Graph XAI: Graph-augmented AI with ADEV*

Ricky Sun[1,†], Yuri Simione[2,†] , Jason Zhang[1,†], and Victor Wang[3,†]

[1] Ultipa, Inc., 2342 Poppyview Ave, San Ramon, CA, 94582, USA
[2] Ultipa, Inc., Viale Egeo 59, 00144, Rome, Italy
[3] Ultipa HK Limited, Building #3, HKSTP, Shatin, Hong Kong SAR

Abstract

Today's big data and AI frameworks face problems like questionable accuracy, shallow data processing depth, black-box in-explainability, and oftentimes low processing speed. This paper summarizes the work of Ultipa, introducing Graph XAI (Graph-augmented AI) and highlighting ADEV (Accuracy, Depth, Explainability, and Velocity). In contrast to many systems that sample data due to inability to traverse datasets thoroughly and quickly, particularly hindered by hotspot supernodes, Ultipa's graph system is designed from data structure and system architecture perspective to allow for ultra-low latency deep penetration, and accuracy is achieved with exhaustive traversal, which also allows for exponentially faster velocity. As graph data are ideally queried and processed using graph query languages and algorithms instead of the two-dimensional SQL and stored procedures, the intuitiveness and explainability are crucial in ensuring ADEV being fulfilled, this paper highlights how Ultipa's graph-native query language facilitates real-time recursive queries like path-finding, K-hopping, auto-networking, or identification of topological structures and communities works hand-in-hand with Ultipa's WebGL-powered graph manager to ensure end-to-end celerity and explainability.

Keywords

Graph XAI, Real-time Deep Data Processing, GQL, Graph Query, Graph Data Modeling, Network Analytics, Graph Database, HTAP, High-density computing, Traversal Boosting

1. Introduction: XAI and ADEV

XAI originally stands for eXplainable Artificial Intelligence, it signifies the needs for explainability against the results generated by AI, as well as the white-box explainability of the processes leading to the results. As we are broadening the adoption of AI across all industries, the meaning of XAI transcends explainability, more meanings are added to it, including but not limited to:

- Accuracy: the computed results should be adequately accurate.
- Depth: the ability to traverse connected data set deeply.
- Velocity: speed at which data is generated, modeled, ingested, and processed [1].

The accuracy problem of AI and big-data systems is commonly attributed to human ignorance or procedural unfairness of systems designs [2] and [4]. Though there are mitigation plans which try to improve the accuracy (and explainability) of such AI/big-data systems, the problem lies with the underpinning system architecture and design philosophy which are inaccurate and unexplainable by nature [3] and [4] and [5]. Specifically, the joint-force of big-data and AI aggravated the problems – data sampling and profiling [6] are widely used, however, a major drawback with sampling is that it may work in one domain, but not in the others. For instance, most ANNs (Artificial Neural Networks) are originally designed to handle images (i.e., photos or

Paestum'24: 26th Int'l Workshop on DOLAP, March 25, 2024, Paestum, Italy

*Corresponding author.
†These authors contributed equally.
ricky@ultipa.com (R. Sun); yuri.simione@ultipa.com (Y. Simione); jason@ultipa.com (J. Zhang) ; victor@ultipa.com (V. Wang)
0009-0004-3025-4500 (R. Sun)

remote-sensing images) with sampling and profiling techniques, but such networks may run into serious inaccuracy problems when dealing with financial datasets. The reason is that to examine a certain account's historical behavior, or to conduct attribution analysis [7] against a banking center, exhaustive traversal of the account's (or the center's) transactions is a must-have, and sampling will be far off from reality. Similar accuracy problems happen in medical industry [8], supply chains, telcos, and power grids.

The depth factor is about how deeply and thoroughly a system can traverse (or penetrate) the given dataset. A major weakness with relational databases and big-data frameworks (being the foundation of today's AI systems) is their poor ability to handle recursive queries, such as joining of tables, due to effect of cartesian product (cardinality), the performance degradation is exponential as the number of tables (roughly equivalent to the depth of the query) [9] and [10] and the sizes of tables increase, which dramatically limits the ability of such system to deeply (recursively) penetrate the data. A horizontally distributed system tends to have much worse performance on network analytics than a single-instance system that's capable of multi-thread processing [11], therefore, the trick for a distributed system to accelerate network analytics is to project data from distributed storage instances to one (or fewer) computing instance(s) where data will be centrally processed [12] and [13] and [14] and [29] – but the data projection process can be time-consuming, making it unfit for real-time decision making. Such limitations have real-world repercussions, for instance, doing attribution analysis [7] with Oracle is an extremely lengthy process when dealing with hundreds of millions of trades (transactions) scattered in dozens of tables. The collapse of Silicon Valley Bank in 2023 is a typical case of failure to conduct timely liquidity risk management, and attribution analysis, to understand the bank's portfolio and liquidity positions and forecast quantitatively and qualitatively on daily or intra-day basis [15].

Velocity concerns the speed at which data is generated, captured, schematized, and processed. The maximum speed of a system is only tested by putting into the context of the maximum depth it reaches within a bounded timeframe. Given the rise of cryptocurrency, blockchain and Web 3.0, numerous performance analysis [16] and [17] and [18] have been conducted against systems implementing such infrastructures. Most existing big-data and NoSQL frameworks (and the RDBMS) are designed with storage-centric mindset where horizontally scalability design is prioritized higher than depth-oriented velocity. Such mindset ensures unsatisfactory velocity when there is need for deep data processing – on the other hand, the popularity of AI introduces added layers of black-box in-explainability when processing data through machine learning or deep learning frameworks [19] and [20] and [21].

This paper focuses on four aspects of XAI which collectively referred to as ADEV (Accuracy, Depth, Explainability and Velocity), and how ADEV can be achieved with a novel graph system design [21].

2. XAI and ADEV How-to

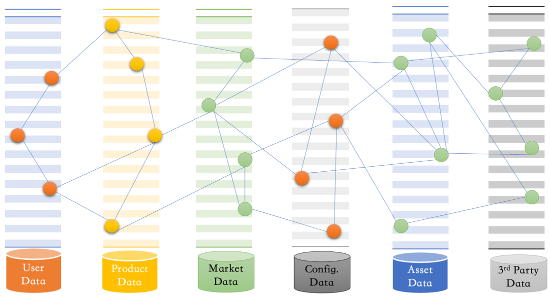

Figure 1: From Relational Tables to Graph(s)

Graph [23] organizes, otherwise siloed, data in a unified, connected, and holistic. Figure 1 illustrates how data can be connected in a graphical way in comparison with the tabular data modeling [24]. Once data is organized in graphical way, there are essentially two types of data operations, which are:

- Meta-data operations: CRUD (Create-Read-Update-Delete), aggregations and filtering, etc.
- High-dimensional operations: which are network oriented, such as finding paths, networks, subgraphs, etc. Note that heterogeneous types of data can be mixed and assembled in one batch of results.

There are 3 types of high-dimensional data operations:

- K-Hopping: traversal of a vertex for K hops per filtering conditions.
- Path-finding: from shortest-path to circular-path (cycle detection), to more sophisticated networking or auto-spreading queries.
- Graph algorithmic operations: such as degree, similarity, community detection, random walking, and graph embeddings.

Taking K-hop as an example, BFS is guaranteed to be accurate and effective, because DFS does not track the depth of shortest-path (hops) from current vertex to the starting vertex and standing at the Kth hop on a DFS path does not guarantee the correctness. The point here is that it's rudimentary to open up the implementation core of any analytical query and make it white-box explainable, to ensure the results as well as the procedures are explainable.

The advances of LLM and GPT have given us the impression (or illusion) that AI soon will be taking over average human beings both on IQ and EQ fronts. Scholars around the world have investigated and criticized the hallucination and black-box problems with LLM/GPT [25] and [26] and [32]. From XAI's perspective, LLM/GPT's hallucination and un-explainability are rooted in their incapability to conduct deep traversals, or causality searches. Figure 2 illustrates that GPT lacks the ability to conduct causality search, which if conducted in a graph database is to find the shortest path between the parties against the dataset that must be part of what GPT [27] has been trained upon. Figure 3 shows that a shortest-path query of up to 5 hops is conducted on a data set populated with Wikipedia data.

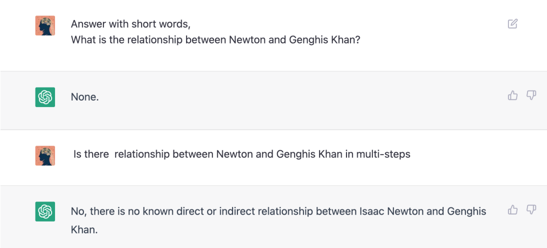

Figure 2: Graph-augmenting LLM

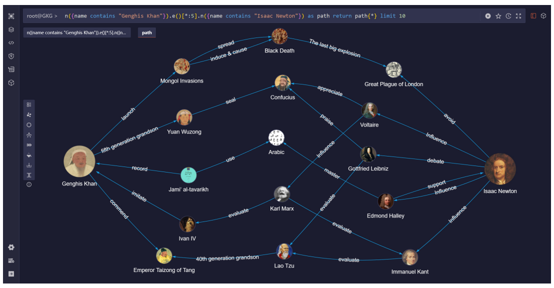

Figure 3: Path-finding (BFS/Shortest Path)

2.1 Graph-augmented: Accuracy

The accuracy problem in big data and AI analytics is multifaceted. We can summarize the accuracy problem into three classes:

- Inadequate computing power: sampling and incomplete traversal of data. Many big-data and AI systems are comfortable handling meta-data operations but not networked analytics.
- Faulty design or implementation: This happens with unvalidated implementations, such as using DFS for K-hop or shortest-path finding or no de-duplication of K-hop, or partial traversal, all of which may cause the results to be redundant, incomplete, or outright wrong.
- Ill Product-Market Fit: system designed for batch-processing, pre-calculation and pre-caching cannot server real-time scenarios.

We use a concrete example to illustrate the accuracy problem when running K-hop queries against a large Twitter-2010 SNS dataset of 42MM vertices and 1470MM edges, the dataset is densely populated with average degree of ~70 (SNS friends of a user), and with the existence of many hotspot supernodes with over 1MM neighbors (high-impact social influencer).

Table 1

Real-time Changing Results w/ Changed Topology

Starting Vertex ID	Ending Vertex ID	Query Depth	K-Hop Before Edge Adding	K-Hop After Edge Adding
20727483	28843543	1	973	974
		3	**27206363**	**27210397**
		6	10028	10027
50329304	21378173	1	4746	4747
		3	**29939223**	**29939314**
		6	9052	9052
26199460	32278263	1	19954	19955
		3	31324330	31324333
		6	3022	3022
1177521	6676222	1	4272	4273
		3	**17139727**	**17139725**
		6	3101	3101
27960125	48271231	1	7	8
		3	**20280156**	**20283107**
		6	25838	25836
30440025	38232241	1	3386	3387
		3	**23120607**	**3121930**
		6	**5437**	**5431**

In Table 1, the Twitter dataset's topology is changed in real-time by connecting a vertex with one of its 3rd-hop connected vertices with a new edge (orange-colored, as illustrated in Figure 4) , and we check the starting vertex's 1st, 3rd and 6th hop neighbors immediately before and after the topology change – in some cases the k-hop results may change dramatically, as shown in the last column in Table 1. In Figure 5, adding an edge between the C001 vertex and C009 will change C001's 1-hop neighbors from 3 to 4, and 2-hop neighbors unchanged, and 3-hop's from 7 to 6. If a system uses pre-processing and caching mechanism, it will continue to read pre-stored (stale) results and not be able to churn out updated query results accurately and instantly. At Ultipa, we designed an HTAP system [22] and [30], to ensure changing topologies is accurately and instantly reflected across all system data structures so that graph operations results can be accurate. This is further discussed in the following section.

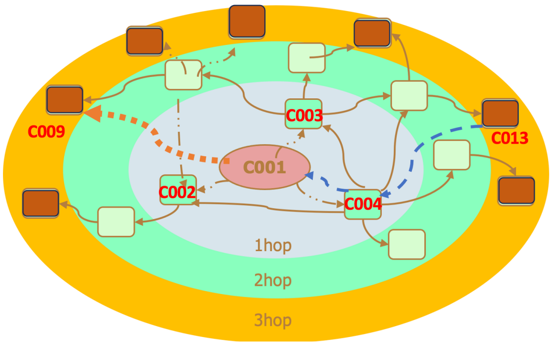

Figure 4: Topology Change Affects Query Results

Data structure plays a pivotal role here, conducting accurate analytics over graphs require both agility and resiliency, such as the ability to handle multi-graph as well as simple-graph, filter by direction of relationships, or attributes tied to vertices and edges, which may affect query results. For instance, if querying K-hop by inbound edges in Figure 4 (with added dashed blue ones), C001 has only one 1-hop neighbor, and one 2-hop neighbor (not the same as K-hop results).

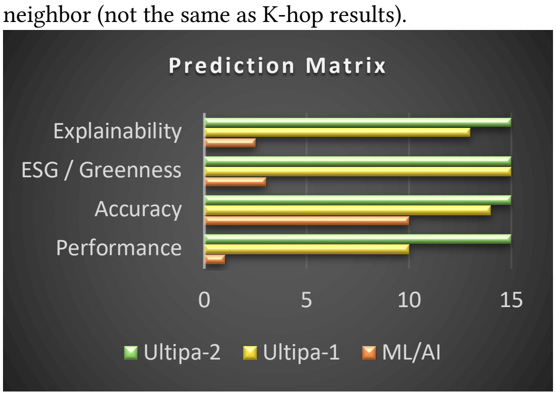

Figure 5: Graph-augmented Predictive Analytics

Accuracy problem is also tied to complicated system architecture, for instance, accuracy is easier to achieve with single-threading, but once multi-threading and data-partitioning-n-parallel-processing are introduced, data structures and systems architectures are more complex, and the results validation become much harder. Taking Louvain community detection algorithm as an example, the original algorithm was designed to function in a serial fashion, and parallel computing will complicate the matter by yielding faster but possibly inaccurate results (numbers of communities).

Accuracy problem can lead to serious ramifications in real-world applications. Taking retail-bank credit card spending (turnover) prediction as an example, Bigdata/DL frameworks are slow and inaccurate, and each 1% mismatch can be equivalent to $1 billion loss of cash reserve (and in-liquidity). Ultipa models card transactions as a graph network and extracts features via graph queries and algorithms such as weighted node degrees, page-rank, and random-walk to improve prediction accuracy. Figure 5 shows two batches of Ultipa graph-based predictions to significantly improve accuracy (40-50% better) and latency (10-15x) over ML/DL methodologies. The main cause for such improvement is the graph-based feature extraction of the transaction network accurately reflects card holders' (and merchants') behavior patterns therefore giving augmented prediction power. In comparison, big-data/ML predictions are still table centric and low-dimensional which can hardly track the supposedly high-dimensional entity behaviors; besides they are slow and tend to go black-box with sophisticated operations.

2.2 Graph-augmented: Depth

Graph's natural strength is to be able to analyze data that are connected, and what really sets one graph system apart from the others is its ability to penetrate the data much more deeply within the same time bound and upon the same underpinning hardware.

In Figure 3, we've illustrated the necessity for deep-traversal. To implement a graph XAI system with real-time deep traversal capability, there are 3 factors to consider:

- Low-latency: data structures that allow for lowest possible access time-complexity, ideally O(1).
- Mutability: read-only data structure and access patterns are easier to design but we must cope with read-n-write scenarios where data are mutable and supporting CRUD operations.
- Parallelization: serial access per query is easy to do, but the existence of supernodes would require parallel and accelerated access on a single query – because one query can lead to graph-wide traversal.

We could use Map or HashMap in C++ to implement the core meta-data data structure, but both are considered highly redundant in terms of memory consumption. We came up with a novel data-structure design of vector_of_vectors as illustrated in Figure 6, essentially, packing all edges connecting with a vertex in a mutable vector, but aggregating inbound and outbound edges in different sections for easier graph traversal. Such data structure can satisfy the needs for mutability and parallel access, and most importantly O(1) time complexity for per-hop data traversal – the computational complexity to visit all neighbors of a vertex is a constant O(1) which will empower exponentially faster deeper traversals.

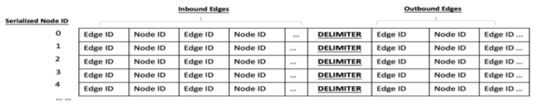

Figure 6: Node-Edge Adjacency Data Structure

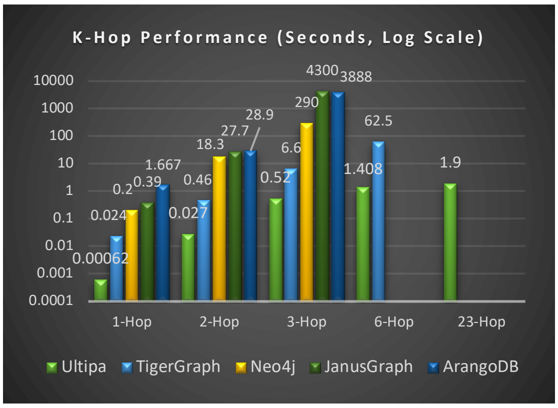

Figure 7: K-hop on Twitter-2010 Dataset

There are other forms of novel acceleration techniques, which we'll introduce in section 2.4 (Velocity). The effect of deep traversal capability is shown in Figure 7, where 1-hop traversal with Ultipa is done in microseconds while the other systems require at least dozens of milliseconds, and at 6-hop, only 2 systems (Ultipa and Tigergraph) can perform while Ultipa is nearly 50 times faster than the other. When the depth reaches 23-hop (this is close to the diameter [31] of the benchmarked dataset), Ultipa is the only system that returns (capped at 45-min), and in real-time (<1.9 seconds, with 99.9999% of the graph traversed, which is equivalent to 1500MM nodes and edges traversed within 2 seconds, indicating the system's capability to cover over 750MM+ nodes and edges per second).

There are many optimizations made in pursuing for real-time recursive deep penetration of dataset. Multi-layer storage-n-computing acceleration is one such optimization, which can be reflected in how critical system resources are consumed. Figure 8 shows that Ultipa uses more static memory but less dynamic memory comparing with other systems, meanwhile goes far more parallel in processing graph queries. A salient benefit of lower dynamic memory is equivalent to better system stability while the other systems risk running into OOM with deep-query processing.

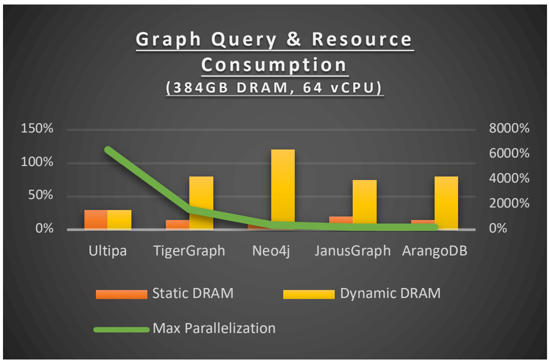

Figure 8: Resource Consumption Comparison

2.3 Graph-augmented: Explainability

Explainability is a corner stone of AI, big data analytics, business intelligence, and decision making. There are 2 important aspects of it:

- Explainable result and intermediate process: which mean the linkage between the source data and the result can be traced and explained in a step-by-step fashion. This also involves the explainability of query languages (SQL or GQL).
- Explainable architecture and system design: which require that the underpinning system architecture to be white-box explainable.

Graph data is meta-data that are organized and connected in high-dimensional ways, and the finest granularity of meta-data boils down to vertices and edges. By organizing and manipulating these meta-data in different dimensions, insights or certain facets of the graph can be generated on the fly. Explainability often demands for reverse thinking process that is to trace backward from the result to the intermediate process, and eventually to the source data (being analyzed) – if any part of the back-tracing process is hardly explainable, we'll suspect that the underpinning system has explainability problem even though the system may work well on many aspects.

Visualization is another important aspect that helps with explainability. Figure 3 shows the graphical results of a multiple-hop path finding with different types of entities and relationships clearly annotated for easy digestion. The query language itself is also important, but it would be pointless if we don't put this in the context of comparing with varied graph query languages.

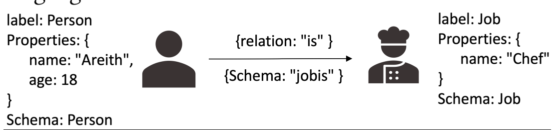

Figure 9: Explainable Query Language: Triplet

Figure 9 describes a simple triplet (vertex-edge-vertex) relationship, and it's plain to tell the differences of intuitiveness by different GQL dialects. Imagine the cognitive-loading differences when doing sophisticated queries using different GQL dialects:

Neo4j Cypher: /* Out-of-line filtering */

```
Match path = (p:Person) – [{relation:"is"}]-(j:Job)
Where p.name = "Areith" && j.name=="Chef"
return path
```

Ultipa GQL (Schematic): /* Inline filtering */

```
n({@person.name == "Areith"}).e({@jobis}).n({@job.name=="Chef "}) as paths return paths
```

Ultipa GQL (Schema-free):

```
n({name == "Areith"}).e().n({name=="Chef "}) as paths return paths
```

Gremlin: /* Extensive chaining */

```
g.V().hasLabel('person').has('name',"areith").outE().hasLabel("job Is").V().hasLabel("job").has("name","chef").path()
```

Tigergraph GSQL: /* SQL-style, 12 lines code for triplet expression */

```
CREATE QUERY areithjob(vertex<word> w) for graph test {
  SetAccum<node> @@nodeSet;
  SetAccum<edge> @@edgeSet;
  Start = {persion.*};
  Result = select j from Start::p - (jobIs:e) - job:j
         WHERE p.name == "areith" AND j.name == "chef"
       accum @@nodeSet += p,
       accum @@nodeSet += j,
       accum @@edgeSet + = e;
  print @@nodeSet;
  print @@edgeSet;
}
```

Even though readers may be subjective on intuitiveness (explainability) of the above GQL dialects, there are several things that hinder easy comprehension such as non-inline filtering, extensive chaining, or mixing up of SQL and C++ procedural programming styles.

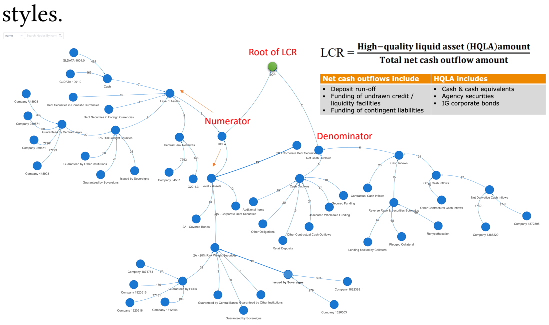

Figure 10: Explainable Data Modeling – Liquidity Risk Management & Attribution Analysis [22]

Explainability can also be reflected in data modeling. Though RDBMS and tables have been main-stream for decades, it can be a disaster to use dozens of tables to serve sophisticated business scenarios like liquidity risk management. Figure 10 shows a novel graph data modeling, where regulated key financial indicator LCR (Liquidity Coverage Ratio) formula is transformed to the graph, intuitively. This enables not only great explainability, but also exponentially accelerated computing of the LCR indicator (due to avoidance of the cartesian-product of joining dozens of tables) with high-density parallel processing, and attribution analysis which is essentially a back-tracing processing with dynamic filtering on the tree-like graph.

2.4 Graph-augmented: Velocity

The velocity aspect of graph XAI is crucial in enabling accelerated deep traversal during big data analytics. There are 3 novel techniques leveraged by Ultipa graph database in accelerating:

- High-density graph computing: adapting parallel computing to graph domain with optimizations to penetrate hotspot supernodes in conjunction with native-graph data structures.
- Data structure optimization: this has been discussed in the XAI's depth aspect, essentially vertex-edge adjacency data structures.
- Traversal boosting: this encompasses multiple facets, primarily optimized redesign of graph query and algorithm traversal logics, with the help of acceleration data structures.

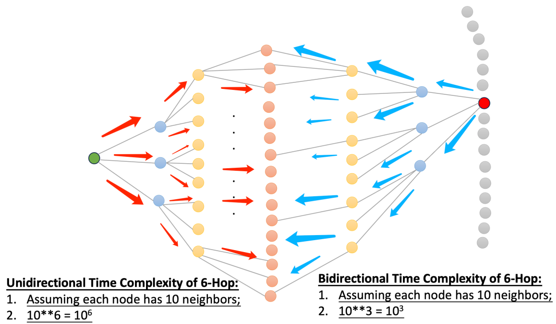

Figure 11: Bidirectional-boosted Path Finding

In Figure 11, a novel graph traversal method is illustrated, instead of traversing from one vertex only, the method allows traversal to be conducted parallelly from both ends. This would exponentially lower the traversal complexity, and the query will return as soon as common neighbors are found in the middle, and the overall theoretical query time-complexity can be exponentially lower. Additionally, in-memory data-structure is used to boost traversal speed (storing temporal neighborhood states). The empirical benchmark data shows that an overall acceleration of 40 times over K-hop neighborhood finding, and 160 time over path-finding are achieved (see Figure 12). As more cores are fired up for denser parallel processing, the acceleration effect is significant (32-core vs. 1-core: 7x for 3-hop, 25x for 6-hop, and 100x for Shortest-path. Note that adding more cores not always yield better performance, in shortest-path finding, 16-core turns out to be slightly faster than 32-core, as communications between more cores tend to add communication costs).

The performance gain of utilizing high-density parallel graph computing and boosted-traversal mechanism can significantly reduce system latency and increase system throughput per QPS and TPS. In real-world applications, Ultipa graph system incorporating these acceleration mechanisms can allow T+1 (1-day) batch processing tasks to be completed in real-time or near-real-time T+0 (same day) fashion, therefore opening up opportunities for broad spectrum business scenario realization (and acceleration).

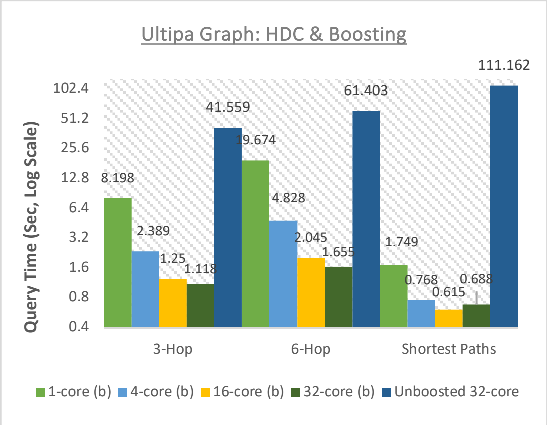

Figure 12: Bidirectional-boosted K-Hop & Path Finding On Billion-scale Data Set (Twitter-2010)

3. Conclusions and Future Work

Big data analysis and AI are undoubtedly integral to our technological landscape. However, several fundamental challenges demand our attention. In this paper, we have demonstrated that explainable AI (XAI) through graph augmentation can provide practical solutions to address key issues (A-D-E-V) in the field.

While significant progress has been made, several avenues remain underexplored. We recommend further investigation in the following areas:

- Dynamic Graph Data Modeling: Optimizing on dynamic graph data modeling based on evolving data and temporal user/query requirements.
- High-Scalable Graph Computing Architecture: Efficiently handling extra large-scale (zillion-scale) graphs.
- TP (Transactional Processing) and AP (Analytics Processing) Fusion: Bridging the gap between transactional and analytical workloads.

These research directions hold promise for future breakthroughs, and we anticipate their incorporation into production systems and scholarly publications.

Acknowledgements

We would like to thank our customers for offering inspirations and opportunities to realizing their relentless business scenarios with our cutting-edge graph XAI solutions, and big kudos to our teammates Lynsey, Zoey, Bin, Pearl, and countless others for their contribution, inputs, and insightful feedback.

References

[1] Rob Kitchin, Gavin McArdle. 2016 What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets. Big Data & Society, 2016, DOI: 10.1177/2053951716631130

[2] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In Conference on Human Factors in Computing Systems - Proceedings, Vol. 2018-April. Association for Computing Machinery. https://doi.org/10.1145/3173574.3174156

[3] Arlind Kadra, Sebastian Pineda Arango, Josif Grabocka. 2023 Breaking the Paradox of Explinaable Deep Learning, 2023, DOI: 10.48550/arXiv.2305.13072

[4] Jon Rueda, Janet D. Rodriguez, Iris P. Jounou, Joaquin Hortal-Carmona. 2022 Just accuracy? Proedual fairness demands explainability in AI-based medical resource allocations, 2022, https://doi.org/10.1007/s00146-022-01614-9

[5] Sarah Lebovitz, Hila Lifshitz-Assaf. 2021 Is AI Ground Truth Really True? The Dangers of Training and Evaluating AI Tools Based on Experts' Know-What, 2021, DOI: 10.25300/ MISQ /2021/16564

[6] Erendira Rendon, Roberto Alejo, Carlos Castorena, Frank J. Isidro-Ortega. 2020 Data Sampling Methods to Deal with Big Data Multi-Class Imbalance Problem, 2020, https://doi.org/10.3390/app10041276

[7] Leonid Kirievsky, Anotoly Kirievsky. 2004 Attribution Analysis: Issues Old and New, 2004, http://dx.doi.org/10.2139/ ssrn.632943

[8] Chittaranjan Andrade. 2021 The Inconvenient Truth About Convenience and Purposive Samples, 2021, https://doi.org/10.1177/0253717620977000

[9] Chad Vicknair, Michael Macias, Zhendong Zhao. 2010 A comparison of a graph database and a relational database: a data prevenancep perspective, 2010, https://doi.org/10.1145/1900008.1900067

[10] Ziniu Wu, Parimajan Negi, Mohammad Alizadeh, Tim Kraska. 2022 FactorJoin: A New Cardinality Estimation Framework for Join queries, 2022, arXiv:2212.05526v1

[11] Frank McSherry, Michael Isard and Derek G. Murray, 2015. Scalability! But at what COST. HotOS XV 2015. DOI: https://dl.acm.org/doi/10.5555/2831090.2831104

[12] Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. OSDI 2004.

[13] Shwet Ketu, Pramod Kumar Mishra, Sonali Agarwal, 2020. Performance Analysis of Distributed Computing Frameworks for Big Data Analytics: Hadoop vs Spark. doi: 10.13053/CyS-24-2-3401

[14] Benoit Dageville, Thierry Cruanes, Marcin Zukowski, Vadim Antonov, 2016. The Snowflake Elastic Data Warehouse. DOI: http://dx.doi.org/10.1145/2882903.2903741

[15] Lai Van Vo, Huong T.T. Le. 2023 From Hero to Zero – The Case of Silicon Valley Bank. Journal of Economics and Business 2023,https://ssrn.com/abstract=4394553

[16] Seongjoon Park, Seounghwan Oh, Hwangnam Kim. 2019 Performance Analysis of DAG-Based Cryptocurrency. IEEE ICC 2019, 10.1109/ICCW.2019.8756973

[17] Huan Yu Wu, Xin Wang, Chentao Yue, Hye-Young Paik, Salil S. Kanhere. 2022 Chain or DAG? Underlying data structures, architectures, topologies and consensus in distributed ledger technology: A review, taxonomy and research issues. Elsevier JSA 2022, https://doi.org/10.1016/j.sysarc.2022.102720

[18] Si Chen, Jinyu Zhang, Rui Shi, Jiaqi Yan. 2018 A Comparative Testing on Performance of Blockchain and Relational Datbase: Foundation of Applying Smart Technology into Current Business Systems. 2018, DOI:10.1007/978-3-319-91125-2_2

[19] Michal Klincewicz, Lily Frank. 2019 Consequences of unexplainable machine learning for the notions of a trusted doctor and patient autonomy. 2019, https://ceur-ws.org/Vol-2681/xaila2019-paper4.pdf

[20] Yavar Batthaee. 2018 The Artificial Intellgience Black Box and Failure of Intent and Causation. Harvard Journal of Law and Technology Vol 31, 2018,https://jolt.law.harvard.edu/assets/articlePDFs/v31/The-Artificial-Intelligence-Black-Box-and-the-Failure-of-Intent-and-Causation-Yavar-Bathaee.pdf

[21] Jean-Marie John-Mathews. 2021 Critical Empirical Study on Black-Box Explainability in AI. 42nd Int'l CIS 2021, https://hal.science/hal-03357663

[22] Ricky Sun, Jamie Chen. 2023 Designing Highly Scalable Graph Database Systems without Exponential Performance Degradation. BiDEDE'23, https://doi.org/10.1145/3579142.3594293

[23] Maciej Besta, Emanuel Peter, Robert Gerstenberger. 2019 Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries: Towards understanding modern graph processing, storage, and analytics. CS.DB 2019, https://doi.org/10.48550/arXiv.1910.09017

[24] Louis Jachiet, Pierre Geneves, Nils Gesbert, Nabil Layaida. 2020 On the Optimization of Recursive Relational Queries: Application to Graph Queries. SIGMOD'20 Portland, OR. DOI: 10.1445/3318464.3380567

[25] Potsawee manakuk, Adian Luise, Mark J.F. Gales. 2023. SELFCHECKGPT: Zero-Resoruce Black-Box Hallucination Detection for Generation LLM. 2023. arXiv:2303.08896v3

[26] Yur zhang, Yafu Li, Leyang Cui, Deng Cai. 2023. Siren's Song in the AI Ocean: A Survey on Hallucinations in LLMs. 2023. DOI:10.48550/arXiv:2309.01219

[27] Partha Pratim Ray. 2023. ChaptGPT: A comprehensive review on background applications, key challenges, bias, ethics, limitations and future scope. 2023. DOI:10.1016/j.iotcps. 2023.04.003

[28] Ricky Sun. 2024. The Essential Criteria of Graph Databases. Elsevier. 2024. ISBN: 978-0-443-14162-1

[29] Manish Kumar Abhishek and D. Rajeswara Rao. 2020. Dynamic Allocation of High Performance Computing Resources. Int'l Journal of Advanced Trends in Computer Science and Engineering. DOI: 10.30534/ijatcse/2020/159932020

[30] Muhammad Attahir Jibril, Alexander Baumstark, Kai-Uwe Sattler. 2022. Adaptive Update Handling for Graph HTAP. 2022 IEEE 38th International Conference on Data Engineering Workshops. ICDEW. DOI: 10.1109/ICDEW55742.2022.00007

[31] Liam Roditty, Virginia Vassilevska Williams. 2012. Approximating the Diameter of a Graph. cs.DS, 2012. arXiv:1207.3622v1

[32] Daniel Vela, Andrew Sharp. 2022. Temporal quality degradation in AI models. Scientific Reports, 2022. DOI: 10.1038/s41598-022-15245-z