Big Data: From Theory to Systems

Wenfei Fan
Shenzhen Institute of Computing Sciences
University of Edinburgh
Beihang University

The 5 V's of Big Data

Big Data: Volume, Variety, Velocity, Veracity, Value

- ✓ Volume: The size of data grows rapidly and continuously
  - China generated 23.9 ZB business data in 2022. It is expected to reach 76.6 ZB in 2027
- ✓ Velocity: "You cannot afford to make decisions based on yesterday's data"
  - Healthcare, retail, financial services, cyber security, …
- ✓ Variety: Relational database D, transaction graph G
  - Can we write a query across D and G in SQL?
- ✓ Veracity: The most challenging issue among the 5V's
  - Real-life data is dirty: semantic inconsistencies, duplicates, stale data, missing links
- ✓ Value : Killer APPs?
  - What practical value can we get out of big data?

The study has raised as many questions as it has answered

The challenges introduced by digital economy

Smart City

- Fusion of data from various models (historical BIM/CIM; and newly collected data)
- Massive data from unreliable data sources
- Real-time analysis in response to updates

Digital Currency

- Heterogeneous queries on big data across different models
- Real-time transaction processing with consistency and reliability requirements
- Data-driven fraud detection and intelligent analysis

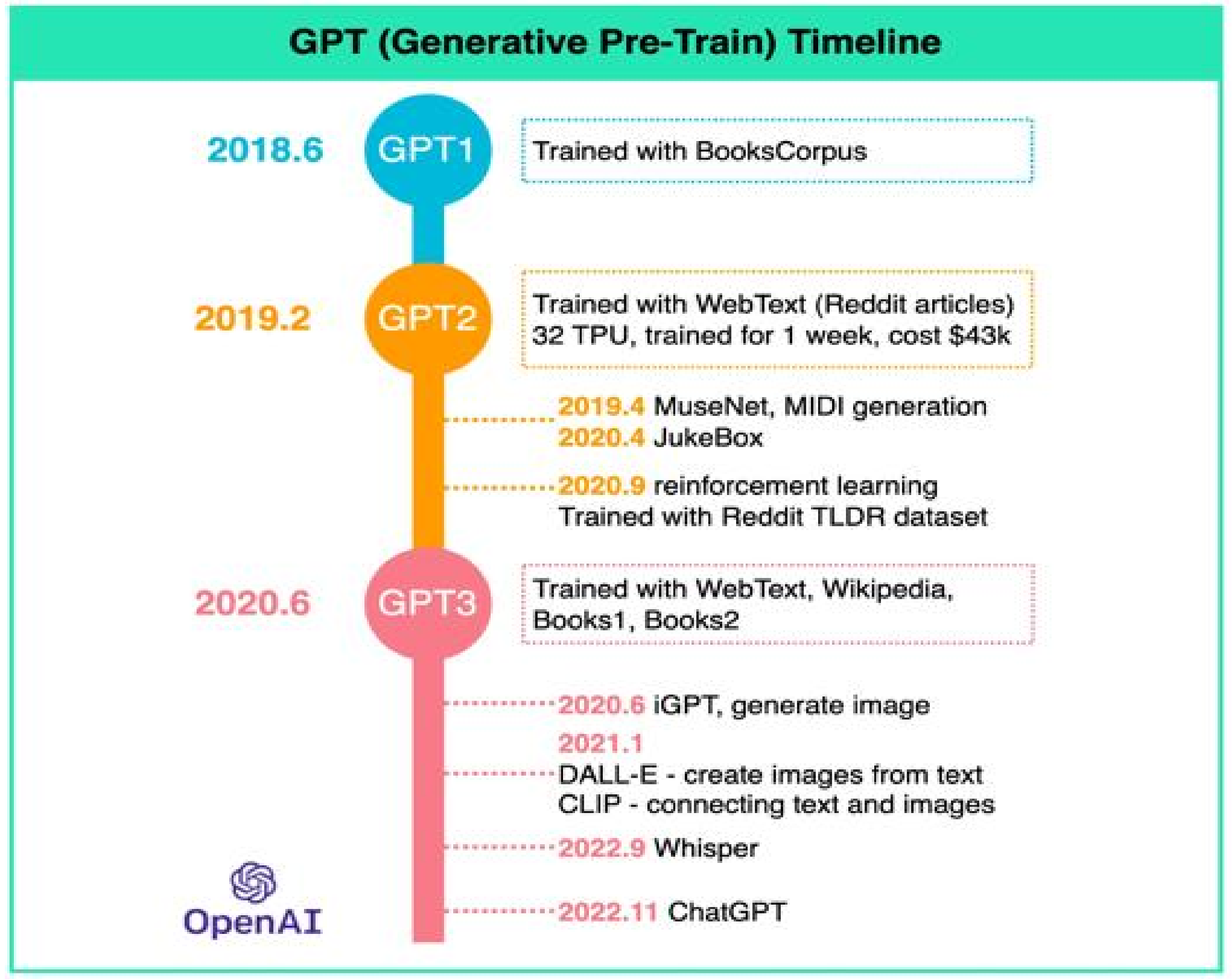
Challenges:

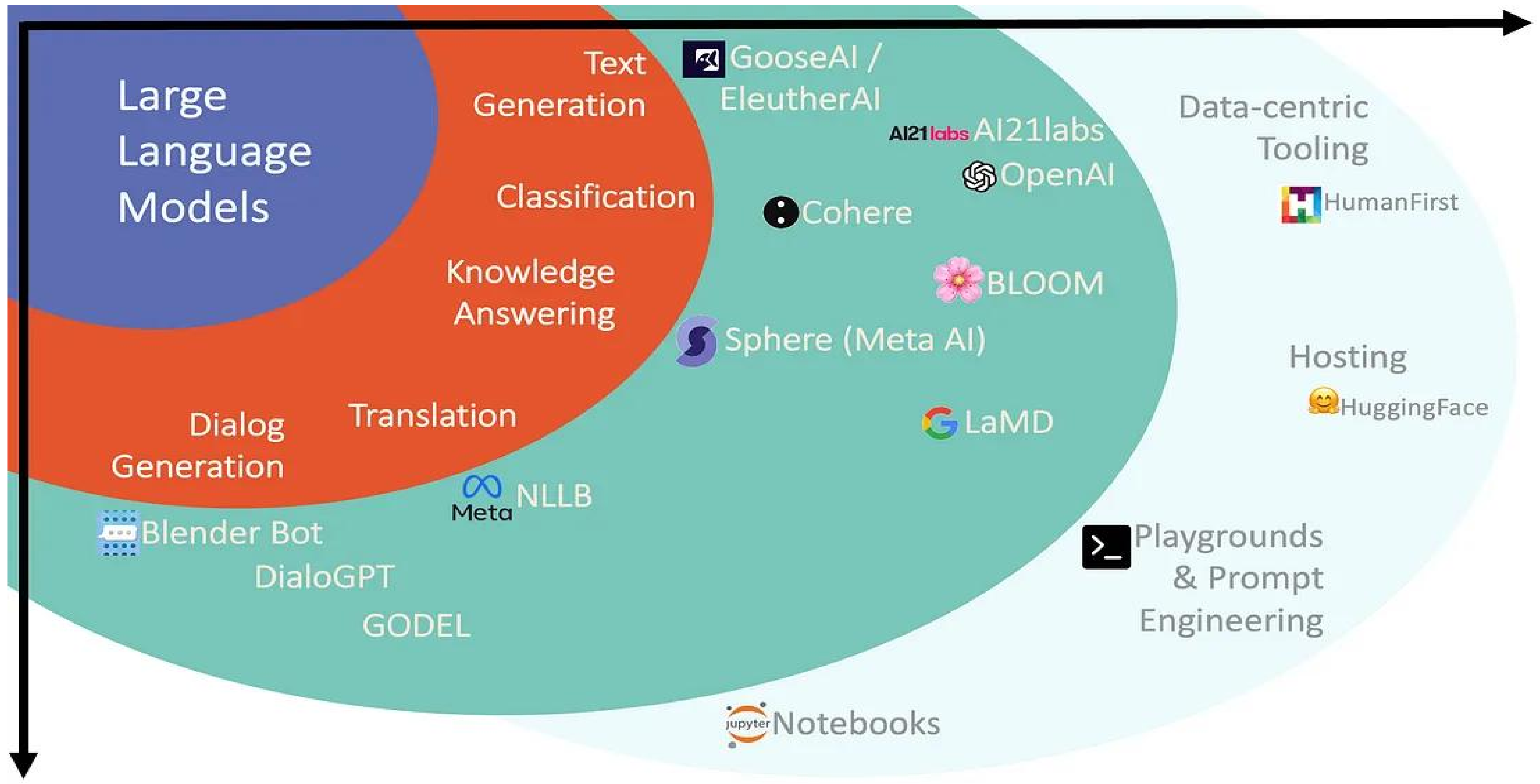
- ✓ How to query big data with limited resources? Volume
- ✓ How to answer queries across heterogeneous data models? Variety
- ✓ How to query dynamic data in response to updates? Velocity
- ✓ How to clean dirty data? Veracity
- ✓ What is benefit of big data analytics? Value

The need for both theory and systems for big data analytics

The challenges introduced by AIGC

- ChatGPT has led to a large number of AIGC startups
- 73% startups in China focus on application domains, and 14% on LLMs.
- Most LLMs are developed via fine-tuning of open-source pre-trained models.





The next step: LLMs for specific application domains. But

✓ Where can we get high-quality data in a specific domain for LLM training?

✓ How can we make LLMs accurate, fair and robust?

✓ Can we interprete ML predictions after all?

To make practical use of AIGC

The systems developed at SICS

✓ Shenzhen Institute of Computing Sciences

- 500+ people, 87% are experienced engineers
- 3 systems and 5 products since 2019
- 95+ papers in TODS, VLDBJ, SIGMOD, VLDB, ICDE, etc;
  60% of the techniques proposed in the papers have been implemented in the systems

Rock: Data quality

Yashan DB: HTAP DBMS

Fishing Fort: Graph analytics

✓ Products: MedHunter, Mirror, Dream Creak, Lemmon Grass, Dasan Pass

An end-to-end solution to big data management

Volume: The solution of YashanDB

- ✓ *Theory: Bounded evaluation (BEAS)*
- ✓ *YashanDB: A database management system for hybrid workload*

The good, the bad and the ugly

- ✓ Traditional computational complexity theory of almost 60 years:
  - The good: polynomial time computable (tractable, PTIME)
  - The bad: NP-complete (intractable)
  - The ugly: PSPACE-hard, EXPTIME-hard, undecidable…

What happens when it comes to big data?

- ✓ Using an SSD of 12G/s, a linear scan of 15TB-dataset takes 20 minutes
- ✓ *O(n) time is already beyond reach on big data in practice!*

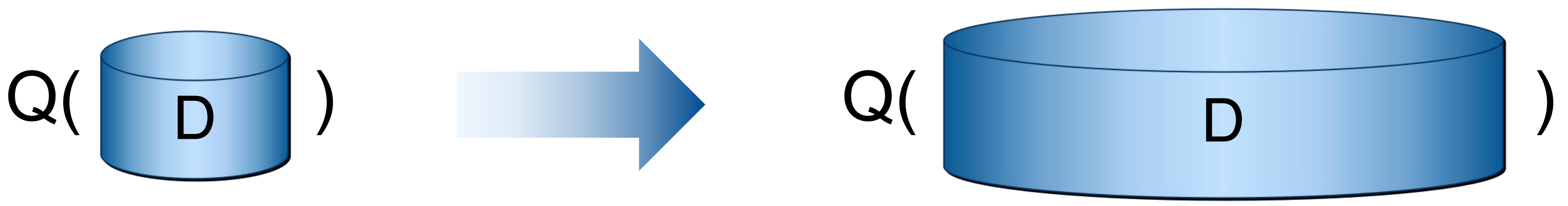
Polynomial time queries may become "intractable" on big data!

Big data: Through the eyes of computation

Computer science is the subject about

the computation of function f(x)

Big data: the data parameter *x* is large: PB or EB



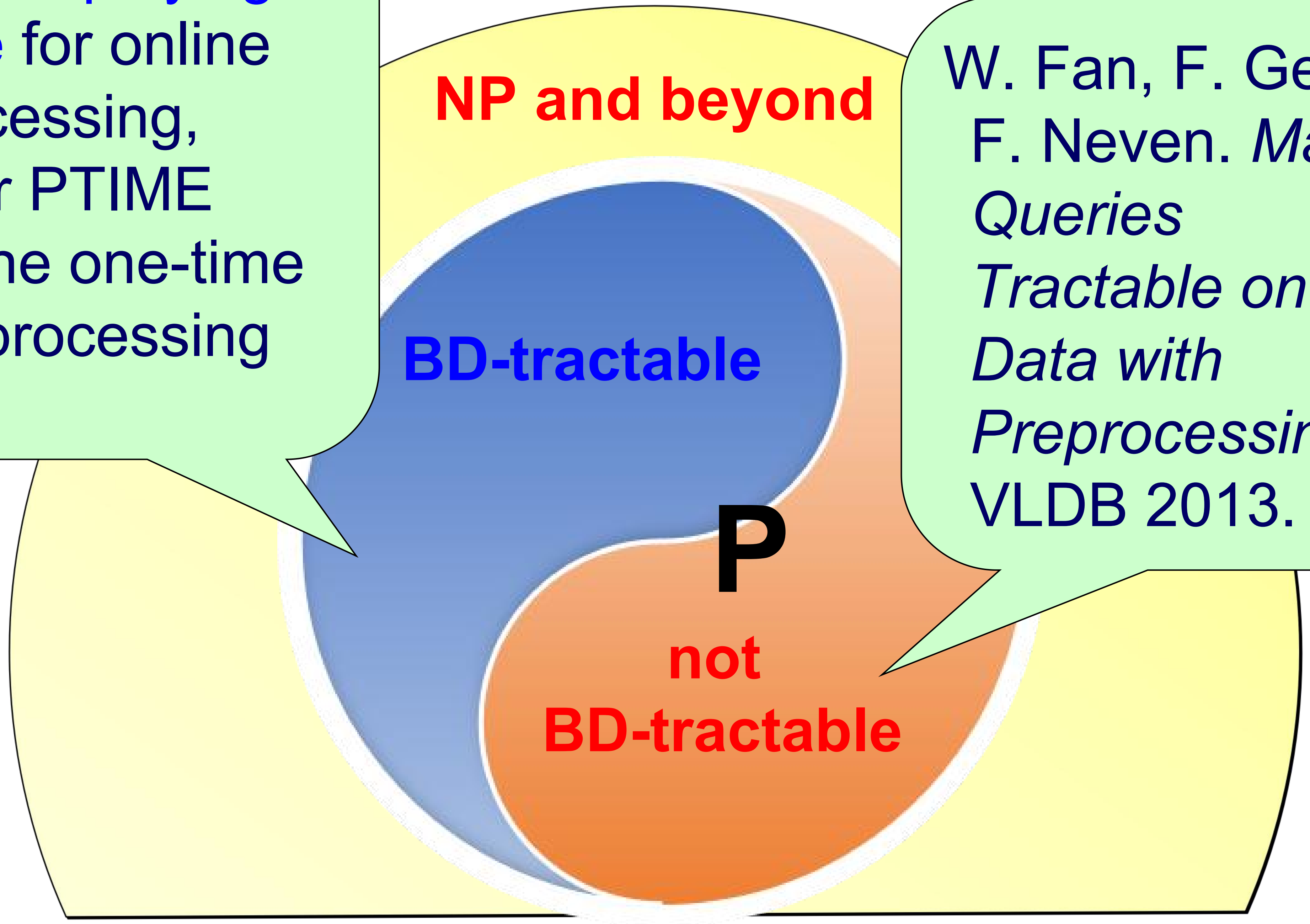
traditional database: GB (10^9B)

big data: PB or even EB (10^{15}B or 10^{18}B)

Fundamental challenges introduced by querying big data?

Tractability revisited for big data

Parallel polylog time for online processing, after PTIME offline one-time preprocessing



W. Fan, F. Geerts, F. Neven. *Making Queries Tractable on Big Data with Preprocessing*, VLDB 2013.

BD-tractable queries: properly contained in P unless P = NC

Open, like P = NP

A departure from classical theory and traditional techniques

Bounded evaluation: Make big data small

A Meta query

✓ Find me the prices of all cafes in NYC where my friend dined in May 2023

select c.price
from friend f, dine d, cafe c
where f.pid1 = p0 and f.pid2 = d.pid and d.mm = May and d.yy = 2023 and d.cid = c.cid and c.city = NYC

Traditional query: 1.4 days

c(price)

JOIN: ⋈

c(cid,city,price/FK)

JOIN: 2000m ⋈

SCAN: 20m

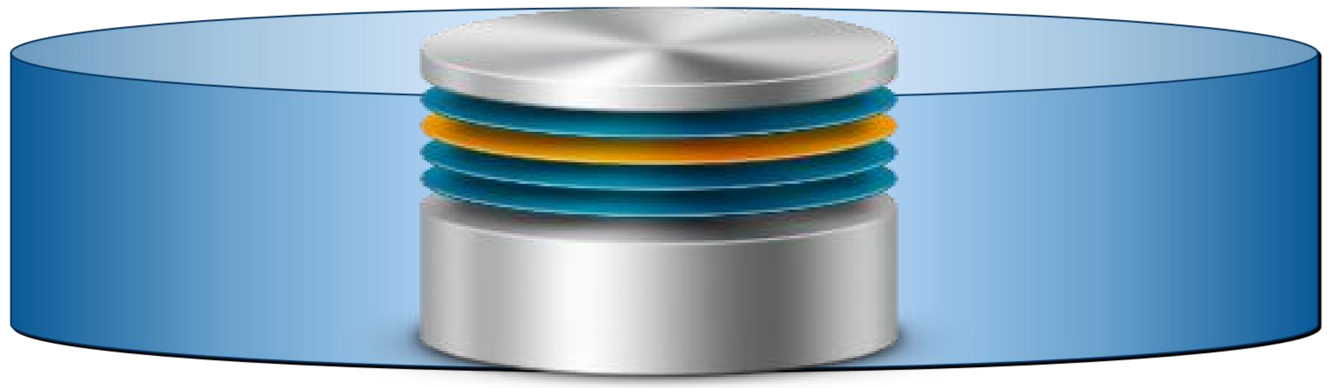
1.5PB

f(**p0**/idx)

d(pid,mm,yy,cid/idx)

100/15TB

15TB



x10⁵

Bounded evaluation: Fetch =<1s

PK（p0）→ Fetch → AC1(5k)

Up to 5K friends

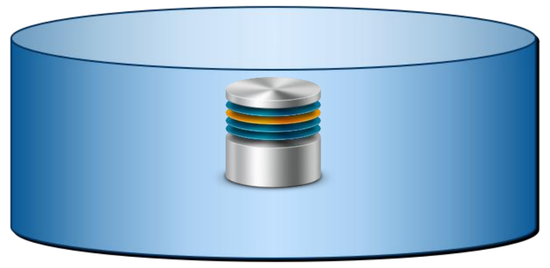
Fetch
At most 31 days

AC3(1*31*5k) ← Fetch ← AC2(31*5k)

One price for one id

Assume 15TB of friend and dine tables.

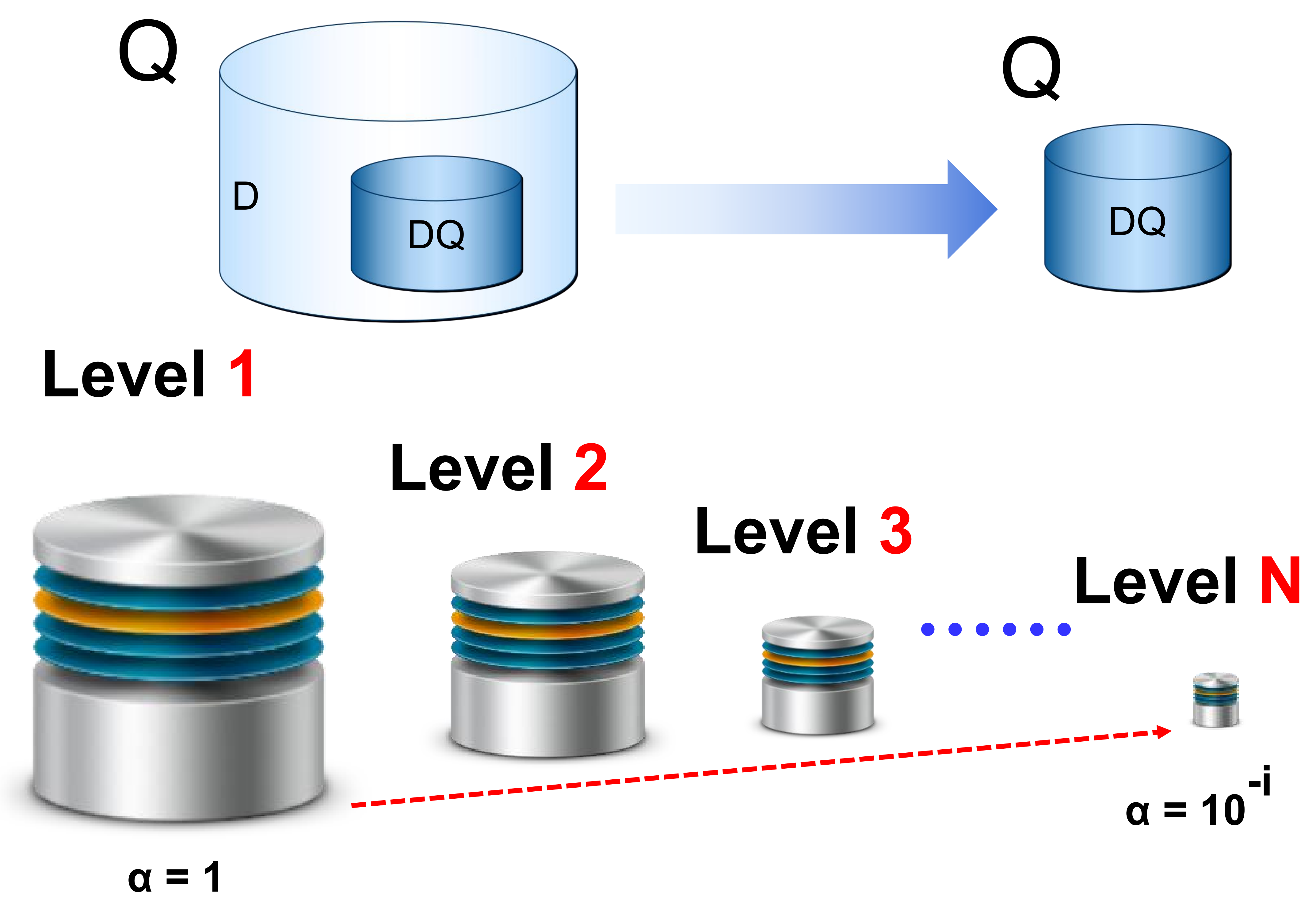
It is 300PB for Meta

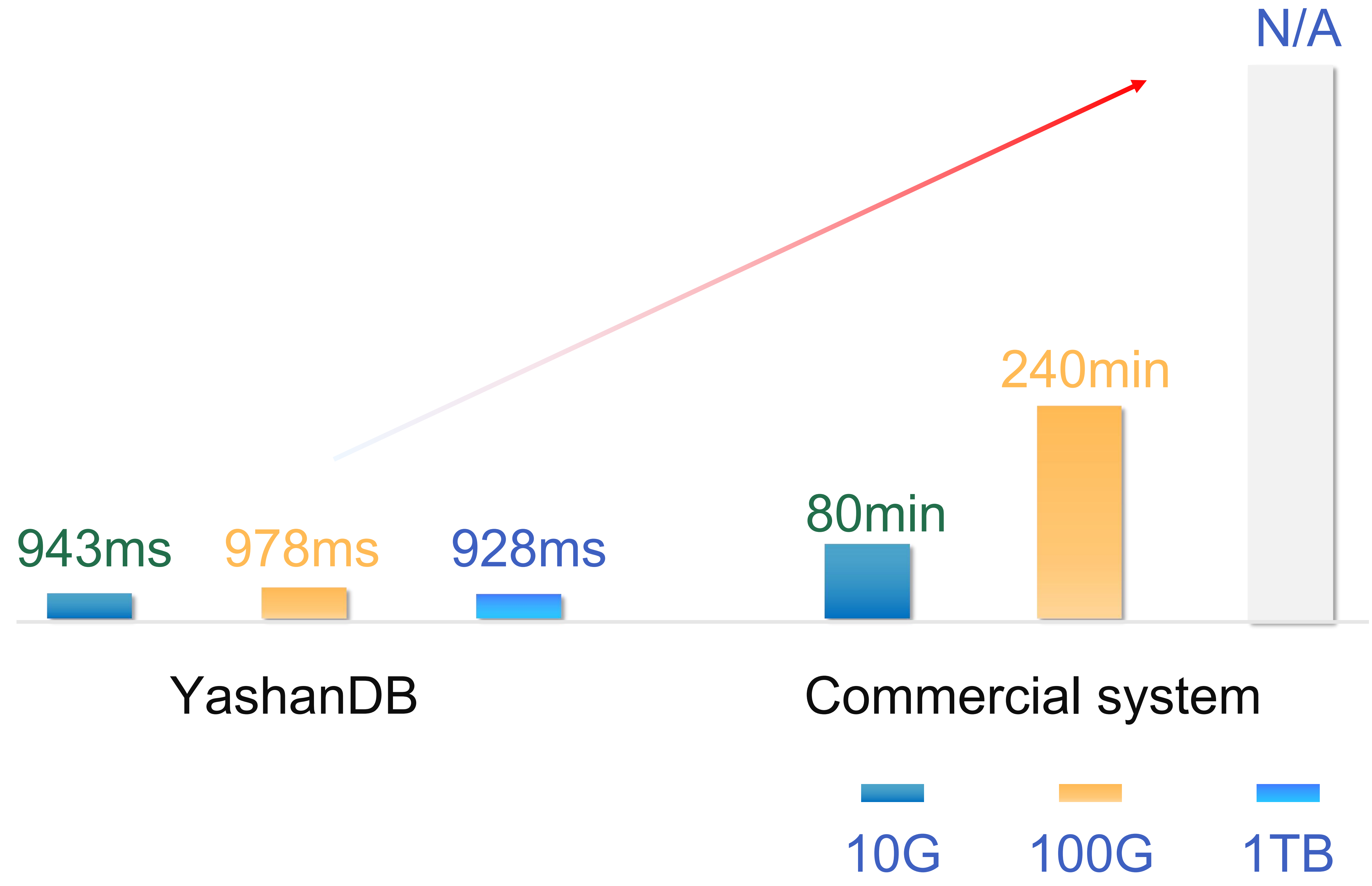


YashanDB

Access a bounded amount of data no matter how big the data grows

Equip database systems with the capacity of big data processing





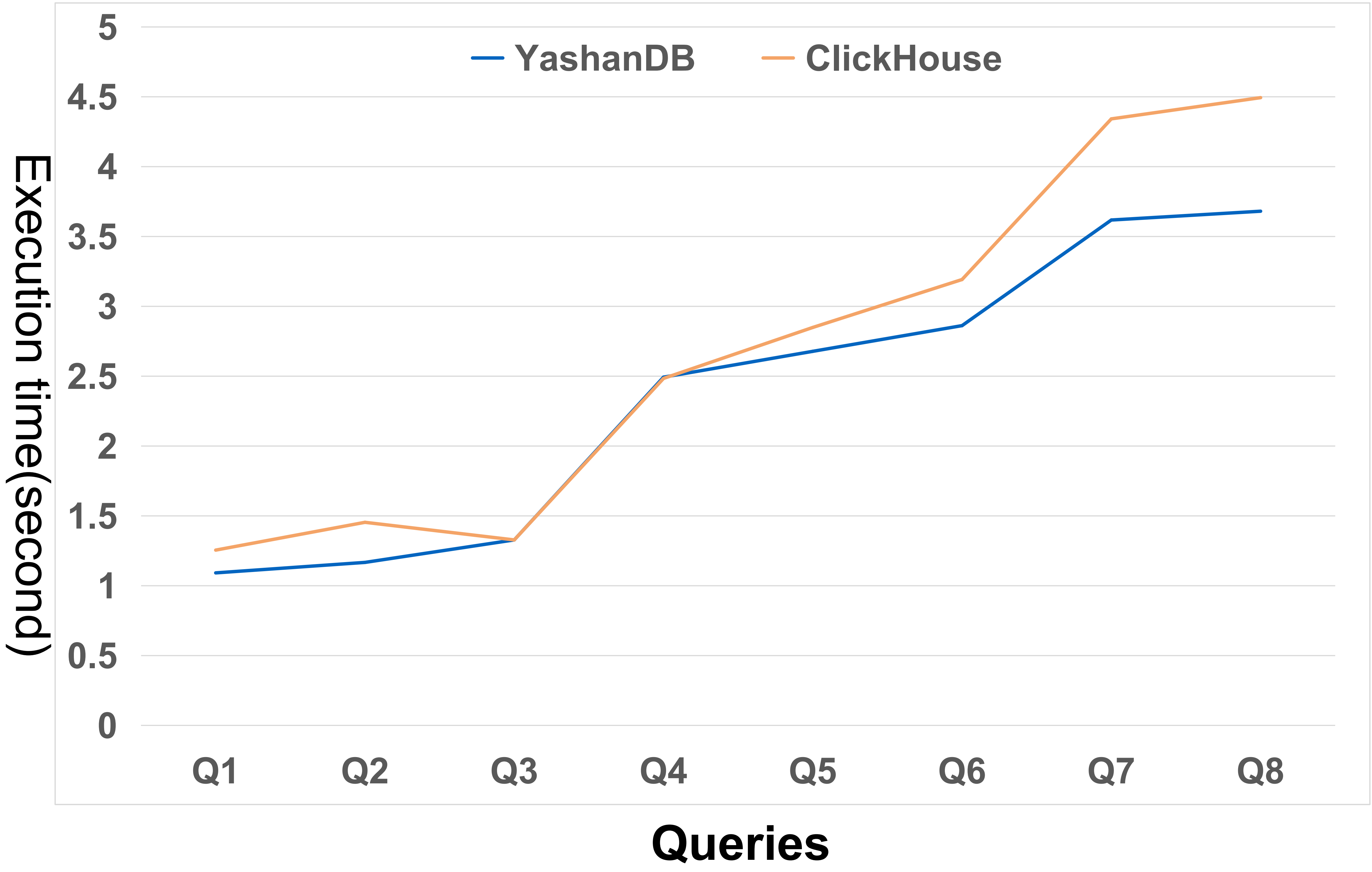
YashanDB in action

✓ **TP:** YashanDB is up to 7X faster than Oracle, and 60X faster than MySQL; Shenzhen Gas Corp. (13 provinces, 60+ cities, 10M households)

SQL	MySQL (ms)	YashanDB(ms)	Oracle(ms)	Improvement
SQL1	33	24	31	**29%**
SQL2	258,968	107,063	253,601	**137%**
SQL3	203,078	100,389	102,189	**2%**
SQL4	304,824	69,527	372,509	**436%**
SQL5	947	101	238	**136%**
SQL6	3,079	2,917	4,185	**43%**
SQL7	311	240	2,119	**783%**
SQL8	3,689	938	2,794	**198%**
SQL9	653,170	156,459	78,222	-50%
SQL10	350	112	216	**93%**
SQL11	3,094	2,533	3,396	**34%**
SQL12	318	25	73	**192%**
SQL13	12,092	196	459	**134%**
SQL14	2,119	67	69	**3%**
SQL15	65	48	51	**6%**

✓ **AP:** 18% faster than ClickHouse

- CDC queries
- X-axis: various queries

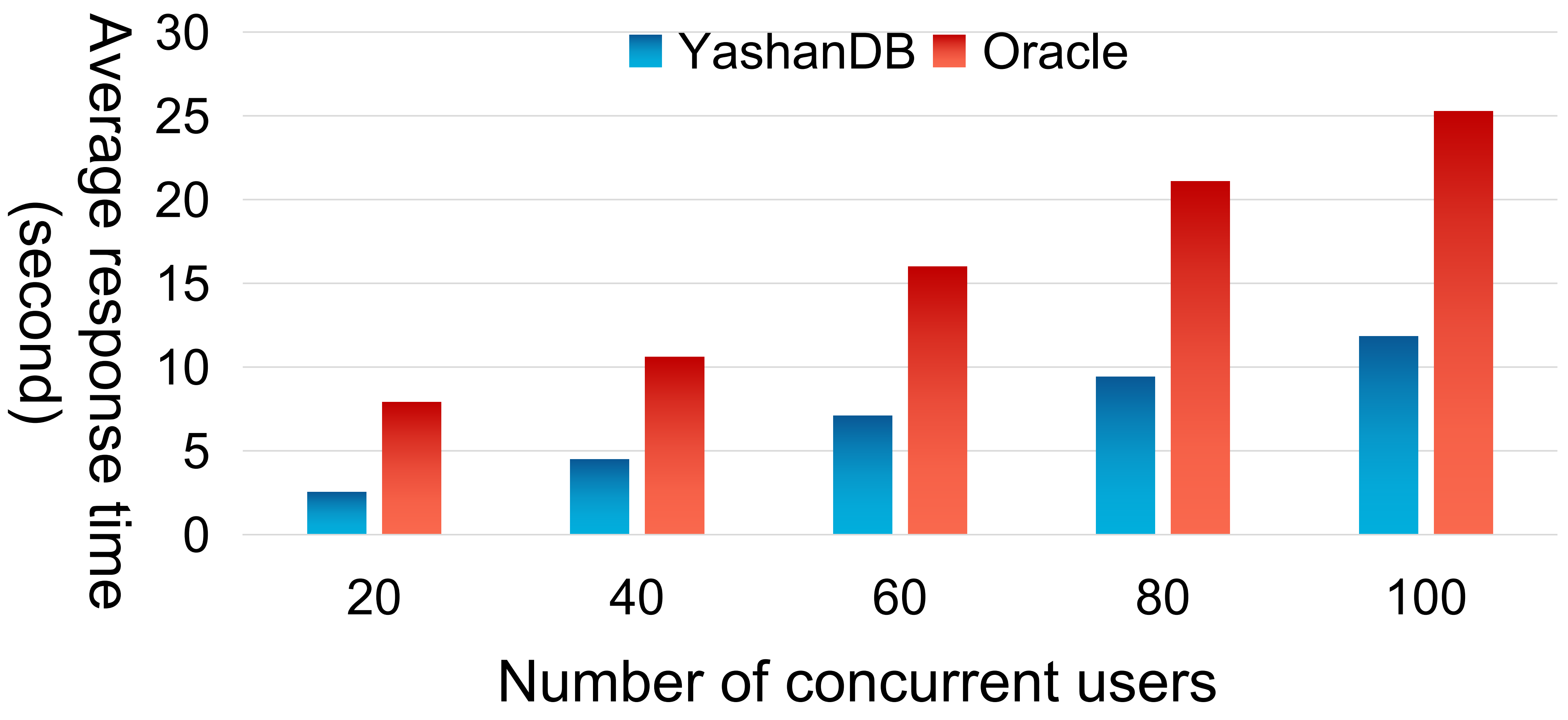


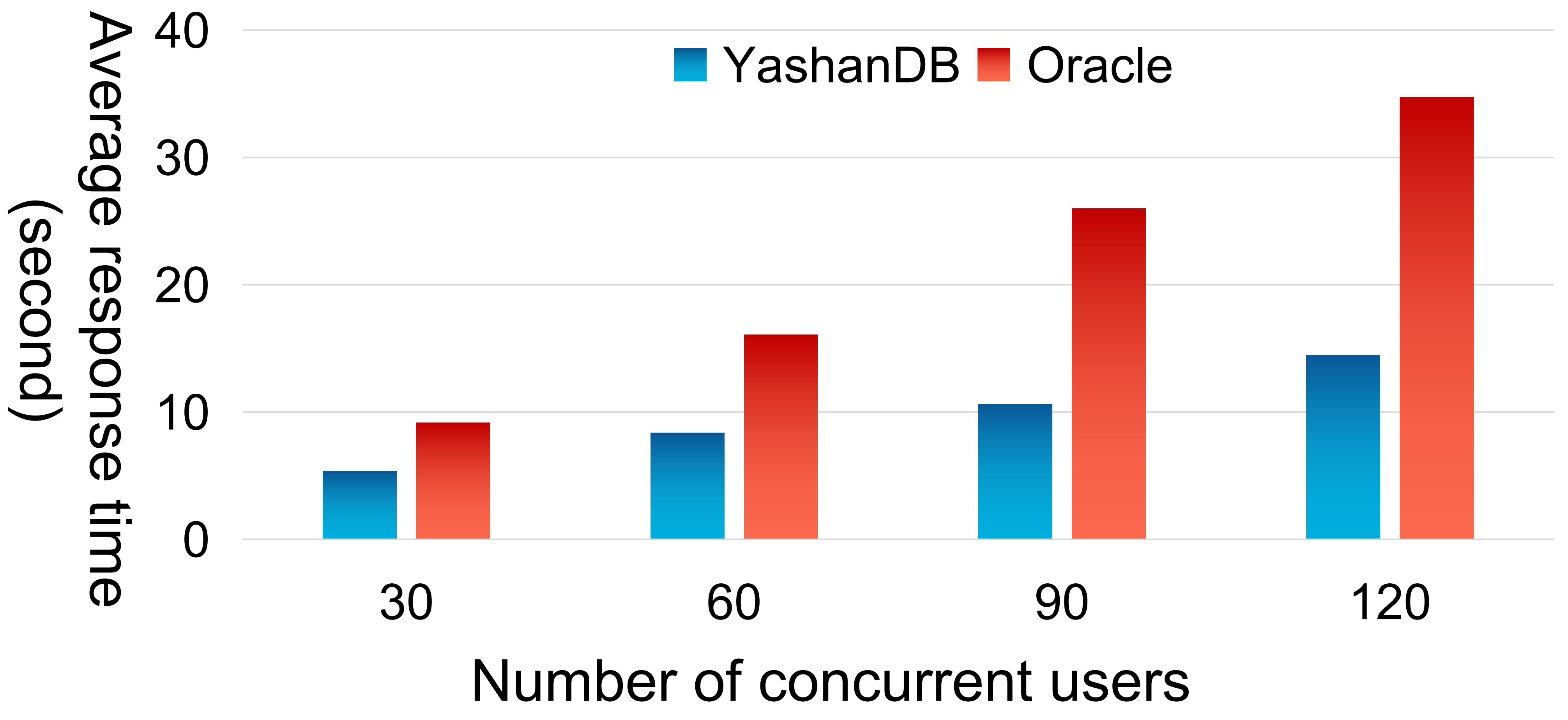
YashanDB outperforms SOTA for both AP and TP

YashanDB in finance (banks, fund, insurance)

- For a workload, YashanDB is 50% faster than Oracle, and its TPS (Transactions per Second) is 2.3X-4.5X better
- Under mixed workloads, YashanDB outperforms Oracle by ~30%

Number of concurrent users	YashanDB		Oracle		**Comparison**
	TPS average	Average response time(s)	TPS average	Average response time(s)	**TPS comparison**
Single workload					
20	795.54	2.55	144.614	7.91	**450%**
40	805.03	4.483	240.528	10.616	**235%**
60	815.23	7.093	246.254	15.994	**231%**
80	838.322	9.423	248.2	21.094	**237%**
100	845.506	11.84	245.521	25.286	**244%**
Mixed workload					
30	59.517	5.364	42.813	9.18	**39%**
60	55.693	8.386	43.027	16.09	**29%**
90	59.206	10.616	42.457	26.005	**39%**
120	54.451	14.453	43.019	34.738	**26%**





Variety: Queries across relations and graphs

- ✓ *Theory: Heterogeneous Entity Resolution (HER)*
- ✓ *YashanDB: Semantic join across relations and graphs*

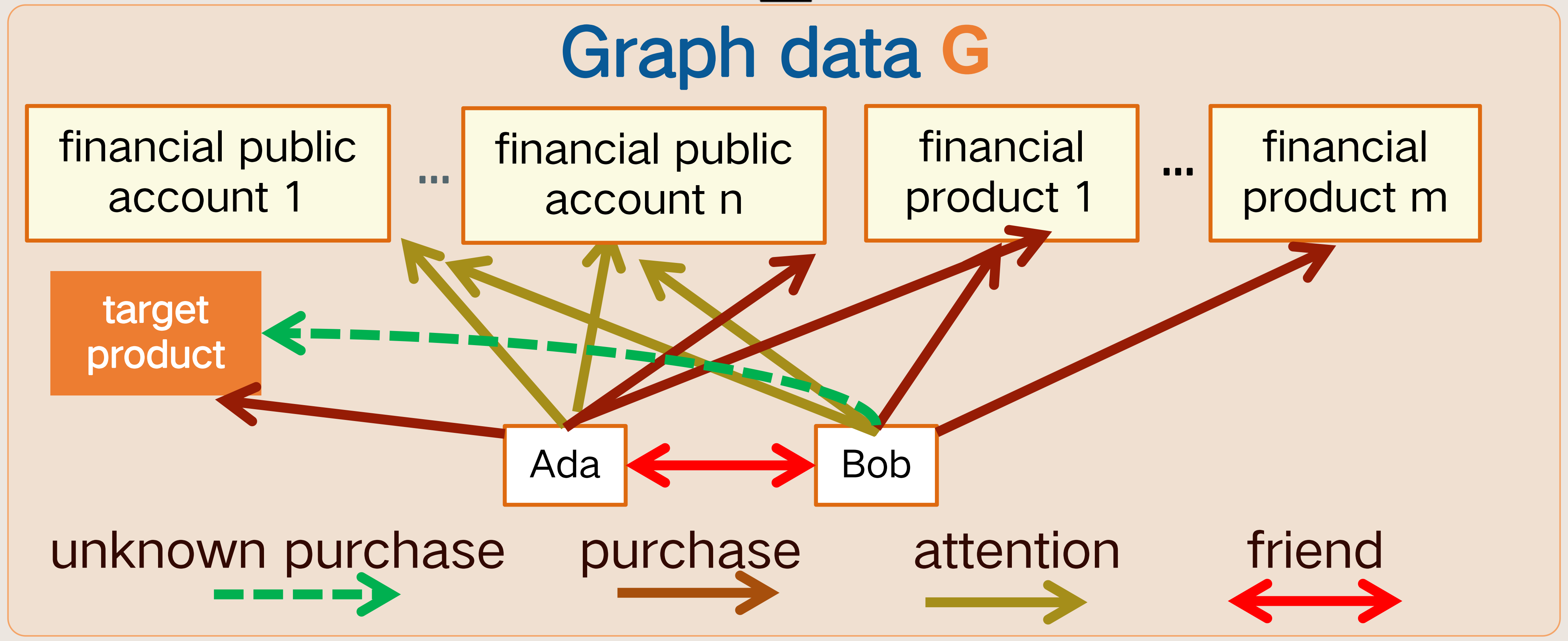
Heterogeneous queries across relations and graphs

A question raised by FinTech collaborators

User	Credit	preference	credit	purchase
Ada	good	Share	fair	yes
Bob	fair	Fund	excellent	?

Relational database D

FinTech query



- ✓ Customer data: relational database D
- ✓ Transactions: graph G

Recommend financial product fp to Bob?

- ✓ Two conditions:
  1. Bob has good credit (in D),
  2. Ada and Bob have at least three common products in portfolio, and Ada has invested in fp (in G)

How to decide Bob in relation D and Bob in graph G are the same person?

"Can I write the query in SQL"?

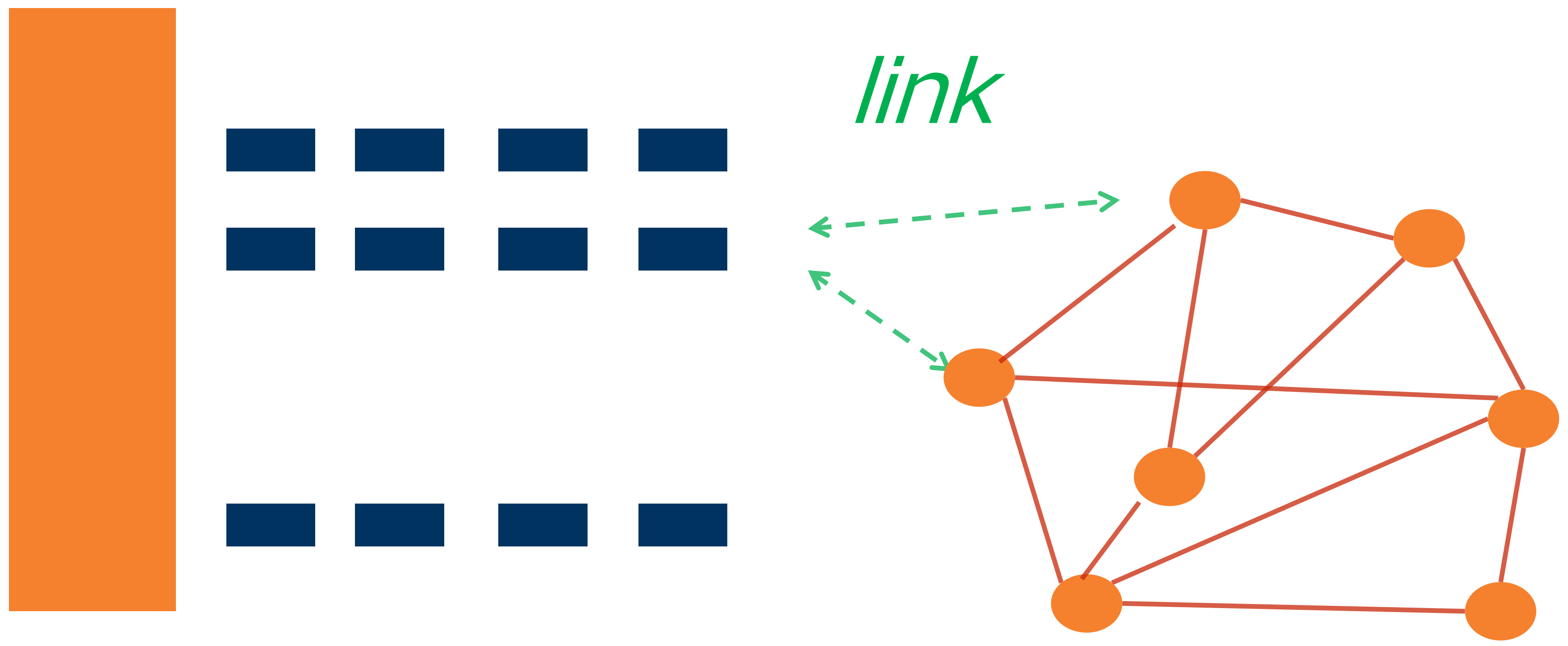
✓ Synthesizing and correlating data across relations and graphs

Variety: The added value of big data comes from diverse sources

Heterogeneous Entity Resolution (HER)

✓Given a tuple t in D and a vertex v in G, check whether t and v refer to the same entity?

- heterogeneous structures
- paths of v vs. attributes of t



✓Parametric simulation: tuple t and vertex v match only if their representative "descendants" are semantically close, pairwise

- ML models: assess semantic closeness of vertices and associations
- Topological matching: inductively defined to collect global information

Robin Milner: graph simulation (path-path)

✓Complexity: O(|D| |G|) time, no more expensive than relational ER

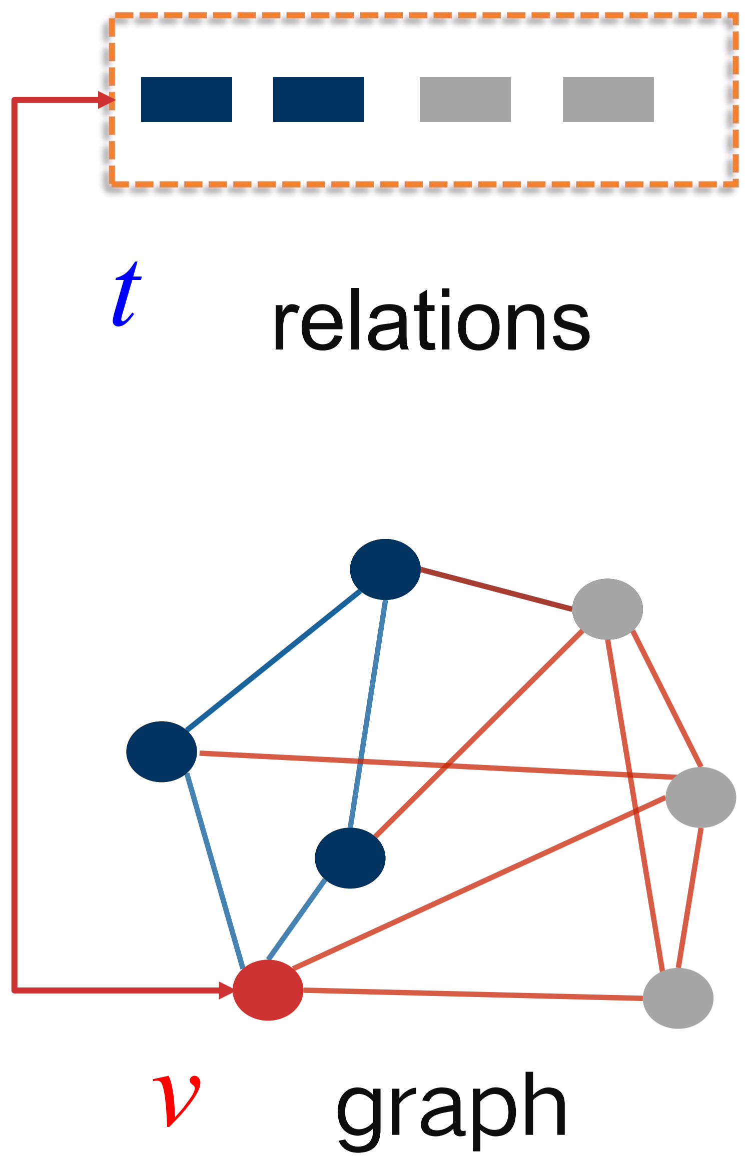
Embedding ML models into topological matching

Semantic joins

✓Semantic join: If tuple t and vertex v refer to the same real-world entity, then we can "join" t and v, and complement tuple t with additional properties of vertex v

- HER: check whether tuple t and vertex v match
- EXT: extract properties of v as additional attributes of t

A semantic extension of natural join in SQL



✓What can we do with semantic join?

- A capacity for RDBMS to query relations and graphs in SQL
- On-demand data integration for data lakes to augment tuples in Q(D) with graph properties

Veracity: Getting high-quality datasets

- ✓*Real-life data is dirty*
- ✓*Rock: Improving data quality*

Real-life data is often dirty

First name	Last name	Address	Mobile number	Area code	City
Mary	Smith	Hutong No.2	158223004	020	Beijing
Mary	Dupont	10 Elm Street	4844731483	610	New York
Mary	Dupont	No.6 Main Street	8143008970	010	null

Q: The dataset is not quite correct. Identify 5 potential problems

(a) Missing value. (b) Semantic consistency. (c) Duplicates. (d) Stale data. (e) Missing tuples.

The U.S. loses $1M per minute due to bad data.

——*DWI*

Dirty data costed the U.S. $3.1 trillion in 2016.

——*IBM*

20%~35% of profit losses are caused by data quality issues.

——*Total Information*

"The #1 problem to big data analytics"

Rock: A data quality system

Rock: Unifying machine learning (ML) and logic deduction

✓Input: A dataset D (relations or graphs)

✓Output: A cleaned dataset D_c for subsequent queries and applications

Central problems:

- Duplicates
- Semantic inconsistencies
- Stale data
- Missing data

Underlying algorithms:

- Rule learning
- Error Detection
- Error correction with certainty
- Incremental learning/training

Criteria:

- Accuracy
- Efficiency
- Scalability
- Interpretability

Data Profiling	Rule learning	Rule Execution	Knowledge
Schema mapping	Sampling	Certain fix	Knowledge graph
Exploratory data analysis	Prior knowledge	Parallelly scalable deep cleaning	Logical rules
Data visualization	Top-K / anytime	Incremental methods	LLMs, ML models

Banks, fund, service providers, logistics, data market

Rock in action

Domain	Pain Points	Rock	Feedback

Logistics

Problems:

- Large data collection: 170K+ tables, 10M+ attributes.
- No data standardization across different departments

Performance:

- Semantic mapping across tables
- Accuracy > 85%, far better than ML models

Bank

Problems:

- Rules are handcrafted by human experts.
- Costly, error-prune, fragile
- Not real-time

Performance:

- Accuracy 97%, from 81%
- Manual effort reduced by 8X

Knowledge base

Problems:

- A lot of duplicates
- Not scalable
- Missing data (null)

Performance:

- Find 450K duplicates in 4.5M entities.
- Accuracy > 95.4%;
- 100X faster than ML models

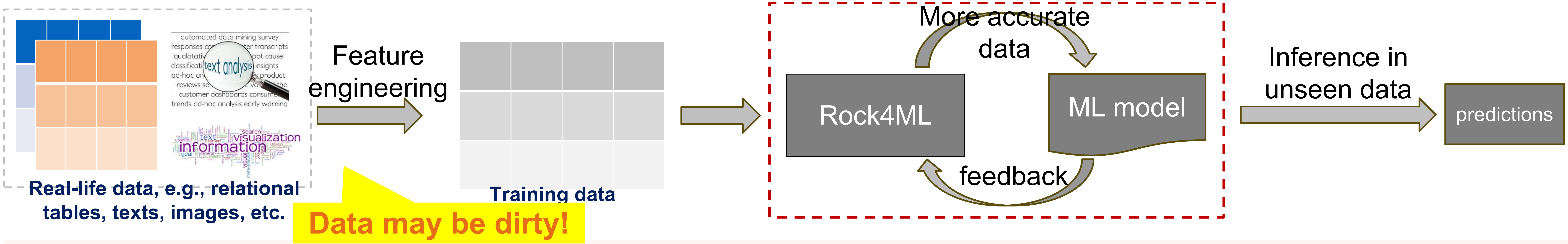
Methods

- Schema mapping
- Rule learning
- ML + logic
- Error detection
- Error correction
- ...

An infrastructure for data transaction market

Rock4ML: Data cleaning for ML

- ✓ Input: A dataset D, and an ML model M (possibly LLM)
- ✓ Output: A cleaned dataset D_c to maximize Accuracy(M, D_c), Fairness(M, D_c) and Robustness(M, D_c)



New challenges:

- ✓ How to clean document data, typical training data for LLMs?
- ✓ How to impute missing labels and correct mislabelled data?
- ✓ How to make blackbox ML models more accurate? E.g., ER may do more harm than good.
- ✓ How to enrich data for ML, e.g., adding adversarial examples to prevent adversarial attacks?

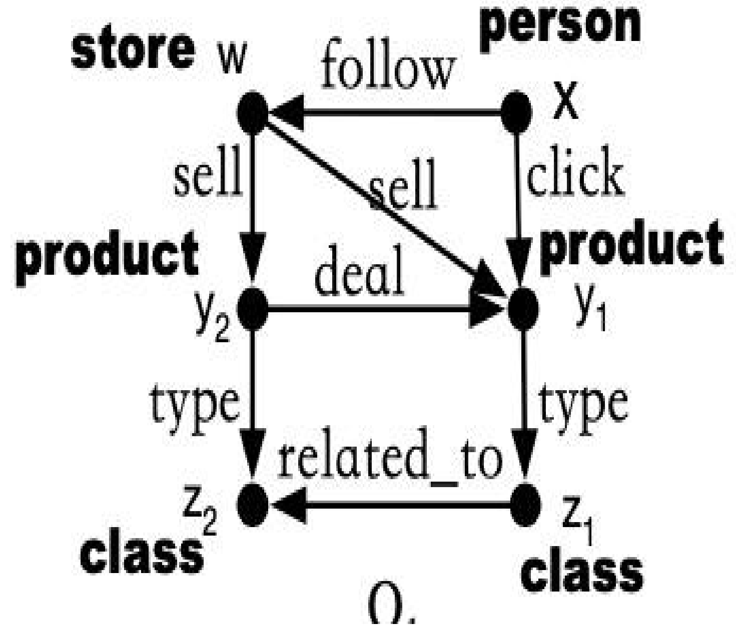
Make ML models more accurate, fair, robust and practical

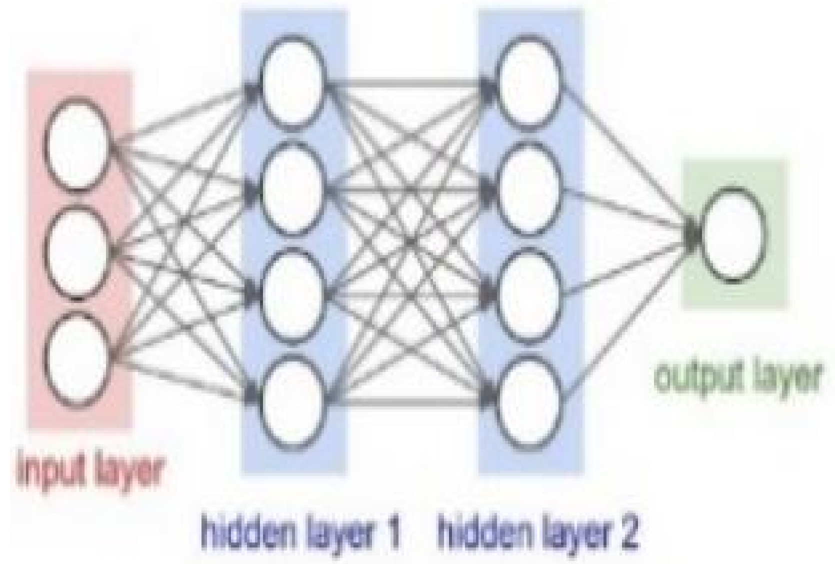
Value: Getting values out of big data

- ✓Method: Machine learning or logic rules?
- ✓Fishing Fort: A model of ML + logic deduction

Fishing Fort: Big graph analytics

Graph association rules (GARs) $Q[\vec{x}](X \rightarrow Y)$





- ✓ Q: graph pattern
- ✓ $\vec{x}$: a list of vertices in Q
- ✓ X, Y: conjunctions of predicates
- ✓ $X \rightarrow Y$: dependency

Predicates:

- ✓ link predicates l(x, y); logic predicates ... orders
- ✓ **ML predicates $M(\vec{x}, \vec{y})$** : ER, similarity ...

Possible for GNN-based models: FO2 with limited counting, for vertex classification and link prediction

Interpret ML predications in terms of

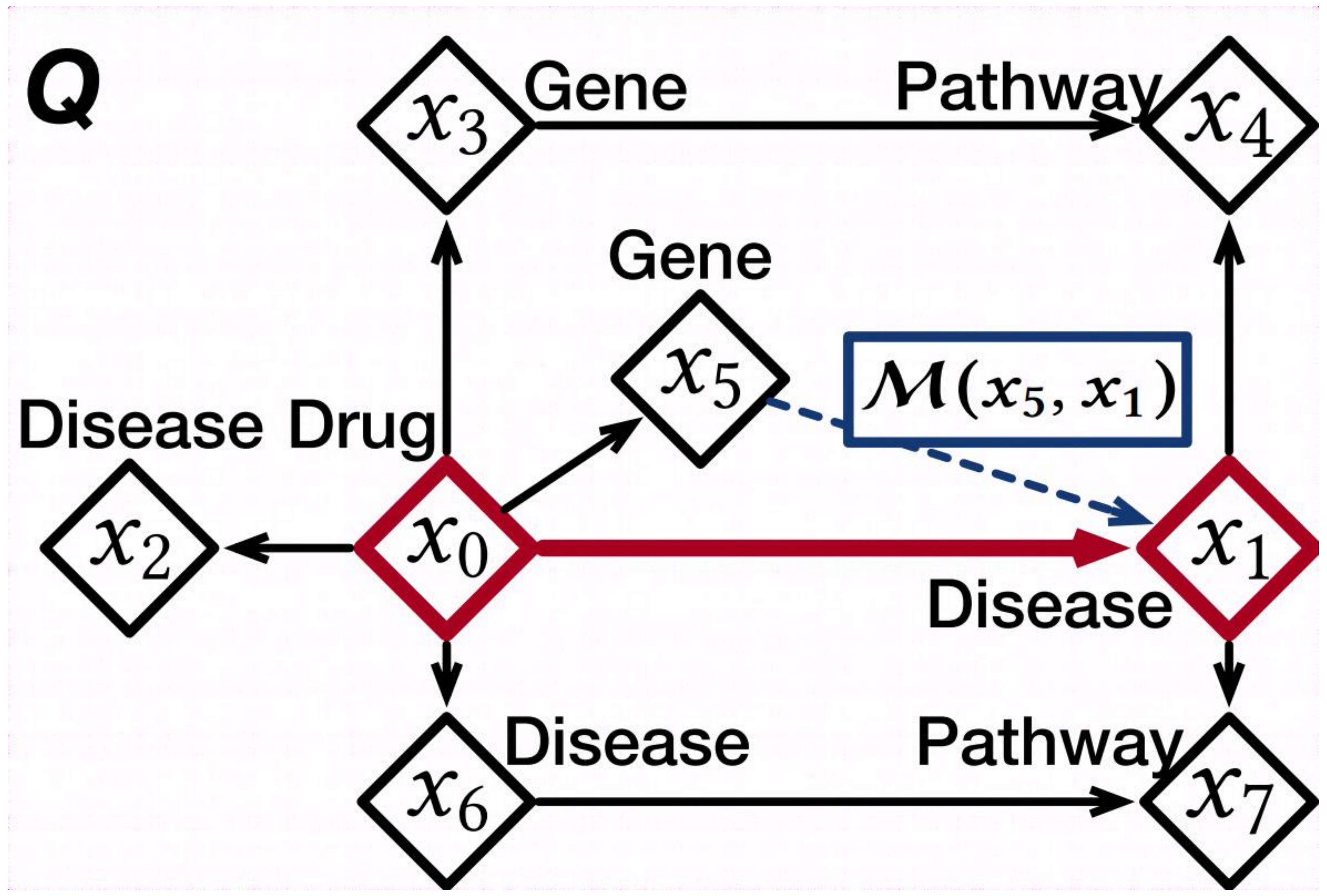
$$Q[\vec{x}]\ (X \rightarrow M(\vec{x}, \vec{y}))$$

MedHunter: Drug repurposing for Parkinson's disease

New drug is costly:

> 10 years, \$1 billion, success rate < 10%

$Q[\vec{x}]\ (X \rightarrow l(x_0, x_1))$



- ✓ CTD (Comparative Toxicogenomics Database)
- ✓ *Identified 5 drugs for Parkinson's disease: 4 with published evidences, 1 under active lab investigation*

Pattern Q and conditions X: drug x_0 may work for Parkinson disease x_1 ***because***

- x_0 has known impact on an inborn genetic blood disease x_2
- x_0 has known effect on skin cancer x_6, which shares an effect pathway with x_1
- x_0 interacts with gene x_3, which shares an effect pathway x_4 with x_1
- x_0 interacts with gene x_5, which has a predicted relationship with x_1

DDA (disease-drug association) = missing/hidden links

MedHunter: PPI prediction for drug development

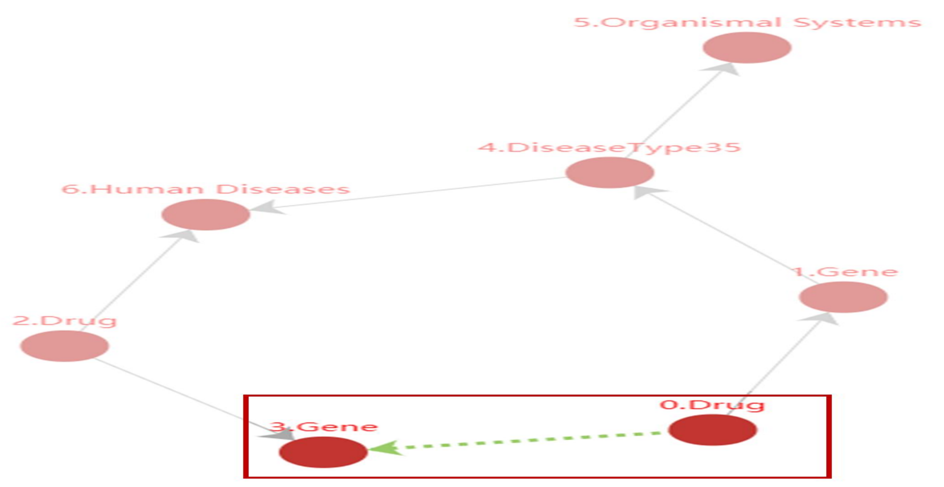
PPI: protein-protein interaction, for peptide-based drugs

- BioGrid data

MedHunter predicted the existence of self-interactions in human protein SYT2 in May 2022.
In the same month, **Nature** published a similar finding: synaptic binding protein SYT2 is **the best target** to block mucin secretion.

MedHunter improves the accuracy of PPI prediction by **43%**

43%



Biogrid_id_ A	Biogrid_id_ B	RGCN_score	FE_score	Official_symbol_A	Official_symbol_B	Entry_name_A	Entry_name_B
126085	124158	0.990037322	0.710992157	SYT2	TNRC18	sq|Q8N910|SYT2_HUMAN	sq|O15417|TNC18_HUMAN
126085	112742	0.983470738	0.728862166	SYT2	TAF7	sq|Q8N910|SYT2_HUMAN	sq|Q15545|TAF7_HUMAN
126085	**126085**	**0.983071744**	**1**	**SYT2**	**SYT2**	**sq|Q8N910|SYT2_HUMAN**	**sq|Q8N910|SYT2_HUMAN**

Early stage drug development: Save time, money and lives

Dream Creak: Lithium-Iron Battery manufacturing

Safety: Only battery cells of the same capacity can be packed in the same module

Capacity grading: The current industry practice

(1) Charge a battery; fully discharge it, cool down. (2) Fully recharge it; decide the capacity

Costly:

- 16 -- 24 hours, 2 phases
- **Energy:** charging, cooling, temperature control, approximately $2.3M for 1GWh
- **Equipment:** e.g., battery sites, approximately $7.4M for 4GWh

The effectiveness of Dream Creak

time	**80%** ↓	energy	**50%** ↓	Error rate	**≤1‰**

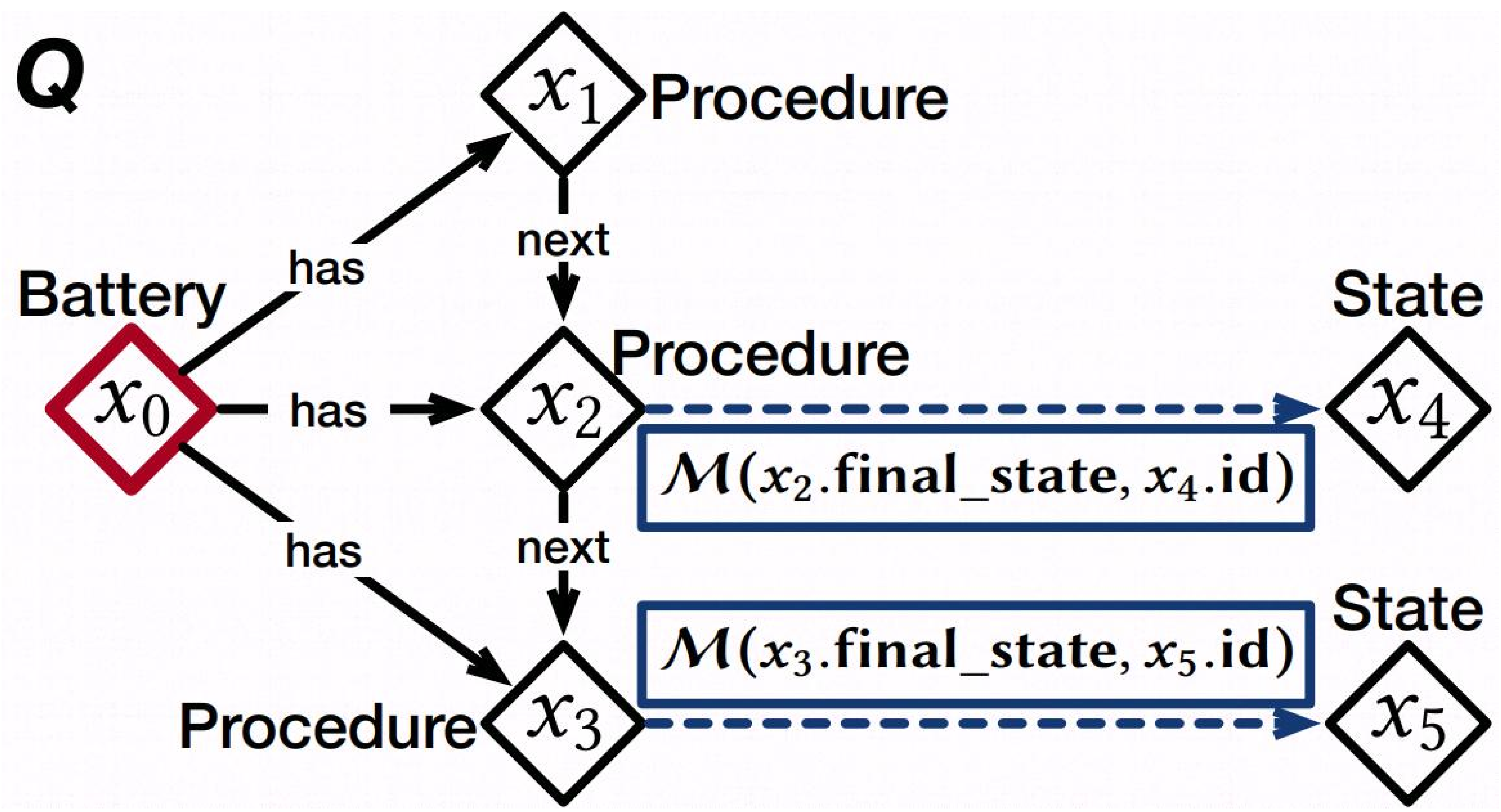
Determine the capacity of battery cells by data analytics

Dream Creak: Speed up the process of capacity grading

$Q[\vec{x}]\ (X \rightarrow x_0.\text{capacity} = 8)$

✓ *Use the data from the 1st phase to decide the capacity, skip the 2nd phase*

✓ *Reduce time from 16-24 hours to 4 hours, saving 80% of equipment cost*



Pattern Q and conditions X decide capacity:

- weight before and after the Electrolyte Filling procedure (x_1)
- charging current and initial voltage for its Formation-A procedure (x_2)
- the Formation-A procedure (x_2) has final state x_4 categorized by ML model M_8
- charging current and initial voltage for its Formation-B procedure (x_3)
- the Formation-B procedure (x_3) has final state x_5 categorized by M_8

Already built in 1GWh production lines

Mirror: Online recommendation

ML models: collaborative filtering (CF), Content-based (CB), hybrid

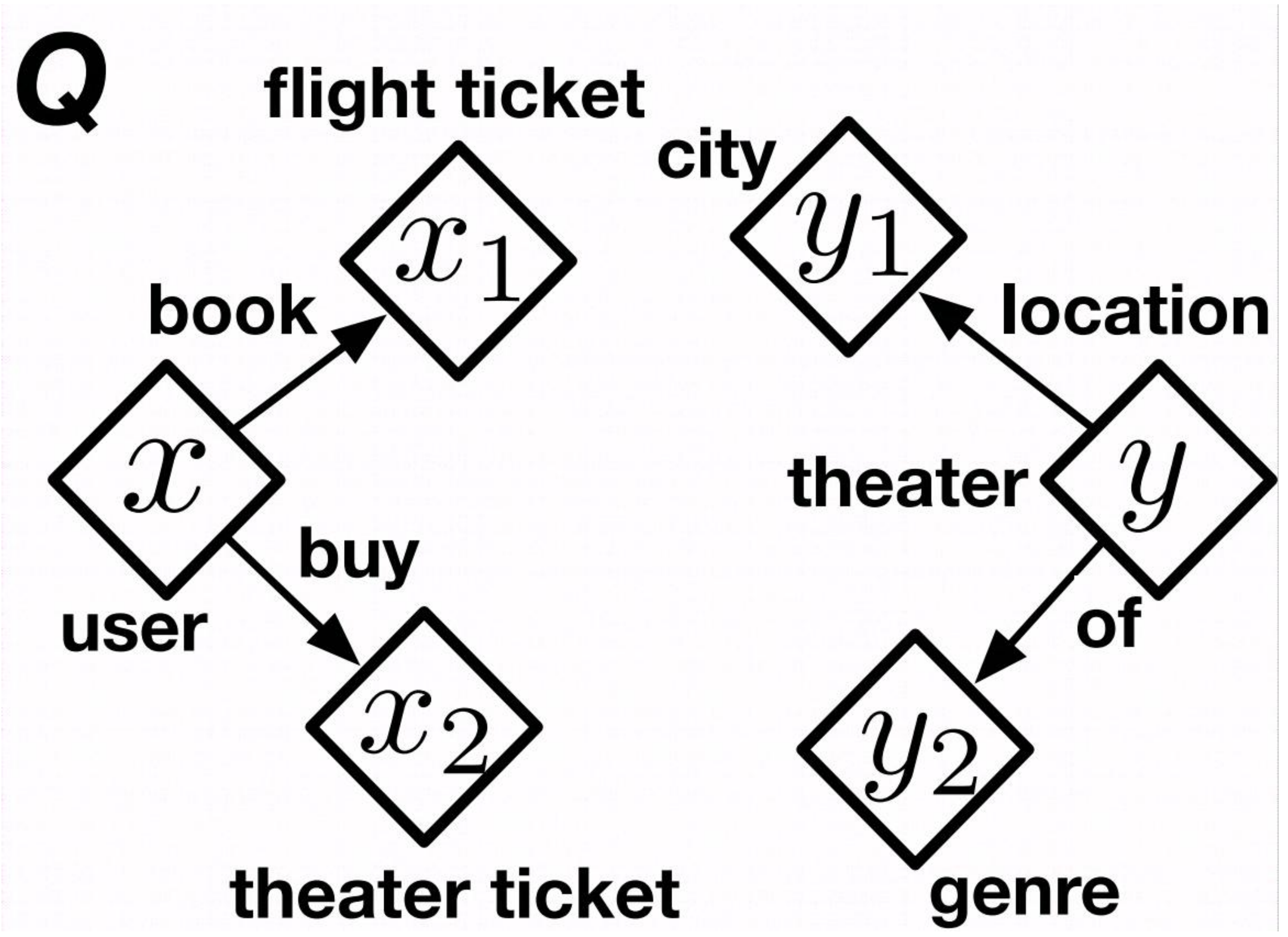
If $M(x, y) \geq \delta$, then recommend item y to user x

$$\mathbf{Q[\vec{x}]\ (M(x, y) \geq 0.6 \wedge x_1.destination = y_1.name \wedge x_2.genre = y_2.name \rightarrow rec(y, x))}$$

M: an existing hybrid model for recommendation
$M(x, y) \geq 0.6$: recommend tickets of theater y to user x since x went to live theater in the past

Reduce FP: override ML recommendation if

- either x travels to a different city, or
- the show of y does not match the preference of user x



Red ... are satisfied

Risk control at a bank: Caught 10384 possible fraudsters

Reduce FPs and FNs by incorporating logic conditions

Dasan Pass: Predicting Cyber Attacks

Attack event prediction allows preemtive defense measures

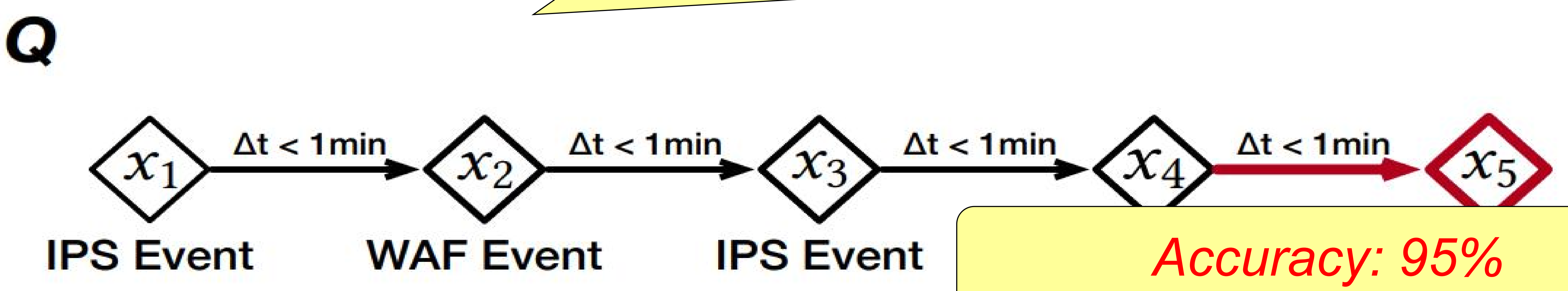
IDS, IPS devices: detect attacks (with delays), cannot make predictions

Dasan Pass: identify attack paths in advance via
- temporal graph analysis
- combined status analy
- auto mining of event-p

Possibly attack during time [t4, t4+1min], if
- An IPS device detects a Nmap scan event on x0 at time t1
- A WAF device detects a scanner operation on x0 at time t2, within 1min of t1
- An IPS device reports a Startracker alert on x0 at time t4, within 1min of t3
- An IPS device detects a WannaRen transmission on x0 at time t3, within 1min of t2

$Q[\vec{x}](x_1.event=$“Nmap sc

$\wedge\ x_3.event=$“Startrack



Accuracy: 95%

Already deployed at a large technology company

Velocity: Incrementalizing algorithms

- ✓*What incremental algorithms are "good"?*
- ✓*How can we develop good incremental algorithms?*

Develop good incremental algorithms

Incremental computation

- Input: Q, G, $Q(G)$, ΔG
- Output: ΔM such that $Q(G \oplus \Delta G) = Q(G) \oplus \Delta M$

Incremental algorithms are hard to write: ad hoc

✓ Many batch algorithms are already in place

✓ Few incremental algorithms have been developed

✓ Fewer incremental algorithms offer performance guarantees

Incrementalization: Deduce an incremental algorithm A_Δ from a batch algorithm *A*

✓ **AFF:** the difference between the data inspected by *A* for computing $Q(G)$ and $Q(G \oplus \Delta G)$

✓ **Bounded relative to *A*:** the cost is expressible as $f(|AFF|, |Q|)$

The inherent cost for incrementalizing batch algorithm A

Practitioners are familiar with the behaviors of algorithm A

Rock & Fishing Fort: Incremental data cleaning and analytics

Incrementalization: Given any fixpoint algorithm A, an incremental A_Δ is deducible from A using the same logic and data structure, such that under a generic condition, A_Δ is

- correct: $A(G \oplus \Delta G) = Q(G) \oplus A_\Delta(Q, G, Q(G), \Delta G)$
- bounded relative to A
- A_Δ uses at most linear timestamps as auxiliary structures

✓ Rock: Incremental rule learning, error detection & correction, temporal deduction

- 9.7X faster than batch algorithms when $|\Delta D| = 5\%|D|$
- Temporal deduction is 7.8% more accurate than other methods

✓ Fishing Fort: fraud detection with real-time transaction data

- Banks, e-commerce, cyber security
- Temporal graphs and event prediction over dynamic data
- 1.2k TPS/core throughput + seconds of latency v.s. > 5h for batch processing

The capacity of coping with dynamic data

Summing up

Big data: Theory and Practice

- ✓ Volume: From the bounded evaluation theory to YashanDB
- ✓ Variety: From HER to SQL across relations and graphs in YashanDB
- ✓ Veracity: Rule learning, error detection and error correction with certainty
- ✓ Value: ML + logic to benefit from both, and interpret ML predictions
- ✓ Velocity: Algorithm incrementalization in Fishing Fort and Rock, to cope with dynamic data

More challenges & Opportunities:

- ✓ **Rock4ML:** Data cleaning for ML
  - Improve downstream ML models for accuracy, fairness and robustness
  - Prepare data for LLM training
- ✓ **YashanDB:** Semantic joins across relations and data of other models
- ✓ **Fishing Fort:** Application domains
  - Quantitative trading, manufacturing industry

Invitation: Join forces to tackle the challenges together

VENI，VIDI，VICI

我来、我见、我征服

深圳计算科学研究院
Shenzhen Institute of Computing Sciences

www.sics.ac.cn