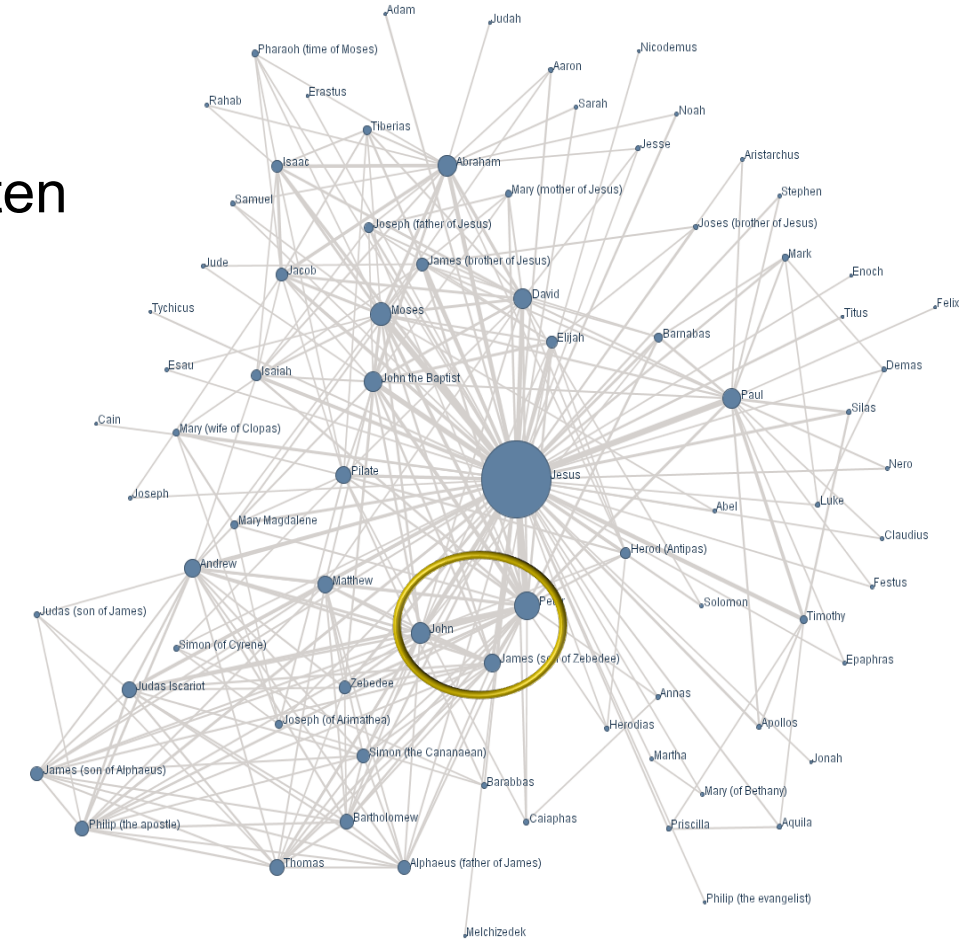
- How to select landmarks?
- How to position landmarks?





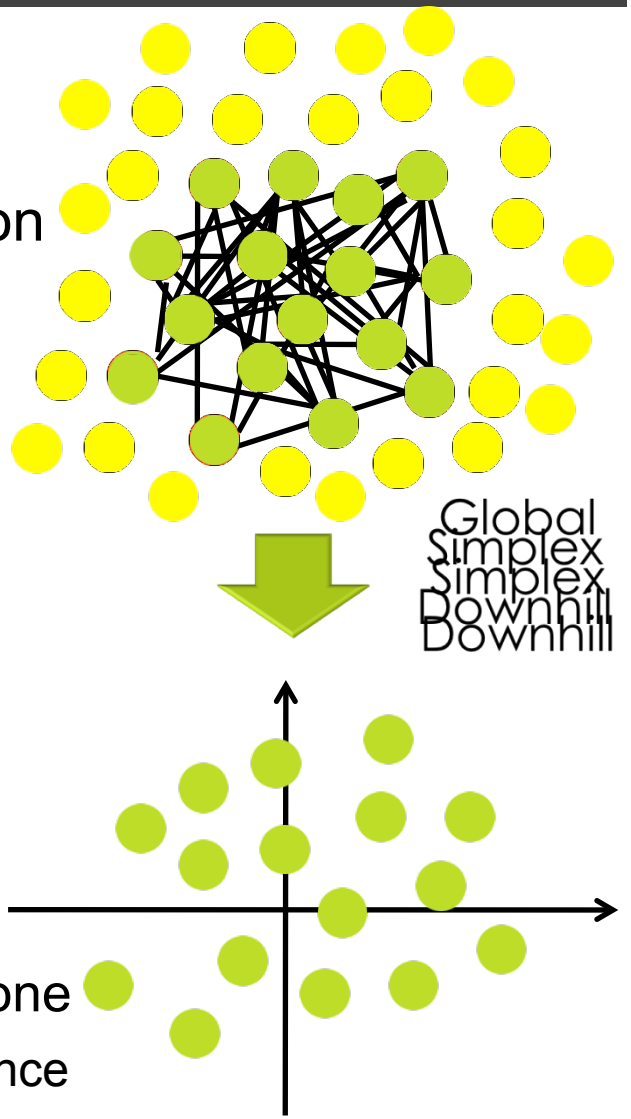
How to Select Landmarks?

- Intuition: highest degree nodes as landmarks
  - "Backbone" of social graph
- Landmark separation
  - Highest degree nodes often connected to each other
  - Need to avoid clusters of landmarks



How to Position Landmarks?

- Naïve solution: Global Simplex Downhill
  - $O(k^2D)$ for k landmarks in D-dimension space
  - However, k can be large for large graphs
- Incremental approach
  - Divide k landmarks into two groups
    - Small initial group L_k (16)
  - Two step computation
    - Initial group: global simplex downhill
    - Remaining landmarks added one by one
      - Use initial landmarks to calibrate distance



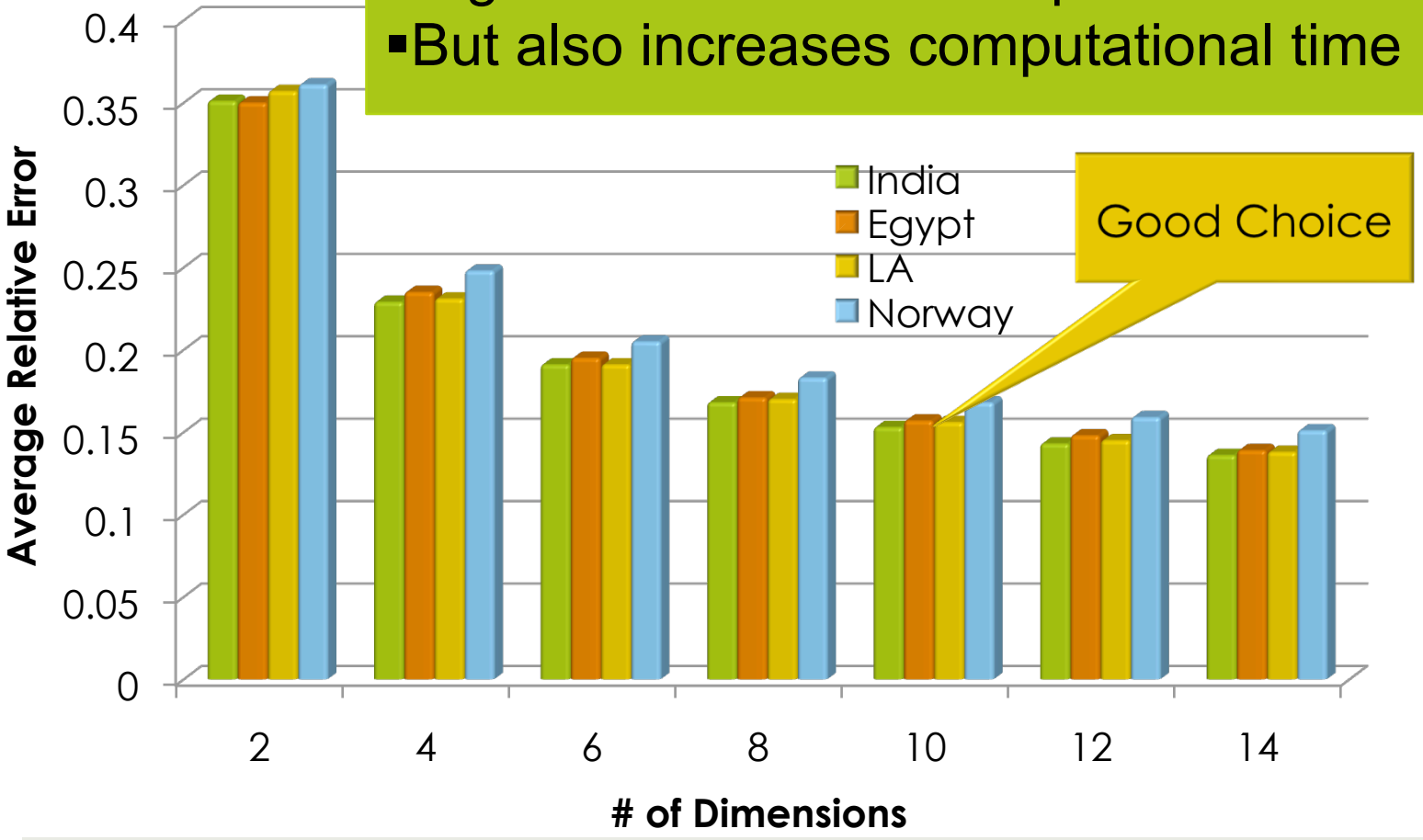
Experimental Setup

- Datasets
  - Four datasets from Facebook regional networks
- Evaluation Metrics
  - Relative Error: $E = \frac{|d^m - d^p|}{d^m}$
    - d^m: actual distance
      d^p: estimated distance computed by Orion
  - Computational Time

Network	Nodes	Edges	Avg. Path Len.
Norway	293K	5,589K	4.2
Egypt	246K	1,618K	5.0
Los Angeles	275K	2,115K	5.1
India	363K	1,556K	6.1

Dimensionality of Coordinates

- Error **< 0.2** when dimension > 6
- Higher dimensions → improved accuracy
- But also increases computational time



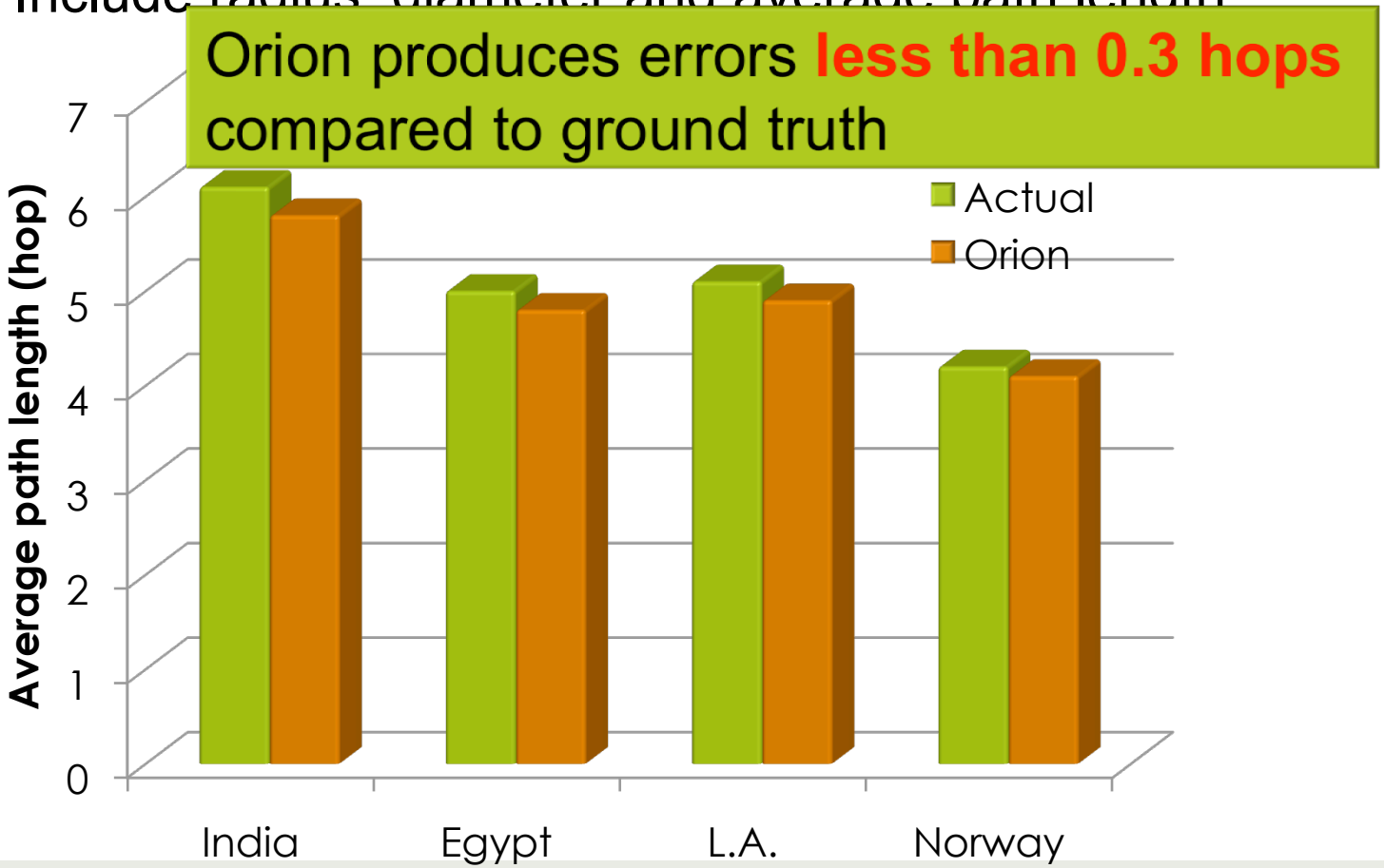
Computational Time

Time	India	Egypt	L.A.	Norway
Orion Preprocessing	9493s	6156s	6967s	7506s
Orion Response	0.0000002s	0.00000002s	0.00000018s	0.00000019s
BFS Response	1.028s	0.75s	1.027s	1.44s

- Orion Preprocessing: to compute coordinates for all nodes
  - One-time cost
  - 2 hours for 300K node graph on 1 cheap commodity server
  - Time scales linearly with graph size
    - Easily parallelized across clusters
- Average time per node-distance query
  - Orion is 7 orders of magnitude faster than BFS

Application: Node Separation Metrics

- Node separation metrics
  - Common tool to analyze graphs
  - Include radius, diameter and average path length



Conclusion

- We propose **Orion**, a scalable graph coordinate system for node distance computation
- Time complexity is low
  - Preprocessing: 2 hours for a 300K node graph
    - Can be parallelized across machine clusters
  - Query Response: 0.2μs to estimate node distances for per query
- Orion can accurately support node-distance based applications

Future / Ongoing Work

- Dynamics in social graphs
  - Investigate the impact of graph dynamics on node distances
  - Use heuristics to incrementally update graph embeddings at run time
- Weighted graphs
  - Examine the use of graph coordinate systems on applications on weighted graphs

Thank You. Questions?