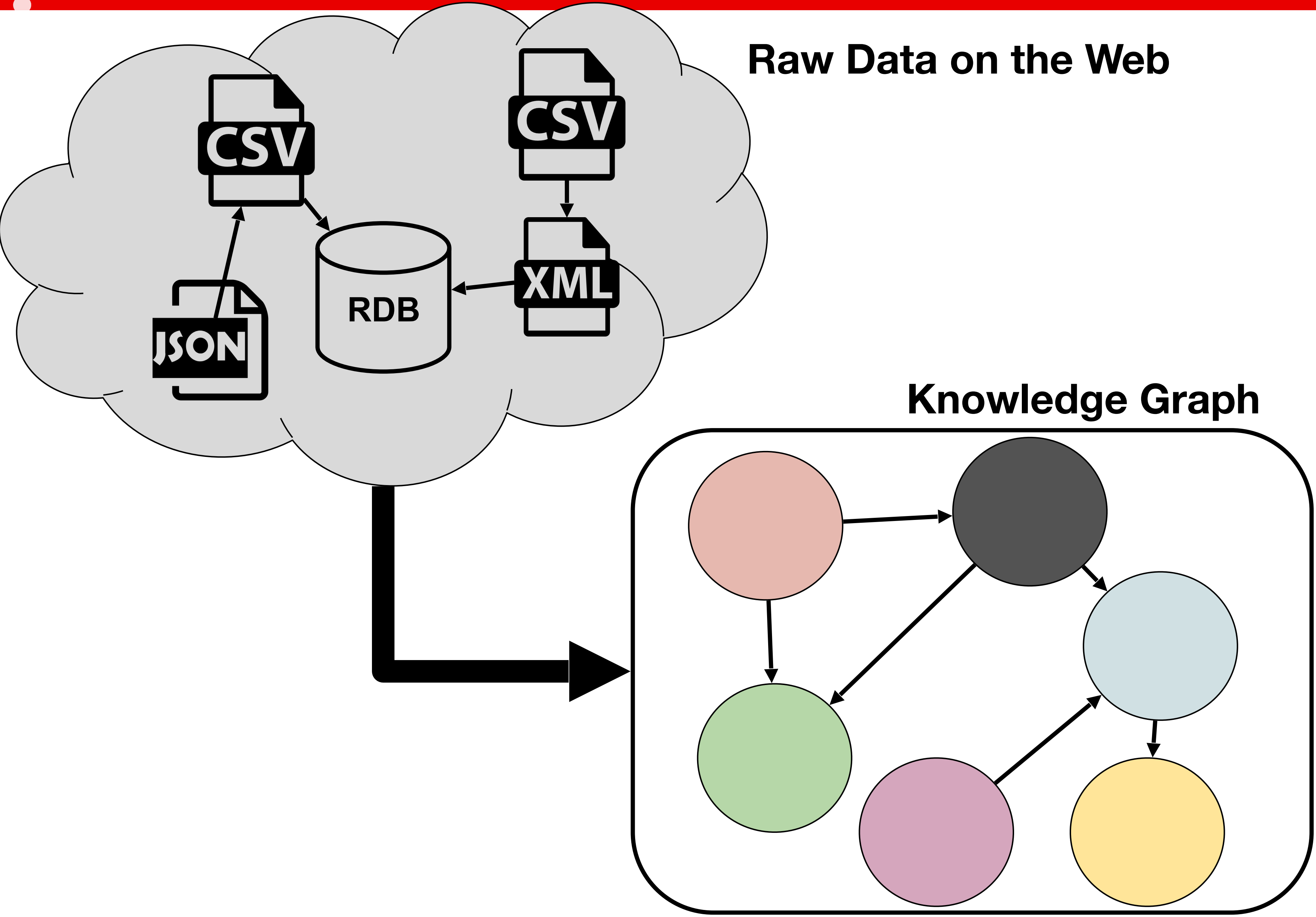
What are the Parameters that Affect the Construction of a Knowledge Graph?

David Chaves-Fraga*, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Kemele M. Endris, L3S Research Center & TIB
Enrique Iglesias, University of Bonn
Oscar Corcho, OEG - UPM
Maria-Esther Vidal, L3S Research Center & TIB

***Work done during the research visit of David Chaves-Fraga to TIB and L3S**

dchaves@fi.upm.es
@dchavesf

22/10/2019
ODBASE@2019 (Rhodes)


Raw Data on the Web
CSV
CSV
XML
JSON
RDB
Knowledge Graph

Knowledge Graph Creation Engines

Triplify

TARQL

SPARQL-Generate

RML-Mapper

CARML

RocketRML

SDM-RDFizer

RMLStreamer

Declarative Knowledge Graph Creation Engines

Triplify

TARQL

SPARQL-Generate

Functional KGC Engines

RML-Mapper

CARML

RocketRML

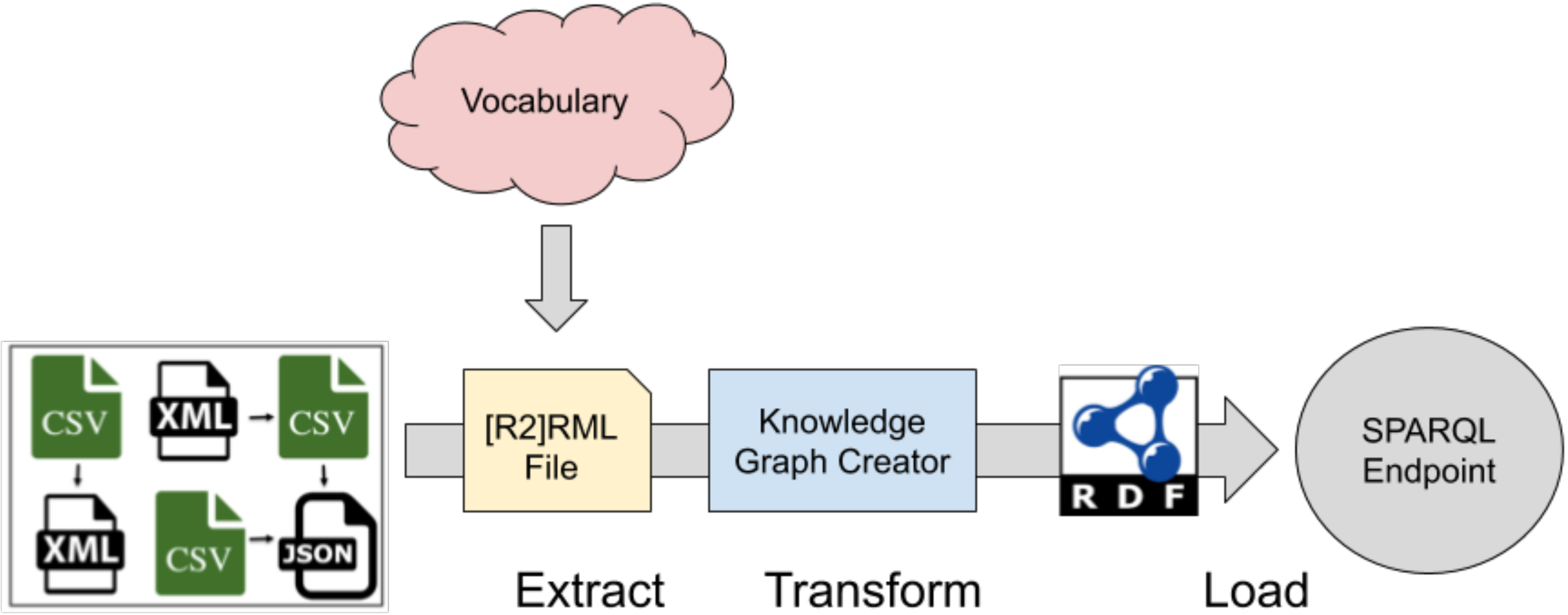
SDM-RDFizer

RMLStreamer

Declarative KGC Engines

RML R2RML

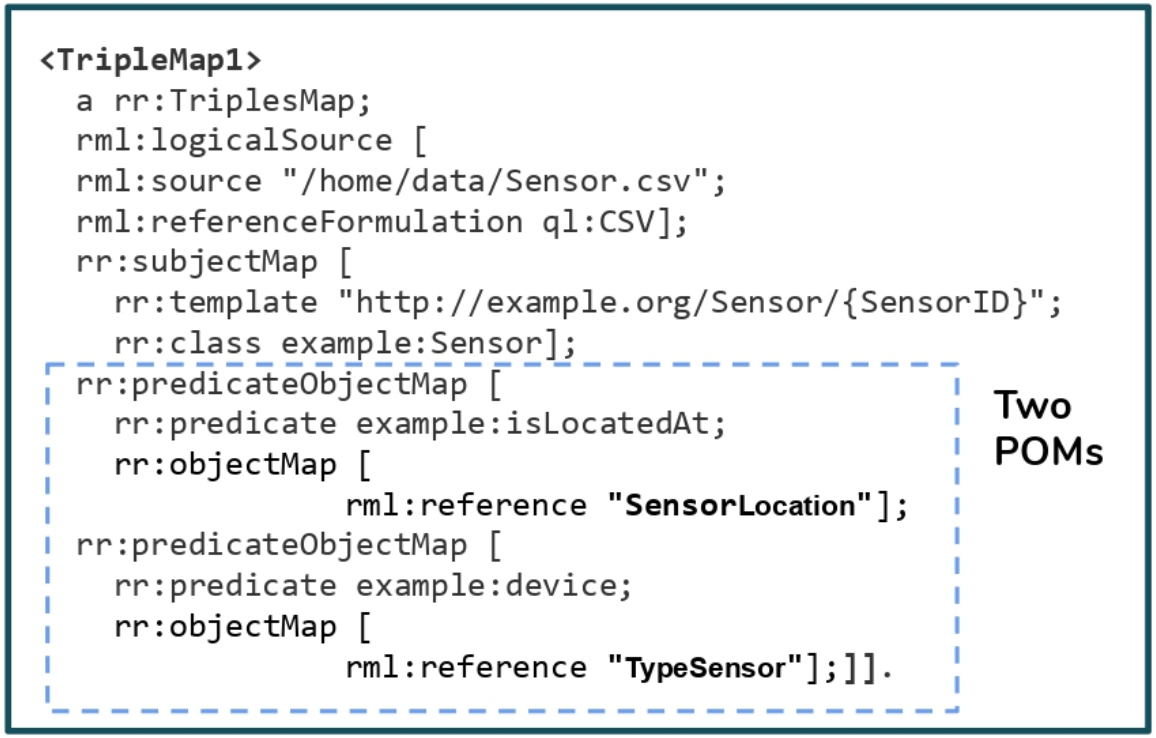
Declarative Knowledge Graph Creation



RDF Mapping Language (RML)

Sensor.csv

SensorID	SensorLocation	TypeSensor
1	loc1	typeA
2	loc2	typeB



```
<TripleMap1>
  a rr:TriplesMap;
  rml:logicalSource [
  rml:source "/home/data/Sensor.csv";
  rml:referenceFormulation ql:CSV];
  rr:subjectMap [
    rr:template "http://example.org/Sensor/{SensorID}";
    rr:class example:Sensor];
  rr:predicateObjectMap [
    rr:predicate example:isLocatedAt;
    rr:objectMap [
                rml:reference "SensorLocation"];
  rr:predicateObjectMap [
    rr:predicate example:device;
    rr:objectMap [
                rml:reference "TypeSensor"];]].
```

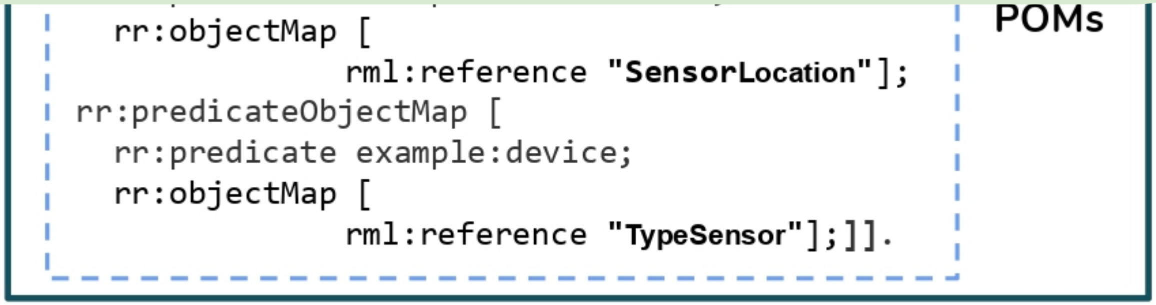
Two POMs

RDF Mapping Language (RML)

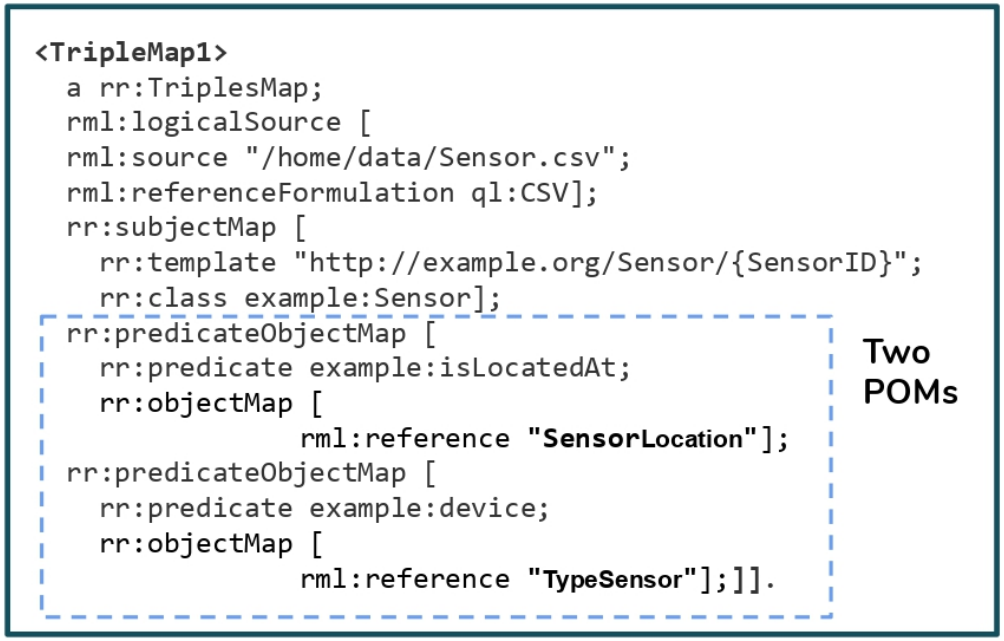
Sensor.csv

SensorID	SensorLocation	TypeSensor

```
ex:Sensor/1    a                ex:Sensor .
ex:Sensor/1    ex:isLocatedAt   “loc1” .
ex:Sensor/1    ex:device        “typeA” .
ex:Sensor/2    a                ex:Sensor .
ex:Sensor/2    ex:isLocatedAt   “loc2” .
ex:Sensor/2    ex:device        “typeB” .
```



```
    rr:objectMap [
                  rml:reference "SensorLocation"];
  rr:predicateObjectMap [
    rr:predicate example:device;
    rr:objectMap [
                  rml:reference "TypeSensor"];]].
```



```
<TripleMap1>
  a rr:TriplesMap;
  rml:logicalSource [
  rml:source "/home/data/Sensor.csv";
  rml:referenceFormulation ql:CSV];
  rr:subjectMap [
    rr:template "http://example.org/Sensor/{SensorID}";
    rr:class example:Sensor];
  rr:predicateObjectMap [
    rr:predicate example:isLocatedAt;
    rr:objectMap [
                rml:reference "SensorLocation"];
  rr:predicateObjectMap [
    rr:predicate example:device;
    rr:objectMap [
                rml:reference "TypeSensor"];]].
```

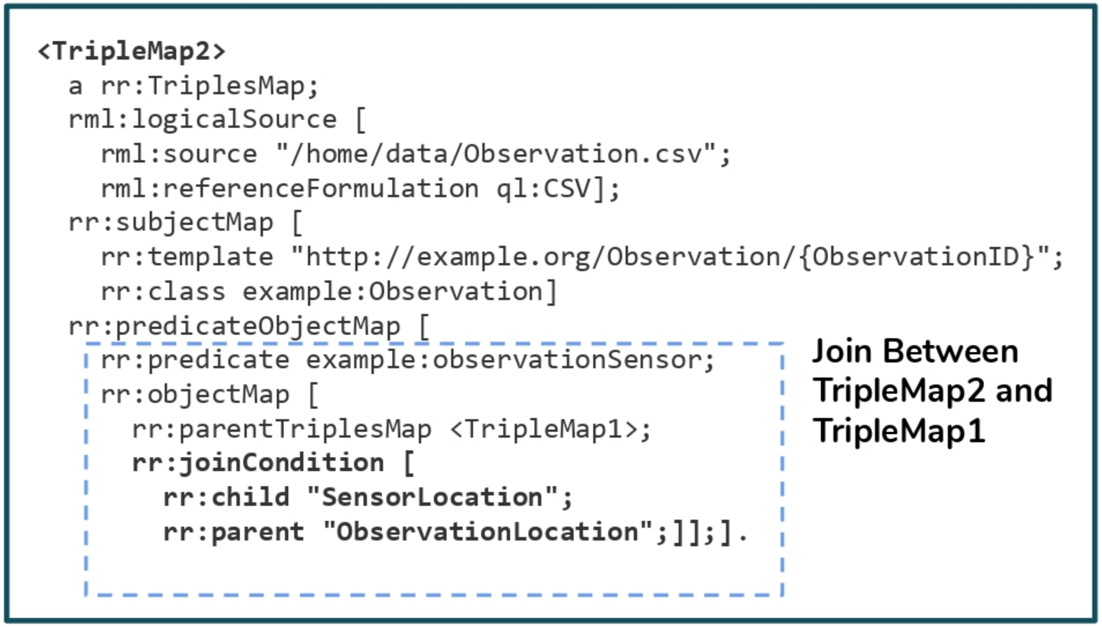
Two POMs

Sensor.csv

SensorID	Sensor Location	Type Sensor
1	loc1	typeA
2	loc2	typeB

Observation.csv

ObservationID	Observation Location
1	loc1
2	loc2



```
<TripleMap2>
  a rr:TriplesMap;
  rml:logicalSource [
    rml:source "/home/data/Observation.csv";
    rml:referenceFormulation ql:CSV];
  rr:subjectMap [
    rr:template "http://example.org/Observation/{ObservationID}";
    rr:class example:Observation]
  rr:predicateObjectMap [
    rr:predicate example:observationSensor;
    rr:objectMap [
      rr:parentTriplesMap <TripleMap1>;
      rr:joinCondition [
        rr:child "SensorLocation";
        rr:parent "ObservationLocation";]];].
```

Join Between TripleMap2 and TripleMap1

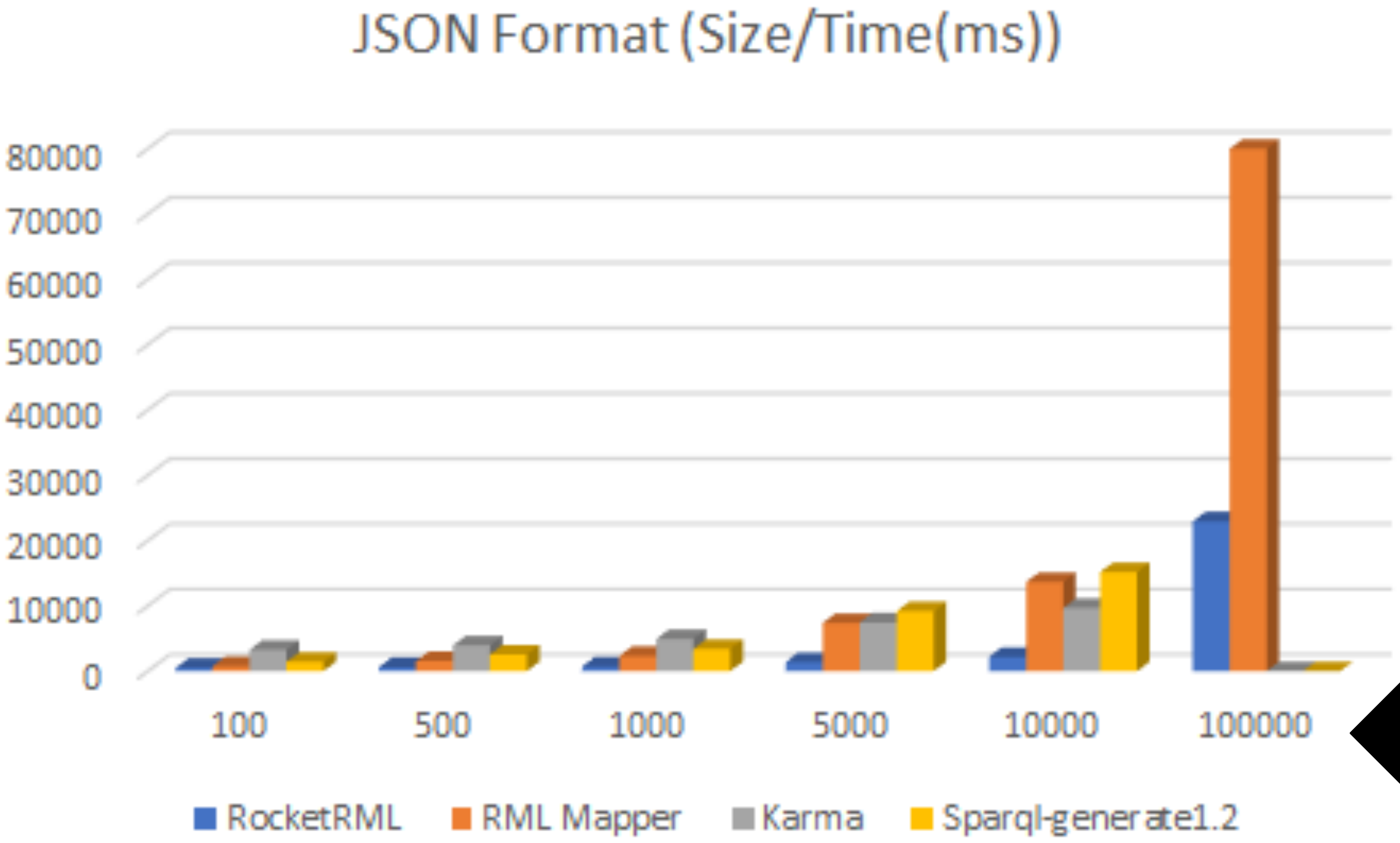
RDF Mapping Language (RML)

```
<TripleMap1>
  a rr:TriplesMap;
  rml:logicalSource [
  rml:source "/home/data/Sensor.csv";
  rml
  r
```

```
ex:Sensor/1         a                      ex:Sensor.
ex:Sensor/1         ex:isLocatedAt         “loc1”.
ex:Sensor/1         ex:device              “typeA”.
ex:Sensor/2         a                      ex:Sensor.
ex:Sensor/2         ex:isLocatedAt          “loc2”.
ex:Sensor/2         ex:device               “typeB”.
ex:Observation/1  a                       ex:Observation .
ex:Observation/1  ex:observationSensor  ex:Sensor/1.
ex:Observation/2  a                       ex:Observation .
ex:Observation/2  ex:observationSensor  ex:Sensor/2.
```

O	
1	
2	loc2

```
rr:joinCondition [
  rr:child "SensorLocation";
  rr:parent "ObservationLocation";]];].
```



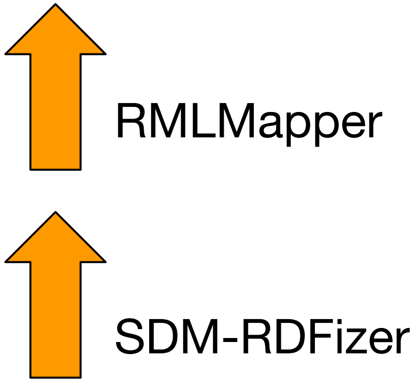
Simsek, U. et al (2019). **RocketRML - A NodeJS Implementation of Use-case Specific RML Mapper.** Proceedings of the 1s International Workshop on Knowledge Graph Building co-located with the 16th Extended Semantic Web Conference

Motivating Example

Size	SDM-RDFizer	RMLMapper
Two POM	1.72	0.92
Five POM	1.85	1.84
Ten POM	1.98	3.46

Motivating Example

Size	SDM-RDFizer	RMLMapper
Two POM	1.72	0.92
Five POM	1.85	1.84
Ten POM	1.98	3.46



RMLMapper

SDM-RDFizer

Motivating Example

Size	SDM-RDFizer	RMLMapper
Two POM	1.72	0.92
Five POM	1.85	1.84
Ten POM	1.98	3.46

↑ RMLMapper

↑ SDM-RDFizer

Join Selectivity	SDM-RDFizer	RMLMapper
High	2.16	38.6
Medium	2.20	40.43
Low	2.19	46.06

Motivating Example

Size	SDM-RDFizer	RMLMapper
Two POM	1.72	0.92
Five POM	1.85	1.84
Ten POM	1.98	3.46

⬆ RMLMapper

⬆ SDM-RDFizer

Join Selectivity	SDM-RDFizer	RMLMapper
High	2.16	38.6
Medium	2.20	40.43
Low	2.19	46.06

⬆ RMLMapper

= SDM-RDFizer

Challenges

- Identify **variables** and **configuration** setups that may provide **accurate** and **well-informed** overview of **knowledge graph creation** engines' performance:
  - mappings,
  - data distribution,
  - serialisation, data format, ...

- Identify **variables** and **configuration** setups that may provide **accurate** and **well-informed** overview of **knowledge graph creation** engines' performance:
    - mappings,
    - data distribution,
    - serialisation, data format, ...

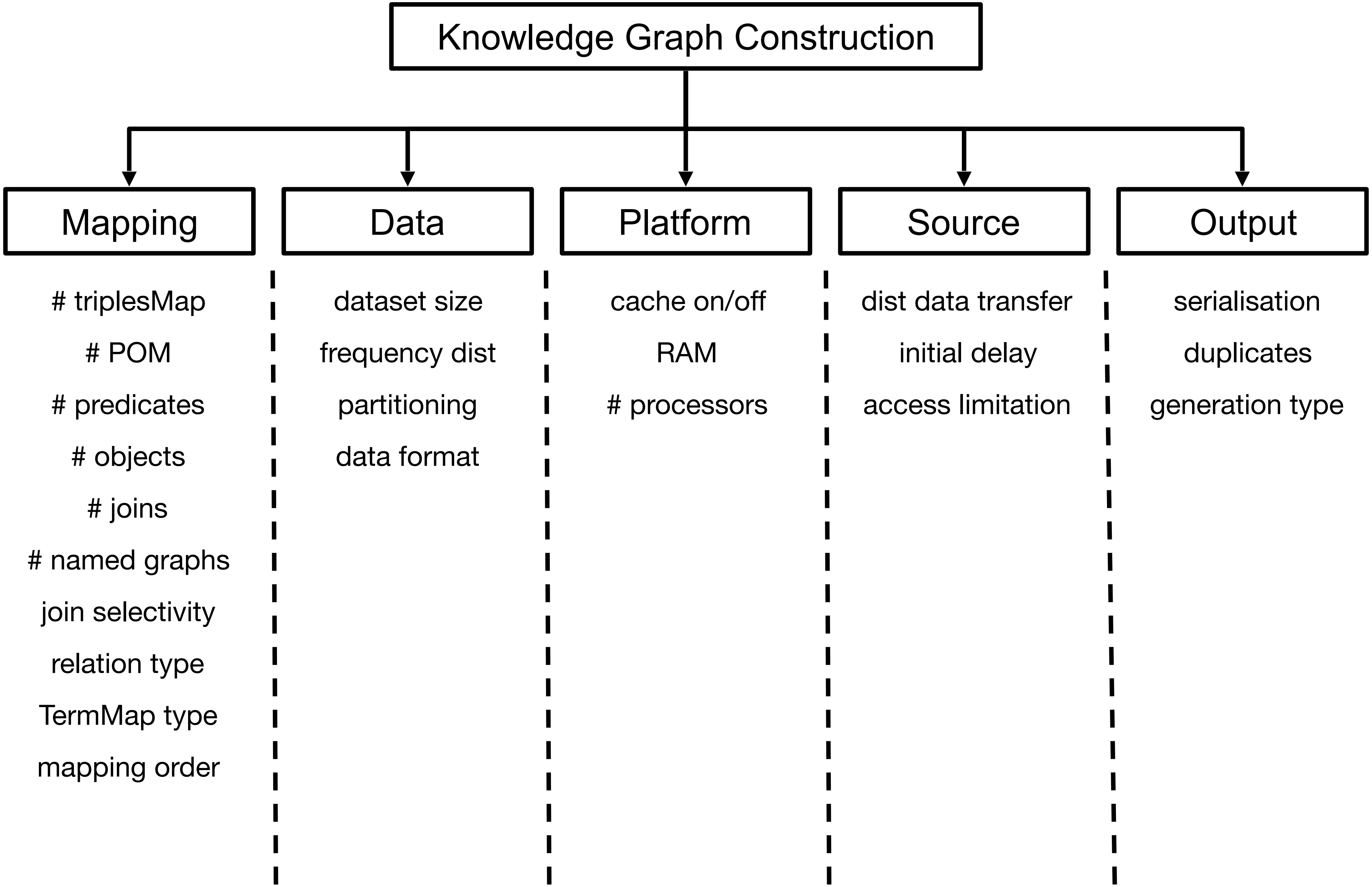
- Empirically **evaluate** the **performance of** the state-of-art engines and study their **behaviour**

Parameters Affecting KG Constructions

- **Independent variables:** need to be specified in a testbed to ensure reproducibility:
  - number of joins, data size, RAM available, serialisation, etc.

- **Observed variables** (measurements):
  - Execution time and completeness.

Independent Variables



Knowledge Graph Construction

Mapping
- # triplesMap
- # POM
- # predicates
- # objects
- # joins
- # named graphs
- join selectivity
- relation type
- TermMap type
- mapping order

Data
- dataset size
- frequency dist
- partitioning
- data format

Platform
- cache on/off
- RAM
- # processors

Source
- dist data transfer
- initial delay
- access limitation

Output
- serialisation
- duplicates
- generation type

Mapping variables

Independent Variables	Observed Variables	
	Execution Time	Completeness
Mapping order	X	
# triplesMap	X	X
# predicateObjectMaps	X	X
# predicates	X	X
# objects	X	X
# joins	X	X
# named graphs	X	X
join selectivity	X	X
relation type	X	X
object TermMap Type	X	

Impact of the Relation Type

Relation Type	RMLMapper	SDM-RDFizer
1-1	42.86	2.19
1-N	43.34	2.19
N-1	43.26	2.15
N-M	78.64	2.33

* (N = 15 in 1-N and N-1, N=M=10 in N-M)

Relation Type	RMLMapper	SDM-RDFizer
1-1	42.86	2.19
1-N	43.34	2.19
N-1	43.26	2.15
N-M	78.64	2.33

SDM-RDFizer **performs better** in N-1 than 1-N

* (N = 15 in 1-N and N-1, N=M=10 in N-M)

Impact of the Relation Type

Relation Type	RMLMapper	SDM-RDFizer
1-1	42.86	2.19
1-N	43.34	2.19
N-1	43.26	2.15
N-M	7[illegible].64	2.[illegible]3

RMLMapper is **not affected** by 1-N and N-1

SDM-RDFizer **performs better** in N-1 than 1-N

* (N = 15 in 1-N and N-1, N=M=10 in N-M)

Impact of the Relation Type

Relation Type	RMLMapper	SDM-RDFizer
1-1	42.86	2.19
1-N	43.34	2.19
N-1	43.26	2.15
N-M	78.64	2.33

* (N = 15 in 1-N and N-1, N=M=10 in N-M)

Impact of the Relation Type

Relation Type	RMLMapper	SDM-RDFizer
1-1	42.86	2.19
1-N	43.34	2.19
N-1	43.26	2.15
N-M	78.64	2.33

Both are **affected** by N-M relations

* (N = 15 in 1-N and N-1, N=M=10 in N-M)

Data variables

Independent Variables	Observed Variables	
	Execution Time	Completeness
dataset size	X	
data frequency distribution	X	
initial delay	X	X
data format	X	X

Impact of the Partitioning

Partitioning Type	RMLMapper	SDM-RDFizer
Horizontal without duplicates	1904.31	4.84
Vertical without duplicates	2067.77	4.73
Horizontal with duplicates	2276.98	5.86
Vertical with duplicates	2024.66	4.98

Impact of the Partitioning

Partitioning Type	RMLMapper	SDM-RDFizer
Horizontal without duplicates	1904.31	4.84
Vertical without duplicates	2067.77	4.73
Horizontal with duplicates	2276.98	5.86
Vertical with duplicates	024.66	4.98

Both behaves **similar** in horizontal partitioning

Impact of the Partitioning

Partitioning Type	RMLMapper	SDM-RDFizer
Horizontal without duplicates	1904.31	4.84
Vertical without duplicates	2067.77	4.73
Horizontal with duplicates	2276.98	5.86
Vertical with duplicates	2024.66	4.98

Impact of the Partitioning

Partitioning Type	RMLMapper	SDM-RDFizer
Horizontal without duplicates	1904.31	4.84
Vertical without duplicates	2067.77	4.73
Horizontal with duplicates	2276.98	5.36
Vertical with duplicates	2024.66	4.98

Different behavior in vertical partitioning

Platform, Source and Output variables

Independent Variables		Observed Variables	
		Execution Time	Completeness
Platform	cache on/off	X	
	RAM available	X	
	# processor	X	
Source	distribution data transfer	X	X
	initial delay	X	
	access limitation	X	X
Output	serialisation	X	X
	duplicates	X	X
	generation type	X	X

Goal: Empirically demonstrate how the behaviour of engines to create knowledge graphs is affected in different configurations and testbeds.

- **RQ1)** What is the effect of mixing different variables in one testbed?
- **RQ2)** What is the impact of considering configurations of different complexity of the same variable in one testbed?
- **RQ3)** Do the different variables and configurations influence in the behaviour of existing knowledge graph creation tools?

Datasets:

- **Naïve:**
  - 2 files, 30 columns per file
- **Relation type:**
  - 1-N, N-1 with N = {1, 5, 10, 15}
  - N-M, N=M={1, 3, 5, 10}
- **Join Duplicates:**
  - Low (5% to 20% duplicates)
  - High (30% to 50% duplicates)
- **Join Selectivity:**
  - High (5% to 20% matches)
  - Low (60% to 100% matches)

Common features:

- **Size:** 1k, 10k and 50k rows
- **Format:** Local CSV files
- **Output:** N-Triples

Resource available at:

https://github.com/SDM-TIB/KGC-Param-Eval

Engines (selected based on RML-Implementation-Report):

- RMLMapper: https://github.com/RMLio/rmlmapper-java
- SDM-RDFizer: https://github.com/SDM-TIB/SDM-RDFizer

Comparison using Pearson’s correlations:

Negative correlation (between 0 and -1)

↳ Trends of execution time of the tools are **opposite**

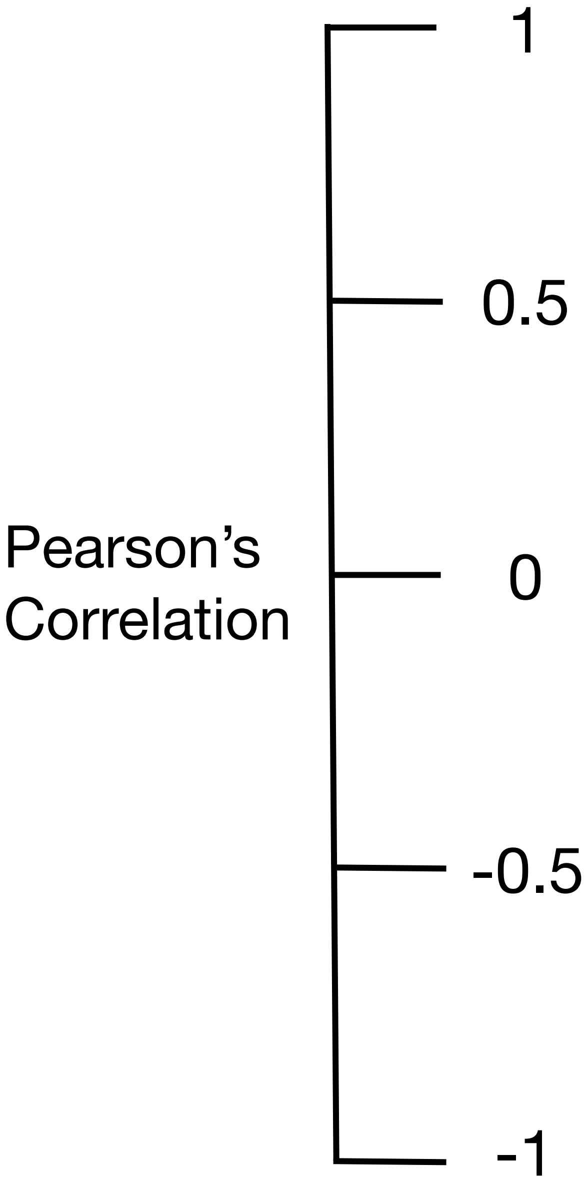
Positive correlation (between 0 and 1)

↳ Trends of execution time of the tools are **similar**

Total positive correlation (1)

↳ Comparing **same** configuration

Experimental setup



Pearson's Correlation

1

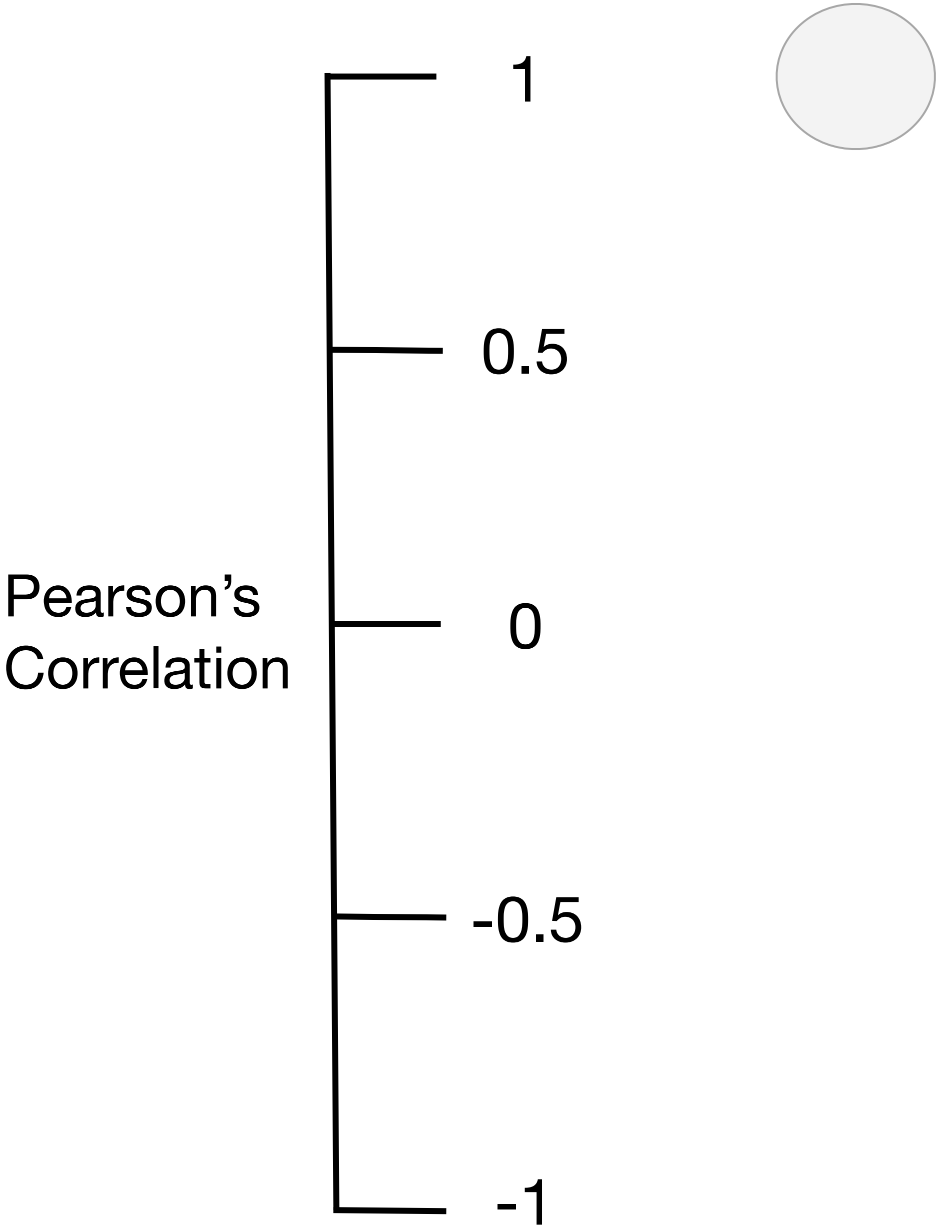
0.5

0

-0.5

-1

Experimental setup



Pearson's Correlation

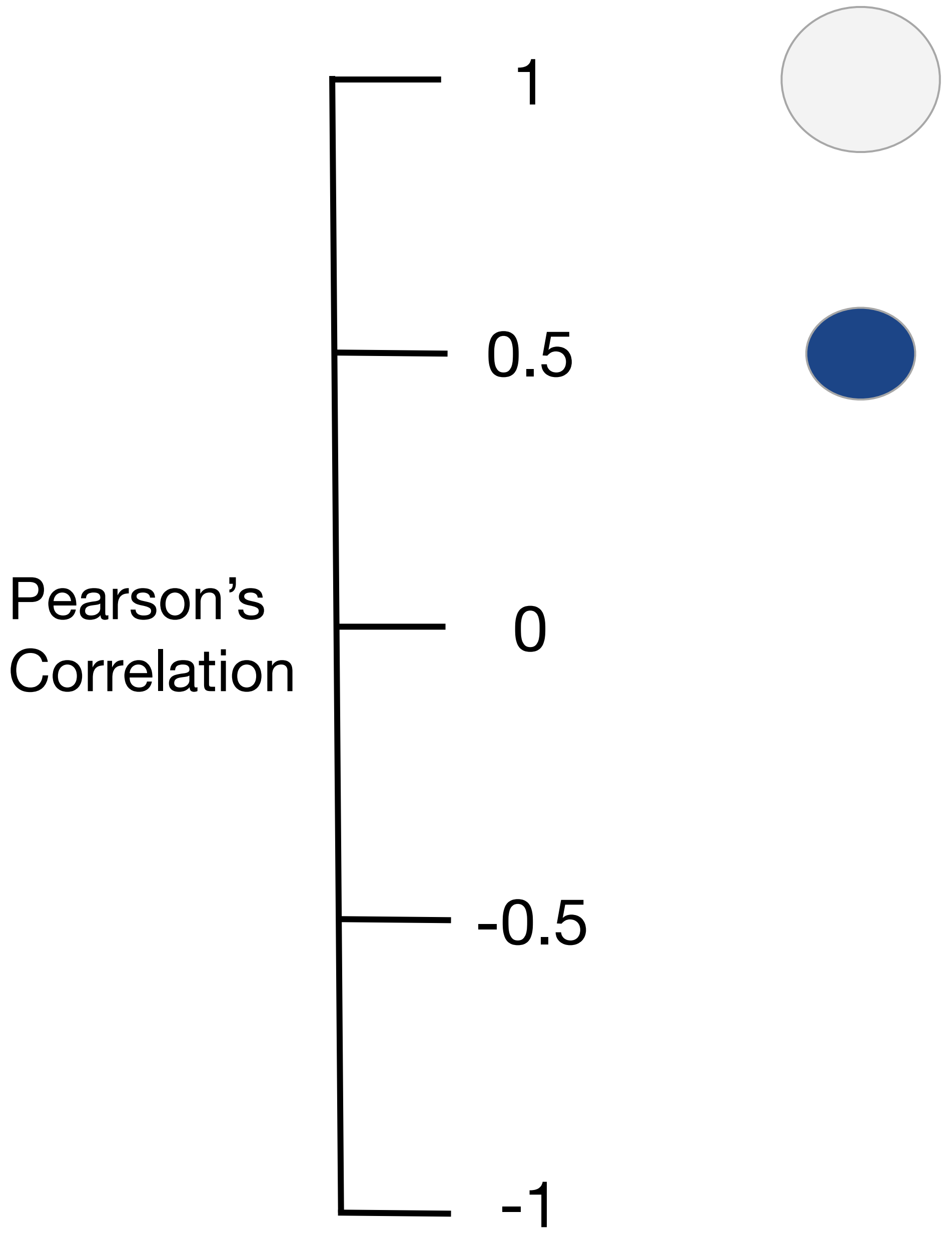
1

0.5

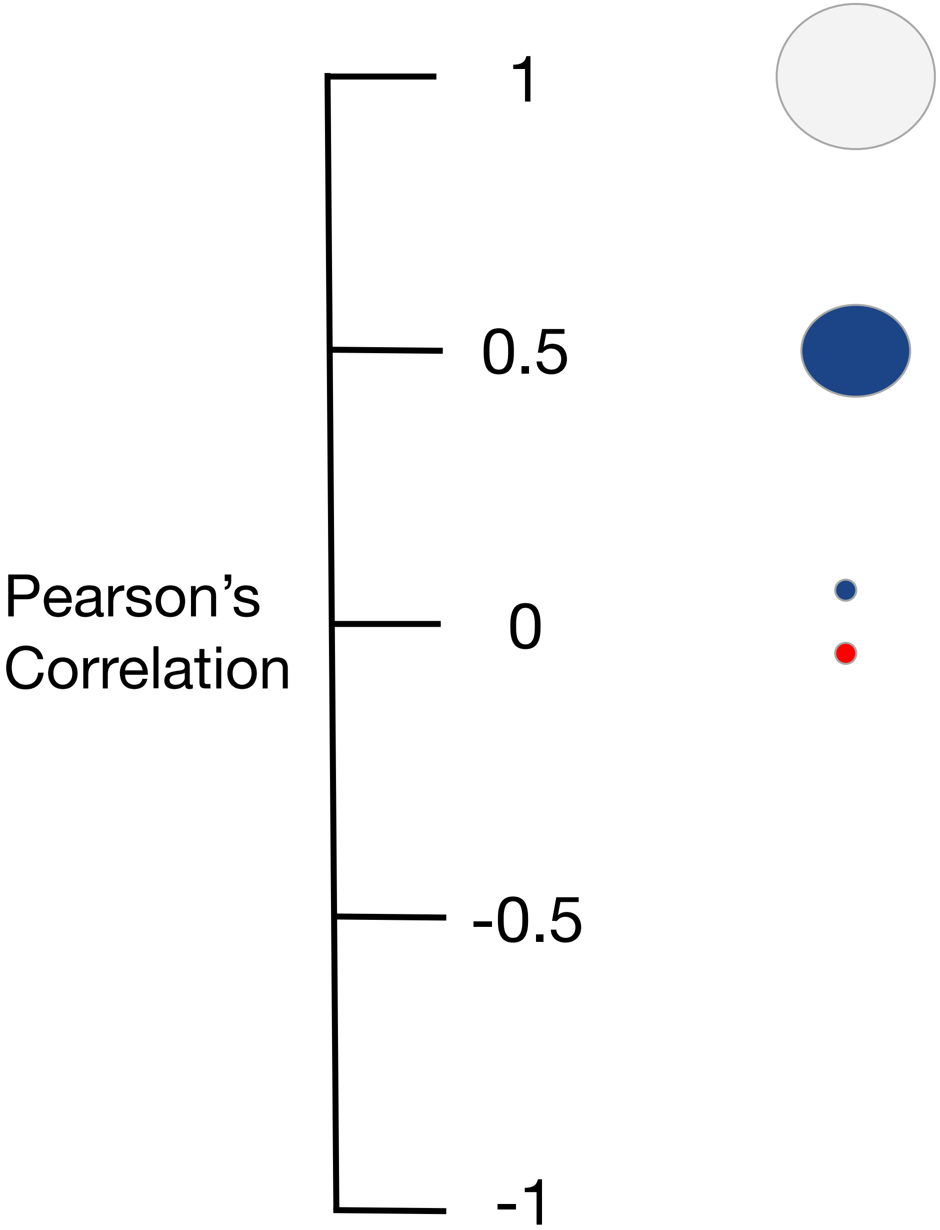
0

-0.5

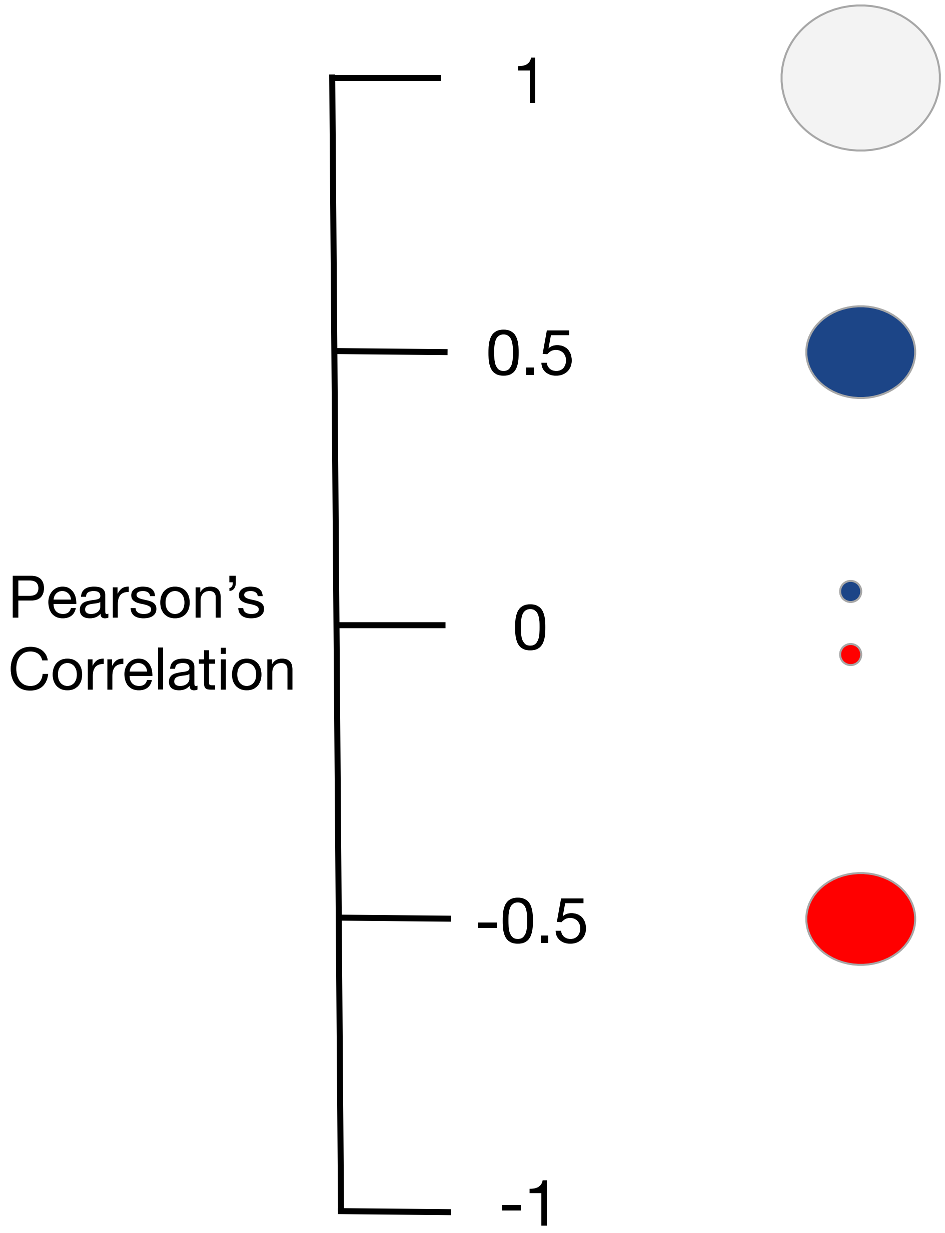
-1

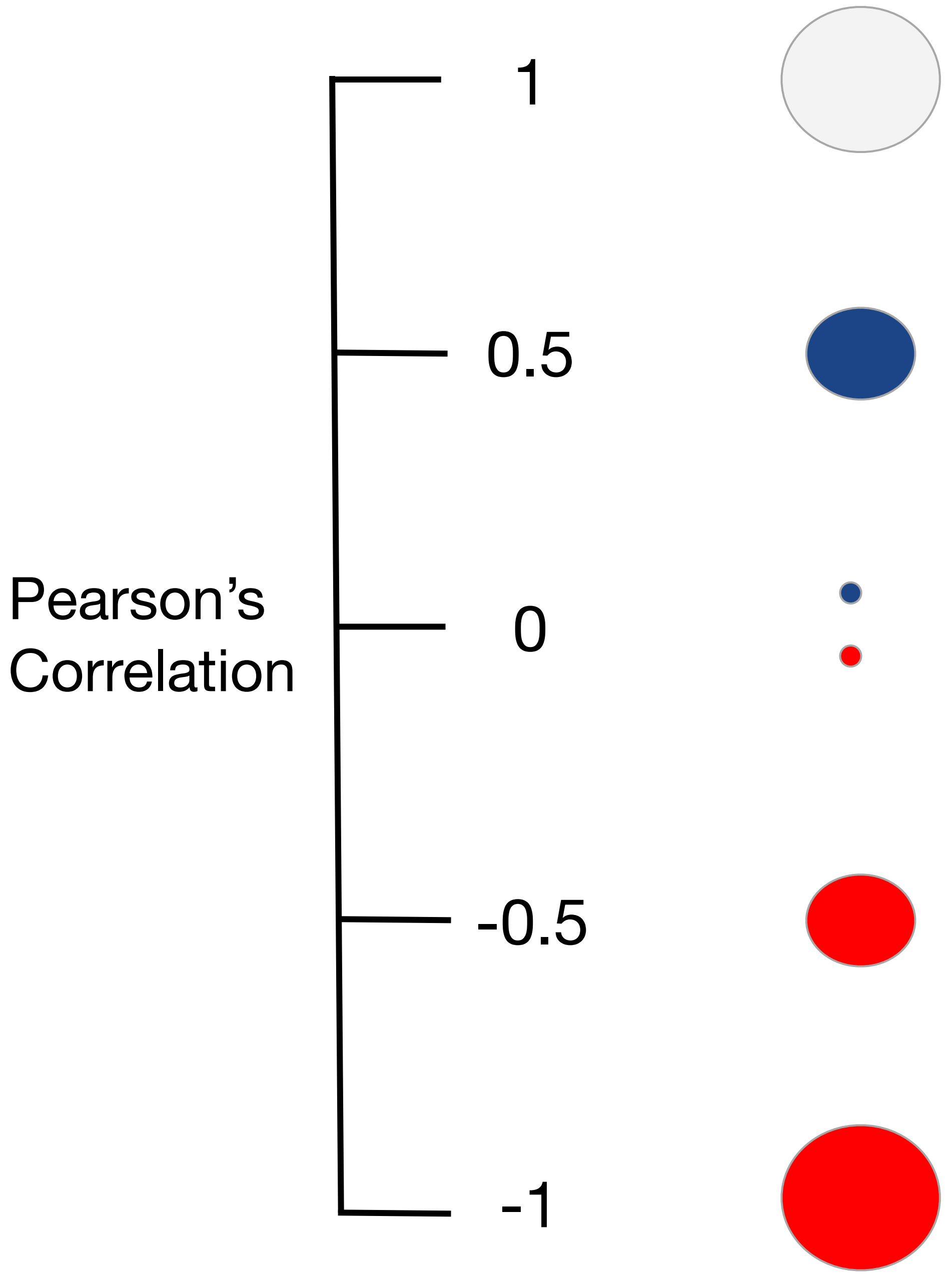

Pearson's
Correlation
1
0.5
0
-0.5
-1

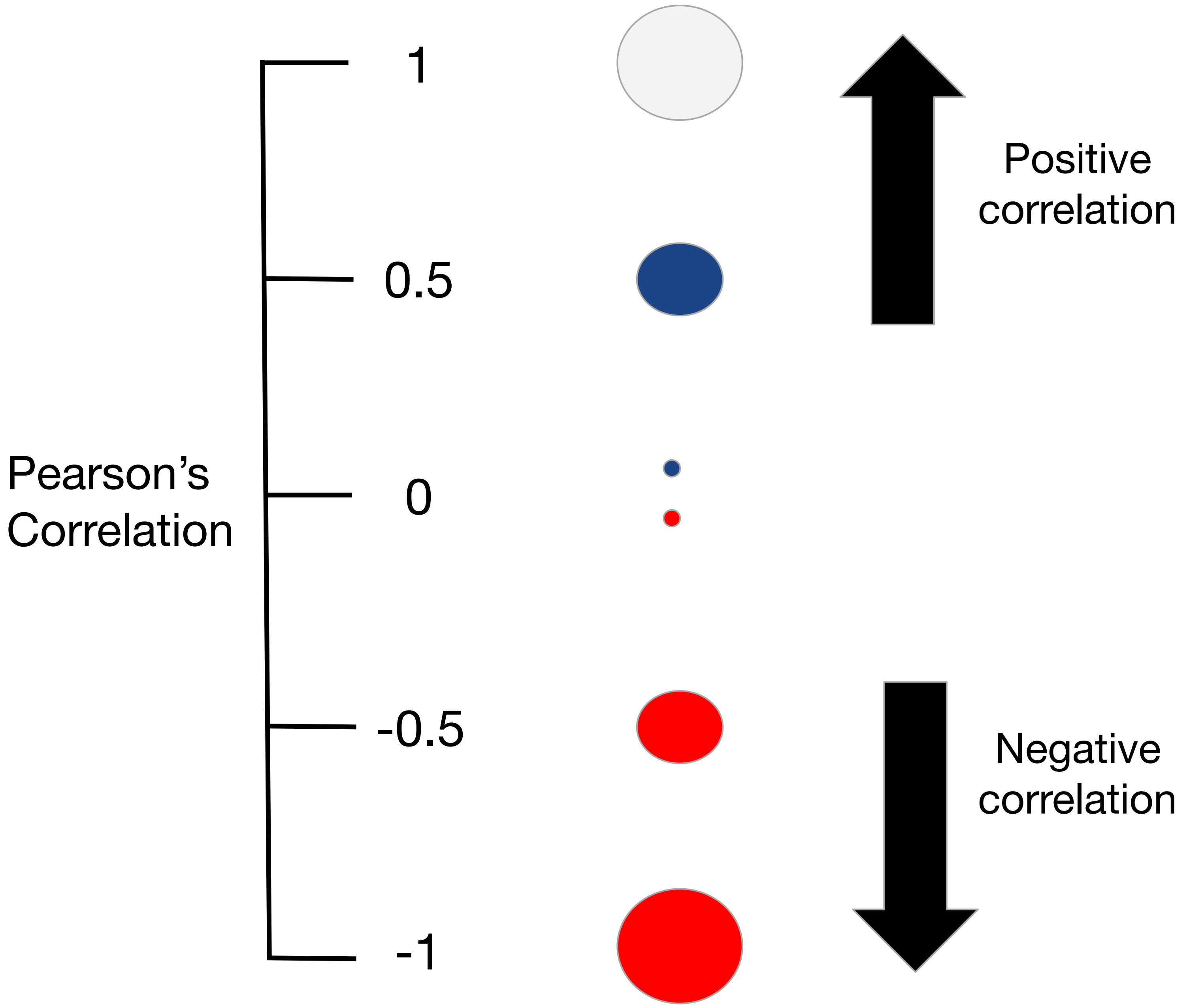
Experimental setup



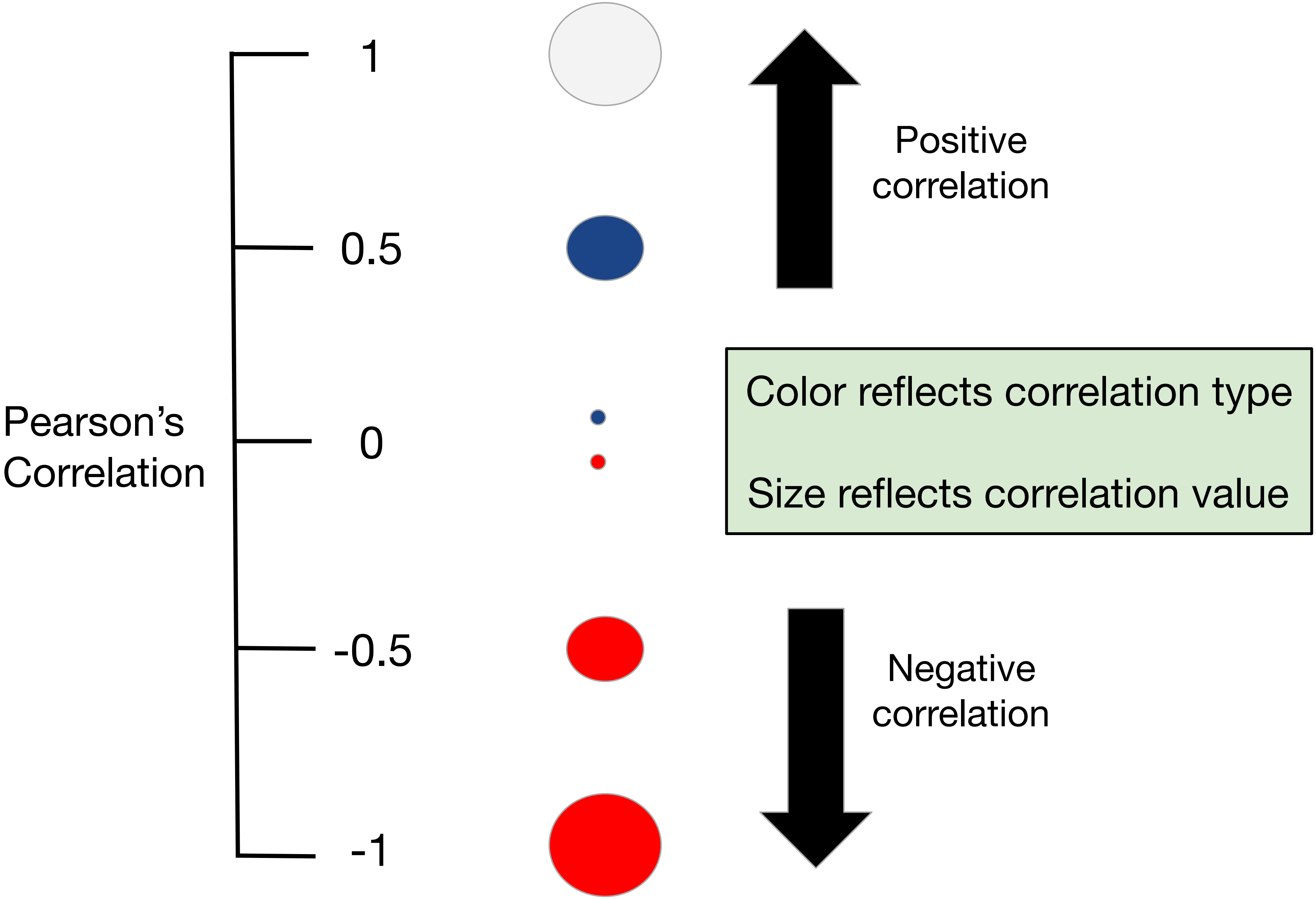
Experimental setup




Pearson's
Correlation
1
0.5
0
-0.5
-1

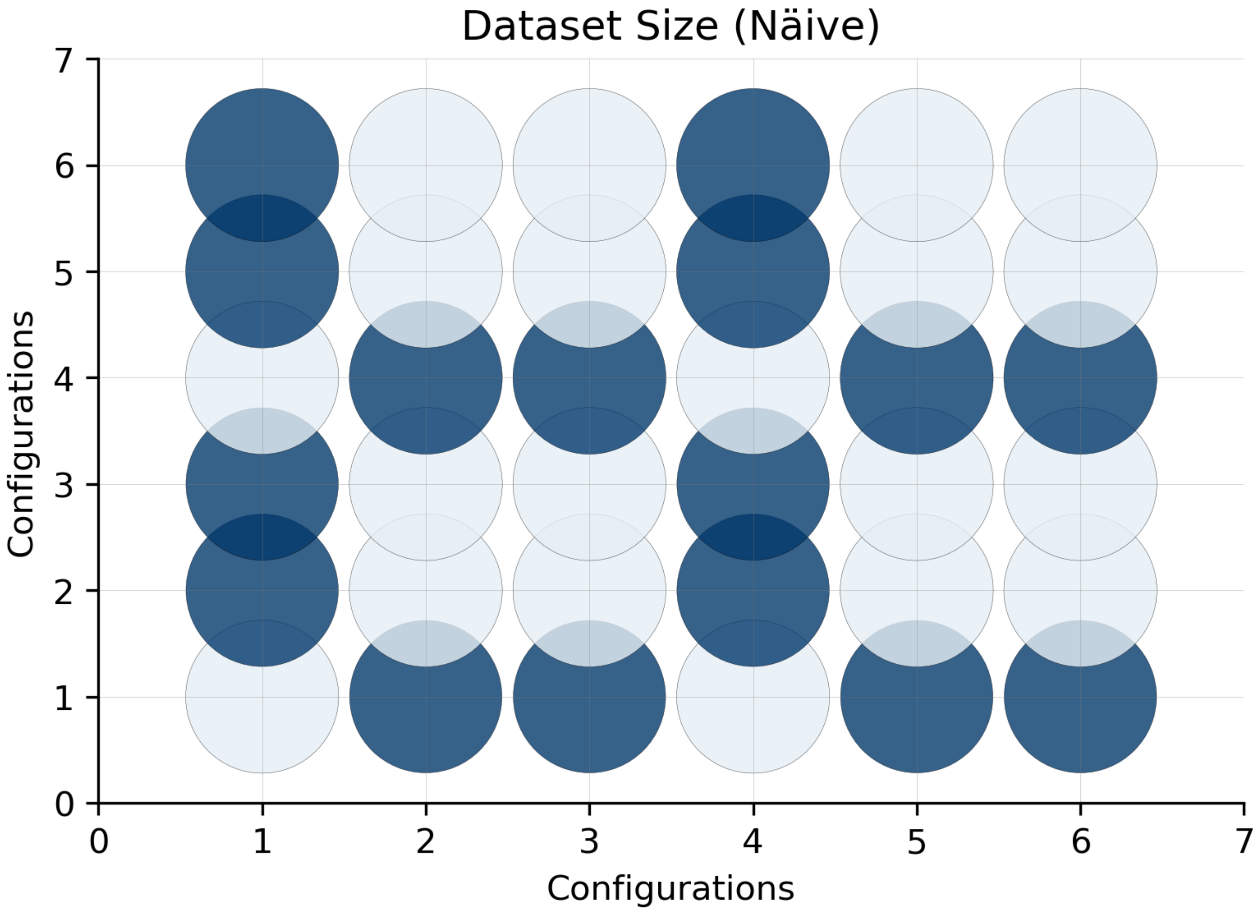

Pearson's
Correlation
1
0.5
0
-0.5
-1
Positive
correlation
Negative
correlation

Experimental setup

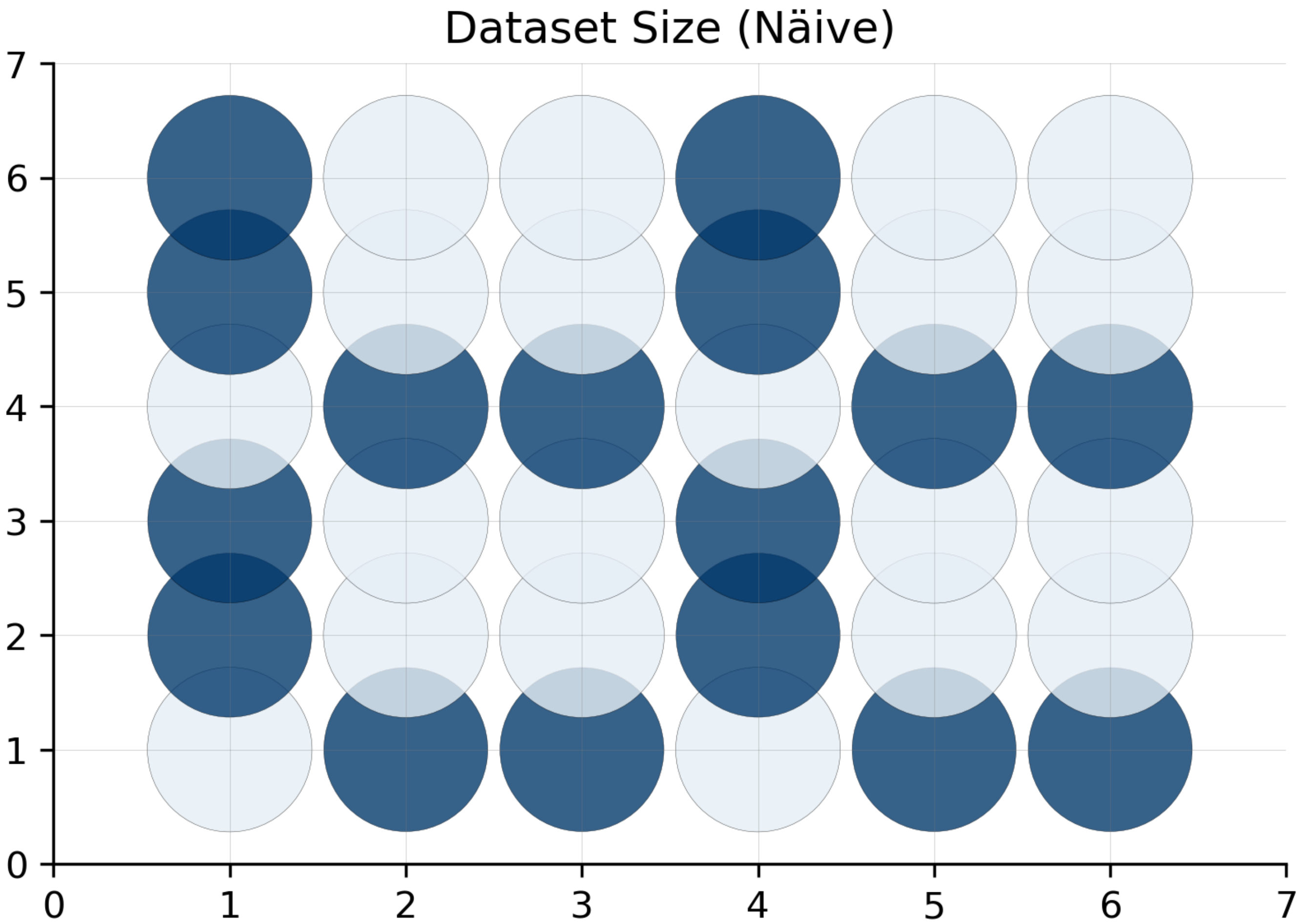


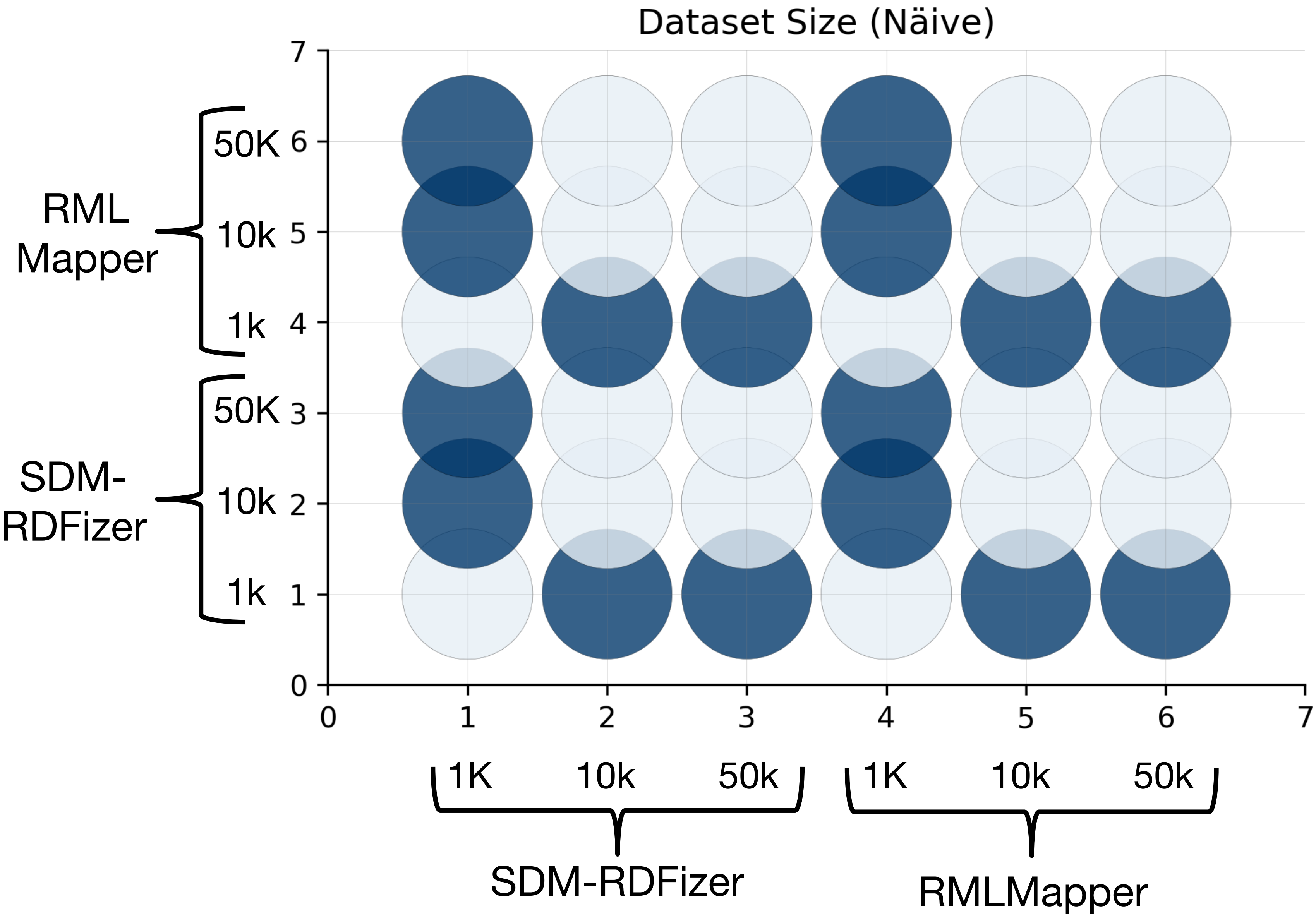
Color reflects correlation type

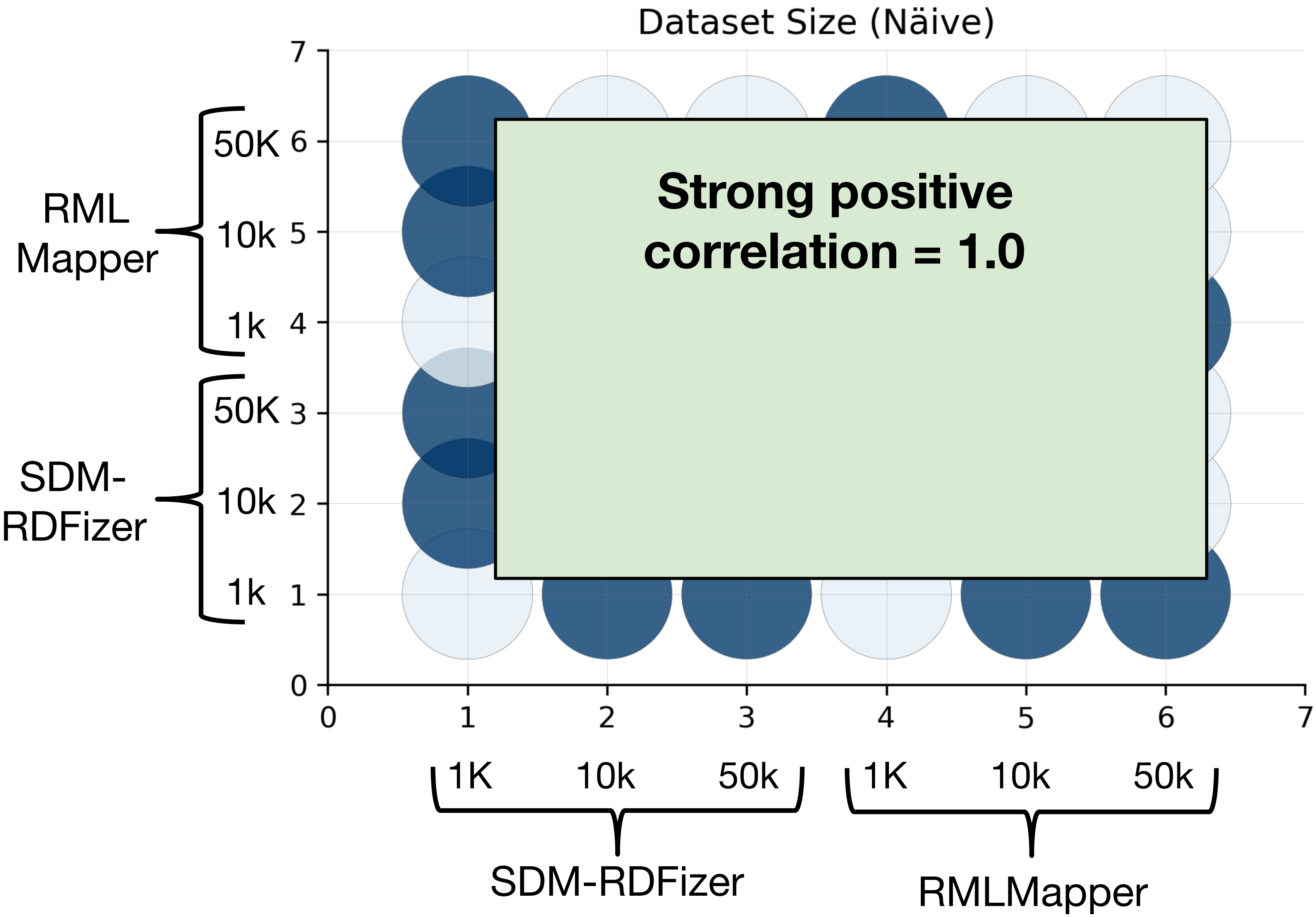
Size reflects correlation value



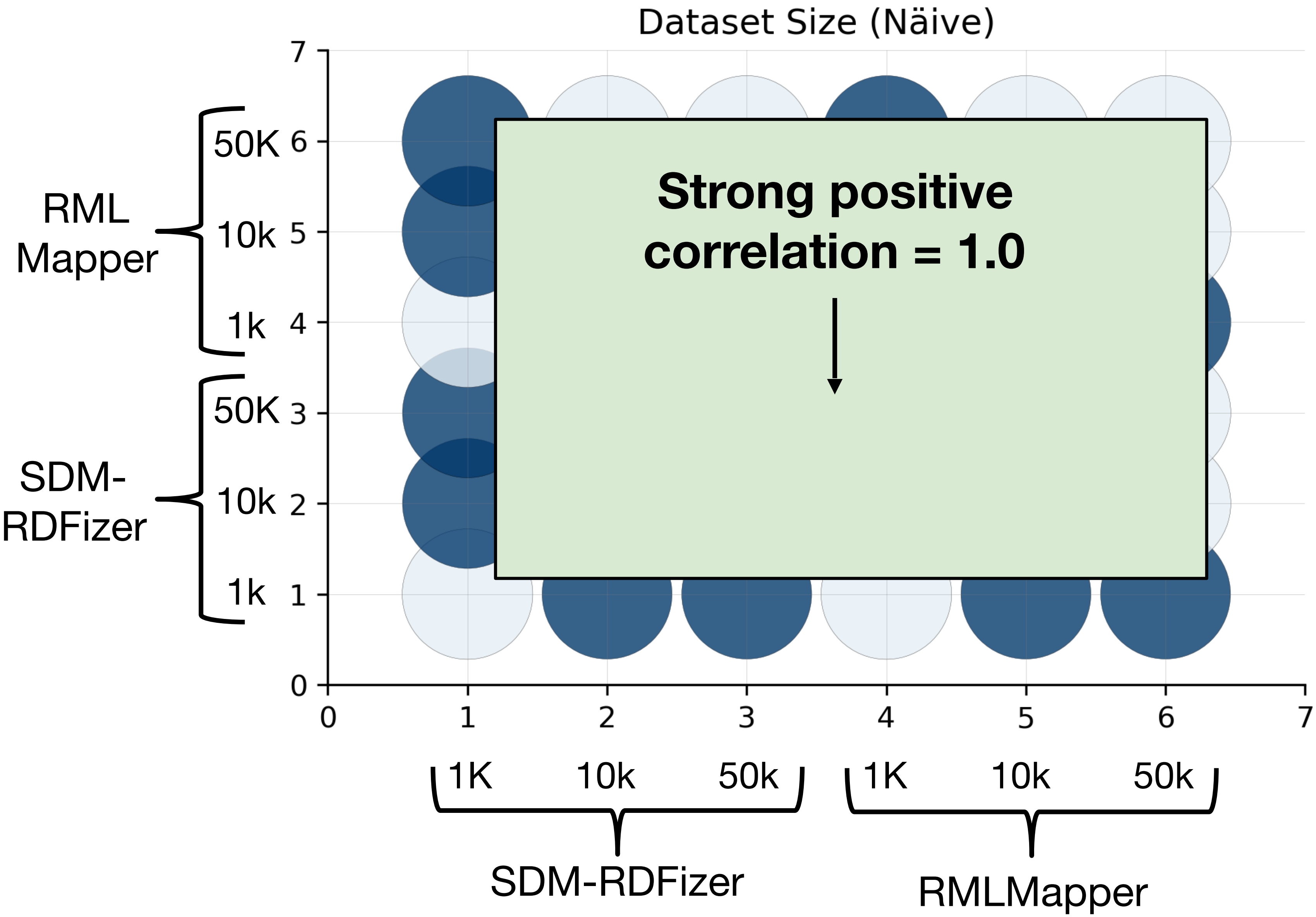
Configurations 1-3: SDM-RDFizer on datasets 1k, 10k, 50k and 30 POM
Configurations 4-6: RMLMapper on datasets 1k, 10k, 50k and 30 POM


Dataset Size (Näive)
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

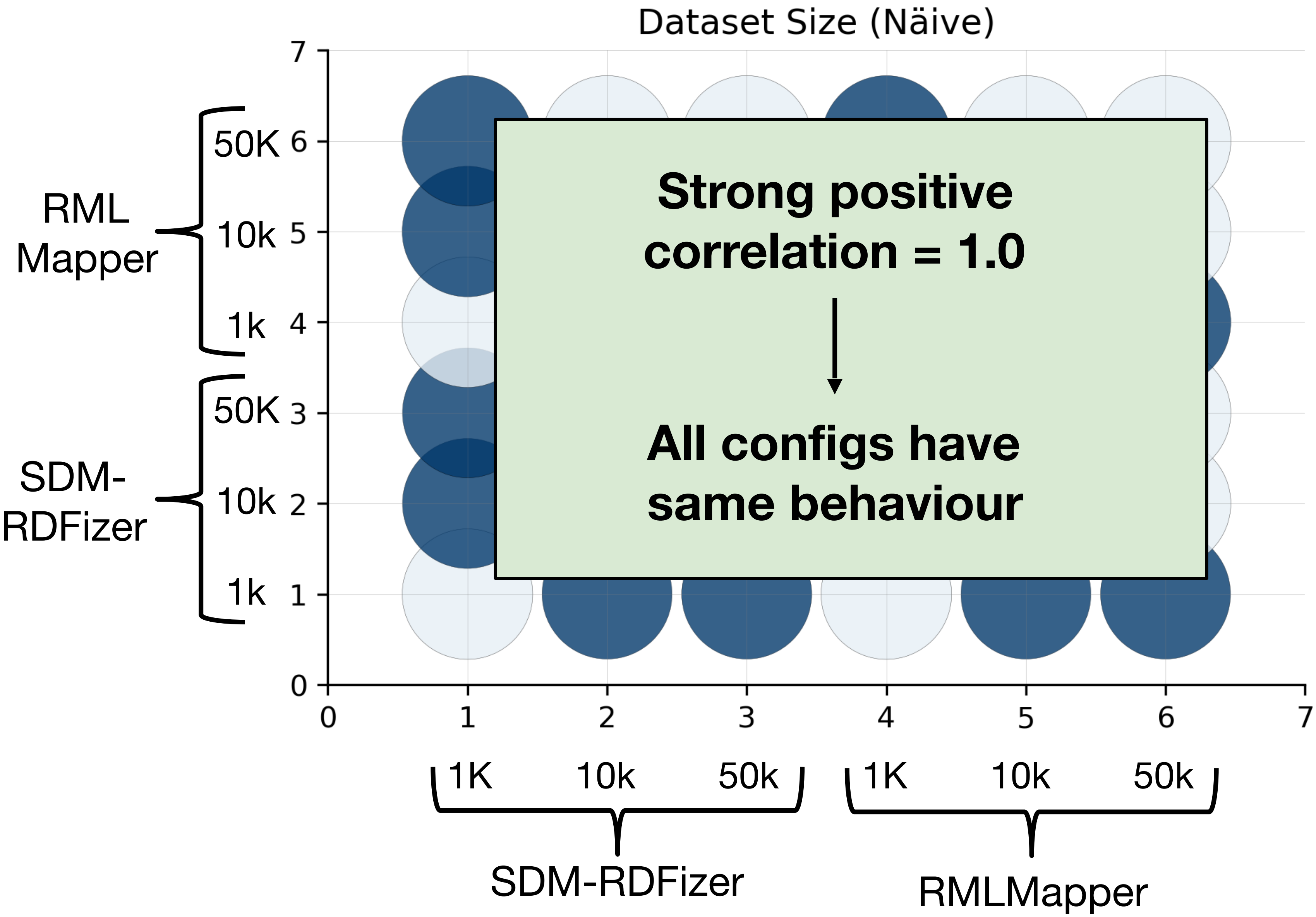

Dataset Size (Näive)
RML
Mapper
SDM-
RDFizer
50K
10k
1k
50K
10k
1k
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
1K
10k
50k
1K
10k
50k
SDM-RDFizer
RMLMapper



Strong positive correlation = 1.0



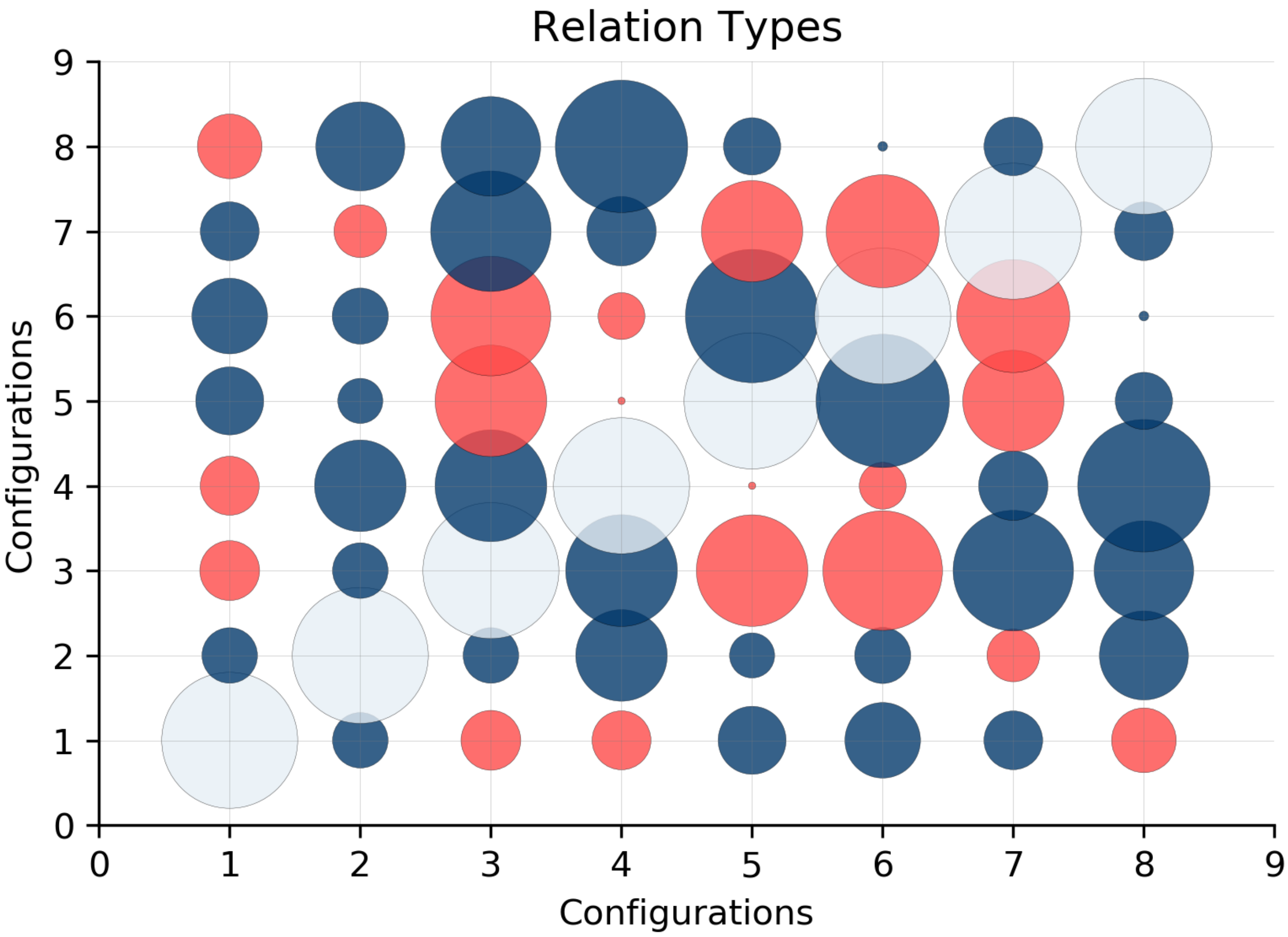
Strong positive correlation = 1.0



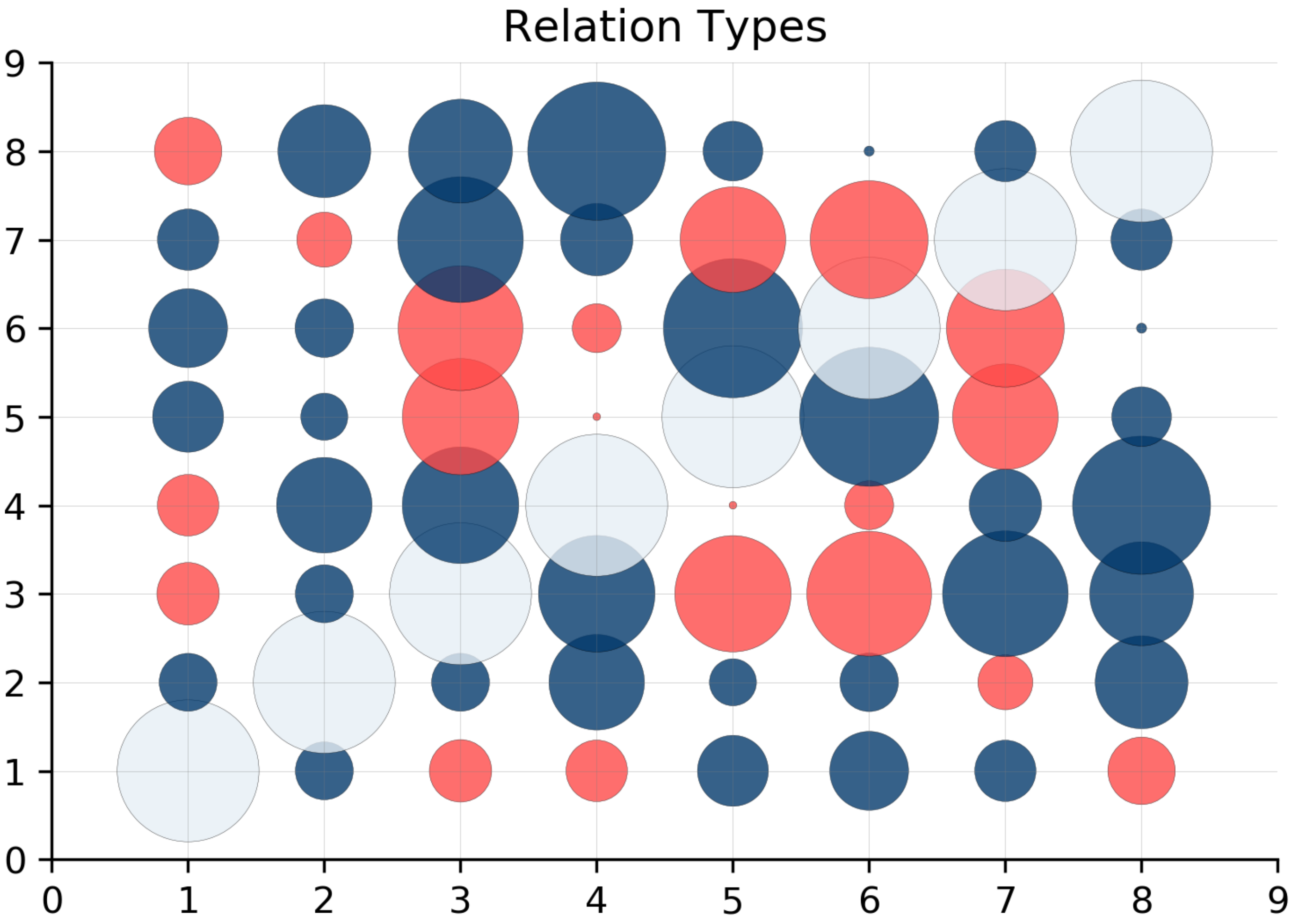
Strong positive correlation = 1.0

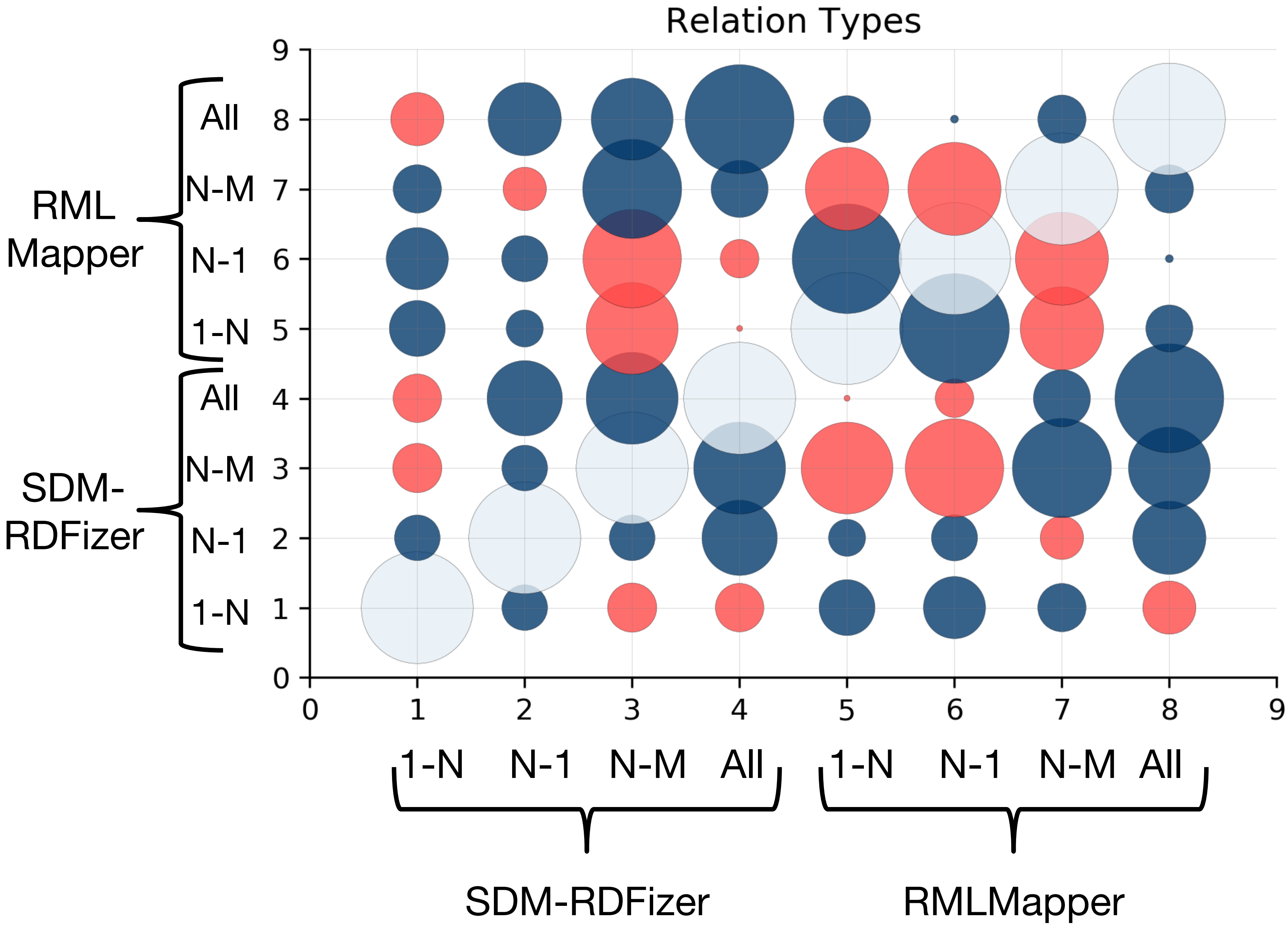
↓

All configs have same behaviour

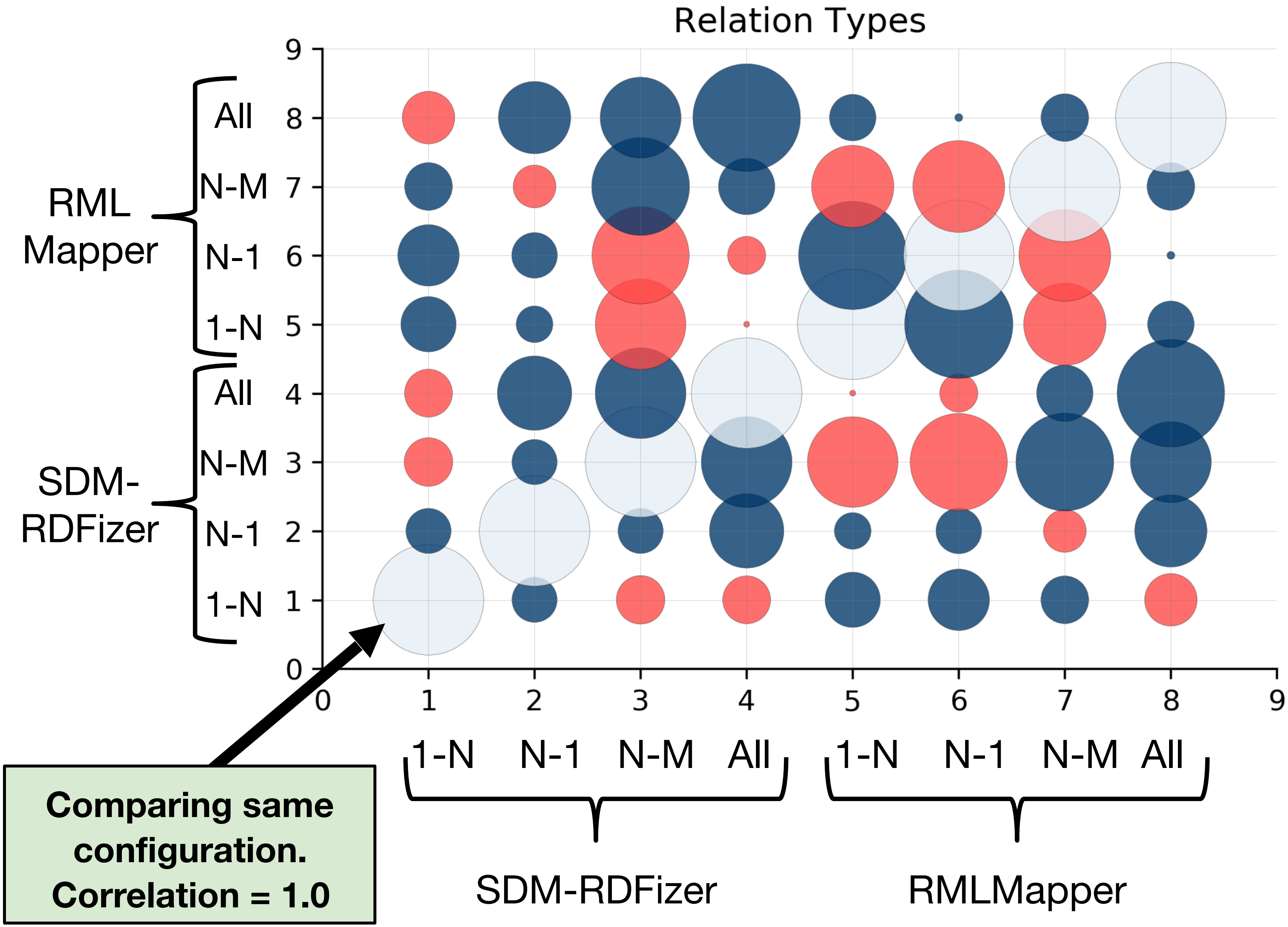


Configurations 1-4: SDM-RDFizer on 1-N, N-1, N-M and combination
Configurations 5-8: RMLMapper on 1-N, N-1, N-M and combination


Relation Types
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9

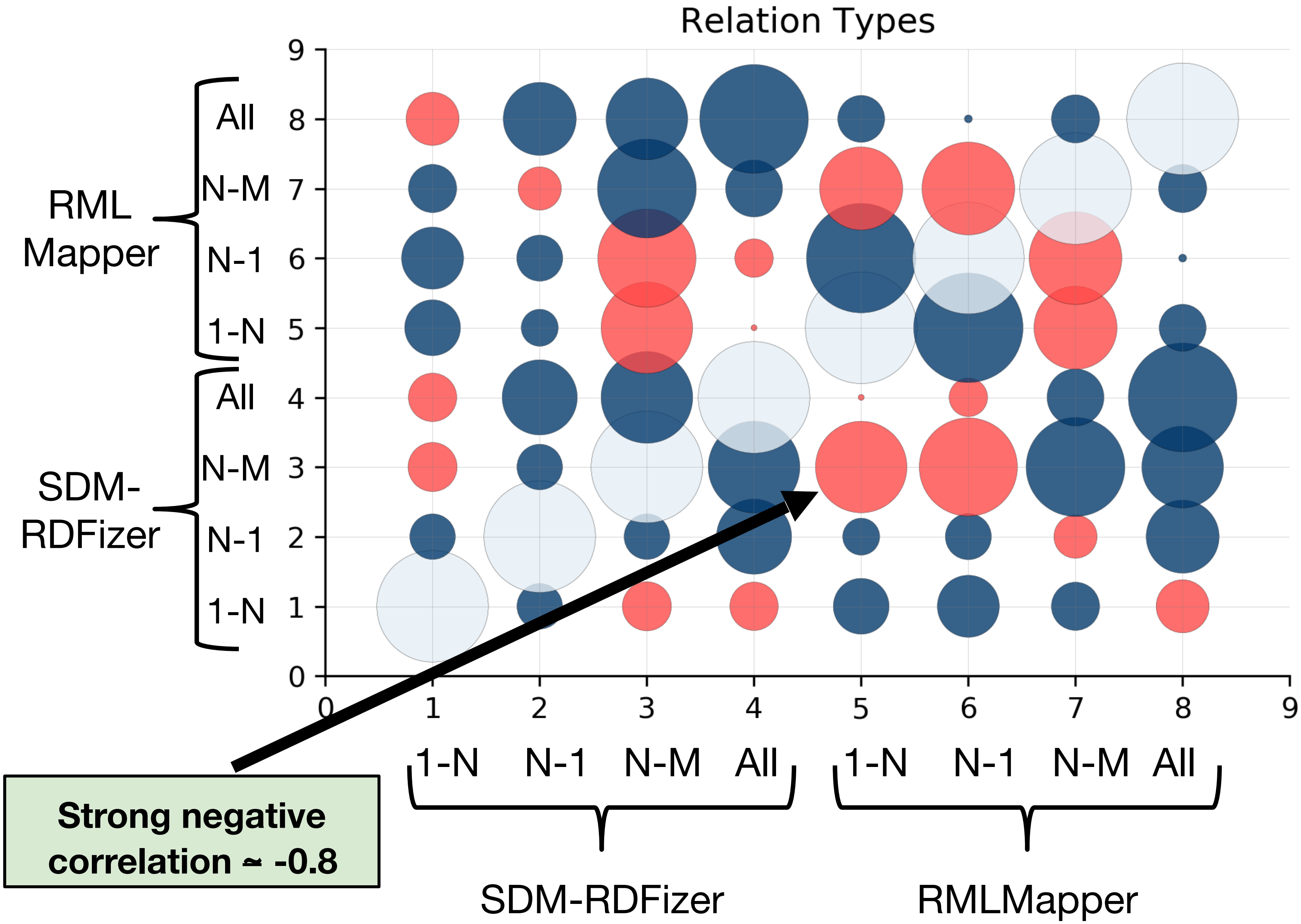

Relation Types
RML Mapper
All
N-M
N-1
1-N
SDM-RDFizer
All
N-M
N-1
1-N
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
1-N
N-1
N-M
All
1-N
N-1
N-M
All
SDM-RDFizer
RMLMapper

Results II: Relation Types 10K

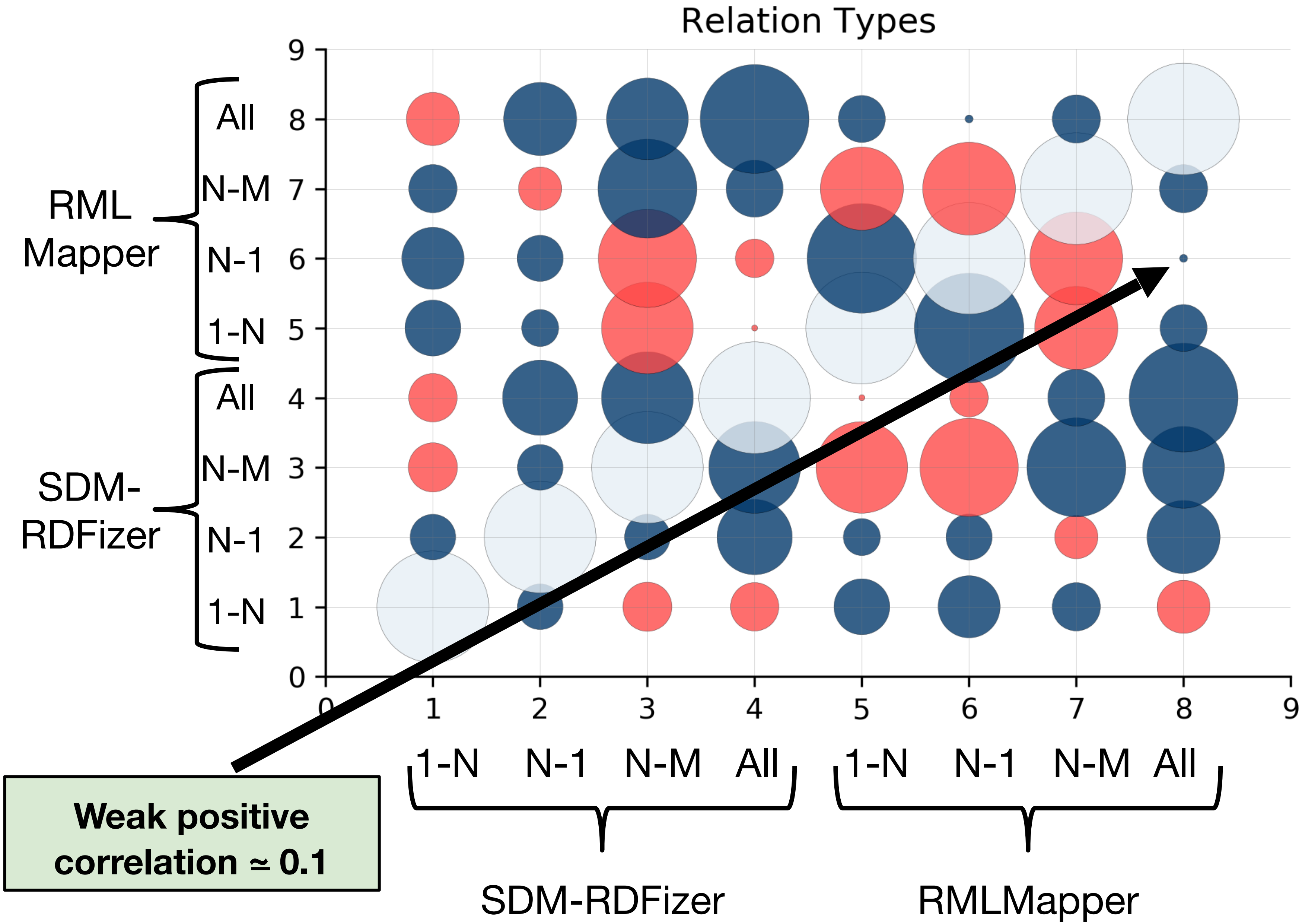


Comparing same configuration. Correlation = 1.0

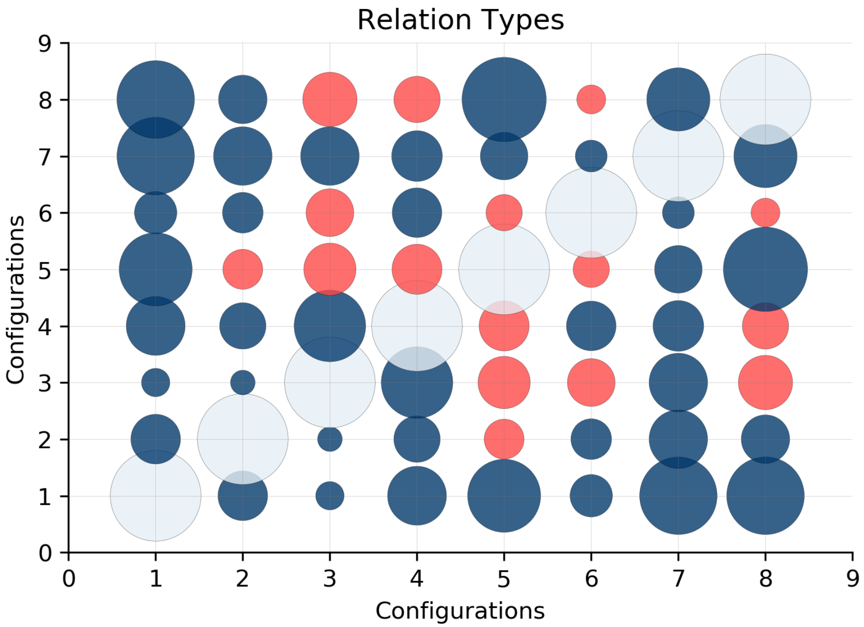
Results II: Relation Types 10K



Strong negative correlation ≃ -0.8

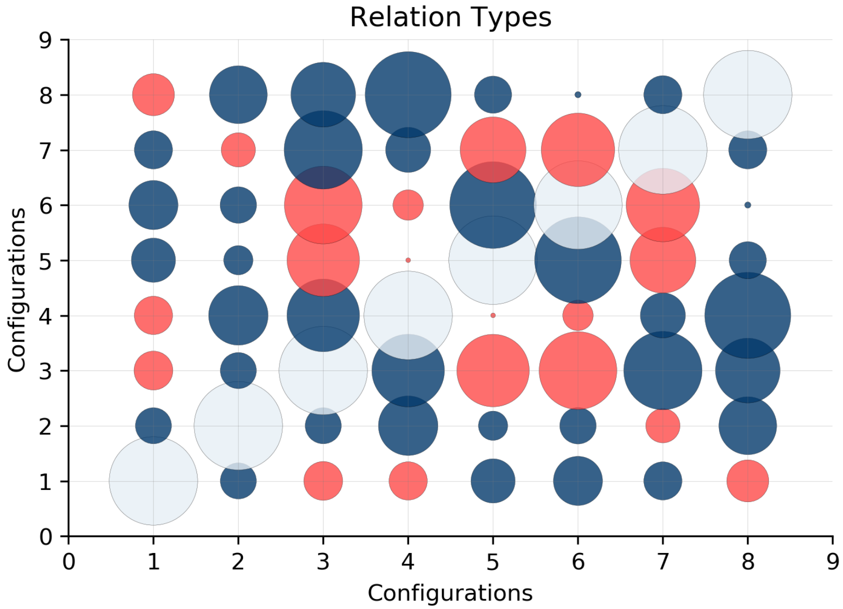


Weak positive correlation ≃ 0.1

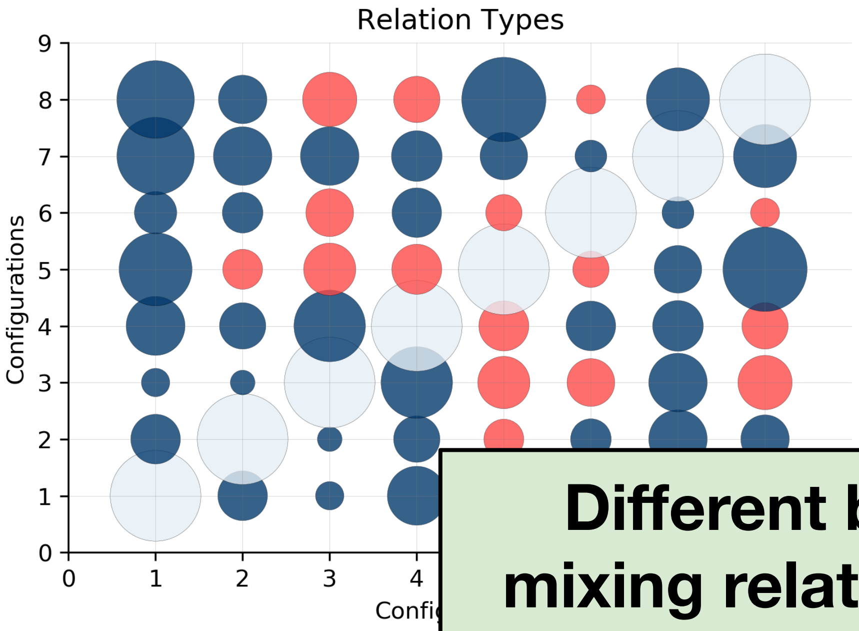


1K rows

10K rows



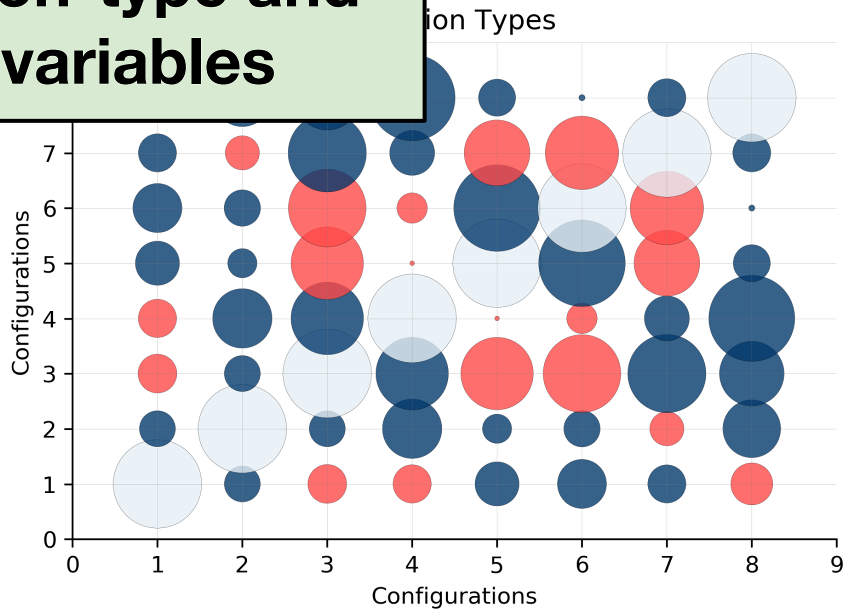
Mixing Parameters in one Testbed

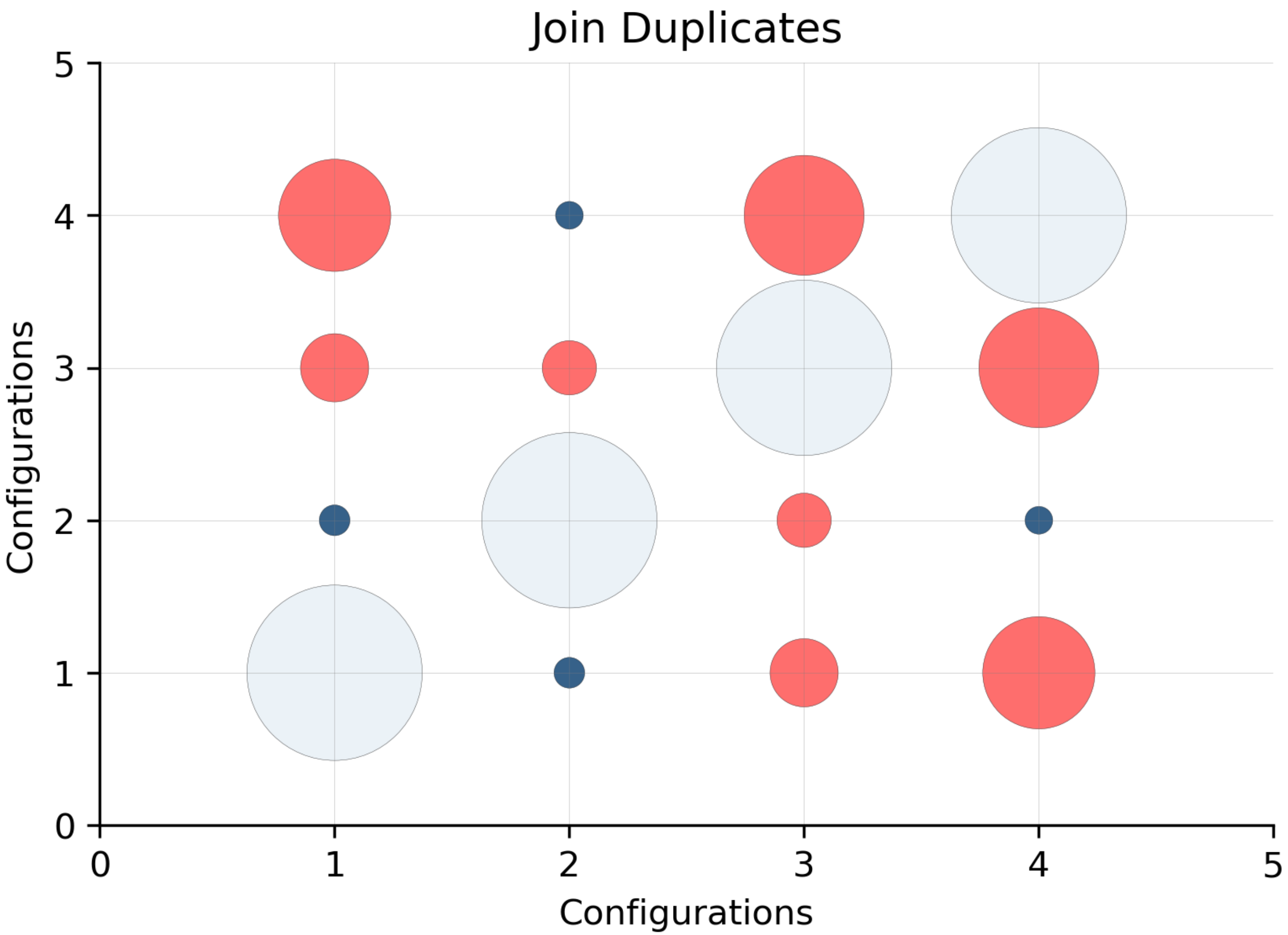


1K rows

Different behaviours mixing relation-type and data size variables

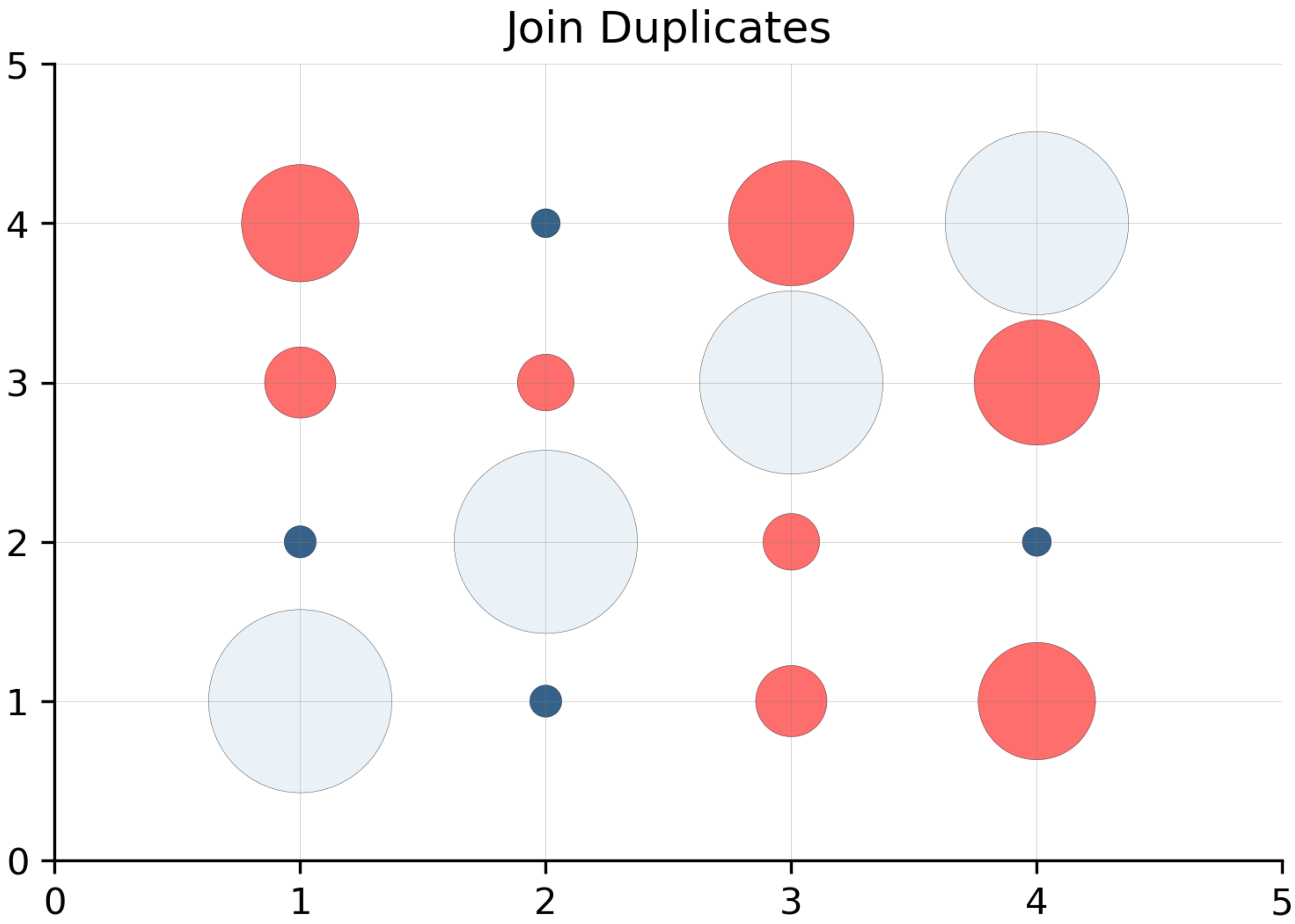
10K rows

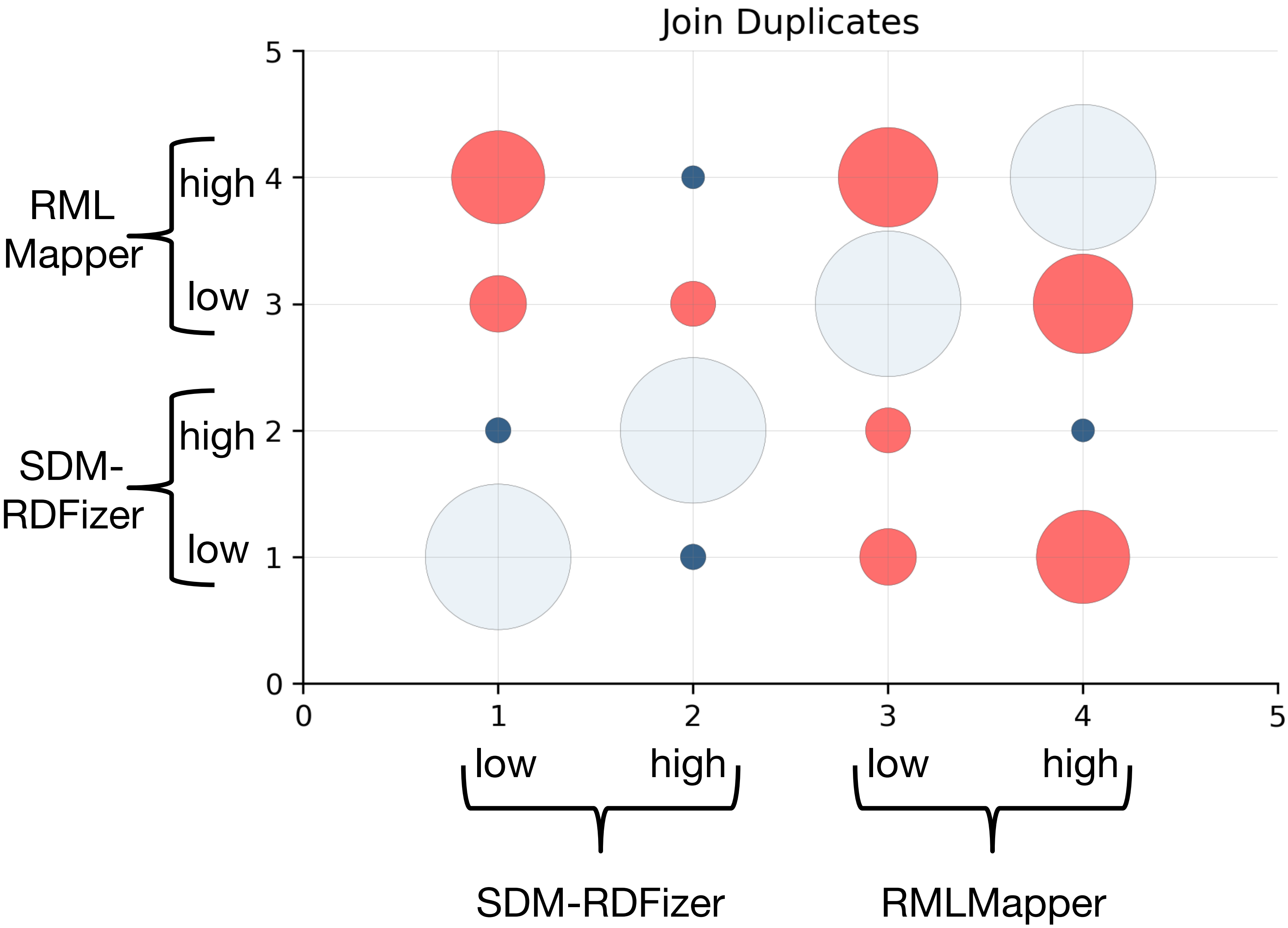


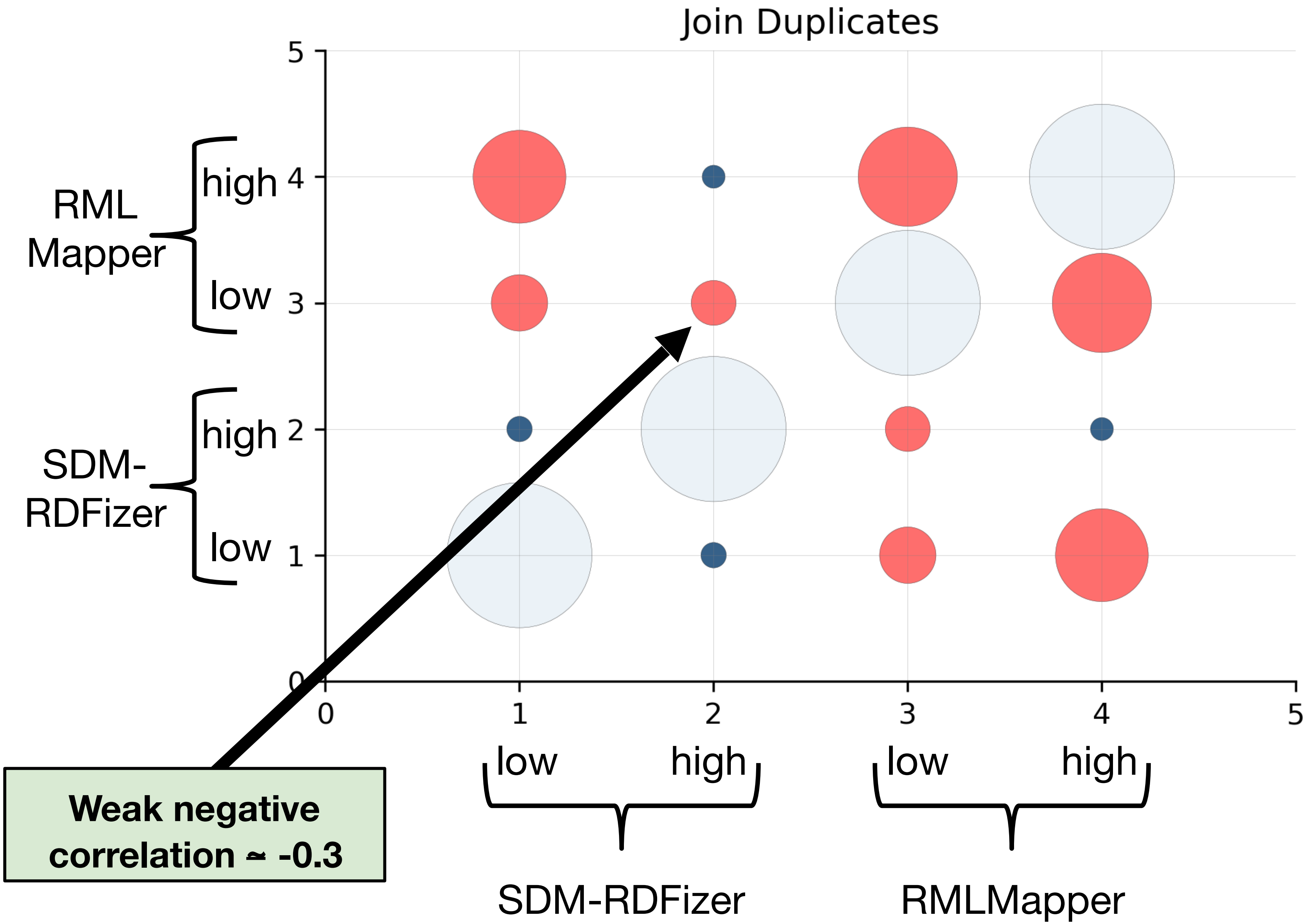


Configurations 1-2: SDM-RDFizer on low and high duplicates
Configurations 3-4: RMLMapper on low and high duplicates

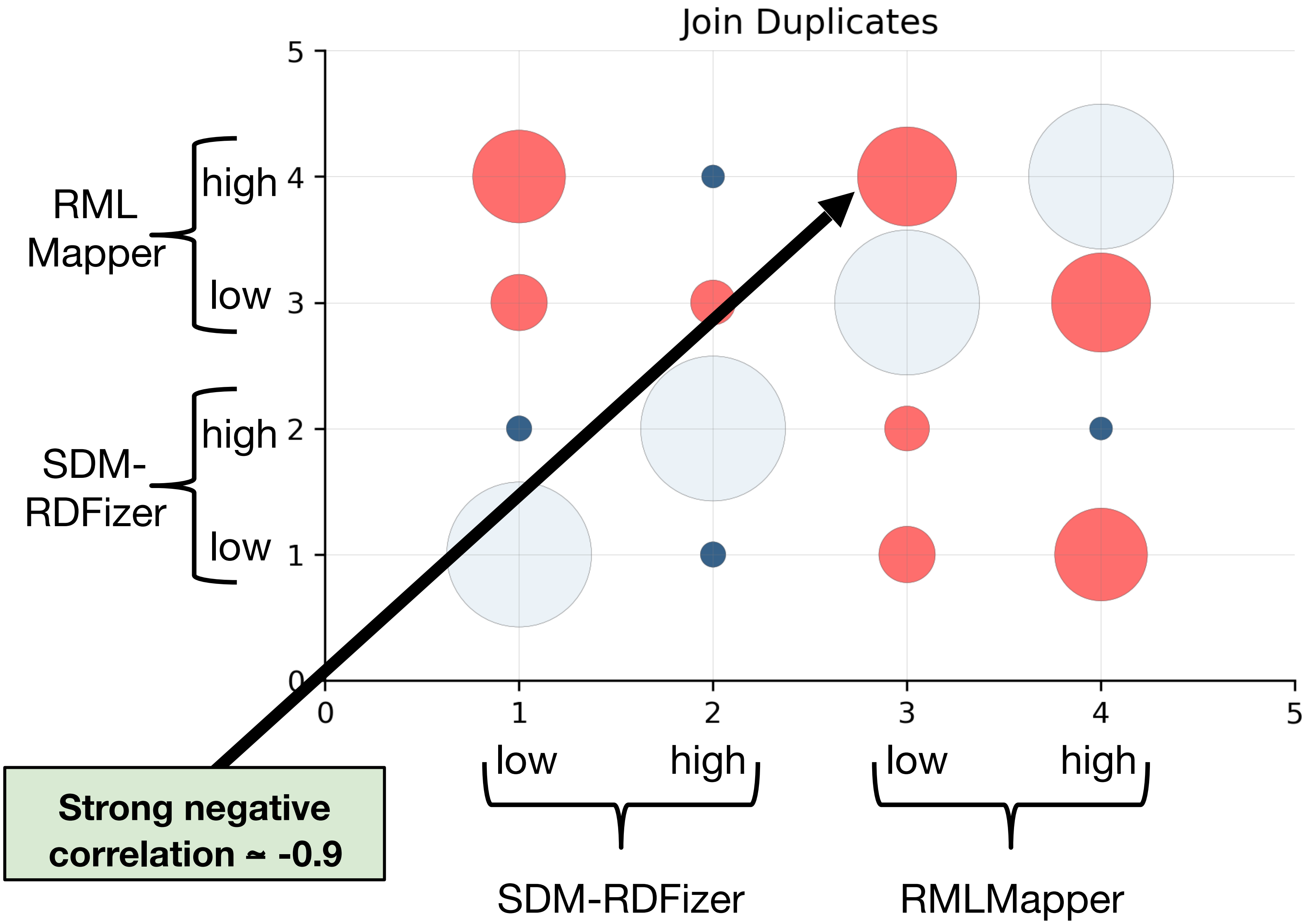
Results III: Duplicates on Join 1K




Join Duplicates
RML
Mapper
high
low
SDM-
RDFizer
high
low
0
1
2
3
4
5
0
1
2
3
4
5
low
high
low
high
SDM-RDFizer
RMLMapper

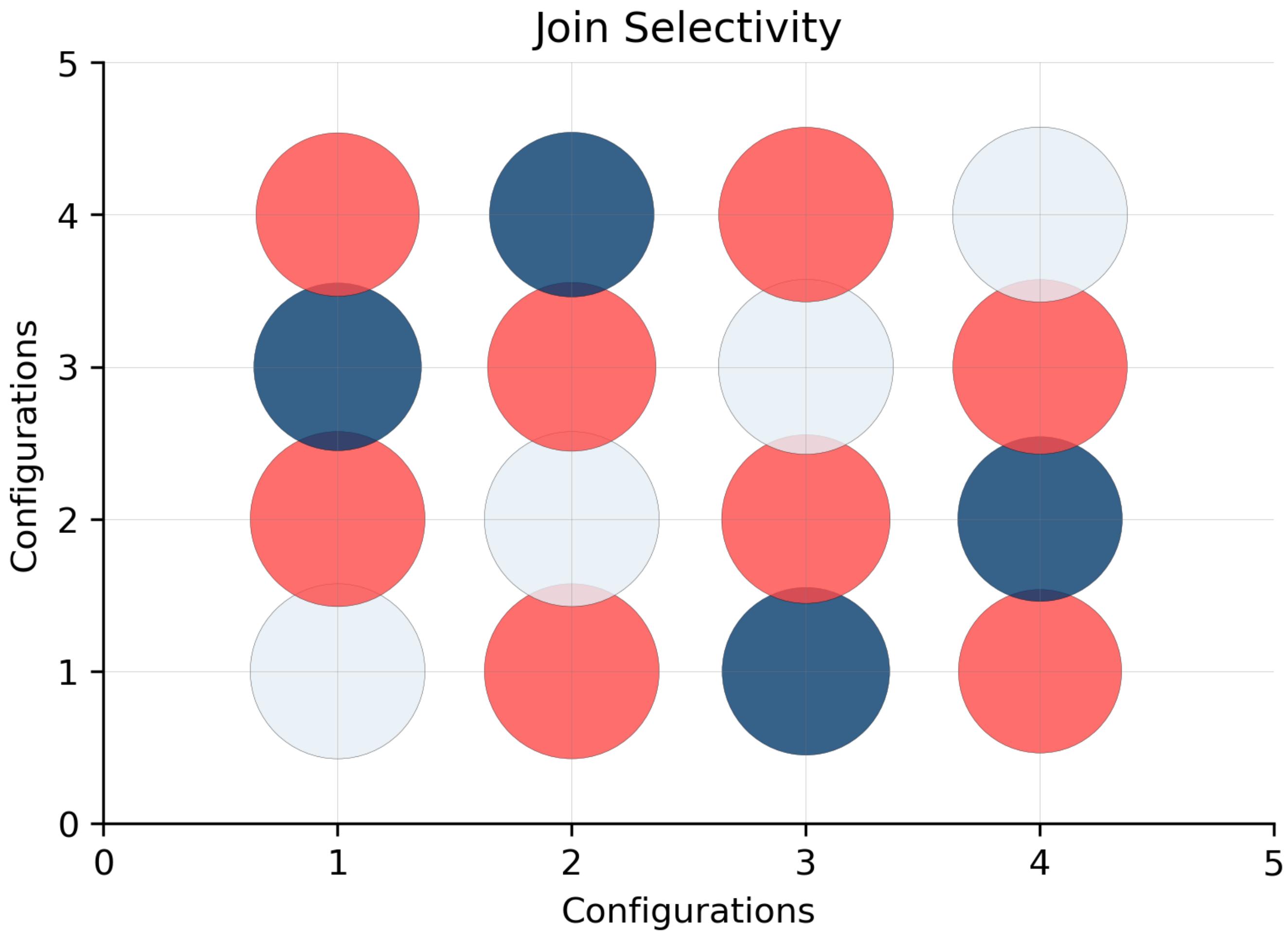


Weak negative correlation ≃ -0.3

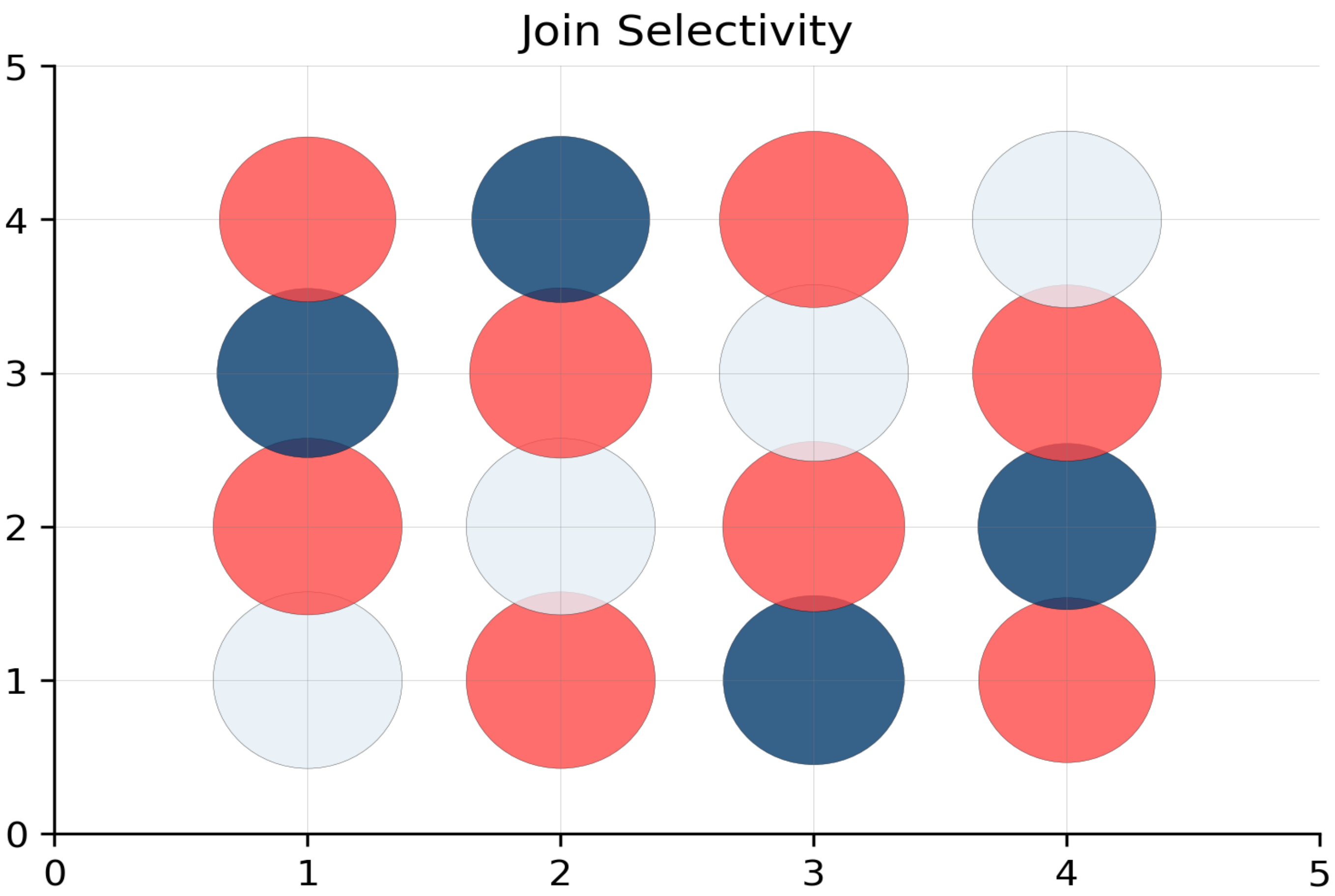


Strong negative correlation ≃ -0.9

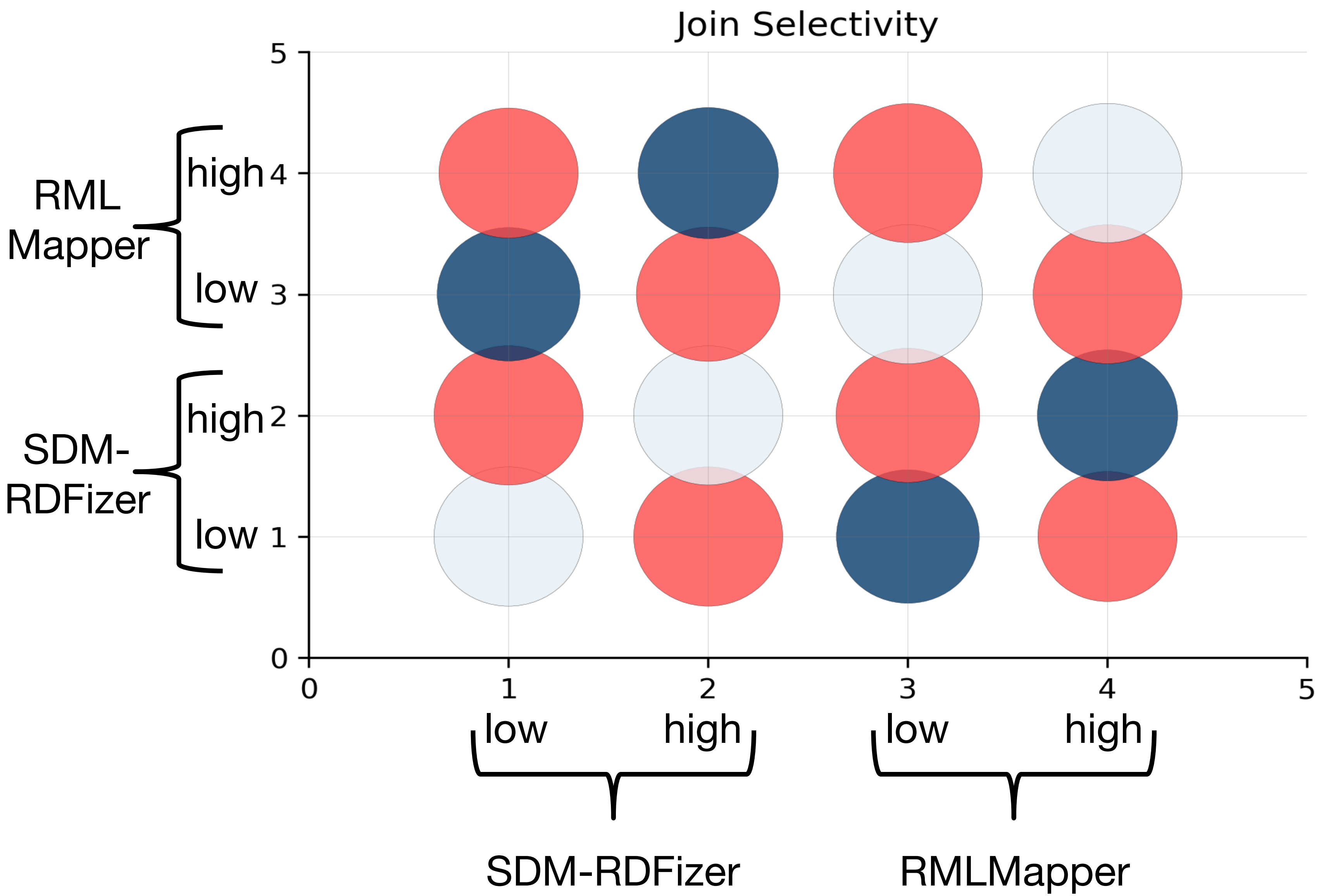
Results IV: Join Selectivity on 1K

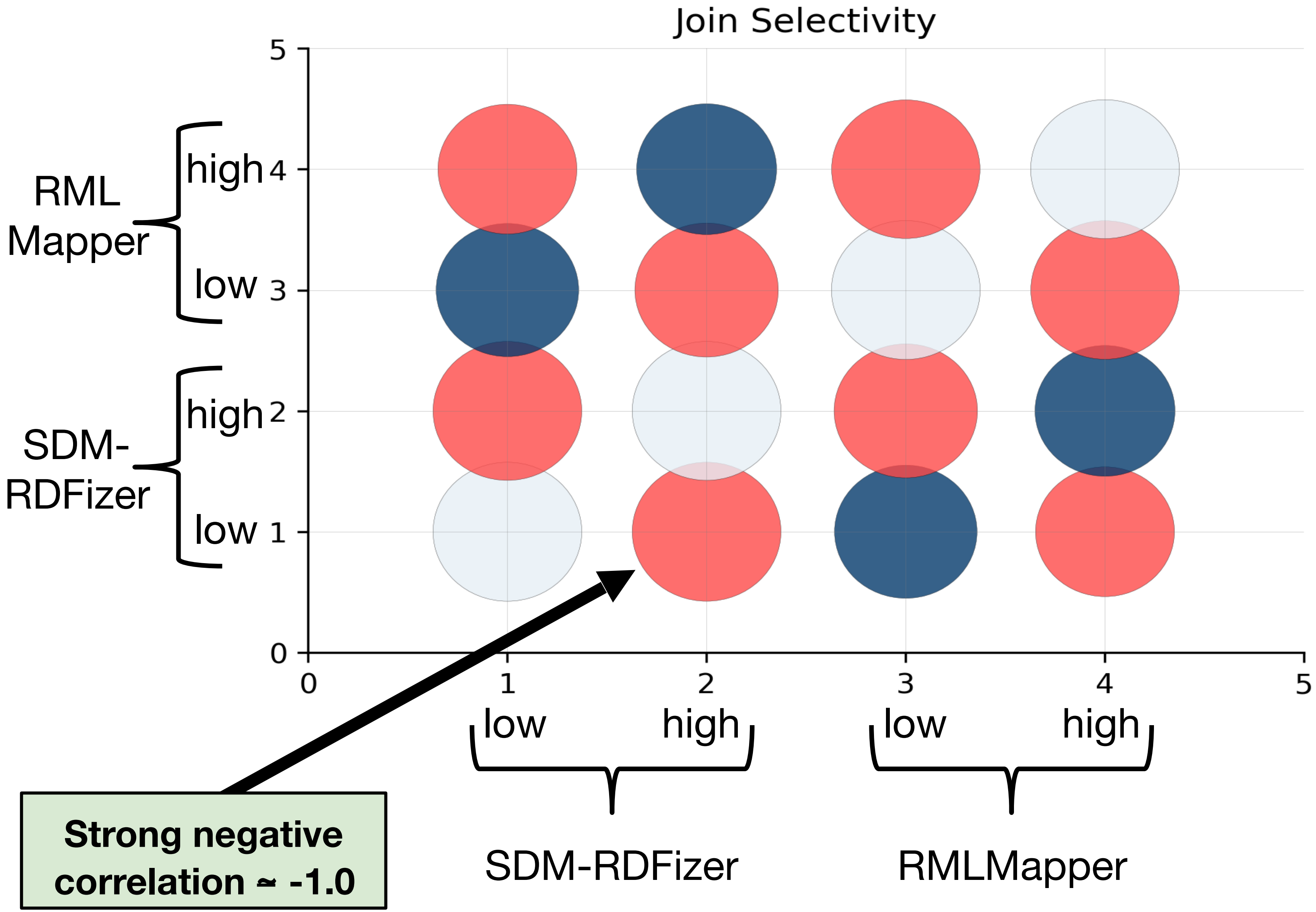


Configurations 1-2: SDM-RDFizer on low and high selectivity
Configurations 3-4: RMLMapper on low and high selectivity

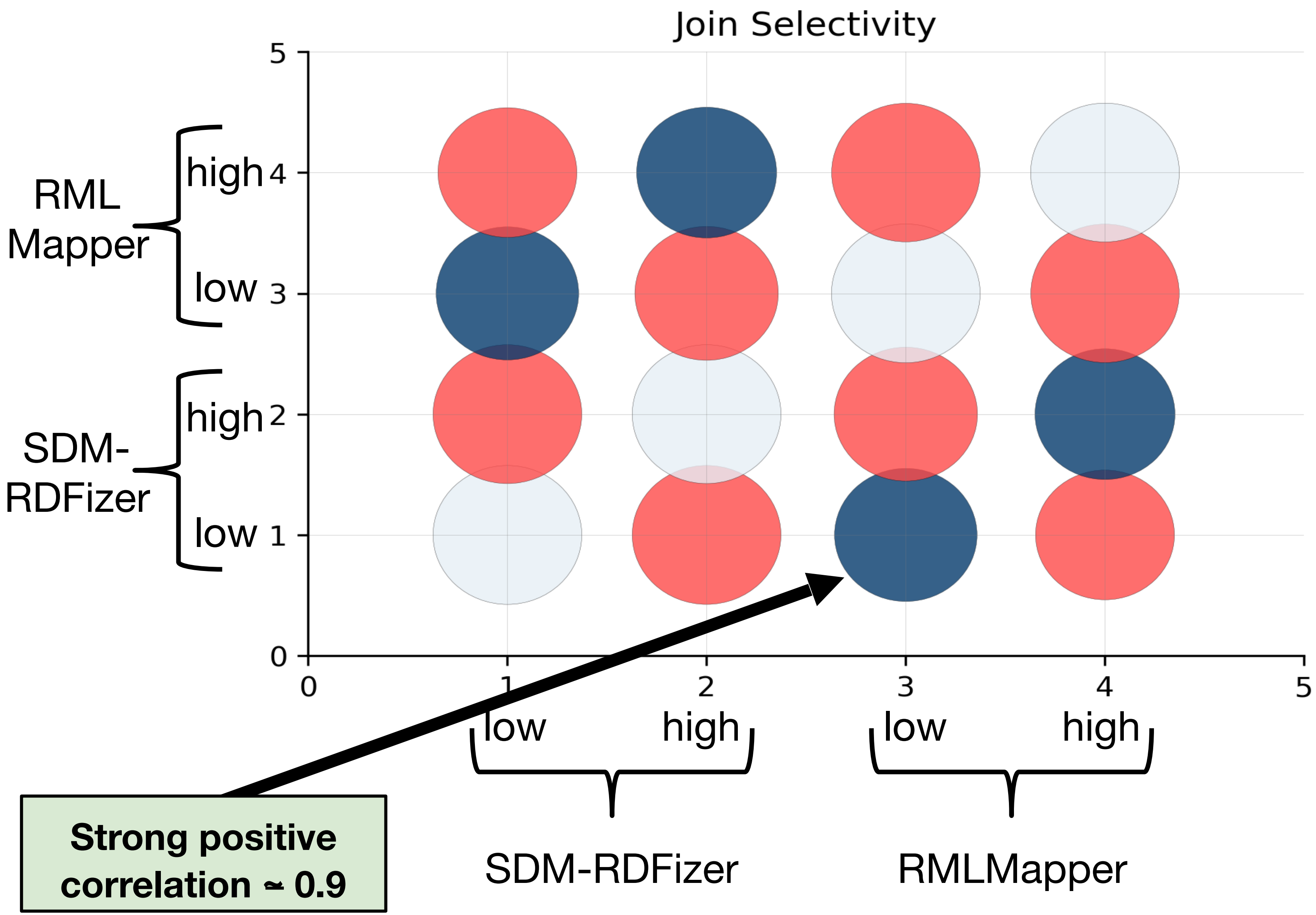

Join Selectivity
0
1
2
3
4
5
0
1
2
3
4
5

Results IV: Join Selectivity on 1K





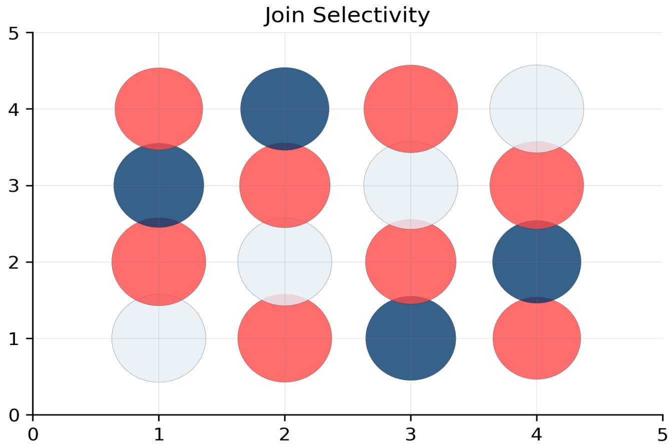
Strong negative correlation ≃ -1.0



Strong positive correlation ≃ 0.9

Conclusions:

- We studied **different parameters and variables** that may affect the **behaviour** of knowledge graph creation engines
- **Empirical evaluation** of knowledge graph creation engines considering the studied parameters:
  - Discover hidden patterns in their behaviours



Future work:

- Define general testbeds to analyse the behaviour of the engines
- Evaluate other tools (e.g. RocketRML) and mapping languages (e.g. R2RML)

What are the Parameters that Affect the Construction of a Knowledge Graph?

David Chaves-Fraga*, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain
Kemele M. Endris, L3S Research Center & TIB
Enrique Iglesias, University of Bonn
Oscar Corcho, OEG - UPM
Maria-Esther Vidal, L3S Research Center & TIB

***Work done during the research visit of David Chaves-Fraga to TIB and L3S**

✉ dchaves@fi.upm.es
@dchavesf

📅 22/10/2019
📍 ODBASE@2019 (Rhodes)