UNIVERSITÉ DE NANTES

LABORATOIRE D'INFORMATIQUE DE NANTES ATLANTIQUE

Answering SPARQL Queries using Views

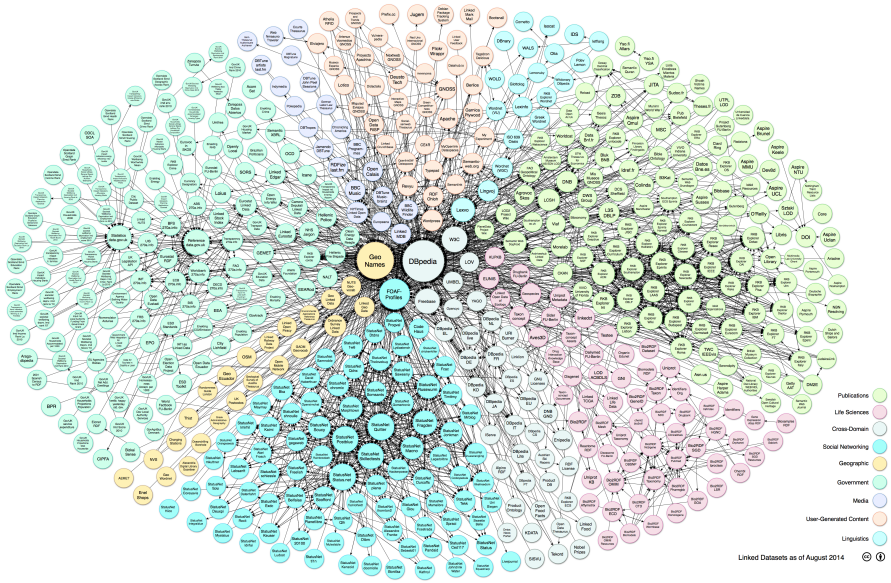
Gabriela Montoya
GDD-Team

Defended on March 11th, 2016

Rapporteurs :	M. Bernd AMANN, Professeur, Université de Pierre et Marie Curie (Paris 6)
	M. Fabien GANDON, Directeur de Recherche, INRIA Sophia Antipolis
Examinateurs :	Mme Pascale KUNTZ, Professeur, Université de Nantes
	Mme Maria-Esther VIDAL, Professeur, Universidad Simón Bolívar
	M. Philippe LAMARRE, Professeur, INSA Lyon
Directeur de thèse :	M. Pascal MOLLI, Professeur, Université de Nantes
Co-encadrante :	Mme Hala SKAF-MOLLI, Maître de Conferences, Université de Nantes

The Linking Open Data cloud

- Huge distributed RDF graph.
- Autonomous data providers publish data in different areas.

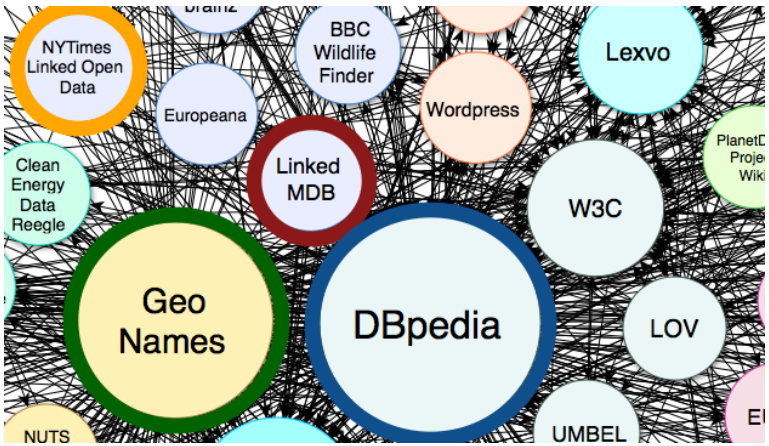


Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak. `http://lod-cloud.net/`

Querying the Linking Open Data cloud

Films, their director, their genre and their director's nationality.

```
SELECT ?f ?d ?g ?n WHERE {
    ?f dbo:director ?d .
    ?d dbo:nationality ?n  .
    ?x owl:sameAs ?f  .
    ?x lmdb:genre ?g
}
```




```
Answers:
 {(f,dbr:Remember_Me,_My_Love),
    (d,dbr:Gabriele_Muccino),
    (g,lmdbfilmgenre:4),
    (n,dbr:Italy)},
 {(f,dbr:L'ultimo_bacio),
  (d,dbr:Gabriele_Muccino),
  (g,lmdbfilmgenre:4),
  (n,dbr:Italy)}
```

Large amounts of data are not in the LOD cloud

- The Deep Web data is around 500 times the size of the Surface Web[1].
- Missing data lead to missing answers.

[1] B. He et al. "Accessing the Deep Web". In: *Commun. ACM* (2007).

Large amounts of data are not in the LOD cloud

- The Deep Web data is around 500 times the size of the Surface Web[1].
- Missing data lead to missing answers.

How to integrate heterogeneous data, and query it using Semantic Web technologies without any statistic about the data distribution?

[1] B. He et al. "Accessing the Deep Web". In: *Commun. ACM* (2007).

Web-Querying Infrastructure does not ensure data availability

- Less than a third of SPARQL endpoints provide availability of 99-100%[2].
- Developers of Semantic Web applications cannot rely on existing technologies to query the LOD cloud.

[2] C. B. Aranda et al. "SPARQL Web-Querying Infrastructure: Ready for Action?" In: *ISWC*. 2013.

Web-Querying Infrastructure does not ensure data availability

- Less than a third of SPARQL endpoints provide availability of 99-100%[2].
- Developers of Semantic Web applications cannot rely on existing technologies to query the LOD cloud.

How to overcome the data availability limitations of SPARQL endpoints?

[2] C. B. Aranda et al. "SPARQL Web-Querying Infrastructure: Ready for Action?" In: *ISWC*. 2013.

Addressed Issues

- **How to integrate heterogeneous data, and query it using Semantic Web technologies without any statistic about the data distribution?**
- How to overcome the data availability limitations of SPARQL endpoints?

Local-As-View (LAV)

- The query is posed using a **global schema**:

  q(O,V,L,P,F):-vendor(O,V),label(V,L),product(O,P),feature(P,F)

- Sources are described with **views over the global schema**:

  s1(P,R,L,F):-producer(P,R),label(R,L),feature(P,F)

  s2(P,O,V,L):-product(O,P),vendor(O,V),label(V,L)

- Query is **rewritten** using the LAV views:

  r1(O,V,L,P,F) :- s2(P,O,V,L), s1(P,R',L',F)

  r2(O,V,L,P,F) :- s2(P,O,V,L'), s1(P,V,L,F)

Local-As-View (LAV)

- The query is posed using a **global schema**:

  q(O,V,L,P,F):-vendor(O,V),label(V,L),product(O,P),feature(P,F)

- Sourc **the** **globa**

  s1(P,R,L

  s2(P,O,V

- Query :

  r1(O,V,L,P,F) :- s2(P,O,V,L), s1(P,R',L',F)

  r2(O,V,L,P,F) :- s2(P,O,V,L'), s1(P,V,L,F)

Each source contribution is specified independently
⇒ Flexibility and Autonomy
⇒ suits a Web scenario[3].

[3] S. Abiteboul et al. *Web Data Management*. Cambridge University Press, 2011.

LAV rewriting approaches

- Existing LAV query rewriters **only** manage **conjunctive queries**.
- The query rewriting problem is **NP-complete** for conjunctive queries[4].
- The number of conjunctive queries that compose the query rewriting may be **exponential**[5].

---

[4] A. Y. Halevy. "Answering queries using views: A survey". In: *VLDB J.* (2001).

[5] S. Abiteboul et al. *Web Data Management*. Cambridge University Press, 2011.

Very high number of rewritings

- $O((M \times V)^N)$, where M, V, and N are the maximal number of view subgoals, the number of views, and the number of query subgoals.

```
q(O,V,L,P,F):-vendor(O,V),label(V,L),product(O,P),feature(P,F)
```

# views	# rewritings
5	60
10	960
20	15,360
40	245,760
80	3.93216e+06
160	6.29146e+07

LAV without rewriting the queries and no restriction to conjunctive queries?

- Execute the **original query** posed by the user.
- Against a **global schema instance** built using the bodies of the views.
- Before query execution $\Rightarrow$ data may be **stale**.
- Complete instance is built during execution $\Rightarrow$ **very expensive** in time and space.
- **Which views to include in a partial global schema instance?**

Intuitions

Views included in the global schema instance:

V={ }

```
r1(O,V,L,P,F) :- v4(P,O,R',V,L), v2(P,R',L',B',F)
r2(O,V,L,P,F) :- v4(P',O,R',V,L), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r3(O,V,L,P,F) :- v4(P,O,R',V,L'), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r4(O,V,L,P,F) :- v4(P',O,R',V,L'), v3(V,L,O',R',V'), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r5(O,V,L,P,F) :- v4(P,O,R',V,L'), v2(P,V,L,B',F)
r6(O,V,L,P,F) :- v4(P',O,R',V,L'), v2(P,V,L,B',F), v3(P,L',O,R',V')
r7(O,V,L,P,F) :- v3(P',L',O,R',V), v4(P,O,R',V,L), v2(P,R',L',B',F)
r8(O,V,L,P,F) :- v3(P,L',O,R',V), v4(P',O',R',V,L), v2(P,R',L',B',F)
r9(O,V,L,P,F) :- v3(V,L,O,R',V), v4(P,O,R',V',L'), v2(P,R',L',B',F)
r10(O,V,L,P,F) :- v3(P,L',O,R',V), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r11(O,V,L,P,F) :- v3(P',L',O,R',V), v2(P,V,L,B',F), v4(P,O,R',V',L')
r12(O,V,L,P,F) :- v3(P,L',O,R',V), v2(P,V,L,B',F)
:
r60(O,V,L,P,F) :- v5(O,V,L,C'), v3(P,L',O,R',V'), v1(P,L',T',F)
```

Evaluating the original query against V produces the same answers as evaluating the rewritings:

R={ }.

Intuitions

Views included in the global schema instance:

V={ }

r1(O,V,L,P,F) :- v4(P,O,R',V,L), v2(P,R',L',B',F)
r2(O,V,L,P,F) :- v4(P',O,R',V,L), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r3(O,V,L,P,F) :- v4(P,O,R',V,L'), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r4(O,V,L,P,F) :- v4(P',O,R',V,L'), v3(V,L,O',R',V'), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r5(O,V,L,P,F) :- v4(P,O,R',V,L'), v2(P,V,L,B',F)
r6(O,V,L,P,F) :- v4(P',O,R',V,L'), v2(P,V,L,B',F), v3(P,L',O,R',V')
r7(O,V,L,P,F) :- v3(P',L',O,R',V), v4(P,O,R',V,L), v2(P,R',L',B',F)
r8(O,V,L,P,F) :- v3(P,L',O,R',V), v4(P',O',R',V,L), v2(P,R',L',B',F)
r9(O,V,L,P,F) :- v3(V,L,O,R',V), v4(P,O,R',V',L'), v2(P,R',L',B',F)
r10(O,V,L,P,F) :- v3(P,L',O,R',V), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r11(O,V,L,P,F) :- v3(P',L',O,R',V), v2(P,V,L,B',F), v4(P,O,R',V',L')
r12(O,V,L,P,F) :- v3(P,L',O,R',V), v2(P,V,L,B',F)

⋮

r60(O,V,L,P,F) :- v5(O,V,L,C'), v3(P,L',O,R',V'), v1(P,L',T',F)

Evaluating the original query against V produces the same answers as evaluating the rewritings:

R={ }.

Intuitions

Views included in the global schema instance:

V={v4 }

r1(O,V,L,P,F) :- v4(P,O,R',V,L), v2(P,R',L',B',F)
r2(O,V,L,P,F) :- v4(P',O,R',V,L), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r3(O,V,L,P,F) :- v4(P,O,R',V,L'), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r4(O,V,L,P,F) :- v4(P',O,R',V,L'), v3(V,L,O',R',V'), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r5(O,V,L,P,F) :- v4(P,O,R',V,L'), v2(P,V,L,B',F)
r6(O,V,L,P,F) :- v4(P',O,R',V,L'), v2(P,V,L,B',F), v3(P,L',O,R',V')
r7(O,V,L,P,F) :- v3(P',L',O,R',V), v4(P,O,R',V,L), v2(P,R',L',B',F)
r8(O,V,L,P,F) :- v3(P,L',O,R',V), v4(P',O',R',V,L), v2(P,R',L',B',F)
r9(O,V,L,P,F) :- v3(V,L,O,R',V), v4(P,O,R',V',L'), v2(P,R',L',B',F)
r10(O,V,L,P,F) :- v3(P,L',O,R',V), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r11(O,V,L,P,F) :- v3(P',L',O,R',V), v2(P,V,L,B',F), v4(P,O,R',V',L')
r12(O,V,L,P,F) :- v3(P,L',O,R',V), v2(P,V,L,B',F)

⋮

r60(O,V,L,P,F) :- v5(O,V,L,C'), v3(P,L',O,R',V'), v1(P,L',T',F)

Evaluating the original query against V produces the same answers as evaluating the rewritings:

R={ }.

Intuitions

Views included in the global schema instance:

$$V=\{v4,\ v2\quad\}$$

r1(O,V,L,P,F) :- v4(P,O,R',V,L), v2(P,R',L',B',F)
r2(O,V,L,P,F) :- v4(P',O,R',V,L), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r3(O,V,L,P,F) :- v4(P,O,R',V,L'), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r4(O,V,L,P,F) :- v4(P',O,R',V,L'), v3(V,L,O',R',V'), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r5(O,V,L,P,F) :- v4(P,O,R',V,L'), v2(P,V,L,B',F)
r6(O,V,L,P,F) :- v4(P',O,R',V,L'), v2(P,V,L,B',F), v3(P,L',O,R',V')
r7(O,V,L,P,F) :- v3(P',L',O,R',V), v4(P,O,R',V,L), v2(P,R',L',B',F)
r8(O,V,L,P,F) :- v3(P,L',O,R',V), v4(P',O',R',V,L), v2(P,R',L',B',F)
r9(O,V,L,P,F) :- v3(V,L,O,R',V), v4(P,O,R',V',L'), v2(P,R',L',B',F)
r10(O,V,L,P,F) :- v3(P,L',O,R',V), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r11(O,V,L,P,F) :- v3(P',L',O,R',V), v2(P,V,L,B',F), v4(P,O,R',V',L')
r12(O,V,L,P,F) :- v3(P,L',O,R',V), v2(P,V,L,B',F)

⋮

r60(O,V,L,P,F) :- v5(O,V,L,C'), v3(P,L',O,R',V'), v1(P,L',T',F)

Evaluating the original query against V produces the same answers as evaluating the rewritings:

$$R=\{r1,\ r5\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\}.$$

Intuitions

Views included in the global schema instance:

$$V=\{v4, v2, v3\}$$

```
r1(O,V,L,P,F) :- v4(P,O,R',V,L), v2(P,R',L',B',F)
r2(O,V,L,P,F) :- v4(P',O,R',V,L), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r3(O,V,L,P,F) :- v4(P,O,R',V,L'), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r4(O,V,L,P,F) :- v4(P',O,R',V,L'), v3(V,L,O',R',V'), v3(P,L',O,R',V'), v2(P,R',L',B',F)
r5(O,V,L,P,F) :- v4(P,O,R',V,L'), v2(P,V,L,B',F)
r6(O,V,L,P,F) :- v4(P',O,R',V,L'), v2(P,V,L,B',F), v3(P,L',O,R',V')
r7(O,V,L,P,F) :- v3(P',L',O,R',V), v4(P,O,R',V,L), v2(P,R',L',B',F)
r8(O,V,L,P,F) :- v3(P,L',O,R',V), v4(P',O',R',V,L), v2(P,R',L',B',F)
r9(O,V,L,P,F) :- v3(V,L,O,R',V), v4(P,O,R',V',L'), v2(P,R',L',B',F)
r10(O,V,L,P,F) :- v3(P,L',O,R',V), v3(V,L,O',R',V'), v2(P,R',L',B',F)
r11(O,V,L,P,F) :- v3(P',L',O,R',V), v2(P,V,L,B',F), v4(P,O,R',V',L')
r12(O,V,L,P,F) :- v3(P,L',O,R',V), v2(P,V,L,B',F)
:
r60(O,V,L,P,F) :- v5(O,V,L,C'), v3(P,L',O,R',V'), v1(P,L',T',F)
```

Evaluating the original query against V produces the same answers as evaluating the rewritings:

$$R=\{r1, r5, r2, r3, r4, r6, r7, r8, r9, r10, r11, r12\}.$$

Research Question

- In which **order** should the query relevant **views** be loaded into a **global schema instance**, built during query execution, in order to use this instance to **answer the query**, and outperform the traditional LAV query rewriting techniques in terms of number of answers produced per time unit?

The Maximal Coverage problem (MaxCov)

Given a SPARQL query, a set of LAV views, V, and an integer, k, obtain V_k, $V_k \subseteq V \wedge |V_k| = k$, such that there is no other subset of V of size k that covers more rewritings than V_k.

SemLAV is a polynomial time solution for MaxCov

- Assumption: all the variables in the LAV views are *distinguishable*.

1. First step of the Bucket Algorithm[6].
2. Views in each bucket are ranked.
3. The k top ranked views are included in V_k.

[6] A. Levy et al. "Querying Heterogeneous Information Sources Using Source Descriptions". In: *VLDB*. 1996.

SemLAV makes a bucket for each query subgoal

SPARQL Query:

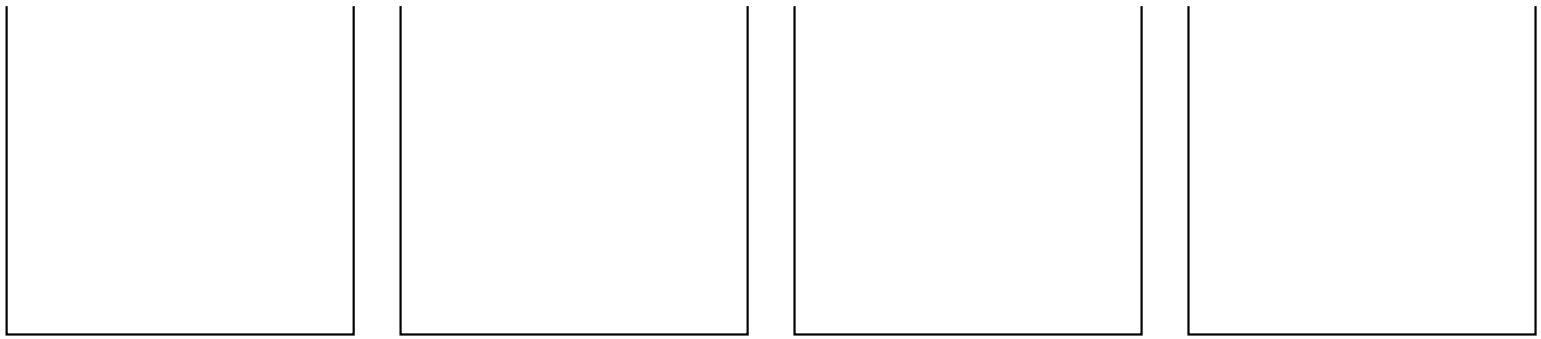
```
SELECT * WHERE {
   ?Offer bsbm:vendor ?Vendor .
   ?Vendor rdfs:label ?Label .
   ?Offer bsbm:product ?Prod .
   ?Prod bsbm:productFeature ?Feat .
}
```

Query subgoals:

{ vendor(O, V), label(V, L),
product(O, P), feature(P, F) }

SemLAV is low cost

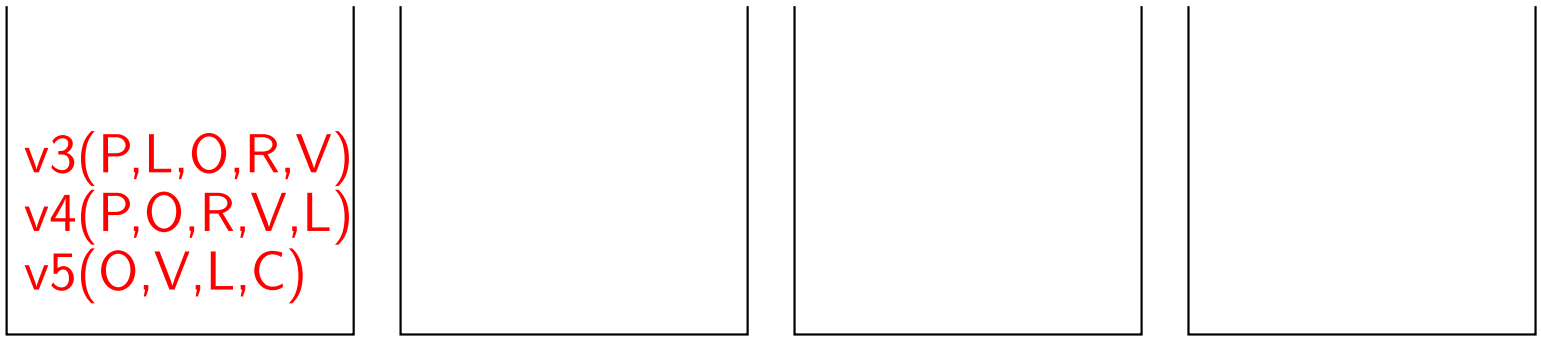
vendor(O, V), label(V,L), product(O,P), feature(P,F)



v1(P,L,T,F) :- label(P,L), type(P,T), feature(P,F)
v2(P,R,L,F) :- producer(P,R), label(R,L), feature(P,F)
v3(P,L,O,V) :- label(P,L), product(O,P), vendor(O,V)
v4(P,O,V,L) :- product(O,P),vendor(O,V),label(V,L)
v5(O,V,L,C) :- vendor(O,V), label(V,L), country(V,C)

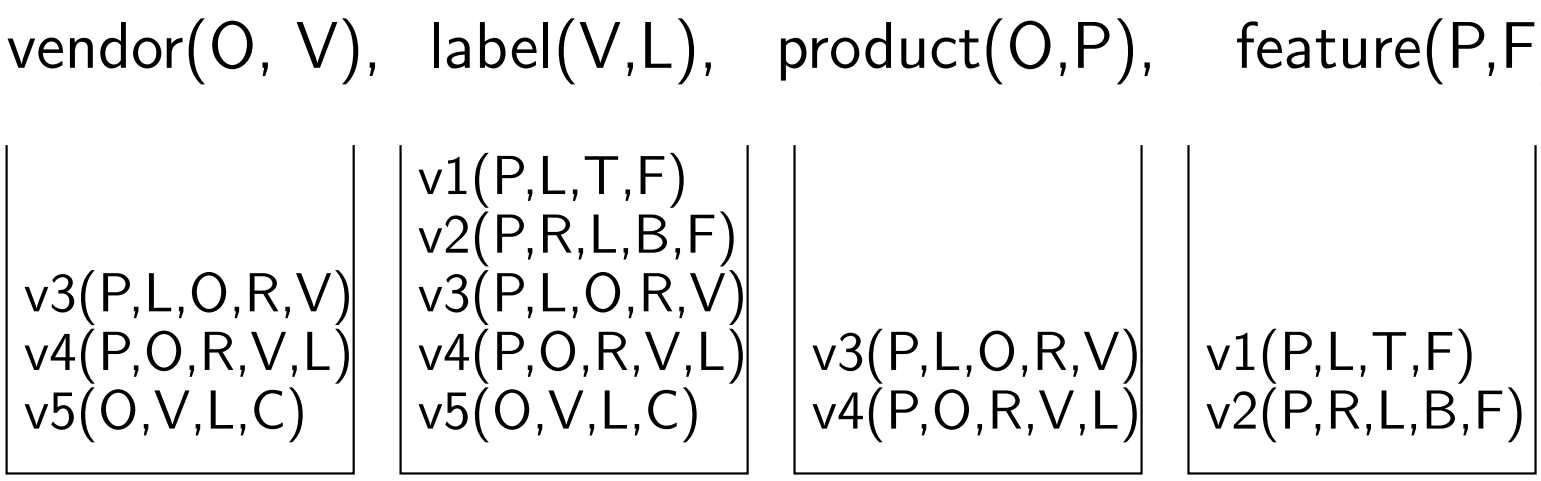
SemLAV is low cost

vendor(O, V), label(V,L), product(O,P), feature(P,F)



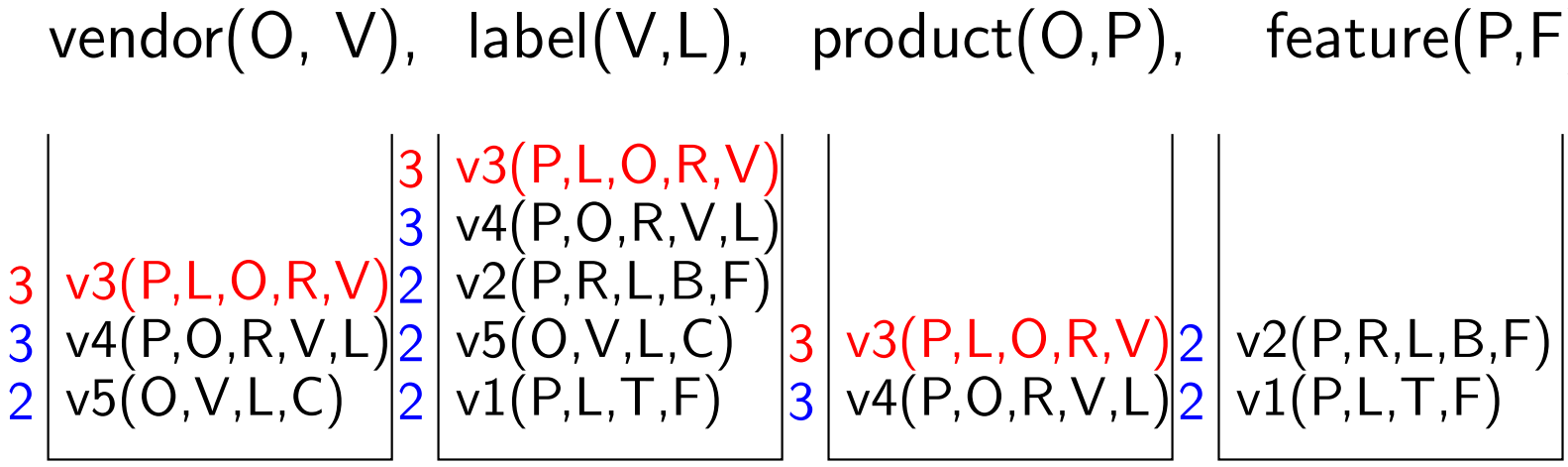
v1(P,L,T,F) :- label(P,L), type(P,T), feature(P,F)
v2(P,R,L,F) :- producer(P,R), label(R,L), feature(P,F)
v3(P,L,O,V) :- label(P,L), product(O,P), vendor(O,V)
v4(P,O,V,L) :- product(O,P),vendor(O,V),label(V,L)
v5(O,V,L,C) :- vendor(O,V), label(V,L), country(V,C)

SemLAV is low cost



v1(P,L,T,F) :- label(P,L), type(P,T), feature(P,F)
v2(P,R,L,F) :- producer(P,R), label(R,L), feature(P,F)
v3(P,L,O,V) :- label(P,L), product(O,P), vendor(O,V)
v4(P,O,V,L) :- product(O,P),vendor(O,V),label(V,L)
v5(O,V,L,C) :- vendor(O,V), label(V,L), country(V,C)

SemLAV is low cost



v1(P,L,T,F) :- label(P,L), type(P,T), feature(P,F)
v2(P,R,L,F) :- producer(P,R), label(R,L), feature(P,F)
v3(P,L,O,V) :- label(P,L), product(O,P), vendor(O,V)
v4(P,O,V,L) :- product(O,P),vendor(O,V),label(V,L)
v5(O,V,L,C) :- vendor(O,V), label(V,L), country(V,C)

SemLAV's ranking strategy covers more query rewritings

vendor(O, V), label(V,L), product(O,P), feature(P,F)

		3	v3(P,L,O,R,V)				
		3	v4(P,O,R,V,L)				
3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)				
3	v4(P,O,R,V,L)	2	v5(O,V,L,C)	3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)
2	v5(O,V,L,C)	2	v1(P,L,T,F)	3	v4(P,O,R,V,L)	2	v1(P,L,T,F)

k	Another Order		SemLAV Order	
	V'_k	# Covered rewritings	V_k	# Covered rewritings
1	v5	0	v3	0
2	v5, v1	0	v3, v2	2
3	v5, v1, v3	6	v3, v2, v4	12
4	v5, v1, v3, v2	8	v3, v2, v4, v1	32
5	v5, v1, v3, v2, v4	60	v3, v2, v4, v1, v5	60

SemLAV's ranking strategy covers more query rewritings

vendor(O, V), label(V,L), product(O,P), feature(P,F)

		3	v3(P,L,O,R,V)				
		3	v4(P,O,R,V,L)				
3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)				
3	v4(P,O,R,V,L)	2	v5(O,V,L,C)	3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)
2	v5(O,V,L,C)	2	v1(P,L,T,F)	3	v4(P,O,R,V,L)	2	v1(P,L,T,F)

k	Another Order		SemLAV Order	
	V'_k	# Covered rewritings	V_k	# Covered rewritings
1	v5	0	v3	0
2	v5, v1	0	v3, v2	2
3	v5, v1, v3	6	v3, v2, v4	12
4	v5, v1, v3, v2	8	v3, v2, v4, v1	32
5	v5, v1, v3, v2, v4	60	v3, v2, v4, v1, v5	60

SemLAV's ranking strategy covers more query rewritings

vendor(O, V), label(V,L), product(O,P), feature(P,F)

		3	v3(P,L,O,R,V)				
		3	v4(P,O,R,V,L)				
3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)				
3	v4(P,O,R,V,L)	2	v5(O,V,L,C)	3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)
2	v5(O,V,L,C)	2	v1(P,L,T,F)	3	v4(P,O,R,V,L)	2	v1(P,L,T,F)

k	Another Order		SemLAV Order	
	V'_k	# Covered rewritings	V_k	# Covered rewritings
1	v5	0	v3	0
2	v5, v1	0	v3, v2	2
3	v5, v1, v3	6	v3, v2, v4	12
4	v5, v1, v3, v2	8	v3, v2, v4, v1	32
5	v5, v1, v3, v2, v4	60	v3, v2, v4, v1, v5	60

SemLAV's ranking strategy covers more query rewritings

vendor(O, V), label(V,L), product(O,P), feature(P,F)

		3	v3(P,L,O,R,V)				
		3	v4(P,O,R,V,L)				
3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)				
3	v4(P,O,R,V,L)	2	v5(O,V,L,C)	3	v3(P,L,O,R,V)	2	v2(P,R,L,B,F)
2	v5(O,V,L,C)	2	v1(P,L,T,F)	3	v4(P,O,R,V,L)	2	v1(P,L,T,F)

k	Another Order		SemLAV Order	
	V'_k	# Covered rewritings	V_k	# Covered rewritings
1	v5	0	v3	0
2	v5, v1	0	v3, v2	2
3	v5, v1, v3	6	v3, v2, v4	12
4	v5, v1, v3, v2	8	v3, v2, v4, v1	32
5	v5, v1, v3, v2, v4	60	v3, v2, v4, v1, v5	60

Experimental Setup

- 10 million triples Berlin Benchmark synthetic dataset[7].
- Existing set of 16 queries and 9 views[8], plus 5 additional views to access all relevant data.
- Linux server, 20 GB of RAM allocated.

---

[7] C. Bizer and A. Schultz. "The Berlin SPARQL Benchmark". In: *Int. J. Semantic Web Inf. Syst.* (2009).

[8] R. Castillo-Espinola. "Indexing RDF data using materialized SPARQL queries". PhD thesis. Humboldt-Universität zu Berlin, 2012.

Experimental Setup

- 14 views divided in 34 parts $\Rightarrow$ 476 views.
- Query rewriter-based approaches: MiniCon[9], MCD-SAT[10], GQR[11].
- Metrics: number of triples, time of the first answer, throughput.
- Timeout: 10 min.

---

[9] R. Pottinger and A. Y. Halevy. "MiniCon: A scalable algorithm for answering queries using views". In: *VLDB J.* (2001).

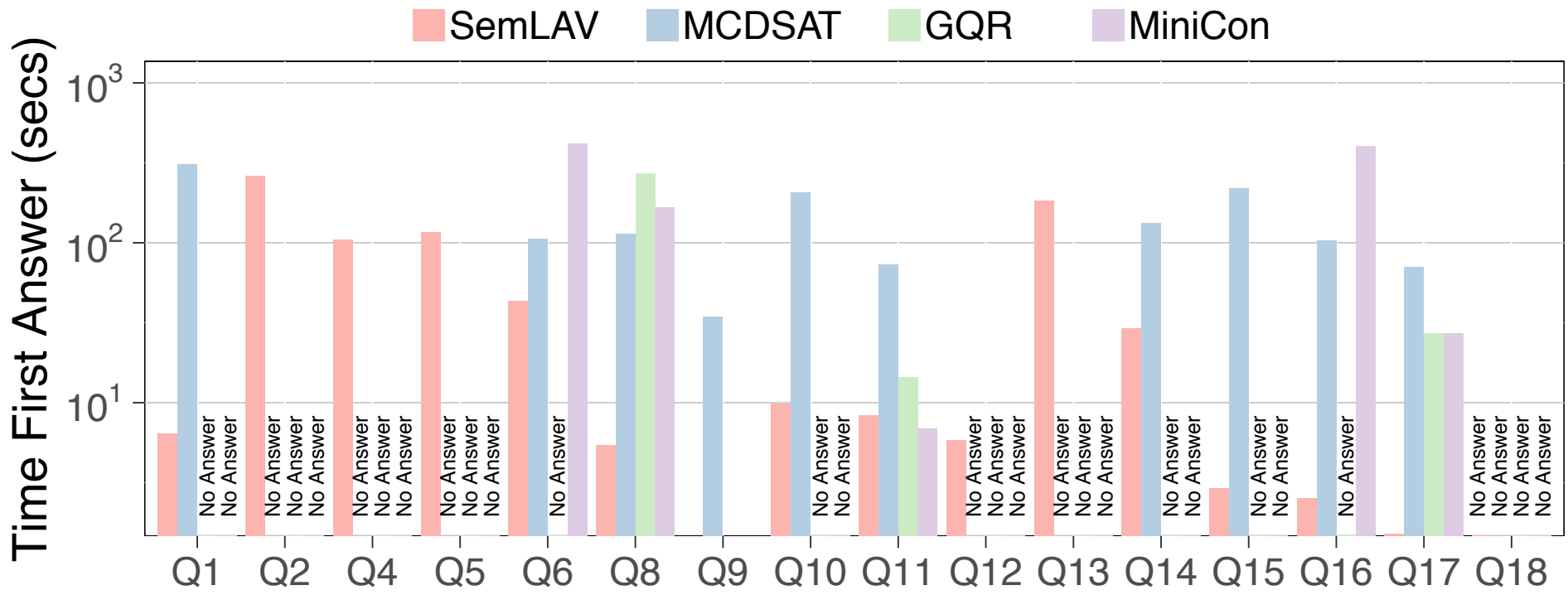
[10] Y. Arvelo et al. "Compilation of Query-Rewriting Problems into Tractable Fragments of Propositional Logic". In: *AAAI*. 2006.

[11] G. Konstantinidis and J. L. Ambite. "Scalable query rewriting: a graph-based approach". In: *SIGMOD Conference*. 2011.

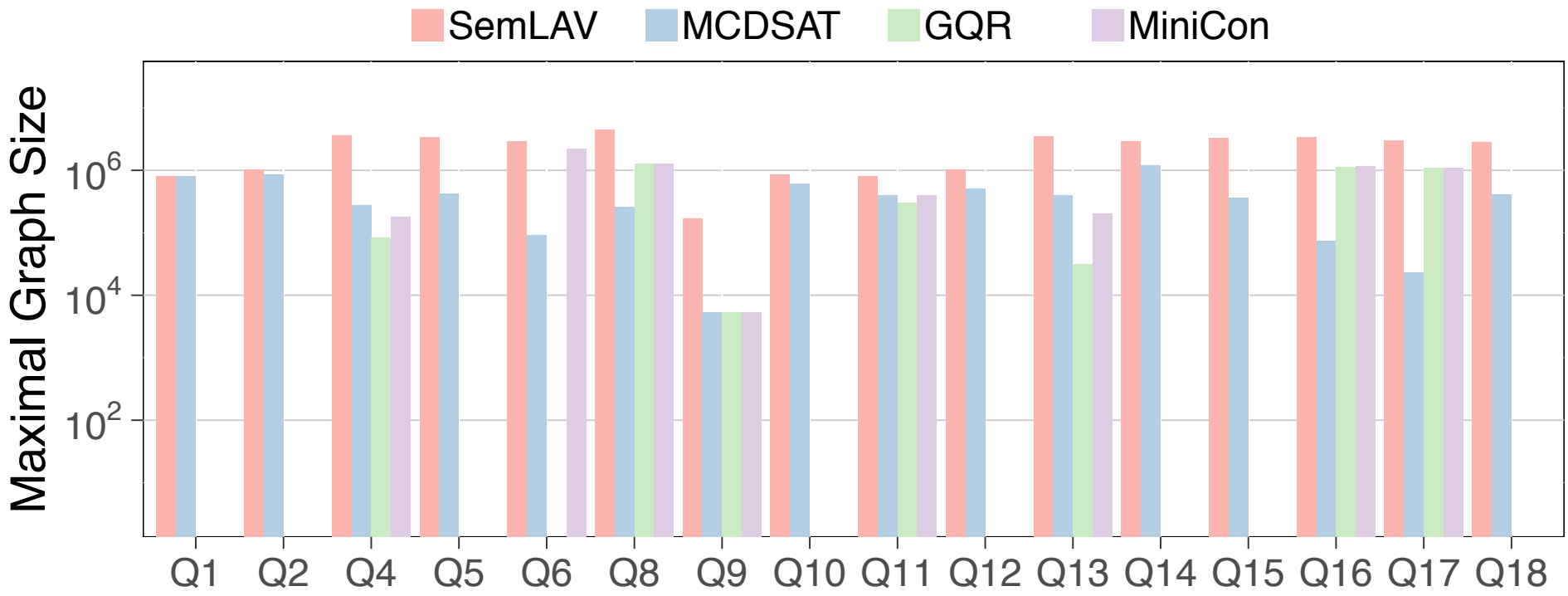
Expected Results

- First answer faster.
- Higher memory consumption.
- Higher answer throughput.

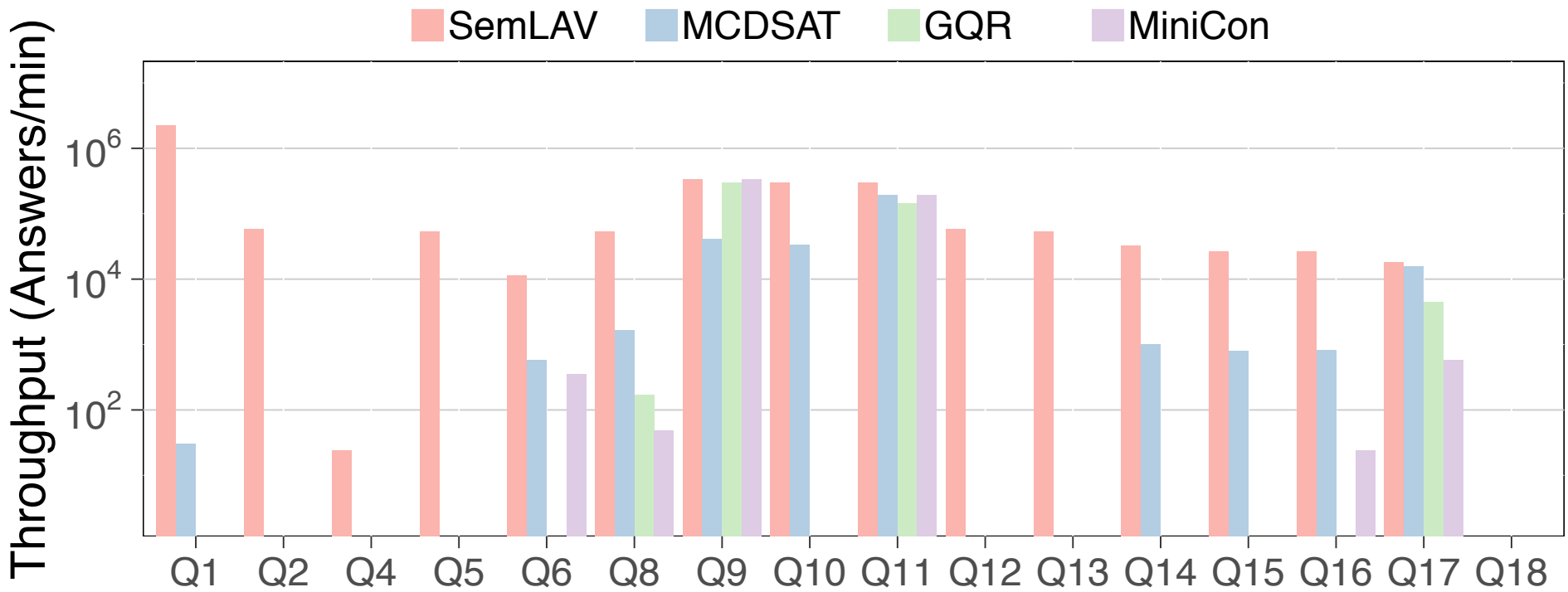
SemLAV produces answers faster than the rewriting-based approaches



SemLAV consumes more memory than the rewriting-based approaches



SemLAV's throughput is higher than the rewriting-based approaches



Conclusions

- SemLAV **avoids expensive** query rewriting generation and evaluation.
- SemLAV **ranks first** the views that can cover more query subgoals, then loading the top ranked views will **more likely produce answers**.

Perspectives

- Improve the implementation by loading views in parallel[12].
- Include strategies to handle memory limitations.

[12] P. Folz et al. "Parallel Data Loading during Querying Deep Web and Linked Open Data with SPARQL". . In: *SSWS*. 2015.

Addressed Issues

- How to integrate heterogeneous data, and query it using Semantic Web technologies without any statistic about the data distribution?
- **How to overcome the data availability limitations of SPARQL endpoints?**

SPARQL endpoints availability issue may be addressed

- Using **fragmentation** and **replication**.
- But in Linked Data, each client declares her federation, then **the replication and fragmentation schema cannot be designed**.
- Federated query engines have to **adapt** to the **opportunities** found during query execution.

Replicated data decreases federated query engines performance

```
select distinct ?p ?m ?n ?d where {
  ?p  dbprop:name ?m .
  ?p  dbprop:nationality ?n .
  ?p  dbprop:doctoralAdvisor ?d
}
```

#DBpedia Replicas	FedX[13] Execution Time (ms)
1	1,392
2	215,907

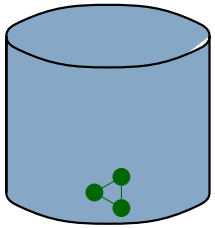
[13] A. Schwarte et al. "FedX: Optimization Techniques for Federated Query Processing on Linked Data". In: *ISWC*. 2011.

Users may replicate only the fragments relevant for their queries

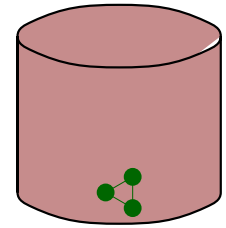
- A triple pattern fragment is defined by the **dataset** it has been replicated from and by a CONSTRUCT query with **a triple pattern**.
  - Fragment with the doctoral advisors triples: <http://**dbpedia**.org/sparql, CONSTRUCT WHERE { **?p dbprop:doctoralAdvisor ?a** }>
- Replicating fragments from different datasets provides **new data localities** and opens new opportunities for optimization.

Querying federations with replicated fragments

DBpedia



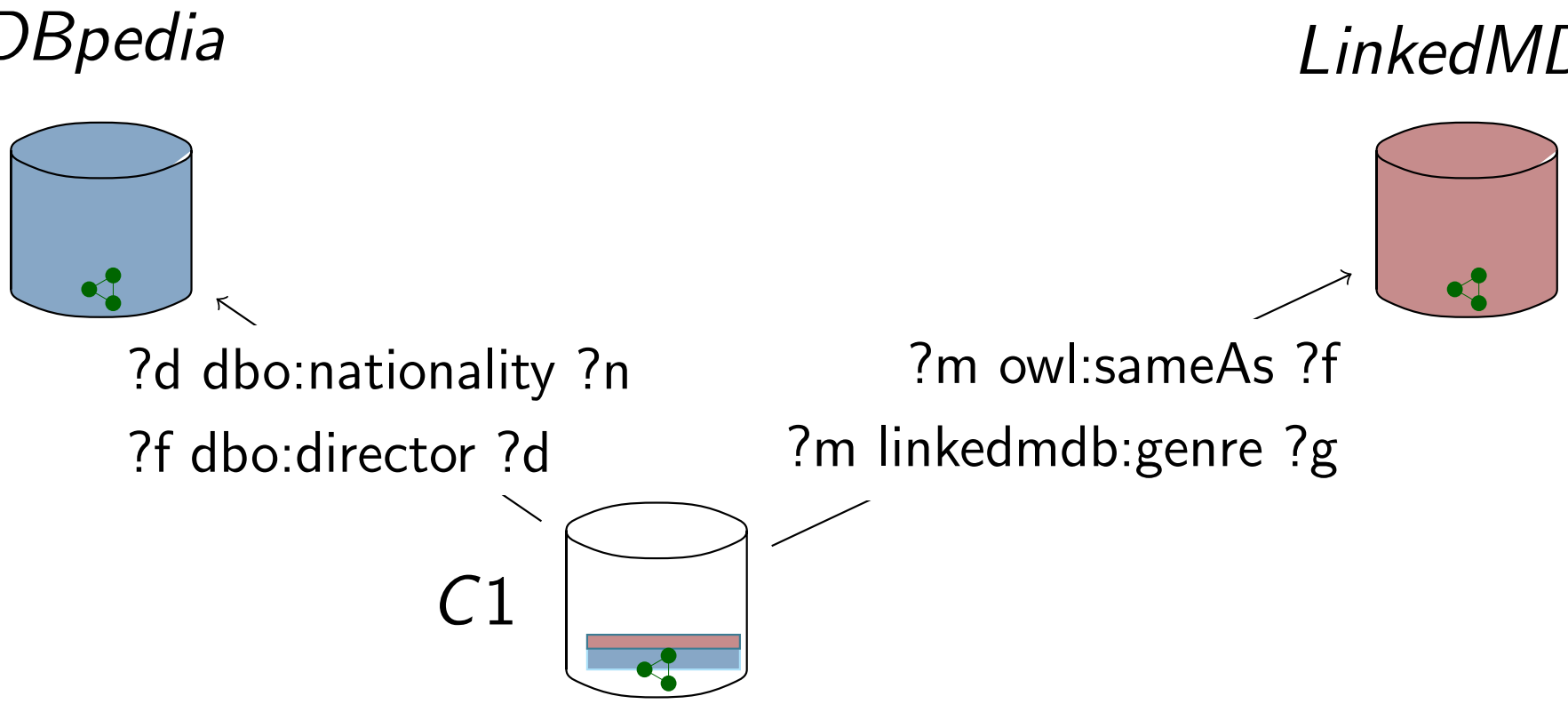
LinkedMDB



```
select distinct * where {
?director dbo : nationality ?nat .
?film dbo : director ?director .
?movie owl : sameAs ?film .
?movie linkedmdb : genre ?genre }
```

client

Querying federations with replicated fragments



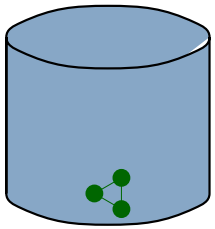

```
select distinct * where {
?director dbo : nationality ?nat .
?film dbo : director ?director .
?movie owl : sameAs ?film .
?movie linkedmdb : genre ?genre }
```

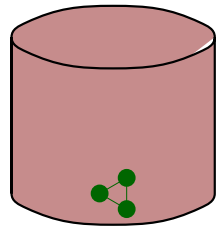
client

Querying federations with replicated fragments

DBpedia

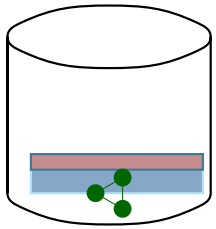
LinkedMDB





Tuples to transfer	s1	s2	s3	s4	s5
DBpedia	166,177	3,229	3,229	0	0
LinkedMDB	76,180	13,430	0	13,430	0
C1	242,357	0	13,430	3,229	48
Execution Time (s)	20.22	2.64	2.50	2.79	0.65

C1



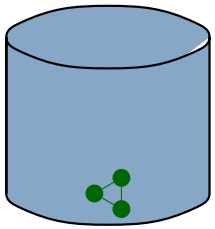
select distinct * **where** {
?director *dbo* : *nationality* ?nat .
?film *dbo* : *director* ?director .
?movie *owl* : *sameAs* ?film .
?movie *linkedmdb* : *genre* ?genre }

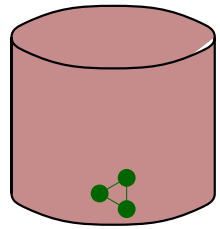
client

Querying federations with replicated fragments

DBpedia

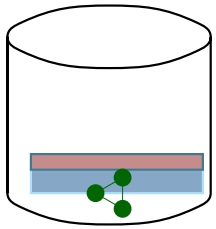
LinkedMDB





Tuples to transfer	s1	s2	s3	s4	s5
DBpedia	166,177	3,229	3,229	0	0
LinkedMDB	76,180	13,430	0	13,430	0
C1	242,357	0	13,430	3,229	48
Execution Time (s)	20.22	2.64	2.50	2.79	0.65

C1



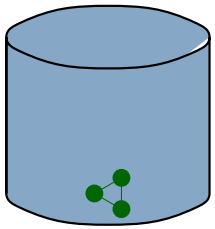
```
select distinct * where {
?director dbo : nationality ?nat .
?film dbo : director ?director .
?movie owl : sameAs ?film .
?movie linkedmdb : genre ?genre }
```

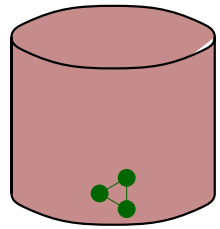
client

Querying federations with replicated fragments

DBpedia

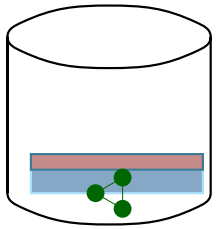
LinkedMDB





Tuples to transfer	s1	s2	s3	s4	s5
DBpedia	166,177	3,229	3,229	0	0
LinkedMDB	76,180	13,430	0	13,430	0
C1	242,357	0	13,430	3,229	48
Execution Time (s)	20.22	2.64	2.50	2.79	0.65

C1



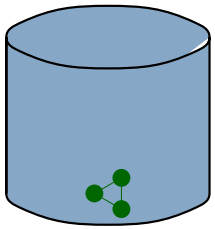
```
select distinct * where {
?director dbo : nationality ?nat .
?film dbo : director ?director .
?movie owl : sameAs ?film .
?movie linkedmdb : genre ?genre }
```

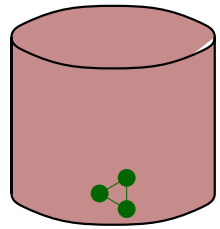
client

Querying federations with replicated fragments

DBpedia

LinkedMDB

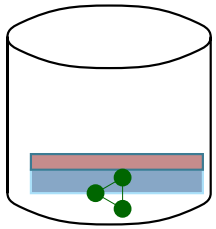




Tuples to transfer	s1	s2	s3	s4	s5
DBpedia	166,177	3,229	3,229	0	0
LinkedMDB	76,180	13,430	0	13,430	0
C1	242,357	0	13,430	3,229	48
Execution Time (s)	20.22	2.64	2.50	2.79	0.65

?d dbo:nationality ?n ?m owl:sameAs ?f
?f dbo:director ?d ?m linkedmdb:genre ?g

C1



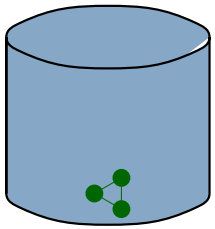
```
select distinct * where {
?director dbo : nationality ?nat .
?film dbo : director ?director .
?movie owl : sameAs ?film .
?movie linkedmdb : genre ?genre }
```

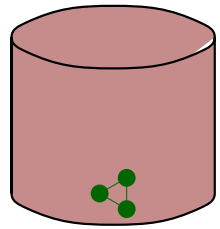
client

Querying federations with replicated fragments

DBpedia

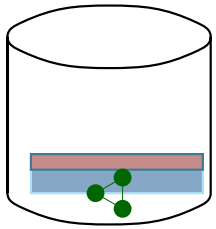
LinkedMDB





Tuples to transfer	s1	s2	s3	s4	s5
DBpedia	166,177	3,229	3,229	0	0
LinkedMDB	76,180	13,430	0	13,430	0
C1	242,357	0	13,430	3,229	48
Execution Time (s)	20.22	2.64	2.50	2.79	0.65

C1



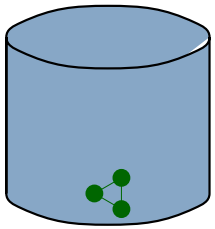
```
select distinct * where {
?director dbo : nationality ?nat .
?film dbo : director ?director .
?movie owl : sameAs ?film .
?movie linkedmdb : genre ?genre }
```

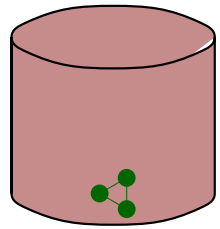
client

Querying federations with replicated fragments

DBpedia

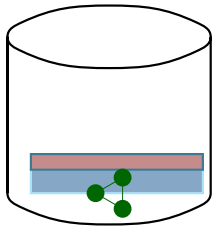
LinkedMDB





Tuples to transfer	s1	s2	s3	s4	s5
DBpedia	166,177	3,229	3,229	0	0
LinkedMDB	76,180	13,430	0	13,430	0
C1	242,357	0	13,430	3,229	48
Execution Time (s)	20.22	2.64	2.50	2.79	0.65

C1



```
select distinct * where {
?director dbo : nationality ?nat .
?film dbo : director ?director .
?movie owl : sameAs ?film .
?movie linkedmdb : genre ?genre }
```

client

Assumptions

- Replicated fragments are perfectly synchronized.
- Data accessible through the endpoints are described as fragments.

Research Question

- How should be changed the source selection of federated query engines in order to query **SPARQL endpoints with replicated fragments** and do not degrade federated query engines performance?

The Source Selection Problem with Fragment Replication (SSP-FR)

Given a SPARQL query and a set of SPARQL endpoints with replicated fragments, choose the SPARQL endpoints to contact for each query triple pattern in order to produce a **complete query answer** and **transfer the minimum amount of data**.

Intuitions

1. The knowledge about **fragment replication** can be used to **reduce** the number of selected sources by federated query engines while producing the same answers.
2. Considering **groups of triple patterns** to be executed together, instead of individual triple patterns, may produce source selections that lead to **transfer less data** from endpoints to the federated query engine.

The FEDRA Source Selection Algorithm

1. Selection of relevant fragments and fragment pruning using **query containment**.
2. Multiple relevant fragments $\rightarrow$ **UNION Reduction**: try to reduce to one fragment.
3. One relevant fragment $\rightarrow$ **BGP Reduction**: reduce to set covering problem to evaluate in as few endpoints as possible.

Non Redundant Relevant Fragment Selection

BGP Triple Pattern		Relevant Fragments	Relevant Endpoints
tp1	?director dbo:nationality ?nat		
tp2	?film dbo:director ?director		
tp3	?movie owl:sameAs ?film		
tp4	?movie linkedmdb:genre ?genre		

f1 : <dbpedia , ?director dbo:nationality ?nat>
f2 : <dbpedia , ?film dbo:director ?director>
f3 : <linkedmdb , ?movie owl:sameAs ?film>
f4 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:4>
f5 : <dbpedia , ?director dbo:nationality dbr:France>
f6 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:12>
fragments mapping = {(f1, {C1}), (f2 ,{C1 , C3}),
(f3 , {C1 , C2}), (f4,{C2,C4}), (f5, {C2}), (f6, {C2,C3})}

Non Redundant Relevant Fragment Selection

BGP Triple Pattern		Relevant Fragments	Relevant Endpoints
tp1	?director dbo:nationality ?nat	f1 f5	{C1} {C2}
tp2	?film dbo:director ?director		
tp3	?movie owl:sameAs ?film		
tp4	?movie linkedmdb:genre ?genre		

f1 : <dbpedia , ?director dbo:nationality ?nat>
f2 : <dbpedia , ?film dbo:director ?director>
f3 : <linkedmdb , ?movie owl:sameAs ?film>
f4 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:4>
f5 : <dbpedia , ?director dbo:nationality dbr:France>
f6 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:12>
fragments mapping = {(f1, {C1}), (f2 ,{C1 , C3}),
(f3 , {C1 , C2}), (f4,{C2,C4}), (f5, {C2}), (f6, {C2,C3})}

Union Reduction

BGP Triple Pattern		Relevant Fragments	Relevant Endpoints
tp1	?director dbo:nationality ?nat	f1 f5	{C1} {C2}
tp2	?film dbo:director ?director	f2	{C1, C3}
tp3	?movie owl:sameAs ?film	f3	{C1, C2}
tp4	?movie linkedmdb:genre ?genre	f4 f6	{C2, C4} {C2, C3}

f1 : <dbpedia , ?director dbo:nationality ?nat>
f2 : <dbpedia , ?film dbo:director ?director>
f3 : <linkedmdb , ?movie owl:sameAs ?film>
f4 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:4>
f5 : <dbpedia , ?director dbo:nationality dbr:France>
f6 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:12>
fragments mapping = {(f1, {C1}), (f2 ,{C1 ,C3}),
(f3 , {C1 ,C2}), (f4,{C2,C4}), (f5, {C2}), (f6, {C2,C3})}

Union Reduction

BGP Triple Pattern		Relevant Fragments	Relevant Endpoints
tp1	?director dbo:nationality ?nat	f1 f5	{C1} {C2}
tp2	?film dbo:director ?director	f2	{C1, C3}
tp3	?movie owl:sameAs ?film	f3	{C1, C2}
tp4	?movie linkedmdb:genre ?genre	f4 f6	{C2, C4} {C2, C3}

f1 : <dbpedia , ?director dbo:nationality ?nat>
f2 : <dbpedia , ?film dbo:director ?director>
f3 : <linkedmdb , ?movie owl:sameAs ?film>
f4 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:4>
f5 : <dbpedia , ?director dbo:nationality dbr:France>
f6 : <linkedmdb , ?movie linkedmdb:genre lmdbgenre:12>
fragments mapping = {(f1, {C1}), (f2 ,{C1 ,C3}),
(f3 , {C1 ,C2}), (f4,{C2,C4}), (f5, {C2}), (f6, {C2,C3})}

BGP Reduction

	BGP Triple Pattern	Relevant Fragments	Relevant Endpoints
tp1	?director dbo:nationality ?nat	f1 f5	{C1} {C2}
tp2	?film dbo:director ?director	f2	{C1, C3}
tp3	?movie owl:sameAs ?film	f3	{C1, C2}
tp4	?movie linkedmdb:genre ?genre	f' f6	{C2, C4} {C2, C3}

$S = \{ tp1, tp2, tp3, tp4 \}$

$C_{C1} = \{ tp1, tp2, tp3 \}$

$C_{C2} = \{ tp3, tp4 \}$

$C_{C3} = \{ tp2 \}$

BGP Reduction

	BGP Triple Pattern	Relevant Fragments	Relevant Endpoints
tp1	?director dbo:nationality ?nat	f1 f5	{C1} {C2}
tp2	?film dbo:director ?director	f2	{C1, C3}
tp3	?movie owl:sameAs ?film	f3	{C1, C2}
tp4	?movie linkedmdb:genre ?genre	f' f6	{C2, C4} {C2, C3}

$S = \{ tp1, tp2, tp3, tp4 \}$

$C_{C1} = \{ tp1, tp2, tp3\}$

$C_{C2} = \{ tp3, tp4\}$

$C_{C3} = \{ tp2 \}$

BGP Reduction

	BGP Triple Pattern	Relevant Fragments	Relevant Endpoints
tp1	?director dbo:nationality ?nat	f1 f5	{C1} {C2}
tp2	?film dbo:director ?director	f2	{C1, C3}
tp3	?movie owl:sameAs ?film	f3	{C1, C2}
tp4	?movie linkedmdb:genre ?genre	f' f6	{C2, C4} {C2, C3}

$S = \{ tp1, tp2, tp3, tp4 \}$

$C_{C1} = \{ tp1, tp2, tp3\}$

$C_{C2} = \{ tp3, tp4\}$

$C_{C3} = \{ tp2 \}$

Experimental Setup

- WatDiv synthetic dataset[14] with 100 thousand and 10 million triples (WatDiv1 and WatDiv100).
- Real datasets: Diseasome, Semantic Web Dog Food (SWDF), Linked Movie Data Base (LMDB), DBpedia GeoCoordinates (Geo).
- Federations of 10 endpoints that access random replicated fragments.
- Execution of 100 random queries.

[14] G. Aluç et al. "Diversified Stress Testing of RDF Data Management Systems". In: *ISWC*. 2014.

Experimental Setup

- Federated query engines: FedX[15] and ANAPSID[16].
- Additional source selection strategies: DAW[17].
- Metrics: number of selected sources and number of transferred tuples.
- Timeout: 30 min.

---

[15] A. Schwarte et al. "FedX: Optimization Techniques for Federated Query Processing on Linked Data". In: *ISWC*. 2011.

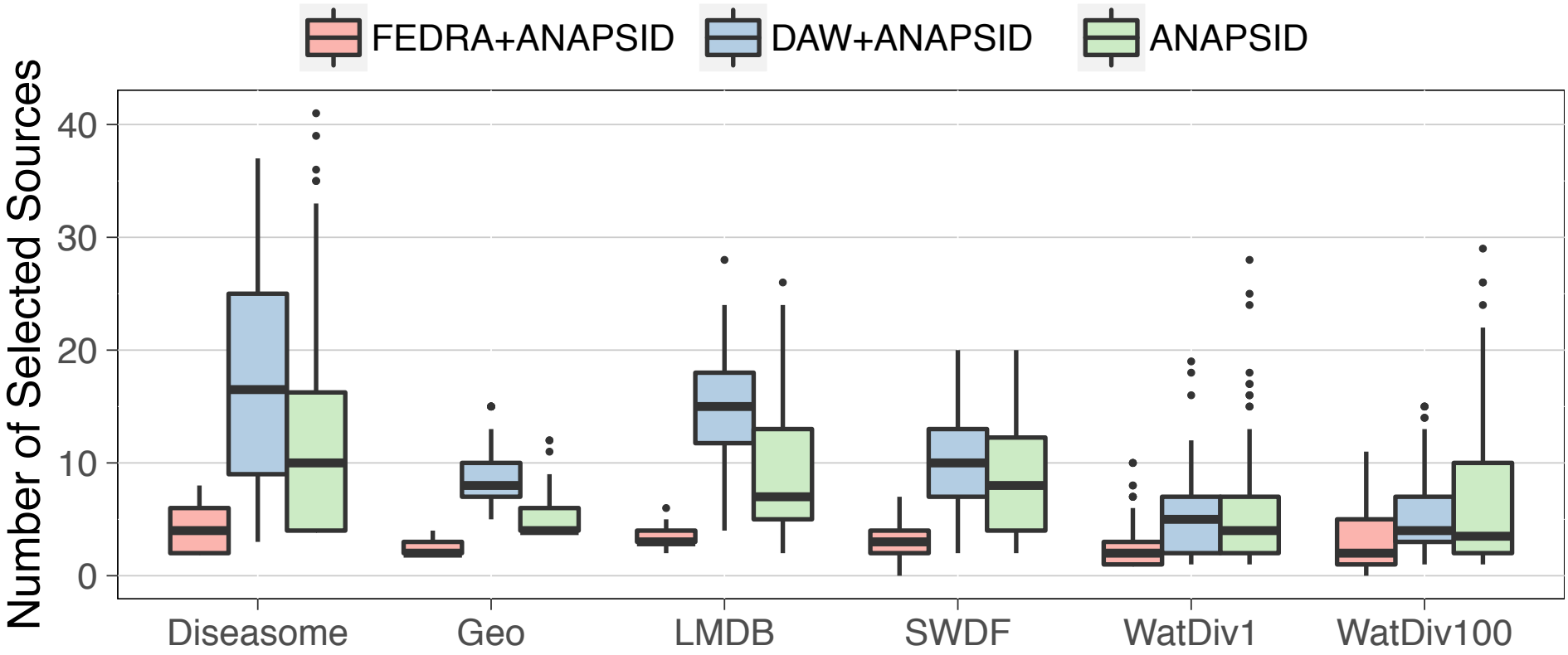
[16] M. Acosta et al. "ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints". In: *ISWC*. 2011.

[17] M. Saleem et al. "DAW: Duplicate-AWare Federated Query Processing over the Web of Data". In: *ISWC*. 2013.

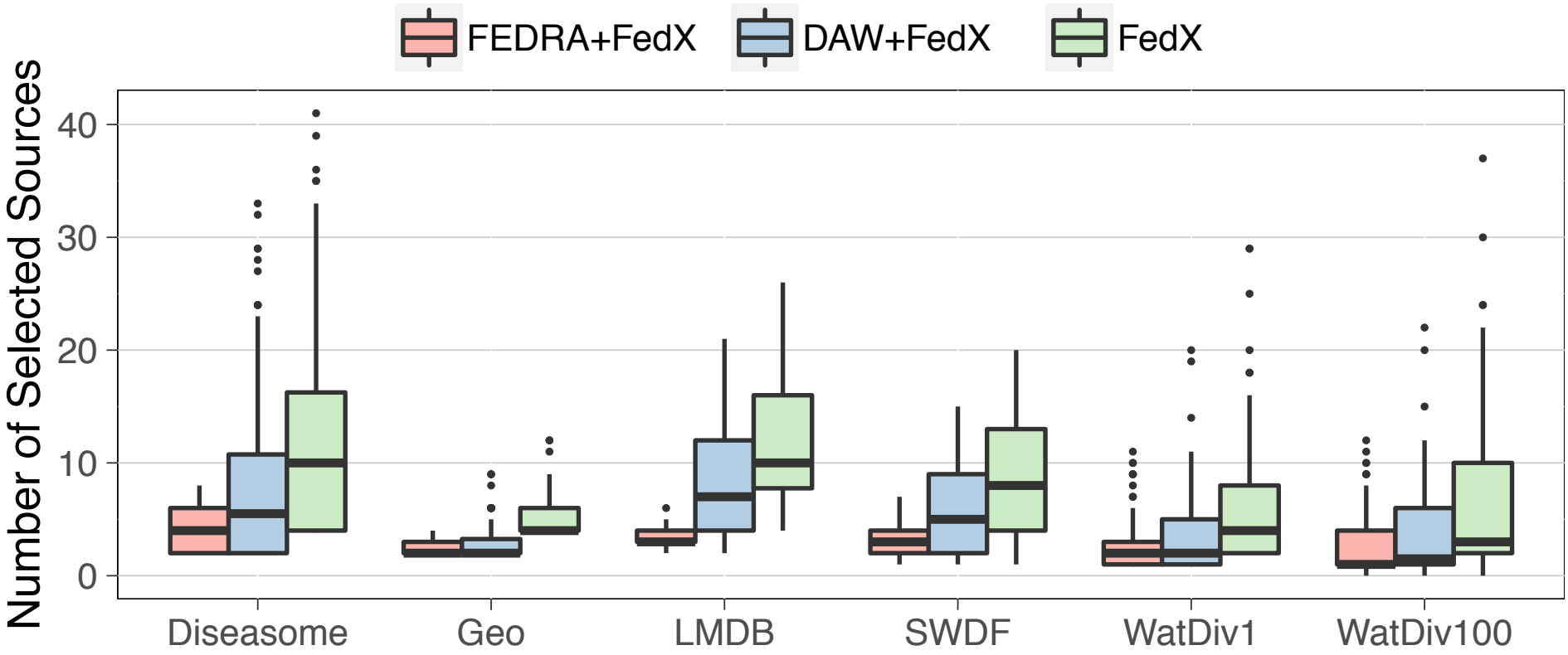
Expected Results

- FEDRA should select significantly less sources.
- FEDRA+FedX and FEDRA+ANAPSID should transfer less tuples.

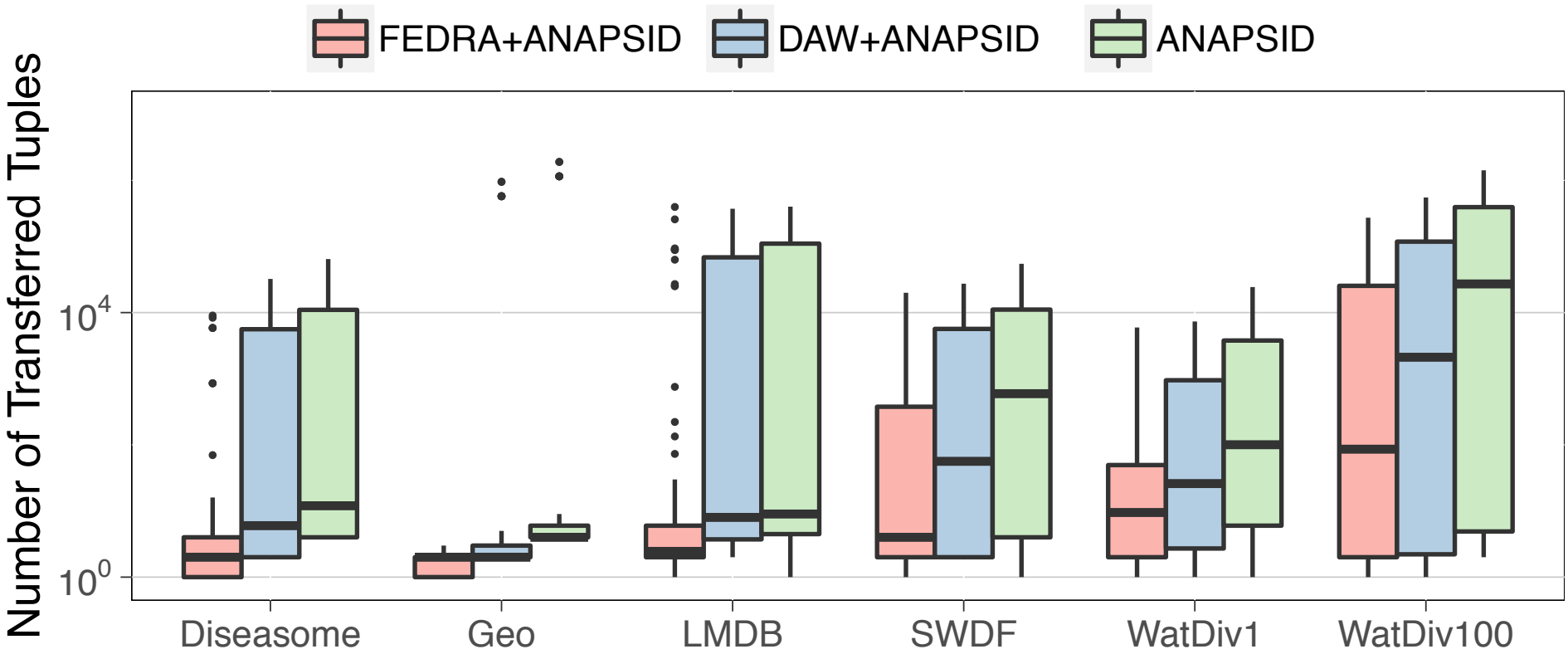
FEDRA uses known replicated fragments to effectively reduce the number of selected sources



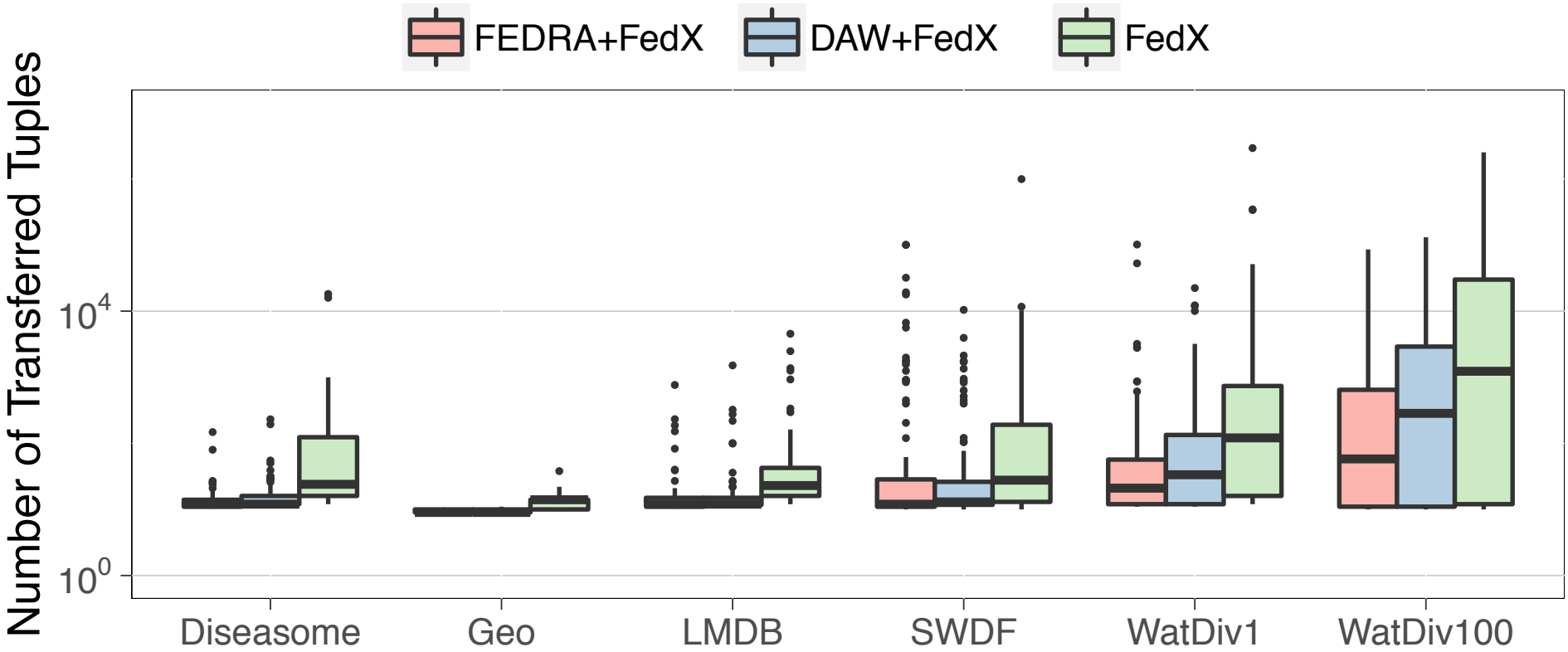
Replicated fragments give FEDRA a perfect summary of endpoints data



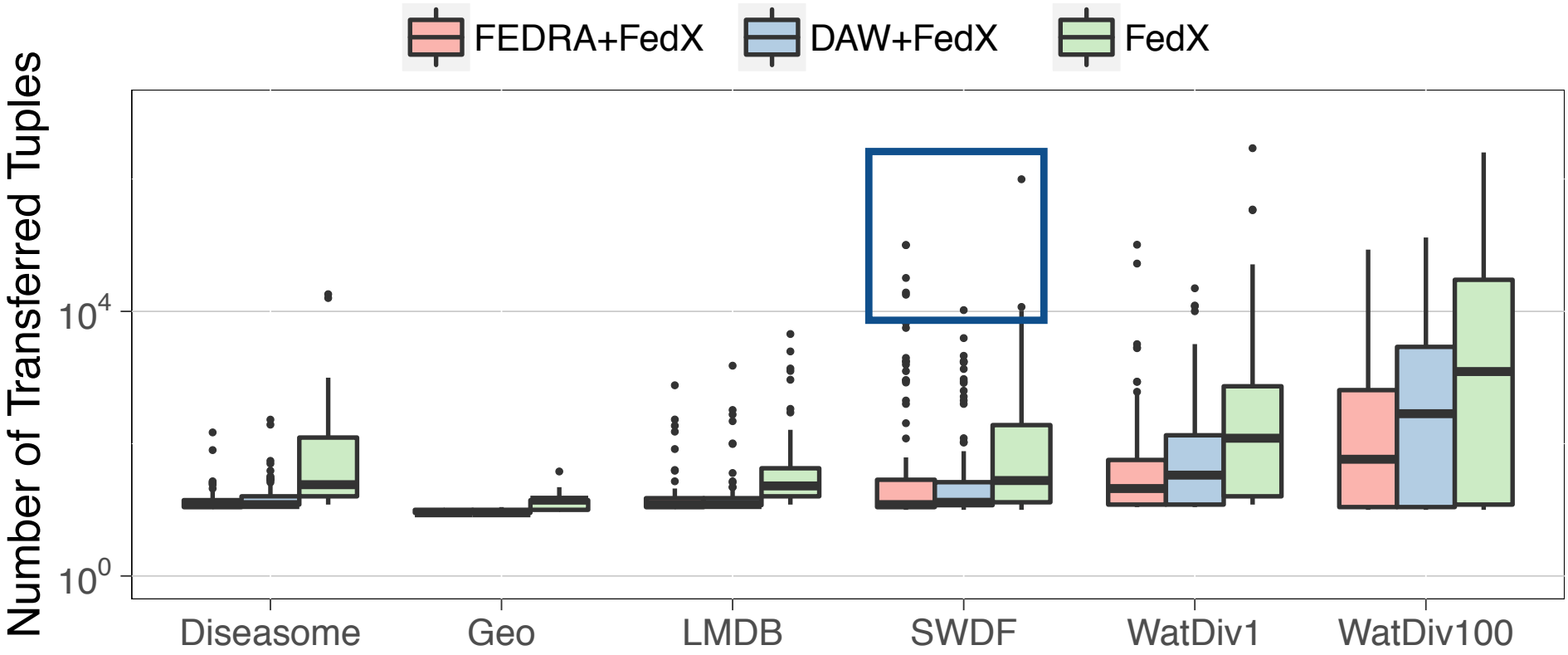
The Number of Transferred Tuples has been reduced



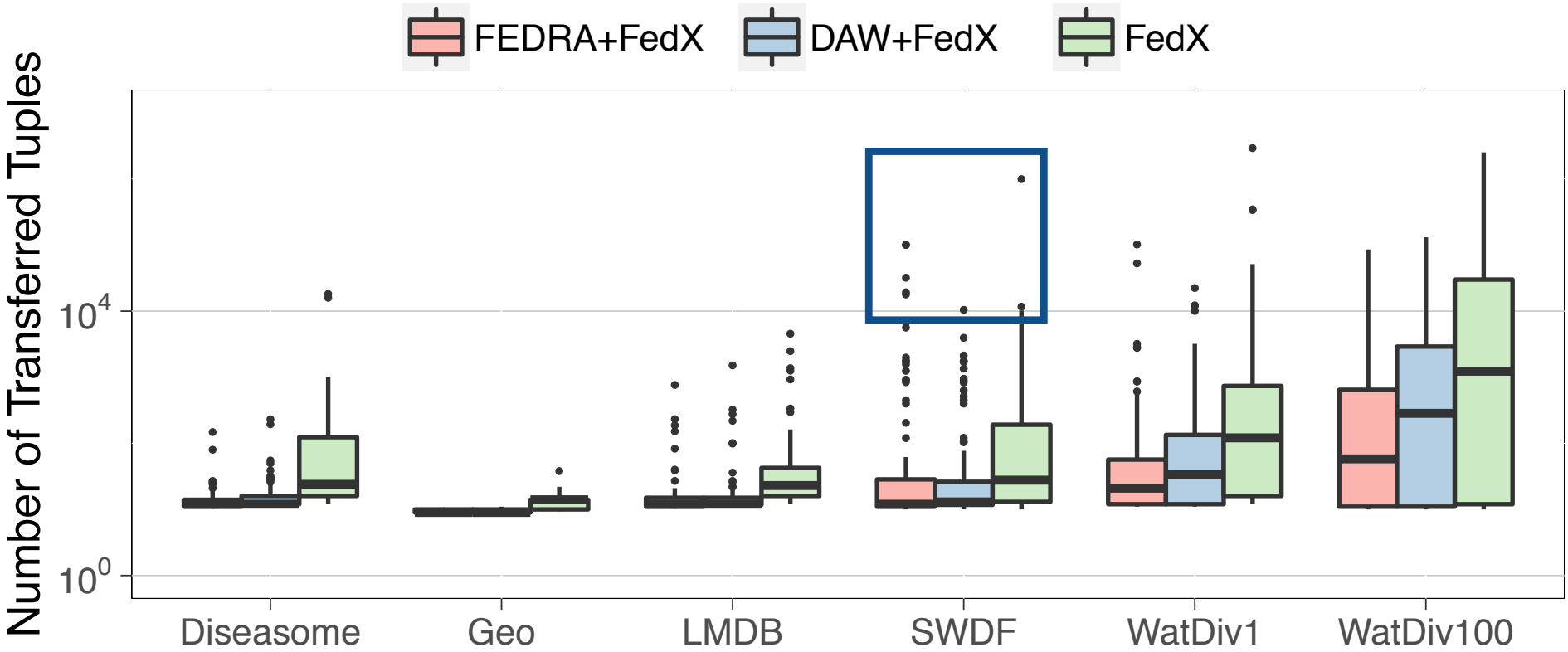
FEDRA has delegated join evaluation to the endpoints



FEDRA has delegated join evaluation to the endpoints

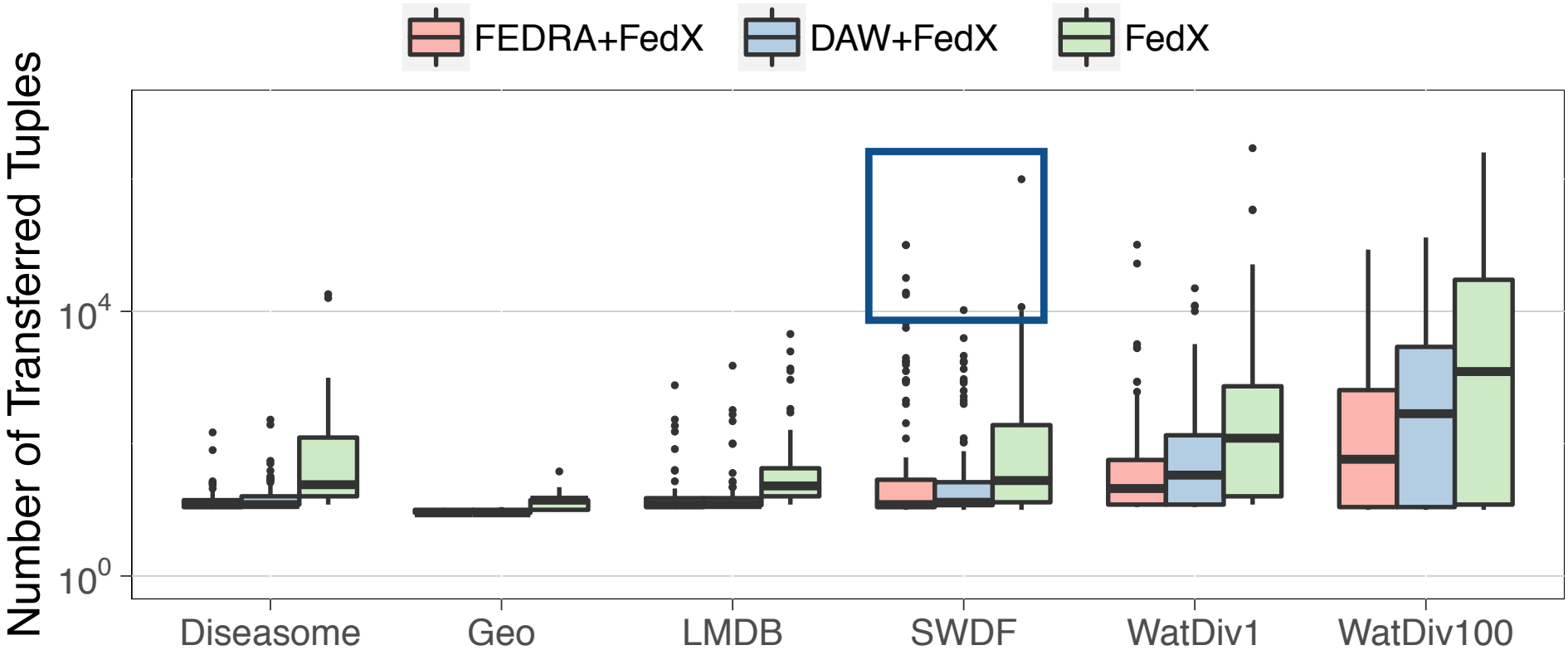


FEDRA has delegated join evaluation to the endpoints



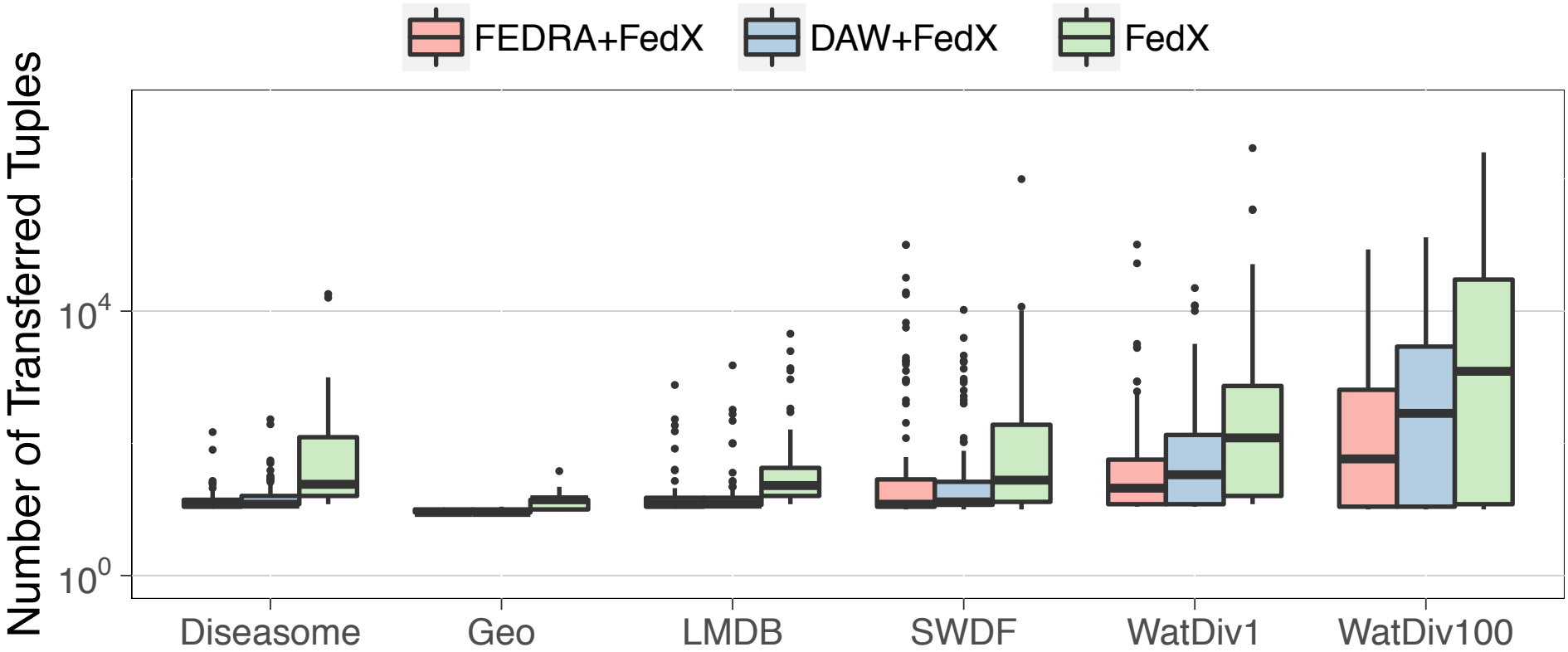
SERVICE <C1> { ?director dbo:nationality ?nat . ?movie owl:sameAs ?film }

FEDRA has delegated join evaluation to the endpoints



SERVICE <C1> { ?director dbo:nationality ?nat }
SERVICE <C1> { ?movie owl:sameAs ?film }
SERVICE <C2> { ?movie owl:sameAs ?film }

FEDRA has delegated join evaluation to the endpoints



Conclusions

- **Partial replication** in Linked Data.
- FEDRA performs a **BGP aware source selection**, and exploits **fragment localities** to **reduce** the number of transferred tuples.
- FEDRA achieves a **great reduction** of the number of selected sources and the number of transferred tuples by ANAPSID and FedX.

Perspective: query decomposition

- Limitation:
  FEDRA **source selection** is performed **before** knowing how the query is **decomposed**.
- Perspective:
  Replication aware **query decomposition** strategies[18].

[18] G. Montoya et al. "Decomposing Federated Queries in presence of Replicated Fragments". In: *JWS* (2016). Under revision.

Perspective: divergent fragments

- Limitation:
  FEDRA works under the assumption:
  replicated fragments are perfectly synchronized.
- Perspective:
  Integrate **metrics** to measure the **staleness** of selected sources[19].

[19] G. Montoya et al. *Fedra: Query Processing for SPARQL Federations with Divergence*. Tech. rep. Université de Nantes, May 2014.

Perspective: cost-based source selection

- Limitation:
  FEDRA **only** considers that **data transfer** should be minimized.
- Perspective:
  Taken into account **other factors**, e.g., preferences about the endpoints[20].

[20] G. Montoya et al. *Fedra: Query Processing for SPARQL Federations with Divergence*. Tech. rep. Université de Nantes, May 2014.

Perspective: take advantage of replicated fragments

- Limitation:
  FEDRA **does not exploit replicated data** to speed up query processing.
- Perspective:
  Share the subquery evaluation among the endpoints that provide the same relevant data[21].

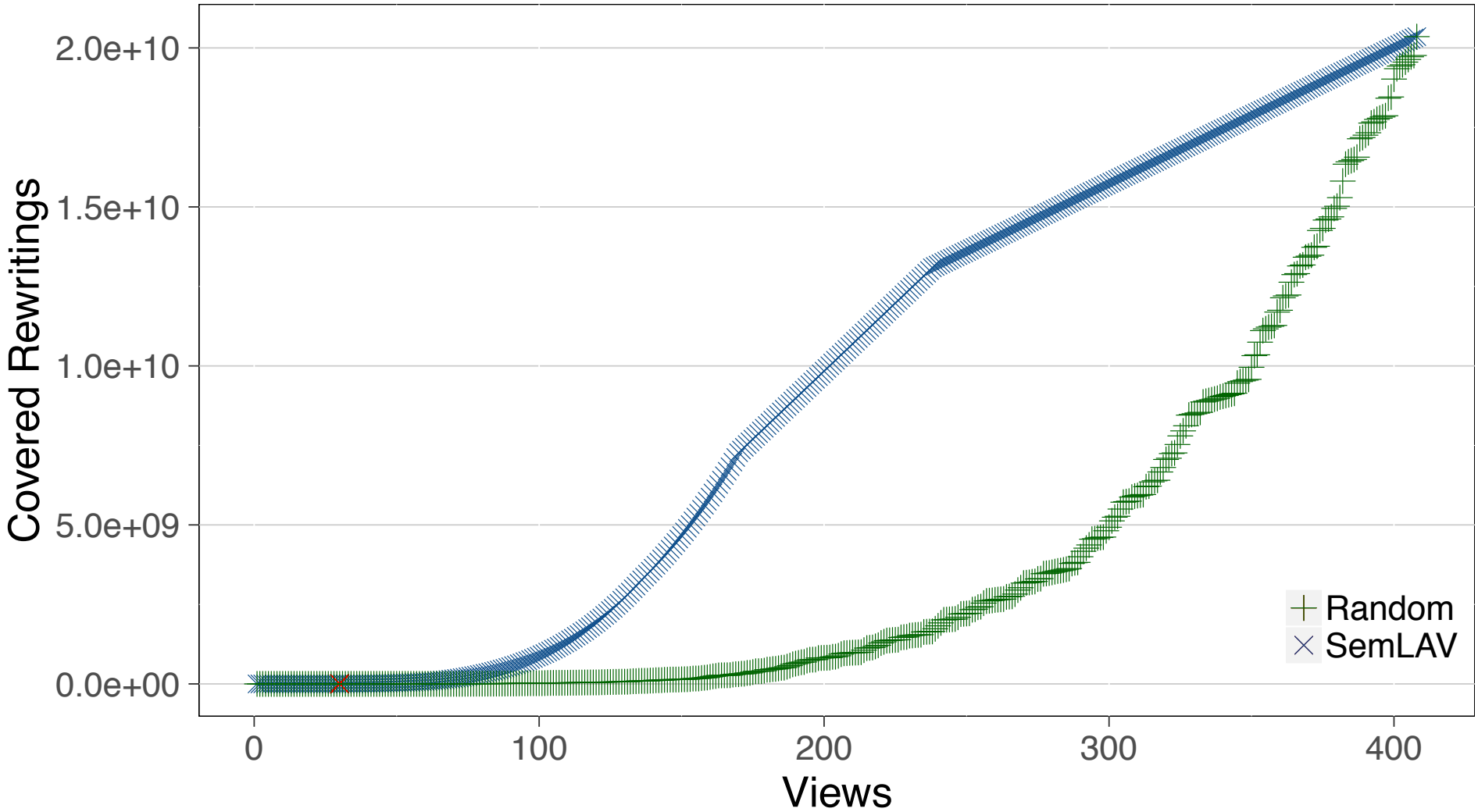
[21] R. Bao and D. Perrai. *Increasing the benefits obtained from replicated data during query execution*. Report. Apr. 2015.

Questions?

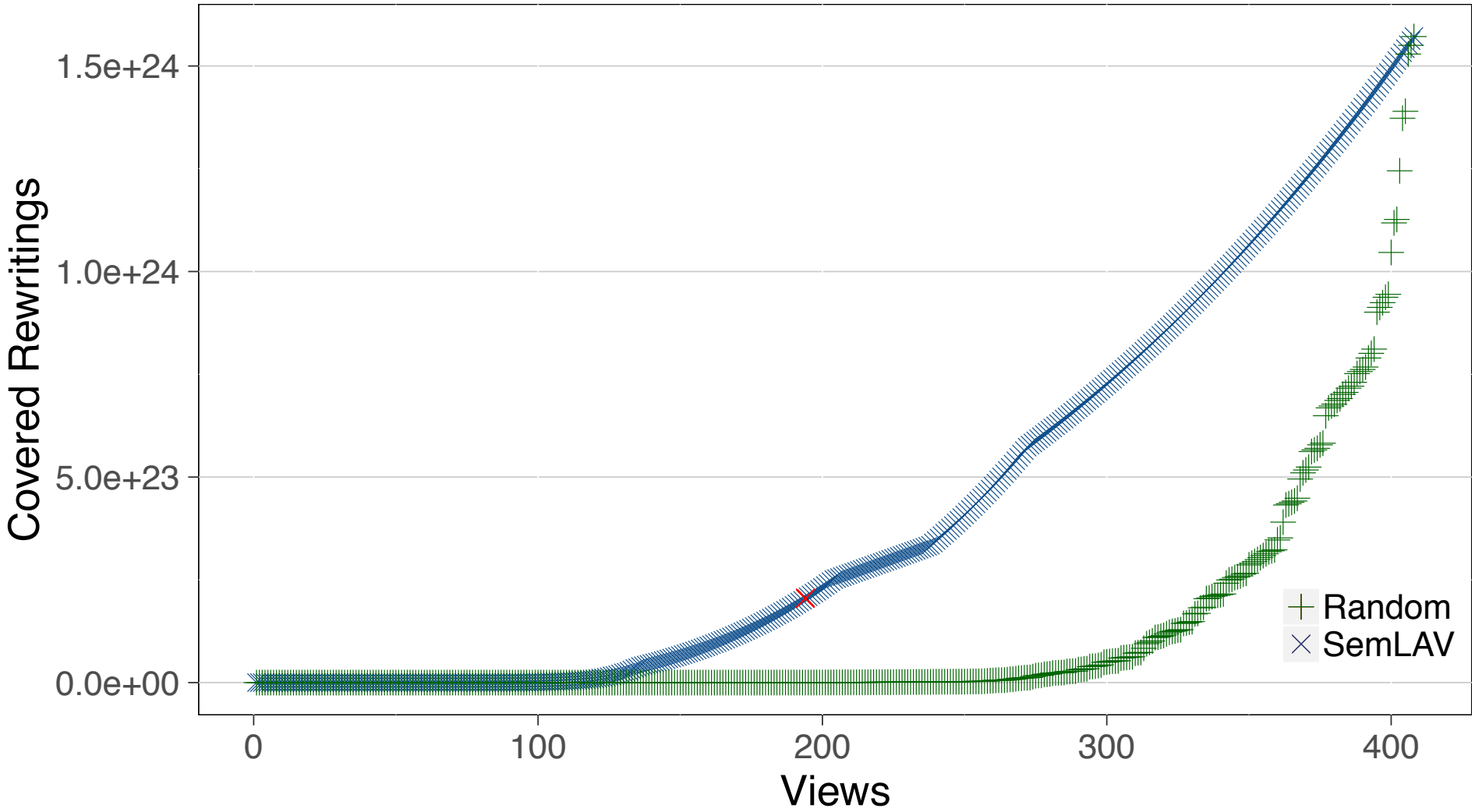
Publications

- Gabriela Montoya, Hala Skaf-Molli, Pascal Molli, and Maria-Esther Vidal. Federated SPARQL Queries Processing with Replicated Fragments. In The Semantic Web-ISWC 2015 - 14th International Semantic Web Conference, pages 36–51, Bethlehem, United States, October 2015. (SSP-FR + FEDRA)
- Gabriela Montoya, Luis Daniel Ibáñez, Hala Skaf-Molli, Pascal Molli, and Maria-Esther Vidal. SemLAV: Local-As-View Mediation for SPARQL Queries. Transactions on Large-Scale Data- and Knowledge-Centered Systems XIII, pages 33–58, 2014 (MaxCov + SemLAV)
- Pauline Folz, Gabriela Montoya, Hala Skaf-Molli, Pascal Molli, and Maria-Esther Vidal. SemLAV: Querying Deep Web and Linked Open Data with SPARQL. In ESWC: Extended Semantic Web Conference, volume 476 of The Semantic Web: ESWC 2014 Satellite Events, pages 332 – 337, Anissaras/Hersonissou, Greece, May 2014 (Demonstration of SemLAV).
- Pauline Folz, Gabriela Montoya, Hala Skaf-Molli, Pascal Molli, and Maria-Esther Vidal. Parallel data loading during querying deep web and linked open data with SPARQL. In Thorsten Liebig and Achille Fokoue, editors, Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 11, 2015., volume 1457 of CEUR Workshop Proceedings, pages 63–74. CEUR-WS.org, 2015 (SemLAV's optimization).
- Gabriela Montoya, Luis Daniel Ibáñez, Hala Skaf-Molli, Pascal Molli, and Maria-Esther Vidal. GUN: An efficient execution strategy for querying the web of data. In Hendrik Decker, Lenka Lhotská, Sebastian Link, Josef Basl, and A Min Tjoa, editors, DEXA (1), volume 8055 of Lecture Notes in Computer Science, pages 180–194. Springer, 2013 (Predecessor of SemLAV).
- Maria-Esther Vidal, Simon Castillo, Maribel Acosta, Gabriela Montoya, and Guillermo Palma. On the Selection of SPARQL Endpoints to Efficiently Execute Federated SPARQL Queries. Transactions on Large-Scale Data- and Knowledge-Centered Systems, 2015.
- Gabriela Montoya. Answering SPARQL Queries using Views. ISWC-DC 2015 The ISWC 2015 Doctoral Consortium, pages 33–40, 2015 (Doctoral Consortium).

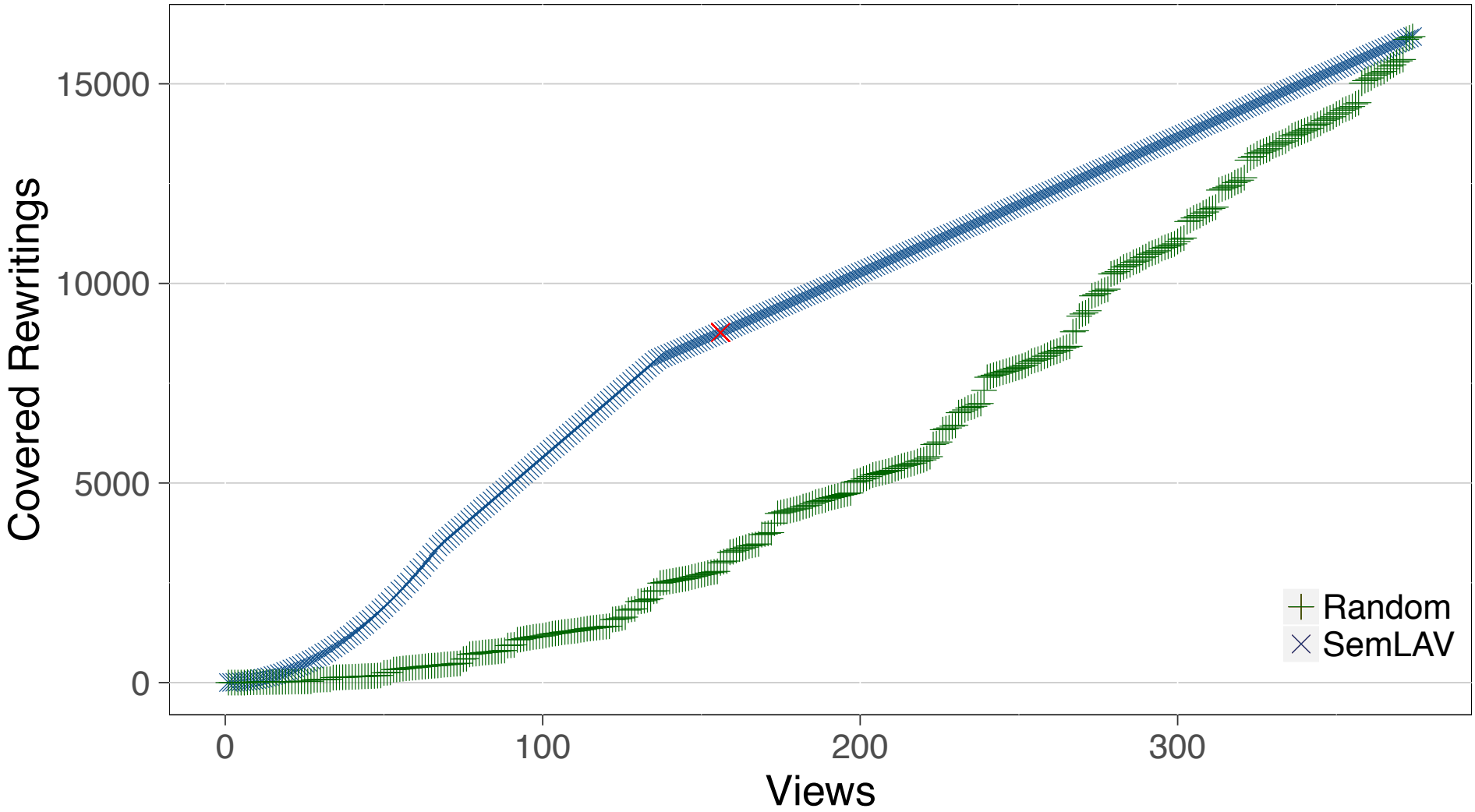
View loading order matters - Q1



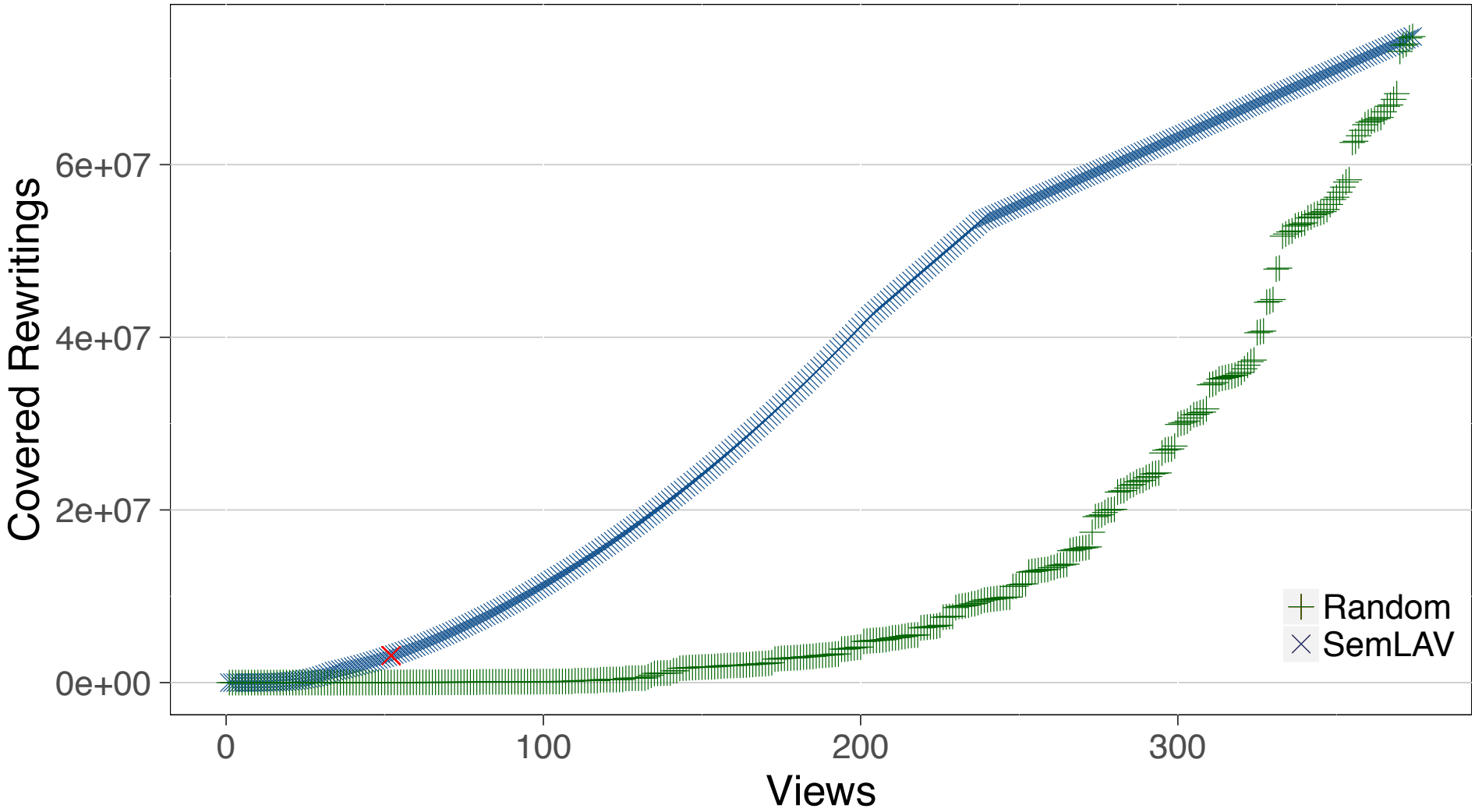
View loading order matters - Q2



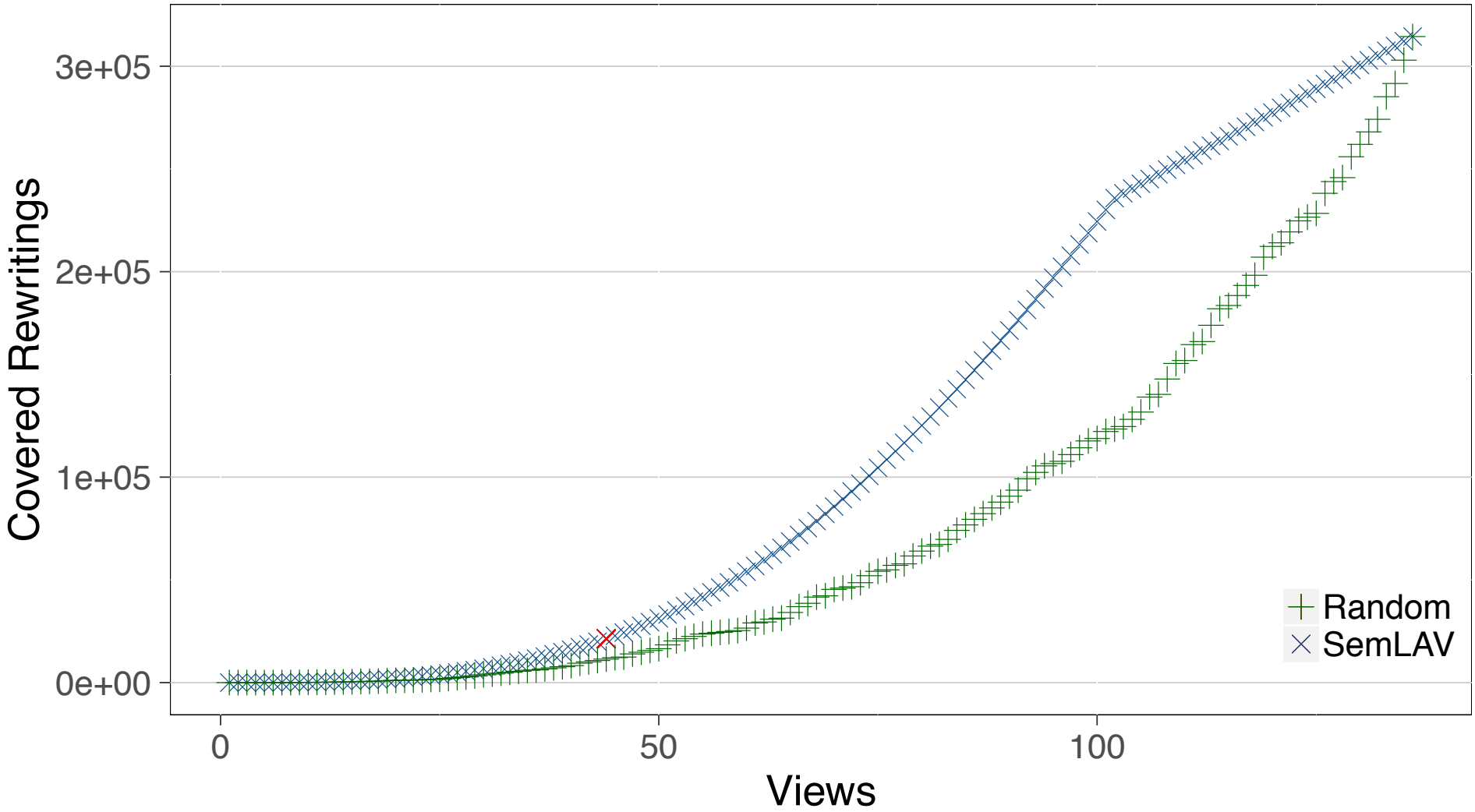
View loading order matters - Q4



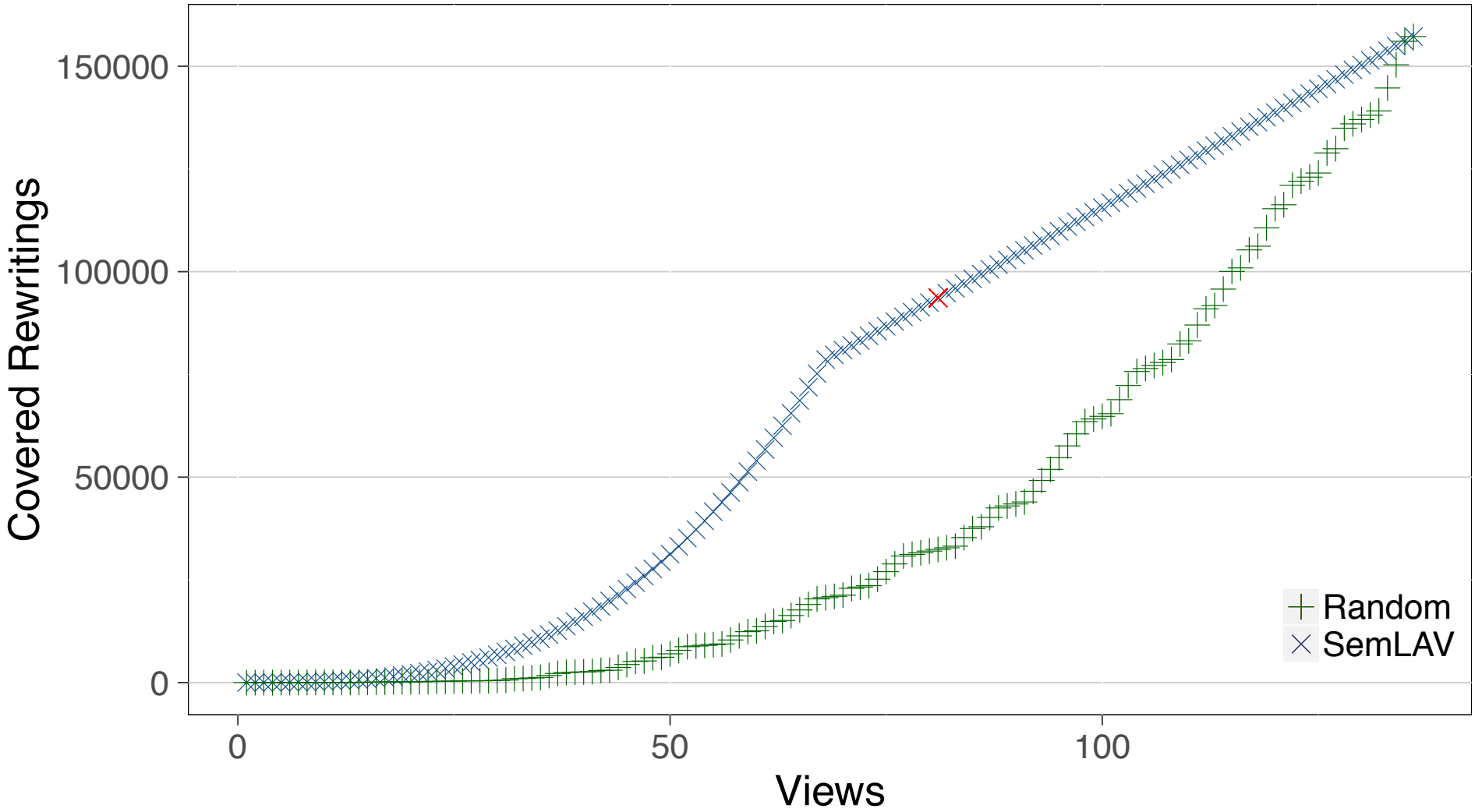
View loading order matters - Q5



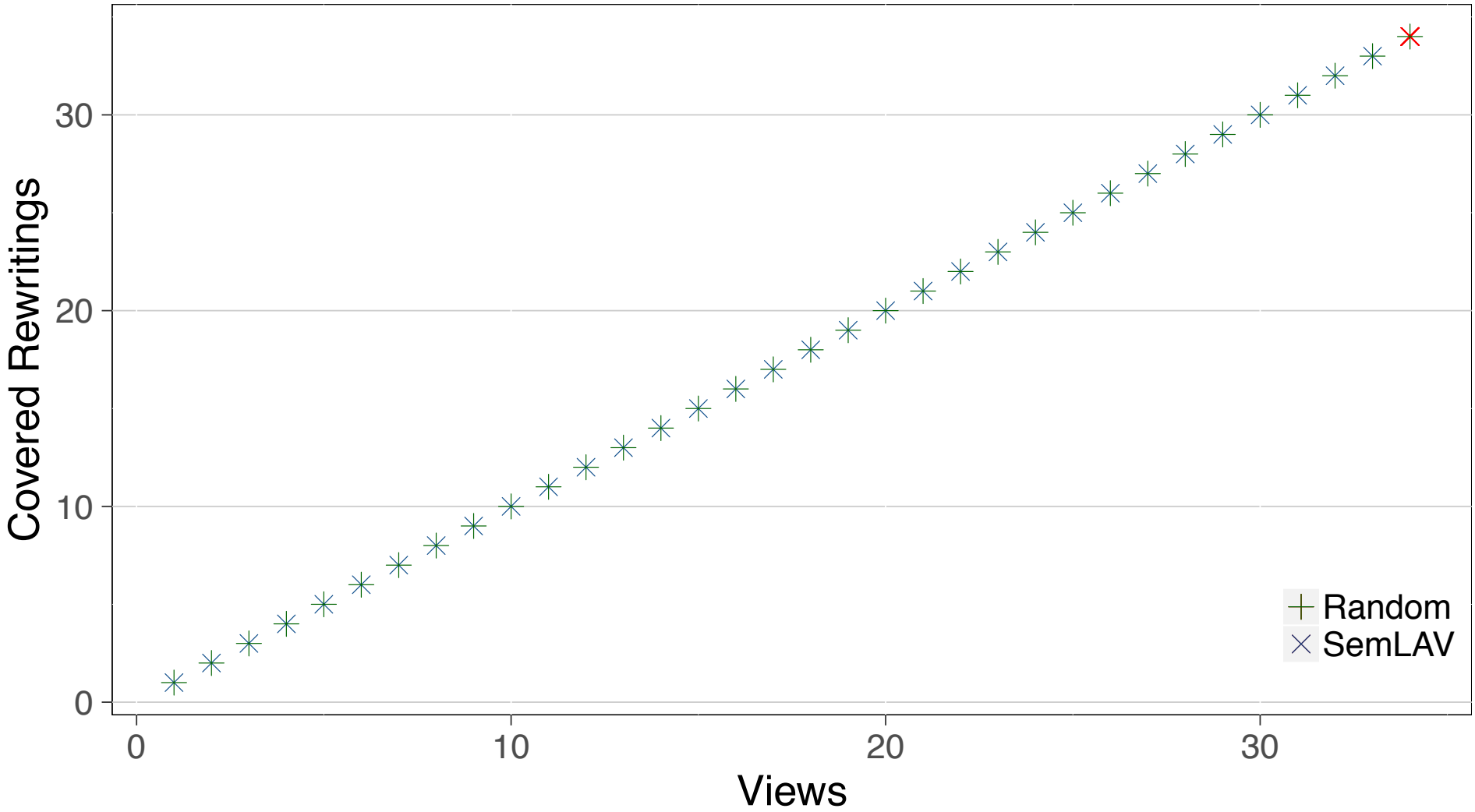
View loading order matters - Q6



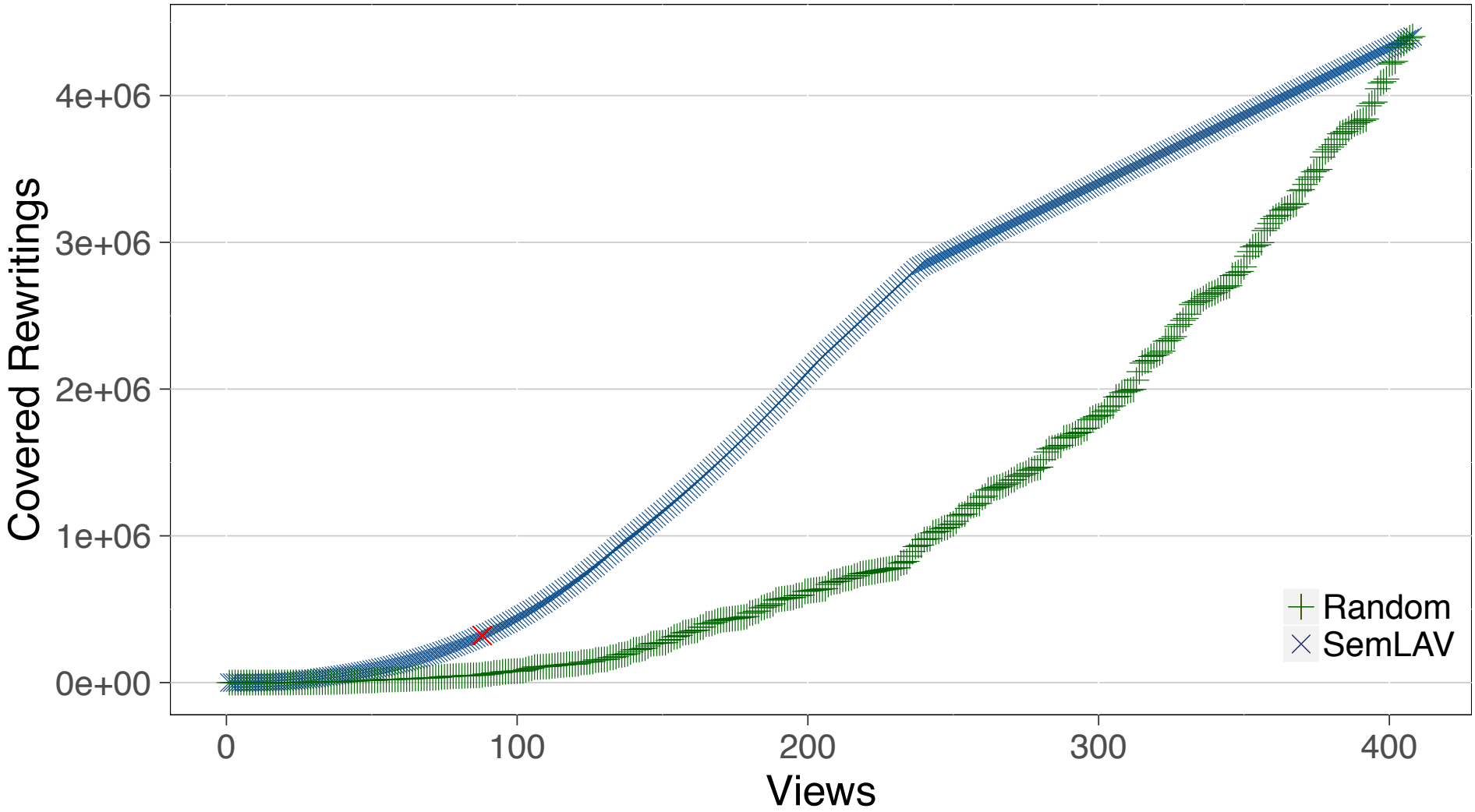
View loading order matters - Q8



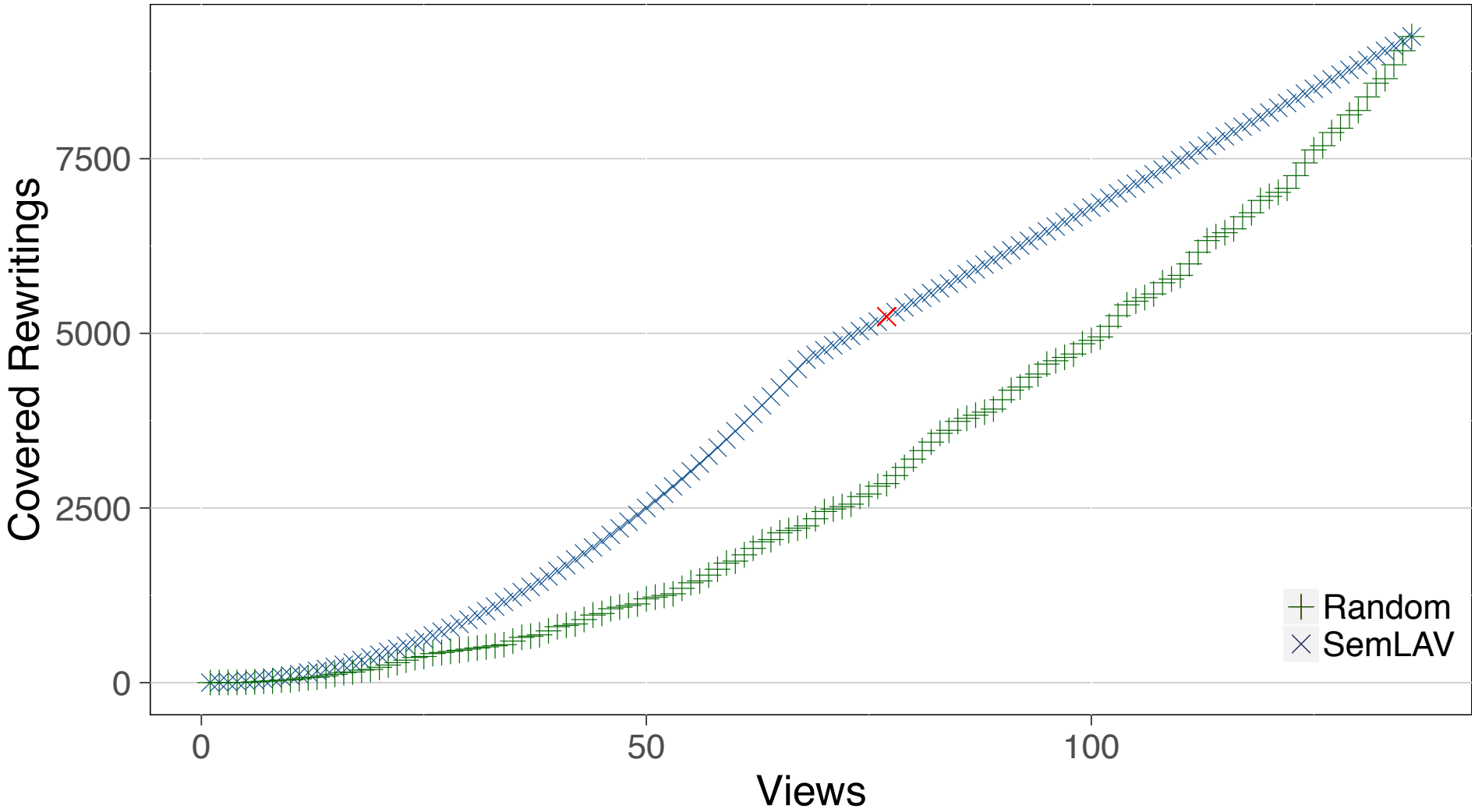
View loading order matters - Q9



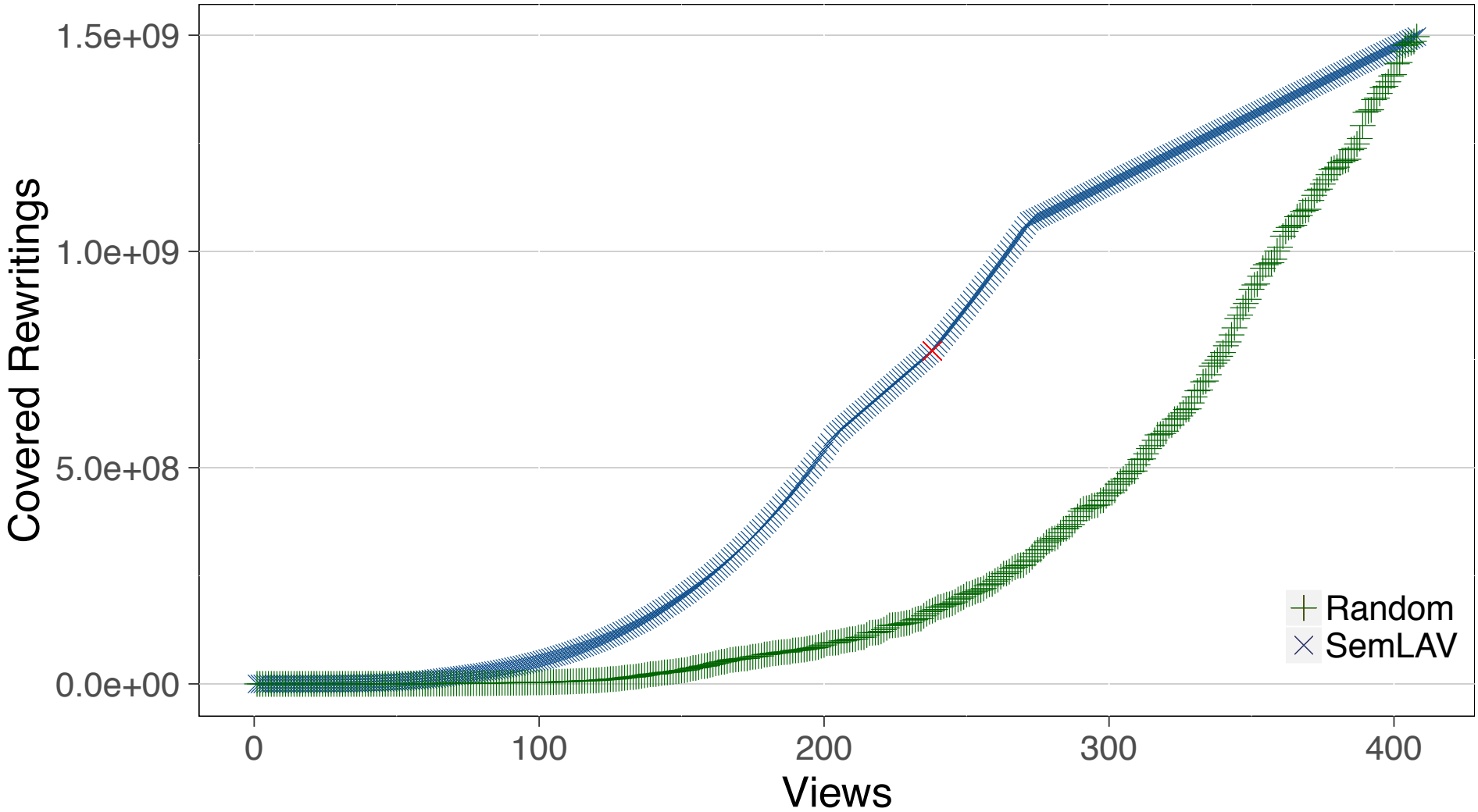
View loading order matters - Q10



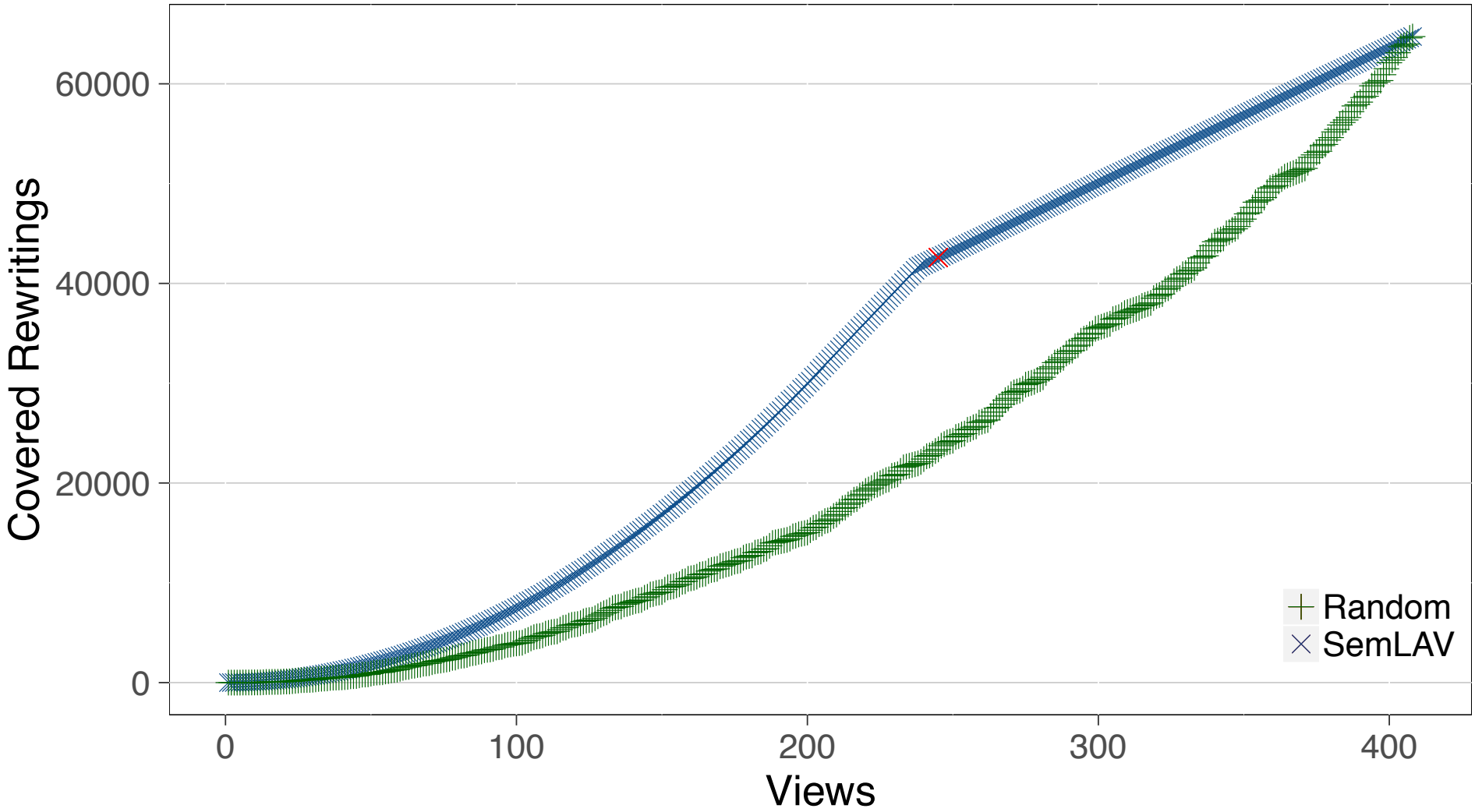
View loading order matters - Q11



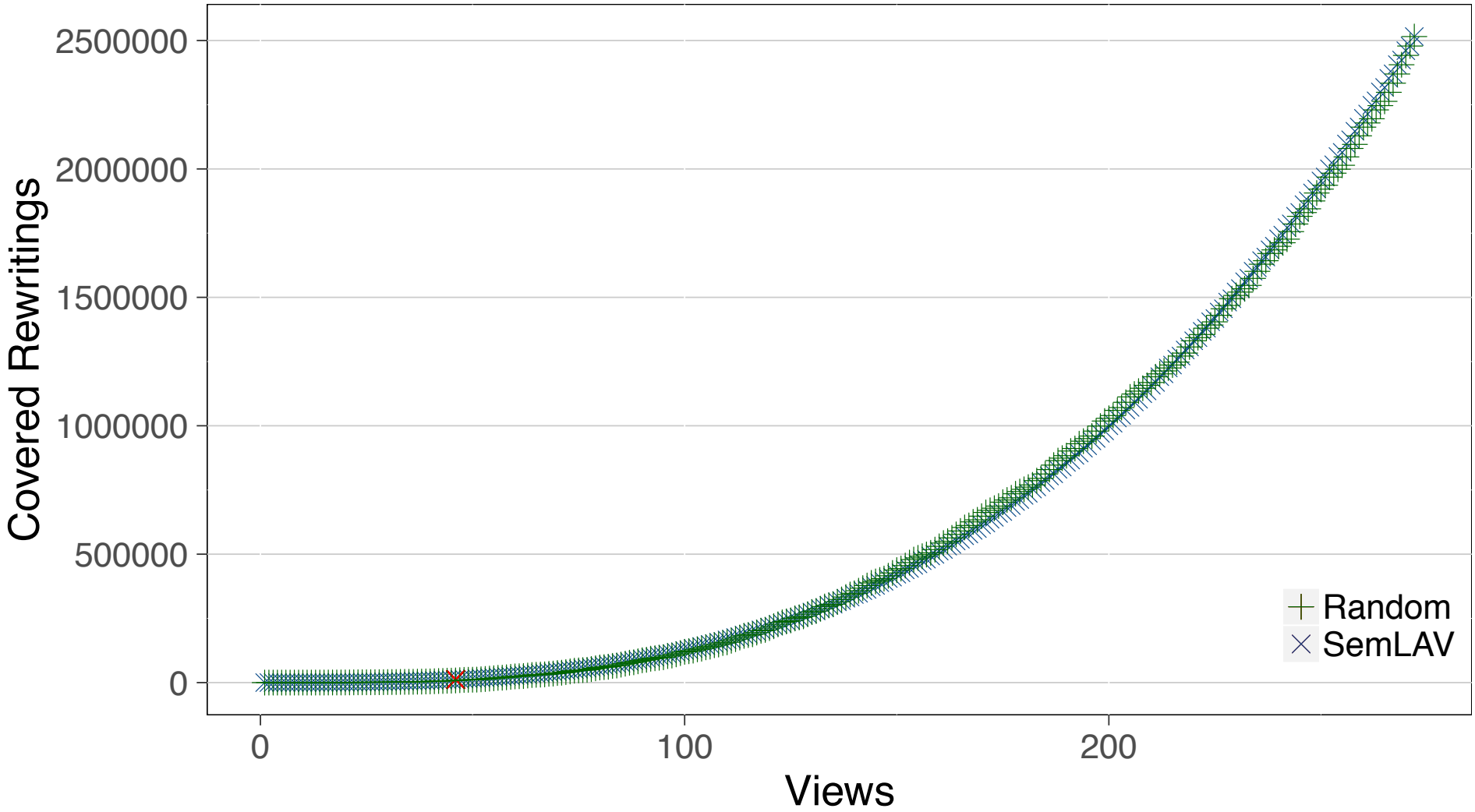
View loading order matters - Q12



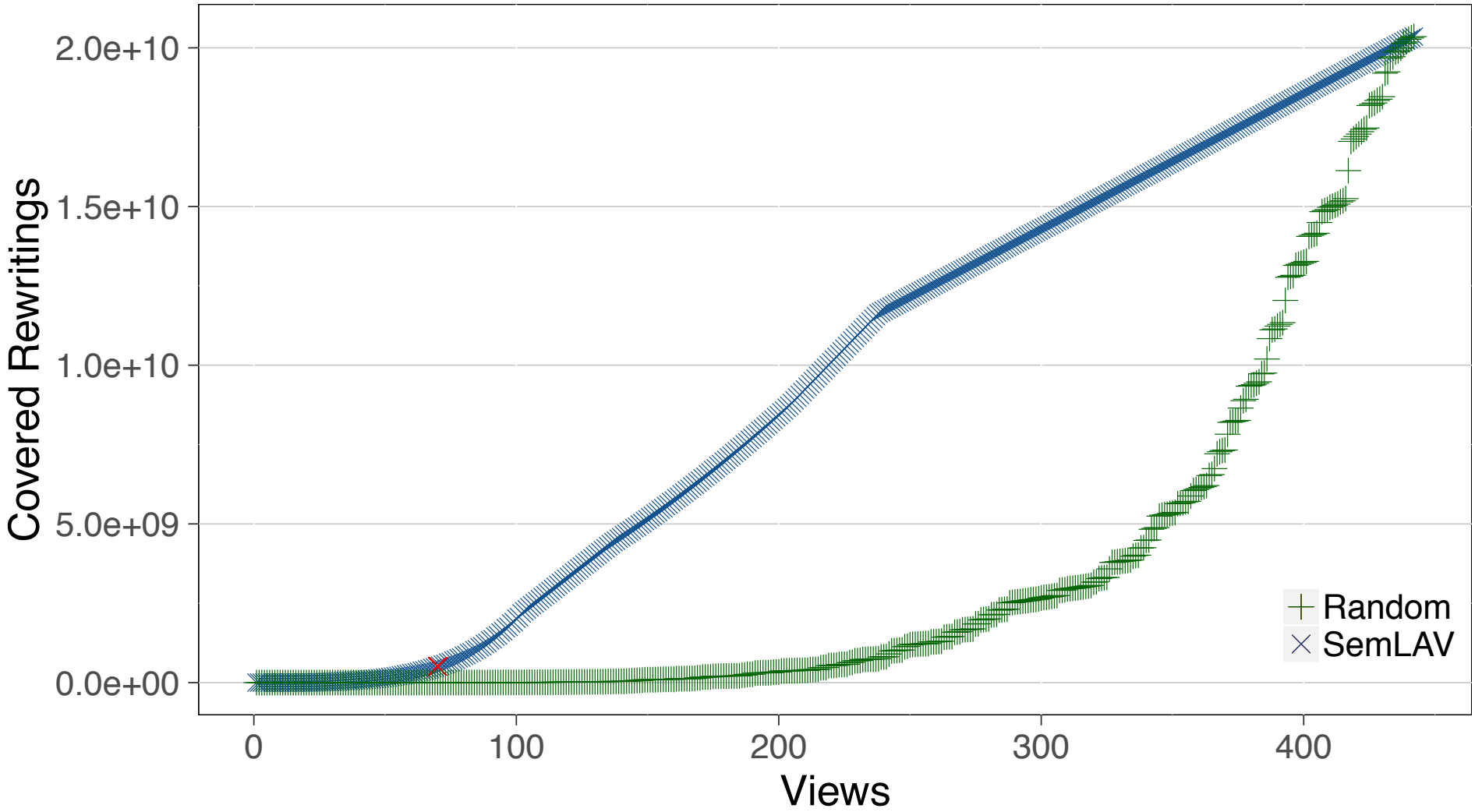
View loading order matters - Q13



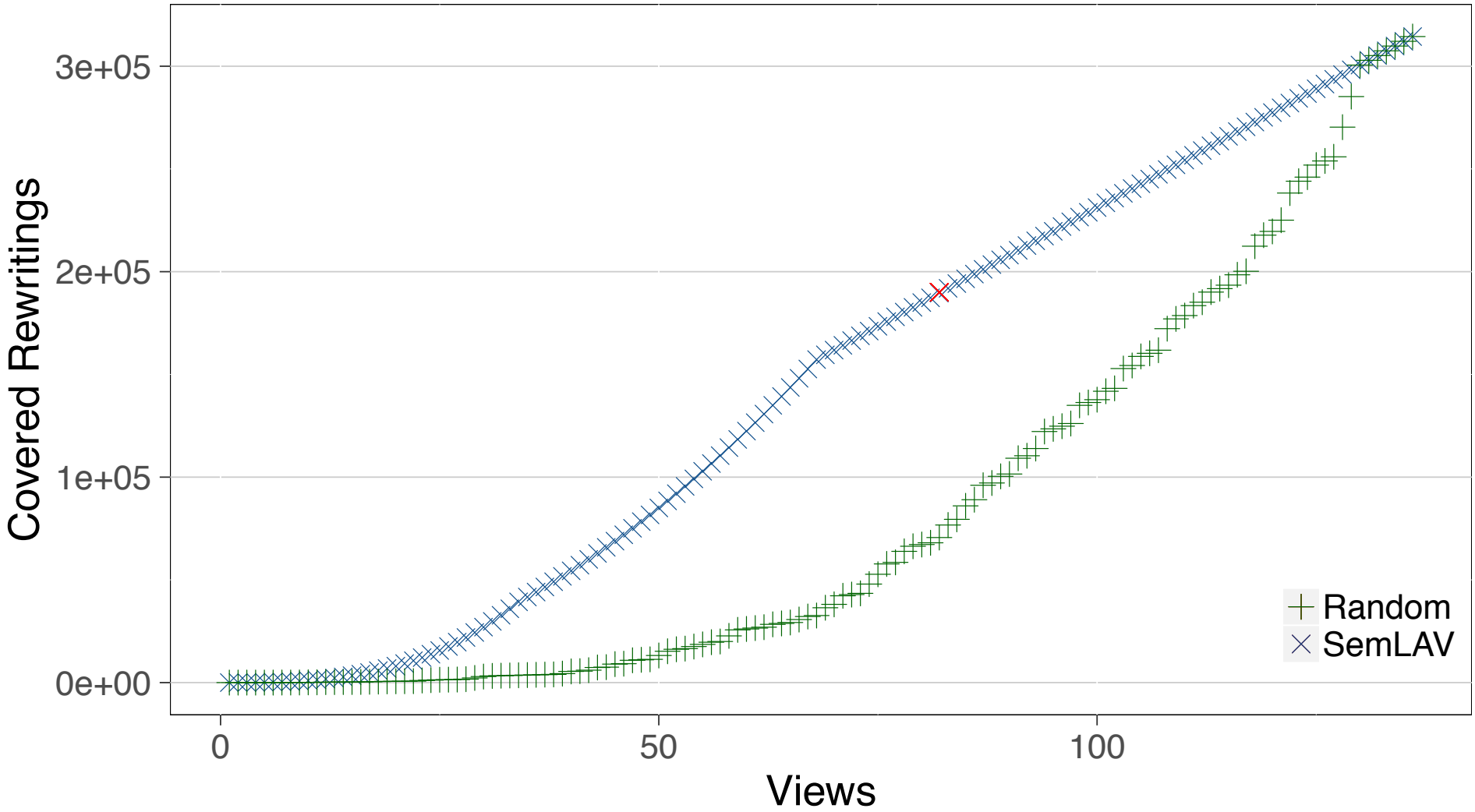
View loading order matters - Q14



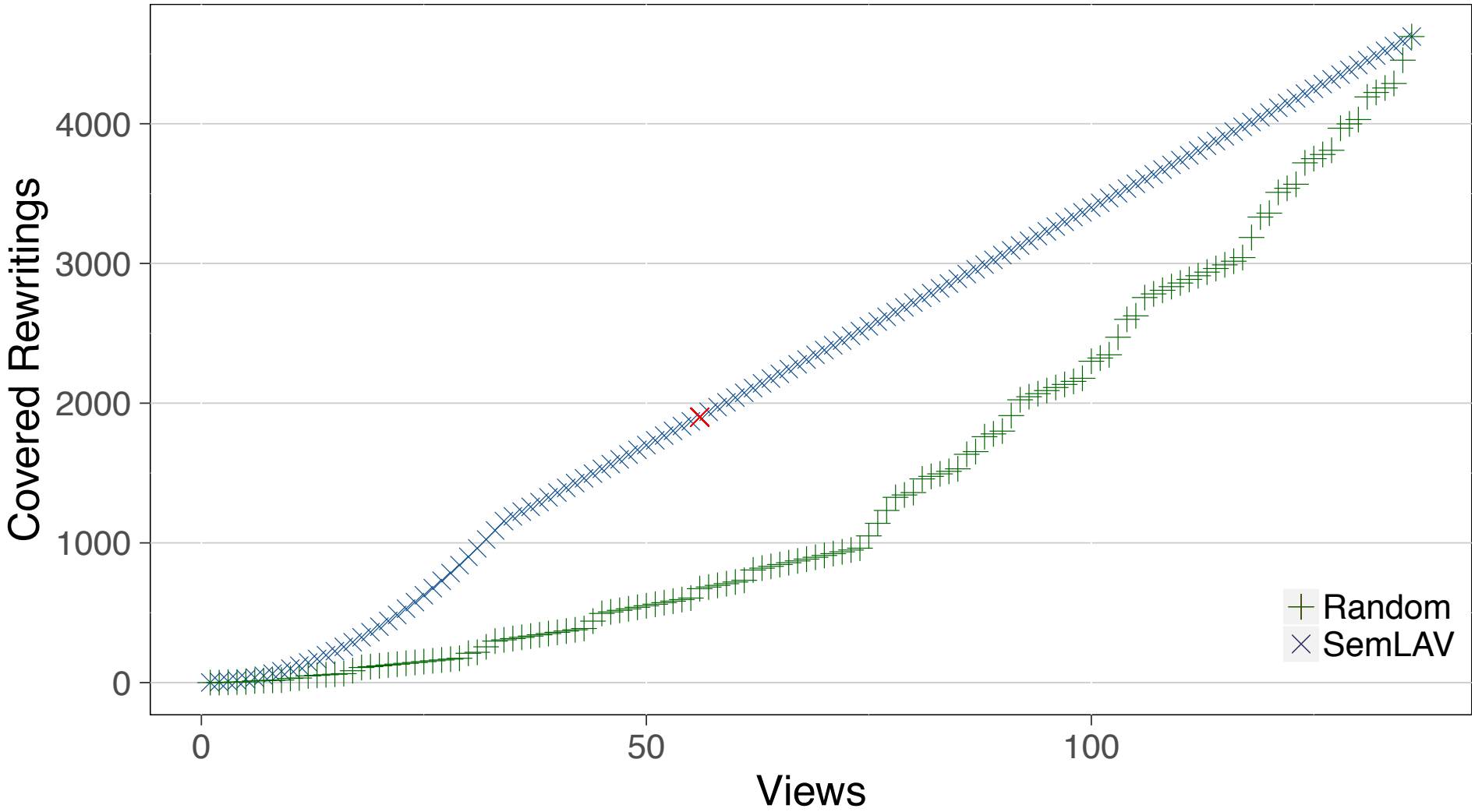
View loading order matters - Q15



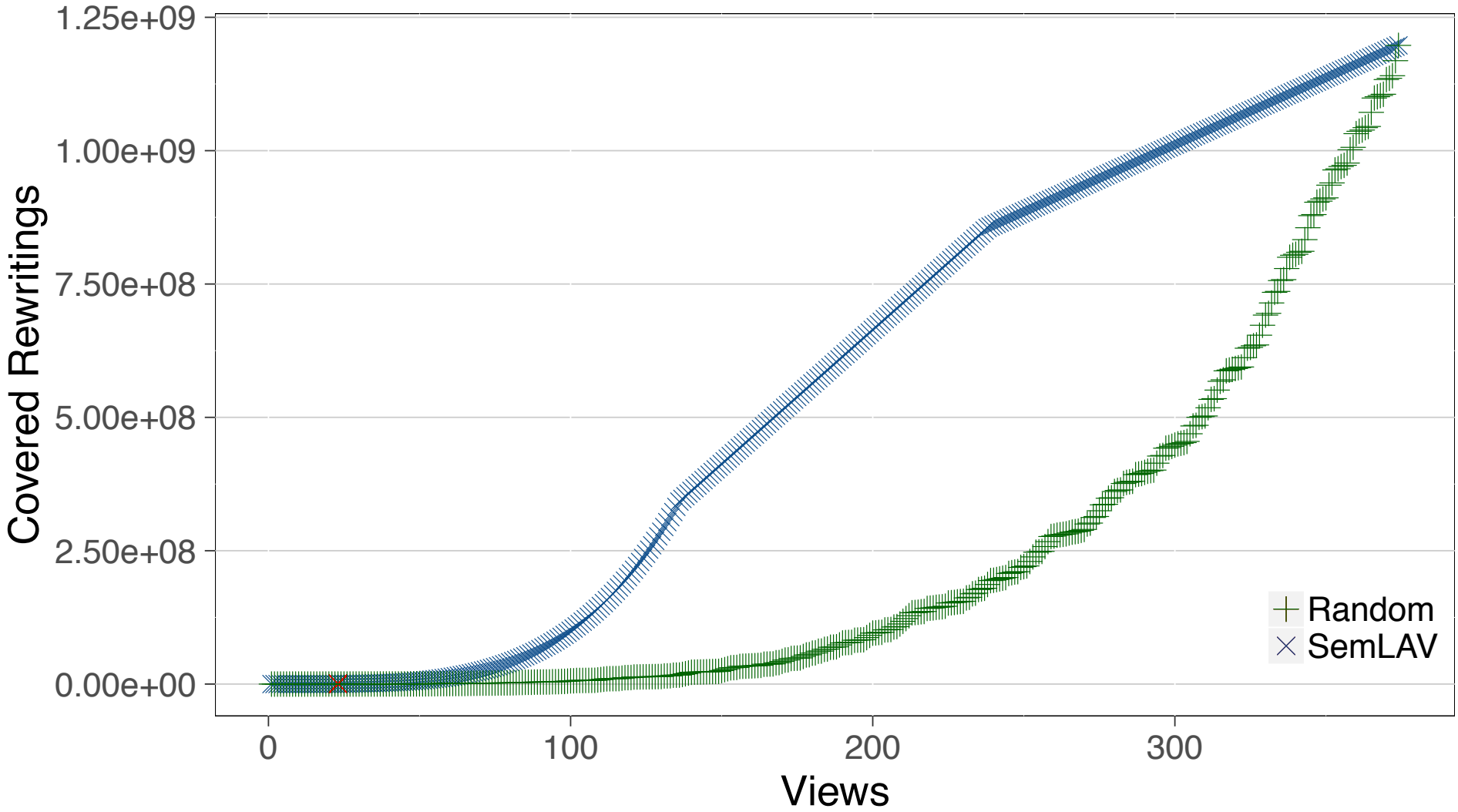
View loading order matters - Q16



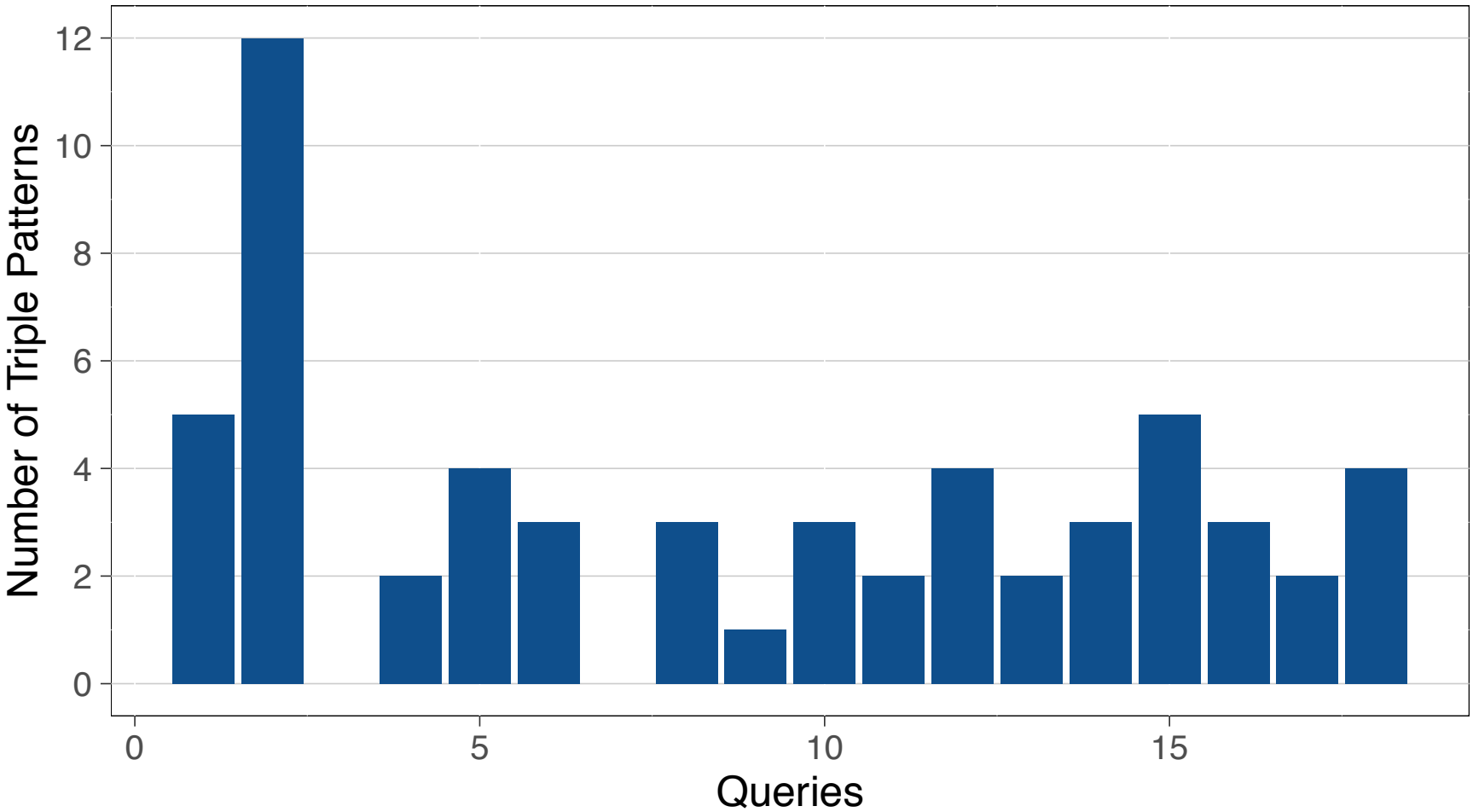
View loading order matters - Q17



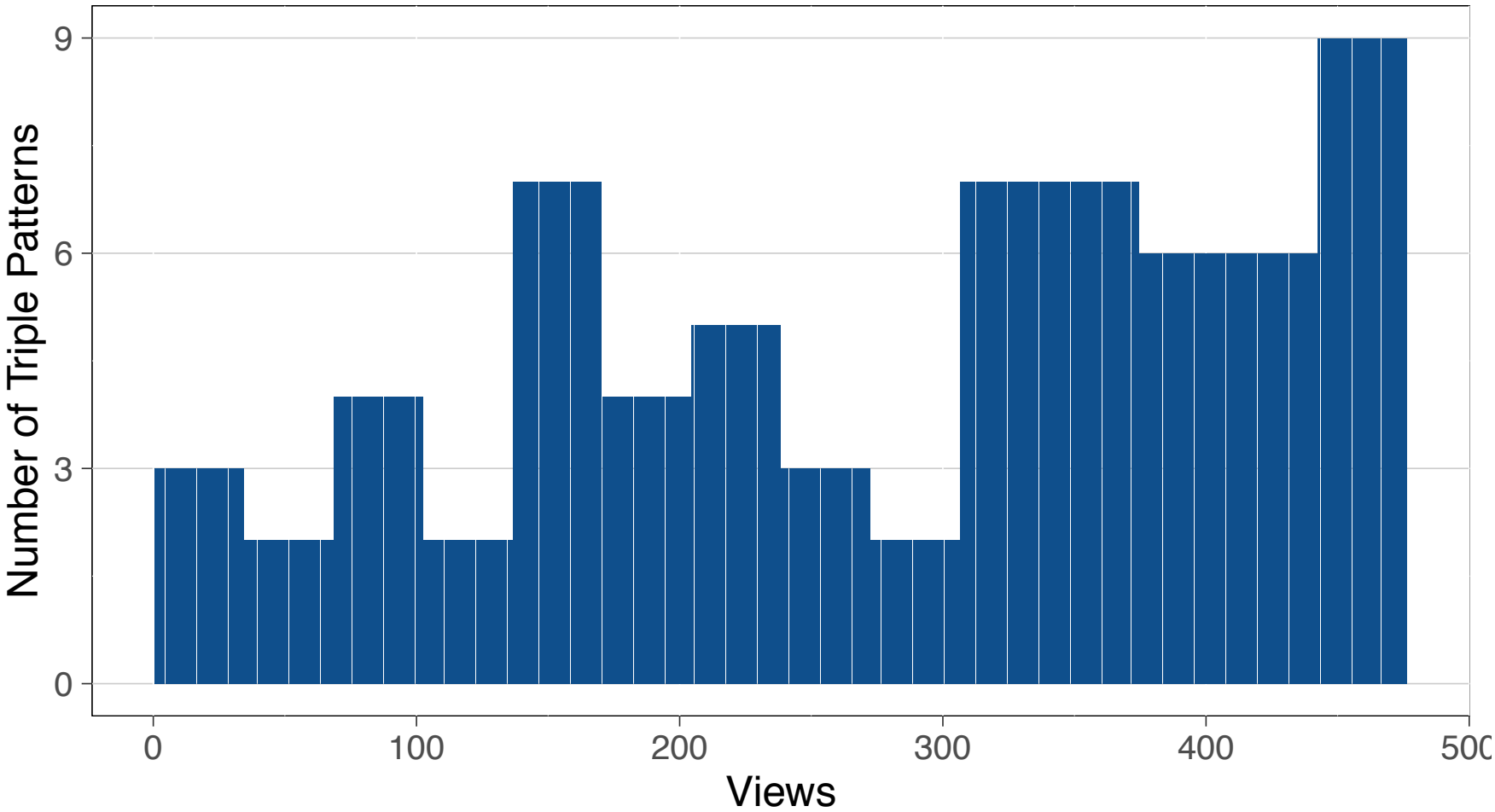
View loading order matters - Q18



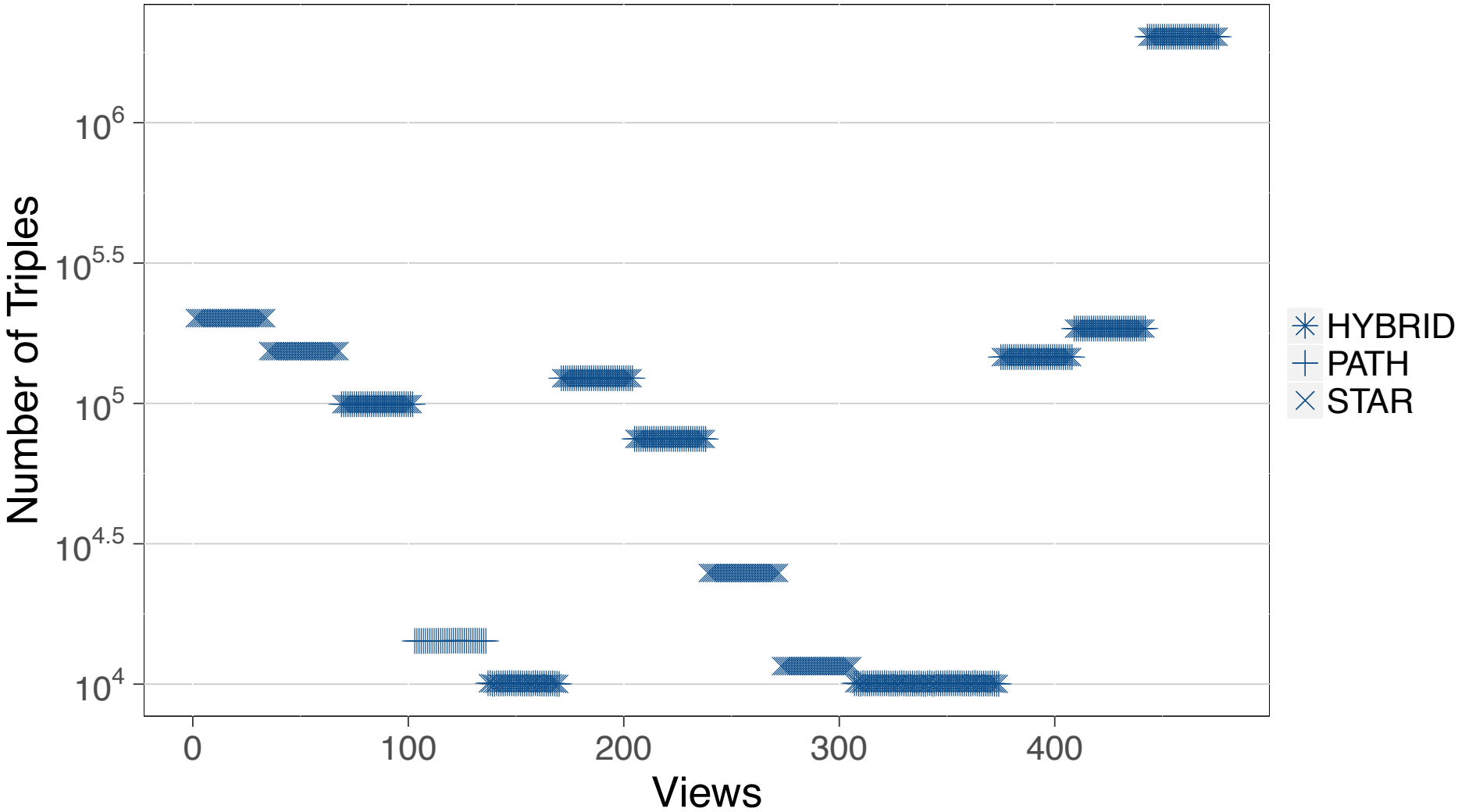
Number of Triple Patterns in Queries



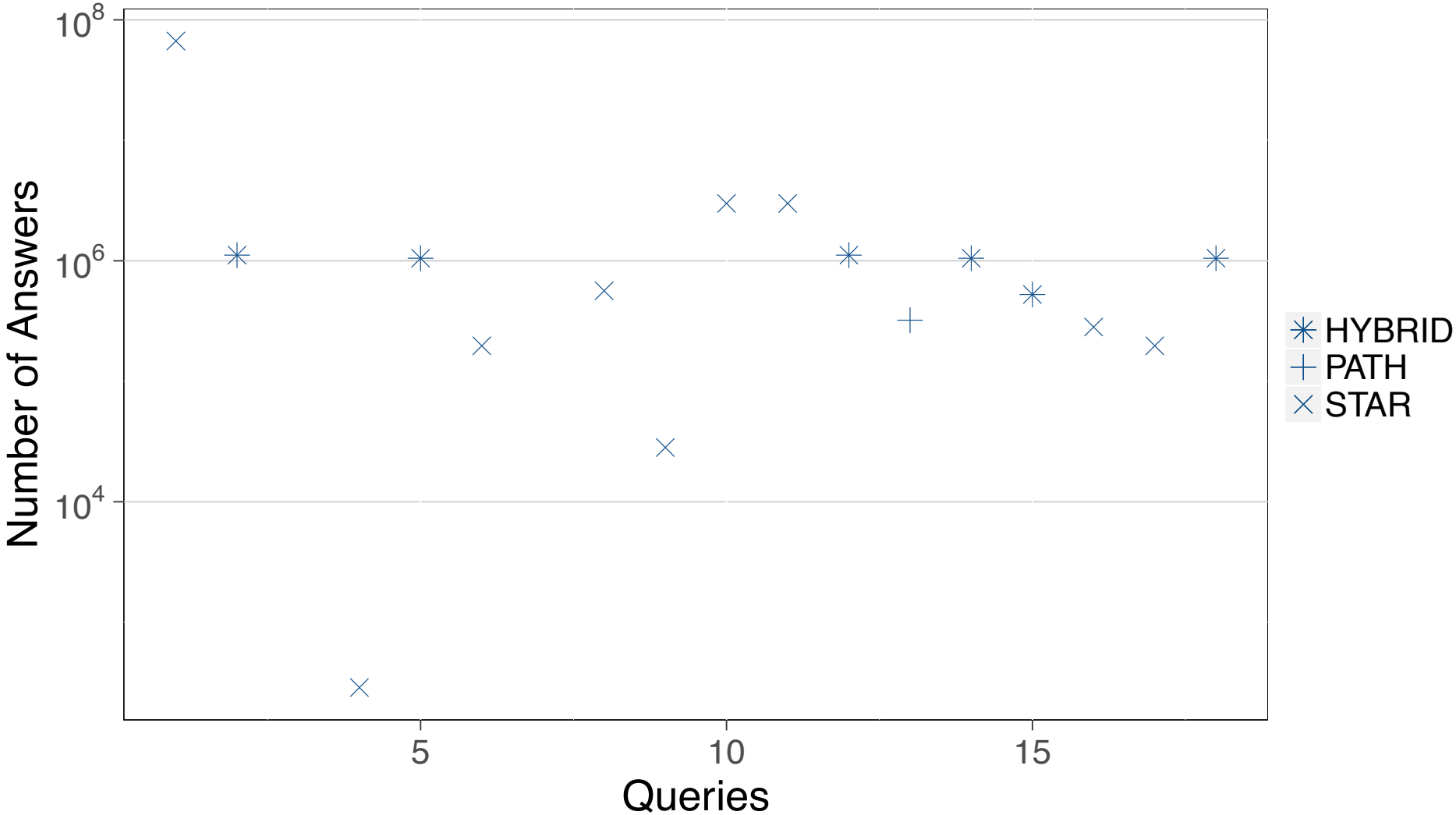
Number of Triple Patterns in Views



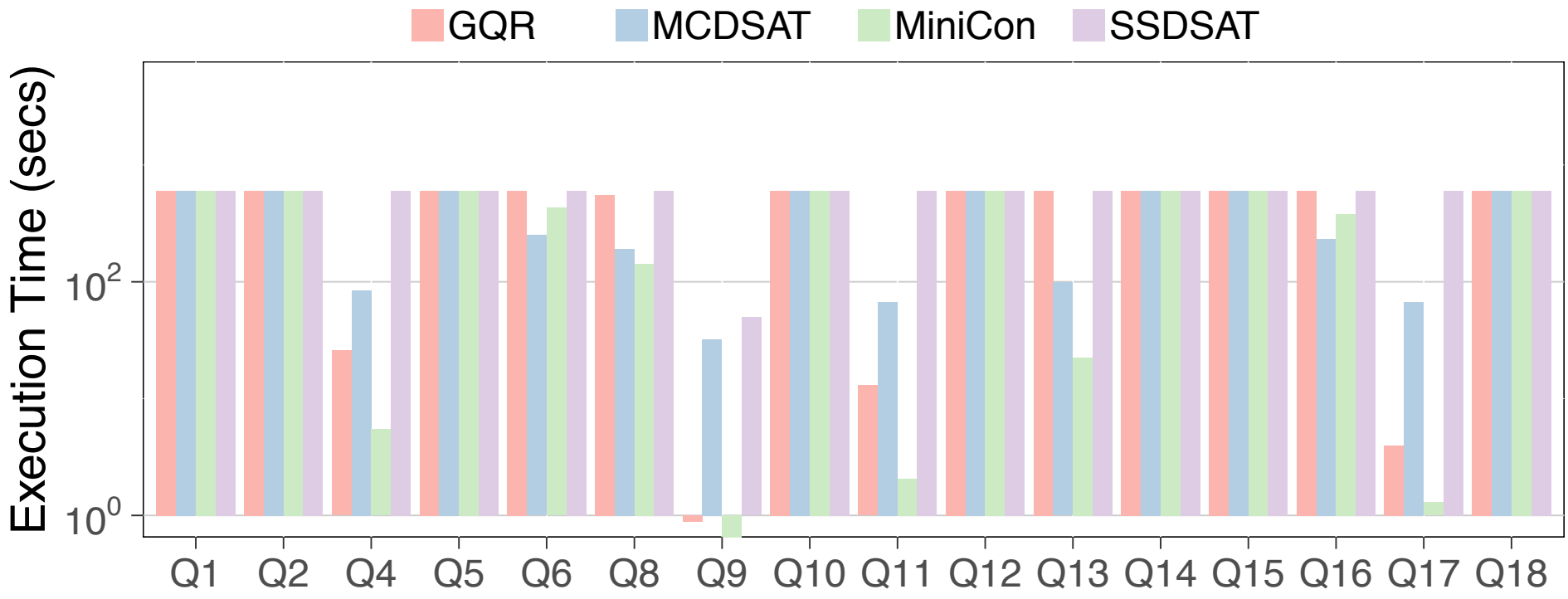
SemLAV views' number of triples



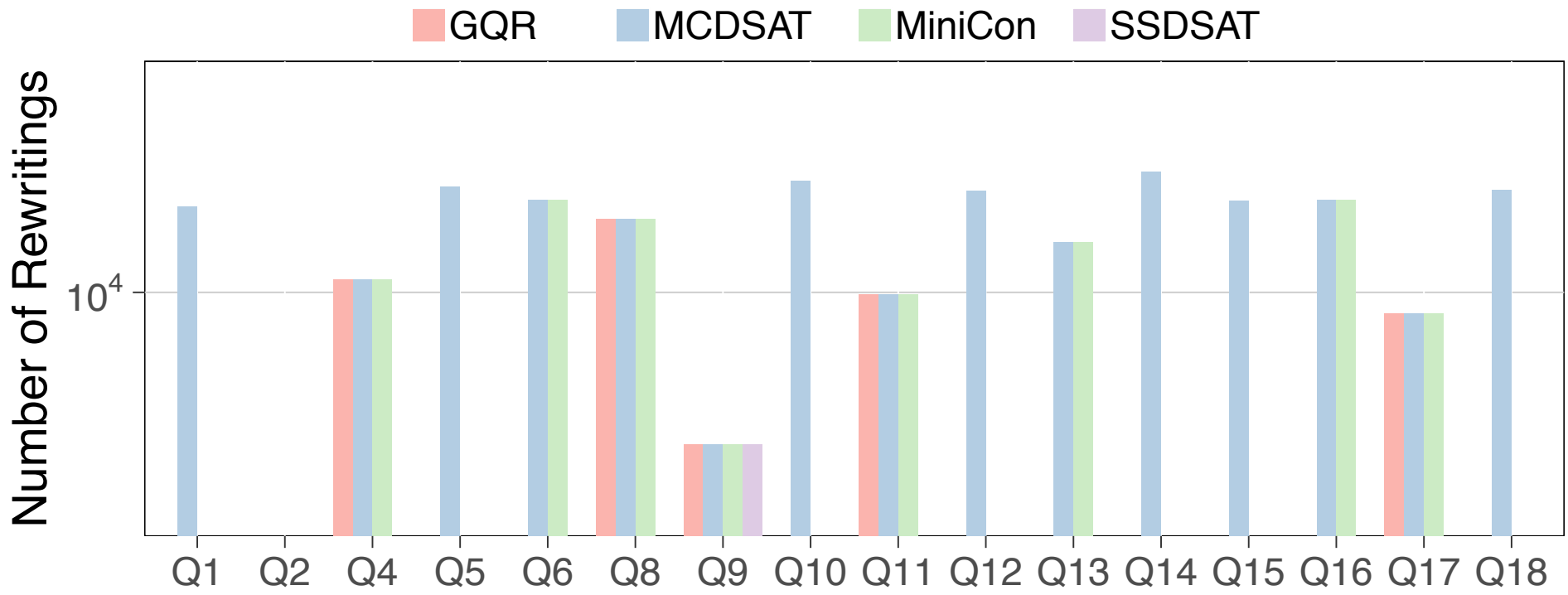
SemLAV queries' number of answers and shape



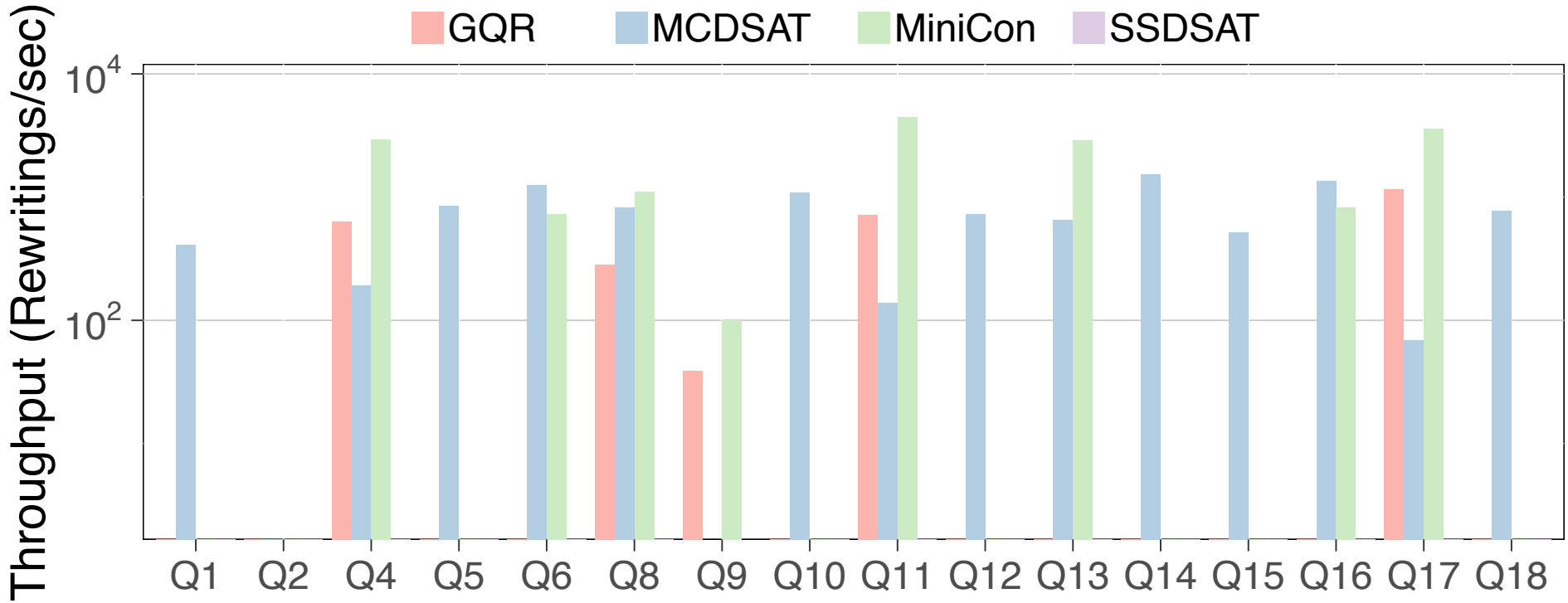
Execution time of the studied rewriters



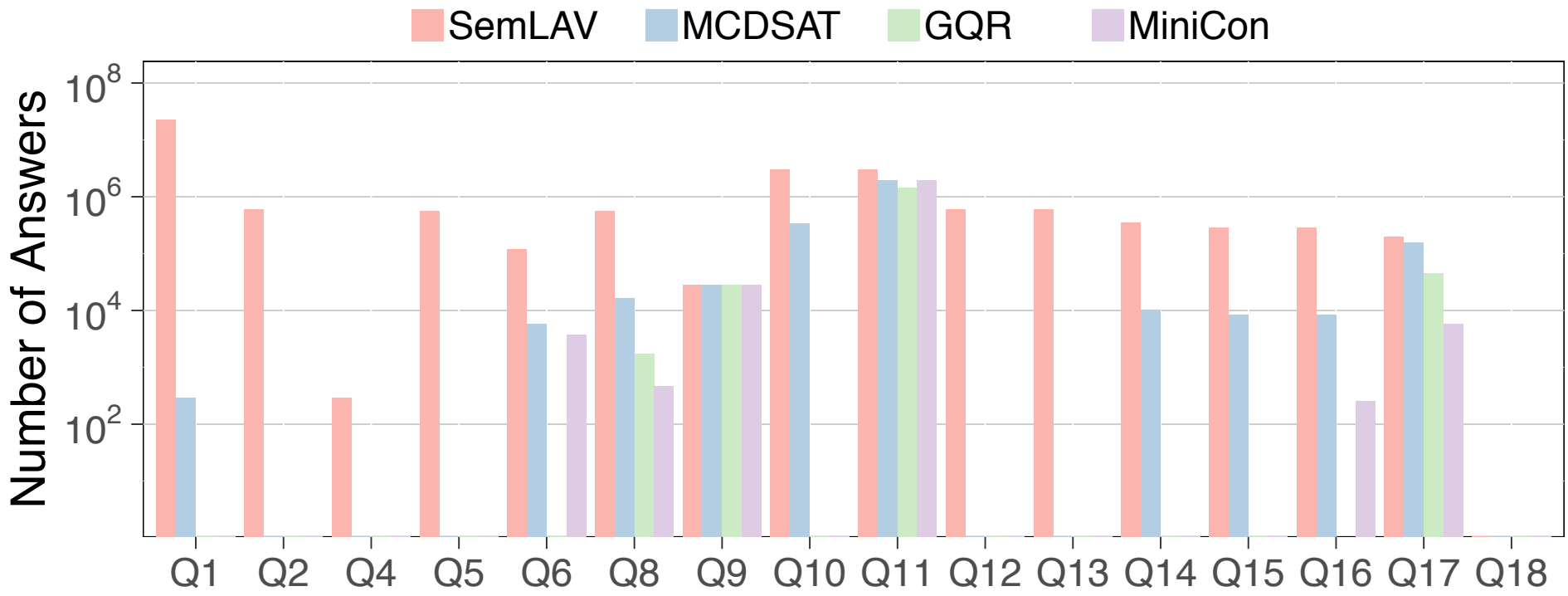
Number of rewritings produced by the studied rewriters



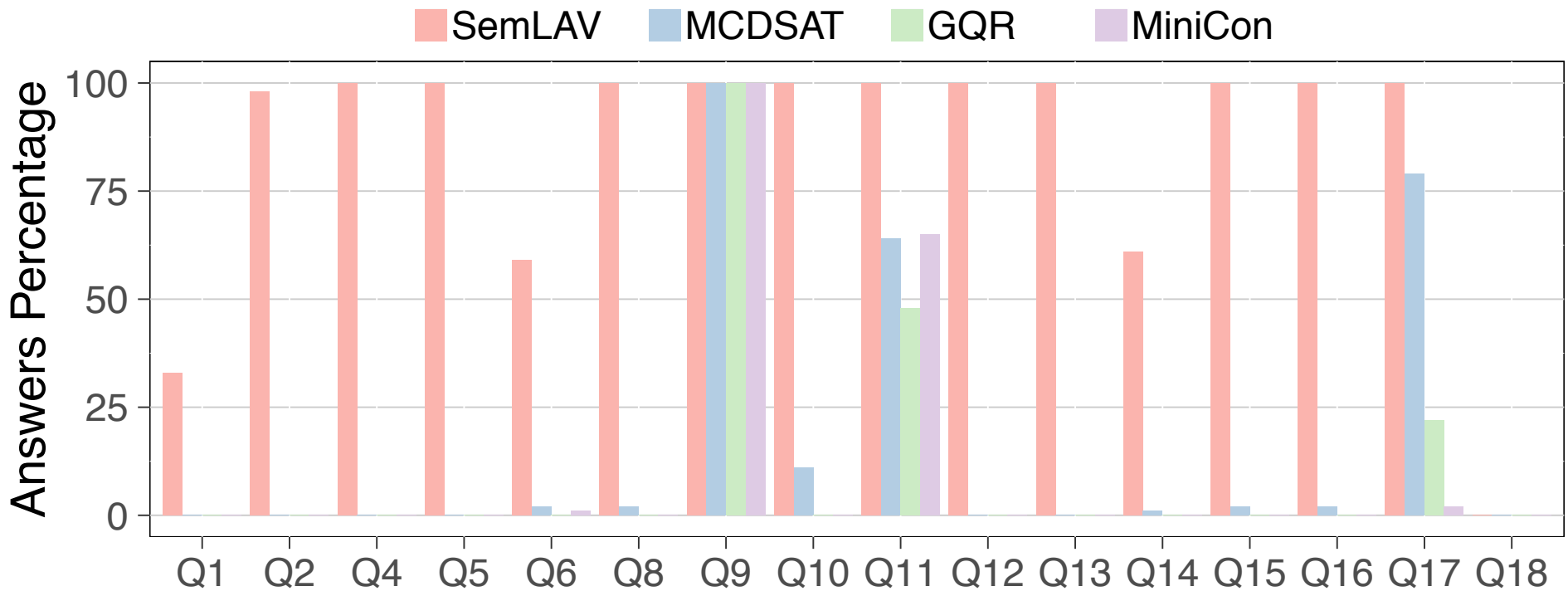
Throughput of the studied rewriters



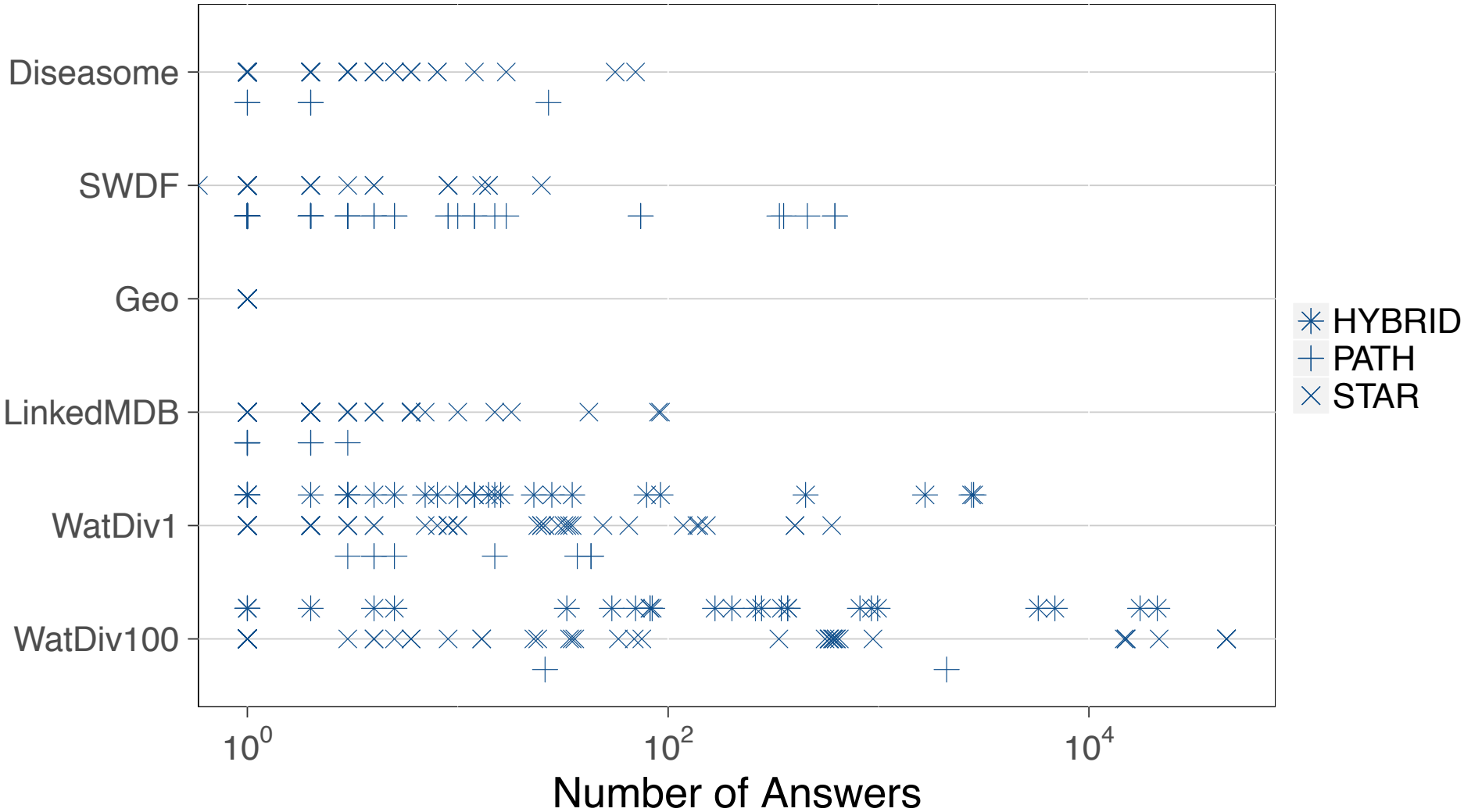
SemLAV produces more answers than the rewriting-based approaches



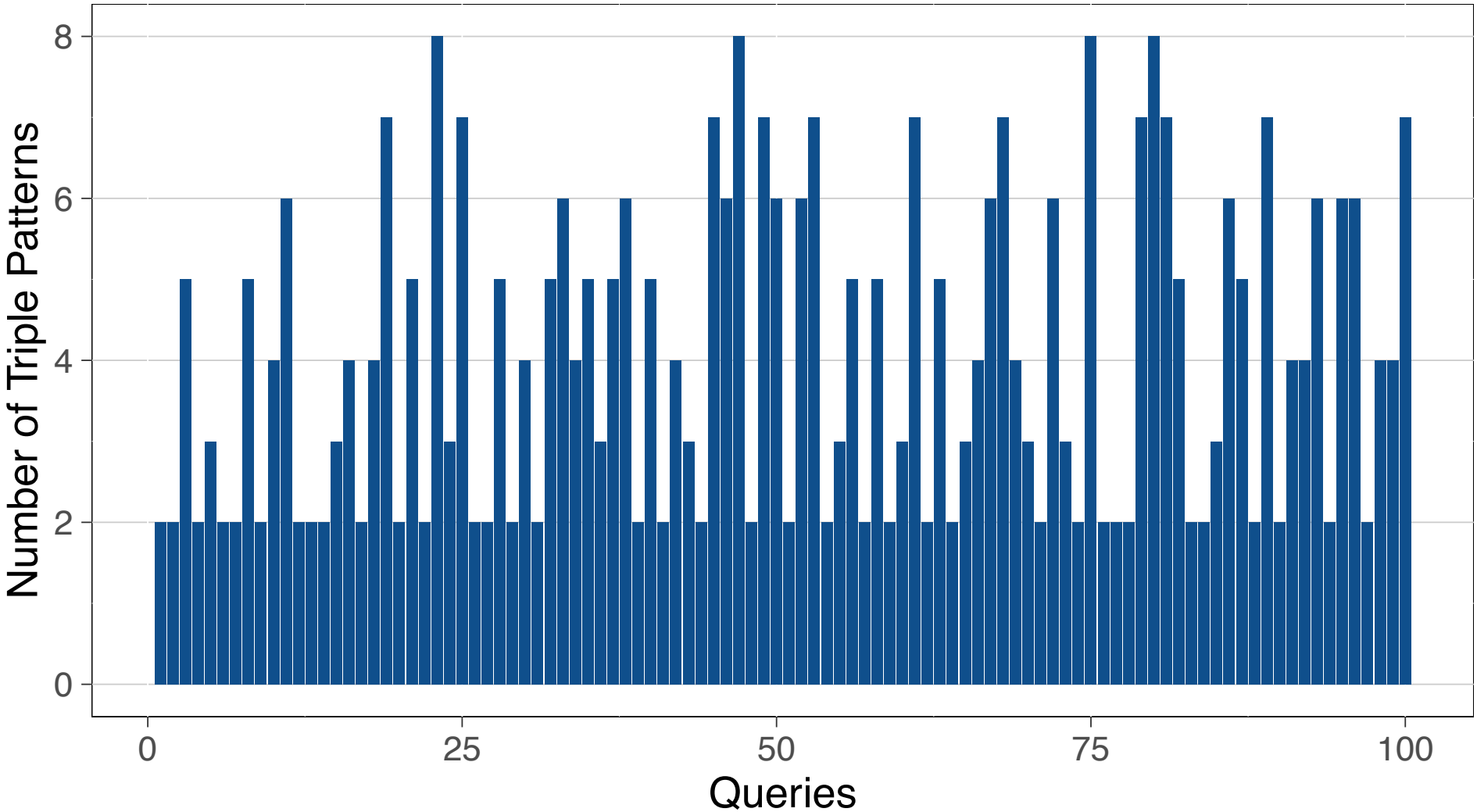
SemLAV produces answers more complete than the rewriting-based approaches



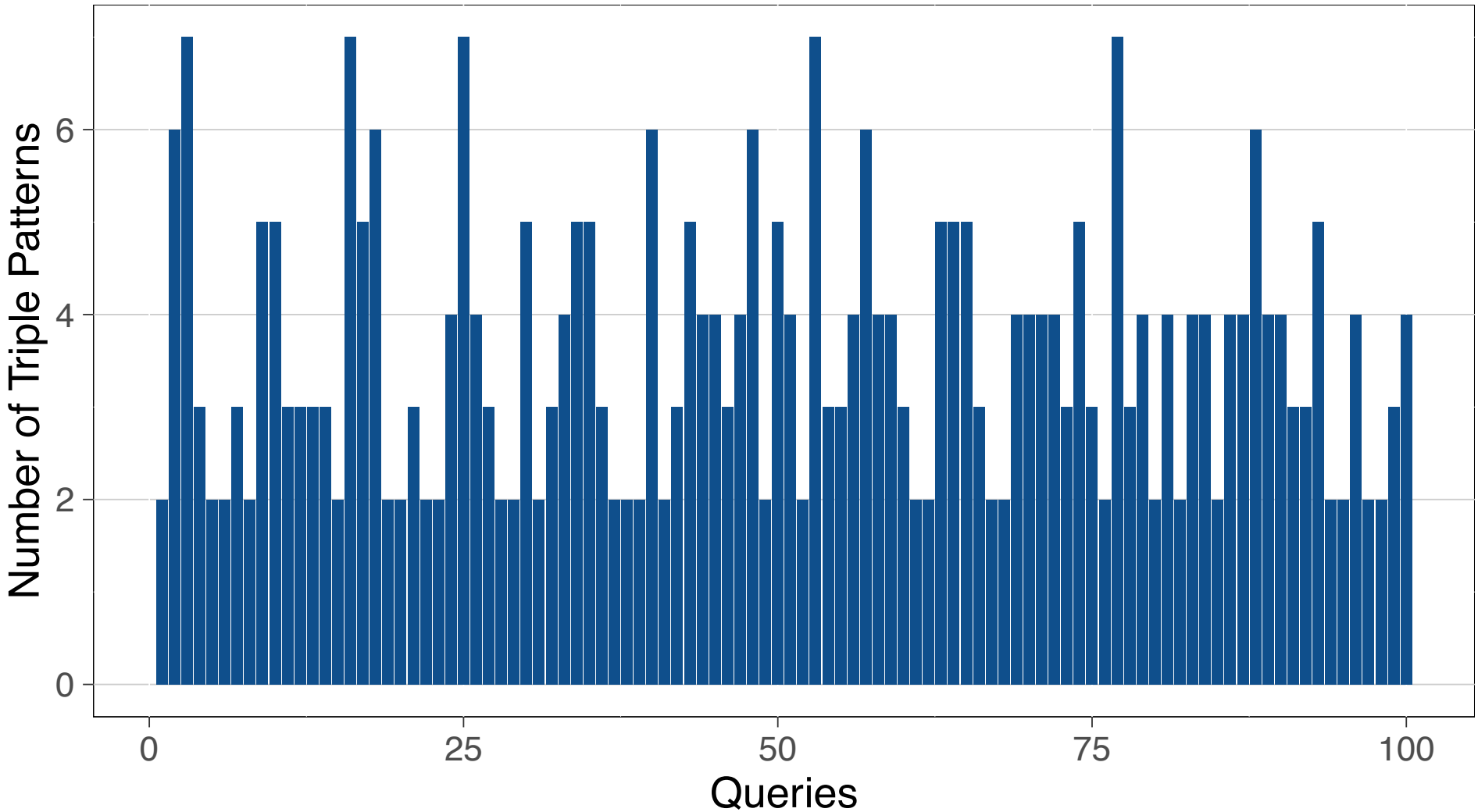
FEDRA queries' number of answers and shape



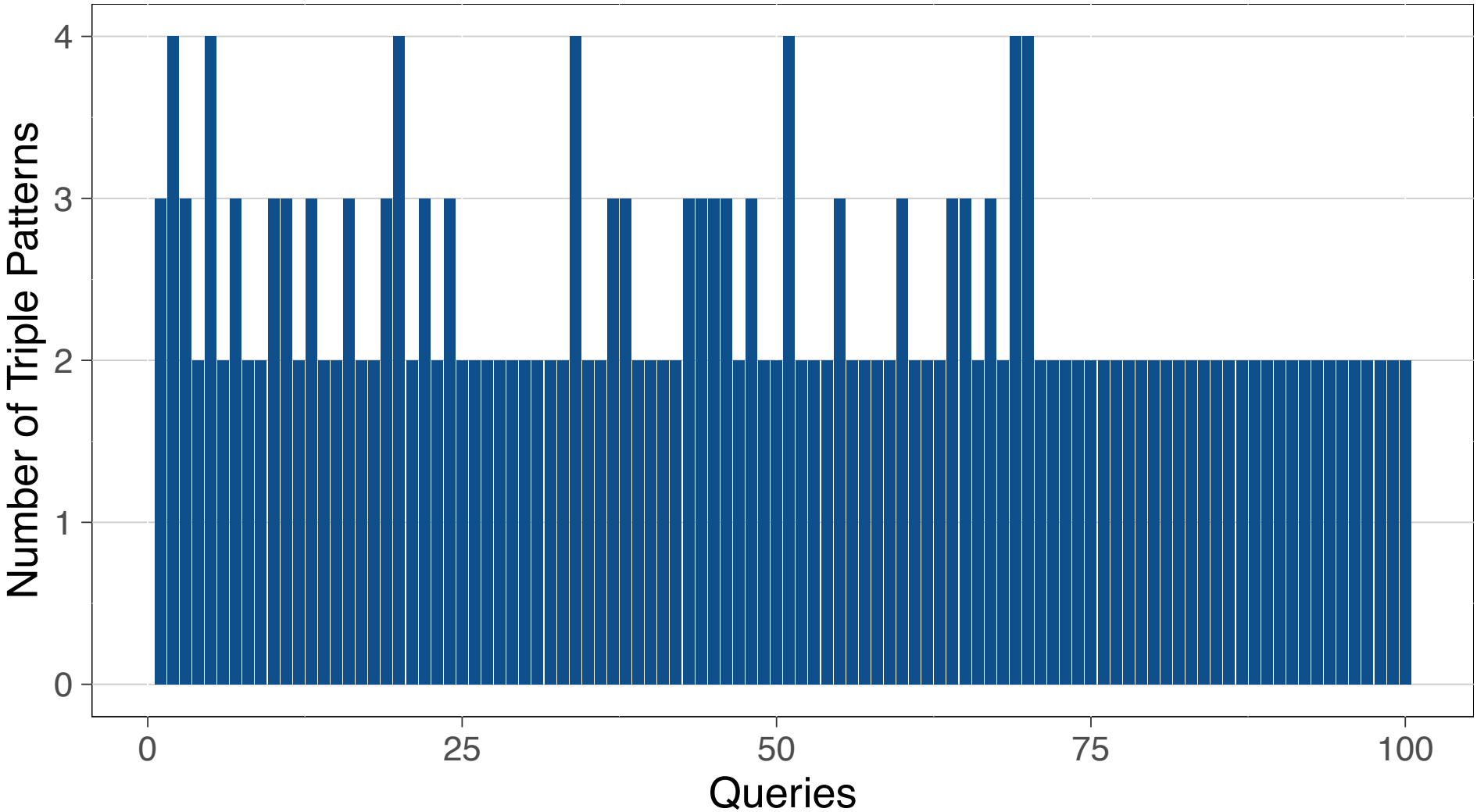
Diseasome - query number of triple patterns



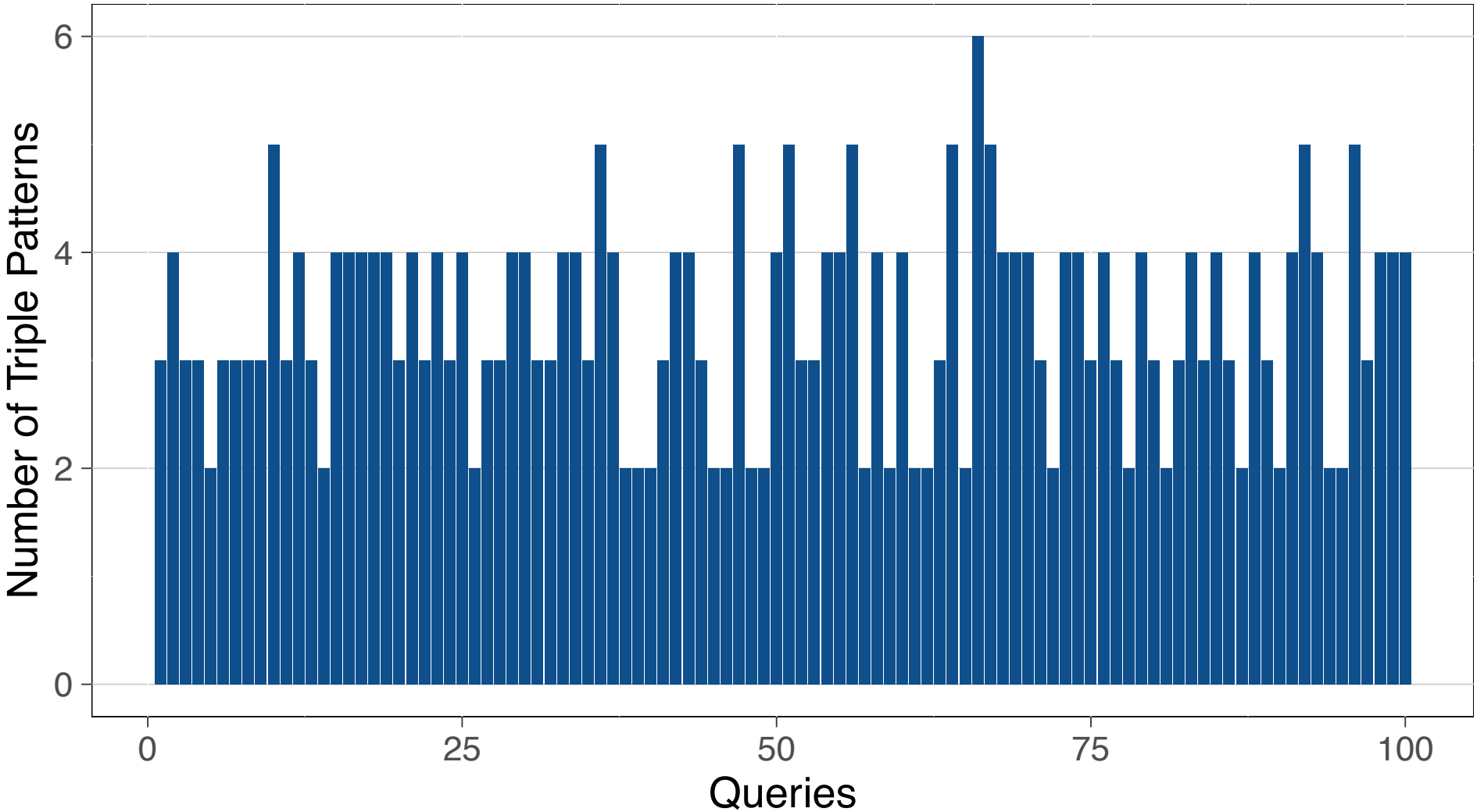
SWDF - query number of triple patterns



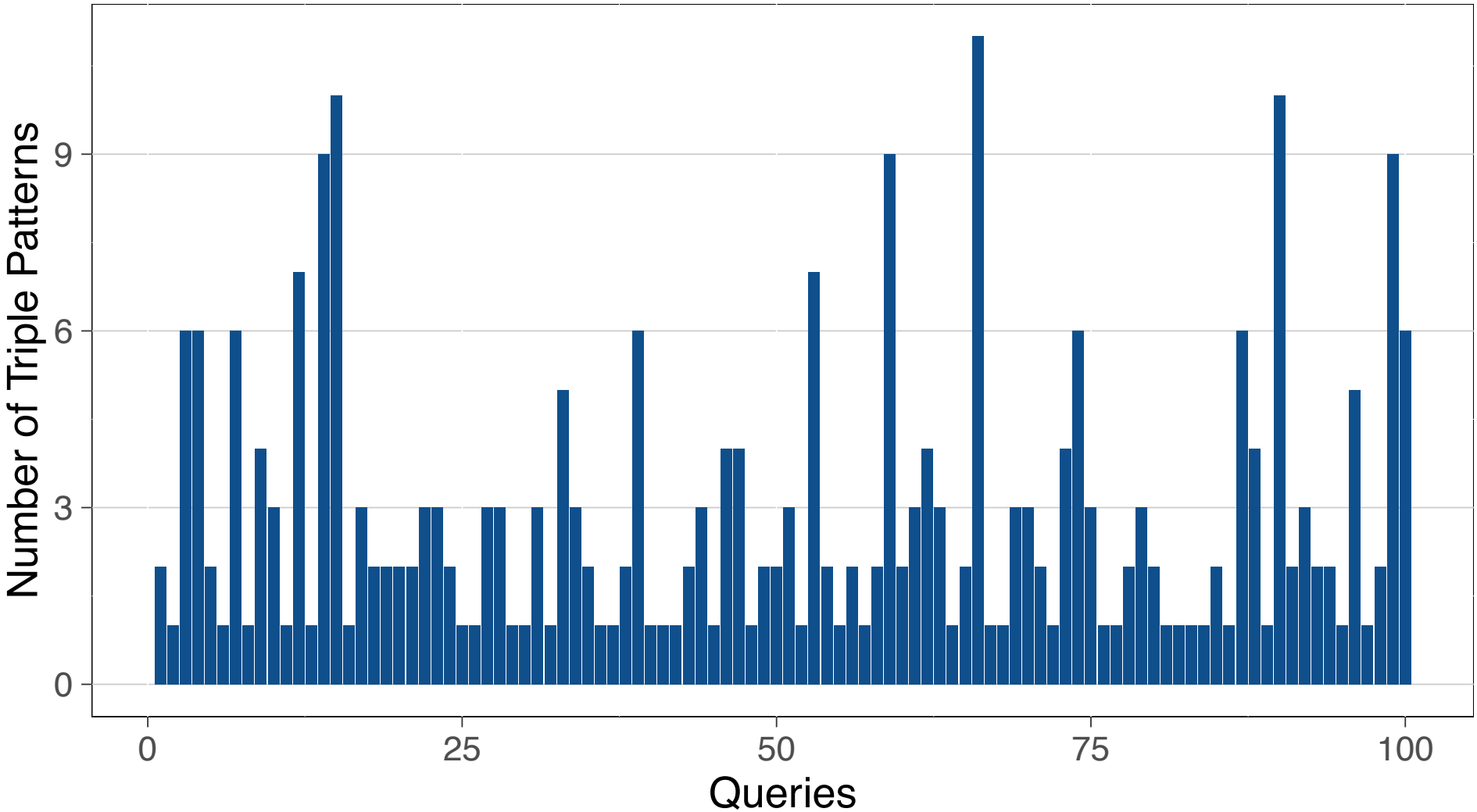
Geocoordinates - query number of triple patterns



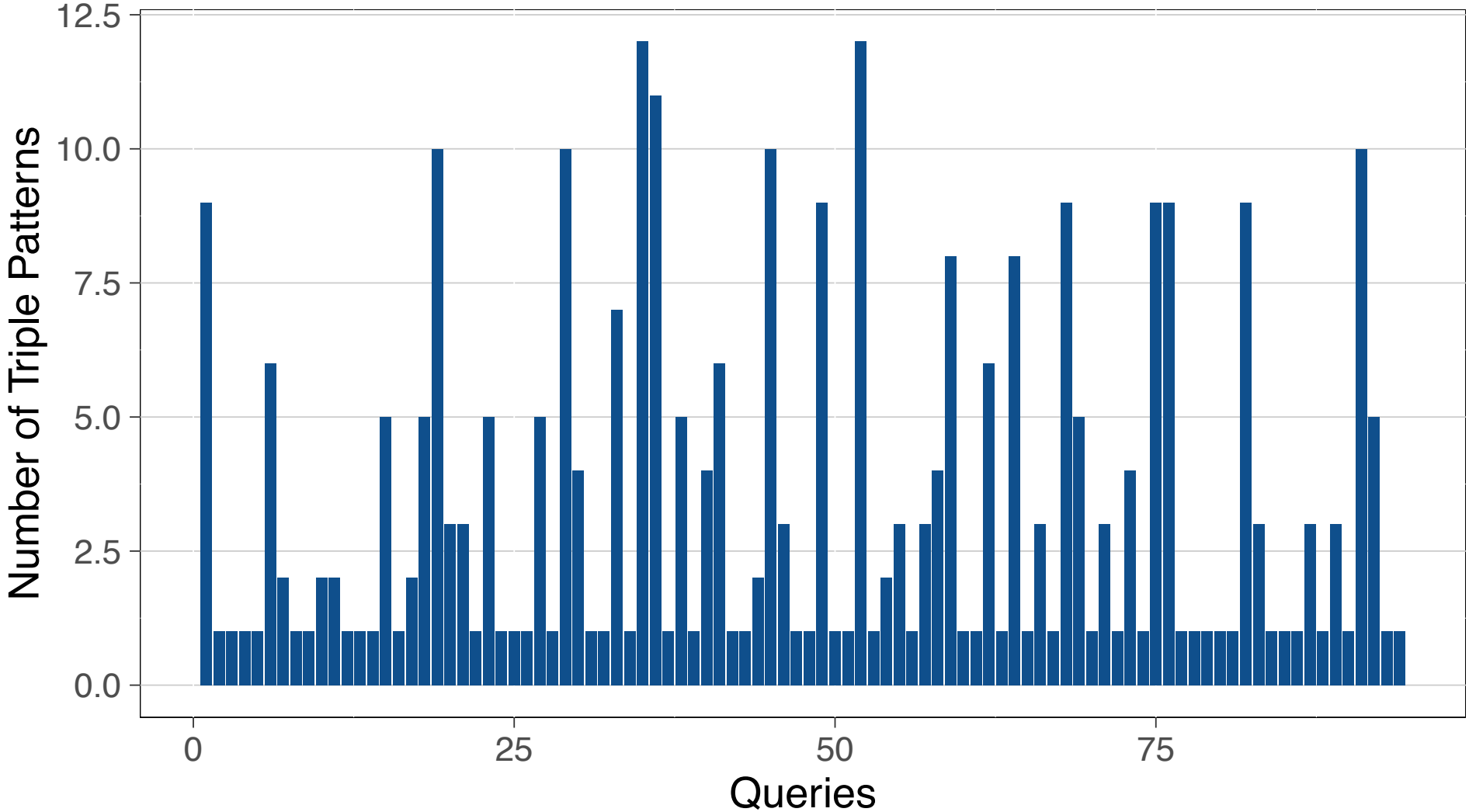
LinkedMDB - query number of triple patterns



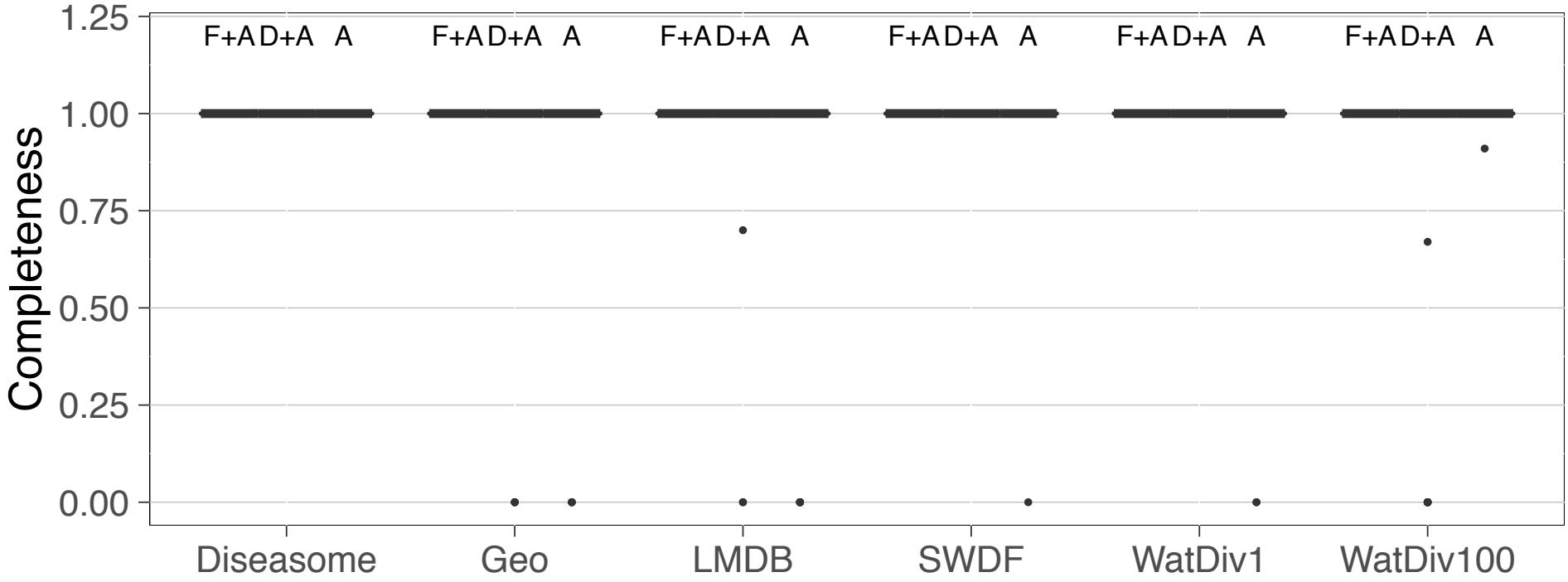
WatDiv1 - query number of triple patterns



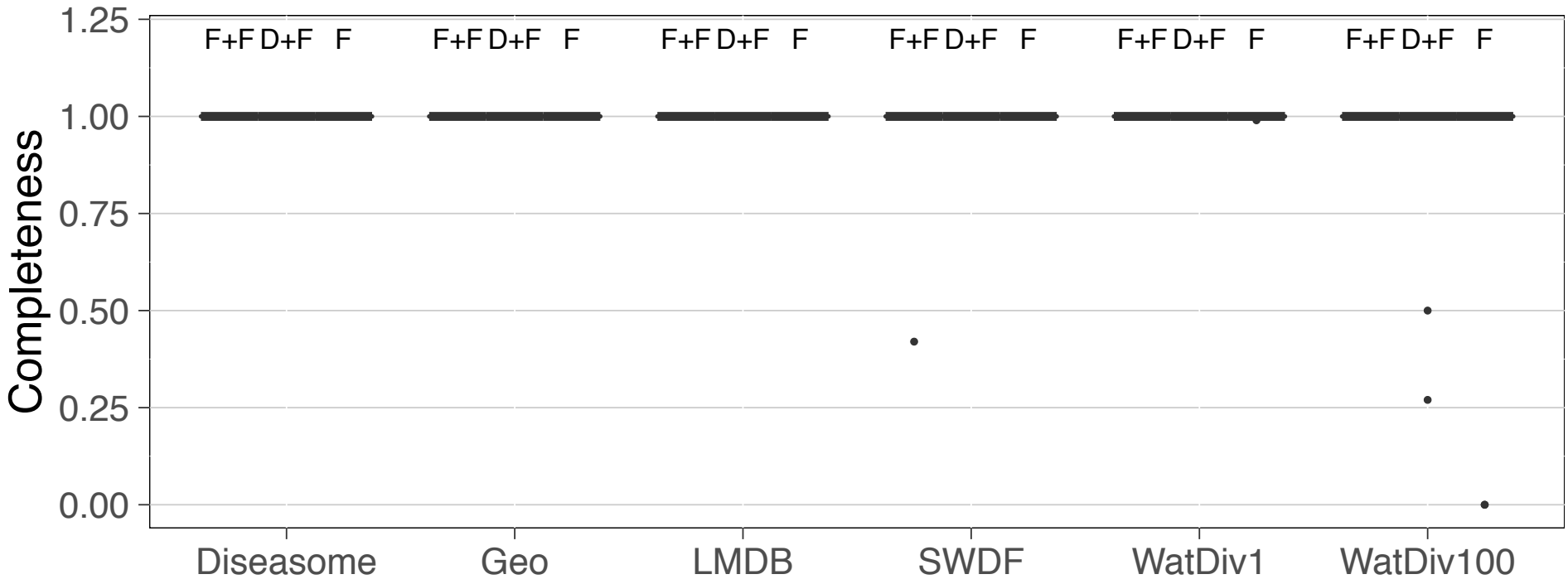
WatDiv100 - query number of triple patterns



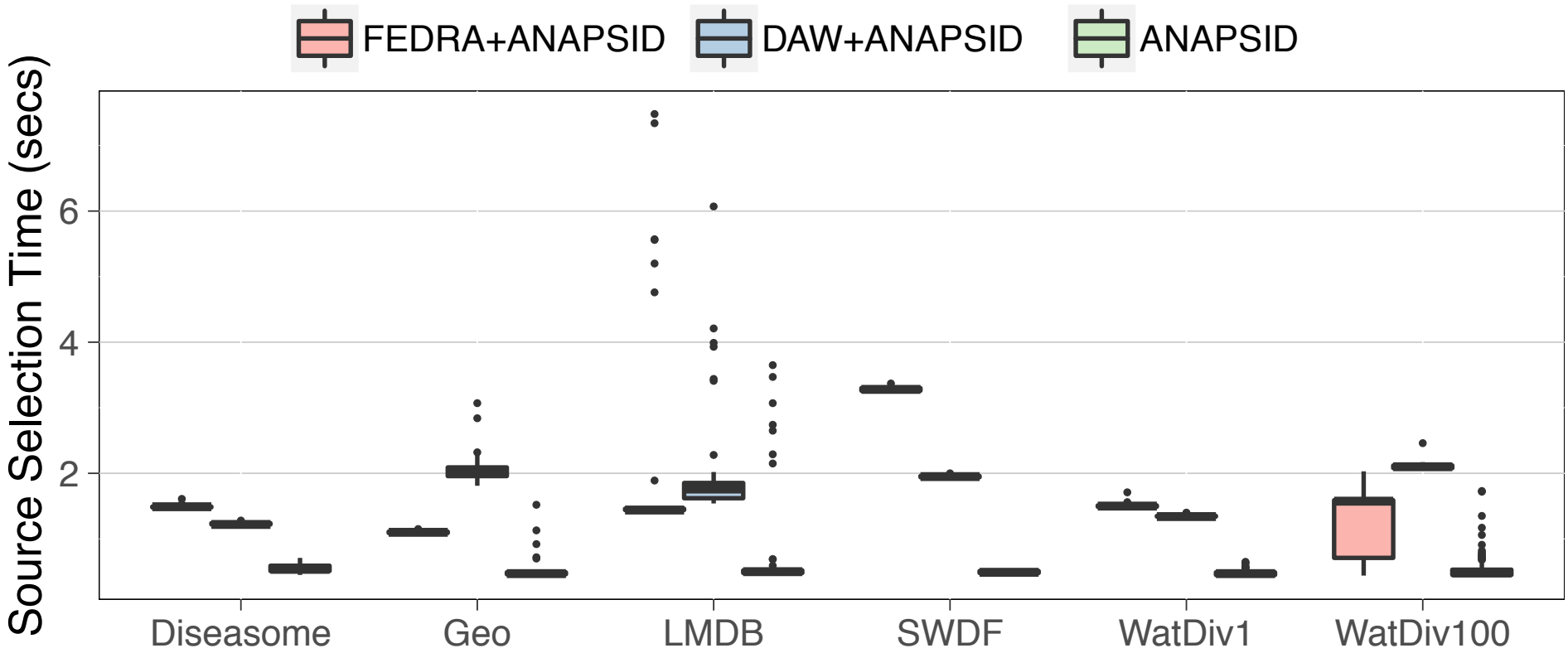
FEDRA increases the number of obtained answers



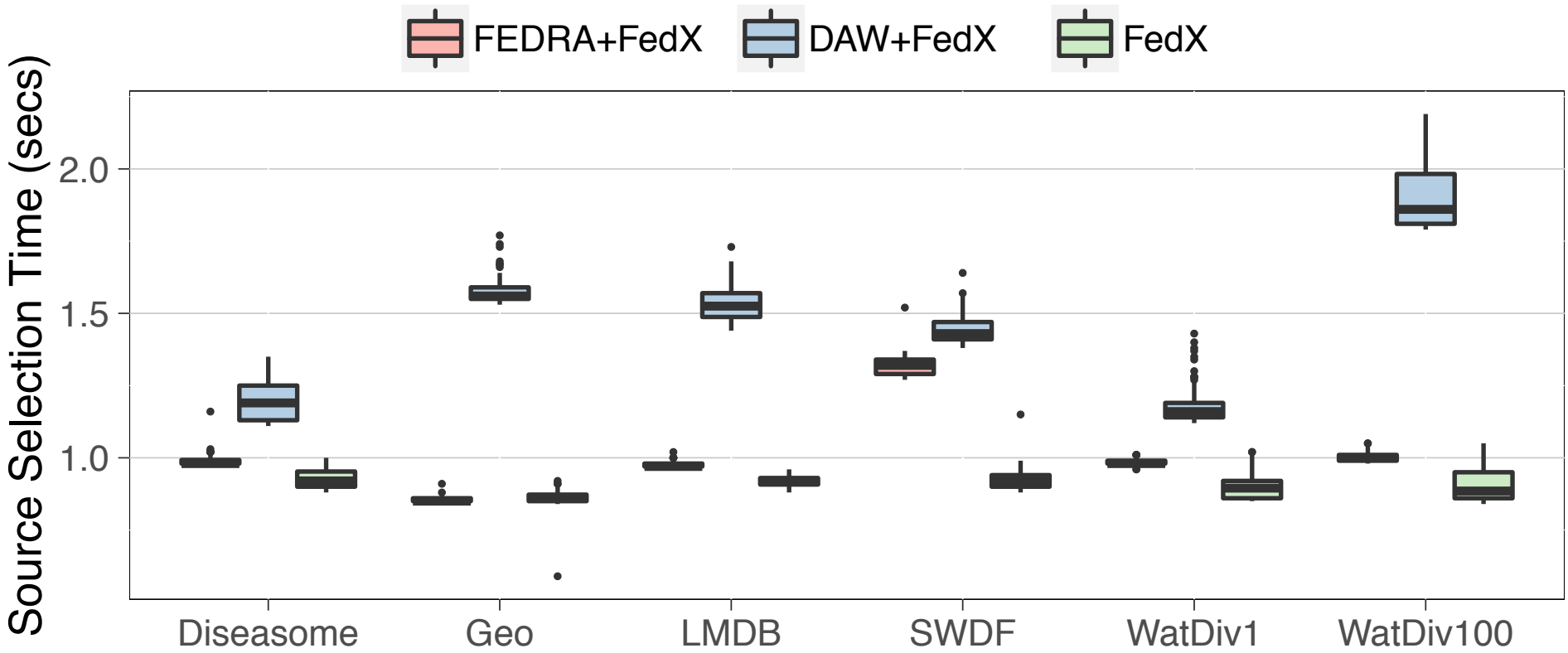
FEDRA does not reduce the number of obtained answers



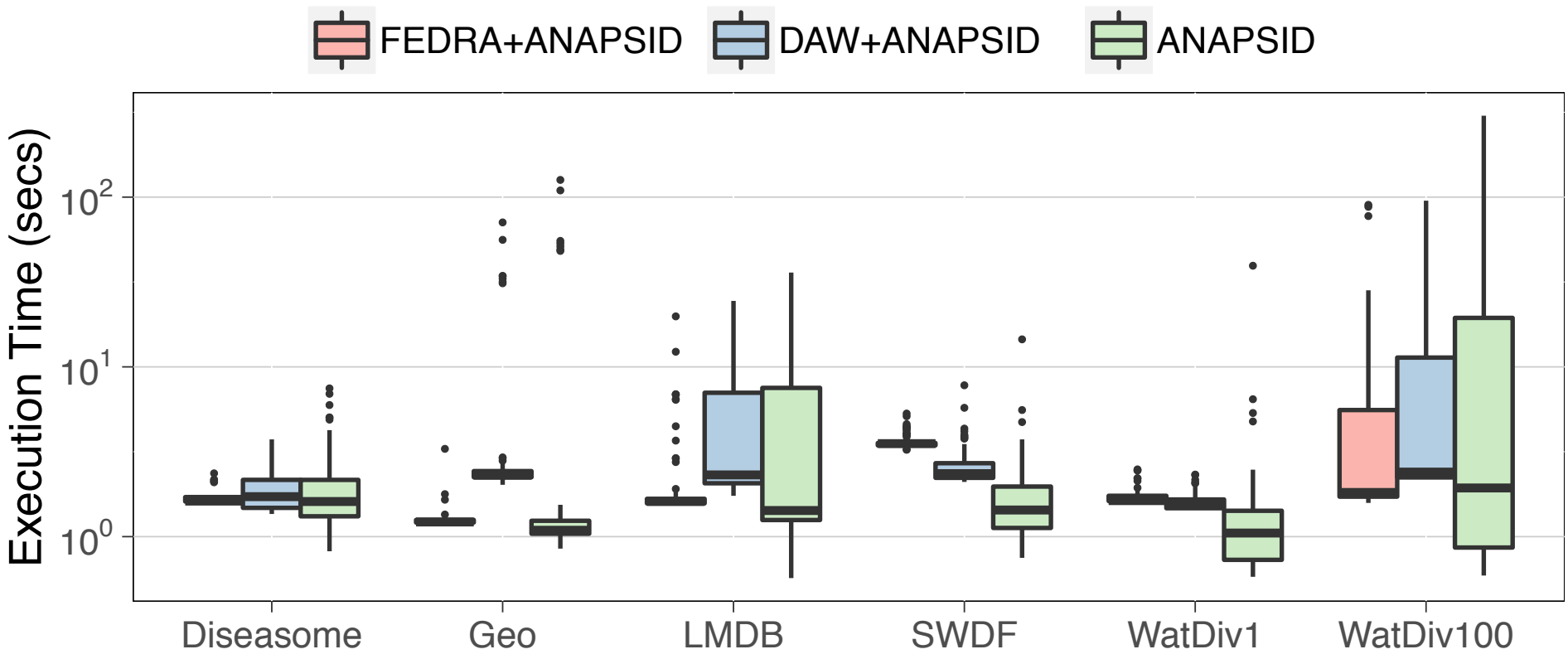
FEDRA source selection is more expensive than ANAPSID's



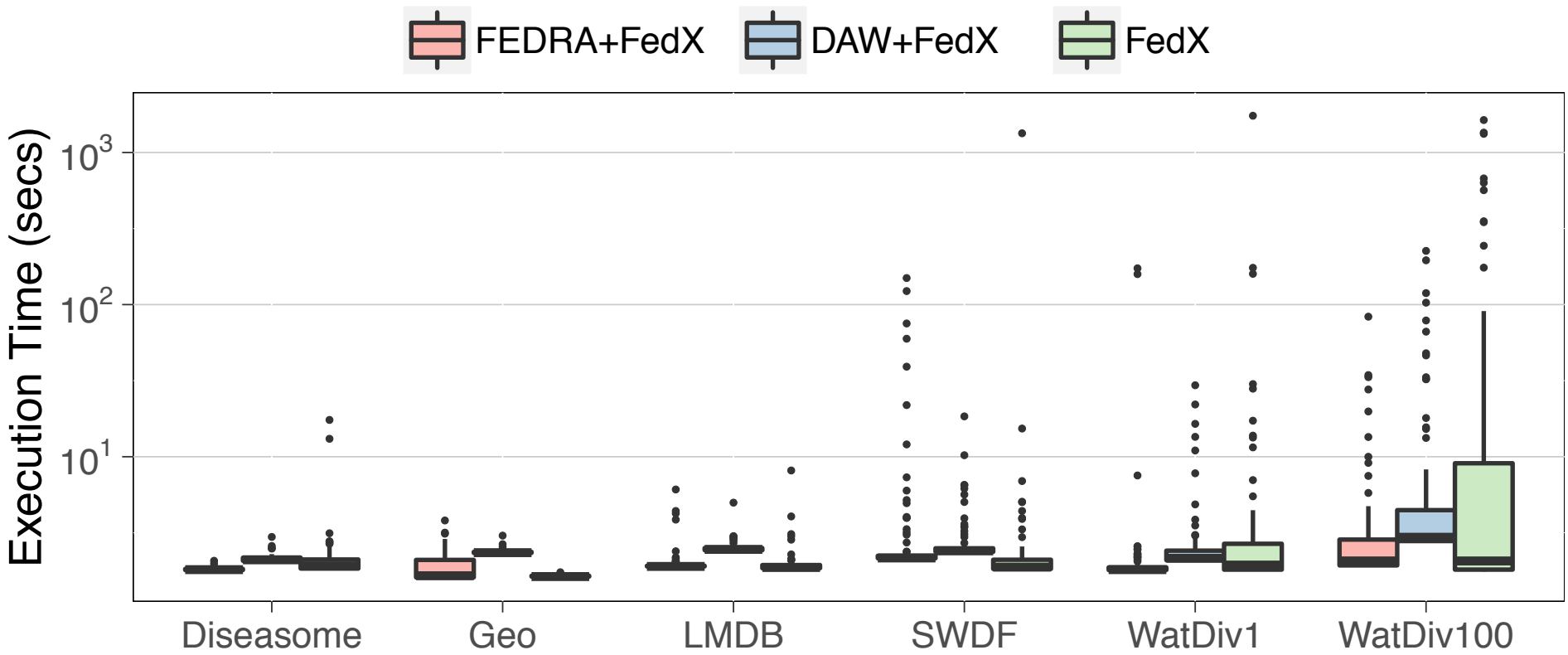
FEDRA source selection is faster than FedX's and DAW's



FEDRA+ANAPSID is faster than ANAPSID for many queries



FEDRA+FedX is faster than FedX and DAW+FedX



FEDRA selects less sources than the engines

The Wilcoxon signed rank test:
H0: FEDRA selects the same number of sources as FedX and ANAPSID
Ha: FEDRA selects less sources than FedX and ANAPSID

Federation	p-value	
	ANAPSID	FedX
Diseasome	$<$ **2.2e-16**	$<$ **2.2e-16**
SWDF	$<$ **2.2e-16**	$<$ **2.2e-16**
LinkedMDB	$<$ **2.2e-16**	$<$ **2.2e-16**
Geocoordinates	$<$ **2.2e-16**	$<$ **2.2e-16**
WatDiv1	**3.52e-016**	$<$ **2.2e-16**
WatDiv100	$<$ **2.2e-16**	**1.27e-015**

For all the federations and engines, the obtained p-values allow discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

FEDRA selects less sources than DAW

The Wilcoxon signed rank test:

H0: FEDRA selects the same number of sources as DAW does

Ha: FEDRA selects less sources than DAW

Federation	p-value	
	ANAPSID	FedX
Diseasome	$<$ **2.2e-16**	**8.371e-09**
SWDF	$<$ **2.2e-16**	**5.386e-11**
LinkedMDB	$<$ **2.2e-16**	**5.254e-11**
Geocoordinates	$<$ **2.2e-16**	**1.301e-05**
WatDiv1	**2.728e-13**	**1.006e-07**
WatDiv100	**4.794e-14**	**1.873e-05**

For all the federations and engines, the obtained p-values allow discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

FEDRA reduces the number of transferred tuples

The Wilcoxon signed rank test:

H0: using sources selected by FEDRA leads to transfer the same number of tuples as using sources selected by ANAPSID and FedX

Ha: using sources selected by FEDRA leads to transfer less tuples than using sources selected by ANAPSID and FedX

Federation	p-value	
	ANAPSID	FedX
Diseasome	$<$ **2.2e-16**	$<$ **2.2e-16**
SWDF	$<$ **2.2e-16**	**5.83e-05**
LinkedMDB	**4.155e-16**	$<$ **2.2e-16**
Geocoordinates	$<$ **2.2e-16**	$<$ **2.2e-16**
WatDiv1	**5.91e-16**	**4.03e-16**
WatDiv100	**4.74e-15**	**2.43e-15**

The obtained p-values allow discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

FEDRA reduction on the number of transferred tuples is greater than DAW's

The Wilcoxon signed rank test:

H0: using sources selected by FEDRA leads to transfer the same number of tuples as using sources selected by DAW

Ha: using sources selected by FEDRA leads to transfer less tuples than using sources selected by DAW

Federation	p-value	
	ANAPSID	FedX
Diseasome	$<$ **2.2e-16**	**6.334e-09**
SWDF	**2.198e-11**	0.6855
LinkedMDB	**1.296e-14**	**3.427e-05**
Geocoordinates	**1.29e-12**	0.5
WatDiv1	**1.188e-07**	**1.494e-07**
WatDiv100	**2.488e-05**	**1.293e-07**

For all the combinations in **bold**, the obtained p-values allow discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

Does FEDRA reduce the execution time?

The Wilcoxon signed rank test:

H0: using sources selected by FEDRA leads to the same query execution time as using sources selected by ANAPSID and FedX

Ha: using sources selected by FEDRA leads to faster query executions

Federation	p-value	
	ANAPSID	FedX
Diseasome	0.1306	$<$ **2.2e-16**
SWDF	1	1
LinkedMDB	0.254	0.9676
Geocoordinates	1	1
WatDiv1	1	**4.12e-13**
WatDiv100	0.0935	0.0011

For all the combinations in **bold**, the obtained p-values allow discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

Is FEDRA's reduction on execution time higher than DAW's?

The Wilcoxon signed rank test:
H0: using sources selected by FEDRA leads to transfer the same execution time as using sources selected by DAW
Ha: using sources selected by FEDRA leads to lower execution times than using sources selected by DAW

Federation	p-value	
	ANAPSID	FedX
Diseasome	**0.0002**	$<$ **2.2e-16**
SWDF	1	**6.79e-06**
LinkedMDB	$<$ **2.2e-16**	**9.22e-15**
Geocoordinates	$<$ **2.2e-16**	**7.87e-13**
WatDiv1	1	**6.32e-16**
WatDiv100	**5.39e-09**	**1.38e-14**

For all the combinations in **bold**, the obtained p-values allow discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

Is FEDRA the faster source selection strategy?

The Wilcoxon signed rank test:
H0: performing FEDRA source selection takes the same time as FedX and ANAPSID source selection strategies.
Ha: FEDRA source selection is faster than the engines source selections.

Federation	p-value	
	ANAPSID	FedX
Diseasome	1	1
SWDF	1	1
LinkedMDB	1	1
Geocoordinates	1	**3.12e-08**
WatDiv1	1	1
WatDiv100	1	1

Only for FedX and Geocoordinates, the obtained p-value allows discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).

If FEDRA source selection faster than DAW's?

The Wilcoxon signed rank test:
H0: performing FEDRA and DAW source selections consume the same time
Ha: performing FEDRA source selection is faster than DAW's.

Federation	p-value	
	ANAPSID	FedX
Diseasome	1	$<$ **2.2e-16**
SWDF	1	$<$ **2.2e-16**
LinkedMDB	**1.28e-11**	$<$ **2.2e-16**
Geocoordinates	$<$ **2.2e-16**	$<$ **2.2e-16**
WatDiv1	1	$<$ **2.2e-16**
WatDiv100	$<$ **2.2e-16**	$<$ **2.2e-16**

For all the combinations in **bold**, the obtained p-values allow discarding the null hypothesis (H0) in favor of the alternative hypothesis (Ha).