SIGMOD Officers, Committees, and Awardees

Chair	Vice-Chair	Secretary/Treasurer
Angela Bonifati	Alexandra Meliou	Manos Atthanassoulis
LIRIS - Département Informatique	College of Information and Computer Science	College of Arts and Sciences
Université Claude Bernard Lyon 1	University of Massachusetts	Boston University
Lyon	Amherst, Massachusetts	Boston, Massachusetts
France	USA	USA
+33-4-72-44-58-24	+1-413-545-3788	+1-617-353-4602
Angela.bonifati <at> univ-lyon1.fr	ameli <at> cs.umass.edu	mathan <at> bu.edu

Jim Gray Doctoral Dissertation Award Committee:

Evaggelia Pitoura (chair), Angela Bonifati (co-chair), Sourav S Bhowmick, Daniel Kang, Georgia Koutrika, Supun Nakandala, Fatma Ozcan, Julia Stoyanovich, and Xiaofang Zhou

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Recipients of the award are the following:

Michael Stonebraker (1992)
David DeWitt (1995)
Serge Abiteboul (1998)
Rudolf Bayer (2001)
Ronald Fagin (2004)
Jennifer Widom (2007)
Umeshwar Dayal (2010)
Stefano Ceri (2013)
Gerhard Weikum (2016)
Anastasia Ailamaki (2019)
Dan Suciu (2022)
Carlo Zaniolo (2025)

Jim Gray (1993)
C. Mohan (1996)
Hector Garcia-Molina (1999)
Patricia Selinger (2002)
Michael Carey (2005)
Moshe Y. Vardi (2008)
Surajit Chaudhuri (2011)
Martin Kersten (2014)
Goetz Graefe (2017)
Beng Chin Ooi (2020)
Joseph M. Hellerstein (2023)

Philip Bernstein (1994)
David Maier (1997)
Rakesh Agrawal (2000)
Don Chamberlin (2003)
Jeffrey D. Ullman (2006)
Masaru Kitsuregawa (2009)
Bruce Lindsay (2012)
Laura Haas (2015)
Raghu Ramakrishnan (2018)
Alon Halevy (2021)
Samuel Madden (2024)

SIGMOD Systems Award

For technical contributions that have had significant impact on the theory or practice of large-scale data management systems.

Michael Stonebraker and Lawrence Rowe (2015); Martin Kersten (2016); Richard Hipp (2017); Jeff Hammerbacher, Ashish Thusoo, Joydeep Sen Sarma; Christopher Olston, Benjamin Reed, and Utkarsh Srivastava (2018); Xiaofeng Bao, Charlie Bell, Murali Brahmadesam, James Corey, Neal Fachan, Raju Gulabani, Anurag Gupta, Kamal Gupta, James Hamilton, Andy Jassy, Tengiz Kharatishvili, Sailesh Krishnamurthy, Yan Leshinsky, Lon Lundgren, Pradeep Madhavarapu, Sandor Maurice, Grant McAlister, Sam McKelvie, Raman Mittal, Debanjan Saha, Swami Sivasubramanian, Stefano Stefani, and Alex Verbitski (2019); Don Anderson, Keith Bostic, Alan Bram, Grg Burd, Michael Cahill, Ron Cohen, Alex Gorrod, George Feinberg, Mark Hayes, Charles Lamb, Linda Lee, Susan LoVerso, John Merrells, Mike Olson, Carol Sandstrom, Steve Sarette, David Schacter, David Segleau, Mario Seltzer, and Mike Ubell (2020); Michael Blanton, Adam Bolton, Bill Boroski, Joel Brownstein, Robert Brunner, Tamas Budavari, Sam Carliles, Jim Gray, Steve Kent, Peter Kunszt, Gerard Lemson, Nolan Li, Dmitry Medvedev, Jeff Munn, Deoyani Nandrekar-Heinis, Maria Nieto-Santisteban, Wil O'Mullane, Victor Paul, Don Slutz, Alex Szalay, Gyula Szokoly, Manu Taghizadeh-Popp, Jordan Raddick, Bonnie Souter, Ani Thakar, Jan Vandenberg, Benjamin Alan Weaver, Anne-Marie Weijmans, Sue Werner, Brian Yanny, Donald York, and the SDSS collaboration (2021); Michael Armbrust, Tathagata Das, Ankur Dave, Wenchen Fan, Michael J. Franklin, Huaxin Gao, Maxim Gekk, Ali Ghodsi, Joseph Gonzalez, Liang-Chi Hsieh, Dongjoon Hyun, Hyukjin Kwon, Xiao Li, Cheng Lian, Yanbo Liang, Xiangrui Meng, Sean Owen, Josh Rosen, Kousuke Saruta, Scott Shenker, Ion Stoica, Takuya Ueshin, Shivaram Venkataraman, Gengliang Wang, Yuming Wang, Patrick Wendell, Reynold Xin, Takeshi Yamamuro, Kent Yao, Matei Zaharia, Ruifeng Zheng, and Shixiong Zhu (2022); Aljoscha Krettek, Andrey Zagrebin, Anton Kalashnikov, Arvid Heise, Asterios Katsifodimos, Jiangji (Becket) Qin, Benchao Li, Bowen Li, Caizhi Weng, ChengXiang Li, Chesnay Schepler, Chiwan Park, Congxian Qiu, Daniel Warneke, Danny Cranmer, David Anderson, David Morávek, Dawid Wysakowicz, Dian Fu, Dong Lin, Eron Wright, Etienne Chauchot, Fabian Hueske, Fabian Paul, Feng Wang, Gabor Somogyi, Gary Yao, Godfrey He, Greg Hogan, Guowei Ma, Gyula Fora, Haohui Mai, Henry Saputra, Hequn Cheng, Igal Shilman, Ingo Bürk, Jamie Grier, Jark Wu, Jincheng Sun, Jing Ge, Jing Zhang, Jingsong Lee, Junhan Yang, Konstantin Knauf, Kostas Kloudas, Kostas Tzoumas, Kete (Kurt) Young, Leonard Xu, Lijie Wang, Lincoln Lee, Lungu Andra, Martijn Visser, Marton Balassi, Matthias J. Sax, Matthias Pohl, Matyas Orhidi, Maximilian Michels, Nico Kruber, Niels Basjes, Paris Carbone, Piotr Nowojski, Qingsheng Ren, Robert Metzger, Roman Khachatryan, Rong Rong, Rui Fan, Rui Li, Sebastian

Schelter, Seif Haridi, Sergey Nuyanzin, Seth Wiesman, Shaoxuan Wang, Shengkai Fang, Shuyi Chen, Sihua Zhou, Stefan Richter, Stephan Ewen, Theodore Vasiloudis, Thomas Weise, Till Rohrmann, Timo Walther, Tzu-Li (Gordon) Tai, Ufuk Celebi, Vasiliki Kalavri, Volker Markl, Wei Zhong, Weijie Guo, Xiaogang Shi, Xiaowei Jiang, Xingbo Huang, Xingcan Cui, Xintong Song, Yang Wang, Yangze Guo, Yingjie Cao, Yu Li, Yuan Mei, Yun Gao, Yun Tang, Yuxia Luo, Zhijiang Wang, Zhipeng Zhang, Zhu Zhu, Zili Chen (2023); Zhaojing Luo, Beng Chin Ooi, Wei Wang, Meihui Zhang, Qingchao Cai, Shaofeng Cai, Gang Chen, Tien Tuan Anh Dinh, Jinyang Gao, Qian Lin, Shicong Lin, Kee Yuan Ngiam, Gene Yan Ooi, Moaz Reyad, Kian-Lee Tan, Anthony K. H. Tung, Sheng Wang, Yuncheng Wu, Zhongle Xie, Naili Xing, Rulin Xing, Wanqi Xue, Sai Ho Yeung, James Yip, Lingze Zeng, Zhaoqi Zhang, Kaiping Zheng, Lei Zhu, Ji Wang (2024); James C. Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, Wilson Hsieh, Sebastian Kanthak, Alexander Lloyd, Sergey Melnik, David Mwaura, Sean Quinlan, Lindsay Rolig, Yasushi Saito, Michal Szymaniak, Christopher Taylor, Ruth Wang, Dale Woodford, David F. Bacon, Shannon Bales, Nico Bruno, Brian F. Cooper, Adam Dickinson, Campbell Fraser, Milind Joshi, Eugene Kogan, Rajesh Rao, David Shue, Marcel van der Holst, Cliff Frey, Damian Reeves, Steve Middlekauff, Mert Akdere, Ben Vandiver, Dan Glick, David Ziegler, Alex Khesin, Dave Weissman, Todd Lipcon, Sean Dorward, Eric Veach (2025).

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)
Jeffrey Ullman (1996)
Raghu Ramakrishnan (1999)
Daniel Rosenkrantz (2001)
Surajit Chaudhuri (2004)
Hans-Jörg Schek (2007)
David Lomet (2010)
H.V. Jagadish (2013)
Samuel Madden (2016)
Ahmed Elmagarmid (2019)
Stratos Idreos (2020)
Dennis Shasha (2020)
K. Selcuk Candan (2023)
Hector Munoz-Avila & Sylvia Spengler (2025)

Gio Wiederhold (1995)
Avi Silberschatz (1997)
Michael Carey (2000)
Richard Snodgrass (2002)
Hongjun Lu (2005)
Klaus R. Dittrich (2008)
Gerhard Weikum (2011)
Kyu-Young Whang (2014)
Yannis E. Ioannidis (2017)
Philipe Bonnet (2020)
Stefan Manegold (2020)
Divesh Srivastava (2021)
Sihem Amer-Yahia (2024)

Yahiko Kambayashi (1995)
Won Kim (1998)
Laura Haas (2000)
Michael Ley (2003)
Tamer Özsu (2006)
Beng Chin Ooi (2009)
Marianne Winslett (2012)
Curtis Dyreson (2015)
Z. Meral Özsoyoğlu (2018)
Juliana Freire (2020)
Ioana Manolescu (2020)
Christian S. Jensen (2022)

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* Recipients of the award are the following:

- **2006** *Winner*: Gerome Miklau. *Honorable Mentions*: Marcelo Arenas and Yanlei Diao
- **2007** *Winner*: Boon Thau Loo. *Honorable Mentions*: Xifeng Yan and Martin Theobald
- **2008** *Winner*: Ariel Fuxman. *Honorable Mentions*: Cong Yu and Nilesh Dalvi
- **2009** *Winner*: Daniel Abadi. *Honorable Mentions*: Bee-Chung Chen and Ashwin Machanavajjhala
- **2010** *Winner:* Christopher Ré. *Honorable Mentions*: Soumyadeb Mitra and Fabian Suchanek
- **2011** *Winner*: Stratos Idreos. *Honorable Mentions*: Todd Green and Karl Schnaitterz
- **2012** *Winner*: Ryan Johnson. *Honorable Mention*: Bogdan Alexe
- **2013** *Winner*: Sudipto Das, *Honorable Mention*: Herodotos Herodotou and Wenchao Zhou
- **2014** *Winners*: Aditya Parameswaran and Andy Pavlo.
- **2015** *Winner*: Alexander Thomson. *Honorable Mentions*: Marina Drosou and Karthik Ramachandra
- **2016** *Winner*: Paris Koutris. *Honorable Mentions*: Pinar Tozun and Alvin Cheung
- **2017** *Winne*r: Peter Bailis. *Honorable Mention*: Immanuel Trummer
- **2018** *Winne*r: Viktor Leis. *Honorable Mention*: Luis Galárraga and Yongjoo Park
- **2019** *Winne*r: Joy Arulraj. *Honorable Mention*: Bas Ketsman
- **2020** *Winner:* Jose Faleiro. *Honorable Mention:* Silu Huang
- **2021** *Winner:* Huanchen Zhang, *Honorable Mentions:* Erfan Zamanian, Maximilian Schleich, and Natacha

Crooks

- **2022** *Winner:* Chenggang Wu, *Honorable Mentions:* Pingcheng Ruan and Kexin Rong
- **2023** *Winner:* Supun Nakandala, *Honorable Mentions:* Benjamin Hilprecht and Zongheng Yang
- **2024** *Winner:* Daniel Kang, *Honorable Mentions:* Wei Dong, Jialin Ding, and Yisu Remy Wang
- **2025** *Winner:* Peng Li, *Honorable Mentions:* Xuanhue Zhou, Aecio Santos, and Meghdad Kurmanji

A complete list of all SIGMOD Awards is available at: **https://sigmod.org/sigmod-awards/**

[Last updated: October 1, 2025]

Editor's Notes

Welcome to the December 2025 issue of the ACM SIGMOD Record!

This issue starts with the Database Principles column featuring an article by Tao and Wang focusing on the theoretical foundation of sampling for natural joins, which are fundamental to database systems. The authors present an elegant and powerful box-tree technique that has been developed by the database-theory community in recent years. The technique yields simple algorithms and data structures, whose theoretical guarantees match the state of the art, for solving the problems of join reporting, small-delay enumeration, and join sampling. The box-tree technique is based on so-called split theorems, which uncover a combinatorial and intrinsically geometric structure underlying natural joins. The article also surveys related work, discusses alternative approaches to join reporting and sampling, and points out directions for future research.

The Reminiscences on Influential Papers column, edited by Pınar Tözün, presents contributions by Genoveva Vargas-Solar, Thomas Heinis, Alexander Boehm, and Pamela Delgado.

The Advice to Mid-Career Researchers column presents a contribution by Khuzaima Daudjee. The reflections shared in the article focus on four themes connected to the author's personal experiences: system building, mentorship, micro-visits, and service. The article provides advice on many aspects of the mid-career stage of life, offering the perspective of looking for what is meaningful and interesting, as well as for new opportunities, including the one of building yourself along the way on this path. The author expresses a hope that some of the reflections shared in the article would also be a useful read for colleagues who are currently at earlier points in their careers than mid-career researchers.

The DBrainstorming column, whose goal is to discuss new and potentially controversial ideas that might be of interest and benefit to the research community, features an article by Raul Castro Fernandez that focuses on understanding and designing data ecosystems. The research agenda presented in the article aims to uncover principles that drive dataflows and to design comprehensive interventions to steer them positively. The author shares preliminary and ongoing work on this topic, outlines opportunities for the database community, and maps out the road ahead in this research space.

The Distinguished Profiles column features an interview with Angela Bonifati, Distinguished Professor of Computer Science at Lyon 1 University, Head of the Database Research Group at the CNRS Liris Research Lab, and a senior member of the Institut Universitaire de France. In the interview, Angela talks about her research focus and professional arc over the years, about the state of the art and future of graph databases, and about her work plans for the prestigious ERC Advanced Grant that she has just received. She discusses working in the industry and her experience with collaborations, as well as research in Europe versus in the rest of the world. Angela also outlines her vision as the new chair of SIGMOD, provides advice on mentoring, and shares her thoughts and experiences concerning work-life balance.

The issue closes with the Open Forum column, which presents an article by Khan and colleagues reporting on the discussion of retrieval-augmented generation (RAG) that was held by a panel at the LLM+Vector Data'25 workshop held jointly with the ICDE-25 conference in Hong Kong, China. The discussion in the panel focused on emerging RAG opportunities at the intersection of data science, data engineering, and data-centric AI/ML methodologies. The panel also outlined open challenges for

real-world LLM-RAG deployment, with the hope that the discussion would inspire future work on the domain's emerging data-management issues.

On behalf of the SIGMOD Record Editorial board, I hope that you enjoy reading the December 2025 issue of the SIGMOD Record!

Your submissions to the SIGMOD Record are welcome via the submission site:
https://mc.manuscriptcentral.com/sigmodrecord

Prior to submission, please read the Editorial Policy on the SIGMOD Record's website:
https://sigmodrecord.org/sigmod-record-editorial-policy/

Rada Chirkova

December 2025

Past SIGMOD Record Editors:

Yanlei Diao (2014-2019)
Mario Nascimento (2005–2007)
Jennifer Widom (1995–1996)
Jon D. Clark (1984–1985)
Randall Rustin (1974-1975)
Ioana Manolescu (2009-2013)
Ling Liu (2000–2004)
Arie Segev (1989–1995)
Thomas J. Cook (1981–1983)
Daniel O'Connell (1971–1973)
Alexandros Labrinidis (2007–2009)
Michael Franklin (1996–2000)
Margaret H. Dunham (1986–1988)
Douglas S. Kerr (1976-1978)
Harrison R. Morse (1969)

Split Theorems for Join Reporting and Sampling

Yufei Tao
Chinese University of Hong Kong
Hong Kong, China
taoyf@cse.cuhk.edu.hk

Ru Wang
Chinese University of Hong Kong
Hong Kong, China
rwang21@cse.cuhk.edu.hk

ABSTRACT

The *box-tree technique* is an elegant and powerful tool for natural join processing. Recently developed by the database theory community, it yields simple algorithms and data structures for solving three fundamental problems — *join reporting*, *small-delay enumeration*, and *join sampling* — with performance matching the best known bounds, up to polylogarithmic factors. This article presents the technique's theoretical foundation in a form accessible to the broader database community. At the core of this foundation are the so-called *split theorems*, which uncover a combinatorial and intrinsically geometric structure underlying natural joins. Two versions of these theorems are proved using elementary, self-contained arguments.

1. INTRODUCTION

This article studies *natural joins*. Consider $t \geq 2$ relations $R_1, R_2, ..., R_t$. Conceptually, a natural join first computes the Cartesian product $R_1 \times R_2 \times ... \times R_t$ and then selects from this product only the tuples satisfying the SQL-style condition "$R_i.A = R_j.A$" for every $1 \leq i < j \leq t$ and every common attribute A of R_i and R_j. This intuitive yet informal definition will suffice for the discussion until Section 2, where a formal definition of the natural join will be introduced.

Natural joins are fundamental to database systems for two main reasons. First, they are indispensable for answering multi-relation queries and serve as the backbone of most query execution plans. Second, they are computationally expensive operations that often dominate the cost of query processing. The design and analysis of join algorithms have long been a central topic in database theory. A solid understanding of their theoretical foundation provides valuable insights that guide the development of high-performance systems.

Join sampling is the operation that draws uniformly at random a tuple from the result of a natural join. The sample must be independent from those returned by all previous join sampling operations — this requirement prevents the "cheating" solution that keeps only one sample element from the join result and returns it for every single operation.

Recent years have witnessed considerable research interest in join sampling, a trend that can be attributed to at least three key factors. First, with the rapid growth in database volumes, the result size of a natural join increases combinatorially. This fundamentally changes the way users interact with databases: it is no longer practical to materialize or display the entire join result. Instead, users are increasingly turning to sampling-based methods to explore large query outputs. Second, many analytical tasks — especially those of a statistical nature — can actually be carried out effectively on samples rather than full join results. Third, join sampling can usually be performed significantly faster than producing the complete join result.

A main objective of this article is to present an elegant *box-tree technique* for join sampling that has been developed by the database theory community in recent years. The technique is simple enough to be taught in a graduate-level course, yet powerful enough to yield sampling algorithms and data structures with theoretical guarantees that match the state of the art up to polylogarithmic factors. The technique finds its root in two *split theorems*, which reveal a beautiful mathematical structure — specifically, a *geometric* structure — underlying natural joins. We will provide self-contained proofs of both theorems using elementary arguments.

The remainder of this article is organized as follows. Section 2 lays the foundation for our discussion by formalizing natural joins, degree constraints, and the problems of join reporting and sampling. Section 3 introduces the polymatroid bound, a key parameter for characterizing the performance of join reporting and sampling algorithms. Section 4 presents the article's main protagonist — the split theorems — and elaborates on their

derivation. Section 5 demonstrates how these theorems can be applied to derive algorithms that perform join sampling and reporting with strong efficiency guarantees. Section 6 surveys related work and discusses alternative approaches to join reporting and sampling. Finally, Section 7 concludes with a summary and some remarks for future research.

2. FORMALIZATION

Math Conventions. The symbol $\mathbb{N}$ represents the set of integers. For an integer $x \geq 1$, the notation $[x]$ denotes the set $\{1, 2, ..., x\}$; as a special case, $[0]$ gives the empty set. Every logarithm $\log(\cdot)$ has base 2.

Tuples and Relations. Let $\mathbf{att}$ be a finite set, where each element is called an *attribute*. For a non-empty set $\mathcal{X} \subseteq \mathbf{att}$ of attributes, a *tuple* over $\mathcal{X}$ is a function $\boldsymbol{u} : \mathcal{X} \to \mathbb{N}$. For any non-empty subset $\mathcal{Y} \subseteq \mathcal{X}$, we define $\boldsymbol{u}[\mathcal{Y}]$ — called the *projection* of $\boldsymbol{u}$ on $\mathcal{Y}$ — as the tuple $\boldsymbol{v}$ over $\mathcal{Y}$ satisfying $\boldsymbol{v}(Y) = \boldsymbol{u}(Y)$ for every attribute $Y \in \mathcal{Y}$. A *relation* R is a set of tuples over the same set $\mathcal{Z}$ of attributes; we refer to $\mathcal{Z}$ as the *schema* of R and represent it as $schema(R)$.

Joins. We define a *join* as a set $\mathcal{Q}$ of relations with distinct schemas. Let $schema(\mathcal{Q})$ be the union of the attributes of the relations in $\mathcal{Q}$, i.e., $schema(\mathcal{Q}) = \bigcup_{R \in Q} schema(R)$. The result of $\mathcal{Q}$ is a relation over $schema(\mathcal{Q})$ formalized as:

$$join(\mathcal{Q}) = \{\text{tuple } \boldsymbol{u} \text{ over } schema(\mathcal{Q}) \mid \forall R \in \mathcal{Q} : \boldsymbol{u}[schema(R)] \in R\}. \quad (1)$$

Degrees. Let R be a relation. For subsets $\mathcal{X}$ and $\mathcal{Y}$ of $schema(R)$ satisfying $\mathcal{X} \subset \mathcal{Y}$ — note: $\mathcal{X}$ is a *proper* subset of $\mathcal{Y}$ — define the $(\mathcal{Y}|\mathcal{X})$*-degree of* R as

$$\deg_{\mathcal{Y}|\mathcal{X}}(R) = \max_{\text{tuple } \boldsymbol{u} \text{ over } \mathcal{X}} \left| \left\{ \boldsymbol{v}[\mathcal{Y}] \mid \boldsymbol{v} \in R, \boldsymbol{v}[\mathcal{X}] = \boldsymbol{u} \right\} \right| \quad (2)$$

For an intuitive explanation, imagine grouping the tuples of R by $\mathcal{X}$ and counting, for each group, how many *distinct* $\mathcal{Y}$-projections therein. Then, the $(\mathcal{Y}|\mathcal{X})$-degree of R equals the maximum count of all groups. Note that, when $\mathcal{X} = \emptyset$, then $\deg_{\mathcal{Y}|\mathcal{X}}(R)$ is simply $|\Pi_{\mathcal{Y}}(R)|$ where Π is the "projection" operator in relational algebra. If in addition $\mathcal{Y} = schema(R)$, then $\deg_{\mathcal{Y}|\mathcal{X}}(R) = |R|$.

Degree Constraints. We define a *degree constraint* as a triple $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}})$ where $\mathcal{X}$ and $\mathcal{Y}$ are subsets of $\mathbf{att}$ satisfying $\mathcal{X} \subset \mathcal{Y}$, and $N_{\mathcal{Y}|\mathcal{X}} \geq 1$ is an integer. A relation R is said to *guard* the constraint if

$$\mathcal{Y} \subseteq schema(R), \text{ and } \deg_{\mathcal{Y}|\mathcal{X}}(R) \leq N_{\mathcal{Y}|\mathcal{X}}.$$

Let DegC be a set of degree constraints. It defines a *constraint dependency graph* $\mathcal{G}$ as follows:

- The vertex set of $\mathcal{G}$ consists of every attribute $Y \in \mathbf{att}$ such that $Y \in \mathcal{Y}$ for at least one constraint $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}}) \in \mathsf{DegC}$.
- For each degree constraint $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}}) \in \mathsf{DegC}$ with $\mathcal{X} \neq \emptyset$, the graph $\mathcal{G}$ has a directed edge from X to Y for every pair $(X, Y) \in \mathcal{X} \times (\mathcal{Y} \setminus \mathcal{X})$.

If $\mathcal{G}$ is an acyclic graph, DegC is said to be *acyclic*; otherwise, DegC is *cyclic*.

The set DegC is said to be *clean* if it does not have two constraints $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}})$ and $(\mathcal{X}', \mathcal{Y}', N_{\mathcal{Y}'|\mathcal{X}'})$ with $\mathcal{X} = \mathcal{X}'$ and $\mathcal{Y} = \mathcal{Y}'$. Intuitively, when two such constraints exist — w.l.o.g., assume $N_{\mathcal{Y}|\mathcal{X}} \leq N_{\mathcal{Y}'|\mathcal{X}'}$ — the constraint $(\mathcal{X}', \mathcal{Y}', N_{\mathcal{Y}'|\mathcal{X}'})$ is redundant and can be removed from DegC.

A join $\mathcal{Q}$ is said to *conform to* DegC — written as $\mathcal{Q} \models \mathsf{DegC}$ — if the two following conditions hold:

- For each relation $R \in \mathcal{Q}$, the set DegC contains a constraint $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}})$ where $\mathcal{X} = \emptyset$, $\mathcal{Y} = schema(R)$, and $N_{\mathcal{Y}|\mathcal{X}} = |R|$. This constraint is referred to as a *cardinality constraint*.
- Every constraint in DegC is guarded by at least one relation in $\mathcal{Q}$.

As an important special case, when DegC consists purely of cardinality constraints, $\mathcal{G}$ has no edges and, therefore, is trivially acyclic. In general, one should avoid the misconception that "an acyclic $\mathcal{G}$ implies $\mathcal{Q}$ being an acyclic join". These two notions of acyclicity are unrelated. Readers unfamiliar with the concept of an acyclic join may refer to [1, Chapter 6.4].

Join Reporting and Sampling. Given (i) a join $\mathcal{Q}$ whose $schema(\mathcal{Q})$ has a constant size (this article focuses on "data complexity") and (ii) a clean, acyclic set DegC of degree constraints such that $\mathcal{Q} \models \mathsf{DegC}$, we will discuss three representative problems:

- *Join reporting.* Compute the join result $join(\mathcal{Q})$, as defined in (1).
- *Small-delay enumeration.* Given a positive integer Δ, preprocess $\mathcal{Q}$ into a data structure that enumerates $join(\mathcal{Q})$ with a *delay* of Δ. Specifically, within $O(\Delta)$ time, the enumeration algorithm must report a tuple in $join(\mathcal{Q})$ or declare $join(\mathcal{Q}) = \emptyset$. Thereafter, within each subsequent $O(\Delta)$ time period, the algorithm must either report a new tuple in $join(\mathcal{Q})$ or declare the end of enumeration.
- *Join sampling.* Preprocess $\mathcal{Q}$ into a data structure that supports sampling a tuple uniformly at random from $join(\mathcal{Q})$. Repeated invocations of the

sampling operation should produce mutually independent samples.

These problems have been studied extensively and often in isolation (see Section 6). As shown in Section 5, they actually share a common geometric structure.

We reserve the symbols IN and OUT for $\sum_{R\in\mathcal{Q}}|R|$ and $|join(\mathcal{Q})|$, respectively, which we refer to as the *input size* and *output size* of $\mathcal{Q}$.

3. POLYMATROID BOUNDS

Fix an arbitrary clean, acyclic set DegC of degree constraints. Define

$$\mathcal{V} = \bigcup_{(\mathcal{X},\mathcal{Y},N_{\mathcal{Y}|\mathcal{X}})\in\mathsf{DegC}} \mathcal{Y}; \tag{3}$$

namely, $\mathcal{V}$ is the vertex set in the constraint dependency graph defined by DegC.

We formulate the *XY-space* of DegC as

$$\text{space}(\mathsf{DegC}) = \{(\mathcal{X},\mathcal{Y}) \mid (\mathcal{X},\mathcal{Y},N_{\mathcal{Y}|\mathcal{X}}) \in \mathsf{DegC}\}.$$

Henceforth, the symbol π will be used to reference a pair $(\mathcal{X},\mathcal{Y})$ in space(DegC). Accordingly, we use

- N_π to represent the value of $N_{\mathcal{Y}|\mathcal{X}}$ in the corresponding constraint $(\mathcal{X},\mathcal{Y},N_{\mathcal{Y}|\mathcal{X}}) \in \mathsf{DegC}$;
- (π, N_π) to represent the constraint $(\mathcal{X},\mathcal{Y},N_{\mathcal{Y}|\mathcal{X}})$;
- $diffatt(\pi)$ to represent the set $\mathcal{Y}\setminus\mathcal{X}$.

The introduction of XY-space permits us to define vectors over it. Specifically, every such vector W has a component W_π, which is a real value, for each pair $\pi \in$ space(DegC). The vector W is said to be a *fractional XY-pair cover* of DegC if it satisfies two conditions:

- $0 \le W_\pi \le 1$ for every $\pi \in \text{space}(\mathsf{DegC})$.
- $\sum_{\substack{\pi\in\text{space}(\mathsf{DegC})\\ \text{s.t. } A\in diffatt(\pi)}} W_\pi \ge 1$ for every $A \in \mathcal{V}$.

Every fractional XY-pair cover W is associated with a *polymatroid value* defined as

$$polymat_W \;=\; \prod_{\pi\in\text{space}(\mathsf{DegC})} N_\pi^{W_\pi}. \tag{4}$$

We now introduce a crucial lemma that relates polymatroid values to joins.

LEMMA 1 ([18, 21]). *Let $\mathcal{Q}$ be an arbitrary join conforming to* DegC. *Given any fractional XY-pair cover W of* DegC*, it holds that $|join(\mathcal{Q})| \le polymat_W$.*

An *optimal fractional XY-pair cover* of DegC is a fractional XY-pair cover W^* minimizing (4). Lemma 1 assures us that if $\mathcal{Q} \models \mathsf{DegC}$, then the output size of $\mathcal{Q}$ is at most the polymatroid value determined by W^*, which we refer to as the *polymatroid bound* of $\mathcal{Q}$.

Polymatroid bounds are asymptotically tight in the worst case. To explain, let us call a clean, acyclic set of degree constraints *realizable* if there exists at least one join that conforms to it. For any realizable set DegC of degree constraints, it is possible [27] to construct a join $\mathcal{Q}$ satisfying the conditions that $\mathcal{Q} \models \mathsf{DegC}$ and $|join(\mathcal{Q})| = \Omega(polymat_{W^*})$, where W^* is an optimal fractional XY-pair cover of DegC.

Relevance to the AGM Bound. Before proceeding, we draw relevance from the polymatroid bound to the "AGM bound" [4], which is well known in the database community nowadays. As will be clear shortly, the concepts introduced earlier in this section generalize those underlying the AGM bound.

Given a join $\mathcal{Q}$, we use $\mathcal{H}$ to denote the hypergraph $(\mathcal{V}_\mathcal{H}, \mathcal{E}_\mathcal{H})$ where

$$\begin{aligned}\mathcal{V}_\mathcal{H} &= schema(\mathcal{Q})\\ \mathcal{E}_\mathcal{H} &= \{schema(R) \mid R\in\mathcal{Q}\}.\end{aligned}$$

For each hyperedge $e \in \mathcal{E}_\mathcal{H}$, let R_e represent the (only) relation $R \in \mathcal{Q}$ with $e = schema(R)$.

Next, we regard $E_\mathcal{H}$ as a "space" and define vectors over it. Every such vector $\mathscr{W}$ has a component $\mathscr{W}_e$, which is a real value, for each hyperedge $e \in \mathcal{E}_\mathcal{H}$. The vector $\mathscr{W}$ is said to be a *fractional edge cover* of $\mathcal{H}$ if it satisfies two conditions:

- $0 \le \mathscr{W}_e \le 1$ for every $e \in \mathcal{E}_\mathcal{H}$.
- $\sum_{e\in\mathcal{E}_\mathcal{H}: A\in e} \mathscr{W}_e \ge 1$ for every $A \in \mathcal{V}_\mathcal{H}$.

Every fractional edge cover $\mathscr{W}$ is associated with an *AGM value* defined as

$$AGM_\mathscr{W} \;=\; \prod_{e\in\mathcal{E}_\mathcal{H}} |R_e|^{\mathscr{W}_e}. \tag{5}$$

As proved in [4], the inequality $|join(\mathcal{Q})| \le AGM_\mathscr{W}$ holds for any fractional edge cover $\mathscr{W}$ of $\mathcal{H}$. An *optimal fractional edge cover* of $\mathcal{H}$ is a fractional edge cover $\mathscr{W}^*$ minimizing (5). The *AGM bound* of $\mathcal{Q}$ is defined as the AGM value of $\mathscr{W}^*$.

The polymatroid bound of $\mathcal{Q}$ specializes to its AGM bound when DegC comprises *only* the cardinality constraints from $\mathcal{Q}$, that is, $\mathsf{DegC} = \{(\emptyset, schema(R), |R|) \mid R \in \mathcal{Q}\}$. In this case, the XY-space space(DegC) — which is $\{(\emptyset, schema(R)) \mid R \in \mathcal{Q}\}$ — has a one-one correspondence to $\mathcal{E}_\mathcal{H}$. Accordingly, every vector

W over space(DegC) corresponds to a distinct vector $\mathscr{W}$ over $\mathcal{E}_{\mathcal{H}}$ and vice versa: for each $R \in \mathcal{Q}$, the two vectors satisfy $W_\pi = \mathscr{W}_e$ where $\pi = (\emptyset, schema(R))$ and $e = schema(R)$. Hence, W is a fractional XY-pair cover of DegC if and only if the corresponding $\mathscr{W}$ is a fractional edge cover of $\mathcal{E}_{\mathcal{H}}$. It thus follows from a comparison of (4) and (5) that the polymatroid bound of $\mathcal{Q}$ is simply its AGM bound.

In general, when DegC contains more than just cardinality constraints, the polymatroid bound of a join $\mathcal{Q} \models$ DegC is never higher than its AGM bound.[1] On the other hand, it is possible [18, 27] to construct joins where the polymatroid bound is significantly less than the AGM bound.

4. SPLIT THEOREMS

In this section, we fix a join $\mathcal{Q}$, and a clean, acyclic set DegC of degree constraints such that $\mathcal{Q} \models$ DegC. Denote by $\mathcal{G}$ the constraint dependency graph of DegC, and by $\mathcal{V}$ the vertex set of $\mathcal{G}$ (see (3)); note that $\mathcal{V} = schema(\mathcal{Q})$. Set $d = |\mathcal{V}|$, and let the attribute sequence

$$A_1, A_2, ..., A_d \tag{6}$$

be an (arbitrary) topological order of $\mathcal{G}$. This is an important order that we will rely on in our discussion. Such an order definitely exists because $\mathcal{G}$ is acyclic. It is worth emphasizing that the order includes every attribute in $schema(\mathcal{Q})$.

Recall that, for each $\pi = (\mathcal{X}, \mathcal{Y})$ in space(DegC), the notation (π, N_π) represents the constraint $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}})$. We denote by R_π the relation in $\mathcal{Q}$ guarding this constraint. Such a relation must exist because $\mathcal{Q} \models$ DegC. In case π is guarded by multiple relations in $\mathcal{Q}$, any one of them may serve as R_π.

4.1 "Boxing" Everything

We use the term *box* to refer to a rectangle B in $\mathbb{N}^d$ of the form $[x_1, y_1] \times [x_2, y_2] \times ... \times [x_d, y_d]$. For any $i \in [d]$, let $B(A_i)$ represent the interval $[x_i, y_i]$, i.e., the projection of B onto the i-th dimension. Particularly, we say that $B(A_i)$ is a *singleton interval* if $x_i = y_i$. If $B(A_i)$ is a singleton interval on all $i \in [d]$, we call B a *singleton box*.

For each relation $R \in \mathcal{Q}$, we define

$$R(B) = \{\text{tuple } \boldsymbol{u} \in R \mid \forall A_i \in schema(R) : \boldsymbol{u}(A_i) \in B(A_i)\}. \tag{7}$$

[1]Given an optimal fractional edge cover $\mathscr{W}^*$ of $\mathcal{H}$, one can always construct a fractional XY-pair cover W of DegC with $polymat_W = AGM_{\mathscr{W}^*}$. Hence, $polymat_{W^*} \leq polymat_W = AGM_{\mathscr{W}^*}$, where W^* is an optimal fractional XY-pair cover of DegC.

Intuitively, $R(B)$ is the subset of tuples in R that are "selected" by B. Accordingly, define

$$\mathcal{Q}(B) = \{R(B) \mid R \in \mathcal{Q}\}. \tag{8}$$

Note that $\mathcal{Q}(B)$ is a join.

Next, we build a set of degree constraints — denoted as DegC(B) — dedicated to $\mathcal{Q}(B)$. Consider an arbitrary constraint $(\pi, N_\pi) \in$ DegC where $\pi = (\mathcal{X}, \mathcal{Y})$. Define

$$N_\pi(B) = \deg_{\mathcal{Y}|\mathcal{X}}(R_\pi(B)) \tag{9}$$

where — let us recall — R_π is the relation in $\mathcal{Q}$ guarding (π, N_π). We can now express DegC(B) formally as:

$$\text{DegC}(B) = \{(\pi, N_\pi(B)) \mid (\pi, N_\pi) \in \text{DegC}\}.$$

The facts below are easy to verify:

- DegC(B) and DegC share the same XY-space.
- Any fractional XY-pair cover of DegC is also a fractional XY-pair cover of DegC(B).
- $\mathcal{Q}(B)$ conforms to DegC(B).

Take any fractional XY-pair cover W of DegC (which must also be a fractional XY-pair cover of DegC(B)). The *polymatroid value of W on B* is defined as

$$polymat_W(B) = \prod_{\pi \in \text{space}(\text{DegC}(B))} (N_\pi(B))^{W_\pi}$$

which can be rewritten as

$$polymat_W(B) = \prod_{\pi \in \text{space}(\text{DegC})} (N_\pi(B))^{W_\pi} \tag{10}$$

because space(DegC) = space(DegC(B)). By Lemma 1, we must have $|join(\mathcal{Q}(B))| \leq polymat_W(B)$.

4.2 The Main Theorems

Given a non-singleton box B, we define its *split dimension* as the smallest $\sigma \in [d]$ such that $B(A_\sigma)$ is not a singleton interval. A set of boxes $B_1, B_2, ..., B_s$ (where $s \geq 2$) is said to *refine* B if

- $B_1, B_2, ..., B_s$ are mutually disjoint, and their union is B;
- for every $i \in [d] \setminus \{\sigma\}$, it holds that $B_1(A_i) = B_2(A_i) = ...B_s(A_i) = B(A_i)$.

We now present the first split theorem:

THEOREM 2. (SIMPLE POLYMATROID SPLIT THEOREM) *Consider a clean, acyclic* DegC *of degree constraints and a join $\mathcal{Q}$ conforming to* DegC. *Denote by W an arbitrary fractional XY-pair cover of* DegC. *Let B be a non-singleton box; and let B_1, B_2*

be any two boxes refining B. We must have $polymat_W(B_1) + polymat_W(B_2) \leq polymat_W(B)$.

As shown in Section 5, the above theorem is already sufficient to obtain join reporting and sampling algorithms with excellent guarantees. Nevertheless, it falls short of revealing the combinatorial structure of the join problem. In particular, the theorem does not ensure that $polymat_W(B_1)$ and $polymat_W(B_2)$ are both substantially lower than $polymat_W(B)$. The next theorem remedies this limitation.

THEOREM 3. (POLYMATROID SPLIT THEOREM) *Consider a clean, acyclic* DegC *of degree constraints and a join $\mathcal{Q}$ conforming to* DegC*. Denote by W an arbitrary fractional XY-pair cover of* DegC*. Let B be a box with $polymat_W(B) \geq 2$. There is a set S of at most $2d+1$ boxes — where $d = |schema(\mathcal{Q})|$ — such that*

(1) the boxes in S are mutually disjoint, and their union is B;

(2) $polymat_W(B') \leq \frac{1}{2} \cdot polymat_W(B)$ holds for every box $B' \in S$;

(3) $\sum_{B' \in S} polymat_W(B') \leq polymat_W(B)$.

When DegC contains only cardinality constraints (i.e., constraints $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}})$ with $\mathcal{X} = \emptyset$), Theorem 3 reduces to the *AGM split theorem* proved by Deng, Lu, and Tao [14].

4.3 Proving the Theorems

This subsection serves as a proof of Theorem 3; the argument also establishes Theorem 2 as a by-product. The material presented here is not required to follow the subsequent sections.

Let us start with an arbitrary non-singleton box B with split dimension σ. Let $\{B_1, B_2, ..., B_s\}$ be a set of $s \geq 2$ boxes refining B. We will prove:

LEMMA 4. *For any fractional XY-pair cover W of* DegC*, it holds that*

$$\sum_{t=1}^{s} polymat_W(B_t) \leq polymat_W(B).$$

PROOF. Let us first review Friedgut's inequality [15]. Fix arbitrary integers p and q at least 1. Take a set of non-negative real values $\{a_{i,j} \mid i \in [p], j \in [q]\}$. In addition, take another set of non-negative real values $\{b_i \mid i \in [q]\}$ satisfying $\sum_{i=1}^{q} b_i \geq 1$. Assuming $0^0 = 0$, Friedgut's inequality states that

$$\sum_{i=1}^{p} \prod_{j=1}^{q} a_{i,j}^{b_j} \leq \prod_{j=1}^{q} \left(\sum_{i=1}^{p} a_{i,j} \right)^{b_j}. \quad (11)$$

Denote by σ the split dimension of B. Define

$$P = \{\pi \in \text{space}(\mathsf{DegC}) \mid A_\sigma \in \mathit{diffatt}(\pi)\}. \quad (12)$$

Recall that each π denotes a pair $(\mathcal{X}, \mathcal{Y})$ and that $\mathit{diffatt}(\pi) = \mathcal{Y} \setminus \mathcal{X}$. Thus, $A_\sigma \in \mathit{diffatt}(\pi)$ tells us that every attribute — say A_i — in $\mathcal{X}$ must have an outedge to A_σ in $\mathcal{G}$ (the constraint dependency graph of DegC). This further implies $i < \sigma$ because the sequence $A_1, A_2, ..., A_d$ is a topological order of $\mathcal{G}$.

For each $t \in [s]$, we have from (10)

$$\begin{aligned} & polymat_W(B_t) \\ = & \prod_{\pi \in \text{space}(\mathsf{DegC})} (N_\pi(B_t))^{W_\pi} \\ = & \prod_{\pi \in \text{space}(\mathsf{DegC}) \setminus P} (N_\pi(B_t))^{W_\pi} \cdot \prod_{\pi \in P} (N_\pi(B_t))^{W_\pi} \\ \leq & \prod_{\pi \in \text{space}(\mathsf{DegC}) \setminus P} (N_\pi(B))^{W_\pi} \cdot \prod_{\pi \in P} (N_\pi(B_t))^{W_\pi} \quad (13) \end{aligned}$$

where the last step used the fact $N_\pi(B_t) \leq N_\pi(B)$ for all $\pi \in \text{space}(\mathsf{DegC})$, which holds true because $B_t \subset B$ (the reader may want to revisit the definition in (9)).

Before proceeding, we need two observations. First, as W is a fractional XY-pair cover of DegC, the definition of P in (12) tells us

$$\sum_{\pi \in P} W_\pi \geq 1. \quad (14)$$

Second, for every $\pi \in P$, we must have

$$\sum_{t=1}^{s} N_\pi(B_t) = N_\pi(B). \quad (15)$$

To explain why, let us assume $\pi = (\mathcal{X}, \mathcal{Y})$. As argued earlier, every attribute in $\mathcal{X}$ precedes A_σ in the topological order $A_1, A_2, ..., A_d$ of $\mathcal{G}$. This, together with σ being the split dimension of B, indicates that $B(A)$ must be a singleton interval on *every* attribute $A \in \mathcal{X}$. As the set $\{B_1, B_2, ..., B_s\}$ refines B, we can assert that $B_t(A)$ is a singleton interval for all $A \in \mathcal{X}$ and $t \in [s]$. Consider now the relation $R_\pi \in \mathcal{Q}$ that guards π. All the tuples in $R_\pi(B)$ must have exactly the same projection on $\mathcal{X}$. Hence, the $(\mathcal{Y}|\mathcal{X})$-degree of $R_\pi(B)$ — which is simply $N_\pi(B)$ — is the number of distinct projections on $\mathcal{Y}$ of the tuples in $R_\pi(B)$. Similarly, for each $t \in [s]$, the $(\mathcal{Y}|\mathcal{X})$-degree of $R_\pi(B_t)$ — which is simply $N_\pi(B_t)$ — is the number of distinct projections on $\mathcal{Y}$ of the tuples in $R_\pi(B_t)$. As the attribute A_σ is in $\mathcal{Y}$, the

set of projections on $\mathcal{Y}$ made by the tuples in $R_\pi(B_t)$ is disjoint from the set made by the tuples in $R_\pi(B_{t'})$, for any $1 \leq t < t' \leq s$. This proves the correctness of (15).

Equipped with these observations, we can derive

$$\begin{aligned}
&\sum_{t=1}^{s} polymat_W(B_t) \\
\leq\ & \prod_{\pi \in \text{space}(\mathsf{DegC}) \setminus P} (N_\pi(B))^{W_\pi} \cdot \sum_{t=1}^{s} \prod_{\pi \in P} (N_\pi(B_t))^{W_\pi} \\
\leq\ & \prod_{\pi \in \text{space}(\mathsf{DegC}) \setminus P} (N_\pi(B))^{W_\pi} \cdot \prod_{\pi \in P} \sum_{t=1}^{s} (N_\pi(B_t))^{W_\pi} \\
=\ & \prod_{\pi \in \text{space}(\mathsf{DegC}) \setminus P} (N_\pi(B))^{W_\pi} \cdot \prod_{\pi \in P} (N_\pi(B))^{W_\pi} \\
=\ & polymat_W(B)
\end{aligned}$$

where the first inequality applied (13), the second applied Friedgut's inequality (12) — the application is made possible by (14) — and the third applied (15). □

Theorem 2 is an immediate corollary of Lemma 4. More effort is needed to prove Theorem 3. The rest of the argument is built on ideas used in [14] to establish the AGM split theorem.

Let us introduce a function $\texttt{replace}(B, i, I)$, where B is a box, i is an integer in $[d]$, and I is an interval of integers. The function yields the box

$$B(A_1) \times ... \times B(A_{i-1}) \times I \times B(A_{i+1}) \times ... \times B(A_d)$$

that is, replacing the projection of B on dimension i with I, while retaining its projections on the other dimensions.

Figure 1 presents an algorithm for splitting a box B into a set S of smaller boxes meeting the requirements of Theorem 3. The procedure, $\texttt{split}(i, B)$, admits two parameters: an integer $i \in [d]$ and a box $B = [x_1, y_1] \times ... \times [x_d, y_d]$. The box must satisfy the condition that its projections on the first $i-1$ attributes should be singleton intervals. Note that the condition is always met when i is set to 1, regardless of B. Indeed, to obtain the aforementioned set S, one should invoke the procedure with $\texttt{split}(1, B)$.

The procedure $\texttt{split}(i, B)$ examines a type of three-way splits along dimension i. Specifically, given an integer $z \in [x_i, y_i]$, we can split B into three boxes:

$$\begin{aligned}
B_{left} &:= \texttt{replace}(B, i, [x_i, z-1]) \\
B_{mid} &:= \texttt{replace}(B, i, [z, z]) \\
B_{right} &:= \texttt{replace}(B, i, [z+1, y_i]).
\end{aligned}$$

Note that B_{left} (resp., B_{right}) does not exist if $z = x_i$ (resp., y_i). The procedure identifies the largest z satisfying $polymat_W(B_{left}) \leq \frac{1}{2} \cdot polymat_W(B)$ (Line 2). Such a z must exist since the condition holds for $z = x_i$; in that case, B_{left} is empty and hence $polymat_W(B_{left}) = 0$. The boxes B_{left} and B_{right}, if they exist, are added to S directly (Line 3 and 7). Let us now turn our attention to B_{mid}. Its projections on the first i dimensions are singleton intervals. If $i = d$, then B_{mid} has shrunk into a singleton box, in which case it is also added to S (Line 5). Otherwise, we recursively invoke $\texttt{split}(i+1, B_{mid})$ to split B_{mid} into a set S' of boxes and then merge S' into S (Line 6).

algorithm $\texttt{split}(i, B)$
/* assume $B = [x_1, y_1] \times ... \times [x_d, y_d]$; it is required that $x_1 = y_1$, $x_2 = y_2$, ..., and $x_{i-1} = y_{i-1}$ */
1. $S \leftarrow \emptyset$
2. $z \leftarrow$ the largest value in $[x_i, y_i]$ s.t. $polymat_W(B_{left}) \leq \frac{1}{2} \cdot polymat_W(B)$ where $B_{left} = \texttt{replace}(B, i, [x_i, z-1])$
3. **if** $z - 1 \geq x_i$ **then** $S \leftarrow S \cup \{B_{left}\}$
4. $B_{mid} \leftarrow \texttt{replace}(B, i, [z, z])$
5. **if** $i = d$ **then** add B_{mid} to S
6. **else** $S \leftarrow S \cup \texttt{split}(i+1, B_{mid})$
7. **if** $z + 1 \leq y_i$ **then** $S \leftarrow S \cup \{B_{right}\}$ where $B_{right} \leftarrow \texttt{replace}(B, i, [z+1, y_i])$
8. **return** S

Figure 1: A split algorithm for Theorem 3

Next, we show that the set S produced by $\texttt{split}(1, B)$ has the three properties in Theorem 3. Property (1) is obvious and omitted.

To prove Property (2), we analyze each type of boxes added to S:

- For every B_{left} created by a recursive call $\texttt{split}(i, B')$ — here, B' is some box inside B generated in the recursion — we have $polymat_W(B_{left}) \leq \frac{1}{2} \cdot polymat_W(B') \leq \frac{1}{2} \cdot polymat_W(B)$. Hence, every B_{left} satisfies Property (2).

- Consider the B_{right} created by $\texttt{split}(i, B')$. By definition of the value z at Line 2, it must hold that $polymat_W(B_{left} \cup B_{mid}) \geq \frac{1}{2} \cdot polymat_W(B')$. By Lemma 4, $polymat_W(B_{left} \cup B_{mid}) + polymat_W(B_{right}) \leq polymat_W(B')$. This implies $polymat_W(B_{right}) \leq \frac{1}{2} \cdot polymat_W(B') \leq \frac{1}{2} \cdot polymat_W(B)$. Hence, every B_{right} satisfies Property (2).

- In all the recursive calls to $\texttt{split}$, the only B_{mid} added to S is the one that has shrunk into a singleton box (Line 5). For that B_{mid}, we

have $polymat_W(B_{mid}) \leq 1$, which is at most $\frac{1}{2} \cdot polymat_W(B)$ because $polymat_W(B) \geq 2$.

Property (3) is a corollary of Lemma 4. At any recursive call $\texttt{split}(i, B')$, the lemma guarantees $polymat_W(B_{left}) + polymat_W(B_{mid}) + polymat_W(B_{right}) \leq polymat_W(B')$. Now, the property follows from a simple inductive argument on i, essentially breaking $polymat_W(B_{mid})$ into the sum of the polymatroid bounds of W on smaller boxes.

The set S returned by $\texttt{split}(1, B)$ has a size no more than $2d + 1$ because we add at most two boxes to S for each $i \in [d-1]$ and at most three for $i = d$. This completes the proof of Theorem 3.

5. THE BOX-TREE TECHNIQUE

This section will explain how to use the split theorems to perform join reporting and sampling. We will strive to use only Theorem 2, which is surprisingly powerful — especially on static data (i.e., no updates) — despite its primitive appearance. In Section 5.4, we include a discussion of when the additional power offered by Theorem 3 becomes useful.

Throughout the section, we fix

- a join $\mathcal{Q}$ and a clean, acyclic set DegC of degree constraints such that $\mathcal{Q} \models$ DegC;
- $A_1, A_2, ..., A_d$ defined in (6) — recall that $d = |schema(\mathcal{Q})|$;
- an optimal fractional XY-pair cover W^* of DegC.

Note that $polymat_{W^*}$ is the polymatroid bound of $\mathcal{Q}$.

We will make two assumptions until Section 5.4.

- **A1:** The input size IN of $\mathcal{Q}$ is a power of 2.
- **A2:** Attribute values lie in the so-called *rank space*. Specifically, for any relation $R \in \mathcal{Q}$, any attribute $A \in schema(R)$, and any tuple $\boldsymbol{u} \in R$, we consider that $\boldsymbol{u}(A)$ is an integer in [IN].

These assumptions allow us to illustrate the core ideas while abstracting away certain implementation details. In fact, they are reasonable when the relations in $\mathcal{Q}$ are static, as explained in Section 5.4.

From now, it is essential to view the join result from a geometric perspective. Every tuple of $join(\mathcal{Q})$ can be seen as a point in the box $[\text{IN}]^d$ — subject to assumption **A2** — whose i-th coordinate ($i \in [d]$) is the tuple's value on A_i. Note, however, that the converse does not hold: not every point in $[\text{IN}]^d$ necessarily corresponds to a tuple of $join(\mathcal{Q})$. We refer to $[\text{IN}]^d$ as the *join space*.

5.1 The Box-Tree

Parameterized by an integer $\Delta \geq 1$, the *box tree* $\mathcal{T}$ is built by splitting the join space recursively. This is a binary tree whose nodes are each associated with a distinct box. Henceforth, we will denote a node using its associated box. For a node B, we define its *polymatroid value* as $polymat_{W^*}(B)$; see (10) for its calculation. The polymatroid value of a node is at least the sum of the polymatroid values of its children.

Specifically, the root of $\mathcal{T}$ is associated with the join space. In general, consider a node associated with box B. If node B has a polymatroid value at most Δ, it becomes a leaf of $\mathcal{T}$. Otherwise, B must be a non-singleton box, which we split *evenly* along its split dimension into two boxes B_1 and B_2 (the reader may want to revisit the definition of the split dimension in Section 4.2). By Theorem 2, $polymat_{W^*}(B_1) + polymat_{W^*}(B_2) \leq polymat_{W^*}(B)$. We then create two child nodes of B, associated with B_1 and B_2, respectively. This completes the definition of the box tree. It is worth mentioning that even splits are always possible due to assumption **A1**.

The tree $\mathcal{T}$ has four properties:

- **P1**: Every internal node has a polymatroid value over Δ, while every leaf node has a polymatroid value at most Δ.
- **P2**: The leaf boxes are mutually disjoint and their union is $[\text{IN}]^d$.
- **P3**: $\mathcal{T}$ has height $O(\log \text{IN})$.
- **P4**: $\mathcal{T}$ has $O((polymat_{W^*}/\Delta) \cdot \log \text{IN})$ nodes.

Properties **P1** and **P2** follow immediately from the definition of $\mathcal{T}$. To understand **P3**, notice that the box of a child node covers half as many points in the join space as that of its parent node. Since the join space has IN^d points (assumption **A2**), it can be split at most $d \log \text{IN} = O(\log \text{IN})$ times before reducing to a singleton box (which must become a leaf because its polymatroid value is at most 1). Hence, the height of $\mathcal{T}$ is $O(\log \text{IN})$.

To prove **P4**, imagine removing all the leaf nodes of $\mathcal{T}$, and let $\mathcal{T}'$ denote the resulting tree (note that $\mathcal{T}'$ need not be binary). Every leaf of $\mathcal{T}'$ is an internal node in $\mathcal{T}$ and hence has a polymatroid value over Δ. The sum of the polymatroid values of all leaves of $\mathcal{T}'$ cannot exceed the polymatroid value of the root, which is precisely $polymat_{W^*}$. Hence, $\mathcal{T}'$ has less than $polymat_{W^*}/\Delta$ leaves, and thus at most $O((polymat_{W^*}/\Delta) \cdot \log \text{IN})$ nodes in total. This proves **P4**, since each node of $\mathcal{T}'$ can parent at most 2 leaves of $\mathcal{T}$.

Henceforth, we assume a *polymatroid oracle* that can calculate $polymat_{W^*}(B)$ for any box B. The implementation of such an oracle will be discussed in Section 5.4. The reader can safely assume that each call takes $O(\log \text{IN})$ time by resorting to a separate data structure of $O(\text{IN})$ space.

5.2 Join Reporting

Property **P2** of the box tree $\mathcal{T}$ implies:

$$\bigcup_{\text{leaf } B} join(\mathcal{Q}(B)) \quad = \quad join(\mathcal{Q}).$$

where $\mathcal{Q}(B)$ is defined in (8). This offers a simple way to compute $join(\mathcal{Q})$. Set $\Delta = 1$ and generate the entire $\mathcal{T}$, which (by property **P4**) requires $O(polymat_{W^*} \log \text{IN})$ CPU times plus $O(polymat_{W^*} \log \text{IN})$ calls to the polymatroid oracle. By property **P1**, each leaf B of $\mathcal{T}$ has a polymatroid value at most 1. This implies that, for each $R \in \mathcal{Q}$, the relation $R(B)$ as defined in (7) has at most a single tuple. Thus, $join(\mathcal{Q}(B))$ can be obtained in $O(1)$ time. This yields a join reporting algorithm with running time $O(polymat_{W^*} \log^2 \text{IN})$, which is worst-case optimal up to an $O(\log^2 \text{IN})$ factor.

Let us now attend to the small-delay enumeration problem (see Section 2), where the goal is to minimize the space of the data structure. To build the structure, first create a box tree $\mathcal{T}$ by setting the parameter Δ to the target delay. Collect the set $\mathcal{L}$ of leaves B whose $\mathcal{Q}(B)$ has a non-empty result. The data structure simply stores $\mathcal{L}$. The space consumption is $O((\text{polylog}_{W^*}/\Delta) \log \text{IN})$ by property **P4**. To enumerate $join(\mathcal{Q})$, for each box $B \in \mathcal{L}$ we compute $join(\mathcal{Q}(B))$ using a worst-case optimal join algorithm [21], which runs in $O(polymat_{W^*}(B)) = O(\Delta)$ time (property **P1**). This fulfills the delay requirement as $join(\mathcal{Q}(B))$ cannot be empty. In comparison, the state of the art [13, Proposition 3] demands $O((\text{AGM}/\Delta) \log \text{IN})$ space, where AGM is the AGM bound of $\mathcal{Q}$. This is never better than $O((\text{polylog}_{W^*}/\Delta) \log \text{IN})$ and can, in fact, be considerably worse.

5.3 Join Sampling

We now tackle the join sampling problem defined in Section 2. Let $\mathcal{T}$ be the box tree defined by $\Delta = 1$. However, rather than materializing the entire $\mathcal{T}$ (as in Section 5.2), this time we store nothing of $\mathcal{T}$, but sample its root-to-leaf paths on an "as-needed" basis.

The `sample` algorithm, shown in Figure 2, begins at the root of $\mathcal{T}$. In general, if the algorithm is currently at an internal node (with box) B, it first obtains the child nodes B_1 and B_2 of B in $\mathcal{T}$ (this requires only splitting B evenly on its split dimension; see Section 5.1). Then,

algorithm `sample`
/* $\mathcal{T}$ is the box tree defined by $\Delta = 1$ */
1. $B \leftarrow [\text{IN}]^d$ /* the join space */
2. **while** $polymat_W(B) > 1$ **do**
3. obtain the child nodes B_1, B_2 of B in $\mathcal{T}$
4. $B^+ \leftarrow$ a rand. value drawn according to (16)-(18)
5. **if** $B^+ = \perp$ **then return** "failure"
6. $B \leftarrow B^+$

/* $polymat_W(B) \leq 1$ here */
7. **if** $join(\mathcal{Q}(B)) \neq \emptyset$ **then**
8. **return** the only tuple in $join(\mathcal{Q}(B))$
9. **else return** "failure"

Figure 2: An algorithm for join sampling

it draws a random value $B^+ \in \{B_1, B_2, \perp\}$ from the following distribution:

$$\mathbf{Pr}[B^+ = B_1] \quad = \quad \frac{polymat_{W^*}(B_1)}{polymat_{W^*}(B)} \tag{16}$$

$$\mathbf{Pr}[B^+ = B_2] \quad = \quad \frac{polymat_{W^*}(B_2)}{polymat_{W^*}(B)} \tag{17}$$

$$\mathbf{Pr}[B^+ = \perp] \quad = \quad 1 - \mathbf{Pr}[B^+ = B_1] - \mathbf{Pr}[B^+ = B_2]. \tag{18}$$

If $B^+ = \perp$, then the algorithm declares "failure" and terminates. Otherwise, it descends to the child B^+.

Now consider that the algorithm has reached a leaf B, whose polymatroid value is at most 1 (property **P1**). As explained in Section 5.2, $join(\mathcal{Q}(B))$ has at most one tuple and can be computed in $O(1)$ time. If $join(\mathcal{Q}(B))$ is not empty, the algorithm returns the only tuple therein; otherwise, it declares "failure".

By Property **P3**, `sample` runs in $O(\log \text{IN})$ time plus $O(\log \text{IN})$ calls to the polymatroid oracle. Whenever `sample` does not declare failure, we say that it *succeeds*. Next, we argue that if the algorithm succeeds, it returns a uniformly random tuple from $join(\mathcal{Q})$.

Fix an arbitrary tuple $\boldsymbol{u} \in join(\mathcal{Q})$. As mentioned at the beginning of Section 5, $\boldsymbol{u}$ can be regarded as a point in the join space. Let B be the leaf of $\mathcal{T}$ whose box covers that point — such a leaf must exist due to Property **P2**. It must hold that

$$polymat_{W^*}(B) = 1. \tag{19}$$

To see why, note that $polymat_{W^*}(B) \leq 1$ by property **P1**, while, on the other hand, $polymat_{W^*}(B) \geq |join(\mathcal{Q}(B))| \geq 1$. Let $B_0, B_1, ..., B_h$ be the nodes on the path of $\mathcal{T}$ from the root to B (i.e., B_0 is the root and

$B_h = B$). The probability that `sample` outputs $\boldsymbol{u}$ is

$$\prod_{i=1}^{h} \frac{polymat_{W^*}(B_i)}{polymat_{W^*}(B_{i-1})} = \frac{polymat_{W^*}(B)}{polymat_{W^*}([\mathrm{IN}]^d)}$$
$$\text{(by (19))} = \frac{1}{polymat_{W^*}}. \quad (20)$$

As the above is identical for all $\boldsymbol{u}$, we conclude that `sample` returns a uniformly random tuple of $join(\mathcal{Q})$ whenever it succeeds.

The success probability of `sample` is

$$\sum_{\boldsymbol{u} \in join(\mathcal{Q})} \mathbf{Pr}[\boldsymbol{u} \text{ is output}] = \frac{\mathrm{OUT}}{polymat_{W^*}}.$$

In expectation, $\frac{polymat_{W^*}}{\mathrm{OUT}}$ repeats of the algorithm are needed to acquire a sample.

A special case occurs when $\mathrm{OUT} = 0$ — this will force us to repeat the algorithm forever. To remedy the issue, we stop using `sample` after $polymat_{W^*}$ repeats and revert to a worst-case optimal join algorithm (e.g., [21]) to compute $join(\mathcal{Q})$ in $O(polymat_{W^*})$ time, which will either confirm $\mathrm{OUT} = 0$ or allow us to obtain a random tuple of $join(\mathcal{Q})$ with no extra cost.

We have not mentioned any data structure. Indeed, the `sample` algorithm itself requires no data structures. Nevertheless, to implement a call to the polymatroid oracle in $O(\log \mathrm{IN})$ time, one still needs a data structure of $O(\mathrm{IN})$ space, as will be discussed in Section 5.4. This makes the expected join sampling time $O(\frac{polymat_{W^*}}{\max\{1,\mathrm{OUT}\}} \log^2 \mathrm{IN})$, which is worse than the state of the art [29] by only a factor of $O(\log^2 \mathrm{IN})$.

5.4 Discussion

We have made two assumptions, **A1** and **A2**, at the beginning of Section 5. These assumptions are quite "harmless" when the relations of $\mathcal{Q}$ are static. First, if IN is not a power of 2, one can add dummy tuples to increase IN to the nearest power of 2. Second, **A2** can be satisfied through *value renaming*: map the tuple values to distinct elements in $[\mathrm{IN}]$, which can be achieved with sorting in $O(\mathrm{IN} \log \mathrm{IN})$ time.

If value renaming is difficult or inconvenient (e.g., when insertions are allowed), however, we would have to work directly with the original tuple values. Suppose, in general, that every value is in a domain $[U]$ for some integer U, which can be arbitrarily larger than IN. The box tree $\mathcal{T}$ described in Section 5.1 still applies, but its height becomes $O(\log U)$; accordingly, the number of nodes in T becomes $O((polymat_{W^*}/\Delta) \log U)$.

This is a place where Theorem 3 is useful. Specifically, it allows us build an alternative box tree $\mathcal{T}$ satisfying properties **P1**-**P4** without assumptions **A1** and **A2**. Parameterized by an integer $\Delta \geq 2$, this $\mathcal{T}$ is a $(2d+1)$-ary tree whose nodes are each associated with a distinct box. As before, the polymatroid value of a node is at least the sum of the polymatroid values of its children. The root of $\mathcal{T}$ is associated with $\mathbb{N}^d$. In general, consider a node associated with box B. If B has a polymatroid value at most Δ, it becomes a leaf of $\mathcal{T}$. Otherwise, we split B into a set S of boxes promised by Theorem 3 and, for each $B' \in S$, create a child of B associated with B'. This completes the definition of the new box tree.

Properties **P1** and **P2** follow immediately from the definition. Regarding property **P3**, the argument in Section 5.1 no longer works. Nevertheless, we can "rescue" the property as follows. First, the polymatroid value of the root is $O(\mathrm{IN}^d)$ (this is a trivial upper bound for the polymatroid bound of $\mathcal{Q}$). Every time we descend a level, the polymatroid value drops by at least half, as guaranteed by Theorem 3. Therefore, the number of levels cannot exceed $O(\log \mathrm{IN})$. Finally, property **P4** can be established using an argument similar to the one in Section 5.1.

All the algorithms in Sections 5.2 and 5.3 can be adapted to work with the new box tree. Splitting a box using Theorem 3 would require $O(\log \mathrm{IN})$ calls [14] to the polymatroid oracle (rather than $O(1)$ calls in Section 5.1). This may cause a slow down in CPU time by a factor of $O(\log \mathrm{IN})$.

Finally, we briefly discuss polymatroid oracles. The essence of implementing such an oracle is to support a *range count* operation on each relation $R \in \mathcal{Q}$. Assume, w.l.o.g., that $schema(R) = \{A_{i_1}, A_{i_2}, ..., A_{i_t}\}$, where $t = |schema(R)|$ and $i_1 < i_2 < ... < i_t$. Define a lexicographic order over R by regarding each tuple $\boldsymbol{u} \in R$ as a string of t "characters": $\boldsymbol{u}(A_{i_1})\boldsymbol{u}(A_{i_2})...\boldsymbol{u}(A_{i_t})$. Given an interval I over the lexicographic order, a range count operation finds the number of tuples whose encodings fall within I. It is elementary to design a structure of $O(\mathrm{IN})$ space that performs the operation in $O(\log \mathrm{IN})$ time and supports an update in $O(\log \mathrm{IN})$ time.

6. A LITERATURE SURVEY

Modern research on natural joins was revitalized by the discovery of the AGM bound — named after Atserias, Grohe, and Marx [4] — which is an upper bound on the join result size that is asymptotically tight in the worst case under data complexity. Ngo et al. [23] designed the first algorithm that computes a join $\mathcal{Q}$ in $O(AGM)$ time, where AGM is the AGM bound of $\mathcal{Q}$. This is

worst-case optimal because even just reporting the result $join(\mathcal{Q})$ may necessitate $\Omega(AGM)$ time, as is implied by the tightness of the AGM bound. Since then, the community has developed quite a large number of join algorithms [3,7,17,20,22,24–26,28] achieving the worst-case optimal time, up to polylogarithmic factors.

The AGM bound, however, tends to be loose in practice. One reason is that it accounts only for cardinality constraints (as are defined in Section 2), while ignoring degree constraints $(\mathcal{X}, \mathcal{Y}, N_{\mathcal{Y}|\mathcal{X}})$ with $\mathcal{X} \neq \emptyset$. In [18], Khamis, Ngo, and Suciu introduced the polymatroid bound, which generalizes the AGM bound by incorporating all degree constraints. In Section 3, we described polymatroid bounds for *acyclic* sets of degree constraints, as they are the focus of this article. However, it should be noted that polymatroid bounds also apply to *cyclic* sets of degree constraints. Khamis, Ngo, and Suciu [18] presented an algorithm named PANDA that can process any join $\mathcal{Q}$ in $O(polymat \operatorname{polylog} \mathrm{IN})$ time, where $polymat$ is the polymatroid bound of $\mathcal{Q}$. In [21], Ngo showed that the time can be improved to $O(polymat)$ when the set of degree constraints is acyclic. The running time is worst-case optimal in the sense explained in Section 3.

Deep and Koutris [13] studied the small-delay enumeration problem in a setup more general than the one defined in Section 2: their formulation allows a selection condition to be provided at runtime. Of particular relevance here is the following variant of their setting: a user supplies a box B and demands $join(\mathcal{Q}(B))$ — as defined in (8) — to be enumerated with a delay of $O(\Delta)$. The goal is to process $\mathcal{Q}$ into a data structure to support this operation for any B. The ideas presented in Section 5.2 can be integrated with the standard *dyadic interval* technique to achieve the purpose, although the space consumption would increase by an $O(\operatorname{polylog} \mathrm{IN})$ factor. Readers interested in this line of work may refer to [31] for alternative space-delay tradeoffs.

For the problem of join sampling, improving over previous results [2,11,12], Deng, Lu, and Tao [14] presented a structure of $O(\mathrm{IN} \operatorname{polylog} \mathrm{IN})$ space that achieves expected sample time $O(\frac{\mathrm{AGM}}{\max\{1,\mathrm{OUT}\}} \operatorname{polylog} \mathrm{IN})$, where $\mathrm{OUT} = |join(\mathcal{Q})|$. They introduced the AGM split theorem, which as mentioned earlier is the "AGM-counterpart" of Theorem 3, and the box tree described in Section 5.4. Kim et al. [19] improved the result of [14] by shaving off the $\operatorname{polylog} \mathrm{IN}$ factor in both the space usage and sample time. Later, Wang and Tao [29] managed to reduce the expected sample time to $O(polymat/\max\{1, \mathrm{OUT}\})$, while keeping the space at $O(\mathrm{IN})$, for joins with acyclic sets of degree constraints. The methods of [19, 29] are rather sophisticated and do not fall into the box-tree framework. Recently, targeting the same type of joins as [29], Capelli, Irwin, and Salvati [7] gave a structure of $O(\mathrm{IN} \log \mathrm{IN})$ space and $O(\frac{polymat}{\max\{1,\mathrm{OUT}\}} \log \mathrm{IN})$ expected sample time. Their method can be understood as a way to implement what was described in Section 5.3.[2]

Deng, Lu, and Tao [14] observed a connection between join sampling and small-delay enumeration. Specifically, they showed that the latter problem can be reduced, with high probability (i.e., at least $1 - 1/\mathrm{IN}^c$ for any arbitrarily large constant), to the former. By applying their reduction to the result of [29], one obtains with high probability a structure of $O(\mathrm{IN})$ space that can enumerate $join(\mathcal{Q})$ with a delay of $O(\frac{polymat}{\max\{1,\mathrm{OUT}\}} \operatorname{polylog} \mathrm{IN})$. Even better, the enumeration actually produces a random permutation of the elements in $join(\mathcal{Q})$.

The three problems defined in Section 2 admit better solutions when the join $\mathcal{Q}$ is "acyclic" as per the definition in [1, Chapter 6.4] (this should not be confused with the notion of acyclicity on degree constraint sets). Specifically, Yannakakis' algorithm [30] settles the join reporting problem in $O(\mathrm{IN} + \mathrm{OUT})$ time. Moreover, the algorithm implies a structure of $O(\mathrm{IN})$ space that can enumerate $join(\mathcal{Q})$ with a constant-time delay. For join sampling, one can preprocess an acyclic $\mathcal{Q}$ into a structure of $O(\mathrm{IN})$ space that guarantees $O(1)$ sample time [32].

In general, given a non-acyclic join $\mathcal{Q}$, it is possible to convert it to an acyclic join $\mathcal{Q}'$ with exactly the same result, i.e., $join(\mathcal{Q}) = join(\mathcal{Q}')$. The conversion takes $O(\mathrm{IN}^{\mathrm{fhtw}(\mathcal{Q})})$ time, where $\mathrm{fhtw}(Q)$ is the *fractional hyper-tree width* of $\mathcal{Q}$. We refer the reader to [16] for the precise calculation of $\mathrm{fhtw}(Q)$ and remark that $\mathrm{fhtw}(Q) > 1$ for every non-acyclic $\mathcal{Q}$. The new join $\mathcal{Q}'$ has an input size of $O(\mathrm{IN}^{\mathrm{fhtw}(\mathcal{Q})})$. By combining this idea with our earlier discussion on acyclic joins, one obtains a structure of $O(\mathrm{IN}^{\mathrm{fhtw}(\mathcal{Q})})$ space that can be used to enumerate $join(\mathcal{Q})$ with a constant-time delay or to sample from it in constant time.

Both the "small-delay enumeration" and "join sampling" problems are closely related to the *direct access* problem. Let $\prec$ be a total order imposed on a join result $join(\mathcal{Q})$. Given an integer $i \in [\mathrm{OUT}]$ (where $\mathrm{OUT} = |join(\mathcal{Q})|$), a direct-access query returns the i-th tuple of $join(\mathcal{Q})$ under the ordering of $\prec$. The goal of the direct access problem is to preprocess $\mathcal{Q}$ into a

[2]The method of [7] treats a relation of t attributes as an "expanded" relation of $t \log \mathrm{IN}$ attributes, each taking values 0 or 1. We find that treatment unnecessary when adopting the box-tree perspective.

data structure that can answer such a query in Δ time, where Δ is a target bound. Given a direct-access structure and the value of OUT, we can enumerate $join(\mathcal{Q})$ with a delay of Δ by issuing direct-access queries with $i = 1, 2, ..., \text{OUT}$, respectively, or sample from $join(\mathcal{Q})$ in Δ time by issuing a query with a random $i \in [\text{OUT}]$. Readers interested in this topic may refer to representative works [5, 6, 8, 9] and the references therein. For small-delay enumeration and join sampling, however, the performance guarantees provided by the existing direct-access methods are subsumed by the results stated earlier.

7. CONCLUSIONS

The box-tree technique is a method for join reporting and sampling developed by the database theory community in recent years. It is intuitive, conceptually clean, and can be deployed easily to design algorithms for join processing. Its theoretical foundation lies in the AGM split theorem introduced by [14]. In this article, we proved two new split theorems that extend the result of [14] by replacing the AGM bound with a polymatroid bound. These new theorems further strengthen the box-tree technique, enabling it to yield algorithms and data structures that rival the current state of the art.

One limitation of the technique is that the box tree $\mathcal{T}$ has $O(\log \text{IN})$ levels — the logarithmic factor prevents the technique from fully matching the best time bounds for join reporting and sampling. This limitation could be eliminated at the risk of losing the technique's elegance: given a box B, we could split it, along its split dimension (see its meaning in Section 4.1), into a maximum number of boxes B' whose $\mathcal{Q}(B')$ — as defined in (8) — has a non-empty size. Such an approach allows us to "reinterpret" the reporting algorithm of [21] and the sampling algorithm of [29] from the box-tree perspective. Nevertheless, we do not yet have an interpretation clean enough for graduate-level classrooms at the moment.

Another, more severe limitation of the technique is that it cannot deal with cyclic sets of degree constraints. The root cause lies in our split theorems, which have been proved only for acyclic sets of degree constraints. We invite the community to address this open problem.

8. REFERENCES

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] Swarup Acharya, Phillip B. Gibbons, Viswanath Poosala, and Sridhar Ramaswamy. Join synopses for approximate query answering. In *Proceedings of ACM Management of Data (SIGMOD)*, pages 275–286, 1999.

[3] Kaleb Alway, Eric Blais, and Semih Salihoglu. Box covers and domain orderings for beyond worst-case join processing. In *Proceedings of International Conference on Database Theory (ICDT)*, pages 3:1–3:23, 2021.

[4] Albert Atserias, Martin Grohe, and Daniel Marx. Size bounds and query plans for relational joins. *SIAM Journal on Computing*, 42(4):1737–1767, 2013.

[5] Guillaume Bagan, Arnaud Durand, Etienne Grandjean, and Frederic Olive. Computing the jth solution of a first-order query. *RAIRO Theor. Informatics Appl.*, 42(1):147–164, 2008.

[6] Karl Bringmann, Nofar Carmeli, and Stefan Mengel. Tight fine-grained bounds for direct access on join queries. *ACM Transactions on Database Systems (TODS)*, 50(1):1:1–1:44, 2025.

[7] Florent Capelli, Oliver Irwin, and Sylvain Salvati. A simple algorithm for worst case optimal join and sampling. In *Proceedings of International Conference on Database Theory (ICDT)*, pages 23:1–23:19, 2025.

[8] Nofar Carmeli, Nikolaos Tziavelis, Wolfgang Gatterbauer, Benny Kimelfeld, and Mirek Riedewald. Tractable orders for direct access to ranked answers of conjunctive queries. *ACM Transactions on Database Systems (TODS)*, 48(1):1:1–1:45, 2023.

[9] Nofar Carmeli, Shai Zeevi, Christoph Berkholz, Alessio Conte, Benny Kimelfeld, and Nicole Schweikardt. Answering (unions of) conjunctive queries using random access and random-order enumeration. *ACM Transactions on Database Systems (TODS)*, 47(3):9:1–9:49, 2022.

[10] Nofar Carmeli, Shai Zeevi, Christoph Berkholz, Benny Kimelfeld, and Nicole Schweikardt. Answering (unions of) conjunctive queries using random access and random-order enumeration. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 393–409, 2020.

[11] Surajit Chaudhuri, Rajeev Motwani, and Vivek R. Narasayya. On random sampling over joins. In *Proceedings of ACM Management of Data (SIGMOD)*, pages 263–274, 1999.

[12] Yu Chen and Ke Yi. Random sampling and size estimation over cyclic joins. In *Proceedings of International Conference on Database Theory (ICDT)*, pages 7:1–7:18, 2020.

[13] Shaleen Deep and Paraschos Koutris. Compressed representations of conjunctive query results. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 307–322, 2018.

[14] Shiyuan Deng, Shangqi Lu, and Yufei Tao. On join sampling and the hardness of combinatorial output-sensitive join algorithms. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 99–111, 2023.

[15] Ehud Friedgut. Hypergraphs, entropy, and inequalities. *Am. Math. Mon.*, 111(9):749–760, 2004.

[16] Martin Grohe and Daniel Marx. Constraint solving via fractional edge covers. *ACM Transactions on Algorithms*, 11(1):4:1–4:20, 2014.

[17] Mahmoud Abo Khamis, Hung Q. Ngo, Christopher Re, and Atri Rudra. Joins via geometric resolutions: Worst case and beyond. *ACM Transactions on Database Systems (TODS)*, 41(4):22:1–22:45, 2016.

[18] Mahmoud Abo Khamis, Hung Q. Ngo, and Dan Suciu. What do shannon-type inequalities, submodular width, and disjunctive datalog have to do with one another? In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 429–444, 2017.

[19] Kyoungmin Kim, Jaehyun Ha, George Fletcher, and Wook-Shin Han. Guaranteeing the $\tilde{O}$(AGM/OUT) runtime for uniform sampling and size estimation over joins. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 113–125, 2023.

[20] Gonzalo Navarro, Juan L. Reutter, and Javiel Rojas-Ledesma. Optimal joins using compact data structures. In *Proceedings of International Conference on Database Theory (ICDT)*, volume 155, pages 21:1–21:21, 2020.

[21] Hung Q. Ngo. Worst-case optimal join algorithms: Techniques, results, and open problems. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 111–124, 2018.

[22] Hung Q. Ngo, Dung T. Nguyen, Christopher Re, and Atri Rudra. Beyond worst-case analysis for joins with minesweeper. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 234–245, 2014.

[23] Hung Q. Ngo, Ely Porat, Christopher Re, and Atri Rudra. Worst-case optimal join algorithms: [extended abstract]. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 37–48, 2012.

[24] Hung Q. Ngo, Ely Porat, Christopher Re, and Atri Rudra. Worst-case optimal join algorithms. *Journal of the ACM (JACM)*, 65(3):16:1–16:40, 2018.

[25] Hung Q. Ngo, Christopher Re, and Atri Rudra. Skew strikes back: new developments in the theory of join algorithms. *SIGMOD Rec.*, 42(4):5–16, 2013.

[26] Dan Olteanu and Jakub Zavodny. Size bounds for factorised representations of query results. *ACM Transactions on Database Systems (TODS)*, 40(1):2:1–2:44, 2015.

[27] Dan Suciu. Applications of information inequalities to database theory problems. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–30, 2023.

[28] Todd L. Veldhuizen. Triejoin: A simple, worst-case optimal join algorithm. In *Proceedings of International Conference on Database Theory (ICDT)*, pages 96–106, 2014.

[29] Ru Wang and Yufei Tao. Join and subgraph sampling under degree constraints. *Journal of Computer and System Sciences (JCSS)*, 155, 2026.

[30] Mihalis Yannakakis. Algorithms for acyclic database schemes. In *Very Large Data Bases, 7th International Conference, September 9-11, 1981, Cannes, France, Proceedings*, pages 82–94, 1981.

[31] Hangdong Zhao, Shaleen Deep, and Paraschos Koutris. Space-time tradeoffs for conjunctive queries with access patterns. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*, pages 59–68, 2023.

[32] Zhuoyue Zhao, Robert Christensen, Feifei Li, Xiao Hu, and Ke Yi. Random sampling over joins revisited. In *Proceedings of ACM Management of Data (SIGMOD)*, pages 1525–1539, 2018.

Reminiscences on Influential Papers

This issue's contributors cover works that are results of paradigm shifts where data-intensive systems expanded beyond relational model, disk-based performance optimizations, one-dimensional data, and "small" scale. Enjoy reading!

While I will keep inviting members of the data management community, and neighboring communities, to contribute to this column, I also welcome unsolicited contributions. Please contact me if you are interested.

Pınar Tözün, *editor*
IT University of Copenhagen, Denmark
pito@itu.dk

Genoveva Vargas-Solar
CNRS, University of Lyon, INSA Lyon, UCBL, LIRIS, UMR5205, Lyon, France
genoveva.vargas-solar@cnrs.fr

Stella Gatziu, Andreas Geppert, and Klaus R. Dittrich.
The SAMOS active DBMS prototype.
In Proceedings of the International Conference on Management of Data (SIGMOD), 1995.

It was the time when the database community was expanding beyond the relational model-embracing object-oriented paradigms, rethinking consistency guarantees, and redesigning DBMS architectures. I began my PhD in the midst of this momentum, focusing on the challenges posed by active rules (Event–Condition–Action - ECA). I joined a team working on early active object-oriented databases via NAOS [1] which was the active rules component of the object oriented database system O_2. Equipped with seminal papers on the first wave of approaches from the 1990s [6], I set out to compare, grasp, and replicate these proposals to prepare for meaningful contributions. My project focused on specifying and detecting simple and composite events arising not only from database operations but also from sources external to the DBMS. Of the surveyed systems, SAMOS (Swiss Active-Mechanism based Object-oriented database System) [3] was particularly influential for its rigorous event model and its efficient detection algorithm for complex event expressions. SAMOS was an active, object-oriented DBMS prototype from the University of Zurich. It added ECA rules on top of a passive OODB (ObjectStore), using a layered architecture so the base DBMS remained unmodified. Rules were compiled, stored as objects, and executed by runtime components (rule manager, composite-event detector, rule execution, SAMOS transactions). SAMOS allowed to specify **primitive** and **composite event** types. Primitive event types included time-based, method, invocations, value changes, transactional, and application-defined event types. Composite event types, were built using an *event algebra* combining *sequence* ($e_1; e_2$), *conjunction* (e_1, e_2), *disjunction* ($e_1 \mid e_2$), *negation* ($\neg e$ within an interval), and *occurrence-reduction* operators (e.g., times(e, n), first occurrence). **Parameters and temporal scopes** (e.g., object identifier, transaction identifier, time windows) constrained which occurrences participated in a composite and when to notify it. **Execution** respected **coupling modes** with respect to the execution scope of a transaction (*immediate*, *deferred* (within or at the end of the triggering transaction), or *decoupled* (separate transaction), while **priorities** order competing rules on the same event. Conditions and actions were expressed in the host OODB's DML (Data Manipulation Language) (e.g., C++/ObjectStore).

In SAMOS, primitive events were notified at runtime via `raise_event` and matched, using indexed descriptors, against the catalog of rules that may respond. Detection of composite events proceeded incrementally: each primitive event advanced a *Colored Petri net* that encoded the event pattern, and a composite was declared when the net reached its terminal marking. Rule execution adopted coupling modes through a dedicated `samTransaction` mechanism, which controlled immediate, deferred, or decoupled firing, handled nested triggers, and ensures recovery. Finally, SAMOS persisted an event history so that complex patterns could span multiple operations and, when configured, even multiple transactions or sessions. The SAMOS event-detection line of work anticipated several problems that later became central in stream processing and continuous querying. In particular, it tackled composite event detection and the management of validity scopes (i.e., detection and notification windows) both essential for observing and reacting to the evolving state of complex systems, whether within a DBMS and its applications, across distributed infrastructures, or in physical domains such as

urban systems, traffic networks, and stock markets.

The work around SAMOS, and active databases more broadly, shaped my research playbook: strategies, methodologies, and results that taught me how to balance revisiting core disciplinary problems with building real systems atop commercial platforms. Equally important, it trained me to move fluidly between theory and implementation. I then had the chance to join the University of Zurich's database group for my postdoc, a hands-on immersion in the very ideas I had studied through their papers and books (and other seminal works), now tested and extended in practice.

In an era where high-velocity data underpins many analytics challenges, searching for and discovering event patterns is central. Consolidating the knowledge built over previous decades provides a solid foundation to address today's expectations and technological shifts (storage architectures, distribution, compute capacity, virtualization, and scale) to push event-centric analytics forward with rigour and out of the box perspectives.

Thomas Heinis
Imperial College London, UK
t.heinis@imperial.ac.uk

Antonin Guttman.
R-trees: A Dynamic Index Structure for Spatial Searching.
In Proceedings of the International Conference on Management of Data (SIGMOD), 1984.

Although some may regard it as a tried and tired topic, I do find spatial indexing genuinely fascinating! It extends the indexing problem into multiple dimensions and introduces a set of challenges that are both distinctive and intellectually compelling. In particular, the total order that underpins most one-dimensional index structures does not carry over to higher dimensions. This paper [4], however, finds an elegant solution that would shape my career and the field.

I first encountered the paper at the start of my postdoc, when I began working on indexing scientific data, and it proved pivotal in setting me on a path toward developing many more spatial indexes. It has influenced not only how I think about indexing, but also how I approach academic writing. At the time, publishing in this area felt especially challenging because the field already seemed crowded - there was no shortage of prior work to contend with. In hindsight, I realise this is probably true of many research topics, but it certainly felt differently then.

As incredible as its impact on my career felt, the importance of the paper for the community, however, is far more important. It's certainly not the first paper on spatial indexing but surely a seminal one in the development and teaching of spatial indexes.

One of the striking aspects of the paper is its clarity and restraint. The premise is simple: many applications store objects that live in multi-dimensional space - rectangles, regions, points, trajectories - and we need an index that supports efficient spatial queries such as "find all objects that intersect this window." Instead of trying to shoehorn these objects into a one-dimensional order with elaborate space-filling curves or ad hoc encodings, Guttman starts from geometry itself: bound each object by a minimal bounding rectangle (MBR) and build a balanced tree where each internal node stores the MBR of its children. The R-tree is, in some sense, the obvious structure once you see it. But before this paper, "obvious" was not how the database community thought about spatial indexing.

The technical core of the paper is a tour through the practical design space of such an index. It does not hide behind a purely theoretical formulation. Instead, it works through the gritty details that any real system builder must face: how to split overflowing nodes, how to choose where to insert a new rectangle so as to control overlap and dead space, and how to propagate structural changes while preserving balance and reasonable page utilization. The split heuristics are not presented as abstract optimization problems, but as concrete trade-offs between CPU time and index quality. As a budding researcher, this was one of the first times I saw a database paper so explicitly embrace "engineering knobs" as first-class research contributions. One of the aspects that would also shape my personal approach to research and its presentation.

Equally important is what the paper does not do. It does not claim that the proposed heuristics are optimal. It does not solve every variant of spatial query or every workload pattern. Instead, it defines a clean abstraction and shows that it is versatile enough to support a variety of queries (point, range, nearest neighbour-like patterns) with competitive performance. This balance between conceptual simplicity and practical completeness is a hallmark of enduring systems work, and R-trees are an exemplary case. There is no question that this paper is and was hugely influential in the community. R-trees quickly became the de facto access method for spatial data in both research prototypes and industrial systems. They influenced spatial extensions in relational databases, underpinned early GIS engines, and inspired a whole family of variants. Many generic index frameworks, explicitly draw on the idea that you can parameterize a balanced tree by a bounding predicate and a set of split/merge policies - an idea that Guttman's design makes tangible. Even today, when we talk about "spatial indexes" in production systems, more often than not we mean "some descendant of an R-tree."

And there are many "descendants" of the R-Tree: R-tree, R+-tree, R*-tree, Hilbert R-tree, Packed R-tree, STR packed R-tree, R-link tree, R1-tree, R-file, 2+3 R-tree, 3DR-tree, MV3R-tree, MVR-tree, HR-tree, RT-tree, PR-tree, TPR-tree, TPR*-tree, BPR-tree, VCI R-tree, ATPR-tree, STR-tree, STRIPES, STRIPES-tree, Bx-tree, Bdual-tree, PR bucket quadtree, IR-tree, uR-tree, GiST, R-tree-over-GiST, SP-GiST, MGiST, MSP-GiST, RD-tree, X-tree, TV-tree, SR-tree, SS-tree, A-tree, P+-tree, Pyramid-technique, Pyramid-tree, VA-file, VA+-file, iDistance, iMinMax(θ), iMinMax(θ), LSB-

tree, LSB forest, D-index, Local Dimensionality Reduction (LDR) index, Global Index (GI), Cluster index, Hybrid tree, M-tree, Slim-tree, PM-tree, Pivoting M-tree, MVP-tree, mvp-tree, VP-tree, GNAT, EGNAT, SAT, M+-tree, DBM-tree, DBM-tree, LAESA, OMNI, B-tree, B+-tree, B*-tree, B-link tree, ISAM, 2-D ISAM, linear hashing, extendible hashing, multidimensional extendible hashing, multidimensional order-preserving linear hashing (MOLHPE), k-d tree, adaptive k-d tree, bintree, BSP-tree, BD-tree, GBD-tree, extended k-d tree, spatial k-d tree (skd-tree), k-d-B-tree, hB-tree, hBP-tree, BV-tree, Cell tree, JP-tree, SP-tree, P-tree, PLOP-hashing, grid file, multilevel grid file, two-level grid file, twin grid file, generalized grid file, multilayer grid file, BANG file, buddy tree, z-ordering, Z-order index, Morton order, Hilbert order, Hilbert curve index, UB-tree, quadtree, PR quadtree, PM quadtree, PMR quadtree, trie, Linear R-tree, Quadratic R-tree, Greene's R-tree, Priority R-tree, Packed Hilbert R-tree, Dynamic Hilbert R-tree, Nearest-X packed R-tree, OMT packed R-tree, TGS R-tree, ND R-tree, Cache-oblivious R-tree, Buffered R-tree, Rsb-tree, RR-tree, RUM-tree, LGUR-tree, FUR-tree, RW-tree, AI+R-tree, 3D R-tree, HR+-tree, MR-tree, STAR-tree, REXP-tree, R1,2D3-tree, R1+2D3-tree, Parametric R-tree, NSI R-tree, STP-tree, GG TPR-tree, HTPR*-tree, RPPF-tree, PCFI+-index, TB-tree, CSE-tree, ANR-tree, FNR-tree, S-TPR-tree, RST-tree, IR2-tree, IRLi-tree, DIR-tree, CIR-tree, CDIR-tree, WIR-tree, WIBR-tree, KR*-tree, SKI, R*-IF, S2I, aR-tree, -tree, S-tree, Y-tree, Z-tree, cR-tree, CR-tree, CR*-tree, Hilbert CR-tree, H-R-tree, MON-tree, UTR-tree, LUR-tree, RUM+-tree, UST-tree, CT-R-tree, Sketch RR-tree, Bulk-load TPR-tree, Horizon TPR-tree, HTPR-tree, LGU TPR-tree, VRTPR-tree, CRTPR-tree, MVTPR-tree, TPRU-tree, VTPR-tree, RP-tree, RTR-tree, TP2R-tree, KR-tree, OST-tree, 3D hybrid R-tree (hybrid 3D R-tree), DR-tree, PA-tree, GS-tree, PK-tree, PH-tree, By-tree, Bs-tree, VR-tree ...

I could go on. Probably not really, but ChatGPT surely could!

The paper itself dates back to 1984 and survives in a charmingly antiquated scanned form. As far as I can tell, it was Guttman's final publication. I tried to contact him, but his LinkedIn profile indicates that he is now retired - and, I feel, very much deservedly so. I find it remarkable that someone could write a seminal, single-author paper - without even his supervisor, Mike Stonebraker - that has attracted more than 12,000 citations, and then choose not to pursue an academic career but instead move into industry - it has the energy of a mic drop moment, but in the context of academia!

Alexander Boehm
SAP SE, Germany
alexander.boehm@sap.com

Thomas Willhalm, Nicolae Popovici, Yazan Boshmaf, Hasso Plattner, Alexander Zeier, and Jan Schaffner.
SIMD-Scan: Ultra Fast in-Memory Table Scan using on-Chip Vector Processing Units.
In Proceedings of the VLDB Endowment, 2009.

I visited VLDB 2009 as a last year PhD student, looking for interesting topics to work on after graduating. One paper especially caught my eye, as the set of authors was quite remarkable: A joint team from Intel, SAP, and the University of Potsdam, including SAP co-founder Hasso Plattner, proposed novel techniques for ultra-fast, hardware-accelerated database scans [9].

With the ultimate goal of creating an in-memory database management system (DBMS) unifying both transactional and analytical workloads in mind (as outlined by Plattner's SIGMOD 2009 keynote [7]), one of the key challenges that the paper addresses is to strike a balance between high data compression and fast query processing. At first glance, these goals seem to be conflicting: High compression is essential for in-memory databases as it reduces the amount of costly DRAM that is required for data storage. However, the decompression effort during query processing creates additional overhead, potentially reducing system performance, which is *the* key value proposition of an in-memory DBMS.

In the paper, the authors show how vectorized processing using Single Instruction, Multiple Data (SIMD) instructions can be used to efficiently decompress bit-encoded data (for bit lengths up to 32 bits). Moreover, they discuss techniques for the fast evaluation of range predicates using vectorized instructions, thereby significantly speeding up the scan operation. A follow-up paper from ADMS 2013 [8] shows how to leverage the more modern AXV2 vector processing instructions up to the point where memory bandwidth is becoming a bottleneck, underlining how meaningful and powerful the proposed concepts still are in the context of today's modern CPUs.

Apart from being impressive engineering, the key insight from the paper is what remarkable opportunities emerge when people collaborate across various domains. By bringing together hardware experts and database engineers, the paper demonstrates the high impact that can be achieved with hardware/software co-design. While the database community has recognized this importance early on as e.g. demonstrated by SIGMOD's DAMON workshop which is successfully running since over 20 years now, papers like the one by Willhalm et al. are an outstanding example of the practical benefits that can be achieved.

To me, similar opportunities exist when looking "up" the processing stack towards the applications and application servers running "on top" of database management systems. Like with hardware/software co-design, there is the potential to optimize the end-to-end system performance by taking a joint look from an application and database perspective, leading to application/database co-design.

Pamela Delgado
HEIG-VD, HES-SO, Switzerland
pamela.delgado@heig-vd.ch

Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica.

Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center.

In Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI), 2011.

I started my PhD during the aftermath of the big data era, where its large-scale nature spawned several interesting systems and data management challenges. Many of the methods and principles introduced during that decade are still relevant, and some *incarnation* of them are used in today's platforms. While some influential papers, like MapReduce [2], are widely recognized, others have had a significant impact in ways that are less immediately visible. One such paper is *Mesos: A platform for Fine-Grained resource sharing in the data center* [5]. As I explored the state of the art in resource management during the early stages of my PhD, this particular work really stood out.

One particular feature of Mesos was its two level scheduling design. The original idea introduced in Mesos was to have a thin coarse-grained scheduling layer and leave the detailed decisions to the corresponding framework. Concretely, the fact of combining more than one type of scheduling co-existing in a cluster was inspiring for our work on hybrid scheduling.

Another aspect of this paper that resonated with me was the principle of including flexibility by design: the ability to accommodate a wide range of use cases (in the case of Mesos, Hadoop, MPI, and Torque batch scheduling) in order to use resources more efficiently. Flexibility is indeed something I still value deeply when designing systems today. This flexibility also applies to hardware heterogeneity, an aspect that is highly relevant in current Machine Learning ecosystems.

One more arguably relevant aspect is data locality. Over the years the importance of this aspect has changed from a non-issue for centralized systems, to crucial for distributed systems under potential network bottleneck limitations. In the GPU and LLM era this is an aspect that has gained even more prominence, motivating the revision of ideas as the ones exposed in Mesos.

Nowadays, one can trace the influence of Mesos in widely used systems for orchestration and deployment of applications, such as Kubernetes. Such systems offer a finer-grained resource management approach when sharing resources—one of the key ideas introduced in Mesos. Another such example is the idea of proposing a two-level scheduler, also seen in systems like YARN.

Of course, the proposed system was not flawless. For instance, Mesos offers resources, but frameworks can reject this offer, which has its impracticalities and challenges. However, the ideas and methods introduced in this work were and still are definitely out-of-the-box and inspiring not just to me, but also to other researchers and systems engineers. The imprint of its design and the principles it introduced have found wide adoption in a broad range of applications.

1. REFERENCES

[1] Christine Collet, Thierry Coupaye, and T Svensen. NAOS-Efficient and modular reactive capabilities in an Object-Oriented Database System. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 132–143, 1994.

[2] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[3] Stella Gatziu, Andreas Geppert, and Klaus R. Dittrich. The SAMOS active DBMS prototype. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, SIGMOD '95, page 480, New York, NY, USA, 1995. Association for Computing Machinery.

[4] Antonin Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, page 47–57, New York, NY, USA, 1984. Association for Computing Machinery.

[5] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, page 295–308, USA, 2011. USENIX Association.

[6] Norman W Paton and Oscar Diaz. Active database systems. *ACM Computing Surveys (CSUR)*, 31(1):63–103, 1999.

[7] Hasso Plattner. A Common Database Approach for OLTP and OLAP Using an in-Memory Column Database. In Ugur Çetintemel, Stanley B. Zdonik, Donald Kossmann, and Nesime Tatbul, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009*, pages 1–2. ACM, 2009.

[8] Thomas Willhalm, Ismail Oukid, Ingo Müller, and Franz Faerber. Vectorizing Database Column Scans with Complex Predicates. In Rajesh Bordawekar, Christian A. Lang, and Bugra Gedik, editors, *International Workshop on Accelerating Data Management Systems Using Modern Processor and Storage Architectures - ADMS 2013, Riva del Garda, Trento, Italy, August 26, 2013*, pages 1–12, 2013.

[9] Thomas Willhalm, Nicolae Popovici, Yazan Boshmaf, Hasso Plattner, Alexander Zeier, and Jan Schaffner. SIMD-Scan: Ultra Fast in-Memory Table Scan using on-Chip Vector Processing Units. *Proc. VLDB Endow.*, 2(1):385–394, 2009.

Being a Mid-Career Academic: Challenges, Rewards, and Reflections

Khuzaima Daudjee
Cheriton School of Computer Science, University of Waterloo, Canada

Reaching mid-career can feel like walking into a meeting where everyone assumes you're chairing, and you don't know that. You then realize that the role you hold in the room has changed, even if you don't feel particularly different. You've earned tenure, your name starts appearing on committees, and people have started asking you for advice as though you know what you're doing. The expectations change from proving you belong to more about deciding what kind of scholar, mentor, and community member you want to be. It is a stage in one's career that usually comes with more autonomy, more responsibility, and perhaps the realization that no one is keeping score except you.

Offering advice here feels a little presumptuous. I don't claim to have great wisdom; just experience, a few lessons learned and plenty still in progress. And because my research background is systems-oriented, my perspective leans toward building stuff and practice, and the belief that many things are clearer once you've built something. Hopefully, some of what I will say next will be useful; ignore what isn't, and feel free to smile politely while doing so (quiet eye-rolling is also acceptable).

Mid-career brings a small but important pause. You start noticing what kind of work you may be proud of, what kind you don't want to repeat, and where you actually want to spend your time. You also realize that no one is steering your trajectory except you, which is both freeing and mildly unsettling.

Much thoughtful advice has already been offered by colleagues on navigating the mid-career stage, so rather than creating a list of recommendations, I'll focus my reflections around four themes from my own experiences. I hope that some of these reflections would be a useful read also for colleagues who are earlier in their careers.

System Building

Let's talk about two flavours of systems. Assume a strong desire to make systems do something they currently don't do well, or should be able to do, given the resources available. For example, some of my work has focused on making systems more adaptive. What do I mean by adaptive? We've long asked two related questions: given a workload, how should system resources be managed? And given system resources, how should we handle workloads? These two intertwined questions suggest that the system itself should learn about a workload as it runs, inferring resource needs on the fly and dynamically adjusting allocation and utilization. This line of thinking has opened up an interesting set of problems for me: from monitoring, modeling, control, and systems architecture.

Another approach is to observe real systems in actual use and ask: wouldn't it be great if this system could do X instead of Y, or X in addition to Y? Consequently, the challenge becomes building the new capability in a way that is principled, robust and well-integrated. Underlying both approaches is a commitment to first-principles system design: identify the bottlenecks, reason about their causes, and then design mechanisms that address them in a generalizable way. A principled system doesn't just work -- it works for the right reasons.

This mindset opens choice in research directions not only because they are 'safe' or acceptable to the community but because they represent problems that we believe matter. The freedom afforded by being mid-career allows for pursuing bolder ideas – problems that may not have obvious benchmarks, baselines or guaranteed outcomes but that have the potential to redefine how systems work. These are often higher-risk directions, but they can be those that lead to significant insight and impact. At this career stage, it is justifiable to trust one's own thoughts of what is important and to build systems you want the world to have, rather than only ones that fit into established categories.

Mentorship

Mentorship is often mentioned in the context of early-career researchers, but it remains just as important at the mid-career stage. Mid-career, the challenges shift. Instead of narrowly focusing on generating publishable results, researchers are often in the phase of making

decisions that define their long-term research agenda and leadership roles. These decisions are harder and generally less structured. A senior mentor can provide an outsider's perspective when considering new research directions, deciding whether to take on administrative responsibilities, joining major grant proposals, or dealing with institutional issues. Mentorship at this stage is less about how to succeed day-to-day and more about how to steer: how to choose the kind of researcher, advisor, and contributor you would like to be over the next decade.

What makes a good senior mentor? Trust and candor. A senior mentor should be someone whose judgment you respect and whose values you can relate to, someone you feel comfortable asking anything, from questions about research directions to sensitive questions about conflict, opportunity, or burnout. Importantly, a good mentor is not someone who simply agrees with you: they challenge your viewpoint and may offer advice you don't necessarily want to hear, but that can help you to see and understand your situation more clearly. The value lies not only in the guidance, but also in the advice that can bring about re-prioritization or a change in perspective. Mid-career mentorship works best when it is grounded in openness and a willingness to consider discomfort as a signal of growth and not of failure.

Micro-Visits

Short, focused, research visits of one to three weeks can be an effective way to establish potential longer-term collaborations. Unlike a sabbatical, they require relatively less work to set up and can be scheduled around teaching or administrative responsibilities. The value of a micro-visit comes from time spent working together with colleagues and their students. These interactions can help to assimilate things that e-mail, Zoom, or even co-authoring a paper cannot: how people think, make decisions, and conduct research. A micro-visit can help gauge whether there are genuine shared interests and an intellectual fit. When these align, micro-visits can naturally scale up into co-advised students, shared grants, multi-year projects, and so on, leading to long-term collaborations built on mutual understanding.

Service

Service often becomes distinctive at the mid-career stage. Roles such as journal associate editorships, chairing PCs and organizing workshops stop being "extra" and become part of how we help set the tone for the field. What I've appreciated in these roles is the chance to work with colleagues whose judgment you come to trust. For example, putting together a call for papers and deciding what kind of work we want to encourage becomes, in some small way, a shared research conversation. These collaborations are often where one can learn quite a bit about how others put value on a piece of work, and how they balance openness with focus.

On the practical side, almost nothing is ever smooth sailing. Deadlines can slip, and reviews can go missing. Sometimes, we need to mediate tensions and deal with unbalanced workloads. There's no glamour in chasing late reviewers or dealing with difficult cases, but it is part of the job. What has helped me is watching how more experienced colleagues handle these issues, how they keep things moving without making them personal, how they simplify processes to avoid recurring pain points, and how they maintain fairness when there isn't a perfect answer. A key aspect of service is about learning how to work with others when things are not perfect.

Summary

Mid-career is less about checking boxes and more about direction. It is a period where uncertainty and freedom can co-exist, where the work becomes driven less by external expectations and more by what feels meaningful and interesting. It offers an opportunity to think and consider not just what we can do, but what we want to do, and how we want to show up for our colleagues, collaborators, and students. As you shape this next phase of your career, don't sacrifice the parts of life that make the work worth doing: your relationships, your health, your interests, and the things that matter to you. A career is not only what we build in papers and systems. It is also what we build in ourselves along the way.

Data Ecology: Understanding and Designing Data Ecosystems

Raul Castro Fernandez
The University of Chicago
raulcf@uchicago.edu

Data profoundly shapes our economic, political, and social ecosystems, yet we have limited control over its influence. Unchecked dataflows among agents can distort or undermine these ecosystems. Analyzing dataflows helps us understand their use and misuse, revealing opportunities to harness their value. Beneficial dataflows, such as hospitals sharing patient data, enhance healthcare outcomes, while harmful dataflows, such as personal data sold to self-interested brokers, cause substantial damage. Equally critical are absent dataflows—such as banks or governments withholding data due to competition or mistrust—that lead to unrealized value. Despite their substantial impact, unified methods to manage dataflows effectively are lacking. Current legal (regulations), economic (incentives), and technical (privacy technologies) interventions are developed independently without a clear evaluation of their collective effectiveness.

Principles of Dataflows. To address this gap, I propose the concept of *Data Ecology*: a research agenda aiming to uncover principles that drive dataflows and to design comprehensive interventions to steer them positively. A central problem of Data Ecology is: Given a desirable outcome for a data ecosystem (such as in a company, city, or government), what interventions will lead agents' actions to that goal? The research agenda comprises three main areas: (i) formalizing the fundamental questions; (ii) designing new interventions (examples provided below); and (iii) evaluating interventions across diverse ecosystems. While some literature has explored these questions, data ecology provides a new lens that brings existing work into a common framework, helping us advance our understanding.

Preliminary and Ongoing Work. My group has explored numerous data ecosystems, including data sharing [1] and data markets [2, 3, 4, 5]. Data markets illustrate the practical value of the Data Ecology approach. Typically, data marketplaces suffer from Arrow's Information Paradox, a situation where sellers refuse to release data before payment (since data is non-rivalrous and easily duplicated), and buyers resist paying before validating the data's usefulness. This uncertainty leads to fewer transactions, limiting overall market efficiency.

Applying Data Ecology's dataflow lens, we identified this uncertainty as the critical bottleneck and developed a technical intervention called data escrow [6] to resolve it. Technically, a data escrow acts as a neutral intermediary: sellers securely register their datasets with the escrow platform, and buyers delegate specific computations to evaluate the dataset's value. For example, the escrow could run a buyer's machine learning model on the seller's dataset, returning only performance metrics (e.g., accuracy improvements) and results that pass a leakage filter. This approach enables buyers to verify data quality and relevance without compromising the seller's proprietary information, effectively eliminating data leakage risks.

Ongoing research extends this work to internal and external data-sharing markets, exploring foundational questions about data's intrinsic value [7], such as why specific dataflows emerge and what constitutes a "good" or beneficial data ecosystem.

Opportunities for the database community. The database research community is uniquely positioned to contribute to technical interventions in Data Ecology. Potential research avenues include: 1) Controlling Dataflows: Investigating privacy-preserving techniques beyond differential privacy and data escrows, such as secure multi-party computation and zero-knowledge proofs, to manage dataflows securely and flexibly. 2) Enhanced Provenance Techniques: Developing novel or adapted provenance systems that can reliably trace dataflows across complex ecosystems, providing controlled transparency where needed. 3) Internal Data Market Design: Designing effective internal marketplaces [2] to efficiently transfer tacit organizational knowledge about data use, quality, and context.

The Road Ahead. The ultimate goal of data ecology is to provide a general theory and mechanisms for understanding and controlling dataflows. Data shapes our world, but the final form need not be fixed. Data ecology's tools are more critical than ever as data's influence on our world broadens and intensifies.

REFERENCES

[1] Raul Castro Fernandez Data-sharing markets: model, protocol, and algorithms to incentivize the formation of data-sharing consortia. *SIGMOD*, 2023.

[2] Raul Castro Fernandez, Pranav Subramaniam, Michael Franklin Data market platforms: Trading data assets to solve data problems. *VLDB*, 2020.

[3] Raul Castro Fernandez Protecting Data Markets from Strategic Buyers. *SIGMOD*, 2022.

[4] Javen Kennedy, Pranav Subramaniam, Sainyam Galhotra, Raul Castro Fernandez. Revisiting Online Data Markets in 2022. A Seller and Buyer Perspective. *SIGMOD Record*, 2022

[5] Boxin Zhao, Boxiang Lyu, Raul Castro Fernandez, Mladen Kolar. Addressing Budget Allocation and Revenue Allocation in Data Market Environments Using an Adaptive Sampling Algorithm. *ICML*, 2023

[6] Siyuan Xia, Zhiru Zhu, Chris Zhu, Jinjin Zhao, Kyle Chard, Aaron Elmore, lan Foster, Michael Franklin, Sanjay Krishnan, Raul Castro Fernandez. Data Station: Delegated, Trustworthy, and Auditable Computation to Enable Data-Sharing Consortia with a Data Escrow. *VLDB*, 2022

[7] Raul Castro Fernandez What is the Value of Data?: A Theory and Systematization. *ACM/IMS Journal of Data Science*, 2025

Angela Bonifati Speaks Out on Research, Grants, and Collaborations

H. V. Jagadish and Vanessa Braganholo

Angela Bonifati

https://perso.liris.cnrs.fr/angela.bonifati/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. It is my pleasure to host Angela Bonifati today. She is a Distinguished Professor of Computer Science at Lyon 1 University, Head of the Database Research Group at the CNRS Liris Research Lab, and a senior member of the Institut Universitaire de France. I'm H. V. Jagadish, and I'm glad that Professor Bonifati is with us today. So, let's dive in. Welcome, Angela!

You have many affiliations, one of which is the French University Institute, which may not be well-known to people outside France. Can you tell us a little about what that is?

Thank you, Jag, for the invitation. I'm really honored to be part of this series and to answer your questions today.

The French University Institute is a service of the French Ministry of Education that recognizes university professors across all domains in France. Only very few individuals are recognized. Around 2% of French university faculty are members of this institute. I'm a senior member, which means the share of members who are older than 40. What I receive as benefits from this institute is a waiver of part of my teaching service for the next 5 years, and also a national award of scientific excellence. There is also a small research grant from which I'm now hiring a postdoc.

Thank you. I was thinking about your research accomplishments (which you have many and we won't have time to talk about all of them today). In my mind, I put them in three main buckets: XML, schema matching, and graph databases. Of course, what you've done is not limited to these three, but this is just to create some structure. Could you tell us a little bit about how your research interests have evolved across these and other areas?

Yes, of course. I'm really happy to talk about this. I started to do research on data management during the XML hype, and this was during my PhD. At the end of my PhD, I began working on schema matching and mapping, and started to become interested in data integration problems. Later on, I was interested to explore the interplay between those areas and I also considered combinations of schema mapping with heterogeneous XML data. Starting from 2010, there was the explosion of Graph Data Management Systems, and I also started to work on these new NoSQL data systems. That was the start of companies such as Neo4j. I then continued working on data integration, schema mapping, and matching.

XML was over, if you remember, around those years, so I kept working on these two areas of graph databases and data integration. Recently, I worked on transforming graph data, where the latter can be obtained from any kind of format (even relational data and time series), to actually integrate this into a property graph. As you can see, the three areas are not siloed, but I always kept studying the interplay between them at an abstract level, with concrete contributions.

I guess in particular, graph databases and more broadly, graph processing and curation, is what you're really focused on lately, as far as I can tell. Can you tell us a little bit about your perspective about these areas? What's interesting and cool about them, what makes you passionate about these?

They are a very interesting new technology for data management, in my opinion. There is a wide landscape of graph engines and graph data systems that have appeared since the inception of graph databases in 2010. What happened is that there were so many tools and so many engines, and also so many query languages behind these tools. And these languages were not unified. That's why, about two years ago, there was this effort, driven by academia and industry, to standardize these languages. In particular, in 2023 and 2024, the first standardized Graph Query languages, namely SQL/PGQ and GQL, appeared as a first version of the standards for query languages for property graphs. And these are the first standards after SQL, after the standards that we know for relational databases.

That's why I see this area as really flourishing now. And I was lucky, because during the pandemic, I was part of the LDBC working group, which was working in liaison with the standardization body. I was working on property graph schemas and constraint languages, and these were published[1] in the Industrial Track of SIGMOD in collaboration with people from the industry. We have to think about the fact that some of these standards will now be implemented in the tools. So all these companies that are part of this effort will take their own language and implement some of the features of the standard. So, that's why I'm really excited about this area. And then, of course, I'm a professor, so I had grants on these topics.

And as you know, there are a lot of things that one can do on top of these standardized languages. I was working on graph transformations using standardized languages, and now, lately, I'm also exploring the boundaries between graph databases and artificial intelligence, especially for curation processes.

In terms of grants and grant-making, you've recently obtained an ERC Advanced Grant, which is a high recognition for leading researchers in Europe. I believe it's the largest kind of grant that researchers in Europe strive for. It's a very competitive thing, so

[1] Renzo Angles et al: PG-Schema: Schemas for Property Graphs. Proc. ACM Manag. Data 1(2): 198:1-198:25 (2023).

congratulations! Can you tell us a little more about your vision for this grant and the research that you are about to start, and what you think this will lead to?

Thank you! Yes, I'm really excited about this grant. As you said, this is the most prestigious individual grant that European researchers can get in their careers, and it's very competitive. So, I was extremely happy to receive this grant that will start on December 1st this year. So the name of this grant is ERC GO-Y (pronounced "Go Why"), and guess what? This grant is on unifying graph databases and causal models used in AI.

It starts from the observation that in our data systems (and this observation can also be made for relational databases), we have been focusing mainly on processing and pre-processing. But here in this grant, I envision a radical shift: having data systems that are actually guided by causality, by being able not only to return results, or to be able to do pre-processing, processing, or transformation of the data, but also to return the explanation behind the processes. And what I see behind the explanation is to be able to return the causes that are behind the effects.

In this 5-year project, I intend to address the development of theoretical foundations and also tools to be able to understand the causes behind the data and to be able to execute and evaluate these causal graph operations in our data systems. I work on graph databases, so when I wrote the grant, I realized that the theory of causation uses directed acyclic graphs to encode cause-and-effect relationships, and this can actually be encoded within a graph database. And this opens up a lot of research topics and data management tasks that, right now, are not there because people do these things with *ad-hoc* scripts. Additionally, they do it for one dataset or one causal question at a time, whereas in data systems, we can work with any dataset, and we can work with any query. Therefore, I think there is a lot of potential there to unify these areas that have been kept separate so far. The GO-Y grant will eventually explore the interplay between data management and causal AI, leading to implement causal processes in a database.

So, you mentioned that this was an individual grant, so it's all you? Are there other collaborators involved?

No, it's just for me and my team in Lyon. It's actually a huge grant, it's almost 2.5 million euros, only for my team. There is no consortium behind this grant.

No wonder it's so prestigious and so difficult!

Turning to some non-technical questions, when you have a big grant like this, the odds of getting it are very low, even for very good people, doing very good work. It's highly competitive, as we were talking about. So, what's your philosophy behind this, and what would your advice be to others who might be thinking about this? Should researchers go after these big prizes with low odds, or is it better to put energy into things that are more likely to succeed, even if they're not as big? How do you balance these?

It's a difficult question. I don't know whether I can give credible advice there, because it depends on the individual experience. You can apply to an ERC depending on the stage of your career. They have the Starting Grants, then they have the Consolidator Grants, which are targeted for mid-career researchers, and then they have the Advanced Grants, which are for senior people like me. I never applied for Starting or Consolidator grants. I was working with smaller grants before, also with the European grants, in a consortium with other people.

I think one needs a bit of training with other grants that are probably much easier to get before going to ERC. In my case, it was useful to have been funded by other agencies before ERC. But, I would certainly recommend it to people who are at a certain stage of their career, and even junior researchers, if they feel they have reached the peak of their career, that they have a lot of contributions, and they are doing a lot of research. Even if the chances are low, nothing is lost, because we know that writing a grant brings us to do some brainstorming, think about our future research topics, and the grant can always be funded by other agencies. So I would definitely recommend researchers who feel they are ready to go for an ERC, to go for an ERC.

I have a bare minimum requirement [for collaborating with Young PhD students and post-docs], that is I want to work with people who can meet the deadlines, and who are very serious about their work.

Can you tell us about your own professional arc? You started in Calabria, and are currently in Lyon, and have

been in multiple places along the way: first in Italy, then in France. How did you decide to stay in some place, or to look for better opportunities?

In my case, I was trying to choose a place based on the actual possibility of doing research in that place. I always wanted to do research. You know that, especially in France, but also in Italy, one can be consumed by other activities, right? It could be an admin or a huge teaching load. I tried to choose the place based on the possibility of doing research, and I always look for endowed research chairs as part of the job positions. These are very common in France, similar to the French University Institute. I had a couple of years in which each year I was basically waiving my teaching service. And this explains why I moved around some places, because whenever I felt that I did not have enough time to do research, then I told myself, "Maybe I should look for a new place."

In terms of countries, I've been lucky because I moved basically from Calabria, in southern Italy, to France, well, to northern France at the very beginning, and then to Lyon, which is in southern France. But the two countries are very similar, too. I also had a few extensive visits in North America and Canada.

I'm quite happy about what I did, because I've always been driven by the intention of doing more research. I think the various parts are part of the whole, right?

> ***What I look for in my collaborations is to learn from others.***

Even though you've been at multiple places, they've all, as far as I know, been academic, or at least quasi-academic, like INRIA, and so on. How do you feel about working in industry or industrial labs?

That is one possibility that I haven't explored. During my PhD, I did an internship in industry. I was at HP Labs in California, and I had a great experience there, both personally and professionally. I was working in Umeshwar Dayal's team for a 3-month internship. We published a paper[2] in the industrial track at VLDB 2001. But as you remember, the industry was hitting a crisis after the year 2000, laying off many people. And even though I liked this industrial experience, I thought I should stay in academia, and I looked for a postdoc. And then, starting from my postdoc, I went directly to academia, and I basically didn't regret it.

So I never felt the need to go into the industry, but of course, I don't exclude it for the future. I've always been collaborating with companies, so everything is open. I don't exclude it.

You've had many collaborators: juniors and seniors, from various places and various levels. But one thing that kind of struck me was that even when you were a junior researcher, you had already developed collaborations with many senior researchers. Could you say a little bit about your overall philosophy for collaborations, and then perhaps also specifically about how it came to be, and how you've had so many senior collaborators early in life? What are your views on that?

When I was at these visiting periods in North America, all the collaborations happened there. I also had collaborations with people in Europe. You are right, maybe my collaborations were always with people who are more senior than I am. What I look for in my collaboration is to learn from others. So, it was not intended, but you know, we learn from each other, and especially from people who are more senior than us; we learn a lot, because they have a lot of experience. This is something that we already experimented with during our PhD, because our PhD advisor has a lot of experience. It worked well. I learned a lot from all my collaborators.

I started these collaborations by chance, let's say. Sometimes I was visiting a place, and I was expecting to collaborate with someone, but this person didn't have time. So I just started talking to another colleague, and then we started to collaborate. You see, I would not say I always choose people who are more senior than I am, but it just happened, and in the end, the result is quite satisfactory. And what are the qualities that I value in my collaborators? If I also consider the collaborators in my team, like the young PhD students or postdocs, I have a bare minimum requirement, that is I want to work with people who can meet the deadlines, and who are very serious about their work. That's all.

That's very succinctly put. So, asking you a geopolitical question: as a senior researcher in Europe, how do you view research in Europe versus the rest of the world?

[2] Angela Bonifati, Fabio Casati, Umeshwar Dayal, Ming-Chien Shan. Warehousing Workflow Data: Challenges and Opportunities. VLDB 2001: 649-652.

Are there differences, both in terms of subject matter and in terms of style, that you'd like to comment on?

That's a difficult question. I'm a European faculty member, and as a researcher and educator, we are all confronted with generative AI and with AI in general. We can see that this AI landscape is rapidly evolving, and everyone is trying to understand how to work with it, so that we can only take the benefits of this new technology, and we can avoid the shortcomings.

Europe has been working on some of the regulations for AI. As you know, the EU AI Act was designed by the European Commission. But what I notice in Europe is that it is difficult to find agreement across European countries, and unless it comes top-down from the European Commission, then it is difficult to get all of the European countries aligned, whereas it would be good to have that. For instance, even in France, different universities or research institutions are deciding their own AI policies. Whereas we could decide together what the AI policy should be, right? This would actually let us control this technology.

It is very difficult to say what is the positioning that we should have in research. I think more investment is needed. In France, for instance, the government has been funding these Research Institutes on AI. But there were a few of them, and they were given to some cities at the detriment of the others. I think more investment is needed in AI if we want to be competitive in Europe with respect to other continents.

I think we need more discussion, too. As computer scientists, we need to discuss at all levels: as educators in our universities, as people who are in charge of deciding the future of conferences, etc. All the actors should be involved in discussions. I think this is bottom-up.

I also wanted to mention something else. I was reading an article about the latest United Nations Summit, where Yoshua Bengio mentioned the fact that now Gen AI is becoming even more critical in security and peace. And now in Europe we are there, right? I mean, we have two wars on our borders. So, I think that, as Europeans, we should really take it seriously. Having our own AI technology is important. With that, we will not only be competitive in the IT sector but will also take care of these critical issues, such as security and peace.

Talking about the research atmosphere more broadly. As a senior woman researcher in a community with very few women, are there comments that you'd like to make to the community as a whole? Both with respect to the database research community overall, and also specifically with respect to your position in France.

I think mentoring is very important. I'm a senior woman researcher in a community with a few women. What I do is I try to mentor my (male or female) collaborators, so that they learn how to navigate through their career milestones. I wrote an article[3] on advice to mid-career researchers in SIGMOD Record that appeared in July 2024. The gist of it is that, when somebody is in their mid-career, they have passed this stage of independent critical thinking. I was mentioning my own experience. I've always been driven by curiosity, curiosity about new research subjects, and then writing, writing down your ideas. These were the suggestions. I don't think this is a female-to-female suggestion. I mean, it could be from female to male, for young people or mid-career researchers. And I think mentoring. Mentoring both ways. I mentor other people, and I'm mentored by people who are senior than me. I think that's the key to having more people join this community.

[As Europeans,] Having our own AI technology is important. With that, we will not only be competitive in the IT sector but will also take care of these critical issues, such as security and peace

I think that's a very positive statement, and that leads me to ask you about your view of the community and your vision for it as the new chair of SIGMOD. What are your plans? What is your vision? What are the things that you'd like to make better for all of us?

Thank you for the question. I'm really honored to have been elected as the SIGMOD Chair. I started my term on July 1st, and in the last few weeks, we have already been discussing many issues. I see SIGMOD as part of the SIG ecosystem of ACM, apart from being an executive committee of the SIGMOD conference, and also co-sponsoring other conferences. The fact that it has these two aspects makes me busy on both sides.

The biggest challenge now on the ACM SIG landscape is the transition to open access. As you know, in January 2026, ACM will start adopting open access. This means

[3] Angela Bonifati: The Long and Winding Road to Mid-Career Academia. SIGMOD Rec. 53(2): 48-51 (2024).

that people who don't belong to institutions that have signed the open access agreement with ACM will have to pay the APC fees. This is a big transition for SIGMOD, and not only for SIGMOD, but for all the other conferences that are sponsored by SIGMOD and all the other ACM SIGs. So I'm really busy with this.

There are also other long-term challenges that are awaiting us, such as AI in data management, ethics in the review processes, and changes in the political landscape that affect our community. All these challenges are equally important, both at the ACM level and at the conference level. So, I intend to work on both aspects: SIGMOD as part of this ACM ecosystem, and SIGMOD as an executive committee of the ACM SIGMOD conference, and all the other SIGMOD-sponsored conferences.

Thank you. Best of luck in terms of what you're planning to do. I hope that's all very successful, and I look forward to it.

Thanks.

I wanted to talk about one of the things that our community has started to address in a very small way, which is helping people balance their professional work with family. A few conferences now have some help with childcare, for example. You have brought your daughter to some events, and you're managing all of the amazing things that you've been doing recently, while also having a young person that you're responsible for at home. So, can you talk about both how you manage your balance in terms of time, as well as are there things that we as a community could do to make this easier on others?

That's a difficult question. I think that the database community already has many initiatives to help young parents. There has been daycare, as you mentioned, in a few of the last editions of conferences. And this is very helpful for parents who want to attend the conference and bring their children. This is very important. This was not the case when my daughter was young. I remember the first years, I could not travel to conferences because I could just not bring her. The first time I brought her to an event, it was in a Dagstuhl Seminar. By chance, by going and checking the Dagstuhl website, I discovered that they offer a babysitter. And then I could take my daughter, because the babysitter would take care of my kid during the entire seminar. She was 5 years old at the time, and she had a great time. When we came back, she said, “Why don't we go back again to that castle?” And the reason why I brought her, of course, was to spend more time with her, because my partner could not keep her during that week, and also because I wanted her to learn about my job. And I think this actually happened.

Since then, whenever I can, if I have an event during summer holidays when she can come, I just bring her. And I think this is great, because she learns about my job, she knows why I'm traveling, why she cannot come, because she's at school, and I'm not at home. So, I would suggest that parents do this. And of course, the community can help there, as they're already doing, and maybe more, with solutions. Solutions for keeping children and having daycare during conferences.

So, going beyond work-life balance, I guess there's just a question of time management and juggling many different things. So, aside from having a school-aged daughter, right now, you're on the VLDB Endowment Board, you're a general chair of VLDB next year, you have a large ERC project that you're just starting, and you have been chair of SIGMOD now for the past couple of months. So, how do you do all of this?

I don't know. I'm a very organized person. I didn't ask myself this question before, but I think I'm very good at time management, and I think I even learned to be better at time management after becoming a parent, because I knew that in the evening, for instance, I could not work because I have to devote time to my kid. So, I just work on various tasks during the day. If I have to say, I do it in a round-robin fashion, in the sense that I dedicate fixed slots during the day to each of these activities. I mean, it worked in the past, so I'm very confident that it will work in the future. Of course, I have these, as you said, big challenges with VLDB 2026, and then ERC, and then SIGMOD chair. I haven't done this before, but I'm still optimistic. Let's see...

By the way, I also enjoy extra work activities, such as spending time with my family, going for long walks, and cooking, especially Italian cooking.

So, I guess we need to have another edition of Speaks Out, where you will share your recipes with us.

Of course! With pleasure, with pleasure.

But for today, I just want to thank you.

Thank you, Jag.

Retrieval-augmented Generation (RAG): What is There for Data Management Researchers?

A discussion on research from a panel at LLM+Vector Data Workshop @ IEEE ICDE 2025

Arijit Khan
AAU

Yuyu Luo
HKUST (GZ)

Wenjie Zhang
UNSW

Minqi Zhou
Huawei

Xiaofang Zhou
HKUST

1. INTRODUCTION

Large language models (LLMs) enable the state-of-the-art in language processing by framing diverse tasks–from code synthesis and healthcare to finance, digital assistance, and scientific discovery–as next-token prediction problems [38, 53, 60, 65, 72, 20, 68, 76, 32]. In addition, LLMs enable automation in data science and engineering, optimizing processes such as data analysis, manipulation, querying, interpretation, research, and education [33, 7, 8, 22, 24, 42, 77, 37, 43, 44, 66, 49].

LLMs encode probabilistic token patterns instead of maintaining explicit knowledge structures, which (1) constrains multi-step reasoning under the next-token prediction paradigm; (2) ties outputs to static, pre-cutoff training data–undermining performance on evolving knowledge tasks; and (3) lacks a built-in factual verification mechanism, resulting in hallucinations [25].

Retrieval-augmented generation (RAG) enhances LLM outputs by integrating dynamic, authoritative external knowledge sources rather than relying solely on static pre-training data [29, 74, 56]. Advanced embedding models convert heterogeneous datasets, e.g., text, multimedia, graphs, and tables into high-dimensional vectors that preserve semantic similarity, storing them in vector databases such as Weaviate, Chroma, FAISS, Milvus, Pinecone, Qdrant, and Vespa [46]. At query time, RAG performs a vector similarity search to retrieve semantically relevant information, appends this context to the LLM prompt for in-context learning, and thus mitigates hallucinations while boosting accuracy, transparency, and capability without costly retraining or fine-tuning.

The retrieval phase in RAG mirrors classical information retrieval–queries return documents based on relevance metrics [48]–but it uniquely injects retrieved content into the generative pipeline for LLM in-context learning, yielding responses that are both contextually coherent and precisely tailored. This approach raises key technical questions: Why vector similarity search via approximate nearest neighbor (ANN) methods performs effectively in high-dimensional spaces [23]; which innovations in indexing and quantization mitigate the curse of dimensionality for dense vectors [26]; and to what extent dataset characteristics, embedding models, or relevance metrics drive retrieval performance. While data management has long addressed high-dimensional time series, spatial, multimedia, and geometric data, the recent surge in vector workloads has catalyzed novel indexing and search algorithms, specialized vector databases, and hardware accelerators [58]. These advances power enterprise RAG deployments–Microsoft Azure AI [6], DoorDash delivery [19], Pinterest's Text-to-SQL [45], and LinkedIn customer service [67]–promising to transform search technologies [36].

Nevertheless, RAG systems remain brittle: Partitioning large external knowledge bases into vectorized chunks dilutes critical details, severs global interrelationships needed for multi-hop queries, and depends on embedding similarity that often retrieves contextually irrelevant content. Consequently, conventional RAG underperforms on queries demanding synthesized, comprehensive insights, driving the emergence of variants, e.g., GraphRAG [21], multimodal RAG [64], Agentic RAG [52], knowledge-augmented generation (KAG) [35], tool-augmented generation [51], table-augmented generation (TAG) [10], and cache-augmented generation (CAG) [16].

At the LLM+Vector Data'25 workshop (ICDE 2025, Hong Kong, China), the focus was on data management challenges and opportunities arising from LLM-vector data interactions [27]. A panel featuring Yuyu Luo (Hong Kong University of Science and Technology, Guangzhou), Wenjie Zhang (University of New South Wales), Minqi Zhou (Huawei), and Xiaofang Zhou (Hong Kong University of Science and Technology), moderated by Arijit Khan (Aalborg University), discussed emerging RAG opportunities at the intersection of data science, data engineering, and data-centric AI/ML methodologies. The session attracted over 100 attendees.

2. OVERVIEW OF PANEL DISCUSSION

The discussion was organized into three broad themes.

Theme 1: Future prospects of RAG, vector data management, and LLMs for data management

Background. Rapid advancements in large language models and their integration with external data systems are transforming the computational landscape [18, 57, 17, 59, 47, 71]. With models like Gemini 1.5 Pro boasting a 2-million token context window [28], questions arise regarding the necessity of traditional RAG architectures when semantic search can be performed by long-context LLMs. Meanwhile, the exponential growth and complexity of vectorized data present new challenges in data management, driving the need for advanced solutions that efficiently store, retrieve, and update heterogeneous data. These trends raise several critical questions about the future direction of the field.

Q1. *With long-context LLMs like Gemini 1.5 Pro–having its impressive 2-million token context window–emerging, how do you foresee RAG architectures evolving to fully exploit these extended capabilities? What are the most significant recent innovations and future trends in RAG that excite you the most?*

Yuyu Luo. Long-context LLMs with million-token context windows create new opportunities for RAG. Instead of segmenting external knowledge into small, isolated chunks, we can retrieve larger, more semantically coherent units. Future RAG pipelines may intelligently determine how much retrieved context to include in the prompt. For example, with the availability of extensive context windows in models such as Gemini 1.5 Pro, entire documents or interconnected knowledge segments could be retrieved and integrated, thus preserving global semantic relationships that previous approaches might have disrupted. Moreover, retrieval strategies could dynamically adapt context selection to efficiently utilize the available context window, ensuring optimal model performance without excessive computational overhead.

However, these expanded context capabilities introduce significant efficiency challenges, particularly around memory management. This opens exciting research opportunities for database researchers. Database-inspired techniques, such as KV caching mechanisms, could optimize the reuse of previously computed representations and mitigate the overhead of repeatedly processing similar context segments [34, 61].

Wenjie Zhang. RAG architectures are evolving to take advantage of the capabilities of long-context LLMs through techniques such as fine-grained retrieval, iterative reasoning, and dynamic compute allocation. Fine-grained proposition retrieval enhances information density by reducing noise, ensuring that only highly relevant content is used by the LLM. Iterative approaches like IterDRAG [70] optimize multi-hop reasoning by refining retrieval and generation cycles. Dynamic compute allocation models intelligently distribute resources, balancing cost and efficiency for large-scale RAG deployments. Hard negative fine-tuning further strengthens robustness by addressing challenging retrieval scenarios, improving accuracy in diverse domains. Emerging trends include multimodal RAG, which integrates text, images, and audio for richer responses, interactive systems that allow real-time user feedback, and integration of knowledge graphs for enhanced structured contexts. They facilitate precise and efficient inference in long-context settings, significantly boosting performance on knowledge-intensive tasks.

Minqi Zhou. The current long-context LLM capability, like Gemini 1.5 Pro, does create a set of new opportunities to enlarge the external information that can be used. Meanwhile, our findings underscore the urgency of refining new RAG techniques for practical deployment. (1) Integrate the document semantics together with the document structure: In the document/mail QA system, it is easier to get the correct answer when taking the total substructure of the document/mail into account. (2) Understand the versions between the document on the same topic: While building systems or projects, it is important to understand the concept evolution in the different versions of the system/project architecture or system design documents. (3) Adhere to the original text: For example, it is a must to return the original text from the issuance of government documents and the issuance of legal documents in the corresponding QA systems. As for the sophisticated RAG systems in practice, they do require different innovations to provide the correct answers (i.e., "no one-size-fits-all").

Xiaofang Zhou. While long-context LLMs showed exciting possibilities, they also introduce new challenges. First, handling such large inputs can significantly increase the time to first token (TTFT), which directly impacts user experience if it becomes too slow. Second, these models still rely heavily on GPU resources, which are considerably more expensive than CPU-based solutions. In my view, this means that CPU and memory-efficient RAG systems still hold a strong advantage when it comes to retrieval efficiency and cost-effective use of computing resources. Rather than seeing RAG and long-context LLMs as competing approaches, I believe they are complementary. Combining them effectively could lead to more powerful and efficient systems. One particularly exciting research direction is how to design optimal system architectures that intelligently integrate RAG with LLMs–leveraging the strengths of both. This hybrid approach has great potential to shape the future of retrieval-augmented generation.

Q2. *What emerging issues are we seeing in vector data management, and how can advanced data management solutions mitigate these complexities?*

Wenjie Zhang. High-dimensional embeddings, such

as those with thousands of dimensions for text, images, or videos, drastically increase storage and computational costs for vector data management. Sparsity in these high-dimensional spaces reduces the efficiency of similarity searches. Recent research on approximate nearest-neighbor search [75] uses hybrid approaches that combine locality-sensitive hashing (LSH)'s rapid coarse filtering with the efficient exploration performance of approximate proximity graph (APG) for large-scale datasets, reducing storage needs and improving query efficiency to support scalable RAG and LLM deployments.

Minqi Zhou. In term of the industry deployment, we find several issues still need to be fixed, especially for the vector index. (1) Vector indexes must support continuous, incremental updates without degrading top-k recall, precision, or query latency–yet existing structures (e.g., IVF, HNSW, HCNNG, DiskANN) exhibit performance drops even on simple insertions. (2) We must also reduce index-building times, which currently stretch into hours once vector counts exceed one billion.

Xiaofang Zhou. Vector data management today faces a wave of interrelated challenges. As models generate billions or even trillions of high-dimensional embeddings, systems must scale accordingly while maintaining efficiency. Storing dense vectors at this scale puts enormous pressure on memory, demanding techniques like ultra-low-bit quantization and intelligent tiering. For instance, frequently accessed ("hot") vectors can be kept in fast memory tiers, while infrequently used ("cold") data can be offloaded to cost-effective storage such as NVMe or cloud object stores. Beyond traditional nearest neighbor search, modern workloads increasingly demand support for set-level queries, hybrid retrieval that combines semantic similarity with symbolic filtering, and operations over multimodal or multilingual embeddings. These introduce significant algorithmic and system-level complexity. Meanwhile, most existing infrastructures are ill-equipped to handle continuous, low-latency updates or support rapid incremental inserts without costly index rebuilds. In my view, addressing these challenges requires more than marginal improvements in algorithms. It calls for end-to-end system design. Real progress will come from holistic hardware-software co-design, where approximate retrieval, adaptive indexing, and tier-based resource management are seamlessly integrated into scalable, production-ready systems.

Q3. *What specific roles will RAG–and broadly, LLMs and generative AI–play in advancing database research and shaping the next-generation DB systems?*

Yuyu Luo. LLMs and RAG are poised to become integral parts of future database systems. A natural entry point is the extensively studied Text-to-SQL task [30, 31], where an LLM translates user questions into executable queries. RAG enhances this pipeline by dynamically retrieving relevant database documentation, such as schema definitions, data dictionaries, and data semantics, to generate accurate SQL queries even for unfamiliar databases [40, 78, 41]. Moving beyond Text-to-SQL, RAG fits naturally here by pulling in relevant data from the database (or other sources) as needed for the LLM to produce accurate answers. This could transform how we do data analytics, making it more accessible – imagine getting a narrative report of your data, compiled via SQL queries and explanatory text from an LLM [50, 69].

Wenjie Zhang. LLMs can synthesize data to fill gaps in databases or expand datasets; and RAG, by retrieving relevant knowledge, ensures that the generated data align with real world scenarios and requirements, thus improving data quality and usability. Additionally, LLMs can analyze users' query patterns and automatically generate optimized execution plans; and RAG can retrieve historical query logs to provide additional insights for optimization, further enhancing query efficiency.

Minqi Zhuo. RAG, LLM, and generative AI will enlarge the database research scope. In the traditional enterprise data warehouse system field, RAG and LLM are able to accelerate the ETL phase for adding new data sources or creating new data marts and to enhance the quality of the data integration from isolated data/database sources inside the enterprise.

Xiaofang Zhou. LLMs are reshaping the role of DB systems–from passive stores of structured data to active participants in semantic retrieval, reasoning, and generation. As users increasingly pose natural-language queries over heterogeneous data–text, tables, images–natural language becomes the new query interface, embedding based retrieval emerges as a core operator, and structured and unstructured data are increasingly fused within a unified retrieval layer. This paradigm shift calls for systems that blend the precision and consistency of structured storage with the flexibility and recall of semantic retrieval. Researchers could begin by developing compound query planners that coordinate exact lookups and approximate searches within a single framework, along with storage engines that natively support both relational and vector data. New benchmarks and evaluation metrics are needed to reflect the multimodal, generative, and user-facing nature of next-generation DB systems. Meanwhile, it's worth keeping in mind a broader question: Next-generation database systems may forgo full human interpretability in favor of machine-driven adaptation, optimization, and reasoning.

Theme 2: Deep dive into the technical frontiers of RAG and synergy with data management

Background. Recent breakthroughs in approximate nearest neighbor search (ANNs) and scalable vector DBs

are revolutionizing the indexing and retrieval of large-scale, high-dimensional, dense vectors, resulting in faster and more precise information synthesis. Concurrently, the development of specialized RAG variants–such as GraphRAG, KG-RAG, Agentic RAG, and multimodal RAG–demonstrates the capacity to enhance LLMs by integrating domain-specific structures and multi-modal inputs. This dynamic landscape warrants technical investigation into merging these innovations into cohesive, high-performance systems.

Q4. *What emerging research directions in approximate nearest neighbor search and vector databases hold the most promise, and how can these developments integrate with modern data management strategies?*

Wenjie Zhang. Emerging research in ANN search and vector databases includes advancing toward scalable, multimodal, and real-time solutions that integrate seamlessly with modern data management tasks in the generative AI era. Key directions include learned index structures for intelligent retrieval, dynamic ANN methods for streaming data, knowledge-aware ANN supports to reason over graph and LLM-augmented systems, multimodal search across text, images, and graphs. These developments are transforming vector search into a core component of next generation data platforms.

Minqi Zhou. In the age of agentic AI, a lot of agents will be created on the devices side, such as phone, Pad, PC, etc., where vector databases would play a critical role in enhancing the semantic search capability. On such devices, how to reduce the energy consumption when building vector index, and how to lower the storage capacity of vector index without sacrificing the top-k search recall/precision, are crucial research directions.

Xiaofang Zhou. In my view, there are many interesting topics, such as Maximum Inner Product Search (MIPS), dynamic indexes that let you add or remove vectors on the fly, and new quantization methods that dramatically cut memory without killing accuracy.

Q5. *With the RAG landscape rapidly diversifying into forms like GraphRAG, KG-RAG, Agentic RAG, multimodal RAG, and beyond, which features do you find most compelling? What emerging frameworks do you envision that can harmonize their unique capabilities?*

Yuyu Luo. Each RAG variant brings something unique to the table–several of their features stand out to me. GraphRAG and KG-RAG are compelling because they retain relationships and structure–rather than retrieving isolated text passages, they leverage connections in a graph or a knowledge base. This means answers can follow a chain of facts or traverse a hierarchy, which is powerful for complex reasoning. Agentic RAG is exciting for giving the LLM more autonomy: The model can iteratively decide what to retrieve or which tool to use next, almost like it's doing its own research. And of course, multimodal RAG extends capabilities beyond text–a system that fetches relevant images or audio along with text can provide much richer responses. To get the best of all worlds, it is necessary to develop unified frameworks capable of integrating multiple specialized retrieval modules [73]. A promising direction involves creating an orchestrator that modularizes various knowledge sources and external tools into a cohesive pipeline, e.g., extracting structured facts from knowledge graphs, retrieving detailed context from vector-based indexes, or invoking computational APIs as needed.

Wenjie Zhang. Graph-based variants, such as Graph RAG and KG-RAG, appeal to me most, because a graph view makes hidden links explicit. In classic text only RAG, the LLM struggles to track cross sentence jumps or connect evidence spread over multiple documents. Adding nodes for entities and edges for their relations turns those implicit jumps into a visible path, so the model can follow a multi hop chain, keep the reasoning transparent, and answer complex questions with higher accuracy [54]. Agentic RAG then lets the model decide which extra tools to invoke, such as search, calculation, and coder, whenever the retrieved context is not yet enough. This keeps the chain-of-thought short and grounded, and it brings the workflow closer to real applications that mix retrieval, analysis, and generation. A complete retrieval layer, however, needs to look beyond text. Human knowledge arrives as images, video, audio, and sensor streams, not just prose. Multimodal RAG aims to meet that reality. If a system can fetch a diagram, a short clip, and a passage of text about the same topic, then combine them through a common representation, its answers become richer and better grounded. Looking forward, I expect a unified framework with an agent planner at the core, a bank of modality specific retrievers, and a shared graph as the hub. The graph would align text, images, audio, and tables in one structure and give the model a single stage for orderly reasoning. Early tests with models that reason over both structured and unstructured data already show clear gains, suggesting that such a unified framework is within reach [55].

Minqi Zhuo. In the time being, a set of RAG frameworks are emerging, like GraphRAG, KG-RAG, Agentic RAG, multimodal RAG, etc., but each of them has its own advantages in solving a set of real application scenarios. Based on our industry deployment observations, as discussed in Q1, we found that it requires different capabilities to serve different application scenarios, e.g., document/mail QA system, the issuance of government documents, or the issuance of legal documents QA system, and it is very difficult to find one framework which is able to server all the purpose.

Xiaofang Zhou. What's most compelling about the evolving RAG landscape is its movement from simple text retrieval toward richer, context-aware reasoning. Each variant–GraphRAG, KG-RAG, Agentic RAG, multimodal RAG–brings a distinct strength: structured knowledge grounding, entity-level disambiguation, reasoning over actions, or cross-modal alignment. To harmonize these capabilities, the main challenges lie in either modular orchestration or alignment of diverse feature spaces. Modular orchestration requires a unified control layer capable of routing sub-queries to specialized retrievers and aligning their outputs semantically. On the other hand, feature alignment demands a unified embedding space where each modality's encoder projects data into comparable vectors. Harmonizing these modalities under one retrieval-planning interface–while managing latency and ensuring trustworthiness–will advance the effectiveness and versatility of RAG systems.

Theme 3: Relevance to Academia and Industry

Background. The rapid evolution of LLMs+RAG solutions is transforming academic research and industrial applications in various sectors. This shift forces researchers to continually innovate while sidestepping common methodological pitfalls and industry to update technical competencies and operational frameworks. Besides, the integration of these technologies prompts critical discussions on fostering an equitable research landscape and mitigating monopolistic influences.

Q6. *Given the rapid surge in LLM literature and emerging solutions, how can academic researchers stay abreast of new trends while ensuring that their work remains impactful? What common pitfalls should new Ph.D. students avoid when entering this domain?*

Yuyu Luo. I suggest regularly scanning key conference proceedings and arXiv for emerging work – often, just reading abstracts or discussion posts can flag important trends without needing to deep-dive into every paper. It's also useful to follow a handful of experts or community newsletters that digest new breakthroughs.

Grounding research in real-world needs and fundamental questions is essential for impact. The open-source movement accelerates innovation, and my research group actively contributes to projects tackling complex data management and analysis [2], including Text-to-SQL systems [4], data analysis agents [1], and LLM agents [3].

For Ph.D. students, common pitfalls include chasing trends without a clear research question and neglecting rigorous evaluation. Impactful work demands careful experimentation and validation, not just demos. I also encourage students to embrace interdisciplinary research at the DATA+AI frontier.

Wenjie Zhang. I think it is essential to embrace new technologies like LLMs to tackle longstanding and emerging real-world data management challenges. Rather than viewing LLMs as separate from traditional database systems, we could explore how they can augment tasks such as query understanding, query answering, and data quality management. Staying impactful means focusing on how these models can address practical pain points in real-world data-centric applications, while grounding solutions in core database principles like efficiency, effectiveness, and scalability. For new Ph.D. students, a common pitfall is either chasing LLM trends without a clear research perspective or overlooking the transformative potential of these models. A solid foundation in data management can guide responsible and innovative applications of LLMs that effectively bridge AI advances with real-world database problems.

Xiaofang Zhou. I believe that embracing new trends is essential, and we should view AI as a valuable opportunity to drive disruptive innovation in the database field, enabling novel approaches and advancements. The scope of database research is continuously evolving, and the impact of a research effort largely depends on whether it addresses a genuine, real-world problem. This underscores the importance for students to have strong hands-on experience with real systems and applications, allowing them to identify new challenges and develop innovative solutions. However, one concern I have regarding the training of future Ph.D. students is that AI tools may lead to a lack of patience among young researchers. Since AI can quickly generate seemingly plausible answers, it may discourage deep critical thinking–an essential skill for any Ph.D. candidate. This could ultimately hinder the development of rigorous analytical and problem-solving abilities.

Q7. *What industry requirements are projected for this area over the next 3-5 years? Which technical skills will organizations prioritize when hiring new graduates?*

Yuyu Luo. In the next 3-5 years, industry demand will grow heavily for professionals capable of bridging data management and artificial intelligence (DATA+AI). Companies will increasingly seek experts proficient in deploying and maintaining LLMs within production environments. Expertise in vector databases and efficient similarity search techniques will be particularly valuable, as organizations aim to construct and optimize data infrastructures (e.g., data retrieval pipelines) for LLM-driven applications. Due to the necessity of scaling these systems, experience in AI/Cloud infrastructure and performance optimization–especially involving GPUs and specialized hardware–will also be highly sought after.

Wenjie Zhang. From my own experience collaborating with industry partners in Australia, one of the most critical requirements is the development of trustworthy and reliable LLM-powered data solutions. Industries

are increasingly interested in applying LLMs to enhance data-driven decision-making, but they are equally concerned about issues such as data privacy, security, model hallucination, explainability, accountability, and regulatory compliance. Besides strong programming and system skills, technical capabilities in areas like LLM fine-tuning, prompt engineering, RAG, data governance, and evaluation of model trustworthiness will be in high demand. Graduates who can bridge AI techniques with rigorous data management practices will be particularly valuable in addressing real-world challenges.

Minqi Zhou. Over the next 3-5 years, enterprise and consumer agents will proliferate, with enterprise agents autonomously executing or augmenting worker tasks by interpreting organizational processes via integrated data pipelines, and consumer agents personalizing experiences through unified, cross-device (e.g., phone, Pad, and PC) data aggregation. How to integrate all these unstructured, isolated data to understand the enterprises and the users is an important direction for the graduates.

Xiaofang Zhou. I believe that over the next 3-5 years, the scale and variety of data we handle will be significantly different from what we see today. With the rapid growth of AI applications, there will be an explosion of high-dimensional data that needs to be generated, processed, and managed. To handle this shift, new systems and tools will need to be developed. Additionally, the increasing prevalence of multimodal and unstructured data will drive the need for innovative storage and indexing strategies. As a result, organizations will seek graduates who not only understand traditional DB systems but are also well-versed in emerging trends like AI-powered DBs and multimodal data processing. Candidates who demonstrate adaptability, cross-disciplinary knowledge, and hands-on experience with AI+DB and system development will stand out in this evolving landscape.

Q8. *What strategies can curb big tech monopolies in the LLM era, and how might advanced database technologies help democratize AI?*

Yuyu Luo. Open-source AI models like LLaMA [63, 62] provide a practical countermeasure against monopolistic control, especially when combined with advanced data management tools that enable seamless integration with high-quality local datasets. RAG exemplifies how such integration boosts utility. Extending crowdsourced data management to LLM-agent systems, where multiple agents collaborate, raises classic cost-quality tradeoffs–an area well-aligned with database researchers' expertise in cost-based optimization. Data-centric AI [37, 13, 39, 12], which focuses on enhancing data quality to drive model performance, offers another fertile ground for database research [15, 14, 11].

Wenjie Zhang. From a technical perspective, advanced database technologies can play a key role in democratizing AI by enabling more efficient, cost-effective, and domain-specific model deployment outside large cloud-based ecosystems. Techniques such as RAG allows smaller organizations to build powerful AI systems without training or hosting large foundation models. Open-source LLMs combined with high-performance data infrastructure can empower researchers, startups, small/medium enterprises, and public institutions to develop tailored solutions using their local data and tasks.

Minqi Zhuo. In agentic AI, deployed agents hinge on three core capabilities: leveraging LLM infrastructures for world understanding, invoking diverse tools via the Model Context Protocol (MCP), and employing database-driven techniques to integrate isolated enterprise or consumer data sources with atomicity, consistency, and isolation–underscoring the critical importance of rigorous data preparation and high-quality integration for democratizing AI.

Xiaofang Zhou. On the application side, open-source LLMs such as DeepSeek and Qwen are increasingly capable of competing with closed-source alternatives. In many sensitive and high-privacy scenarios, there is a strong demand for locally deployed LLMs. In such cases, database solutions supporting these models can be more distributed and flexible compared to monolithic, super-scale databases, better aligning with the needs of privacy-preserving and decentralized AI deployments.

3. CONCLUDING REMARKS

The panel concluded by discussing the other challenges for real-world LLM-RAG deployment: consistency, robustness, privacy, security, and human-in-the-loop – noting that semantically equivalent queries often produce divergent LLM outputs, a problem magnified by varying retrieval orders in iterative or multimodal RAG workflows. Achieving robustness requires end-to-end guarantees for query execution and formal validation of retrieval completeness and soundness across the pipeline. Addressing privacy and security calls for advanced privacy-preserving methods and fine-grained access controls over data, embeddings, model parameters, and query logs. Ensuring trust and transparency demands tight human-DB interfaces and provenance-based explanations citing source documents. We hope that this discussion and open challenges will inspire future works on the domain's emerging data management issues.

Some of the panelists and the moderator of this panel (1) will co-organize and participate in a future Dagstuhl seminar on "Managing Vector Data for Retrieval Augmented Generation: Systems and Algorithms" [9]; and (2) will also co-organize the second edition of the LLM + Vector Data workshop at ICDE 2026 [5] for initiating more discussion and interdisciplinary collaboration.

4. REFERENCES

[1] DIAL@HKUST(GZ), DeepFund Project, https://github.com/HKUSTDial/DeepFund

[2] DIAL@HKUST(GZ), https://github.com/HKUSTDial

[3] DIAL@HKUST(GZ), OpenManus Project, https://github.com/FoundationAgents/OpenManus

[4] DIAL@HKUST(GZ), Text-to-SQL Project, https://github.com/HKUSTDial/NL2SQL_Handbook

[5] LLM + Vector Data @ ICDE 2026, Second International Workshop on Coupling of Large Language Models with Vector Data Management: Agentic RAG Edition, https://llmvdb2.github.io/

[6] Retrieval augmented generation (rag) in azure ai search. https://learn.microsoft.com/en-us/azure/search/retrieval-augmented-generation-overview?tabs=docs, 2025.

[7] S. Amer-Yahia, J. Bogojeska, R. Facchinetti, V. Franceschi, A. Gionis, K. Hose, G. Koutrika, R. Kouyos, M. Lissandrini, S. Maniu, K. Mirylenka, D. Mottin, T. Palpanas, M. Rigotti, and Y. Velegrakis. Towards reliable conversational data analytics. In *International Conference on Extending Database Technology (EDBT)*, pages 962–969, 2025.

[8] S. Amer-Yahia, A. Bonifati, L. Chen, G. Li, K. Shim, J. Xu, and X. Yang. From large language models to databases and back: A discussion on research and education. *SIGMOD Rec.*, 52(3):49–56, 2023.

[9] S. Amer-Yahia, A. Khan, W. Lehner, S. Mehrotra, and W. Zhang. Managing vector data for retrieval augmented generation: Systems and algorithms (Dagstuhl Seminar 26161). https://www.dagstuhl.de/en/seminars/seminar-calendar/seminar-details/26161, 2026.

[10] A. Biswal, L. Patel, S. Jha, A. Kamsetty, S. Liu, J. E. Gonzalez, C. Guestrin, and M. Zaharia. Text2SQL is not enough: Unifying AI and databases with TAG. *CoRR*, abs/2408.14717, 2024.

[11] C. Chai, K. Jin, N. Tang, J. Fan, D. Miao, J. Wang, Y. Luo, G. Li, Y. Yuan, and G. Wang. Cost-effective missing value imputation for data-effective machine learning. *ACM Trans. Database Syst.*, 50(3), May 2025.

[12] C. Chai, J. Liu, N. Tang, J. Fan, D. Miao, J. Wang, Y. Luo, and G. Li. Goodcore: Data-effective and data-efficient machine learning through coreset selection over incomplete data. *Proc. ACM Manag. Data*, 1(2):157:1–157:27, 2023.

[13] C. Chai, J. Liu, N. Tang, G. Li, and Y. Luo. Selective data acquisition in the wild for model charging. *Proc. VLDB Endow.*, 15(7):1466–1478, 2022.

[14] C. Chai, N. Tang, J. Fan, and Y. Luo. Demystifying artificial intelligence for data preparation. In *SIGMOD Conference Companion*, pages 13–20. ACM, 2023.

[15] C. Chai, J. Wang, Y. Luo, Z. Niu, and G. Li. Data management for machine learning: A survey. *IEEE Trans. Knowl. Data Eng.*, 35(5):4646–4667, 2023.

[16] B. J. Chan, C. Chen, J. Cheng, and H. Huang. Don't do RAG: when cache-augmented generation is all you need for knowledge tasks. In *ACM Web Conference (WWW)*, pages 893–897, 2025.

[17] S. Chen, J. Fan, B. Wu, N. Tang, C. Deng, P. Wang, Y. Li, J. Tan, F. Li, J. Zhou, and X. Du. Automatic database configuration debugging using retrieval-augmented language models. *Proc. ACM Manag. Data*, 3(1):13:1–13:27, 2025.

[18] P. Christmann and G. Weikum. RAG-based question answering over heterogeneous data and text. *IEEE Data Eng. Bull.*, 48(4):71–86, 2024.

[19] S. Das, R. Saboo, C. S. K. Vadrevu, B. Wang, and S. Xu. Applications of llms in e-commerce search and product knowledge graph: The doordash case study. In *ACM International Conference on Web Search and Data Mining (WSDM)*, pages 1163–1164, 2024.

[20] X. L. Dong, S. Moon, Y. E. Xu, K. Malik, and Z. Yu. Towards next-generation intelligent assistants leveraging LLM techniques. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 5792–5793, 2023.

[21] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130, 2024.

[22] R. C. Fernandez, A. J. Elmore, M. J. Franklin, S. Krishnan, and C. Tan. How large language models will disrupt data management. *Proc. VLDB Endow.*, 16(11):3302–3309, 2023.

[23] M. Grohe. word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 1–16, 2020.

[24] A. Y. Halevy, Y. Choi, A. Floratou, M. J. Franklin, N. F. Noy, and H. Wang. Will llms reshape, supercharge, or kill data science? *Proc. VLDB*

Endow., 16(12):4114–4115, 2023.

[25] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. 43(2), 2025.

[26] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Annual ACM Symposium on the Theory of Computing (STOC)*, pages 604–613, 1998.

[27] A. Khan, Y. Wang, W. Zhang, Y. Tian, and M. T. Özsu. LLM + vector data: Coupling of large language models with vector data management for enhancing data science. In *Workshops at the International Conference on Data Engineering (ICDEW)*, 2025.

[28] L. Kilpatrick, S. B. Mallick, and R. Kofman. Gemini 1.5 Pro 2M context window, code execution capabilities, and Gemma 2 are available today. `https://developers.googleblog.com/en/new-features-for-the-gemini-api-and-google-ai-studio/`, 2024.

[29] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[30] B. Li, Y. Luo, C. Chai, G. Li, and N. Tang. The dawn of natural language to SQL: are we fully ready? [experiment, analysis & benchmark]. *Proc. VLDB Endow.*, 17(11):3318–3331, 2024.

[31] B. Li, J. Zhang, J. Fan, Y. Xu, C. Chen, N. Tang, and Y. Luo. Alpha-sql: Zero-shot text-to-sql using monte carlo tree search. In *Forty-Second International Conference on Machine Learning, ICML 2025, Vancouver, Canada, July 13-19, 2025*. OpenReview.net, 2025.

[32] C. Li, Y. Shi, Y. Luo, and N. Tang. Deepfund: Will LLM be professional at fund investment? A live arena perspective. *CoRR*, abs/2503.18313, 2025.

[33] G. Li, X. Zhou, and X. Zhao. LLM for data management. *Proc. VLDB Endow.*, 17(12):4213–4216, 2024.

[34] H. Li, Y. Li, A. Tian, T. Tang, Z. Xu, X. Chen, N. Hu, W. Dong, Q. Li, and L. Chen. A survey on large language model acceleration based on kv cache management, 2025.

[35] L. Liang, Z. Bo, Z. Gui, Z. Zhu, L. Zhong, P. Zhao, M. Sun, Z. Zhang, J. Zhou, W. Chen, W. Zhang, and H. Chen. KAG: boosting llms in professional domains via knowledge augmented generation. In *ACM Web Conference (WWW)*, pages 334–343, 2025.

[36] J. Lin, P. Gupta, W. Horn, and G. Mishne. Musings about the future of search: A return to the past? *CoRR*, abs/2412.18956, 2024.

[37] X. Lin, Y. Qi, Y. Zhu, T. Palpanas, C. Chai, N. Tang, and Y. Luo. LEAD: iterative data selection for efficient LLM instruction tuning. *CoRR*, abs/2505.07437, 2025.

[38] B. Liu, X. Li, J. Zhang, J. Wang, T. He, S. Hong, H. Liu, S. Zhang, K. Song, K. Zhu, Y. Cheng, S. Wang, X. Wang, Y. Luo, H. Jin, P. Zhang, O. Liu, J. Chen, H. Zhang, Z. Yu, H. Shi, B. Li, D. Wu, F. Teng, X. Jia, J. Xu, J. Xiang, Y. Lin, T. Liu, T. Liu, Y. Su, H. Sun, G. Berseth, J. Nie, I. Foster, L. T. Ward, Q. Wu, Y. Gu, M. Zhuge, X. Tang, H. Wang, J. You, C. Wang, J. Pei, Q. Yang, X. Qi, and C. Wu. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *CoRR*, abs/2504.01990, 2025.

[39] J. Liu, C. Chai, Y. Luo, Y. Lou, J. Feng, and N. Tang. Feature augmentation with reinforcement learning. In *ICDE*, pages 3360–3372. IEEE, 2022.

[40] X. Liu, S. Shen, B. Li, P. Ma, R. Jiang, Y. Zhang, J. Fan, G. Li, N. Tang, and Y. Luo. A survey of text-to-sql in the era of llms: Where are we, and where are we going?, 2025.

[41] X. Liu, S. Shen, B. Li, N. Tang, and Y. Luo. Nl2sql-bugs: A benchmark for detecting semantic errors in NL2SQL translation. *CoRR*, abs/2503.11984, 2025.

[42] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. In *ICDE*, pages 101–112. IEEE Computer Society, 2018.

[43] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin. Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks. In *SIGMOD Conference*, pages 1235–1247. ACM, 2021.

[44] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin. Natural language to visualization by neural machine translation. *IEEE Trans. Vis. Comput. Graph.*, 28(1):217–226, 2022.

[45] A. Obeng, J. Zhong, and C. Gu. How we built Text-to-SQL at Pinterest. `https://medium.com/pinterest-engineering/how-we-built-text-to-sql-at-pinterest-30bad30dabff`, 2024.

[46] J. J. Pan, J. Wang, and G. Li. Survey of vector database management systems. *VLDB J.*, 33(5):1591–1615, 2024.

[47] L. Patel, S. Jha, C. Guestrin, and M. Zaharia.

LOTUS: enabling semantic queries with llms over tables of unstructured and structured data. *CoRR*, abs/2407.11418, 2024.

[48] F. Petroni, F. Siciliano, F. Silvestri, and G. Trappolini. Report on the 1st workshop on information retrieval's role in rag systems (ir-rag 2024) at sigir 2024. *SIGIR Forum*, 58(2):1–12, 2025.

[49] X. Qin, Y. Luo, N. Tang, and G. Li. Making data visualization more efficient and effective: a survey. *VLDB J.*, 29(1):93–117, 2020.

[50] L. Shen, H. Li, Y. Wang, T. Luo, Y. Luo, and H. Qu. Data playwright: Authoring data videos with annotated narration. *CoRR*, abs/2410.03093, 2024.

[51] J. Shim, G. Seo, C. Lim, and Y. Jo. Tooldial: Multi-turn dialogue generation method for tool-augmented language models. In *International Conference on Learning Representations (ICLR)*, 2025.

[52] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei. Agentic retrieval-augmented generation: A survey on agentic RAG. *CoRR*, abs/2501.09136, 2025.

[53] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972), 2023.

[54] X. TAN, X. WANG, Q. LIU, X. XU, X. YUAN, and W. ZHANG. Paths-over-graph: Knowledge graph empowered large language model reasoning. In *PROCEEDINGS OF THE ACM ON WEB CONFERENCE 2025*, 2025.

[55] X. TAN, X. WANG, Q. LIU, X. XU, X. YUAN, L. ZHU, and W. ZHANG. Hydra: Structured cross-source enhanced large language model reasoning. *ARXIV PREPRINT ARXIV:2505.17464*, 2025.

[56] N. Tang, C. Yang, J. Fan, L. Cao, Y. Luo, and A. Y. Halevy. Verifai: Verified generative AI. In *CIDR*. www.cidrdb.org, 2024.

[57] N. Tang, C. Yang, Z. Zhang, Y. Luo, J. Fan, L. Cao, S. Madden, and A. Y. Halevy. Symphony: Towards trustworthy question answering and verification using RAG over multimodal data lakes. *IEEE Data Eng. Bull.*, 48(4):135–146, 2024.

[58] Y. Tian, Z. Yue, R. Zhang, X. Zhao, B. Zheng, and X. Zhou. Approximate nearest neighbor search in high dimensional vector databases: Current research and future directions. *IEEE Data Eng. Bull.*, 47(3):39–54, 2023.

[59] J. Wang, G. Li, and J. Feng. idatalake: An llm-powered analytics system on data lakes. *IEEE Data Eng. Bull.*, 49(1):57–69, 2025.

[60] N. Wang, H. Yang, and C. D. Wang. FinGPT: Instruction tuning benchmark for open-source large language models in financial datasets. *NeurIPS Workshop on Instruction Tuning and Instruction Following*, 2023.

[61] B. Wu, J. Shi, Y. Wu, N. Tang, and Y. Luo. Transxssm: A hybrid transformer state space model with unified rotary position embedding. *CoRR*, abs/2506.09507, 2025.

[62] B. Wu, J. Shi, Y. Wu, N. Tang, and Y. Luo. Transxssm: A hybrid transformer state space model with unified rotary position embedding, 2025.

[63] Y. Wu, J. Shi, B. Wu, J. Zhang, X. Lin, N. Tang, and Y. Luo. Concise reasoning, big gains: Pruning long reasoning trace with difficulty-aware prompting, 2025.

[64] P. Xia, K. Zhu, H. Li, H. Zhu, Y. Li, G. Li, L. Zhang, and H. Yao. RULE: Reliable multimodal RAG for factuality in medical vision language models. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1081–1093, 2024.

[65] C. Xiao, X. Hu, Z. Liu, C. Tu, and M. Sun. Lawformer: A pre-trained language model for Chinese legal long documents. *AI Open*, 2:79–84, 2021.

[66] Y. Xie, Y. Luo, G. Li, and N. Tang. Haichart: Human and AI paired visualization system. *Proc. VLDB Endow.*, 17(11):3178–3191, 2024.

[67] Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, and Z. Li. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2905–2909, 2024.

[68] K. Yang, Y. Tian, N. Peng, and D. Klein. Re3: Generating longer stories with recursive reprompting and revision. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4393–4479, 2022.

[69] Y. Ye, J. Hao, Y. Hou, Z. Wang, S. Xiao, Y. Luo, and W. Zeng. Generative AI for visualization: State of the art and future directions. *Vis. Informatics*, 8(1):43–66, 2024.

[70] Z. Yue, H. Zhuang, A. Bai, K. Hui, R. Jagerman, H. Zeng, Z. Qin, D. Wang, X. Wang, and M. Bendersky. Inference scaling for long-context retrieval augmented generation. In *International Conference on Learning Representations (ICLR)*, 2025.

[71] J. Zhang, J. Xiang, Z. Yu, F. Teng, X. Chen, J. Chen, M. Zhuge, X. Cheng, S. Hong, J. Wang, B. Zheng, B. Liu, Y. Luo, and C. Wu. Aflow: Automating agentic workflow generation. In *ICLR*. OpenReview.net, 2025.

[72] Z. Zhang, C. Chen, B. Liu, C. Liao, Z. Gong, H. Yu, J. Li, and R. Wang. Unifying the perspectives of NLP and software engineering: A survey on language models for code. *Transactions on Machine Learning Research*, 2024.

[73] Z. Zhang, Z. Liang, Y. Wu, T. Lin, Y. Luo, and N. Tang. Datamosaic: Explainable and verifiable multi-modal data analytics through extract-reason-verify. *CoRR*, abs/2504.10036, 2025.

[74] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, and B. Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.

[75] X. Zhao, Y. Tian, K. Huang, B. Zheng, and X. Zhou. Towards efficient index construction and approximate nearest neighbor search in high-dimensional spaces. In *PVLDB*, 2023.

[76] T. Zheng, Z. Deng, H. T. Tsang, W. Wang, J. Bai, Z. Wang, and Y. Song. From automation to autonomy: A survey on large language models in scientific discovery. *arXiv preprint arXiv:2505.13259*, 2025.

[77] Y. Zhu, S. Du, B. Li, Y. Luo, and N. Tang. Are large language models good statisticians? In *NeurIPS*, 2024.

[78] Y. Zhu, R. Jiang, B. Li, N. Tang, and Y. Luo. Elliesql: Cost-efficient text-to-sql with complexity-aware routing. *COLM*, 2025.