G-CARE:

A Framework for Performance Benchmarking of Cardinality Estimation Techniques for Subgraph Matching

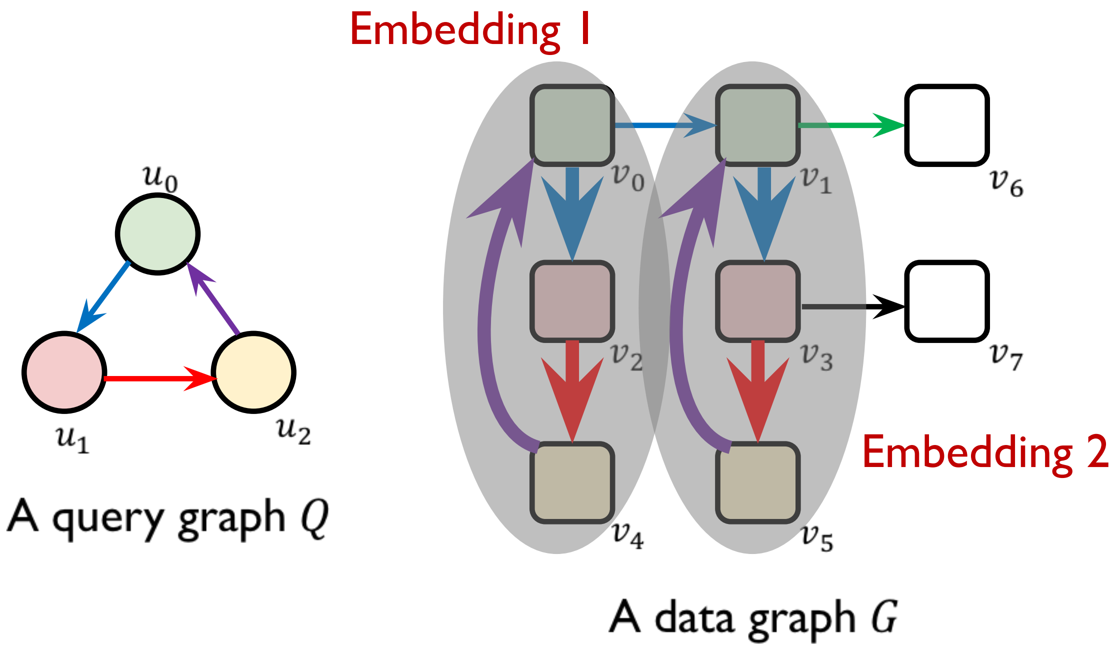
June, 2020

POSTECH: **Yeonsu Park**, Seongyun Ko, Kyoungmin Kim, Kijae Hong, Wook-Shin Han

NTU: Sourav S Bhowmick

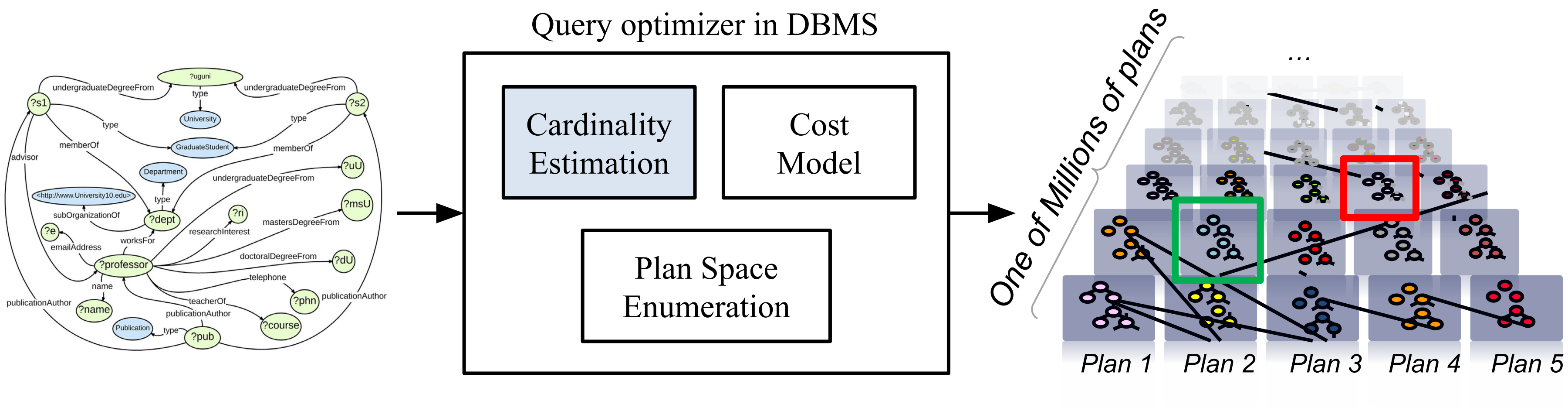
Subgraph Matching and Cardinality

- Subgraph Matching is one of the most important graph queries
- Cardinality is defined as the number of *embeddings*



Cardinality Estimation is Essential for Query Optimization

- Exact cardinality estimation is important to determine the accurate execution cost of query plans [1]
- The cardinality of (intermediate) results and input graphs are used as inputs to the query optimizer cost models (i.e., Neo4j, Oracle PGX, RDF-3X)



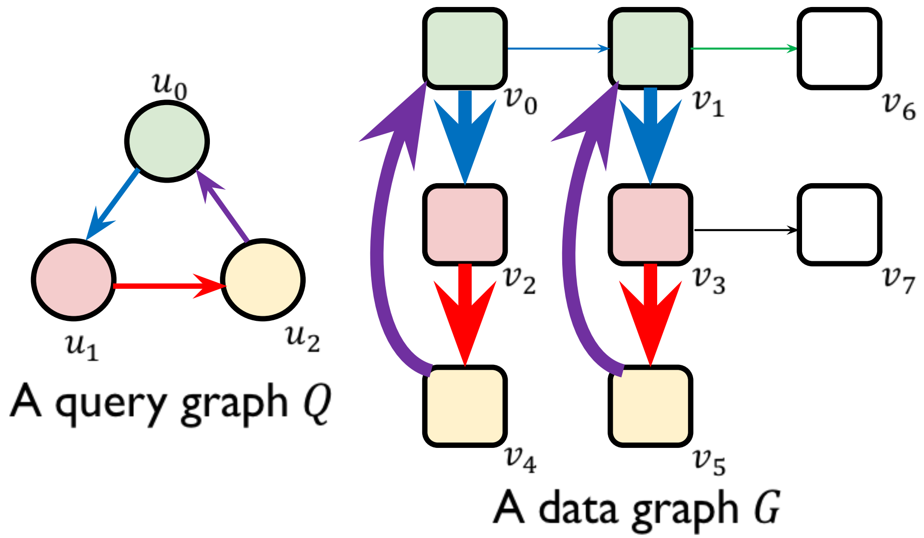
[1] Moerkotte, Guido, Thomas Neumann, and Gabriele Steidl. "Preventing bad plans by bounding the impact of cardinality estimation errors." (VLDB' 09)
* Image from David J. Dewitt's slides

Motivation

- Existing techniques are not compared under a common framework
- Existing literature has overlooked important query features
  - Each of surveyed 54 papers has used only a subset of important query features: **query size, query topology, and query result size**
- Incomprehensive comparisons
  - The cardinality estimation techniques for graph and relational data are developed separately and, thus, not compared to each other

Card. Estimation for Relational Queries Solves the Same Problem

Graph Model



Relational Model

```
SELECT R_blue.src AS u_0,
       R_red.src AS u_1,
       R_purple.src AS u_2,
FROM   R_blue, R_red, R_purple,
       R_green, R_pink, R_yellow
WHERE
R_blue.dst = R_red.src AND
R_red.dst = R_purple.src AND
R_purple.dst = R_blue.src AND
R_blue.src = R_green.v AND
R_red.src = R_pink.v AND
R_purple.src = R_yellow.v AND
```

A relational query Q_{rel}

edge label

R_{blue}

src	dst
0	1
0	2
1	3

R_{red}

src	dst
2	4
3	5

R_{purple}

src	dst
4	0
5	1

vertex label

R_{green}

v
0
1

R_{pink}

v
2
3

R_{yellow}

v
4
5

A database D_G

[2] Abadi, Daniel J., et al. "Scalable semantic web data management using vertical partitioning." (VLDB' 07)

Contributions

- A common framework for implementing Card. Est. Techniques
  - Graph and relational techniques for subgraph matching
  - Summary and sampling-based
- Thorough performance evaluation
  - Real-world and synthetic datasets
  - Various query features (query sizes, topologies, result sizes)
- Intriguing and unexpected findings
  - WanderJoin designed for online aggregation outperforms the others

G-CARE Framework

Card. Estimation Techniques Implemented in G-CARE

- For graph data
  - Characteristic Sets [ICDE'11]
  - IMPR [ICDM'16]
  - SumRDF [WWW'18]

- For relational data
- Card. estimation techniques
  - Correlated Sampling [VLDB'15]
  - Join Sampling with Upper Bound (Modification from [3])
  - BoundSketch [SIGMOD'19]
- Online aggregation technique
  - WanderJoin [SIGMOD'16]

[3] Zhao, Zhuoyue, et al. "Random sampling over joins revisited." (SIGMOD' 18)

Summary-based Techniques

- For graph data
  - Characteristic Sets [ICDE'11]
  - IMPR [ICDM'16]
  - SumRDF [WWW'18]

- For relational data
- Card. estimation techniques
  - Correlated Sampling [VLDB'15]
  - Join Sampling with Upper Bound (Modification from [3])
  - BoundSketch [SIGMOD'19]

- Online aggregation technique
  - WanderJoin [SIGMOD'16]

[3] Zhao, Zhuoyue, et al. "Random sampling over joins revisited." (SIGMOD' 18)

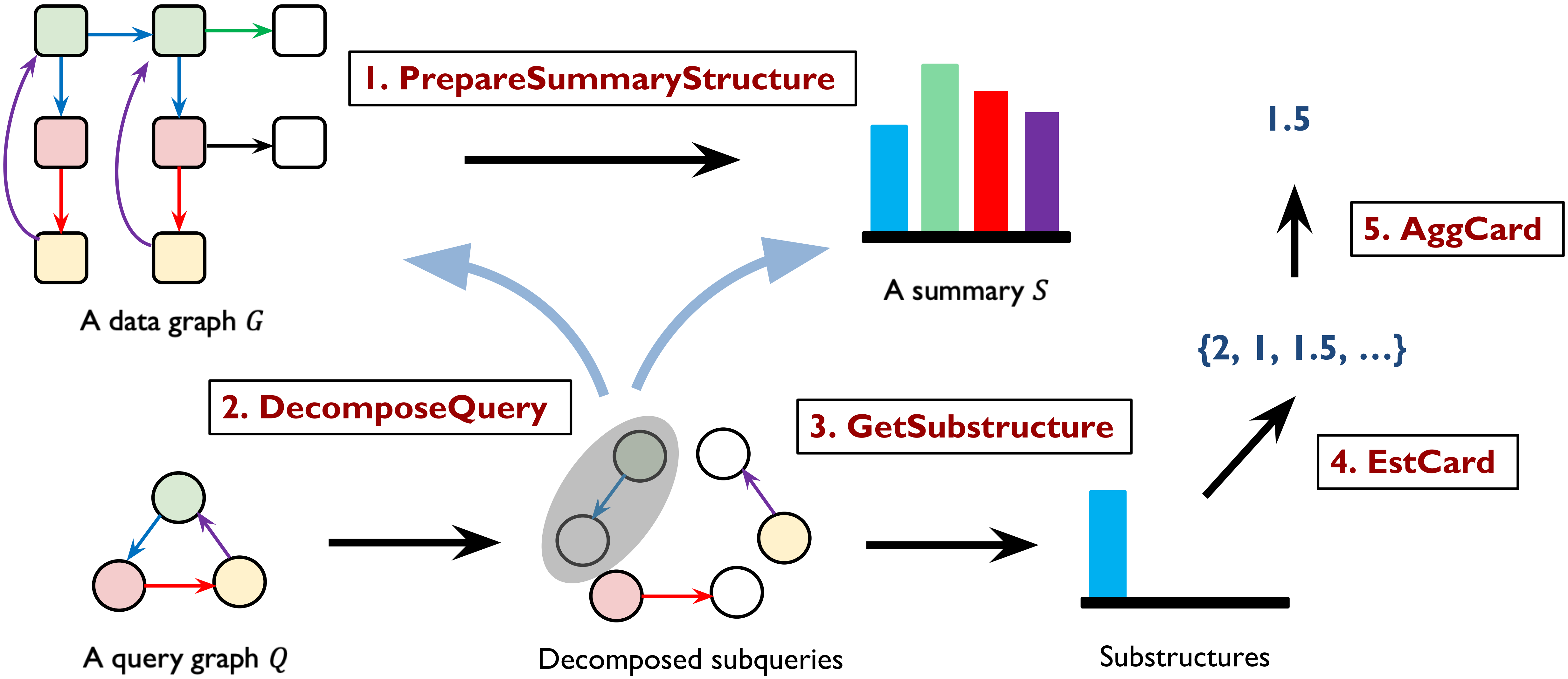
Sampling-based Techniques

- For graph data
  - Characteristic Sets [ICDE'11]
  - IMPR [ICDM'16]
  - SumRDF [WWW'18]

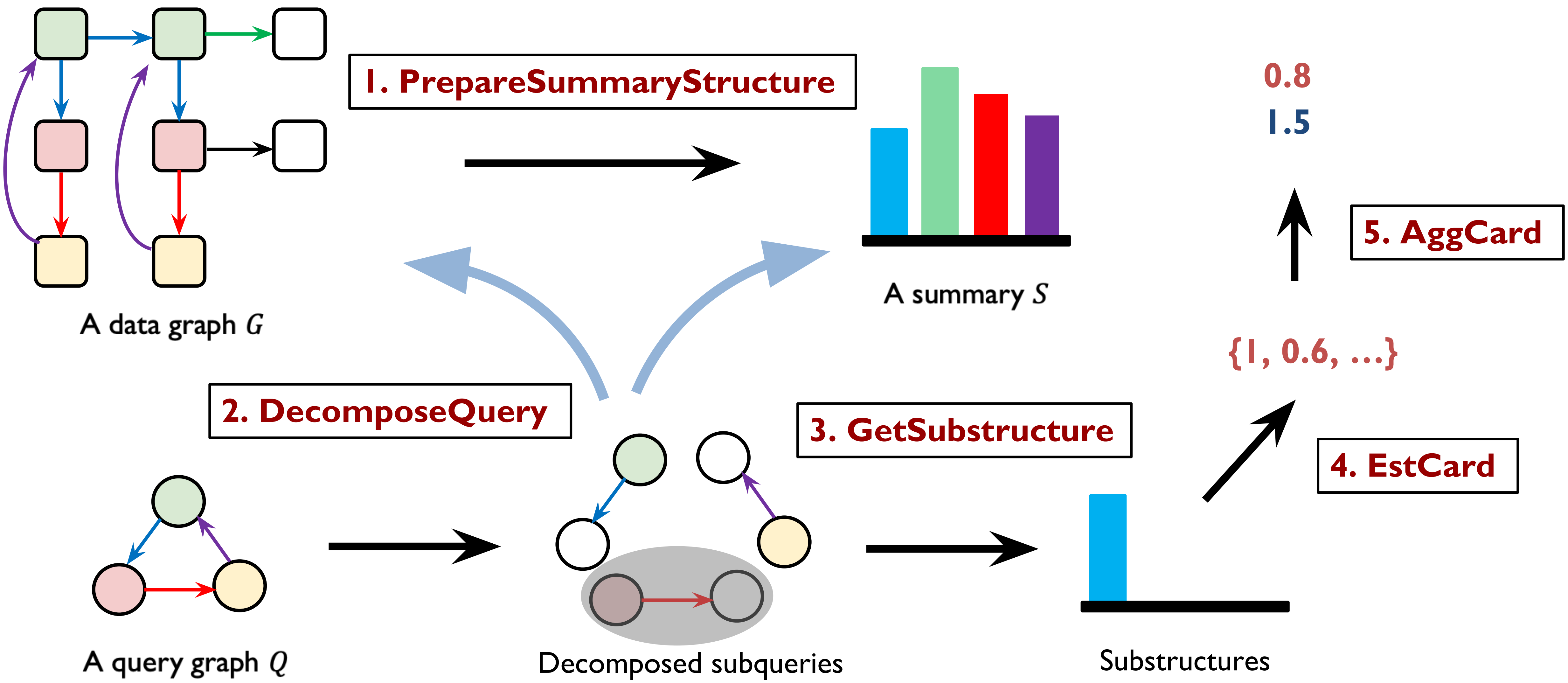
- For relational data
- Card. estimation techniques
  - Correlated Sampling [VLDB'15]
  - Join Sampling with Upper Bound (Modification from [3])
  - BoundSketch [SIGMOD'19]
- Online aggregation technique
  - WanderJoin [SIGMOD'16]

[3] Zhao, Zhuoyue, et al. "Random sampling over joins revisited." (SIGMOD' 18)

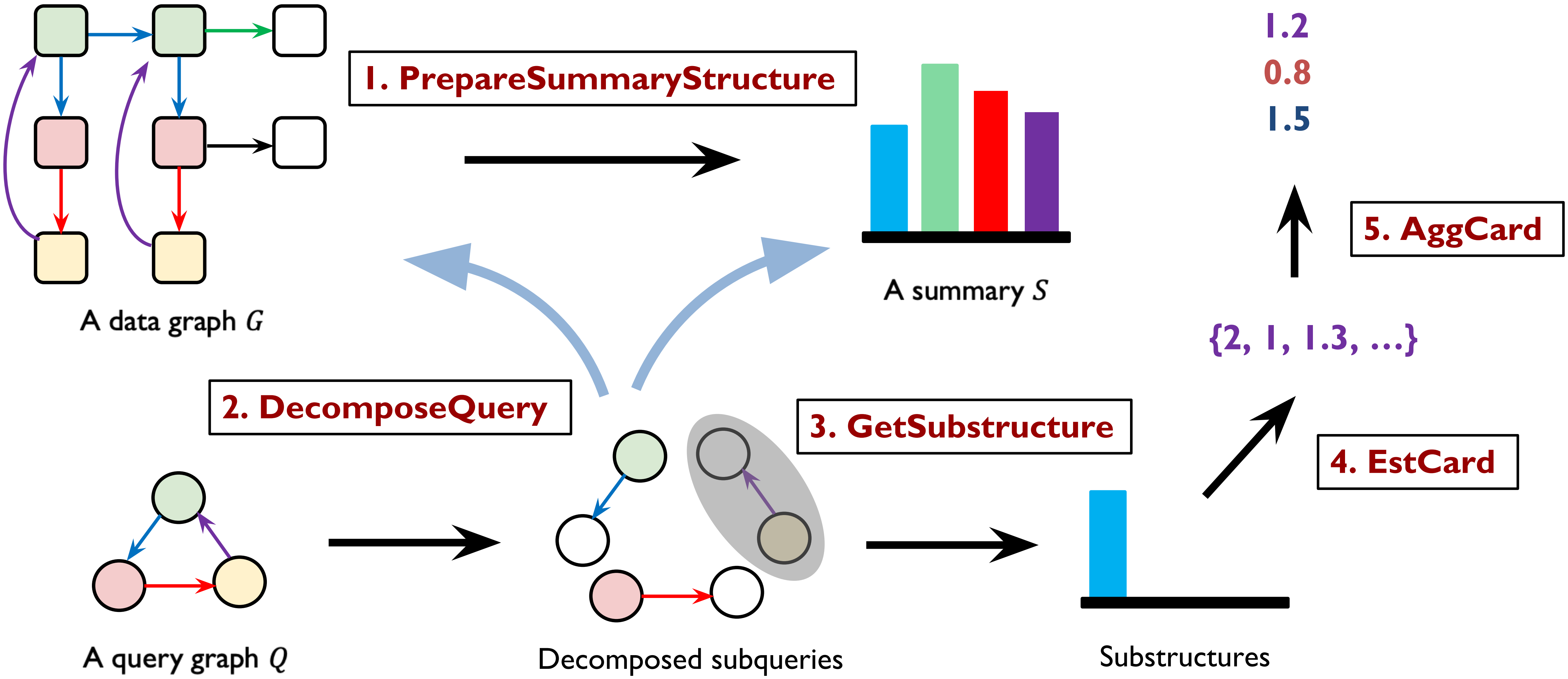
G-CARE Framework



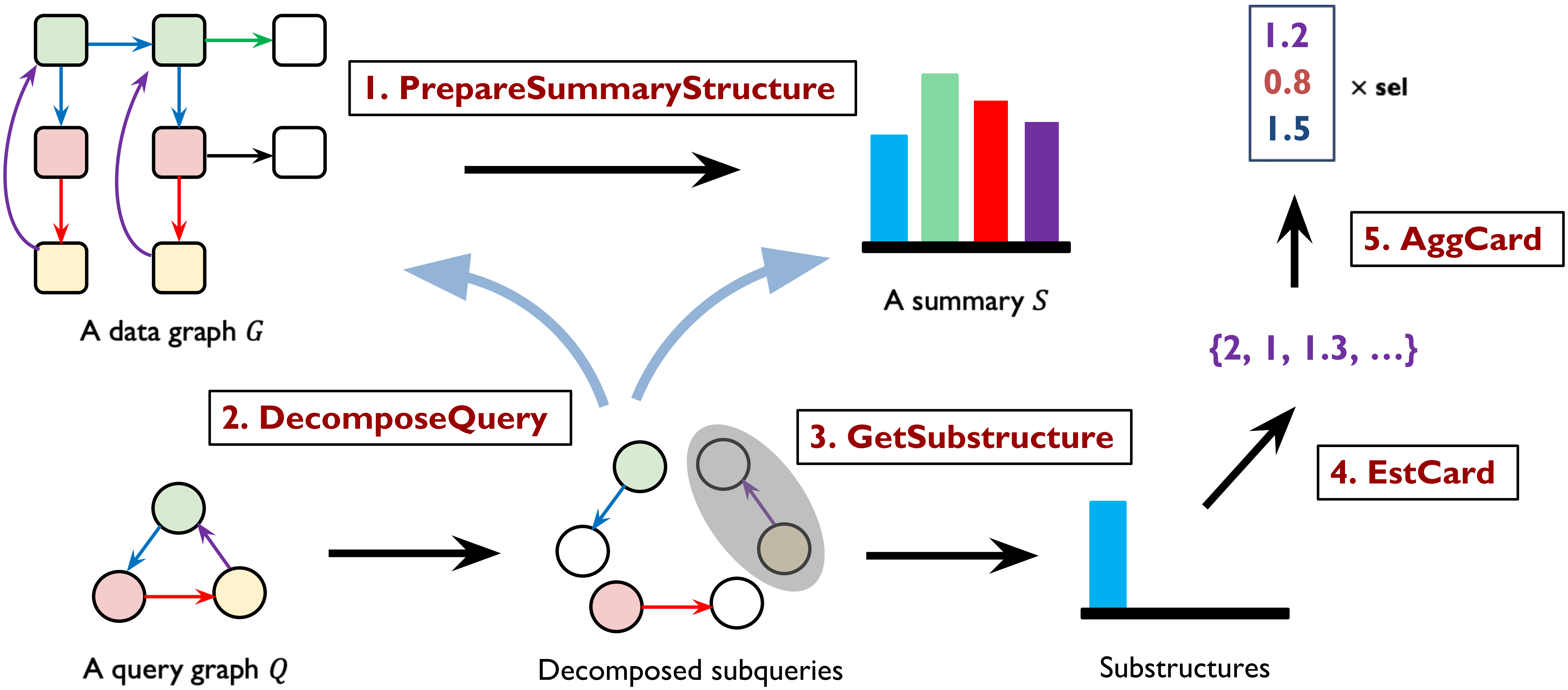
G-CARE Framework



G-CARE Framework



G-CARE Framework



Chracteristic Sets [ICDE'11] and Wander Join [SIGMOD'16]

Characteristic Sets

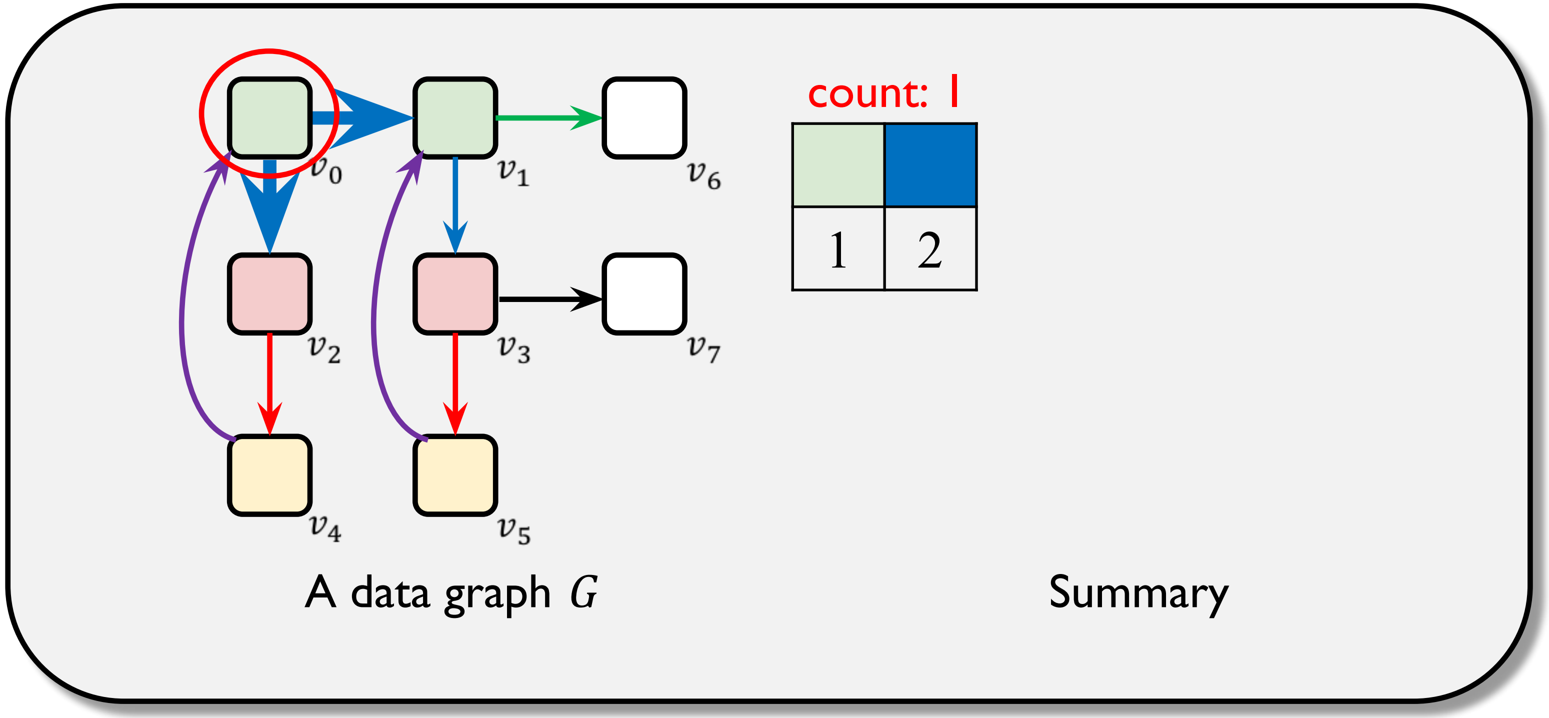
Summary for **Graph Data**

WanderJoin

Sampling for **Relational Data**

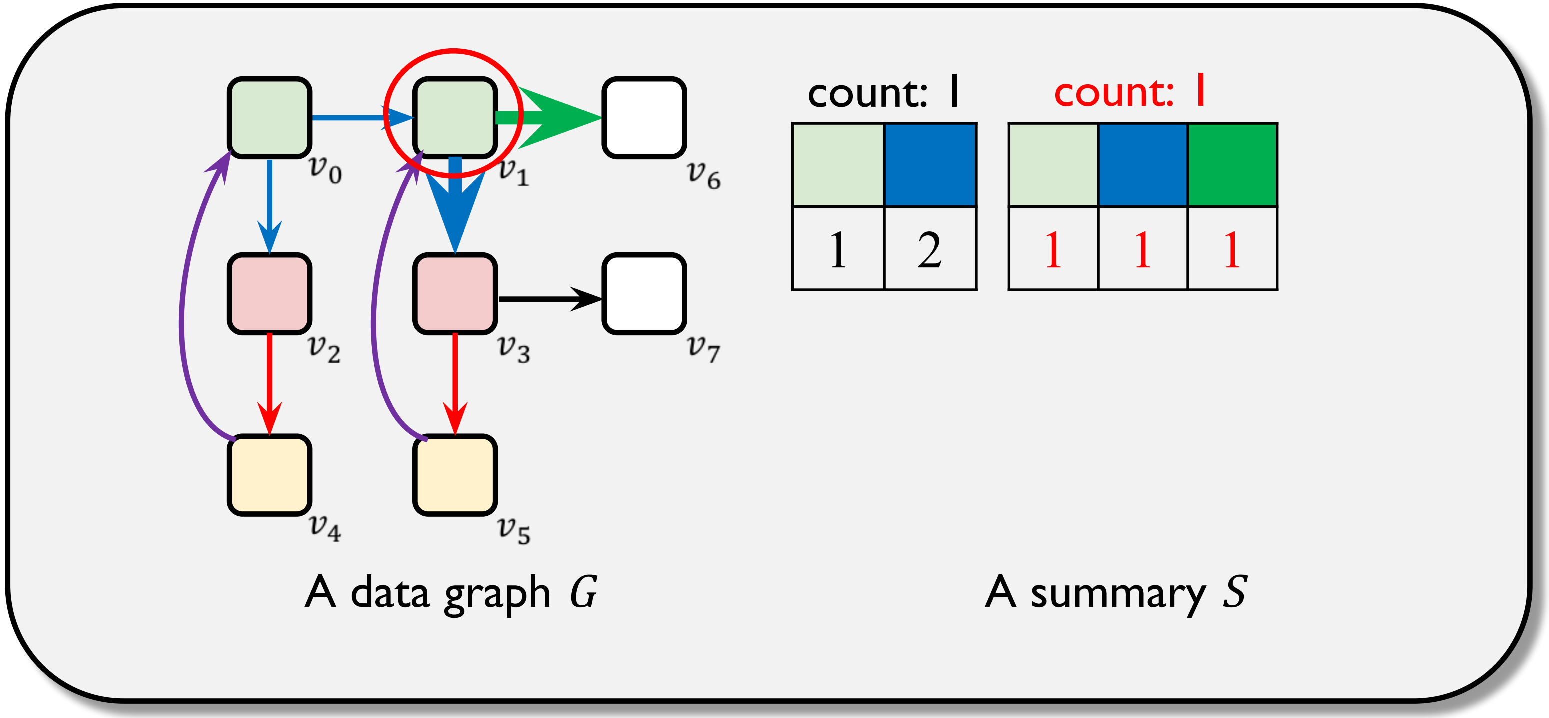
PrepareSummaryStructure

- Create summary structure from the data graph
  - Input: a data graph G
  - Output: a summary S



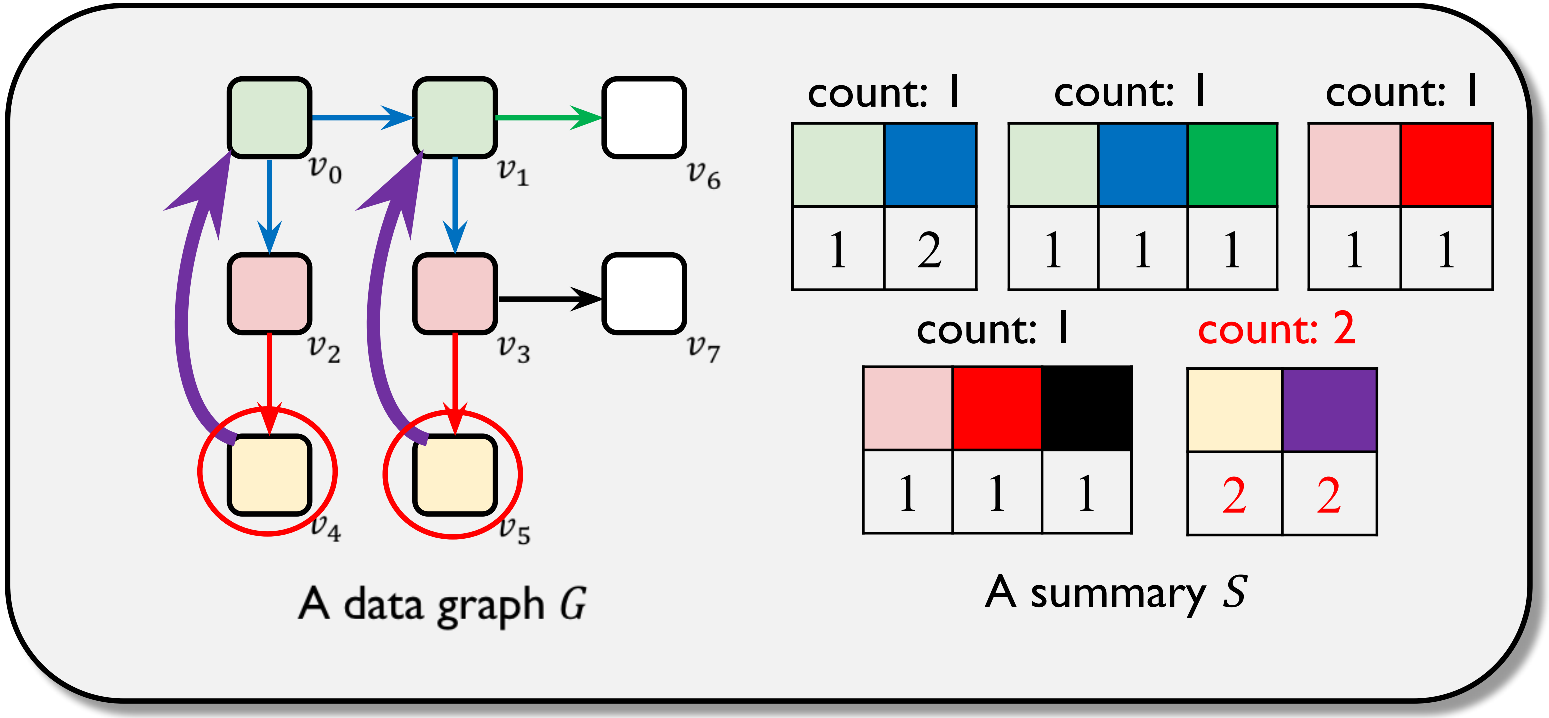
PrepareSummaryStructure

- Create summary structure from the data graph
  - Input: a data graph G
  - Output: a summary S



PrepareSummaryStructure

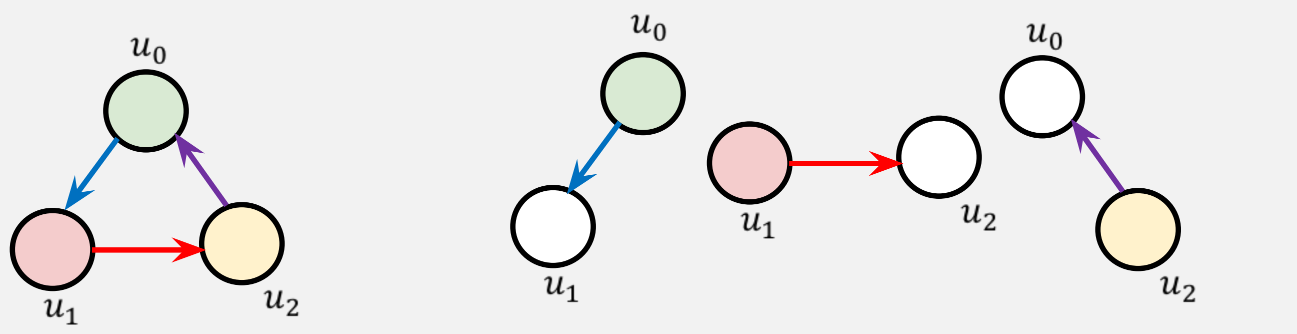
- Create summary structure from the data graph
  - Input: a data graph G
  - Output: a summary S



DecomposeQuery

- Decompose a given query Q into subqueries $(q_0, \ldots, q_{m-1})$

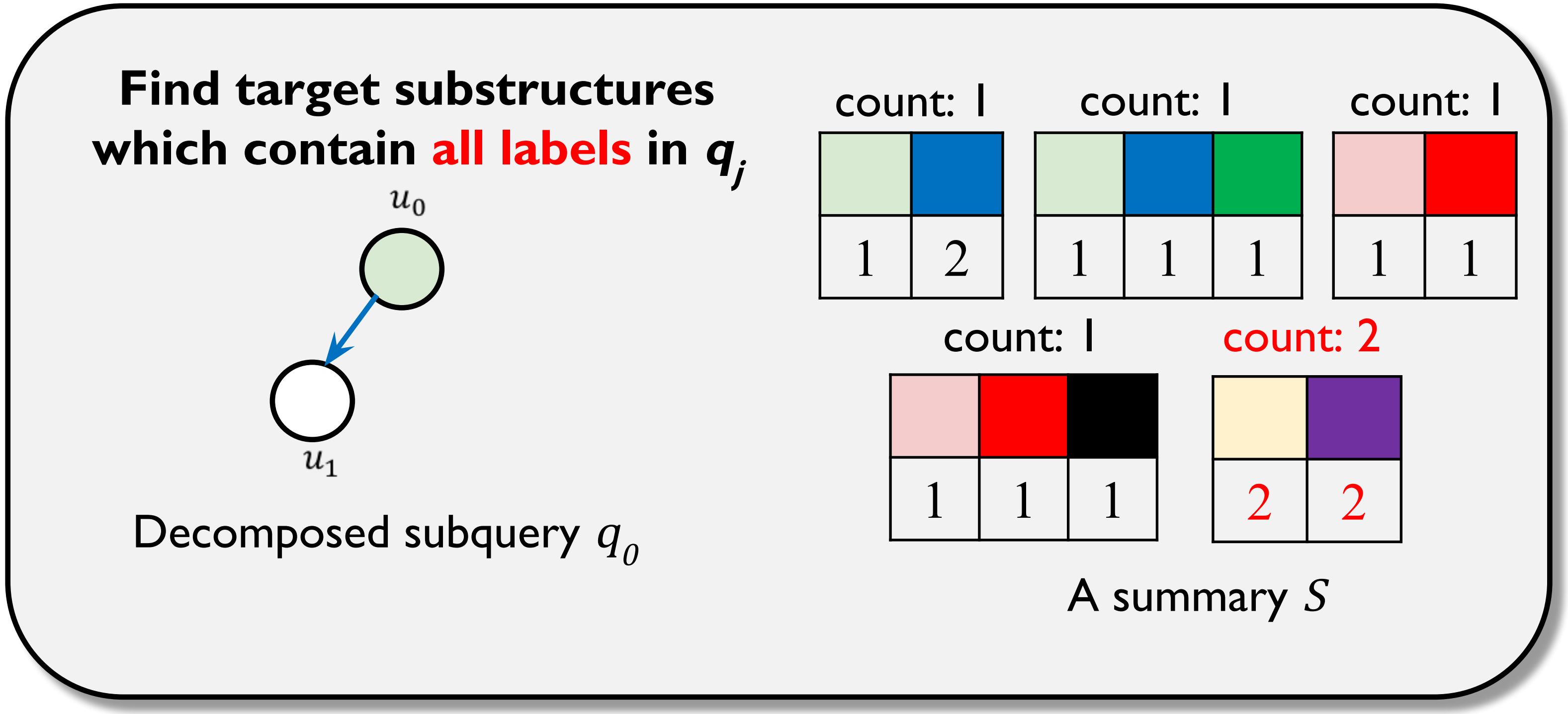
Decompose a query into the star-shaped subqueries



A query graph Q Decomposed subqueries q_0, q_1, and q_2

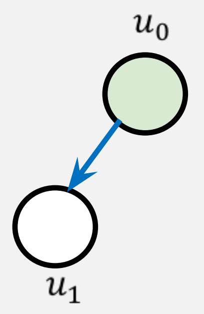
GetSubstructure

- Obtain a series of target substructures for q_j

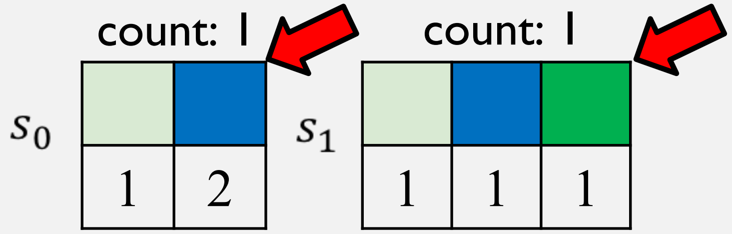


GetSubstructure

- Obtain a series of target substructures for q_j



Decomposed subquery q_0



Find target substructures which contain all vertex and edge labels in q_j

EstCard

- Estimate cardinality of q_j for each target substructure
- Store the estimated cardinality into a vector called *cardVec*

How many star-shaped structures in the data graph correspond to each target substructure

count: 1

1	2

- For s_0, estimate is $1 \cdot \frac{2}{1} = 2$
- For s_1, estimate is $1 \cdot \frac{1}{1} = 1$

Avg. number of edges labeled by blue connected to each center vertex

count: 1

1	1	1

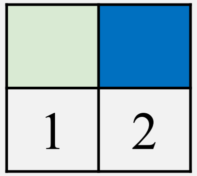
# of center vertices corresponding to each target substructure

AggCard

- Estimate the cardinality of q_j by aggregating over $cardVec$ using aggregation operator

SUM for aggregation

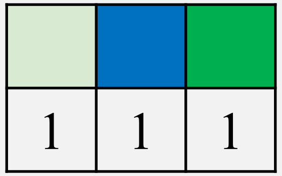
count: 1



- For s_0, estimate is $1 \cdot \frac{2}{1} = 2$
- For s_1, estimate is $1 \cdot \frac{1}{1} = 1$

$\Rightarrow 2 + 1 = 3$

count: 1



Chracteristic Sets [ICDE'11] and Wander Join [SIGMOD'16]

Characteristic Sets

Summary for **Graph Data**

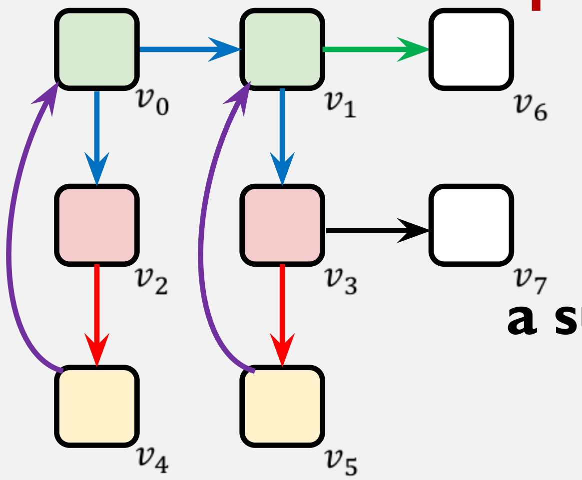
WanderJoin

Sampling for **Relational Data**

GetSubstructure

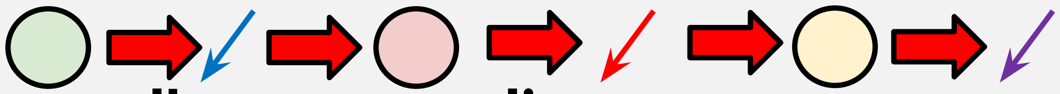
- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$



A data graph G

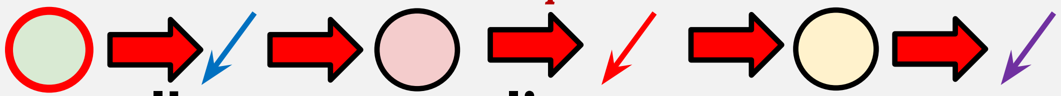
For WanderJoin,
a substructure = a random walk

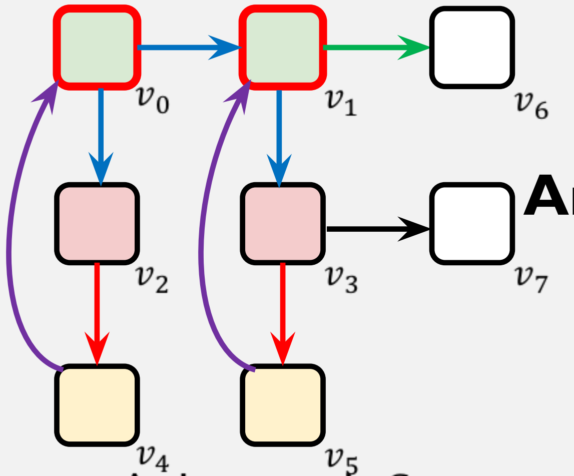
Given the walk order  **,**
perform random walks as a sampling process

GetSubstructure

- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$

Given the walk order  **, perform random walks as a sampling process**



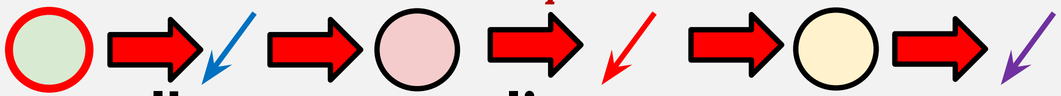
A data graph G

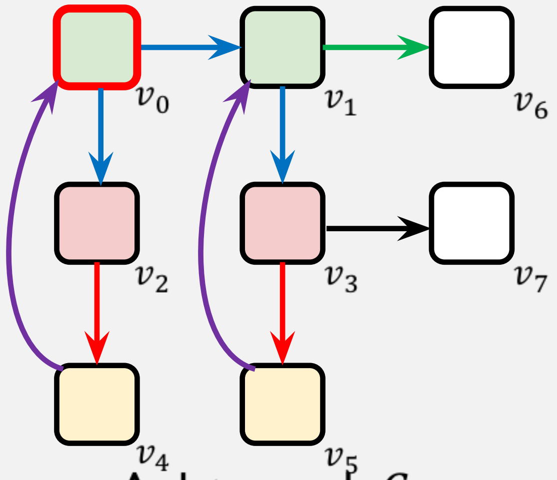
Among the two green vertices, choose one randomly

GetSubstructure

- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$

Given the walk order  **,**
perform random walks as a sampling process



A data graph G

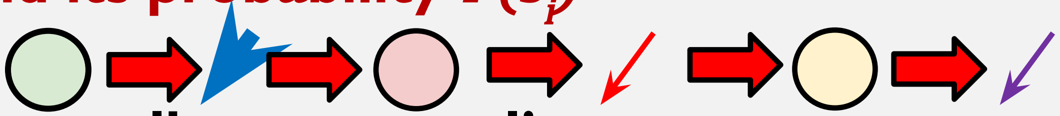
$$s_0 = v_0$$

$$P(s_0) = \frac{1}{2}$$

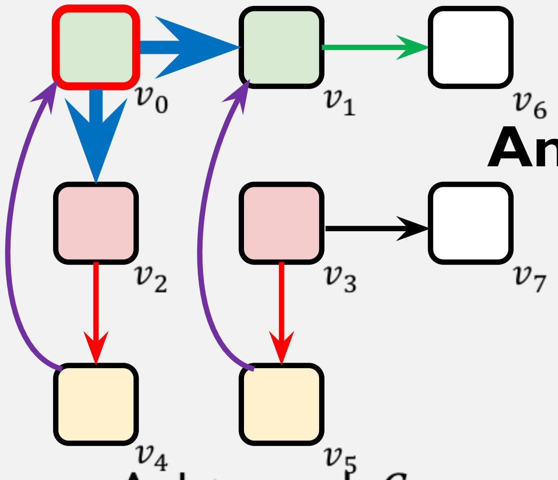
GetSubstructure

- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$

Given the walk order  **,**

perform random walks as a sampling process



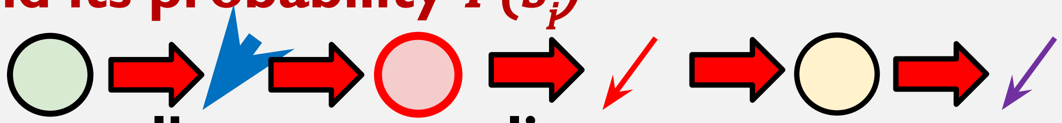
A data graph G

Among the two blue edges of v_0, choose one randomly

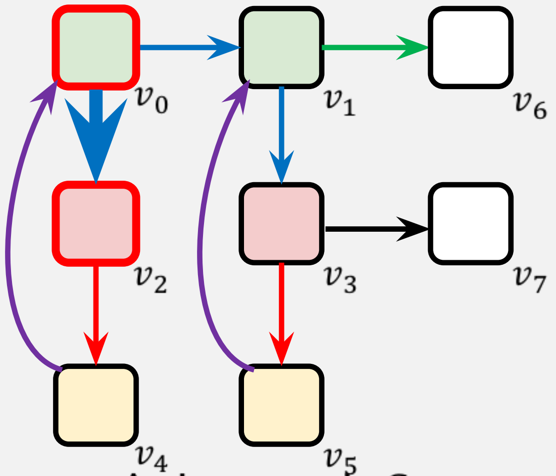
GetSubstructure

- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$

Given the walk order  **,**

perform random walks as a sampling process



A data graph G

$$s_0 = v_0 \rightarrow v_2$$

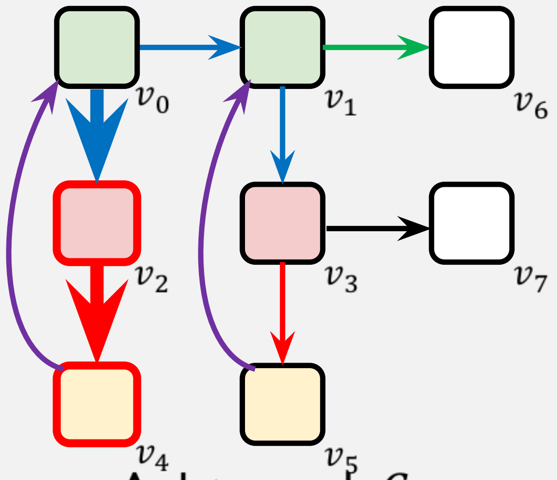
$$P(s_0) = \frac{1}{2} \cdot \frac{1}{2}$$

GetSubstructure

- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$

Given the walk order ,
perform random walks as a sampling process



A data graph G

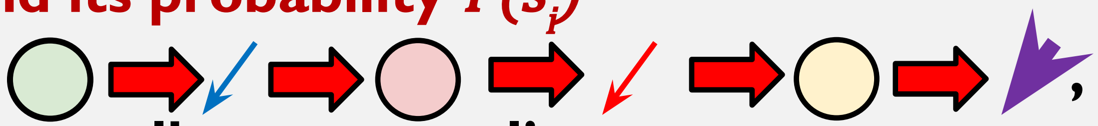
$s_0 = v_0 \rightarrow v_2 \rightarrow v_4$

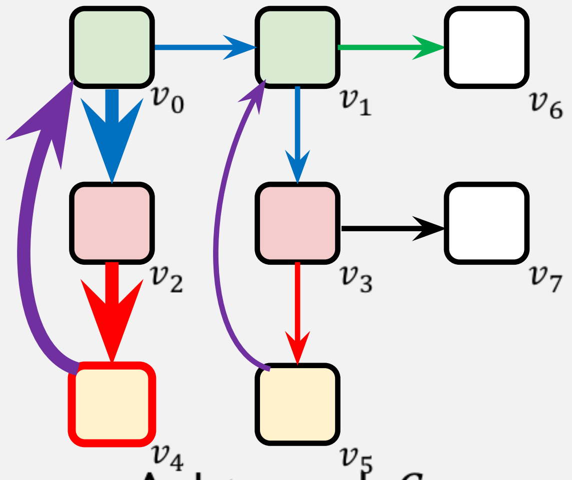
$$P(s_0) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{1}$$

GetSubstructure

- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$

Given the walk order **, perform random walks as a sampling process**



A data graph G

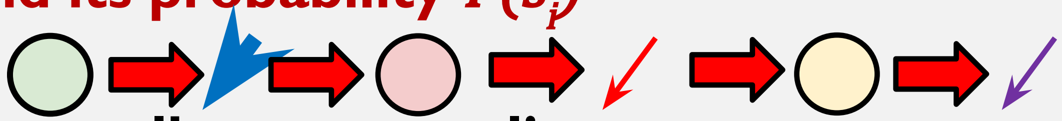
$$s_0 = v_0 \to v_2 \to v_4 \to v_0$$

$$P(s_0) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{1} \cdot \frac{1}{1}$$

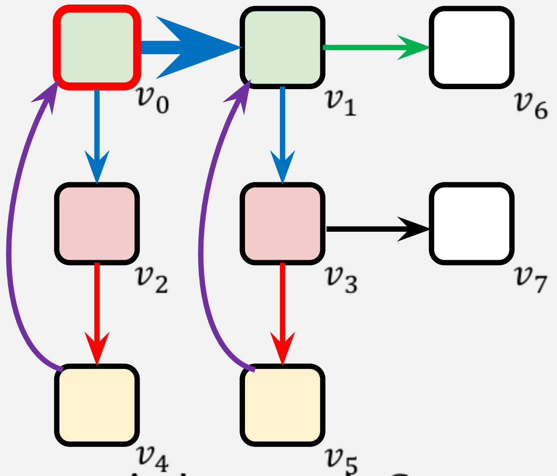
GetSubstructure

- Obtain a series of target substructures for q_j

Return a random walk s_i as a list of edges and its probability $P(s_i)$

Given the walk order  **,**

perform random walks as a sampling process



A data graph G

$s_1 = v_0 \rightarrow v_1$

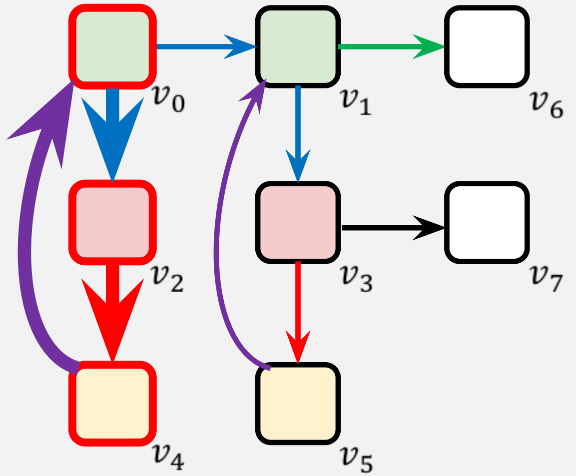
$P(s_1) = \frac{1}{2} \cdot \frac{1}{2}$

Sampling Failure!

EstCard

- Estimate cardinality of q_j for each target substructure
- Store the estimated cardinality into a vector called *cardVec*

Estimates the card. using HT estimator [4]



$s_0 = v_0 \rightarrow v_2 \rightarrow v_4 \rightarrow v_0$

For s_0, estimate is $\frac{1}{P(s_0)} = 4$

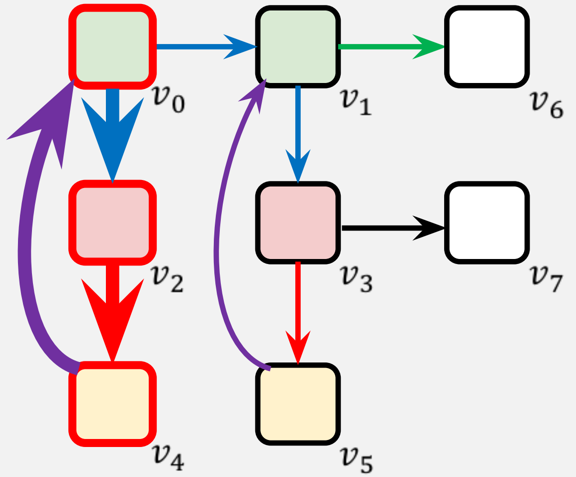
For s_1, estimate is 0
(sampling failure)

[4] Horvitz, Daniel G., and Donovan J. Thompson. "A generalization of sampling without replacement from a finite universe." (JASA' 52)

AggCard

- Estimate the cardinality of q_j by aggregating over *cardVec* using aggregation operator

AVG for aggregation



$s_0 = v_0 \rightarrow v_2 \rightarrow v_4 \rightarrow v_0$

For s_0, estimate is $\frac{1}{P(s_0)} = 4$

For s_1, estimate is 0 (sampling failure)

$$\frac{4+0}{2} = 2$$

Experimental Results
& Important Findings

Experimental Setup

- Datasets & Querysets

- Metrics
  - Accuracy test: q-error [6]
  - Efficiency test: elapsed time

Table 1: Parameters used in the experiments.

Dataset	RDF: LUBM, YAGO, DBpedia Non-RDF: AIDS, Human
Query Topology [5]	Chain, Star, Tree, Cycle, Clique, Petal, Flower, Graph
Query Result Size	$(0, 10], (10, 10^2], (10^2, 10^3], (10^3, 10^4], (10^4, 10^5], (10^5, 10^6]$
Query Size	3, 6, 9, 12
Sampling Ratio	3, 1, 0.3, 0.1, 0.03, 0.01 [%]

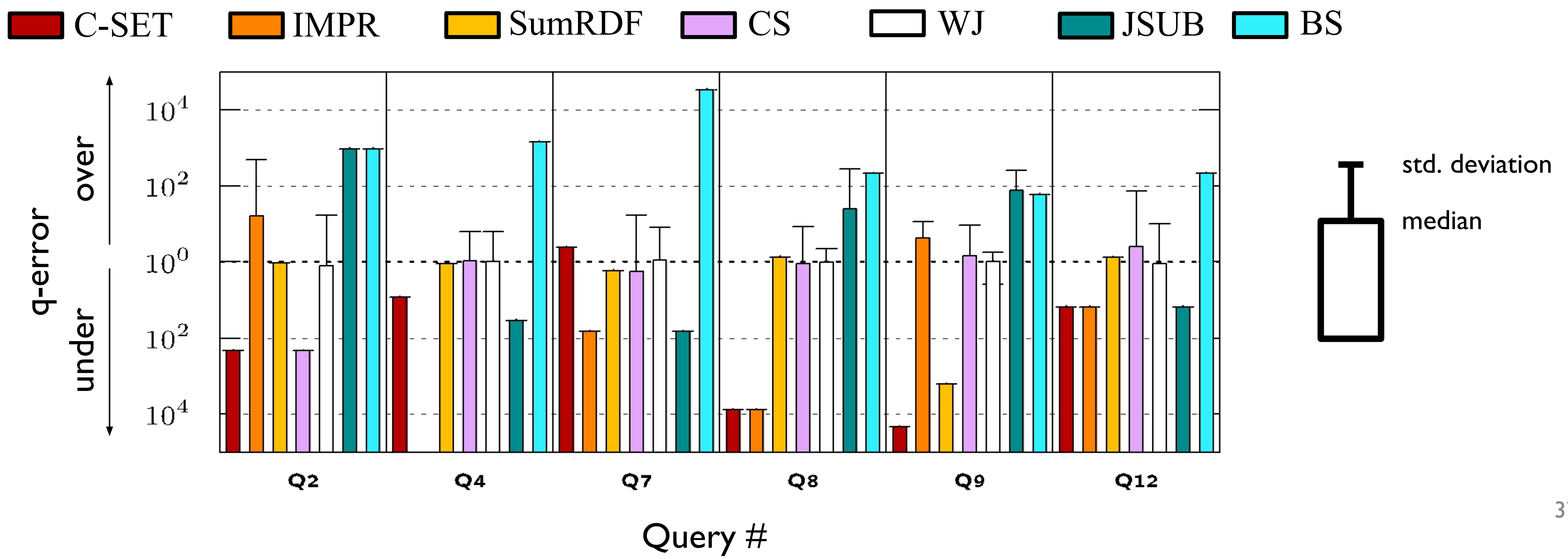
Table 2: Statistics of datasets.

	LUBM	YAGO	DBpedia	AIDS	Human
# of graphs	1	1	1	10K	1
# of vertices	2.6M	12.8M	66.9M	254K	4.7K
# of edges	12.3M	15.8M	225M	548K	86K
Avg. degree	9.35	2.47	6.75	4.31	36.92
Max. degree	0.9M	0.25M	7.3M	22	771
# of distinct v. labels	35	188K	244	50	89
# of distinct e. labels	35	91	39.6K	4	0
Max triples per pred.	2.3M	8.3K	98.7M	270K	-
Min triples per pred.	1	2	1	2.6K	-

[5] Bonifati, Angela, Wim Martens, and Thomas Timm. "An analytical study of large SPARQL query logs." (VLDB' 17)
[6] Moerkotte, Guido, Thomas Neumann, and Gabriele Steidl. "Preventing bad plans by bounding the impact of cardinality estimation errors." *(VLDB'* 09)

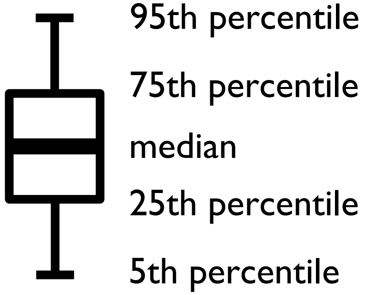
Accuracy Evaluation for LUBM

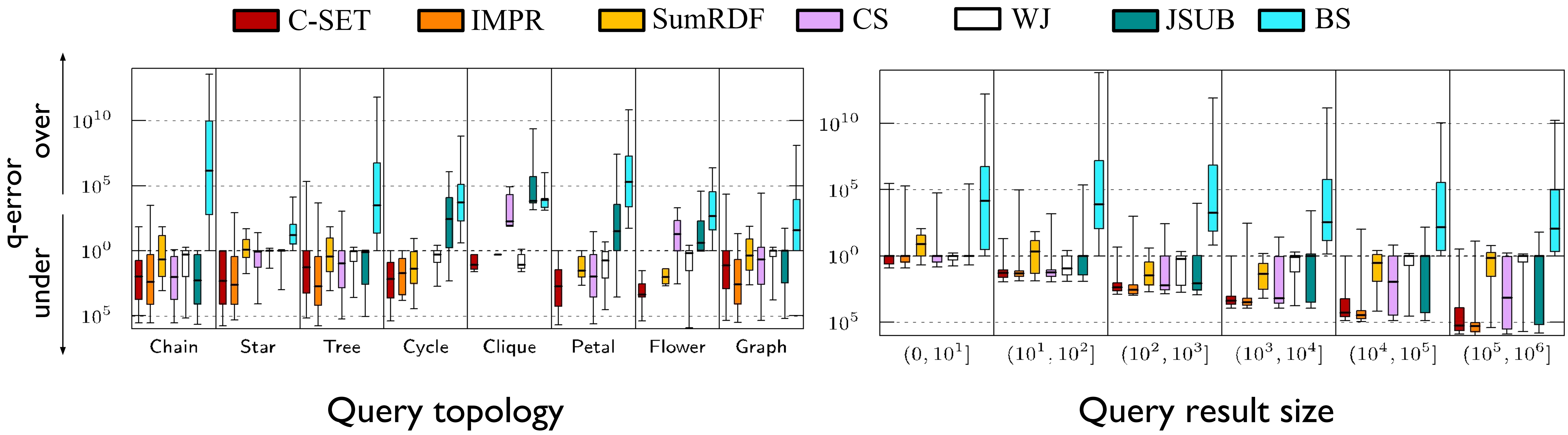
- Surprisingly, WanderJoin (WJ), an online aggregation technique, shows the best accuracy results than the other techniques
- SumRDF performs comparable to WJ, but under-estimates Q9



Accuracy Evaluation for YAGO

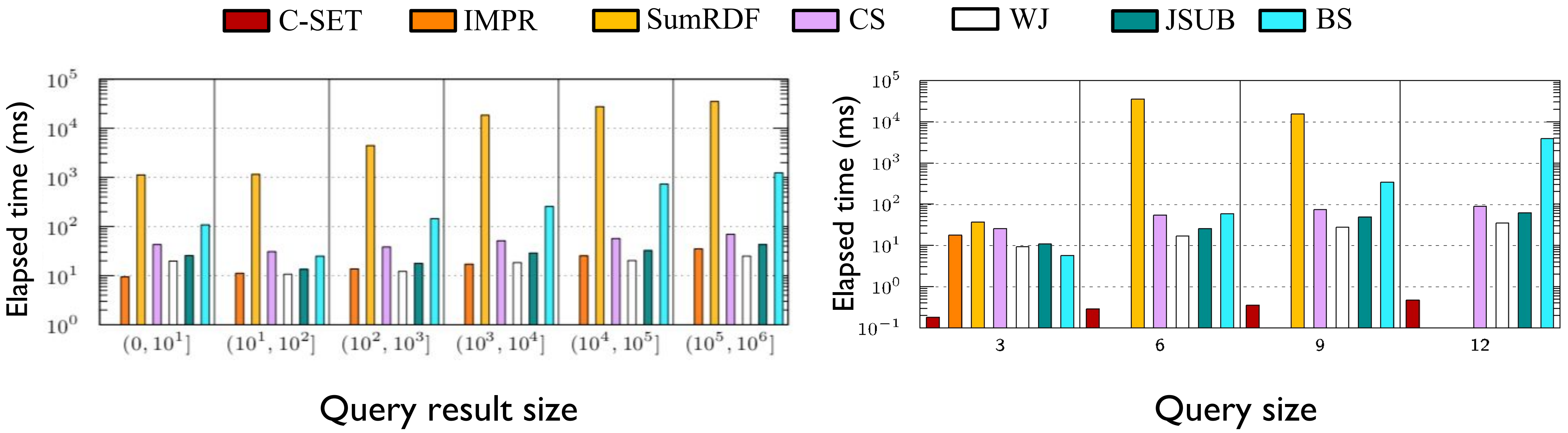
Again, WJ outperforms the other techniques!





Efficiency Evaluation for AIDS

- Similar result for non-RDF datasets
- C-SET is the fastest and WJ is the second fastest



Conclusion

The 1st experimental study which evaluates and analyzes the state-of-the-art card. estimation techniques for subgraph matching

- Unexpected results
  - Existing techniques have serious problems in terms of accuracy and efficiency
  - A simple sampling method, which is based on an online aggregation technique designed for relational data, consistently outperforms the existing techniques

- Avenues of research
  - Integrate the benefits of WanderJoin with native graph-based techniques
  - Hybrid system that leverages native graph stores for query processing but utilizes a relational framework for cardinality estimation

Thank you

References

Chen, X., & Lui, J. C. (2016, December). Mining graphlet counts in online social networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on* (pp. 71-80). IEEE.

Neumann, T., & Moerkotte, G. (2011, April). Characteristic sets: Accurate cardinality estimation for RDF queries with multiple joins. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on* (pp. 984-994). IEEE.

Stefanoni, G., Motik, B., & Kostylev, E. V. (2018, April). Estimating the Cardinality of Conjunctive Queries over RDF Data Using Graph Summarisation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web* (pp. 1043-1052). International World Wide Web Conferences Steering Committee.

References

Vengerov, D., Menck, A. C., Zait, M., & Chakkappen, S. P. (2015). Join size estimation subject to filter conditions. *Proceedings of the VLDB Endowment*, *8*(12), 1530-1541.

Li, F., Wu, B., Yi, K., & Zhao, Z. (2016, June). Wander join: Online aggregation via random walks. In *Proceedings of the 2016 International Conference on Management of Data* (pp. 615-629). ACM.

Zhao, Z., Christensen, R., Li, F., Hu, X., & Yi, K. (2018, May). Random Sampling over Joins Revisited. In *Proceedings of the 2018 International Conference on Management of Data* (pp. 1525-1539). ACM.

Cai, W., Balazinska, M., & Suciu, D. (2019, June). Pessimistic Cardinality Estimation: Tighter Upper Bounds for Intermediate Join Cardinalities. In *Proceedings of the 2019 International Conference on Management of Data* (pp. 18-35). ACM.