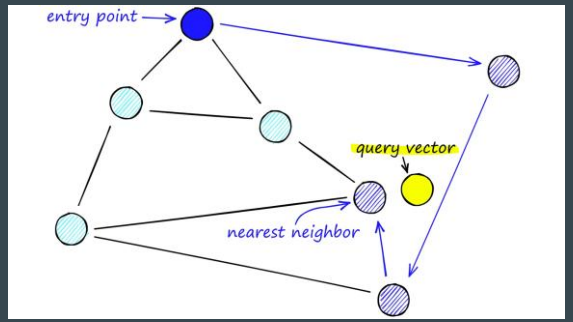
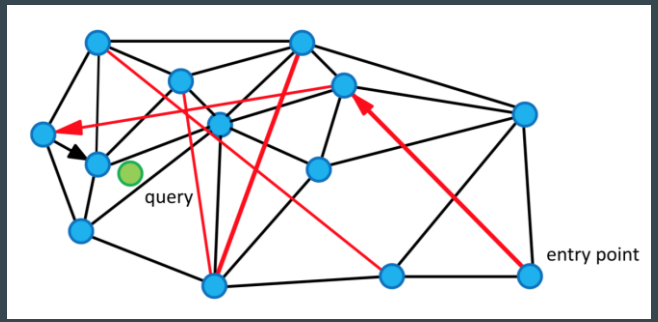
Distance Adaptive Beam Search for Provably Accurate Graph-Based Nearest Neighbor Search

Yousef Al-Jazzazi, Haya Diwan, Jinrui Gou,
Christopher Musco, Cameron Musco, Torsten Suel

Nearest Neighbor Search (NNS)

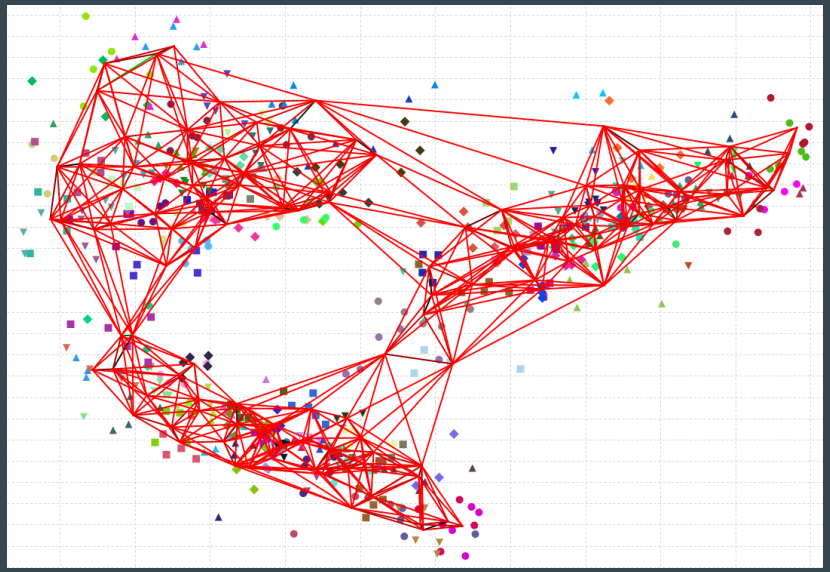
- Objective: Find the closest k points to a query in a dataset
- Applications: recommendation systems, image/music recognition, etc.
  - Very hard to solve exactly in high dimension
  - Lots of interest in fast approximate methods based on techniques like locality sensitive hashing, product quantization, etc.
- Graph-based methods dominate recent benchmarks
  - Hierarchical Navigable Small World (HNSW, 2016), Disk-based Approximate Nearest Neighbor (DiskANN/Vamana, 2019), Navigating Spreading-out Graph (NSG, 2019)





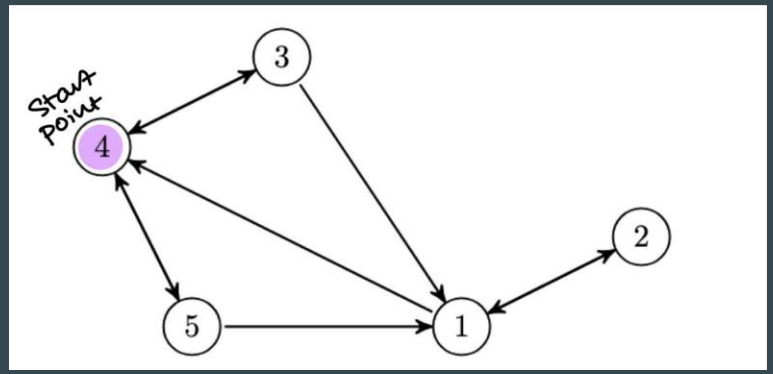
Motivation

- Graph-based ANN methods (HNSW, DiskANN/Vamana, NSG) perform excellently in practice. However, they lack provable performance guarantees



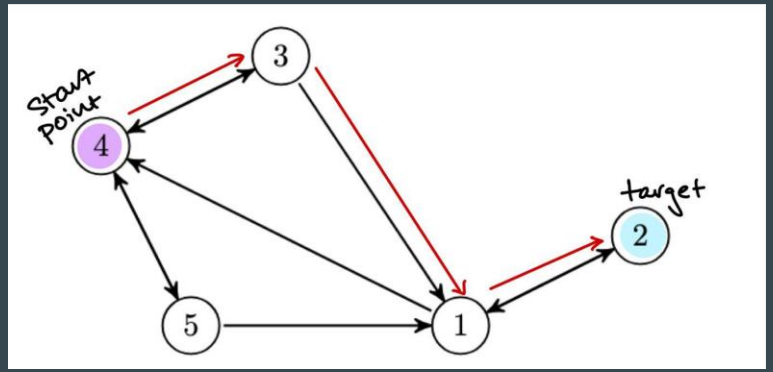
Navigable Graphs

- To analyze search performance, we need to study the properties of search graphs
  - Most common property in ANNS methods: navigability
- Definition: A directed graph G is navigable if for any source node i and target node j ≠ i, there is a node k among i's neighbors that is closer to j.
  - We can greedily move towards the target in each step
- Guarantees monotonic progress toward any target point in the data set



Navigable Graphs

- To analyze search performance, we need to study the properties of search graphs
  - Most common property in ANNS methods: navigability
- Definition: A directed graph G is navigable if for any source node i and target node j ≠ i, there is a node k among i's neighbors that is closer to j.
  - We can greedily move towards the target in each step
- Guarantees monotonic progress toward any target point in the data set



Approximate Nearest Neighbor

- Solving exact nearest neighbor is always not feasible in practice
- Most graph-based search algorithms return approximate nearest neighbors
- Returned nodes have distance within the constant factor of the optimal solution

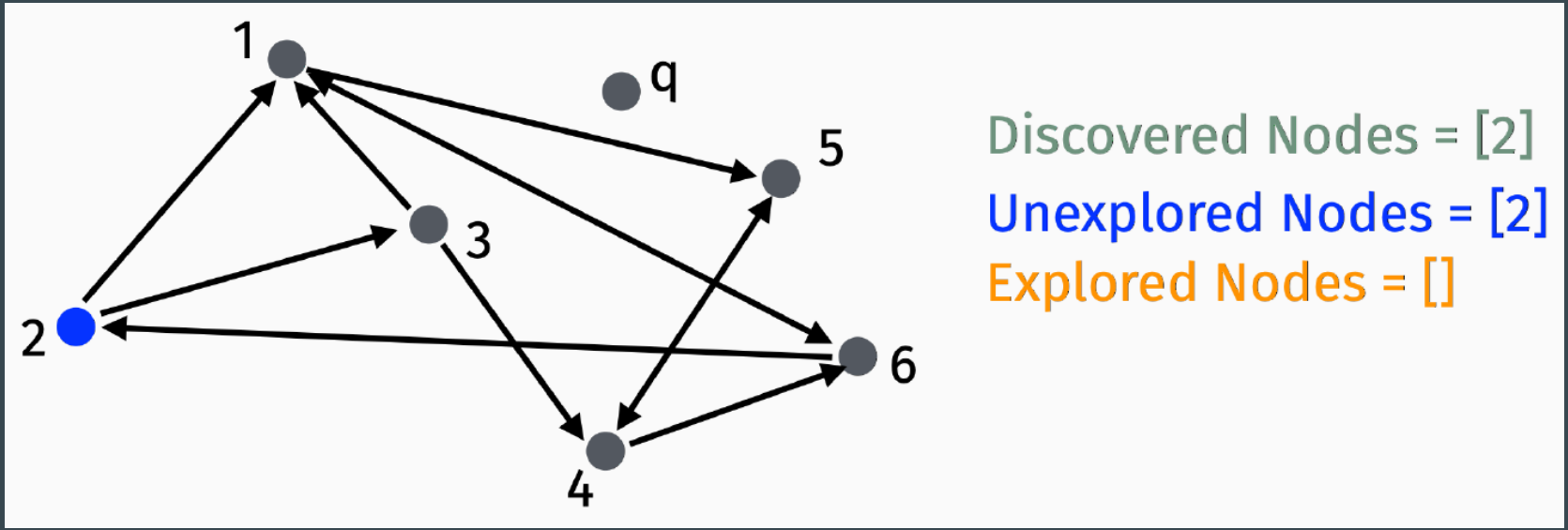
Can greedy search provide good approximate nearest neighbors when running on navigable graphs?

Beam Search

- Greedy search + Navigability doesn't guarantee ANN performance
  - Only ensures we can reach exact target that is already in the graph. However in practice, query point is usually not in the dataset.
  - Greedy search can fail to find good neighbors in some cases.
- Stronger variant of greedy search: beam search
  - With beam-width b, maintain $b \geq k$ candidate nodes in a priority queue C (unexplored nodes), also keep track of a set D (discovered nodes) that stores the nodes have been discovered
    - Always explore closest unexplored candidate node in C in each step
    - Stop when there are at least b nodes in D is closer to q than current exploring node

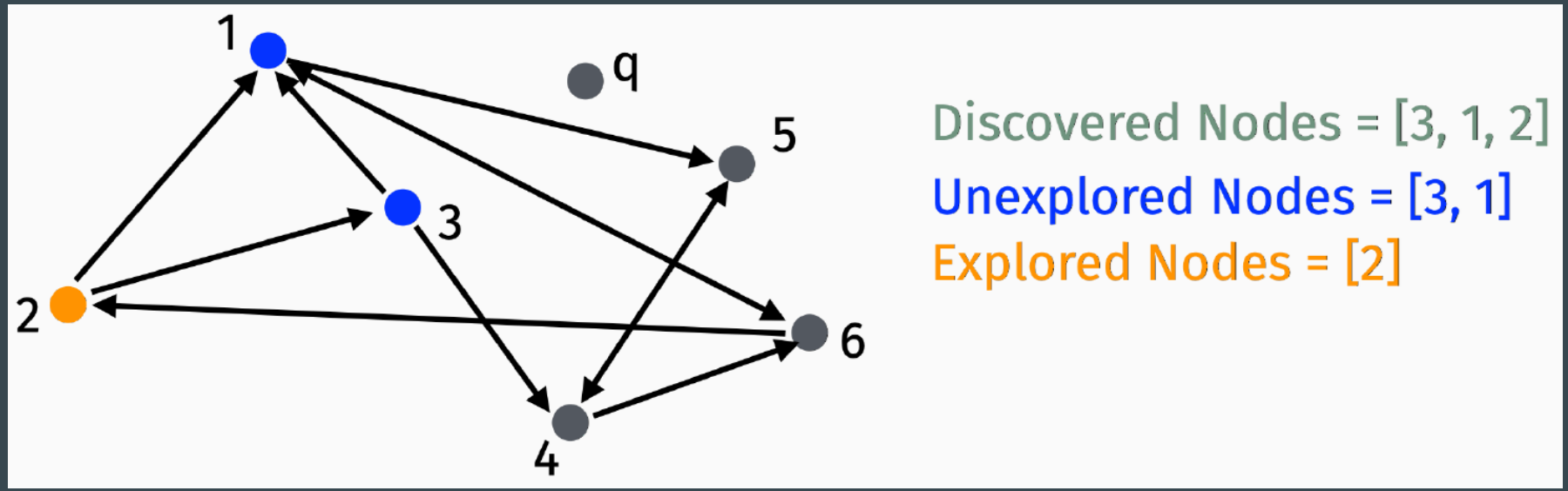
Beam Search (Example)

- Every time we explore a node, we compute the distance between q and all of the node's neighbors



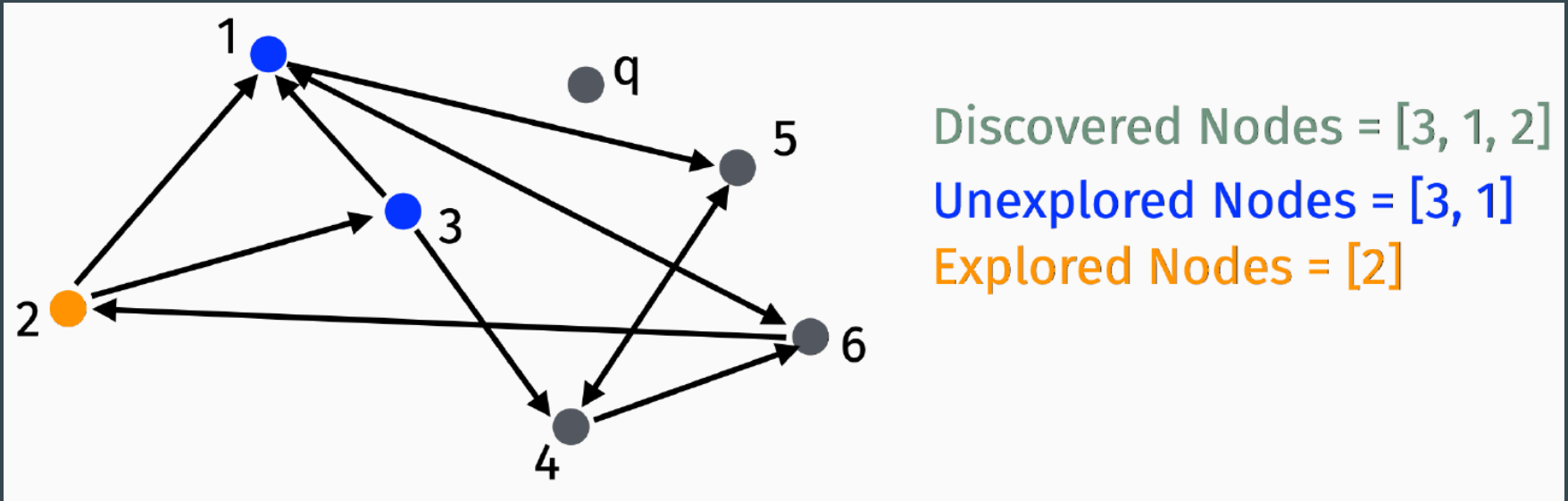
Beam Search (Example)

- Every time we explore a node, we compute the distance between q and all of the node's neighbors



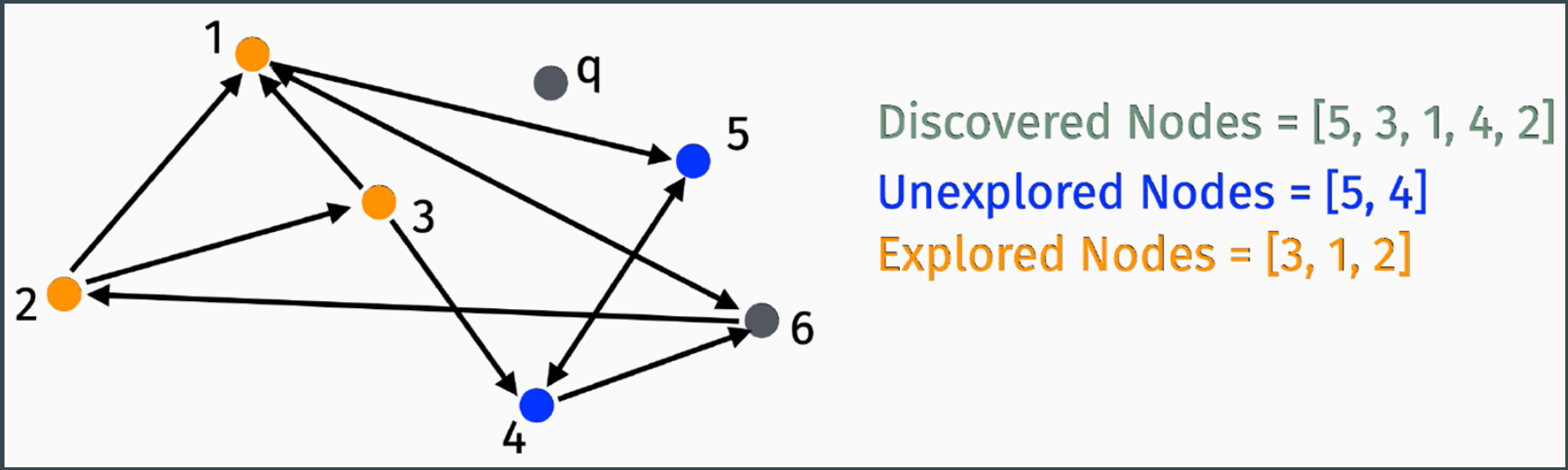
Beam Search (Example)

- Every time we explore a node, we compute the distance between q and all of the node's neighbors



Beam Search (Example)

- Once the search terminates, return the k best results in Discovered Nodes.



Is Beam Search Good Enough?

- We prove lower bound shows that for beam search, w must be set to n to find good approximate nearest neighbors in worst case.
- This means the search will scan whole dataset, which necessitates new search method.

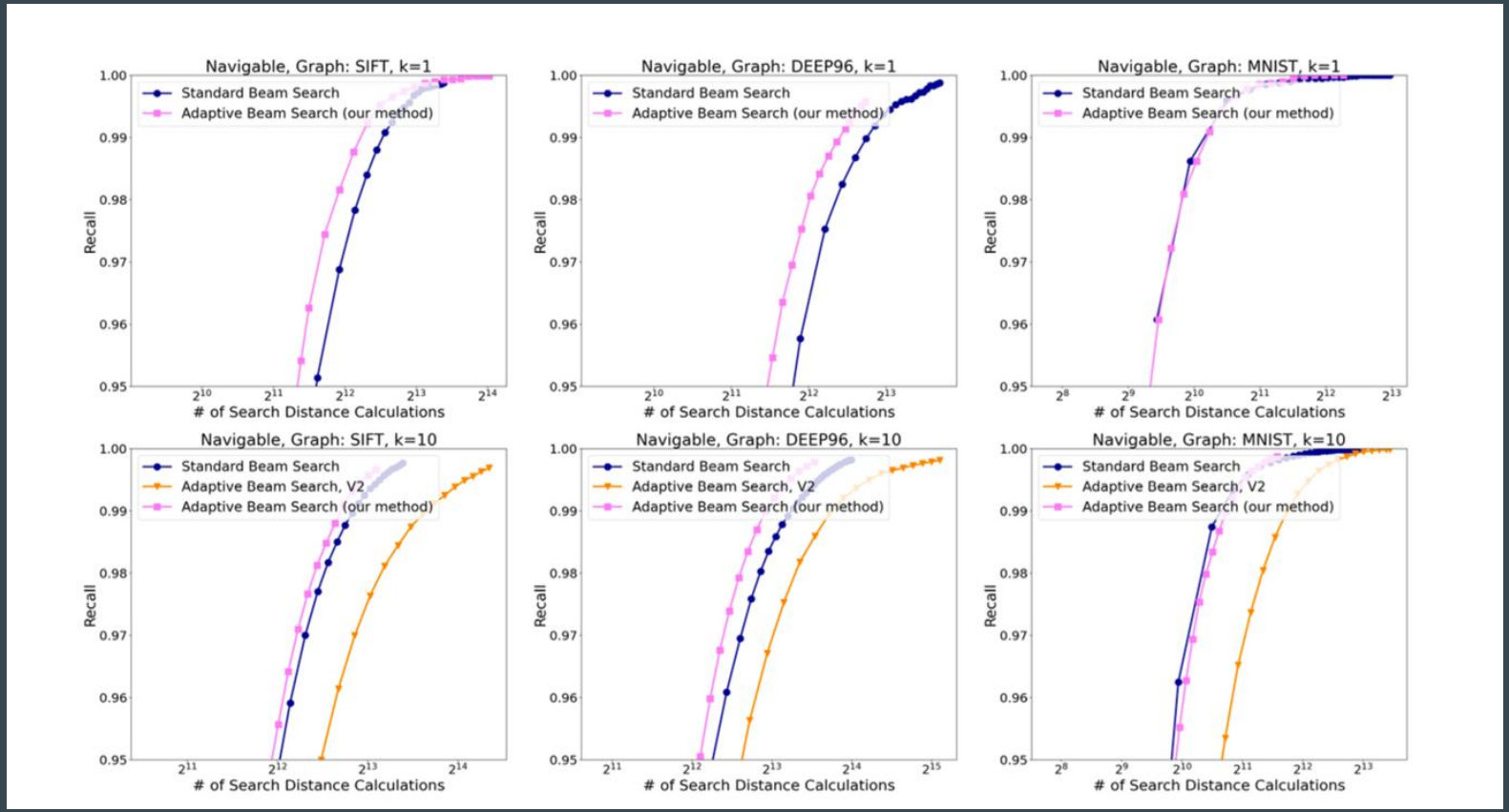
Adaptive Beam Search

- Use smarter termination condition
  - Let q is the query, x is the current exploring point, and a discovered points set D, constant factor γ:
    - Beam search, with beam-width $b \geq k$, terminates if there are at least:
      - b items $j_1, ..., j_b \in D$ with $d(q, j_i) \leq d(q, x)$
    - Adaptive beam search with parameter γ, terminates if there are at least:
      - k items $j_1, ..., j_k \in D$ with $(1+\gamma) \cdot d(q, j_i) \leq d(q, x)$
- Main theorem: Adaptive beam search with approximate factor γ returns approximate nearest neighbor x^* (when $k = 1$) with provable $2/\gamma$-approximation
  - i.e. $d(q, x^*) \leq 2/\gamma \cdot \min d(q, v)$, for v in $G/\{x^*\}$
  - For larger k, the $2/\gamma$-approximation still holds

Adaptive Beam Search

- Result shows that navigability leads to provable ANN
- Adaptive Beam Search also works well in practice, as shown by experiments
- We constructed approximately navigable graphs for common data sets
- We also used common graph constructions to construct non-navigable graphs (HNSW, Vamana/DiskANN, NSG)

Experimental Results - Navigable Graphs



Experimental Results - Heuristic Graphs