SIGMOD Officers, Committees, and Awardees

Chair
Juliana Freire
Computer Science & Engineering
New York University
Brooklyn, New York
USA
+1 646 997 4128
juliana.freire <at> nyu.edu

Vice-Chair
Ihab Francis Ilyas
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario
CANADA
+1 519 888 4567 ext. 33145
ilyas <at> uwaterloo.ca

Secretary/Treasurer
Fatma Ozcan
IBM Research
Almaden Research Center
San Jose, California
USA
+1 408 927 2737
fozcan <at> us.ibm.com

SIGMOD Executive Committee:
Juliana Freire (Chair), Ihab Francis Ilyas (Vice-Chair), Fatma Ozcan (Treasurer), K. Selçuk Candan, Rada Chirkova, Chris Jermaine, Wang-Chiew Tan, AnHai Doan, Leonid Libkin, and Curtis Dyreson

Advisory Board:
Yannis Ioannidis (Chair), Phil Bernstein, Surajit Chaudhuri, Rakesh Agrawal, Joe Hellerstein, Mike Franklin, Laura Haas, Renee Miller, John Wilkes, Chris Olsten, AnHai Doan, Tamer Özsu, Gerhard Weikum, Stefano Ceri, Beng Chin Ooi, Timos Sellis, Sunita Sarawagi, Stratos Idreos, and Tim Kraska

SIGMOD Information Director:
Curtis Dyreson, Utah State University

Associate Information Directors:
Huiping Cao, Georgia Koutrika, Wim Martens, and Sourav S Bhowmick

SIGMOD Record Editor-in-Chief:
Rada Chirkova, NC State University

SIGMOD Record Associate Editors:
Azza Abouzied, Lyublena Antova, Vanessa Braganholo, Aaron J. Elmore, Wim Martens, Kyriakos Mouratidis, Dan Olteanu, Divesh Srivastava, Pınar Tözün, Immanuel Trummer, Yannis Velegrakis, Marianne Winslett, and Jun Yang

SIGMOD Conference Coordinator:
K. Selçuk Candan, Arizona State University

PODS Executive Committee:
Dan Suciu (Chair), Tova Milo, Diego Calvanese, Wang-Chiew Tan, Rick Hull, and Floris Geerts

Sister Society Liaisons:
Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment), Christian Jensen (IEEE TKDE)

SIGMOD Awards Committee:
M. Tamer Özsu (Chair), Stefano Ceri, Yanlei Diao, Volker Markl, Renee Miller, and Sunita Sarawagi

Jim Gray Doctoral Dissertation Award Committee:
Pınar Tözün (co-Chair), Viktor Leis (co-Chair), Peter Bailis, Alexandra Meliou, Bailu Ding, Vanessa Braganholo, Immanuel Trummer, and Joy Arulraj

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)
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)
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)

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)

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)
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)
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)

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

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

[Last updated: June 30, 2020]

Editor's Notes

Welcome to the June 2020 issue of the ACM SIGMOD Record!

This issue starts with two articles in the Database Principles column. The first article, by Barceló, Kostylev, Monet, Pérez, Reutter, and Silva, surveys recent results concerning architectures of graph neural networks (GNNs) in terms of their ability to classify nodes over graphs. GNNs have recently been proven to be very efficient in many applications, but their theoretical properties are not yet well understood. The work described in the article contributes to better understanding of GNNs, in particular of their power to express node classifiers in graphs. The formal results outlined in the article connect the expressive power of GNNs to unary logical formulas, thus bridging the gap between structure-aware machine-learning architectures, on the one hand, and classic database-query formalisms, on the other hand. The authors also report on experimental corroborations of their results, with the code available online, and discuss open problems and future work in the area.

The second article in the Database Principles column, by Schwentick, Vortmeier, and Zeume, focuses on the problem of dynamic query maintenance, that is of whether query answers can be maintained in response to changes in the database data, by using first-order update rules and potentially auxiliary data. The authors study the problem from the perspective of dynamic complexity theory, and present results centering on the reachability (transitive-closure) query in graphs. The exposition starts from the simplest case and then progresses in a clear sequence of steps each building on previous steps. The article outlines useful techniques, as well as positive and impossibility results, and also discusses implications for regular-path and other types of queries. The authors propose guidelines for determining whether a given query can be dynamically maintained using first-order update rules. The article also presents open problems and provides a discussion of related and further work.

The Vision column presents an article by Amer-Yahia and colleagues on ways to make AI machines work in Future of Work (FoW) scenarios. AI systems are increasingly used for the benefit of humans, and this article focuses specifically on using AI systems to enable human work in both physical and virtual workplaces. Bringing humans to the frontier of FoW would contribute to increasing their trust in AI systems. In the process, human perception could shift to using such systems as a source of self-improvement and better work performance, thus positively shaping national and societal outputs. To make this happen, FoW platforms need to be redesigned, and human workers should be encouraged to take on more supervisory roles, which would allow them to provide corrective feedback to AI systems. The article outlines intellectual challenges that need to be addressed to achieve this vision, including the imperative to capture human capabilities, as well as declarative specification of job-related and workforce-related requirements. The authors also map the intellectual challenges to data-management areas, and use this perspective to review related work.

The Surveys column features an article by Jandre, Diirr, and Braganholo that studies the types of provenance solutions that are available in software tools designed to enable collaborative in-silico research. Collaboration is essential in science, and the emergence of accessible computers and computer networks over the past decades has allowed long-distance collaboration. In fact, it has also increased the number of scientific experiments conducted in silico. The various data and metadata that are collected about objects and activities encountered during in-silico experiments logically belong in provenance databases. The article formulates two main provenance-related challenges in collaborative in-silico experiments and the two corresponding research questions, and provides a taxonomy for, as well as an extended comparison of, state-of-the-art approaches and

provenance-aware models that are available for conducting such experiments. The authors provide literature-based answers to the two research questions, and discuss further challenges and opportunities based on the gaps identified in the survey. The findings presented in the article generate insights that may be useful for researchers interested in the area.

The Distinguished Profiles column features Susan Davidson, professor at the University of Pennsylvania. Sue is an ACM Fellow, a Corresponding Fellow of the Royal Society of Edinburgh, and the recipient of the 2017 IEEE Technical Committee on Data Engineering Impact Award. Her Ph.D. is from Princeton University. In this interview, Sue talks about her research on data provenance and its connections to the problems of data citation and of fake news. She shares how she got interested in bioinformatics and computing, and outlines ideas for doing research that helps domain science or industry. Sue also provides insights on a range of other topics, including what CRA accomplishments she is most proud of, as well as strategies for engaging more women in computer science, including promotion of undergraduate research and sense of community. She talks about balancing work and family, discusses what she would do if she magically had extra time, and gives advice for fledgling and mid-career database researchers.

This issue features a report by Kondylakis, Stefanidis, Rao, and Parry on the outcomes of the Second International Workshop on Semantic Web Meets Health Data Management (SWH 2019). The workshop took place in Auckland, New Zealand in conjunction with the 18th International Semantic Web Conference (ISWC 2019). The SWH workshop aimed to bring together an interdisciplinary audience, to discuss challenges in healthcare data management and to propose novel and practical solutions for next-generation data-driven healthcare systems. The article summarizes the outcomes of the workshop and outlines key observations and emerging research directions.

The issue closes with a SIGMOD Executive Committee statement on racism, and a second-round call for SIGMOD 2021 research papers.

On behalf of the SIGMOD Record Editorial board, I hope that you enjoy reading the June 2020 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

June 2020

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)

The Expressive Power of Graph Neural Networks as a Query Language

Pablo Barceló[*]
IMC, PUC & IMFD Chile

Egor V. Kostylev
University of Oxford

Mikaël Monet
IMFD Chile

Jorge Pérez[†]
DCC, UChile & IMFD Chile

Juan L. Reutter[‡]
DCC, PUC & IMFD Chile

Juan-Pablo Silva
DCC, UChile

ABSTRACT

In this paper we survey our recent results characterizing various graph neural network (GNN) architectures in terms of their ability to *classify* nodes over graphs, for classifiers based on unary logical formulas– or queries. We focus on the language FOC_2, a well-studied fragment of FO. This choice is motivated by the fact that FOC_2 is related to the Weisfeiler-Lehman (WL) test for checking graph isomorphism, which has the same ability as GNNs for distinguishing nodes on graphs. We unveil the exact relationship between FOC_2 and GNNs in terms of node classification. To tackle this problem, we start by studying a popular basic class of GNNs, which we call AC-GNNs, in which the features of each node in a graph are updated, in successive layers, according only to the features of its neighbors. We prove that the unary FOC_2 formulas that can be captured by an AC-GNN are exactly those that can be expressed in its guarded fragment, which in turn corresponds to graded modal logic. This result implies in particular that AC-GNNs are too weak to capture all FOC_2 formulas. We then seek for what needs to be added to AC-GNNs for capturing all FOC_2. We show that it suffices to add readouts layers, which allow updating the node features not only in terms of its neighbors, but also in terms of a global attribute vector. We call GNNs with readouts ACR-GNNs. We also describe experiments that validate our findings by showing that, on synthetic data conforming to FOC_2 but not to graded modal logic, AC-GNNs struggle to fit in while ACR-GNNs can generalise even to graphs of sizes not seen during training.

[*]Institute for Mathematical and Computational Engineering, School of Engineering, Faculty of Mathematics, Pontificia Universidad Catolica de Chile.
[†]Department of Computer Science, University of Chile.
[‡]Department of Computer Science, School of Engineering, Pontificia Universidad Catolica de Chile.

1. INTRODUCTION

Graph neural networks (*GNNs*), which were introduced about a decade ago [21, 29], are a class of artificial neural network architectures that has recently become popular for a wide range of applications dealing with structured data, such as molecule classification, knowledge graph completion, and Web page ranking [6, 13, 17, 30]. The main idea behind GNNs is that the connections between neurons are not arbitrary but reflect the structure of the input data, which is given as a graph. Specifically, each node in the graph is associated a neuron, and the forward propagation of the neuron's data depends on the connections–or neighbors–of this neuron in the graph. This approach is motivated by convolutional and recurrent neural networks, and actually generalizes both of them [6].

Despite the fact that GNNs have recently been proven very efficient in many applications, their theoretical properties are not yet well-understood. We focus on the *expressive power* of GNNs, and concentrate on the ability of GNNs to express *node classifiers*, that is, functions assigning 1 (true) or 0 (false) to every node in a graph. More precisely, let us assume we have a GNN whose last layer behaves like a classifier: for every node v of the graph the last layer simply outputs a number 0 or 1. We say that this GNN can express a particular node classifier f, if for every graph G we have that the computation of the GNN assigns to every node v in G the value $f(v)$. This leads us to the following question:

> What type of node classifiers can be expressed as GNNs?

In the context of databases, one can see a graph as a *graph database* [27, 5], and a classifier f as a query language: On input graph (database) G, the

query would return all the nodes in G that are classified as true by f. Thus, answering the question above implies understanding what type of queries can be expressed by GNNs.

Our first observation draws from an interesting result published independently by Morris et al. [23] and Xu et al. [34] that establishes a connection between GNNs and the *Weisfeiler-Lehman* (*WL*) test for checking graph isomorphism. The WL test works by constructing a labeling of the nodes of the graph, in an incremental fashion, and then decides whether two graphs are isomorphic by comparing the labeling of each graph. To state the connection between GNNs and this test, consider the popular GNN architecture that updates the feature vector of each graph node by combining it with the (aggregation of) the feature vectors of its neighbors. We call such GNNs *aggregate-combine GNNs*, or *AC-GNNs*. The authors of these papers independently observe that the node labeling produced by the WL test always refines the labeling produced by any GNN. More precisely, if two nodes are labeled the same by the algorithm underlying the WL test, then the feature vectors of these nodes produced by any AC-GNN will always be the same. Moreover, there are AC-GNNs that can reproduce the WL labeling, and hence AC-GNNs can be as powerful as the WL test for distinguishing nodes.

In terms of queries, these connections give us a sort of upper bound: we see that AC-GNNs can only express queries that agree with the WL test, in the sense that all nodes assigned the same WL label are either all part of the answer, or none of them is. However, this gives us little in terms of understanding the actual queries that can be expressed by GNNs.

To pursue further in this topic, we concentrate on queries expressible in first-order logic. For AC-GNNs, a meaningful starting point to measure their expressive power is the logic FOC_2, the two-variable fragment of FO extended with counting quantifiers of the form $\exists^{\geq N} x\, \varphi(x)$, which state that there are at least N nodes satisfying formula φ [7].[1] This choice of FOC_2 is justified by a classical result establishing a tight connection between FOC_2 and WL: two nodes in a graph are classified the same by the WL test if and only if they satisfy exactly the same unary FOC_2 formulas [7].

Given the connection between AC-GNNs and WL on the one hand, and that between WL and FOC_2 on the other hand, it is natural to think that the expressivity of AC-GNNs coincides with that of FOC_2, at least in terms of classifiers or unary queries. Surprisingly, this is not the case; indeed, we will see that there are many FOC_2 unary formulas that cannot be expressed by AC-GNNs. This leaves us with the following natural questions. First, what is the largest fragment of FOC_2 that can be captured by AC-GNNs? Second, is there an extension of AC-GNNs that allows to express all FOC_2 (unary) formulas? In this paper we provide answers to these two questions. The following are the main results outlined in this paper.

First, we characterize exactly the fragment of FOC_2 that can be expressed as AC-GNNs. This fragment corresponds to *graded modal logic* [9], or, equivalently, to the *description logic* $\mathcal{ALCQ}$, which has received considerable attention in the knowledge representation community [2, 3]. What is more, we show that formulas of this kind can be expressed in terms of a particularly simple class of GNNs, which we call *homogeneous* AC-GNNs. We present these results in Section 4.

Second, we extend the AC-GNN architecture in a simple way by allowing global *readouts*, where in each layer we also compute a feature vector for the whole graph and combine it with local aggregations; we call these *aggregate-combine-readout GNNs*, or ACR-GNNs. These networks are a special case of the networks proposed by Battaglia et al. [6] for relational reasoning over graph representations. In this setting, we prove that each FOC_2 formula can be captured by an ACR-GNN. In this setting, we also prove that each FOC_2 formula can be captured by an ACR-GNN using a single readout. These results are presented in Section 5.

Finally, we experimentally validate our findings in Section 6, where we show that the theoretical expressiveness of ACR-GNNs, as well as the differences between AC-GNNs and ACR-GNNs, can be observed when we learn from examples. In particular, we show that on synthetic graph data conforming to FOC_2 formulas, AC-GNNs struggle to fit the training data while ACR-GNNs can generalize even to graphs of sizes not seen during training.

Remark. This paper summarizes recent results published by the same authors in a machine learning conference paper [4]; however, the presentation is adapted to a reader in the database community.

2. PRELIMINARIES

In this section we describe the architecture of basic GNN classifiers, AC-GNNs, and introduce other related notions. We consider the problem of Boolean

[1] Note that every formula in FOC_2 can also be expressed in FO, albeit with more than just two variables.

node classification in graphs, where we wish to classify each graph node as true or false on the base of the structure of its neighborhood. We concentrate on undirected graphs without self-loops and multiedges and where each node is assigned with a unique color from a finite set; however, our results can be generalised to directed edge-colored multigraphs with loops in a straightforward way.

Graph neural networks. The basic architecture for GNNs, and the one studied in recent articles on GNN expressibility [23, 34], consists of a sequence of *layers* that combine the feature vectors of every node with the multiset of feature vectors of its neighbors, as formalized in the following definition.

DEFINITION 2.1. *An* aggregate-combine GNN *(*AC-GNN*)* $\mathcal{A}$ *with* $L \geq 1$ *layers is specified by two sets of functions,* $\{\mathrm{AGG}^{(i)}\}_{i=1}^{L}$ *and* $\{\mathrm{COM}^{(i)}\}_{i=1}^{L}$*, called* aggregation *and* combination *functions, respectively, and a* classification *function* CLS*. Each aggregation function* $\mathrm{AGG}^{(i)}$ *takes a multiset of (rational) vectors and returns one such vector, each combination function* $\mathrm{COM}^{(i)}$ *takes a pair or vectors and returns one vector, and the classification function* CLS *takes a vector and returns a Boolean value,* true *or* false *(the vector dimensions of these functions are assumed to match the semantics of the GNN as defined next).*

An AC-GNN $\mathcal{A}$ takes a graph G as input and computes feature vectors $\boldsymbol{x}_v^{(i)}$, for each node v of G and each layer $i = 1, \ldots, L$, via the recursive formula

$$\boldsymbol{x}_v^{(i)} = \mathrm{COM}^{(i)}\Big(\boldsymbol{x}_v^{(i-1)}, \mathrm{AGG}^{(i)}\big(\{\!\!\{\boldsymbol{x}_u^{(i-1)} \mid u \in \mathcal{N}_G(v)\}\!\!\}\big)\Big),$$

where $\mathcal{N}_G(v)$ is the neighborhood of v in G and the initial vector $\boldsymbol{x}_v^{(0)}$ is the one-hot encoding of the color of v in G (i.e., the dimension of $\boldsymbol{x}_v^{(0)}$ is the number of possible colors and $\boldsymbol{x}_v^{(0)}$ has the k-th component 1 if its color has number k and 0 otherwise). Finally, each node v of G is classified as true or false according to CLS applied to $\boldsymbol{x}_v^{(L)}$. We define $\mathcal{A}(G, v) := \mathrm{CLS}(\boldsymbol{x}_v^{(L)})$, for each node v in G.

Aggregation, combination, classification functions. Many possible aggregation, combination, and classification functions exist, which produce different classes of GNNs [14, 17, 23, 34]. A simple, yet common choice is to consider the sum of feature vectors as the aggregation function, the sign of one of the elements in $\boldsymbol{x}_v^{(L)}$ as the classification function, and the combination function

$$\mathrm{COM}^{(i)}(\boldsymbol{x}_1, \boldsymbol{x}_2) = f\big(\boldsymbol{x}_1 \boldsymbol{C}^{(i)} + \boldsymbol{x}_2 \boldsymbol{A}^{(i)} + \boldsymbol{b}^{(i)}\big), \quad (1)$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are row vectors, $\boldsymbol{C}^{(i)}$ and $\boldsymbol{A}^{(i)}$ are matrices of parameters (of appropriate dimensions), $\boldsymbol{b}^{(i)}$ is a *bias* row vector of parameters, and f is a *non-linear* function, such as (truncated) ReLU or sigmoid [22]. We call *simple* an AC-GNN using these functions. Note that the parameters in $\boldsymbol{C}^{(i)}$, $\boldsymbol{A}^{(i)}$, and $\boldsymbol{b}^{(i)}$ are usually found during the training of the GNN (e.g., using standard ML techniques [22]). We say that an AC-GNN is *homogeneous* if all $\mathrm{AGG}^{(i)}$ are the same and all $\mathrm{COM}^{(i)}$ are the same (i.e., share the same parameters across layers). In most of our positive results we construct simple and homogeneous GNNs, while our negative results hold in general, i.e., for GNNs with arbitrary aggregation, combination, and classification functions and that are not necessarily homogeneous.

We note that besides node classification, which we consider in this paper, one can use GNNs to classify whole graphs. This can be done, for example, by considering that the classification function CLS inputs the multiset $\{\!\!\{\boldsymbol{x}_v^{(L)}\}\!\!\}$ of feature vectors over all nodes v in the graph and outputs a classification of the whole graph. In this case the classification function is often called *readout* [23, 34]. In this paper, however, we use the term "readout" to refer to functions applied globally on intermediate layers of ACR-GNNs (i.e., GNNs that are more expressive than AC-GNNs, see Section 5).

Weisfeiler-Lehman. The *Weisfeiler-Lehman* (*WL*) test (also called *node coloring*) is a powerful heuristic used to solve the graph isomorphism problem [7, 32], or, for our purposes, to determine whether the neighborhoods of two nodes in a graph are structurally close. Formally, the L-round WL algorithm takes as input a (node-colored) graph G and iteratively assigns, for L rounds, a new color to every node in the graph in such a way that the color of a node assigned in round i is uniquely and unambiguously defined by (i.e., has a one-to-one correspondence with) its own color in round $i-1$ and with the multiset of colors of its neighbors in G in round $i-1$. The result of the algorithm is the coloring of the nodes after round L; then, the multisets of the resulting colors in two graphs can be compared for testing their (non-)isomorphism [7, 32]. An important observation is that the rounds of the WL algorithm can be seen as the layers of an AC-GNN whose aggregation and combination functions are all injective [23, 34]. Furthermore, as independently

shown by Morris et al. [23] and Xu et al. [34], an AC-GNN classification can never contradict the WL test, in the following sense.

PROPOSITION 2.2. *If the L-round WL algorithm assigns the same color to two nodes in a graph, then every AC-GNN with L-layers classifies both nodes the same (i.e., either both as* true *or both as* false*).*

3. GNNS AND LOGIC

Our study relates the expressive power of GNNs to that of classifiers formalized as unary formulas in first order logic with equality (FO) and some of its fragments. It is well-known that FO logic underlies many standard database query languages, such as SQL, and thus our work bridges the gap between structure-aware machine learning architectures on the one side and classic declarative query formalisms on the other side.

Since we concentrate on undirected node-colored graphs, we consider the signature consisting of a single binary predicate Edge and unary predicates corresponding to the possible node colors, as well as assume that all the logical structures encode such graphs (in particular, the interpretation of Edge is always symmetric). As formalized in the following definition, we say that a GNN classifier (i.e., an AC-GNN or a GNN of more expressive architecture as described later) captures a logical classifier when both classifiers agree on every node in every graph.

DEFINITION 3.1. *A GNN classifier $\mathcal{A}$* captures *a logical formula $\varphi(x)$ if for every graph G and node v in G, it holds that $\mathcal{A}(G,v) = \mathsf{true}$ if and only if $(G,v) \models \varphi$.*

3.1 Logic FOC_2 and the WL test

As we have outlined in the introduction, we focus on formulas in FOC_2, the fragment of FO logic that only allows formulas with two variables, but in turn permits the use of *counting quantifiers* [7]. Such quantifiers have the form $\exists^{\geq N}$ for a positive integer N, and a formula $\exists^{\geq N} x\, \varphi(x)$ holds if there are *at least N different nodes* for which φ holds. For example, in FOC_2 we can express a formula that checks whether x is a red node, and there is another node that is not connected to x and that has at least two blue neighbors:

$$\gamma(x) \coloneqq \mathsf{Red}(x) \wedge \exists y\big(\neg\mathsf{Edge}(x,y) \wedge \exists^{\geq 2} x\big[\,\mathsf{Edge}(y,x) \wedge \mathsf{Blue}(x)\,\big]\big).$$

Despite that FOC_2 is not a syntactic fragment of FO logic due to the counting quantifiers, it is a semantic fragment, because these quantifiers can be expressed via usual existential quantifiers and disequalities. For example, the formula $\gamma(x)$ above can be written in FO as

$$\beta(x) \coloneqq \mathsf{Red}(x) \wedge \exists y\big(\neg\mathsf{Edge}(x,y) \wedge \exists z_1 \exists z_2\big[\,\mathsf{Edge}(y,z_1) \wedge \mathsf{Edge}(y,z_2) \wedge z_1 \neq z_2 \wedge \mathsf{Blue}(z_1) \wedge \mathsf{Blue}(z_2)\,\big]\big).$$

Note, however, that this rewriting is possible only by means of increasing the number of used variables, and it is easy to see that this formula cannot be expressed in FO_2, the fragment of FO that allows only two variables (and no counting quantifiers). On the other hand, FO is strictly more expressive than FOC_2; this is witnessed, for example, by a formula checking whether a graph has a triangle as a subgraph.

The following result, which is due to Cai et al. [7], establishes a classical connection between FOC_2 and the WL test. Together with Proposition 2.2, it provides a justification for our choice of the logic FOC_2 for measuring the expressiveness of AC-GNNs.

PROPOSITION 3.2. *For every graph G and nodes u, v in G, we have that u and v agree on every* FOC_2 *unary formula if and only if the WL algorithm colors v and u the same after arbitrary many rounds.*

3.2 FOC_2 and AC-GNN classifiers

Having Propositions 2.2 and 3.2 at hand, one may be tempted to combine them and claim that every FOC_2 formula can be captured by an AC-GNN. Yet this is not the case, as we show in Proposition 3.3 below. In fact, while it is true that two nodes are indistinguishable by the WL test if and only if they are indistinguishable by FOC_2 (Proposition 3.2), and if the former holds then such nodes cannot be distinguished by AC-GNNs (Proposition 2.2), this by no means tells us that every FOC_2 formula can be captured by an AC-GNN.

PROPOSITION 3.3. *There are* FOC_2 *formulas that are not captured by any AC-GNN. In fact, this holds even for FO formulas using only two variables and no counting quantifiers.*

PROOF. Consider the formula $\alpha(v) \coloneqq \mathsf{Red}(v) \wedge \exists x\ \mathsf{Green}(x)$. We will show by contradiction that there is no AC-GNN that captures α, no matter which aggregation, combination, and final classification functions are allowed. Indeed, assume that $\mathcal{A}$ is an AC-GNN capturing α, and let L be its number of layers. Consider the graph G that is a chain of $L+2$ nodes colored Red, and consider the first node v_0 in that chain. Since $\mathcal{A}$ captures α, and since $(G, v_0) \not\models$

α, we have that $\mathcal{A}$ labels v_0 with false, that is, $\mathcal{A}(G, v_0) = \mathsf{false}$. Now, consider the graph G' obtained from G by coloring the last node in the chain with Green (instead of Red). Then one can easily show that $\mathcal{A}$ again labels v_0 by false in G'. But we have $(G', v_0) \models \alpha$, a contradiction. $\square$

The above proof relies on the following weakness of AC-GNNs: if the number of layers is fixed (i.e., does not depend on the input graph), then the information of the color of a node v cannot travel further than at distance L from v. Nevertheless, we can show that the same holds even when we consider AC-GNNs that dispose of an arbitrary number of layers (for instance, one may want to run a homogeneous AC-GNN for $f(|E|)$ layers for each graph $G = (V, E)$, for a fixed function f). Assume again by way of contradiction that $\mathcal{A}$ is such an extended AC-GNN capturing α. Consider the graph G consisting of two disconnected nodes v, u, with v colored Red and y colored Green. Then, since $(G, v) \models \alpha$, we have $\mathcal{A}(G, v) = \mathsf{true}$. Now consider the graph G' obtained from G by changing the color of u from Green to Red. Observe that, since the two nodes are not connected, we will again have $\mathcal{A}(G', v) = \mathsf{true}$, contradicting the fact that $(G', v) \not\models \alpha$ and that $\mathcal{A}$ is supposed to capture α.

From these proofs we get two pieces of intuition. One problem is that an AC-GNN has only a fixed number L of layers and hence the information of local aggregations cannot travel further than at distance L of every node along edges in the graph. But there are times when no number of layer suffices, simply because two nodes may be disconnected in the graph. This negative result opens up the following questions.

1. What kind of FOC_2 formulas can be captured by AC-GNNs?
2. Can we capture FOC_2 classifiers with GNNs using a simple extension of AC-GNNs?

We answer these questions in the next two sections.

4. EXPRESSIVE POWER OF AC-GNNS

Towards answering our first question, we recall that the problem with AC-GNN classifiers is that they are local, in the sense that they cannot see across a distance beyond their number of layers. Thus, if we want to understand which queries this architecture is capable of expressing, we must consider logics built with similar limitations in mind. And indeed, in this section we show that AC-GNNs capture any FOC_2 formula as long as they satisfy such a locality property. This happens to be a well-known restriction of FOC_2 that corresponds to *graded modal logic* [9] or, equivalently, to the description logic $\mathcal{ALCQ}$ [2], which is fundamental for knowledge representation: for instance, the OWL 2 Web Ontology Language [24, 31] relies on $\mathcal{ALCQ}$.

The idea of graded modal logic is to force all subformulas to be *guarded* by the edge predicate Edge. This means that one cannot express in graded modal logic arbitrary formulas of the form $\psi(x) = \exists y\, \varphi(y)$ (that is, whether there is some node that satisfies property φ). Instead, one is allowed to check whether some *neighbor* y of the node x where the formula is being evaluated satisfies φ. For instance, we are allowed to express the formula $\psi(x) = \exists y\, (\mathsf{Edge}(x, y) \wedge \varphi(y))$ in the logic as in this case $\varphi(y)$ is guarded by $\mathsf{Edge}(x, y)$.

We can formally define this logic using FOC_2 syntax as follows (note that both graded modal logic and $\mathcal{ALCQ}$ have their own syntaxes, but we stick to the general FO syntax for uniformity).

Definition 4.1. *A* graded modal logic formula *is either* $\mathrm{Col}(x)$*, for* Col *a node color, or one of the following, where* φ *and* ψ *are graded modal logic formulas and* N *is a positive integer:*

$$\neg\varphi(x), \quad \varphi(x) \wedge \psi(x), \quad \exists^{\geq N} y\, (\mathsf{Edge}(x, y) \wedge \varphi(y)).$$

For example, the formula

$$\delta(x) \;\; := \;\; \mathsf{Red}(x) \;\wedge\; \exists y \left(\mathsf{Edge}(x, y) \;\wedge\; \mathsf{Blue}(y)\right)$$

is in graded modal logic, but the formula

$$\gamma(x) := \mathsf{Red}(x) \wedge \exists y\big(\neg\mathsf{Edge}(x, y) \wedge \exists^{\geq 2} x\big[\,\mathsf{Edge}(y, x) \wedge \mathsf{Blue}(x)\,\big]\big).$$

of Section 3 is not, because the use of $\neg\mathsf{Edge}(x, y)$ as a guard is disallowed. Observe that all graded modal logic formulas are unary by definition, so all of them define unary queries. As promised, we now show that AC-GNNs can indeed capture all graded modal logic classifiers.

Proposition 4.2. *Each graded modal logic classifier is captured by a simple homogeneous AC-GNN.*

Proof sketch. The key idea of the construction is that the components of a node's feature vector can represent the subformulas of the captured logical classifier that hold in the node. An AC-GNN then can implement a standard dynamic programming algorithm over the graph G such that, after k layers, it declares a feature in a node v to be 1 iff v satisfies the corresponding subformula φ over G.

Let $\varphi(x)$ be a formula in graded modal logic. Let $\text{sub}(\varphi) = (\varphi_1, \varphi_2, \ldots, \varphi_L)$ be an enumeration of the sub-formulas of φ such that if φ_k is a sub-formula of φ_ℓ then $k \leq \ell$. We show how to construct a simple and homogeneous AC-GNN $\mathcal{A}_\varphi$ capturing $\varphi(x)$. As mentioned, the idea is that $\mathcal{A}_\varphi$ uses feature vectors in $\mathbb{R}^L$ such that every component of those vectors represents a different formula in $\text{sub}(\varphi)$. Then $\mathcal{A}_\varphi$ will update the feature vector $\boldsymbol{x}_v^{(i)}$ of node v ensuring that component ℓ of $\boldsymbol{x}_v^{(i)}$ gets a value 1 if and only if the formula φ_ℓ is satisfied in node v, for every $i \geq l$. We note that $\varphi = \varphi_L$ and thus, the last component of each feature vector after evaluating L layers in every node gets a value 1 if and only if the node satisfies φ. We will then be able to use a final classification function CLS that simply extracts that particular component.

The simple homogeneous AC-GNN $\mathcal{A}_\varphi$ has L layers and uses aggregation and combine functions

$$\begin{aligned} \text{AGG}(X) &= \sum_{\boldsymbol{x} \in X} \boldsymbol{x}, \\ \text{COM}(\boldsymbol{x}, \boldsymbol{y}) &= \sigma\big(\boldsymbol{x}\boldsymbol{C} + \boldsymbol{y}\boldsymbol{A} + \boldsymbol{b}\big), \end{aligned}$$

where $\boldsymbol{A}, \boldsymbol{C} \in \mathbb{R}^{L \times L}$, and $\boldsymbol{b} \in \mathbb{R}^L$ are defined next, and σ is the truncated ReLU activation defined by $\sigma(x) := \min(\max(0, x), 1)$. The entries of the ℓ-th columns of $\boldsymbol{A}, \boldsymbol{C}$, and $\boldsymbol{b}$ depend on the sub-formulas of φ as follows:

- if $\varphi_\ell(x) = \text{Col}(x)$ with Col one of the (base) colors, then $C_{\ell\ell} = 1$,
- if $\varphi_\ell(x) = \varphi_j(x) \wedge \varphi_k(x)$ then $C_{j\ell} = C_{k\ell} = 1$ and $b_\ell = -1$,
- if $\varphi_\ell(x) = \neg\varphi_k(x)$ then $C_{k\ell} = -1$ and $b_\ell = 1$,
- if $\varphi_\ell(x) = \exists^{\geq N}(E(x, y) \wedge \varphi_k(y))$ then $A_{k\ell} = 1$ and $b_\ell = -N + 1$,

and all other values in the ℓ-th columns of $\boldsymbol{A}, \boldsymbol{C}$, and $\boldsymbol{b}$ are 0.

To show correctness, let $G = (V, E)$ be a colored graph. For every node v in G we consider the initial feature vector $\boldsymbol{x}_v^{(0)} = (x_1, \ldots, x_L)$ such that $x_\ell = 1$ if sub-formula φ_ℓ is the initial color assigned to v, and $x_\ell = 0$ otherwise. By definition, AC-GNN $\mathcal{A}_\varphi$ will iterate the aggregation and combine functions defined above for L rounds (L layers) to produce feature vectors $\boldsymbol{x}_v^{(i)}$ for every node $v \in G$ and $i = 1, \ldots, L$. All that remains is to show that for every $\varphi_\ell \in \text{sub}(\varphi)$, every $i \in \{\ell, \ldots, L\}$, and every node v in G it holds that:

$(\boldsymbol{x}_v^{(i)})_\ell = 1$ if $(G, v) \models \varphi_\ell$ and $(\boldsymbol{x}_v^{(i)})_\ell = 0$ otherwise,

where $(\boldsymbol{x}_v^{(i)})_\ell$ is the ℓ-th component of $\boldsymbol{x}_v^{(i)}$. But this can easily be proved by induction on the number of sub-formulas of every φ_ℓ. □

An interesting open question is whether the same kind of construction can be done with AC-GNNs using different aggregate and combine operators from the ones we consider here; for instance, using max instead of sum to aggregate the feature vectors of the neighbors, or using other non-linearities such as sigmoid.

Interestingly, the relationship between AC-GNNs and graded modal logic goes further: we can show that graded modal logic is the *largest* class of FO logical classifiers captured by AC-GNNs—that is, the only FO formulas that AC-GNNs are able to learn accurately are those in graded modal logic.

THEOREM 4.3. *A logical classifier is captured by AC-GNNs if and only if it can be expressed in graded modal logic.*

The backward direction of this theorem is Proposition 4.2. On the other hand, the proof of the forward direction is based on a van Benthem & Rosen characterization obtained by Otto [26, Theorem 2.2] for finite graphs, stating that an FO formula can be expressed in graded modal logic if and only if the formula only depends on the *unraveling* of the nodes, which in turn correspond to the colors assigned by the WL test. While the setting considered by Otto is slightly different from ours (in particular, we consider directed graphs, as opposed to undirected), these differences can be shown to be inessential, and the proof carries over to this setting. We point out that the forward direction holds no matter which aggregate and combine operators are considered—that is, this is a limitation of the AC-GNN architecture, not of the specific functions that one chooses to update the features.

5. GNNS FOR CAPTURING FOC_2

In this section we tackle our second question: Which GNN architectures do we need to capture all FOC_2 classifiers? Recall that the main shortcoming of AC-GNNs for expressing such classifiers is their local behavior. A natural way to avoid this behavior is to allow for a global feature computation on each layer of the GNN. This is called a *global attribute* computation in the framework of [6]. Following the recent GNN literature [13, 23, 34], we refer to this global operation as a *readout*. We begin with formalizing the GNN architecture with readouts. We then show how readouts serve in capturing all of FOC_2, and finish with an observation on the number of readouts needed in these neural networks.

5.1 GNNs with global readouts

Our definition of GNNs with readouts is a generalization of the Definition 2.1 for AC-GNNs.

DEFINITION 5.1. *An* aggregate-combine-readout GNN *(*ACR-GNN*) with L layers extends AC-GNNs by* readout *functions* $\{\text{READ}^{(i)}\}_{i=1}^{L}$*, which aggregate the (multiset of the) current feature vectors of all the nodes in a graph to a single vector; additionally, the combination functions* $\text{COM}^{(i)}$ *take three arguments rather than two. Then, the feature vector* $\boldsymbol{x}_v^{(i)}$ *of each node v in a graph G on each layer i is computed by the following recursive formula, where V is the set of all nodes in G:*

$$\boldsymbol{x}_v^{(i)} = \text{COM}^{(i)}\Big(\boldsymbol{x}_v^{(i-1)}, \text{AGG}^{(i)}\left(\{\!\{\boldsymbol{x}_u^{(i-1)} \mid u \in \mathcal{N}_G(v)\}\!\}\right), \text{READ}^{(i)}\left(\{\!\{\boldsymbol{x}_u^{(i-1)} \mid u \in V\}\!\}\right)\Big).$$

Intuitively, every layer in an ACR-GNN first computes (i.e., "reads out") the aggregation over all the nodes in G; then, for every node v, it computes the aggregation over the neighbors of v; and finally it combines the features of v with the two aggregation vectors.

All the notions about AC-GNNs extend to ACR-GNNs in a straightforward way; for example, a *simple* ACR-GNN uses the sum as the function $\text{READ}^{(i)}$ in each layer, and the following combination function, generalizing Equation (1):

$$\text{COM}^{(i)}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = f\big(\boldsymbol{x}_1\boldsymbol{C}^{(i)} + \boldsymbol{x}_2\boldsymbol{A}^{(i)} + \boldsymbol{x}_3\boldsymbol{R}^{(i)} + \boldsymbol{b}^{(i)}\big),$$

where $\boldsymbol{R}^{(i)}$ is one more matrix of parameters.

5.2 ACR-GNNs and FOC_2

To see how readout functions could help in capturing non-local properties, consider again the formula $\gamma(x)$ from above, that assigns true to every red node v unless there is another node not connected with v having at least two blue neighbors. It is easy to show, by adapting the proof of Proposition 3.3, that no AC-GNN can capture this classifier. However, using a single readout and local aggregations, one can implement this classifier as follows. Let B be the property "having at least 2 blue neighbors". Then an ACR-GNN that implements $\gamma(x)$ can first use a local aggregation to store in the feature of every node if the node satisfies B, then use a readout function to count the nodes satisfying B in the whole graph, and finally use another local aggregation to count neighbors of every node satisfying B. Then γ is obtained by classifying as true every red node having less neighbors satisfying B than the total number of nodes satisfying B in the whole graph. It turns out that the usage of readout functions is enough to capture all non-local properties of FOC_2 classifiers.

THEOREM 5.2. *Each* FOC_2 *classifier can be captured by a simple homogeneous ACR-GNN.*

PROOF SKETCH. As an intermediate step in the proof, we use a characterization of FOC_2 using an extended version of graded modal logic, which was obtained by Lutz et al. [19], and relates FO_2 with a modal logic that can use parameters to navigate to all nodes not connected, or different to, the current node. This connection can be extended to FOC_2 and the counting version of this modal logic, which is denoted as $\mathcal{EMLC}$.

Next, we show how to capture any $\mathcal{EMLC}$ formula with an ACR-GNN. Since $\mathcal{EMLC}$ formulas are essentially the extension of graded modal logic with these negated modalities, we can reuse most of the proof of Proposition 4.2. The novelty is that we use readouts to take care of subformulas with negated modalities. Thus, readout functions are only used to deal with subformulas asserting the existence of a node that is not connected to the current node in the graph, just as we did for classifier $\gamma(x)$. □

Note that Proposition 4.2 has two directions while Theorem 5.2 just one; we leave as a challenging open problem the other direction of Theorem 5.2—that is, whether the FOC_2 classifiers are exactly the logical classifiers (i.e., FO logic unary formulas) captured by ACR-GNNs.

5.3 The number of layers with readouts

The proof of Theorem 5.2 constructs ACR-GNNs whose number of layers depends on the size of the formula being captured; moreover, readouts are used on unboundedly many (in some cases all) layers of these GNNs. Given that a global computation can be costly, one might wonder whether this is really needed, or if it is possible to cope with all the complexity of such classifiers by performing only a few readouts. We next show the surprising fact that just one readout is actually always enough. However, this reduction in the number of readouts comes at the cost of severely complicating the resulting GNN.

DEFINITION 5.3. *An* aggregate-combine GNN with final readout *(*AC-FR-GNN*) is the same as an ACR-GNN except that only the final layer uses a readout function.*

The following theorem formalizes the result of this section.

THEOREM 5.4. *Each* FOC_2 *classifier is captured by an AC-FR-GNN.*

The AC-FR-GNN construction in the proof of this theorem is not based on the idea of evaluating the formula incrementally along layers, as in the proofs of Proposition 4.2 and Theorem 5.2, and it is not simple (note that AC-FR-GNNs are never homogeneous). Instead, it is based on a refinement of the GIN architecture proposed by Xu et al. [34] (which is also used in the proof of the second claim of Proposition 2.2) to obtain as much information as possible about the local neighborhood in graphs, followed by a readout and combination functions that use this information to deal with non-local constructs in formulas. The first component we build is an AC-GNN that computes an invertible function mapping each node to a number representing its neighborhood (how big is this neighborhood depends on the classifier to be captured). This information is aggregated so that we know for each different type of a neighborhood how many times it appears in the graph. We then use the combine function to evaluate FOC_2 formulas by decoding back the neighborhoods.

6. EXPERIMENTAL RESULTS

In this section we report on our experiments, which are aimed to empirically validate our theoretical findings.

6.1 Overview and Set Up

Our main motivation was to show that the theoretical expressiveness of ACR-GNNs, as well as the difference between AC- and ACR-GNNs, can actually be observed when we learn from examples. To this end, we performed two sets of experiments on synthetic data: experiments to show that ACR-GNNs can learn a very simple FOC_2 node classifier that AC-GNNs cannot learn, and experiments involving complex FOC_2 classifiers that need more intermediate readouts to be learned. Besides testing simple AC-GNNs, we also tested the GIN network proposed by Xu et al. [34] (we used the implementation by Fey and Lenssen [11] and adapted it to classify nodes).

We performed these two experiments using synthetic graphs with five initial colors; these graphs are divided in three sets: train set with 5000 graphs with 50 to 100 nodes, test set with 500 graphs with a similar number of nodes, and another test set with 500 graphs about twice larger than the train set.

We tried several configurations for the aggregation, combination and readout functions, but observed a consistent pattern in which the setting of simple AC(R)-GNNs with ReLU activation as described above produced the most accurate results. Besides this, we did not do any hyperparameter search and did not use any regularisation. Accuracy in our experiments is computed as the total number of nodes correctly classified among all nodes in all the graphs in the dataset. In addition, we report on our preliminary experiments on a real-life *Protein-Protein Interaction* (*PPI*) benchmark [36], where we did not observe an improvement of ACR-GNNs over AC-GNNs.

We implemented our experiments using the PyTorch Geometric library [11]. In all cases we trained with a batch-size of 128, and run up to 50 epochs with the Adam optimizer and default PyTorch parameters.[2]

6.2 Separating AC-GNNs and ACR-GNNs

In our first set of experiments we considered a very simple FOC_2 classifier defined by

$$\alpha(x) \coloneqq \mathsf{Red}(x) \wedge \exists y\ \mathsf{Blue}(y),$$

which is satisfied by every red node in a graph provided that the graph contains at least one blue node. This classifier is not expressible in graded modal logic, so we expected very good performance from ACR-GNNs but difficulties for AC-GNNs.

We tested the GNN architectures with two classes of graphs. First, we considered line-shaped graphs, each of which has $2n$ nodes $v_1, \ldots, v_{2n}$ such that each v_i is connected to v_{i+1}, and such that only nodes $v_1, \ldots, v_n$ can be colored blue and only others can be colored red. Second, we considered Erdös-Renyi random graphs of two flavors: the graphs with the same number of nodes and edges, and the graphs where the number of edges is twice the number of nodes. In every set we had 50% of graphs containing no blue node, and others containing a fixed small number of blue nodes (typically less than five). Also, to ensure that there is a significant number of nodes satisfying the formula, we forced graphs to have at least $1/4$ of its nodes colored red.

The results of these experiments are shown in Table 1. As we can see there, already ACR-GNNs with a single layer showed perfect performance for both types of graphs (ACR-1 in Table 1). This was what we expected given the simplicity of the property being checked. In contrast, AC-GNNs and GINs (shown in Table 1 as AC-L and GIN-L, representing AC-GNNs and GINs with L layers) struggle to fit the data. For the case of the line-shaped graph, they were not able to fit the train data even by al-

[2]All our code and data can be accessed online at https://github.com/juanpablos/GNN-logic.

	Line-Shaped Train	Line-Shaped Test		Erdös-Renyi Train	Erdös-Renyi Test	
		same-size	bigger		same-size	bigger
AC-5	0.887	0.886	0.892	0.951	0.949	0.929
AC-7	0.892	0.892	0.897	0.967	0.965	0.958
GIN-5	0.861	0.861	0.867	0.830	0.831	0.817
GIN-7	0.863	0.864	0.870	0.818	0.819	0.813
ACR-1	1.000	1.000	1.000	1.000	1.000	1.000

Table 1: Results on synthetic data for nodes labeled by classifier $\alpha(x) \coloneqq \mathsf{Red}(x) \wedge \exists y\ \mathsf{Blue}(y)$

lowing 7 layers. For the case of random graphs, the performance with 7 layers was considerably better but still did not fit the data perfectly. We allowed AC-GNNs with 7 layers to run for more epochs but the results did not improve.

In a closer look at the performance for different connectivities of E-R graphs, we found an improvement for AC-GNNs when we train them with more dense graphs (i.e., when the number of edges increases while the number of nodes stays the same). This is consistent with the fact that AC-GNNs are able to move information of local aggregations to distances up to their number of layers. This combined with the fact that random graphs that are more dense make the maximum distances between nodes shorter, may explain the boost in performance for AC-GNNs.

6.3 Complex FOC_2 Properties

In the second experiment we consider classifiers $\alpha_i(x)$ constructed as

$$\alpha_0(x) \coloneqq \mathsf{Blue}(x), \tag{2}$$

$$\alpha_i(x) \coloneqq \exists^{[N_i,M_i]} y \left(\alpha_{i-1}(y) \wedge \neg \mathsf{Edge}(x,y)\right), \tag{3}$$

where $\exists^{[N,M]}$ stands for "there are exactly between N and M nodes" satisfying a given property. Observe that each $\alpha_i(x)$ is in FOC_2, as $\exists^{[N,M]}$ can be expressed by combining $\exists^{\geq N}$ and $\neg\exists^{\geq M+1}$; however, the classifiers α_i, for $i \geq 1$, are not expressible in graded modal logic. In particular, we concentrated on $\alpha_1(x)$, $\alpha_2(x)$ and $\alpha_3(x)$ with $[N_1, M_1]$, $[N_2, M_2]$ and $[N_3, M_3]$ being $[8, 10]$, $[10, 20]$ and $[10, 20]$ (the choice of these intervals is technical, it results in the number of satisfying nodes in the graphs as described below).

We considered sets of Erdös-Renyi random graphs with the number of edges about 7 times greater than the number of nodes (i.e., more dense that in the first experiments), and colored to ensure that approximately one half of all nodes in the graphs in the set satisfy each of $\alpha_1(x)$, $\alpha_2(x)$ and $\alpha_3(x)$.

Our results, given in Table 2, show that when increasing i (i.e., the quantifier depth of the classifiers α_i) more layers are needed to increase train and test accuracy. We report ACR-GNNs performance up to 3 layers (ACR-L in Table 2) as beyond that we did not see any significant improvement. We also note that for the bigger test set, AC-GNNs and GINs are unable to substantially depart from a trivial baseline of 50%. We tested these networks with up to 10 layers but only report the best results on the bigger test set. We also test AC-FR-GNNs with two and three layers (AC-FR-L in Table 2). As we expected, although theoretically using a single readout gives the same expressive power as using several of them (Theorem 5.4), in practice more than a single readout can actually help the learning process of complex properties.

6.4 Experiments with PPI benchmark

Finally, we also tested AC- and ACR-GNNs on the PPI benchmark [36]. We chose PPI since it is a node classification benchmark with different graphs in the train set (as opposed to other popular benchmarks for node classification such as Core or Citeseer that have a single graph). Although the best results we obtained for both classes of GNNs on PPI were quite high (AC-GNNs: 97.5 F1, ACR-GNNs: 95.4 F1 in the test set), we did not observe an improvement of ACR-GNNs over AC-GNNs.

However, Chen et al. have recently observed that commonly used benchmarks are inadequate for testing advanced GNN variants, and ACR-GNNs might be suffering from this fact [8]. Thus, the fact that we do not observe any improvement may be an artefact of the simplicity of the benchmark. We left as future work a more thorough testing and tuning of ACR-GNNs for real data.

7. FINAL REMARKS

Our results show the theoretical advantages of mixing local and global information when classifying nodes in a graph. Recent works have also observed these advantages in practice; e.g., Deng et al. used global-context aware local descriptors to classify objects in 3D point clouds [10], You et

	α_1 Train	α_1 Test		α_2 Train	α_2 Test		α_3 Train	α_3 Test	
		same-size	bigger		same-size	bigger		same-size	bigger
AC	0.839	0.826	0.671	0.694	0.695	0.667	0.657	0.636	0.632
GIN	0.567	0.566	0.536	0.689	0.693	0.672	0.656	0.643	0.580
AC-FR-2	1.000	1.000	1.000	0.863	0.860	0.694	0.788	0.775	0.770
AC-FR-3	1.000	1.000	0.825	0.840	0.823	0.604	0.787	0.767	0.771
ACR-1	1.000	1.000	1.000	0.827	0.834	0.726	0.760	0.762	0.773
ACR-2	1.000	1.000	1.000	0.895	0.897	0.770	0.800	0.799	0.771
ACR-3	1.000	1.000	1.000	0.903	0.902	0.836	0.817	0.802	0.748

Table 2: Results on Erdös-Renyi graphs with nodes labeled according to classifiers α_i

al. construct node features by computing shortest-path distances to a set of distant anchor nodes [35], and Haonan et al. introduced the idea of a "star node" storing global information of the graph [15].

As mentioned before, our work is close in spirit to that of [34] and [23] establishing the correspondence between the WL test and GNNs. In contrast to our work, they focus on graph classification and do not consider the relationship with logical classifiers.

Regarding our results on the links between AC-GNNs and graded modal logic (Theorem 4.3), we point out that very recent work [1] establishes close relationships between GNNs and certain classes of *distributed local algorithms*. These in turn have been shown to have strong correspondences with modal logics [16]. Hence, it may be the case that variants of our Proposition 4.2 could be obtained by combining these two lines of work (but it is not clear if this combination would yield AC-GNNs that are *simple*), and we believe this is an interesting direction for future work. Moreover, we also don't know how to bridge our work with that of distributed algorithms when we add *non-local computations* to GNNs (such as the readouts that we consider).

Morris et al. [23] also studied k-GNNs, which are inspired by the k-dimensional WL test. In k-GNNs, graphs are considered as structures connecting k-tuples of nodes instead of just pairs of nodes. We plan to study how our results on logical classifiers relate to k-GNNs, in particular, with respect to the logic FOC_k that extends FOC_2 by allowing formulas with k variables, for each fixed $k > 1$. Recent work has also explored the extraction of finite state representations from recurrent neural networks as a way of explaining them [33, 18, 25]. We would like to study how our results can be applied for extracting logical formulas from GNNs as possible explanations for their computations.

We would like to remark that studying GNNs continues to be an important topic in the community, with new advances reported every year. The latest results involve the study of more complex GNN architectures, that take us beyond AC-GNNs and even k-GNNs. This extra power may come, e.g., as a result of allowing random information to be computed for each node [28], allowing for more complex aggregating functions [20], or different schemes of port assignments in a distributed setting [12].

Funding All authors but Kostylev are funded the Millennium Institute for Foundational Research on Data[3]. Barceló and Pérez are funded by Fondecyt grant 1200967.

8. REFERENCES

[1] Approximation ratios of graph neural networks for combinatorial problems.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.

[3] F. Baader and C. Lutz. Description logic. In *Handbook of modal logic*, pages 757–819. North-Holland, 2007.

[4] P. Barceló, E. V. Kostylev, M. Monet, J. Pérez, J. Reutter, and J. P. Silva. The logical expressiveness of graph neural networks. In *International Conference on Learning Representations*, 2019.

[5] P. Barceló Baeza. Querying graph databases. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI symposium on Principles of database systems*, pages 175–188, 2013.

[6] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. F. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, Ç. Gülçehre, H. F. Song, A. J. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. R. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and

[3] https://imfd.cl/en/

R. Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

[7] J.-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[8] T. Chen, S. Bian, and Y. Sun. Are powerful graph neural nets necessary? A dissection on graph classification. *CoRR*, abs/1905.04579, 2019.

[9] M. de Rijke. A note on graded modal logic. *Studia Logica*, 64(2):271–283, 2000.

[10] H. Deng, T. Birdal, and S. Ilic. PPFnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018*, pages 195–205, 2018.

[11] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. *CoRR*, abs/1903.02428, 2019.

[12] V. K. Garg, S. Jegelka, and T. Jaakkola. Generalization and representational limits of graph neural networks. *arXiv preprint arXiv:2002.06157*, 2020.

[13] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August, 2017*, pages 1263–1272, 2017.

[14] W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, NIPS 2017, Long Beach, CA, USA, December 4–9, 2017*, pages 1024–1034, 2017.

[15] L. Haonan, S. H. Huang, T. Ye, and G. Xiuyan. Graph star net for generalized multi-task learning. *arXiv preprint arXiv:1906.12330*, 2019.

[16] L. Hella, M. Järvisalo, A. Kuusisto, J. Laurinharju, T. Lempiäinen, K. Luosto, J. Suomela, and J. Virtema. Weak models of distributed computing, with connections to modal logic. *Distributed Computing*, 28(1):31–53, 2015.

[17] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017*, 2017.

[18] A. Koul, S. Greydanus, and A. Fern. Learning finite state representations of recurrent policy networks. In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*, 2019.

[19] C. Lutz, U. Sattler, and F. Wolter. Modal logic and the two-variable fragment. In *Proceedings of the International Workshop on Computer Science Logic, CSL 2001, Paris, France, September 10–13, 2001*, pages 247–261. Springer, 2001.

[20] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks, 2019.

[21] C. Merkwirth and T. Lengauer. Automatic generation of complementary descriptors with molecular graph networks. *J. of Chemical Information and Modeling*, 45(5):1159–1168, 2005.

[22] T. M. Mitchell et al. Machine learning, 1997.

[23] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman go neural: higher-order graph neural networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 – February 1, 2019*, pages 4602–4609, 2019.

[24] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web ontology language profiles (second edition). W3C recommendation, W3C, 2012.

[25] C. Oliva and L. F. Lago-Fernández. On the interpretation of recurrent neural networks as finite state machines. In *Part I of the Proceedings of the 28th International Conference on Artificial Neural Networks, ICANN 2019, Munich, Germany, September 17–19, 2019*, pages 312–323. Springer, 2019.

[26] M. Otto. Graded modal logic and counting bisimulation. `https://www2.mathematik.tu-darmstadt.de/~otto/papers/cml19.pdf`, 2019.

[27] I. Robinson, J. Webber, and E. Eifrem. *Graph databases*. " O'Reilly Media, Inc.", 2013.

[28] R. Sato, M. Yamada, and H. Kashima. Random features strengthen graph neural networks. *arXiv preprint arXiv:2002.03155*, 2020.

[29] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The

graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2009.
[30] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *Proceedings of The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018*, pages 593–607, 2018.
[31] W3C OWL Working Group. OWL 2 Web ontology language document overview (second edition). W3C recommendation, W3C, 2012.
[32] B. Y. Weisfeiler and A. A. Leman. A Reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968. Translated from Russian.
[33] G. Weiss, Y. Goldberg, and E. Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018*, pages 5244–5253, 2018.
[34] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How Powerful are graph neural networks? In *Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*, 2019.
[35] J. You, R. Ying, and J. Leskovec. Position-aware graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA, June 9–15, 2019*, pages 7134–7143, 2019.
[36] M. Zitnik and J. Leskovec. Predicting multicellular function through multi-layer tissue networks. *CoRR*, abs/1707.04638, 2017.

Sketches of Dynamic Complexity

Thomas Schwentick
TU Dortmund
thomas.schwentick@tu-dortmund.de

Nils Vortmeier
TU Dortmund
nils.vortmeier@tu-dortmund.de

Thomas Zeume
Ruhr University Bochum
thomas.zeume@rub.de

ABSTRACT

How can the result of a query be updated after changing a database? This is a fundamental task for database management systems which ideally takes previously computed information into account. In dynamic complexity theory, it is studied from a theoretical perspective where updates are specified by rules written in first-order logic.

In this article we sketch recent techniques and results from dynamic complexity theory with a focus on the reachability query.

1. INTRODUCTION

Assume you are running a very traditional relational DBMS that supports all queries that can be expressed in the relational algebra, but nothing else.[1] Then you precisely understand what kinds of queries you can pose and which you cannot: you are limited to queries that are expressible in first-order logic.

You think that it might be helpful that you are interested in continuously asking the same query. Maybe the database you maintain is actually a graph database and you would be interested to evaluate a fixed set of regular path queries all over again. Maybe you also know that changes to your database are not very frequent. Is there a way to cope with your queries without writing programs or installing that graph database engine?

This is the setting that is assumed in this article and the setting of dynamic complexity as introduced by Patnaik and Immerman [34] and similarly by Dong and Su [13] in the early nineties: there is an initially empty database, tuples can be inserted and deleted and after each change of the database, the answer to some fixed query needs to be computed with first-order logic means. Besides the "real" relations, the database can have additional, auxiliary relations, one of which always represents the query answer to the standing query. After each change step your database can apply first-order queries to update these auxiliary relations.

This setting is similar to other typical database settings, but it differs from a typical incremental query maintenance setting in that it addresses queries that are *not* expressible in the relational algebra, and from a typical view maintenance setting because auxiliary relations are allowed[2].

In this article, we want to report on some progress that dynamic complexity has seen during the last years. Besides the result that reachability on directed graphs *can* be maintained in this framework, much of the research has focussed on new techniques and the extension of the framework towards bulk changes, as opposed to single-tuple changes. The ability to maintain regular path queries is one outcome of this line of work.

We develop the framework incrementally while giving sketches[3] of some recent and some older key results and techniques. Most of these results concern the reachability query Reach on directed or undirected graphs. This query maps a graph $G = (V, E)$ to the transitive closure of the edge relation E. In other words, Reach(G) is the binary relation that contains a pair (u, v) of nodes if there is a non-empty path from u to v in G. An immediate consequence of these results for maintaining reachability is that regular path queries can be maintained as well, see Sketch 10.

In this article we borrow from several talks we presented in the last few years as well as from some of our articles [8, 11, 10, 9]. For recent, more complete expositions of the current state of the art of dynamic complexity we refer to [40, 43].

2. MAINTAINING REACHABILITY

We start with the very simple scenario where only single edges can be inserted into the graph (database).

Sketch 1:

Single-edge insertions into directed graphs

Since we aim at updating the standing query Reach, the transitive closure of the edge relation is stored as

[1] This assumption is not entirely realistic but very common in foundational database research.

[2] We emphasise that recent higher-order incremental view maintenance frameworks also use auxiliary views [29, 33].

[3] As in "brief description", not as used in, e.g., streaming algorithms.

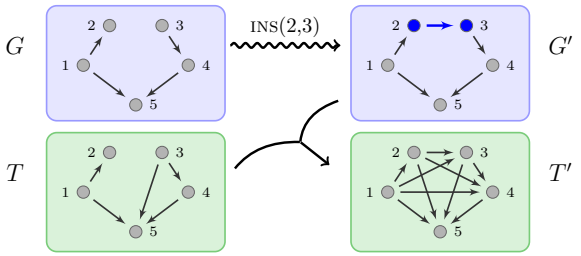


Figure 1: The dynamic scenario. After inserting edge $(2,3)$ there is a path from $x = 1$ to $y = 4$ thanks to the previously existing paths from 1 to 2 and from 3 to 4.

an auxiliary relation.

How can we update the transitive closure of a graph after inserting a single edge? After inserting an edge (u, v), there is a path from a node x to a node y if there has been a path before the insertion or if there were paths from x to u and from v to y, cf. Figure 1. Thus, if T denotes the auxiliary relation that stores the transitive closure of E, the update that needs to be applied can be specified as follows.

on insert (u, v) **update** T **as**

$$T'(x, y) \stackrel{\text{def}}{=} T(x, y) \vee \big(T(x, u) \wedge T(v, y)\big)$$

The semantics is that, after inserting the edge (u, v), the relation T is replaced according to the query $T(x, y) \vee \big(T(x, u) \wedge T(v, y)\big)$.

We call the above rule an *update rule*. A *dynamic (first-order) program* can use finitely many auxiliary relations $R_1, \ldots, R_m$ and provides a *(first-order) update rule* for each of these relations and each admissible change operation. In the above case, the only admissible *change operation* is insertion of edges.[4] Each update rule can access the edge relation and (the current versions of the) relations $R_1, \ldots, R_m$.

Most often, admissible change operations are insertions or deletions of edges. But the exact form, e.g., whether single-tuple or bulk changes are allowed and how they are specified, depends on the context. When an actual change occurs, the program updates its auxiliary relations by simultaneously applying their respective update rules for the underlying change operation.

A dynamic program *maintains* the result of a query if some designated auxiliary relation stores the result of the query after all possible sequences of admissible changes. As an example, the above (single-rule) program maintains the query Reach on directed graphs under single-edge insertions. We emphasise that this particular rule does not even use quantifiers.

[4]We note that each *change operation* can be instantiated by actual *changes*, e.g., the insertion of a concrete edge.

The class DynFO consists of all pairs $(\mathcal{Q}, \Delta)$ such that the query $\mathcal{Q}$ can be maintained by a dynamic first-order program under the set Δ of admissible changes. For a pair $(\mathcal{Q}, \Delta) \in$ DynFO we usually say that $\mathcal{Q}$ is in DynFO under Δ-change operations. As we have just seen, Reach is in DynFO under single-edge insertions.

2.1 Undirected Reachability: from single to bulk changes

Allowing insertions *and* deletions offers "full change power" in the sense that each graph can be transformed into each other graph (with the same vertex set). The question whether Reach can be maintained when edges can be inserted *and* deleted had been a driving force for research in dynamic complexity for twenty years. It turned out that reachability under edge insertions and deletions cannot be maintained in the same simple fashion as in the insertion-only case, and we present some early-known barriers in Section 3.

For now, we concentrate on the easier case of maintaining reachability for undirected graphs. We show how this query can be maintained under insertions and deletions of single edges, and generalise this result to more complex changes. We will come back to reachability for directed graphs in Subsection 2.2.

Besides its elementary update rule, reachability under edge insertions is simple in another sense: it only needs the query answer relation itself as auxiliary relation, i.e., the transitive closure of the edge relation. Trying to maintain a query without any further relations than the query relation itself is a natural first step in the search for a dynamic program. Unfortunately this does not work out for undirected reachability under insertions and deletions [14, Theorem 5.7]. Intuitively, this is because the transitive closure might not yield much information. For instance, the transitive closure of a cycle is a full binary relation which is not helpful for deriving the new transitive closure after deleting two edges from the cycle.

If, as in this case, the query relation does not suffice, one often sees what kind of information is "missing" and one can try to maintain the query by adding another auxiliary relation. This approach could be termed *iterated wishing*: to maintain a certain query, you might wish you had a certain auxiliary relation R_1 available, so you assume you have it, and then you check whether also R_1 can be maintained; if you fail, you wish for more helpful auxiliary relations, and so on.

Sketch 2:

Single-edge insertions and deletions in undirected graphs

After seeing that the transitive closure itself does not suffice, it seems natural to "wish for" a spanning for-

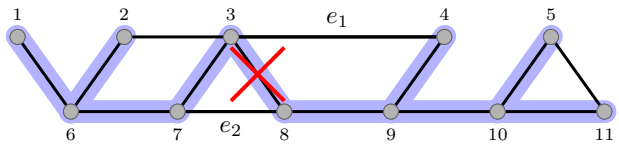


Figure 2: Deleting a single edge from an undirected graph. A spanning forest is highlighted in blue. After removal of $(3,8)$, edges $(3,4)$ and $(7,8)$ are potential new edges for the spanning forest, of which the lexicographically smallest edge $(3,4)$ is chosen by the update rule.

est in order to maintain reachability for undirected graphs under insertions *and deletions*. It turns out that one more wish is helpful: its accompanying in-between relation.[5] More precisely, the following two auxiliary relations can be used to maintain Reach on undirected graphs [35, Theorem 4.1]:

- a binary relation F with $(a,b) \in F$ if (a,b) is an edge of a (fixed) spanning forest; and
- a ternary relation B with $(a,b,c) \in B$ if b is in between a and c in the spanning forest stored in F, so, if a and c are in the same connected component and b is part of the unique path from a to c in the spanning forest.

We now have to verify that F and B can be updated after inserting or deleting an edge, thereby establishing the following result [35, Theorem 4.1].

Proposition 2.1. Reach *on undirected graphs is in* DynFO *under single-edge insertions and deletions.*

Updating the auxiliary relations after edge insertions is very similar to the approach of Sketch 1: if the new edge connects two connected components then it is added to the spanning forest F, otherwise nothing changes. Likewise, deletions of edges not in F are easy to handle, as they leave the auxiliary relations unchanged. We therefore focus on deletions of edges of the spanning forest, see Figure 2.

The formula

$$\psi(x,y,u,v) = E(x,y) \wedge B(x,u,y) \wedge B(x,v,y)$$

expresses that the nodes x and y of the edge (x,y) are in different connected components of the spanning forest after removing (u,v) and thus (x,y) is a candidate for "repairing" the spanning forest.

However, only one such edge can be added to the spanning forest and therefore some tie-breaker is needed. To this end, it is helpful to maintain a linear order on the vertices and to add the lexicographically smallest edge. The linear order is yet another auxiliary relation one can wish for and which can be updated easily: since the edge relation is initially empty, a linear order on the non-isolated nodes can be built based on the order of edge insertions [17]. In fact, not only a linear order can be established in this way, but also 3-ary relations that encode the corresponding addition and multiplication operations.[6]

So far we only considered *simple change operations*, that is, single-tuple changes. This is a typical model, not only in dynamic complexity, but also in dynamic algorithms. However, to deal with realistic scenarios in particular in database contexts, it would be helpful to maintain queries under more complex change operations. That is, change operations should be able to insert or delete sets of tuples.

Obviously, one can not hope for "arbitrary changes": if they were allowed, then one could produce any arbitrary graph in one step from the empty graph.[7] Thus a query can only be maintainable under arbitrary changes if it can actually be explicitly expressed, statically.

Therefore, one has to lower expectations and restrict complex change operations in one way or another. We will consider size-restricted change operations later on, but start here with *first-order definable* change operations. An *insertion query* is specified by a first-order formula $\varphi(x,y,\bar{z})$ and a tuple $\bar{c}$ of elements. It defines the set of edges that are inserted into the edge relation by the set of all tuples (a,b) that satisfy the formula $\varphi(a,b,\bar{c})$. We emphasise that there is no a priori bound on the number of edges that are inserted in such a step.

From a databases point of view, first-order definable change operations (in the spirit of SQL updates) are a very natural kind of complex change operations.

Sketch 3:

Definable insertions into undirected graphs

It turns out that the reachability query on undirected graphs can be maintained under single-edge deletions and first-order definable insertions. More precisely, the result is as follows [39, Theorem 4.2].

Theorem 2.2. *For each finite set Δ of insertion queries,* Reach *on undirected graphs is in* DynFO *under single-edge deletions and under insertions defined by the queries in Δ.*

[5] In fact, it can be seen from the proof of [14, Theorem 5.7] that a spanning forest alone is also not sufficient.

[6] That is, e.g., if a is the smallest node and b the second smallest node with respect to this order, then the triple (a,a,b) is in the ternary relation for the corresponding addition, basically encoding $1+1=2$.

[7] We recall that the empty graph has nodes but not edges.

The dynamic program uses the spanning forest approach as presented in Sketch 2 and relies on a very simple observation, illustrated by Figure 3: if there is a new path between nodes u and v after a first-order defined insertion, then there is such a path in which the number of new edges is bounded by a constant m that only depends on the quantifier depth of the formula defining the insertion. Therefore, for checking whether there is a path between u and v, a first-order update rule can guess at most m newly inserted edges and combine them with previously existing paths.

In general, the constant m can be large, but if the insertion formula is a union of ℓ conjunction queries, it is bounded by 2ℓ [39, Proposition 4.3]. An evaluation of a prototypical implementation [39, Section 5] shows that dynamic programs for insertions defined by small unions of conjunctive queries perform well in some scenarios in comparison with other methods of answering Reach on undirected graphs.

Another obvious restriction of complex changes is to bound the number of edges that can be inserted or deleted in one step. We next consider the insertion of sets of edges as operation and restrict it to sets of $\mathcal{O}(\log n)$ many edges. We assume that a linear order and its corresponding addition and multiplication relations are given as "built-in relations", since they cannot be computed incrementally as before. We make this assumption transparent and write DynFO$(+, \times)$ for the class of queries that are maintained by dynamic programs with access to built-in $\leq$, $+$ and $\times$.

The technique used for such operations can be understood as a *simulation of monadic second-order logic*. Monadic second-order logic MSO extends first-order logic by quantification over sets. The basic idea of the simulation is that a subset of a set of logarithmically many nodes can be encoded by one node, since a node basically corresponds to a bit string of length $\log n$. Therefore, set quantification over such small sets can be simulated by node quantification over the full graph. The connection between node subsets and nodes can be drawn with the help of the built-in linear order and the arithmetic relations $+$ and $\times$, as the bit string representation of a node can be expressed from them by first-order formulas, see [26, Theorem 1.17].

Sketch 4:

Log-size insertions to undirected graphs

It turns out that reachability on undirected graphs can be maintained under single-edge deletions and insertions of $\mathcal{O}(\log n)$ edges.

Proposition 2.3. Reach *on undirected graphs is in* DynFO$(+, \times)$ *under single-edge deletions and insertions of* $\mathcal{O}(\log n)$ *edges, where* n *is the number of nodes of the graph.*

We use the spanning forest approach as presented in Sketch 2, and re-use its maintenance rules after single-edge deletions. We describe how the auxiliary relations can be updated after $\log n$ many edges are inserted into a graph G. A corresponding update rule may use first-order quantification on the graph and set quantification over a subgraph of size $\mathcal{O}(\log n)$. As sketched above, this MSO quantification can actually be simulated by first-order update rules.

As Reach is MSO-expressible, an update rule can express for each pair a, b of nodes that are affected by the change (i.e. that are adjacent to a new edge after the change) whether they are connected via already existing paths and newly inserted edges. The spanning forest is then updated as follows: a newly inserted edge (a, b) becomes a spanning forest edge if a and b are not connected in the graph which consists of all previously existing edges and all new edges that are lexicographically smaller than (a, b). The in-betweenness relation can be updated similarly in a straightforward fashion, since a node b is between a and c in the new spanning forest if a and b as well as b and c are connected, but a and c become disconnected without b.

As a matter of fact, reachability on *directed graphs* can also be maintained under such insertion operation, at least in the absence of deletions.

The previous example naturally leads to the question of which sizes of bulk changes can be handled by a dynamic program for reachability. It turns out that the reachability query cannot be maintained under changes of more than polylogarithmically many edges.

Sketch 5:

An impossibility result for bulk changes

It turns out that classical lower bound results for the size of AC^0-circuits almost immediately yield upper bounds for the sizes of bulk changes that can be handled. Here, an AC^0-circuit for an input of size n may use polynomially many $\vee$-, $\wedge$-, and $\neg$-gates (with possibly unbounded fan-in) arranged in a circuit of constant depth.

The idea is simple. A classical result by Smolensky states that for computing the parity of the number of ones occurring in a bit string of length n, an AC^0 circuit of depth d requires $2^{\Omega(n^{1/2d})}$ many gates (see [27, Theorem 12.27] for a modern exposition). A simple, well-known reduction yields that deciding reachability for graphs with n edges which are disjoint unions of (undirected) paths also requires AC^0 circuits of size $2^{\Omega(n^{1/2d})}$. Indeed, computing the parity of the number of ones in $w = a_1 \cdots a_n$ can be reduced to

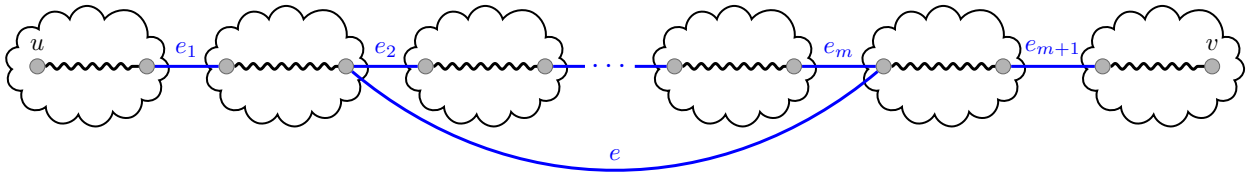


Figure 3: Illustration of the observation that if, after a first-order defined insertion, a path in a graph uses many new edges $e_1, \ldots, e_{m+1}$, there must exist a shortcut via a new edge e with fewer new edges.

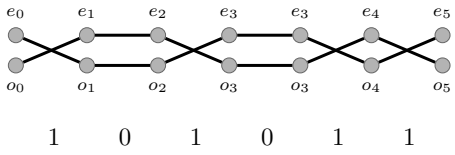
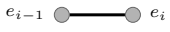
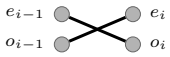


Figure 4: Illustration of the reduction from parity to reachability for the string $w = 101011$.

reachability as follows. A graph G for w can be constructed by converting each bit a_i into a small widget W_i with nodes $e_{i-1}, o_{i-1}, e_i, o_i$ as follows:

- if $a_i = 0$ then $W_i =$ 
- if $a_i = 1$ then $W_i =$ 

Now, for a bit string $w = a_1 \ldots a_n$ there is a path from e_0 to e_n iff w has an even number of ones, see Figure 4 for an example. Furthermore, graphs obtained in this fashion are disjoint unions of two paths.

This lower bound translates into a lower bound for first-order formulas via the correspondence of AC^0 and first-order logic due to [2].

Theorem 2.4. *Let $f(n) \in \log^{\omega(1)} n$ be a function from $\mathbb{N}$ to $\mathbb{N}$. There is no first-order formula with access to built-in relations that defines reachability in graphs with at most $f(n)$ edges, even for disjoint unions of (undirected) paths.*

Since from any formula that updates the result of a query after an insertion of $f(n)$ tuples into an initially empty input relation one can construct a formula that defines the query for inputs of size $f(n)$, the following corollary is immediate [9, Corollary 2].

Corollary 2.5. *Let $f(n) \in \log^{\omega(1)} n$ be a function from $\mathbb{N}$ to $\mathbb{N}$. Then reachability (even in disjoint unions of (undirected) paths) cannot be maintained in* DynFO *for bulk changes of size up to $f(n)$, even if the auxiliary relations may be initialised arbitrarily.*

We have seen how to maintain reachability under $\mathcal{O}(\log n)$ edge insertions, and that dynamic first-order programs cannot maintain reachability under insertions of more than a polylogarithmic number of edges. For reachability on undirected graphs this gap can be closed: this query can be maintained under insertions and deletions of polylogarithmic size.

Sketch 6:

Polylog-size changes to undirected graphs

To maintain reachability in undirected graphs under edge changes of polylogarithmic size, the technique of Sketch 4 can be extended. There, we used simulations of MSO formulas on subgraphs of logarithmic size. We do not know whether such simulations are also possible for subgraphs of polylogarithmic size, but we observe next that on subgraphs of this size, NL-computations can be simulated.

First of all, it can be observed that Reach over subgraphs of polylogarithmic size can be expressed by a first-order formula over the whole graph. This follows from the well-known result (see for example [5, p. 613]) that for every $d \in \mathbb{N}$ there is a uniform circuit family for computing the transitive closure of a graph with N nodes using circuits of depth $2d$ and size $N^{\mathcal{O}(N^{1/d})}$. If the subgraph in question has $N \stackrel{\text{def}}{=} \log^c n$ nodes, for some $c \in \mathbb{N}$, we can choose $d \stackrel{\text{def}}{=} 2c$, and the circuit size

$$N^{\mathcal{O}(N^{1/d})} = (\log^c n)^{\mathcal{O}((\log^c n)^{1/d})} = (\log n)^{\mathcal{O}((\log n)^{c/2c})}$$
$$= 2^{\mathcal{O}(\log\log n \sqrt{\log n})} \subseteq 2^{\mathcal{O}(\log n)} = n^{\mathcal{O}(1)}$$

is polynomial in n. This uniform AC^0 circuit family computing reachability for subgraphs of size $\log^c n$ can be turned into an FO(+,×)-formula thanks to [2]. Since Reach is complete for NL under first-order reductions, see [26, Theorem 3.16], first-order logic can thus express all NL-computable queries on graphs of polylogarithmic size.

Now we can sketch the proof idea of the following result [9, Theorem 6].

Theorem 2.6. Reach *on undirected graphs is in* DynFO$(+, \times)$ *under insertions and deletions of*

$\log^c n$ *many edges, for every fixed* $c \in \mathbb{N}$. *Here,* n *is the number of nodes of the graph.*

Again, we employ the spanning forest approach from Sketch 2. When a polylogarithmic number of edges is inserted into a graph, the update rule defines a spanning forest on the at most polylogarithmic number of connected components that get connected by this change, which is possible in first-order logic by NL-simulation as explained above. For each edge in this spanning forest, the lexicographically smallest edge between corresponding components is selected to become part of the spanning forest of the whole graph. The in-between relation is updated accordingly by combining the auxiliary information with in-betweenness information for the spanning forest on the connected components, which again can be expressed directly in first-order logic.

The update after a deletion of polylogarithmically many edges is not much harder. In a first step, the edges are deleted from the spanning forest, and its in-between information is adjusted. Only a polylogarithmic number of connected components of the spanning forest are affected by this step. For them, the update rule checks in a second step whether they can be re-connected by existing non-spanning-tree edges of the graph. This step works exactly as the update for edge insertions.

Just as for Sketch 4, Reach on directed graphs can be maintained under insertions of polylogarithmically many edges, with similar techniques (but in the absence of edge deletions).

2.2 Current frontiers of directed reachability

Turning to directed graphs, we first give a glimpse of an idea how reachability can be maintained under single edge insertions and deletions.

Sketch 7:

Single-edge insertions and deletions in directed graphs

The long-standing question [7, 14, 12, 23, 25, 35, 45] whether reachability on general directed graphs is in DynFO was settled in [8].

Theorem 2.7. Reach *is in* DynFO *under insertions and deletions of single edges.*

The underlying idea is to first reduce the reachability query to a linear-algebraic problem, and then to show that this problem can be maintained with first-order update rules. This approach works if DynFO is closed under the applied reductions it uses, which is guaranteed if they obey two conditions: that they are definable in first-order logic and that one change in the source structure only induces $\mathcal{O}(1)$ changes in the target structure. Such *bounded first-order (bfo) reductions* were introduced in [35].

Step 1: Reduction to FullRank.

Problem:	FullRank
Input:	An $n \times n$-matrix C
Question:	Is the rank of C equal to n?

This step is very similar to reductions used by Cook (for studying the NC-hierarchy) and Laubner (for studying extensions of first-order logic by linear-algebraic operators) [6, 30]. To facilitate subsequent generalisations, we describe the reduction to FullRank by two reductions with another intermediate problem. We defer to [8] for further details.

Suppose that A is the adjacency matrix of a graph G. The number of paths of length i from s to t in G corresponds to the value of the s-t-entry of A^i, the i-th power of the adjacency matrix. The matrix $I - \frac{1}{n}A$ is invertible (since diagonally dominant) and its inverse can be written, analogously to standard geometric series, as:

$$(I - \frac{1}{n}A)^{-1} = \sum_{i=0}^{\infty} \frac{1}{n^i} A^i$$

Hence, there is a path from s to t if the s-t-entry of the inverse of $C \stackrel{\text{def}}{=} I - \frac{1}{n}A$ is not zero. This yields a bfo-reduction from Reach to the problem MatrixInverse$^{\neq 0}$, which we define as:

Problem:	MatrixInverse$^{\neq 0}$
Input:	Invertible $n \times n$-matrix C; $s, t \leq n$
Question:	Is the s-t-entry of C^{-1} not 0?

The problem MatrixInverse$^{\neq 0}$ can then be reduced to FullRank: by Cramer's rule, an entry of a matrix C^{-1} is non-zero if and only if the determinant of some submatrix of C is non-zero, which is equivalent to the question whether this submatrix has full rank. We refer to [8] for details and for a verification that the reductions are actual bfo-reductions.

Step 2: Maintaining FullRank. Now our goal is to update whether a matrix C has full rank under changes of one entry. This can be done similarly as described in [19]: we maintain matrices B, E such that $BC = E$, B is invertible, and E is in reduced row-echelon form. It turns out that, modulo small primes, the matrices B and E can be updated using a constant number of simple matrix operations under changes of single entries of C. Reachability can then ultimately be maintained by maintaining the full rank property for a suitable number of such small primes.

This result has since been simplified and improved. The following technique for reducing the conceptual re-

quirements for maintaining a query has been useful for this purpose, and it has also been applied to show a number of other maintenance results.

Sketch 8:

The muddling technique

The *muddling technique* exploits that under certain conditions it suffices to maintain the result of a query for polylogarithmically many change steps, as opposed to arbitrarily many. In the following, we only consider queries that are *domain independent* in the sense that the query result does not change for an instance when additional, isolated elements are added to the domain. A query is $(\mathrm{NL}, f(n))$-*maintainable*, if it can be maintained for $f(n)$ change steps starting from an arbitrary database instance and auxiliary data initialised by an NL computation.

THEOREM 2.8 (MUDDLING LEMMA [10, 11, 40]). *Let $\mathcal{Q}$ be a domain independent query that is $(\mathrm{NL}, \log n)$-maintainable under some set Δ of change operations.*[8]

a) If Δ is a set of single-tuple change operations then $(\mathcal{Q}, \Delta)$ is in DYNFO.

b) If $\mathcal{Q} \in \mathrm{NL}$ and Δ is a set of bulk change operations of size at most $\log^d n$, for an arbitrary $d \in \mathbb{N}$, then $(\mathcal{Q}, \Delta)$ is in $\mathrm{DYNFO}(+, \times)$.

As an example, the Muddling Lemma allows to prove that a query is in DYNFO by showing that it can be maintained for $\log n$ many steps, starting from an arbitrary graph G with n nodes, with the help of auxiliary relations that can be obtained from G by some NL computation.

A result that highlights the power of this technique is that all queries expressible in monadic second order logic are in DYNFO under changes of single tuples, if the database (always) has bounded treewidth [10]. We do not know how to maintain a tree decomposition with first-order formulas, yet the muddling lemma allows to pre-compute a tree decomposition in LOGSPACE. It can be shown that a query result can then be maintained for $\log n$ steps.[9]

For the reachability query on undirected graphs, the maintenance strategy could be lifted from single-edge changes to changes of polylogarithmic size. It is an immediate question to what extent this is possible for directed graphs.

We recall that one can maintain REACH under insertions of polylogarithmic size with the techniques of Sketch 6, but this result does not allow for any edge deletions. It turns out that bulk insertions *and* deletions are indeed possible, but the allowed number of changed edges so far falls short of $\log n$.

Sketch 9:

Almost $\log n$ insertions and deletions into directed graphs

We now show how the approach of Sketch 7 can be adapted such that, using the muddling technique, reachability on directed graphs can be maintained under a non-constant number of edge insertions and deletions [11, Theorem 1].

THEOREM 2.9. REACH *is in* $\mathrm{DYNFO}(+, \times)$ *under insertions and deletions of edges that affect $\mathcal{O}(\frac{\log n}{\log \log n})$ nodes, on graphs with n nodes.*

In Sketch 7 we explained how REACH can be reduced to $\mathrm{MATRIXINVERSE}^{\neq 0}$ and to FULLRANK, and how to maintain the latter. Here, the idea is to maintain $\mathrm{MATRIXINVERSE}^{\neq 0}$ directly, by maintaining (sufficient information on) the inverse C^{-1} of the input matrix C. Therefore, our goal is to update the inverse C^{-1} when C changes to some matrix $C + \Delta C$.

Suppose that ΔC is a change matrix that encodes edge insertions and deletions that affect $k \stackrel{\text{def}}{=} \frac{\log n}{\log \log n}$ nodes of the graph. Then, ΔC has at most k non-zero rows and columns, and can be written as a matrix product $\Delta C = UBV$ where B has dimension $k \times k$. The update of C^{-1} to $(C + \Delta C)^{-1}$ with $\Delta C = UBV$ is described by the Sherman-Morrison-Woodbury identity (cf. [24]) as

$$(C + \Delta C)^{-1} = (C + UBV)^{-1} = C^{-1} - C^{-1}U(I + BVC^{-1}U)^{-1}BVC^{-1}.$$

To implement the right-hand-side of this identity as a dynamic program with first-order update rules, some obstacles have to be eliminated. First, literally computing the identity is not possible in first-order logic, since entries in C^{-1} can be exponentially large, and multiplying such numbers is not possible with first-order formulas even in the presence of arithmetic on the domain. A workaround is to compute C^{-1} modulo polynomially many, polynomially bounded primes: an entry of C^{-1} is non-zero if and only if it is non-zero modulo one of the primes.

Since $I + BVC^{-1}U$ is a $k \times k$ matrix, its inverse can be computed in AC^0 over $\mathbb{Z}_p$, for every prime p that is polynomially bounded in n — if it is invertible. However, although the occurring matrices are

[8] The result actually even holds for $(\mathrm{AC}^c, \log^c n)$-maintainable queries, for an arbitrary $c \in \mathbb{N}$, and the proof uses the fact that AC^c-circuits correspond to fixed-point computations with $\mathcal{O}(\log^c n)$ iterations, cf. Section 5 in [26].

[9] In fact, for the MSO result an *annotated* tree decomposition is needed and therefore a stronger version of the Muddling Lemma is used.

all invertible over $\mathbb{Q}$, they may not be invertible over $\mathbb{Z}_p$ for some primes p. If this is the case for a prime, the auxiliary relations for this prime become invalid and cannot be used any more. But thanks to the muddling technique it suffices to maintain the query for a polylogarithmic number of change steps, and it is possible to guarantee that a sufficiently large number of primes survives for that many rounds, to get the final result.

It is an open question whether reachability on directed graphs can be maintained under insertions and deletions of logarithmically or even poly-log many edges using first-order update rules. By allowing update rules from stronger logics than first-order logic, this becomes possible: with additional majority quantifiers one can maintain reachability on all directed graphs under changes of poly-log size [11]; for certain classes of directed graphs, additional parity quantifiers are sufficient [9].

Sketch 10:

Regular path queries

Attentive readers might have observed a gap in our reasoning, as presented so far: our motivating scenario involved graph databases and regular path queries but throughout this article, we studied mere reachability queries on graphs without edge labels. However, it turns out that the maintainability of the latter is actually the key for maintaining regular path queries (and then conjunctive regular path queries and unions therefore, and so on). This is because the evaluation of a regular path query can be reduced to the reachability query in a very simple fashion [28].

Indeed, this is doable by considering the product of the actual graph with an automaton for a regular language. More precisely, if A is an NFA that decides the regular language R underlying the regular path query at hand, and if D is a graph database with edges labelled by the alphabet used by A, then the question whether there is an R-path from u to v boils down to the question whether the node (s_f, v) is reachable from the node (s_0, u) in the synchronised product $A \times D$. The nodes of that product are pairs (s, w) of a state of A and a node from D and there is an edge from (s_1, w_1) to (s_2, w_2) if, for some symbol a, there is an a-transition from s_1 to s_2 in A and an a-labelled edge from w_1 to w_2 in D. Furthermore, s_0 and s_f are the unique initial and final states of A, respectively.

This reduction is actually a bounded-first order reduction, since each single change in D only induces at most size(A) many, first-order definable, changes in $D \times A$. Therefore, maintainability of Reach on directed graphs yields maintainability of the R-path query on graph databases, for every R. As an example, we get the following corollary from Theorem 2.9.

Corollary 2.10. *Let $\mathcal{Q}$ be a regular path query. Then $\mathcal{Q}$ is in* DynFO$(+, \times)$ *under insertions and deletions of edges that affect $\mathcal{O}(\frac{\log n}{\log \log n})$ nodes, where n is the number of nodes of the graph database.*

3. QUERY MAINTENANCE BARRIERS

First-order update rules are surprisingly powerful. Above, we explored the reachability query and saw that it can be updated with such rules, even in cases, where complex changes are allowed. Also the tree isomorphism query [17], all MSO queries on bounded tree-width graphs [10], and all context-free languages [22] can be maintained in DynFO.

This leads to the natural question: Is there a barrier for the power of dynamic programs, besides the easy observation that all queries in DynFO are computable in polynomial time? Proving such barriers is a challenging task already in static settings, and it is therefore not surprising that so far there are only preliminary answers. Much of the work on barriers for dynamic programs was done in the quest of finding out whether reachability is in DynFO. For this reason we focus on results that establish barriers for updating reachability in scenarios with restricted resources such as small auxiliary data and restricted updated rules.

While we can rely on several methods for proving barriers of inexpressibility for first-order logic in static scenarios, our tool set for dynamic lower bounds is much less developed. Classical methods for static inexpressibility include Ehrenfeucht-Fraïssé games and locality-based arguments [16, 31] as well as circuit-based methods [27] that exploit the connection between first-order logic and constant-depth circuits. Parity (of a unary relation) and reachability are standard examples for queries, that are provably not expressible in first-order logic. Yet, both queries are contained in DynFO.

In the following, we outline two tools for dynamic lower bounds: (a) exploitation of static lower bounds and (b) a locality method for restricted update rules.

3.1 Exploiting static methods

Many non-maintainability results for DynFO were shown by contradiction with the help of known static lower bounds. More precisely, it was shown that if there was a dynamic program for a particular query $\mathcal{Q}_{\text{dynamic}}$, then some $\mathcal{Q}_{\text{static}}$ would be expressible in first-order logic, maybe in the presence of "helpful relations", contradicting known inexpressibility results.

After making the notion of *helpful relations* precise, we present two instantiations of this technique which were used to establish that queries cannot be maintained by first-order updates when the arity – and therefore the size – of auxiliary relations is restricted.

A query over schema τ is *definable with helpful relations* over schema τ_{help} if there is a formula φ over $\tau \cup \tau_{\text{help}}$ such that for each database $\mathcal{D}$ over τ there is a database $\mathcal{D}_{\text{help}}$ over τ_{help} such that evaluating φ on $(\mathcal{D}, \mathcal{D}_{\text{help}})$ yields the result of the query on $\mathcal{D}$.

A first set of non-maintainability results for DynFO with unary auxiliary relations can be derived from inexpressibility results for existential, monadic second-order logic EMSO. This logic extends first-order logic by existential quantification of unary relations. Roughly speaking, inexpressibility results for EMSO often transfer to inexpressibility results for first-order logic with unary helpful relations, which in turn allow proving barriers for DynFO with unary auxiliary relations.

Sketch 11:

Unary auxiliary relations do not suffice for Reach

EMSO formulas can not define the transitive closure of simple paths (e.g., implicitly in [18]). By basically the same arguments as in [18] it can be shown that, for each first-order formula $\varphi(x, y)$ and each large enough n, there is no tuple H of help relations, such that φ defines the transitive closure over E on (G, H), where G is just a simple path [14, Theorem 4.3]. This "static" inexpressibility result implies the following "dynamic" inexpressibility result.

Theorem 3.1 ([14]). Reach *is not in* DynFO *with only unary auxiliary relations and one binary auxiliary relation for storing the transitive closure.*

Towards a contradiction, suppose Reach can be maintained in DynFO under edge deletions with unary auxiliary relations in addition to the binary relation T for the transitive closure. Then the first-order update rule for edge deletions can be used to construct a formula φ that defines the transitive closure on simple paths, using unary help relations. Indeed, let G be a simple path on nodes $1, \ldots, n$. The help relations H can simply be chosen as the unary auxiliary relations used by the dynamic program for the cycle C that extends E by the edge $(n, 1)$. The formula φ results from the update rule for deletions, by replacing every atom $T(x, y)$ by $\top$, since the transitive closure relation of a cycle is the full binary relation. This yields the desired contradiction.

We next give an example for deriving dynamic inexpressibility results from circuit lower bounds.

Sketch 12:

An arity hierarchy for auxiliary relations

It is well-known that the parity of n bits cannot be computed by constant-depth circuits of polynomial size [1, 21]. A less known result by Cai [4] extends this to the presence of "helpful bits": given n bit strings of length n^6, a constant-depth circuit of polynomial size cannot compute the parity of each of these strings even with $n-1$ help bits (which may depend on the bit stings). This result translates to first-order logic where, roughly speaking, the help bits translate to helpful relations: there is a query over a $6k$-ary schema which cannot be expressed by a first-order formula with $(k-1)$-ary help relations, for all $k \in \mathbb{N}$. Again, a barrier for DynFO follows immediately.

Theorem 3.2 ([14, 15]). *Let $k \geq 2$. There is a query over a $(3k+1)$-ary schema which can be maintained in* DynFO *with k-ary auxiliary relations, but not with $(k-1)$-ary auxiliary relations. In particular,* DynFO *has a strict arity hierarchy.*

It is open whether DynFO has a strict arity hierarchy over a fixed schema.[10]

3.2 Locality methods for restricted update rules

The above techniques work for first-order update rules, yet only for restricted arities: for queries on graphs, we currently only know how to prove barriers with respect to unary auxiliary relations. We now present a technique that allows proving barriers for high-arity auxiliary relations, yet it can only be applied to quantifier-free update rules and slight extensions thereof.

Quantifier-free update rules might seem unreasonably weak, but it turns out that they are not entirely powerless. As an example, in Sketch 1 we showed how reachability on directed graphs can be maintained under edge insertions with quantifier-free update rules. Also, membership of strings in regular language can be maintained without quantifiers [22].[11]

The *Substructure Method* encapsulates the weakness of quantifier-free update rules as a technical lemma. We sketch it next and give three applications.

Sketch 13:

Barriers with the Substructure Method

The intuition of the Substructure Method is as follows: suppose $\mathcal{S}$ is the current state of a dynamic program, i.e., $\mathcal{S}$ is a structure consisting of the input database and the auxiliary database. When updating an auxiliary tuple $\vec{c}$ after modifying a tuple $\vec{d}$, a quantifier-free update rule only has access to $\vec{c}$ and $\vec{d}$.[12] Thus, if a sequence of modifications changes only tuples from a substructure $\mathcal{A}$ of $\mathcal{S}$, then the auxiliary

[10] Such a strict hierarchy has been established for update rules without quantifiers [43, 41].

[11] In fact, the regular languages can be *characterised* by this property.

data of $\mathcal{A}$ is not affected by information outside $\mathcal{A}$. In particular, two isomorphic substructures $\mathcal{A}$ and $\mathcal{B}$ remain isomorphic, when corresponding modifications are applied to them.

LEMMA 3.3 (SUBSTRUCTURE LEMMA [22, 44]). *Let $\mathcal{P}$ be a dynamic program with quantifier-free update rules and let $\mathcal{S}$ and $\mathcal{T}$ be states of $\mathcal{P}$ with isomorphic substructures $\mathcal{A}$ and $\mathcal{B}$, respectively. Then the substructures $\mathcal{A}$ and $\mathcal{B}$ are still isomorphic after applying isomorphism-respecting changes α to $\mathcal{A}$ and β to $\mathcal{B}$. In particular, if $\mathcal{P}$ has a Boolean relation Q for storing a query result, then Q has the same value in the resulting states.*

Now, to prove that a query $\mathcal{Q}$ cannot be maintained with quantifier-free update rules using the Substructure Method, one can proceed as follows. Assume, towards a contradiction, that there is a program for $\mathcal{Q}$. Then, find two states $\mathcal{S}$ and $\mathcal{T}$ of a dynamic program with two isomorphic substructures $\mathcal{A}$ and $\mathcal{B}$, respectively, such that applying two corresponding modification sequences α and β to $\mathcal{A}$ and $\mathcal{B}$ yields one structure $\mathcal{S}'$ in $\mathcal{Q}$ and one structure $\mathcal{T}'$ not in $\mathcal{Q}$. By the Substructure Lemma, this is a contradiction.

The challenge is to find suitable structures $\mathcal{S}$ and $\mathcal{T}$ for a query at hand. Several combinatorial techniques have been used for finding such structures for which we provide examples. The proof of the following result combines the Substructure Method with a simple counting argument [22, Proposition 6.2].

THEOREM 3.4. *Alternating reachability cannot be maintained with quantifier-free update rules.*

By combining Ramsey's Theorem and Higman's Lemma to find suitable structures with isomorphic substructures, a barrier for REACH can be shown, though only for restricted auxiliary relations [45, Theorem 4.7].

THEOREM 3.5. REACH *cannot be maintained with quantifier-free update rules and binary auxiliary relations.*

It is open whether REACH can be maintained with quantifier-free update rules under single edge modifications. However, combining the Substructure Lemma with upper and lower for Ramsey numbers, a technique introduced in [43], it can be shown that REACH cannot be maintained without quantifiers under moderate definable changes [39, Theorem 7.3].

THEOREM 3.6. REACH *cannot be maintained with quantifier-free update rules under changes defined by quantifier-free first-order formulas.*

4. SUMMARY AND FURTHER WORK

We have presented several DYNFO maintainability results for the reachability query, along with the techniques that are used to construct the corresponding dynamic programs. As discussed in Sketch 10, these results can readily be translated into maintenance results for regular path queries. Further DYNFO maintainability results, also for extensions of regular path queries, are given in [42, 32, 3].

Of course, the *lower bounds* from Section 3 directly hold for regular path queries. Apart from the barriers discussed there, some further challenges become visible when trying to construct dynamic programs for graph database queries.

We emphasise that the maintenance results for REACH do not imply that all NL-computable queries are in DYNFO, although REACH is NL-complete. This is because DYNFO is only (known to be) closed under *bounded* first-order reductions and REACH is provably not NL-complete under these reductions [35].

However, relatively easy graph queries can be shown to be NL-complete under bfo reductions, as for example the $(a[bc])^*$ *query* that selects all pairs (u, v) of nodes such that there is an a-labelled path from u to v, and every intermediate node on this path is the start of a path of length 2 labelled bc. This query can be defined via *nested regular expressions* [36], instead of regular expressions used to define regular path queries.

If the $(a[bc])^*$ query is shown to be in DYNFO under single-edge changes, then so are all NL queries. As there are queries with much smaller static complexity than NL which are not known to be in DYNFO [41], such a result seems unlikely. It is not even known whether the $(a[bc])^*$ query can be maintained when only insertions of single edges are allowed.

On a more technical note, we remark that the settings of [34] and [13] are slightly different, in that the latter allows to change the set of nodes of the graph. It turns out that this difference hardly matters for single-tuple changes and only mildly for more complex changes. Maintainability of queries usually coincides in both settings. We stuck here to the setting of [34], mainly because of its simplicity.

How to approach query maintenance in DynFO?

We have seen some queries that can be maintained in DYNFO and others where this question is open. How should one try to find out whether a given query Q is in DYNFO? Although there is no truly systematic approach to showing that a given query is in DYNFO, the following guiding questions can serve as a heuristic on how to start.

[12] In general, the database could also have some constants. But we assume here, that it does not, for simplicity.

(a) Is the static complexity of Q above NC?

- If this is the case, e.g. if Q is P-hard, the chances of successfully maintaining it are low.[13]

(b) Is the query hard under bfo reductions for some class $\mathcal{C}$ above AC^0, e.g. NL or LOGSPACE?

- If that is the case, it will likely still be difficult to show that Q is in DynFO, since that would imply that all queries from $\mathcal{C}$ are in DynFO.

(c) Otherwise, the methods described in this article might be successful. Probably it does not hurt to try the muddling technique first.

Future work

Some open questions that might be worth tackling are the following.

Open question 1. *Can reachability on directed graphs be maintained with first-order update rules under changes of polylogarithmic size?*

Open question 2. *Can minimal distances and witness paths between nodes of a graph be maintained with first-order update rules?*

Open question 3. *Can reachability be maintained with quantifier-free update formulas under single-edge deletions?*

Besides maintenance of reachability and related queries, other aspects of dynamic complexity have been studied as well. These include static analysis of dynamic programs [38] as well as connections to information extraction [20] and parameterised complexity [37].

Another exciting research question is to bridge the gap between dynamic complexity and dynamic algorithms: most of the above results are pure expressibility results and the dynamic programs are not very efficient with respect to their overall work. In future work we plan to investigate under which circumstances queries can be maintained in a work-efficient fashion.

5. REFERENCES

[1] M. Ajtai. Σ^1_1 formulae on finite structures. *Ann. of Pure and Applied Logic*, 24:1–48, 1983.

[2] D. A. M. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *J. Comput. Syst. Sci.*, 41(3):274–306, 1990.

[3] P. Bouyer and V. Jugé. Dynamic complexity of the Dyck reachability. In *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017*, pages 265–280, 2017.

[13] However, there are some rather artificial P-complete problems that are in DynFO [35].

[4] J. Cai. Lower bounds for constant-depth circuits in the presence of help bits. *Inf. Process. Lett.*, 36(2):79–83, 1990.

[5] X. Chen, I. C. Oliveira, R. A. Servedio, and L. Tan. Near-optimal small-depth lower bounds for small distance connectivity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 612–625. ACM, 2016.

[6] S. A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1-3):2–21, 1985.

[7] S. Datta, W. Hesse, and R. Kulkarni. Dynamic complexity of directed reachability and other problems. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Part I*, pages 356–367, 2014.

[8] S. Datta, R. Kulkarni, A. Mukherjee, T. Schwentick, and T. Zeume. Reachability is in DynFO. *J. ACM*, 65(5):33:1–33:24, 2018.

[9] S. Datta, P. Kumar, A. Mukherjee, A. Tawari, N. Vortmeier, and T. Zeume. Dynamic complexity of reachability: How many changes can we handle? In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, pages 122:1–122:19, 2020.

[10] S. Datta, A. Mukherjee, T. Schwentick, N. Vortmeier, and T. Zeume. A strategy for dynamic programs: Start over and muddle through. *Logical Methods in Computer Science*, 15(2), 2019.

[11] S. Datta, A. Mukherjee, N. Vortmeier, and T. Zeume. Reachability and distances under multiple changes. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, pages 120:1–120:14, 2018.

[12] G. Dong, L. Libkin, J. Su, and L. Wong. Maintaining transitive closure of graphs in SQL. *International Journal of Information Technology*, 51(1):46, 1999.

[13] G. Dong and J. Su. First-order incremental evaluation of datalog queries. In *Database Programming Languages (DBPL-4), Proceedings of the Fourth International Workshop on Database Programming Languages - Object Models and Languages*, pages 295–308, 1993.

[14] G. Dong and J. Su. Arity bounds in first-order incremental evaluation and definition of polynomial time database queries. *J. Comput. Syst. Sci.*, 57(3):289–308, 1998.

[15] G. Dong and L. Zhang. Separating auxiliary arity hierarchy of first-order incremental evaluation systems using (3k+1)-ary input relations. *Int. J. Found. Comput. Sci.*, 11(4):573–578, 2000.

[16] H.-D. Ebbinghaus and J. Flum. *Finite model theory*. Springer Science & Business Media, 2005.
[17] K. Etessami. Dynamic tree isomorphism via first-order updates. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS 1998*, pages 235–243, 1998.
[18] R. Fagin. Monadic generalized spectra. *Math. Log. Q.*, 21(1):89–96, 1975.
[19] G. S. Frandsen and P. F. Frandsen. Dynamic matrix rank. *Theor. Comput. Sci.*, 410(41):4085–4093, 2009.
[20] D. D. Freydenberger and S. M. Thompson. Dynamic complexity of document spanners. In *23rd International Conference on Database Theory, ICDT 2020*, pages 11:1–11:21, 2020.
[21] M. L. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.
[22] W. Gelade, M. Marquardt, and T. Schwentick. The dynamic complexity of formal languages. *ACM Trans. Comput. Log.*, 13(3):19, 2012.
[23] E. Grädel and S. Siebertz. Dynamic definability. In *15th International Conference on Database Theory, ICDT 2012*, pages 236–248, 2012.
[24] H. V. Henderson and S. R. Searle. On deriving the inverse of a sum of matrices. *Siam Review*, 23(1):53–60, 1981.
[25] W. Hesse. The dynamic complexity of transitive closure is in DynTC^0. *Theor. Comput. Sci.*, 296(3):473–485, 2003.
[26] N. Immerman. *Descriptive Complexity*. Springer, 1999.
[27] S. Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer Science & Business Media, 2012.
[28] D. Kähler and T. Wilke. Program complexity of dynamic LTL model checking. In *Computer Science Logic, CSL 2003*, pages 271–284, 2003.
[29] C. Koch, Y. Ahmad, O. Kennedy, M. Nikolic, A. Nötzli, D. Lupei, and A. Shaikhha. DBToaster: higher-order delta processing for dynamic, frequently fresh views. *VLDB J.*, 23(2):253–278, 2014.
[30] B. Laubner. *The structure of graphs and new logics for the characterization of Polynomial Time*. PhD thesis, Humboldt University of Berlin, 2011.
[31] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.
[32] P. Muñoz, N. Vortmeier, and T. Zeume. Dynamic graph queries. In *19th International Conference on Database Theory, ICDT 2016*, pages 14:1–14:18, 2016.
[33] M. Nikolic and D. Olteanu. Incremental view maintenance with triple lock factorization benefits. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018*, pages 365–380. ACM, 2018.
[34] S. Patnaik and N. Immerman. Dyn-FO: A parallel, dynamic complexity class. In *Proceedings of the Thirteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS 1994*, pages 210–221, 1994.
[35] S. Patnaik and N. Immerman. Dyn-FO: A parallel, dynamic complexity class. *J. Comput. Syst. Sci.*, 55(2):199–209, 1997.
[36] J. Pérez, M. Arenas, and C. Gutiérrez. nSPARQL: A navigational language for RDF. *J. Web Semant.*, 8(4):255–270, 2010.
[37] J. Schmidt, T. Schwentick, N. Vortmeier, T. Zeume, and I. Kokkinis. Dynamic complexity meets parameterised algorithms. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020*, pages 36:1–36:17, 2020.
[38] T. Schwentick, N. Vortmeier, and T. Zeume. Static analysis for logic-based dynamic programs. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015*, pages 308–324, 2015.
[39] T. Schwentick, N. Vortmeier, and T. Zeume. Dynamic complexity under definable changes. *ACM Trans. Database Syst.*, 43(3):12:1–12:38, 2018.
[40] N. Vortmeier. *Dynamic expressibility under complex changes*. PhD thesis, TU Dortmund University, Germany, 2019.
[41] N. Vortmeier and T. Zeume. Dynamic complexity of parity exists queries. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020*, pages 37:1–37:16, 2020.
[42] V. Weber and T. Schwentick. Dynamic complexity theory revisited. *Theory Comput. Syst.*, 40(4):355–377, 2007.
[43] T. Zeume. *Small Dynamic Complexity Classes: An Investigation into Dynamic Descriptive Complexity*, volume 10110 of *Lecture Notes in Computer Science*. Springer, 2017.
[44] T. Zeume and T. Schwentick. Dynamic conjunctive queries. In *Proc. 17th International Conference on Database Theory (ICDT 2014)*, pages 38–49, 2014.
[45] T. Zeume and T. Schwentick. On the quantifier-free dynamic complexity of reachability. *Inf. Comput.*, 240:108–129, 2015.

Making AI Machines Work for Humans in FoW

Sihem Amer-Yahia (CNRS, Univ. Grenoble Alpes, France), Senjuti Basu Roy (NJIT, USA), Lei Chen (HKUST, Hong Kong), Atsuyuki Morishima (Univ. of Tsukuba, Japan), James Abello Monedero (Rutgers University, USA), Pierre Bourhis (CNRS, CRIStAL, France), François Charoy (Univ. of Lorraine, INRIA, CNRS, France), Marina Danilevsky (IBM Research - Almaden, USA), Gautam Das (Univ. of Texas at Arlington, USA), Gianluca Demartini (Univ. of Queensland, Australia), Abhishek Dubey (Vanderbilt Univ., USA), Shady Elbassuoni (American Univ. of Beirut, Lebanon), David Gross-Amblard (Rennes 1 University, France), Emilie Hoareau (University Grenoble Alpes, France), Munenari Inoguchi (Univ. of Toyama, Japan), Jared Kenworthy (Univ. of Texas at Arlington, USA), Itaru Kitahara (Univ. of Tsukuba, Japan), Dongwon Lee (Pennsylvania State Univ., USA), Yunyao Li (IBM Research - Almaden, USA), Ria Mae Borromeo (UP Open Univ., Philippines), Paolo Papotti (EURECOM, France), Raghav Rao (Univ. of Texas at San Antonio, USA), Sudeepa Roy (Duke Univ., USA), Pierre Senellart (ENS, PSL University, France), Keishi Tajima (Kyoto Univ., Japan), Saravanan Thirumuruganathan (QCRI Qatar), Marion Tommasi (INRIA, France), Kazutoshi Umemoto (The Univ. of Tokyo, Japan), Andrea Wiggins (Univ. of Nebraska Omaha, USA), Koichiro Yoshida (CrowdWorks Inc., Japan)

1. OUR VISION

The Future of Work (FoW) is witnessing an evolution where AI systems (broadly machines or businesses) are used to the benefit of humans. Work here refers to all forms of paid and unpaid labor in both physical and virtual workplaces and that is enabled by AI systems. This covers crowdsourcing platforms such as Amazon Mechanical Turk, online labor marketplaces such as TaskRabbit and Qapa, but also regular jobs in physical workplaces. Bringing humans back to the frontier of FoW will increase their trust in AI systems and shift their perception to use them as a source of self-improvement, ensure better work performance, and positively shape social and economic outcomes of a society and a nation. To enable that, physical and virtual workplaces will need to capture human traits, behavior, evolving needs, and provide jobs to all. Attitudes, values, opinions regarding the processes and policies will need to be assessed and considered in the design of FoW ecosystems.

AI machines will become more specialized, more closely integrated and interoperable, and will automate many otherwise trivial tasks, as well as taking over more sophisticated functions that are currently done by humans only (e.g., onboarding and socializing). As intelligent systems are increasingly powerful and pervasive in augmenting, supporting, and sometimes replacing human work, making AI machines empower humans is necessary. This will leave workers with more time on exercising and refining human-specific skills, such as creativity and intuition and increasing the amount of specialized, highly-skilled work that they are able to handle by streamlining many supporting processes. This requires to rethink the design of FoW platforms to assist workers in continuously acquiring and improving skills through onboarding, upskilling and work delegation. Workers will take a more supervisory role, both over their work as well as the performance of AI machines that support their work, with their feedback providing corrective input that is used to continuously improve worker satisfaction and process performance.

2. INTELLECTUAL CHALLENGES

IC1: Capturing Human Capabilities.

In FoW, everyone can be a worker or an employer. Workers' perceptions of the fairness of recruitment, selection, allocation, and compensation processes will be crucial. Such perceptions must be measured to optimize not only the computational aspects of work, but also the human elements. This is a case where the measurement of key variables can be informed by social scientists and relevant theories, and put into practice by the computational communities.

New challenges at the crossroads of psychology,

social science, organization studies and computational solutions will arise. These include questions such as the degree to which the variables capturing perceived fairness and transparency affect the satisfaction of workers and employers across different types of work and different platforms? Which cultural backgrounds best predict individual work metrics, and which combinations of human traits are predictors of collaborative work [4, 17]?

Addressing these questions will require adapting organizational commitment frameworks to different work contexts [1]. In particular, a major research question concerns the validation of theories from traditional workplaces in virtual marketplaces. From a modeling and computational perspectives, we need to rethink storage structures to easily update human factors, job assignment algorithms by making them adaptive, and querying capabilities to extract human capabilities over time. Additionally, as the number of human factors that are relevant to optimization are latent, subjective factors such as motivation, collaborativeness are not easy to acquire and learn. Current models of consent to tracking are all-or-nothing and there may not exist a one-size-fits-all solution. Additionally, FoW design needs to account for legal and social expectations.

IC2: Stakeholder Requirements.

FoW platforms must allow the declarative specification of job-related and workforce-related requirements. For instance, employers can only partially specify which workers to hire for their jobs (in AMT, they can specify a threshold for acceptance ratio but no other conditions, in Qapa, they can specify skills, location and qualifications but they are limited to what the proposed form lets them specify). Workers cannot specify which employers they want to ban (a recurring discussion point on TurkerNation).

Employers need to specify (i) jobs, (ii) execution requirements, such as skills, knowledge, and experiences, and (iii) delivery requirements, such as deadlines. They should also be able to express complex jobs requiring coordination among workers [14]. They need tools to estimate the available workforce on platforms and to predict how commitment and quality level they can expect from potential workers for a given job. The diversity of jobs constitutes a challenge in those estimations and predictions. Moreover, it is sometimes difficult for employers to translate their needs into concrete job specifications. It may also happen that employers obtain unexpected outputs because of some ambiguity in job descriptions [25], in which case, automatic verification using previous practices and communication channels between employers and workers, must be leveraged.

Workers should be able to specify jobs they want [20] and express expected rewards, deadlines, and required skills. They may also rely on AI machines to request which knowledge and skills they can acquire through jobs, and what sequence of jobs they could complete to further their career.

Platforms should be able to specify how to match workers and jobs and manage immoral jobs [8]. Sometimes, such jobs are decomposed into smaller ones, so that each piece does not look inappropriate, and AI algorithms for analyzing relations between jobs posted on multiple platforms are needed.

IC3: Social Processes.

Digital labor platforms change the dynamic of employer-worker and worker-worker relationships by creating an anonymous mediation between them. This weakens traditional workplace relationships.

Worker-worker and worker-employer communication constitute the social life at work. Workers exchange information and discuss job opportunities. They discuss with employers for clarification, feedback and training. Improving their ability to communicate in the workplace is essential for the success of FoW. Given that different workplaces, be they physical or virtual, have different credential systems, managing the skills portfolio of workers is a key challenge. FoW platforms should help workers not only share a CV of their work (like [19]) but also transfer their portfolio in a trustworthy manner. Additionally, onboarding for newcomers could be fully automated through AI machines or enabled via the ability to ask questions to more experienced workers. Upskilling is at least as important as onboarding. This process could be realized by AI machines that determine tutorials suitable for precision learning but also arranging job allocations in sequences of increasing difficulty. Prior work in CSCW related to onboarding has shown that, for example, retention of new Wikipedia editors is impacted by welcome messages from humans but not from bots [15].

Workers and jobs. Current platforms provide sophisticated services for task assignment but very little for other dimensions of task management like task delegation, task abandonment or team formation to complete complex tasks requiring different skills. In FoW, workers should be able to delegate part of their jobs to other workers or to AI machines join or leave teams of workers as they see fit. Incentives for interoperability is a policy issue that we do not address. This could be done through the market

(as employers demand it) or through government (when major economy like EU creates regulations).

IC4: Platform Ecosystems.

Online labor markets are pervading every domain ranging from mobility (e.g. Uber), rental (e.g. Airbnb), food delivery (e.g. doordash), and freelance services (e.g. Fiverr, TaskRabbit). It is not possible for a single platform to support all these domains. Instead, due to specialized requirements there are different platforms for each domain. Within each domain and across domains, these platforms have to interoperate. That will enable different worker recruitment channels to reach diverse workers [11],[2]. At any time during job completion, AI machines should help workers if they wish to switch between tasks.

The technical challenges of interoperability include agreeing on schemas and APIs [10]. They should determine a class of interchangeable queries to exchange information on workers, employers and jobs. Such ecosystems would include (i) platforms where the actual work is performed, (ii) platforms similar to LinkedIn where workers can display their completed jobs along with credentials for skills to demonstrate their expertise, (iii) platforms for matching workers to jobs scattered across other platforms, and (iv) platforms that serve as an online water-cooler where workers negotiate for employment benefits.

IC5: Computation Capabilities.

The first computational challenge is to support the design of adaptive utility functions and evaluation mechanisms, for both workers and employers, that support a variety of work types: human services, human supervision, data analysis, content creation, etc. These utility functions capture the benefit of getting involved in a platform for workers, employers and platforms by modeling preferences and constraints. AI machines must help in refining them from worker activity and feedback by leveraging methods from game theory and active learning.

One needs to aim for a global optimization in the long term, taking into account the utility functions of workers, employers, and the platform itself (these utility functions potentially evolve over time). Such optimization will be concerned with monitoring and evaluating the long-term health of the ecosystem, and especially in detecting and addressing bad actors. Employers may harm the platform by contributing fake or malicious tasks; workers may make malicious contributions, intentionally or unintentionally; and even the platform itself may be guilty of bias in work assignment or validation. Identifying such potentially harmful actions will require advancements in signal identification, outlier detection, pattern mining, and techniques for natural language understanding. As new regulations come into being to address such bad actors, they will require the availability of detailed provenance information regarding job assignment, performance evaluation, and complaints, among others.

A central focus must be placed on efficient and incremental management of the creation, storage, access, and protection of the necessary data to enable platform computation while respecting all stakeholders' privacy and well-being. This requires to monitor and mine streaming data about workers continuously and provide provenance tracking to faithfully record who produced which data, what decisions were made. This data will be leveraged in adapting AI machines to evolving human traits and needs as well as for auditing and fairness purposes.

IC6: Benchmark and Metrics.

Benchmarks and metrics for FoW will need to be developed to measure the effectiveness of humans and work interaction at various stages such as the discovery, matching, and interaction of jobs and workers. We envision benchmarks that take social and computational criteria into the metric design. The *social criteria* include social impact, capital advancement, criticality, accessibility, and robustness, while the *computational criteria* include effectiveness and efficiency. In addition, these benchmarks should be able to assess the effectiveness of human-human, human-machine, and human-job interactions.

One of the challenges is to measure human factors, such as cognitive overhead reflecting how interested workers are in their jobs, or retention which indicates whether the jobs lead to boredom and fatigue. Designed metrics may cover one or more criteria. For example, precision and recall measure effectiveness, equity measures easy and universal access to employment for a wide range of users (including users with disabilities or without access to mobile phones), and criticality evaluates whether a job is time critical.

Developing benchmarks requires understanding the context of various job marketplaces by conducting extensive surveys, and generation of synthetic datasets that correctly reflect real-world applications. Additionally, benchmarks must cover diverse applications. They need to capture subjective human factors allowing deviations and reproducibility, sup-

porting interactions of humans with the available AI machines, and creation of *adversarial benchmarks* to evaluate the robustness of the platforms. Worker satisfaction must be assessed for continued participation of humans in the ecosystem.

IC7: Ethics.

Empowering workers and protecting their rights and privacy should be at the heart of FoW. This is a critical challenge since while platforms have a global reach, policies and regulations remain local for the most part. Advances in cybersecurity can be used to address privacy and access control mechanisms to guarantee that the right actors have visibility of the right data. Platforms should provide different privacy settings and be transparent about what worker data is exposed and to whom. Employers should be transparent about what the work is for, and how the work outcome will be utilized. They should also be able to protect their confidential information when needed. Fair compensation for workers, including base payments, bonuses, benefits and insurance should be ensured and regulated by law. Workers should have the freedom to choose the compensation type they deem acceptable. Finally, job allocation should be transparent, fair and explainable by design. Worker's sensitive attributes that might bias the job allocation process should be protected. Auditing mechanisms to ensure compliance with fair, transparent and explainable job allocation and compensation need to be developed and adopted.

In terms of fairness, an interdisciplinary approach will be required to develop novel methods to assess and quantify algorithmic fairness in job allocation practices. For example, looking at bias trade-offs between fully-algorithmic vs human-in-the-loop job allocation approaches where algorithmic bias could be different from implicit bias in humans. This will also result in higher levels of algorithmic transparency for job allocation where decisions should be easy to explain independently of whether they have been made by humans or by AI machines. Processes should be in place to specify how to best address unfair cases, e.g., by means of additional rewards for workers or novel/better job opportunities. We also envision novel methods to make job allocation distribution (i.e., the *long tail effect* where few workers complete most of the available jobs) and time spent on jobs more transparent to workers and external actors like compliancy agents. For example, visual analytics dashboards that communicate to workers how much time they spent and how much money they have earned on a platform with warnings on risks for addiction or unfair payments.

3. RELATED WORK

Kittur et al. discussed various challenges that prevent crowdsourcing from being a viable career [13]. This has inspired many follow-ups and there has been major upheaval in online labor marketplaces after [13] was published. The gig economy has become a major source of employment in various domains. Furthermore, [13] specifically focused on online paid crowdsourcing such as AMT. In contrast, our work casts a wider net. Our proposal affects not just a worker in AMT, a fully virtual marketplace, but also workers in virtual/physical labor markets such as Uber drivers and plumbers hired via TaskRabbit and Qapa.

Social Computing Positioning. Initial work [2010-2020] focused on obtaining reliable results from unreliable workers or developing algorithms for involving crowd workers on diverse tasks. Recently, there has been increasing interest in making crowdsourcing platforms a better place for both workers and requestors. A key issue in making crowdsourcing as a viable career is low pay that is often less than minimum wage in many jurisdictions. The work in [23] enables a simple way that allows a requester to pay minimum wage in AMT. IC7 discusses the issue of fairness from a wider lens beyond pay. The work in [12] surveys 360 workers and identifies the various techniques such as the usage of scripts and tools that workers use to increase their pay. IC4 discusses a working environment in the near future where AI agents act as worker surrogates to improve their experience.

Other examples in [7], [6], [5] seek to build frameworks that enable the use of crowds to solve heterogeneous tasks and optimize simultaneously for cost-quality-time. However, these are often skewed towards one stakeholder such as an employer or worker. In IC2, we identify mechanisms to obtain the requirements of all stakeholders that help in the design of equitable platforms. [22] and [19] propose alternate mechanisms for worker reputation. In IC4, we discuss a generic approach of platform ecosystems that allow a worker to seamlessly move between platforms. There has been a lot of work (e.g., [9]) on understanding the various factors affecting quality of work. Recent efforts such as [3] explore ways to improve worker's skill development through coaching while [21] discusses efficient mechanisms to teach crowd workers new skills. IC1 proposes mechanisms to capture skills (among other human factors) efficiently while IC3 talks about the challenges of upskilling. We advocate for a ma-

Data Modeling	IC1, IC2
Declarative Languages	IC1, IC2, IC5
Indexing	IC1, IC5
Data Integration	IC4
Recommendations	IC3
Data Mining	IC1
Optimization	IC3, IC4, IC5
Benchmarks	IC6
Transactions	IC4

Table 1: ICs and Data Management Challenges

jor change in how platforms are designed to enable these.

Data Management Positioning. Since FOW is more than just crowdsourcing, and much of the lower-level work will be done by AI, data management problems related to AI are a major part of our challenges [16]. Similarly, how to enable human-in-the-loop machine learning at scale and fully integrate it into business processes poses many data management challenges [24].

Table 1 summarizes core data management challenges and their relationship to our ICs. Prior works such as Deco or Qurk focused on cost based optimization for homogenous microtasks. While recent work such as Cioppino [7] generalize them to multiple heterogeneous tasks that run in parallel. SmartCrowd [18] takes human factors into account for optimization. One of the central challenges is building a FoW platform that is modular, extensible and efficient. It must be able to leverage data management techniques such as query optimization, indexing for speeding up the algorithms. Incorporating a diverse set of human and AI crowd workers requires a fundamental rethink of task assignment algorithms. Finally, it is important to develop quantitative benchmarks for each of the ICs so that the progress could be tracked.

The computational platforms necessary to support FoW will require distributed processing of transparent, and immutable time stamped records of transactional data. Blockchain technologies could enable the necessary computational artifacts to support monitor supply chains, payments processing, money transfers, reward mechanisms, digital IDs, data sharing and backup, copyrights and royalty protection, digital voting, regulations and compliance, workers rights, equity trading, management of accessible devices, secure access to belongings, etc.

As platforms become more specialized (example of CrowdFlower, a general-purpose platform that became Figure Eight, solely dedicated to data generation for AI), the trend of claiming to support one of the intellectual challenges we describe is going to increase.

4. REFERENCES

[1] P. J. Bateman, P. H. Gray, and B. S. Butler. Research note—the impact of community commitment on participation in online communities. *Information systems research*, 22(4):841–854, 2011.

[2] M. Brambilla, S. Ceri, A. Mauri, and R. Volonterio. Adaptive and interoperable crowdsourcing. *IEEE Internet Computing*, 19(5):36–44, 2015.

[3] C.-W. Chiang, A. Kasunic, and S. Savage. Crowd coach: Peer coaching for crowd workers' skill growth. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–17, 2018.

[4] L. Coursey, B. Williams, J. Kenworthy, P. Paulus, and S. Doboli. Diversity and group creativity in an online, asynchronous environment. *J. Creat. Behav*, 2017.

[5] D. Difallah, A. Checco, G. Demartini, and P. Cudré-Mauroux. Deadline-aware fair scheduling for multi-tenant crowd-powered systems. *ACM Transactions on Social Computing*, 2(1):1–29, 2019.

[6] K. Goel, S. Rajpal, and M. Mausam. Octopus: A framework for cost-quality-time optimization in crowdsourcing. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*, 2017.

[7] D. Haas and M. J. Franklin. Cioppino: Multi-tenant crowd management. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*, 2017.

[8] C. G. Harris. Dirty deeds done dirt cheap: a darker side to crowdsourcing. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 1314–1317. IEEE, 2011.

[9] K. Hata, R. Krishna, L. Fei-Fei, and M. S. Bernstein. A glimpse far into the future: Understanding long-term crowd worker quality. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 889–901, 2017.

[10] P. G. Ipeirotis and J. J. Horton. The need for standardization in crowdsourcing. In

Proceedings of the workshop on crowdsourcing and human computation at CHI, 2011.

[11] J. Jarrett and M. B. Blake. Interoperability and scalability for worker-job matching across crowdsourcing platforms. In *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 3–8. IEEE, 2017.

[12] T. Kaplan, S. Saito, K. Hara, and J. P. Bigham. Striving to earn more: a survey of work strategies and tool use among crowd workers. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*, 2018.

[13] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1301–1318, 2013.

[14] A. Kulkarni, M. Can, and B. Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the acm 2012 conference on computer supported cooperative work*, pages 1003–1012, 2012.

[15] J. T. Morgan and A. Halfaker. Evaluating the impact of the wikipedia teahouse on newcomer socialization and retention. In *Proceedings of the 14th International Symposium on Open Collaboration*, pages 1–7, 2018.

[16] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich. Data management challenges in production machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1723–1726, 2017.

[17] Y. Ren, J. Chen, and J. Riedl. The impact and evolution of group diversity in online open collaboration. *Management Science*, 62(6):1668–1686, 2016.

[18] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal*, 24(4):467–491, 2015.

[19] C. Sarasua and M. Thimm. Crowd work cv: Recognition for micro work. In *International Conference on Social Informatics*, pages 429–437. Springer, 2014.

[20] T. Schulze, S. Krug, and M. Schader. Workers' task choice in crowdsourcing and human computation markets. 2012.

[21] N.-C. Wang, D. Hicks, and K. Luther. Exploring trade-offs between learning and productivity in crowdsourced history. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–24, 2018.

[22] M. E. Whiting, D. Gamage, S. N. S. Gaikwad, A. Gilbee, S. Goyal, A. Ballav, D. Majeti, N. Chhibber, A. Richmond-Fuller, F. Vargus, et al. Crowd guilds: Worker-led reputation and feedback on crowdsourcing platforms. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1902–1913, 2017.

[23] M. E. Whiting, G. Hugh, and M. S. Bernstein. Fair work: Crowd work minimum wage with one line of code. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 197–206, 2019.

[24] D. Xin, L. Ma, J. Liu, S. Macke, S. Song, and A. Parameswaran. Accelerating human-in-the-loop machine learning: challenges and opportunities. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning*, pages 1–4, 2018.

[25] Y. Yamakata, K. Tajima, and S. Mori. A case study on start-up of dataset construction: In case of recipe named entity corpus. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3564–3567. IEEE, 2018.

Provenance in Collaborative in Silico Scientific Research: a Survey

Eduardo Jandre
Instituto de Computação
Universidade Federal Fluminense, UFF, Brazil
eduardojandre@id.uff.br

Bruna Diirr
Programa de Pós-Graduação em Informática
Universidade Federal do Estado do Rio de Janeiro, UNIRIO, Brazil
bruna.diirr@uniriotec.br

Vanessa Braganholo
Instituto de Computação
Universidade Federal Fluminense, UFF, Brazil
vanessa@ic.uff.br

ABSTRACT

Science is a collaborative activity by definition. Research is usually conducted by several scientists working together, and this behavior has been intensified in recent years. Furthermore, experiments are increasingly performed in silico, which demands proper support tools. Provenance-aware Workflow Management Systems and script-based tools have been popular ways of running in silico experiments, but these tools often neglect the collaboration aspect. Even solutions that aim at collaborative experiments do not always address the collaborators' needs. Literature shows surveys discussing subjects related to in silico experiments. However, they either focus on provenance collection and applications, thus treating collaboration as just another possible application, or focus on Workflow Management Systems, only listing collaboration as a possible challenge. This article surveys available tools and approaches that aim at aiding scientists to conduct collaborative in silico experiments. Particularly, we focus on challenges related to the provenance of these collaborative experiments. We devise a taxonomy with the aspects of collaboration in scientific research and discuss each of these aspects. We also identify literature gaps that provide future opportunities.

1. INTRODUCTION

Scientific knowledge is built incrementally and cumulatively. To discover something new, scientists have to extensively study their fields to understand the current state of the art. Additionally, an important part of the scientific process is the communication of the work done and the outcomes reached, which allows the scientific community to analyze and review other scientist's research and the obtained results. This process is essential because it allows other people to double-check the ideas, find flaws, or reproduce the achieved results, besides enabling the use of acquired knowledge in future discoveries [8]. Hence, collaboration plays a key role in scientific research and knowledge acquisition.

"Scientific collaboration can be defined as interaction taking place within a social context among two or more scientists that facilitates the sharing of meaning and completion of tasks with respect to a mutually shared, super-ordinate goal" [54]. Therefore, scientific collaboration occurs not only after the publication but especially in ongoing research. Research is usually carried out by several scientists working together. Indeed, collaboration is often encouraged and even required by research funding agencies [54].

Wuchty, Jones, and Uzzi [66] analyze almost 20 million publications from the mid-50s to the early 21st century, and conclude that the production of publications by teams of collaborators has increased over time and that these teams have grown in size. Also, the authors conclude that publications produced in teams usually receive more citations on average than publications made by a single author, even when self-citations are ignored [66].

At the same time, computer technology has advanced hugely. Computers have become cheaper and more accessible, and computer networks have spread all around the world. This movement produced two direct effects: (i) it allowed collaboration to occur not just between people nearby but also between people located all around the world; and (ii) it increased the number of scientific experiments conducted in silico.

In silico experiments typically demand more support from data management and software engineering tools when compared to other experiment classes (in vivo, in vitro, and in virtuo) [60]. Workflow Management Systems [3, 26, 67] and Script-based systems [17, 36, 47] (referred in this work as *Experiment Management Systems*) have been popular ways of running such experiments. However, collaboration is still one of the challenges in the area [16, 27, 31].

The data related to in silico experiments are not limited to the results of the experiment but also include the logical sequence of performed activities; parameters used; intermediary results of activities; information about the execution environment; etc. [25]. It is common for these data to be collected and stored in a provenance database. Provenance is a broad concept that can be applied in many disciplines and is usually linked to the origin of an object or data. It can be seen as a set of metadata that describes not only the object or data itself but also the activities applied in its production process. Bringing the concept into scientific research, it refers to information on how the experiment was performed and how the research results were recorded [31]. This should also include records of how the collaboration was conducted.

Provenance gathering is a common feature in many Experiment Management Systems [3, 17, 26, 31, 36, 47, 67]. However, when focusing on collaborative experiments, two challenges emerge: (C1) how to collect provenance in a collaborative experiment (this comprises collecting provenance of actions of scientists that may be working in different parts of the experiment or different geographical locations and machines); and (C2) how provenance can be used to make collaboration easier in this environment.

The main goal of this article is to map the state-of-the-art approaches and provenance-aware models that are available to conduct in silico collaborative experiments. We aim at investigating how they address challenges C1 and C2. To do so, we plan to answer the following research questions: (R1) How do existing tools store and collect provenance in a collaborative experiment?; (R2) how do existing tools use provenance to make collaboration easier in scientific experiments?. The research question R1 and R2 are respectively linked to challenges C1 and C2.

To answer these questions, we make a snowballing [30] based survey. We evaluate 170 publications and select 20 approaches and 7 surveys. To be selected, an approach has to satisfy the following criteria: (i) has collaboration as a focus (i.e., the problem to be solved or the subject of a survey); or (ii) has provenance as a focus while discussing collaboration features; and (iii) is in the context of in silico scientific experiments. The surveys were used to reinforce this work's motivation and as a benchmark. From the 20 selected approaches, 15 are tools for collaborative experiments, 2 are provenance-aware data models for collaborative experiments, and 3 approaches present both a tool and a provenance-aware data model for collaborative experiments.

This article contrasts with existing surveys [6, 16, 27, 31, 37, 51, 66] as follows. This work differs from Lu and Zhang's work [37] and Belloum et al. [6] by bringing a more detailed and up-to-date view of the work in the area. Besides that, Belloum et al. discuss the challenges to support e-science collaborative experiments with a closer look at the experiment life cycle, but it only addresses the tools provided by the VL-e project. Wuchty et al. [66] aim to demonstrate that teams have been increasingly dominating the scientific research in the production of knowledge, without addressing available tools and research that helps the execution of this type of experiment. On the other hand, Davidson and Freire [16] and Gil et al. [27] focus on the challenges and opportunities existing in the Workflow Management Systems research, without detailing the available tools. Other publications focus on provenance collection and its applications, and collaboration merely appears as one of the possible applications of provenance [31, 51]. As opposed to that, this survey focuses on provenance-related aspects of collaboration.

The article proceeds as follows: Section 2 presents an analysis of the existing provenance models that aim to precisely represent collaborative research; Section 3 discusses some aspects of collaborative research and proposes a taxonomy to capture the aspects that may influence collaboration in the scientific research scenario; Section 4 discusses publications and opportunities in the field; and Section 5 concludes the article.

2. PROVENANCE MODELS

Provenance is a broad concept and can be seen from different perspectives. Ragan et al. [51] classify provenance in five types: *Data provenance* (the history of changes and movement of data); *Visualization provenance* (the history of graphic views and visualization states); *Interaction provenance* (the history of user interaction with a system); *Insight provenance* (the history of cognitive outcomes and information derived from the analysis process); and *Rationale provenance* (the history of reasoning and intentions behind decisions, hypotheses, and interactions) [51].

Collaboration brings additional challenges in collecting and storing provenance. The first challenge (C1) resides in how to collect provenance in a collaborative experiment. It involves collecting data, interaction, and visualization provenance from multiple devices since scientists usually work on their workstation. Few initiatives capture provenance from multiple devices [18, 20, 64], but they usually

focus in high-performance settings, where a single user executes parts of the experiment in the cloud, cluster, or grid. This is different from having several scientists working on their local workstations, where there is usually no central control. Collecting this provenance could be useful in several situations, such as giving credit to those involved in the research [31], auditing the research, enabling the reproducibility of the experiment and providing relevant information that allows each member of a group to better understand the actions of other members in the context of a collaborative scientific experiment. Another challenge (C2) resides in how to use this provenance to make collaboration easier in a collaborative environment.

The first step to overcoming these challenges is providing a provenance model that can properly represent the research collaboration aspects. This model needs to represent four main aspects [37]: (i) Distribution (D) – Collaboration typically involves resources from multiple organizations; (ii) Heterogeneity (H) – Provenance produced by different workflows may have different formats. Even those that conform to the same schema may evolve during the experiment life cycle; (iii) Multilevel (M) – Experiments usually have complex tasks that are modeled hierarchically (e.g., using sub-workflows, or by functions calling functions in a script). Although this is not a specificity of collaborative experiments, the provenance model should store this hierarchy; (iv) Collaboration (C) – The model must support new user iterations and collaboration standards, besides storing information about these collaborations.

The term collaborative workflow has been used with multiple meanings in the literature. It is understood both as the *collaboration between workflows* or the *collaboration between workflow users* [37]. Collaboration between workflow users is the direct collaboration of users in the context of a scientific workflow. On the other hand, a collaboration between workflows refers to the indirect use of data produced by another workflow. This suggests an implicit collaboration, when collaboration occurs through the data published by another researcher.

Altintas et al. [1, 2] propose the provenance model shown in Figure 1, which is capable of capturing *implicit collaborations* within a scientific experiment. The model predicts the identification of workflows dependency from the relations between dataflows input and output, and also helps to identify contributions from users who collaborate on a project based on records of past executions. The authors extend OPM (Open Provenance Model) [44]

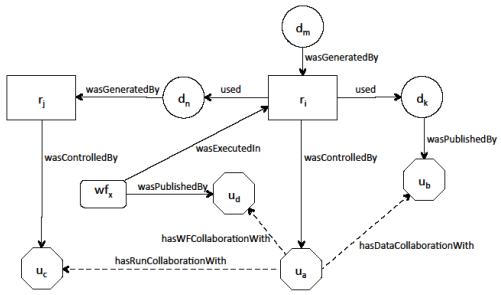


Figure 1: An abstract model of collaborative provenance nodes and dependencies using the extended Open Provenance Model [2]

to record user interactions when publishing data and workflows, which is essential for identifying the various types of user collaboration. This model explicitly represents *collaboration amongst users* (agents u_i in the figure) and which users were responsible for each run of the experiment (r_i in the figure). According to Ragan et al.'s classification [51], it captures *data* and *interaction* provenance. The approach also proposes a query language, which is an extension of the QLP (Query Language for Provenance) [5].

Missier et al. [43] propose a model that facilitates the sharing of provenance in collaborative environments. The model aims to provide end-to-end support for *implicit collaborations*. The approach treats sharing as an action from which provenance has to be preserved, i.e., the focus is to register the provenance of the data sharing process. To do so, the model adds new information to provenance traces, *stitching* common parts of those traces. With this, the model can represent cases when scientists use data that was produced by another scientist's workflow, even when they come from heterogeneous workflow systems. This model can represent *data* and *interaction* provenance [51].

Zhang et al. [68], Confucius [70], and ProvDB [41] present provenance models and tools that track collaboration provenance. Zhang et al. [68] propose the Collaborative Provenance Model (CPM), which is an extension of PROV-DM (PROV Data Model) [45]. Figure 2 shows that the model explicitly represents *Person* and *Group* of *Person* (a collaborating group), besides versions of *Workflow*, *Processor*, and *Data Links*. It also captures which user operates which workflow version, process version, and data link version. The model captures *data* and *interaction* provenance [51].

Confucius [70] introduces a provenance ontology (Figure 3). The ontology aims at supporting the capture and record of scientific workflow composition and user interactions during the process of

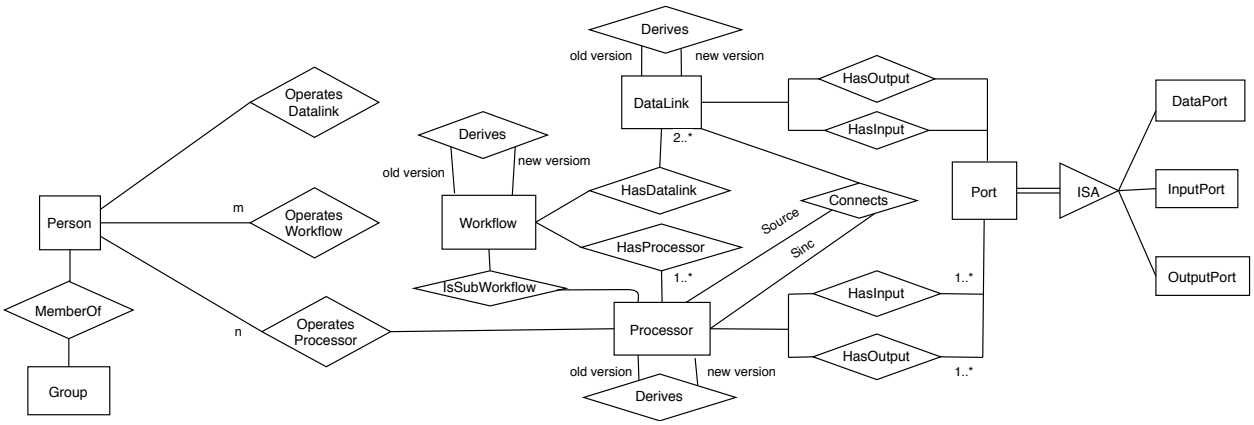


Figure 2: Collaborative Provenance Model (CPM) [68]

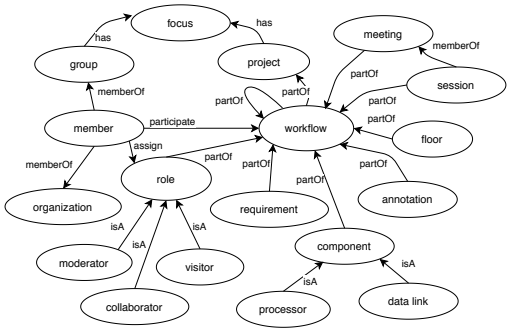


Figure 3: Collaborative workflow composition provenance ontology [70]

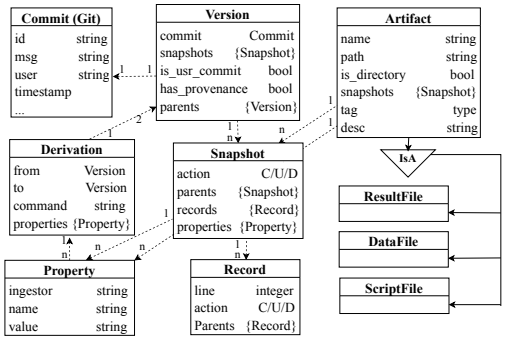


Figure 4: ProvDB Conceptual Data Model [41]

a collaborative workflow composition. The provenance is stored in a provenance repository on the central node of Confucius. Note that the ontology can represent workflows and their components, and roles of people in the collaboration groups. As for Ragan et al.'s classification [51], this model can represent *data* and *interaction* provenance, besides the remaining types through *annotations*.

ProvDB [41] proposes a provenance model with a schema-later approach, providing a base schema that can be extended by arbitrary properties as key-value pairs (Figure 4). Note that these values can be complex, such as a JSON document. The information to the base schema is collected through Git and the built-in ingestors, and additional information can be added through custom ingestors or by user's annotations. When the user runs a command using ProvDB, the system verifies the registered ingestors and executes them. The ingestors can analyze the before- and after-state of the artifacts produced by the command to generate provenance information about the executed command. The model deals with *data* and *interaction* provenance [51] and can deal with all other types of provenance using the ingestors.

Table 1 summarizes how each model supports the collaboration aspects mentioned at the beginning of this section. All the models present limitations when representing some aspects of collaboration. Altintas et al. [1, 2] present a model capable of capturing user collaborations but lack support for the other analyzed items. Confucius [70] and CPM [68] do not adequately treat the heterogeneity of collaboration, not being able to deal with different workflow formats. Confucius also does not deal with workflow evolution. Missier et al. [43] present limitations in dealing with workflow evolution and representing the multilevel hierarchy. ProvDB [41] is the only one providing support for all the analyzed aspects, but it does that making use of extended properties in a key-value schema. Regarding Ragan et al.'s [51] classification, only Confucius and ProvDB can capture all types of provenance, but they do that by using annotations or extended properties. This kind of schema could make things hard and inefficient to query. Another important aspect

Table 1: Summary of the Collaborative Provenance Models

Provenance Model	Provenance Types [51]	Aspects of Collaboration			
		D	H	M	C
Altintas et al. [1, 2]	Data; Interaction	No	No	No	Yes
CPM [68]	Data; Interaction	Yes	Evolution Only	Yes	Yes
Missier et al. [43]	Data; Interaction	Yes	Different schema only	No	Yes
Confucius [38, 61, 67, 70]	All*	Yes	No	Yes	Yes
ProvDB [41]	All*	Yes*	Yes*	Yes*	Yes

*Modeled as extended properties

is that the models just provide a form of storing the information generated in collaborative research and do not necessarily provide a way of collecting them. We also notice that the models supported by a tool [41, 68, 70] can store some provenance on collaboration, but the tool may not fully capture it.

In this section, we show several provenance models that are able to store in part (or in total) collaboration aspects of scientific experiments. However, in order to properly answer our two research questions, we need more insights. In the next section, we discuss how the existing approaches capture and use this information to foster collaboration.

3. COLLABORATION IN SCIENTIFIC RESEARCH

Scientific research is a complex activity per se, and collaboration in this environment becomes a challenging task. To better understand these challenges, we independently analyze the aspects that may influence collaboration in the scientific research scenario. We develop a taxonomy (Figure 5) by examining the 20 approaches we selected, capturing, and categorizing their similarities and differences. We then standardize and enrich the categorization based on other publications [39, 50, 53].

The first branch of the taxonomy is *Experiment Phases*, which is defined in different ways by different authors [6, 39]. In this survey, we use the classification proposed by Mattoso et al. [39], where scientific experiments go through three phases: *composition*, *execution*, and *analysis*. During *composition*, scientists structure and configure the entire experiment, establishing the logical sequence of activities, the type of input data to be provided, and the type of output data. During *execution*, scientists materialize the experiment, define the required input data to run the experiment, trigger its execution (usually carried out by an Experiment Management System), and get the results to be analyzed. During *analysis*, scientists study the gathered data from prior phases [39] aiming at proving or refuting their hypothesis. Each of these experiment phases may involve different forms of collaboration, as discussed in Section 3.1. Provenance plays an important role in each phase, so it is important to keep track of all the user interaction and data transformations on a provenance database.

The second branch of the taxonomy regards the *temporal* aspect of collaboration. This aspect is related to the experience of time and the temporal organization of activities [53]. In a collaborative environment, some tasks need to be synchronized, while others can be done asynchronously. Section 3.2 analyzes if and how existing approaches allow collaborative tasks to occur in real-time or asynchronously.

The third branch is *concurrency control*, which has been extensively studied in the context of databases [52, 15, 23], operating systems [58], and software development [55, 9, 40]. Although the conduction of scientific experiments has its peculiarities, the taxonomy uses ideas that govern version control systems once the problems that may arise when accessing a resource during an experiment resembles the ones that are dealt with by such systems. There are two main concurrency control policies to allow simultaneous work on version control systems: optimistic and pessimistic policy [50]. In pessimistic policies, the artifact that needs to be accessed by several users is restricted to be changed by a single user at a time (i.e., the artifact is locked to a specific user and is only released when the interaction is finished). In optimistic policies, artifacts can be updated in parallel, and users need to merge the changes when conflicts occur. Each of these policies has advantages and disadvantages, and the choice of the most appropriate policy depends on the concurrency frequency, as well as the effort required to merge the artifacts [50]. Section 3.3 discusses how existing approaches deal with concurrency control.

The fourth branch of the taxonomy regards the *sharing* of conceived ideas as well as results and experiments. This allows other researchers to develop new research using these ideas [8]. Although this process is practically mandatory in research, there

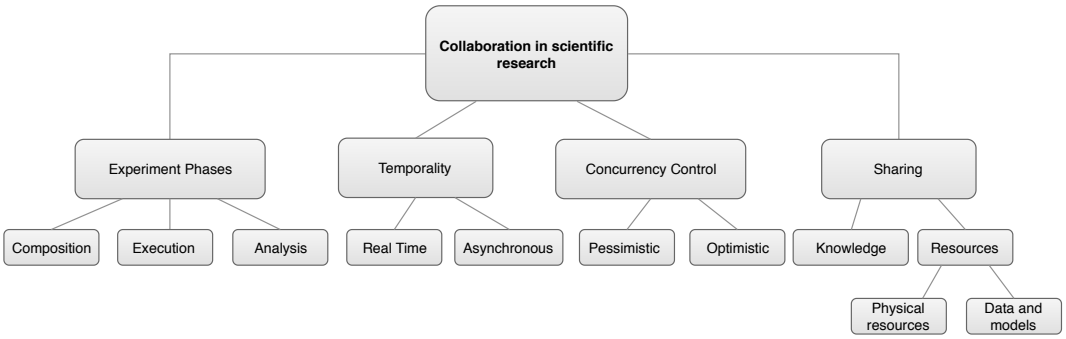


Figure 5: Taxonomy of collaboration in scientific research

is a considerable variation in what is shared, which may facilitate or hinder the research reuse. Some forms of sharing within research would be knowledge sharing, as in publications; data and models sharing, such as sharing a database obtained after some research; and physical resources sharing, such as what happens in the case of institutions sharing a supercomputer. For these different types of sharing (in particular, knowledge, data, or models) to succeed, provenance data is crucial. Without it, the shared information comes out of context and may be useless. Section 3.4 evaluates which of these sharing forms the existing approaches are prepared to deal with, and how this occurs.

Note that all branches of this taxonomy are connected to challenges C1 and C2. They need to be taken into consideration both when collecting provenance (C1) and using this provenance to make collaboration easier (C2). Note also that all branches of the taxonomy are related to data and interaction provenance [51].

Table 2 presents the selected approaches and classifies them according to our taxonomy. This classification considers the aspects addressed in each approach and not the solution maturity of a specific aspect. Thus, two solutions can be equivalently classified, but this does not mean they have the same robustness level. We also evaluated if these tools collect provenance and, when it is possible, classify which type of provenance these tools support. On the next subsections, we detail each of the taxonomy branches and how the surveyed approaches fit them, besides briefly discussing the provenance support of those tools.

3.1 Experiment Phases

Most of the approaches tackle collaboration in the composition phase, while the execution and analysis phases have been receiving less attention.

Composition. This phase has two sub-phases: *conception* and *reuse* [39]. *Conception* aims at producing a high-level representation of the scientific experiment protocol, which is afterward refined and instantiated as a concrete implementation [39] in the form of a workflow or script. *Reuse* consists of retrieving an existing component and adapting it to a new purpose [39].

Some proposals support the *conception* sub-phase [26, 22, 68, 70, 32, 46, 41, 13]. VisTrails [26] is a provenance-aware Workflow Management System that implements support for the collaborative composition of the workflow. Ellkvist et al. [22] and Zhang et al. [68] introduce VisTrails extensions that unleash real-time collaboration on the composition phase of the experiment. Confucius [70] extends Taverna [32] to allow the collaborative composition of workflows by using a client-server architecture that communicates using a service-oriented architecture and XML messages. Mostaeen et al. [46] propose a fine-grained lock scheme that aims to increase efficiency in workflow conception by reducing the waiting time for lock release. ProvDB [41] uses Git to allow the user to collaborate on experiment conception. It also enriches the information collected using ingestors. CoCalc is a virtual workspace for calculations, research, collaboration, and for authoring documents [13], which provides a web portal where scientists can share files with multiple collaborators. This includes Jupyther notebooks, where multiple scientists can simultaneously edit scripts in real-time.

Regarding the *reuse* sub-phase, many of the selected publications focus on the sharing aspect, thus allowing scientists to share a component, a workflow, or a dataset with their peers. That is the case of CAMERA [4], e-ScienceNet [12], myExperiment [28], OpenML [62], Dataverse [35], Collaborative PL-Science [48] and ViroLab [7]. ViroLab [7] provides a way for sharing script components of a workflow. The remaining approaches focus on experiments represented as workflows.

Execution. RASA [42] is the only solution that

Table 2: Aspects of Collaboration in the surveyed Approaches

Approach	Aspects of collaboration				
	Experiment Phase	**Temporality**	**Concurrency Control**	**Sharing**	**Provenance Support**
Confucius [38, 61, 67, 70]	Composition	Asynchronous; Real Time	Pessimistic	Data and models	Data; Interaction
myExperiment [19, 28, 29]	Composition and Analysis	Asynchronous	N/A	Data and models; Knowledge	Yes**
CAMERA [4, 57]	Composition and Analysis	Asynchronous	N/A	Data and models; Knowledge	Yes**
e-ScienceNet [10, 11, 12]	Composition	Asynchronous	N/A	Data and models; Knowledge	No
Collaborative PL-Science [48]	Composition and Analysis	Asynchronous	N/A	Data and models; Knowledge	No
Ellkvist et al. [22]	Composition	Real Time	Optimistic	Data and models	Data
VisTrails [26]	Composition	Asynchronous	Optimistic	N/A	Data
NoCoV [63]	Analysis	Asynchronous; Real Time	N/A	N/A	No
RASA [42]	Execution	Asynchronous	N/A	Physical resources	No
Wood, Wright, and Brodlie [65]	Analysis	Real Time	N/A	N/A	No
ViroLab [7]	Composition	Asynchronous	N/A	Data and models	Yes*
J. Zhang et al [68]	Composition	Real Time	Pessimistic	Data and models	Data; Interaction
Mostaeen et al. [46]	Composition	N/A	Pessimistic	N/A	No
ProvDB [41]	Composition	Asynchronous	Optimistic	Data and models	Data; Interaction
Dataverse [35]	Composition and Analysis	Asynchronous	N/A	Data and models	Yes**
OpenML [62]	Composition and Analysis	Asynchronous	N/A	Data and models	No
CoCalc [13]	Composition and Analysis	Asynchronous; Real Time	Optimistic	Data and models; Knowledge	Data; Interaction
Sumatra [17]	Analysis	Real Time	N/A	Data and models	Data

*No details are provided to correctly classify which provenance types are collected
**Stores data collected by other tools

addresses collaboration in the execution phase of the experiment. RASA is a framework that coordinates the use of scientific instruments, being able to dynamically adapting workflows during the experiment execution according to the needs of the scientists and the equipment.

Analysis. The analysis phase has three sub-phases: *query*, *visualization*, and *discovery* [39]. During *Query*, scientists can relate data and extract information of both the experiment results and provenance data. *Visualization* simplifies the analysis of large volumes of raw data. Data is often projected in graphs or maps to simplify the identification of patterns and the reasoning over the data. During *discovery*, scientists evaluate query results and visual data to draw conclusions about the entire experiment, aiming at checking if the hypothesis is likely to be correct or if it should be refuted. For this, scientists must analyze the experiment as a whole, including all the executions of the experiment (trials) [47].

OpenML [62], CAMERA [4] and myExperiment [28] provide *query* support. They offer a mechanism for sharing not just the workflow components but also other data, such as results and provenance datasets. The myExperiment platform also allows scientists to interact with each other and discuss the shared results. These approaches support the *discovery* sub-phase since they provide a mechanism to analyze and discuss the experiment as a whole. Although not described in the paper [17], Sumatra provides some support to collaboration [56]. It allows different users to share the same provenance database and provides some query features to support the *query* sub-phase.

NoCoV [63] and Wood, Wright, and Brodlie [65] support the *visualization* sub-phase. NoCoV (Notification-service-based Collaborative Visualization) uses a client-server architecture to provide mechanisms for the collaborative visualization of experi-

ment data. The pipeline controller (server) is responsible for synchronizing the clients' visualization, and multiple clients can connect to it simultaneously. The clients could be a pipeline editor (which can update the visualization pipeline) or a parameter control client (which can only adjust visualization parameters). Wood, Wright, and Brodlie [65] propose a collaborative approach on top of IRIS Explorer [24] that allows multiple scientists to interact over a visualization collaboratively.

CoCalc [13] supports the *query, discovery, and visualization* sub-phases. It allows scientists to query the results of the experiment and its history, besides other data. Scientists can also visualize the results using Jupyther notebooks and libraries, such as matplotlib. They can also use chat rooms to discuss the experiment and reason about it.

Dataverse [35] focuses on creating an infrastructure to share datasets related to scientific publications. It provides the data to be used in the *query, discovery, and visualization* sub-phases, although it does not explicitly deal with them.

3.2 Temporality

Starting with the approaches that implement *asynchronous* interactions, CAMERA [4], myExperiment [28], e-ScienceNet [10], Collaborative PL-Science [48], ViroLab [7], Dataverse [35] and OpenML [62] provide solutions focused on the sharing of data and components, where a scientist can publish workflows, components or datasets. These published artifacts become available for other scientists to reuse them *asynchronously*. On VisTrails [26], each version of the workflow is treated as a node in a version tree. Nodes are never modified or deleted (each modification generates a new node in the tree). To collaboratively compose a workflow, scientists can asynchronously work in their local copy of the workflow and synchronize it with another scientist's copy when needed. However, if two scientists modify the same workflow before synchronizing it, this generates multiple disjoint versions, which can be problematic since the changes could be complementary. When this occurs, the scientist should re-implement part of the workflow. ProvDB [41] is a client-server application that uses Git to support version management tasks as well as distributed and decentralized management of individual repositories. Each user makes the necessary modifications to her local repository and, asynchronously, synchronizes them using Git.

We have also identified several proposals that provide *real-time* collaboration. Ellkvist et al. [22] implement a solution based on a client/server architecture, where the server is a MySQL database, and the client is a modified version of VisTrails, that consists of a mechanism to unleash real-time collaboration during workflow composition. The server is used as a shared database to synchronize the versions among the scientists. When one scientist makes a modification, it is saved on the shared database and the other clients are automatically notified to update their local versions. Although implemented in VisTrails, the authors argue that their solution could be implemented in other provenance-aware Workflow Management Systems. Zhang et al. [68] also implement a plugin to VisTrails, which allows any changes made by one scientist to be immediately reflected on all other collaborators' screens. The approach communicates with VisTrails through third-party packages and the VisTrails API. It utilizes Git to provide a new version tree over the existing VisTrails History View. Wood, Wright, and Brodlie [65] present a real-time approach based on a client-server architecture, which allows scientists to visualize an experiment collaboratively. Users can share and alter visualization parameters and visualization pipelines so they can see other users' changes in real-time. Participants may also disconnect single modules from their group to allow periods of independent work on a subset of the pipeline while remaining in contact with the rest of the session. Sumatra [17] provides a way of sharing the provenance database in real-time. The information is shared as soon as it is collected. However, the solution still has several limitations and, in some scenarios, even data loss is possible.

Three solutions work in both real-time and asynchronous scenarios. Confucius [70] provides a solution inspired by a protocol of human communication called Robert's Rules of Order, which is a set of rules created by Henry M. Robert in 1876 to run effective and orderly meetings with maximum fairness to all members [33]. Confucius implements that with a locking strategy that controls which scientist has the right to interact at a given time in a real-time collaboration session. Confucius also maintains a database on the central node that is used for storing provenance of collaboration and workflow evolution, which allows asynchronous collaboration. NoCoV [63] is implemented in a service-oriented architecture that uses notification Web services to synchronize clients and server. When someone alters the visualization pipeline, the pipeline controller notifies other clients, so everyone sees the same visualization in real-time. To transmit information between the pipeline controller and the client, it uses skML [21], an XML-based dataflow

description language. NoCoV uses the stateful Web Services provided by GlobusToolkit 4 (GT4) [59]. Using this stateful feature, the state of the pipeline is persisted and users can retrieve the saved pipeline to continue the work of other users, thus achieving asynchronous collaboration. CoCalc [13] provides a solution based on a web portal where scientists can simultaneously compose scripts in real-time. All changes are immediately synchronized with others. It saves files and data in its cloud infrastructure, so scientists can leave the session and rejoin when needed (allowing asynchronous work).

3.3 Concurrency Control

All approaches providing a mechanism for concurrency control focus on the composition phase of the scientific experiment.

Starting with the approaches that implement the *pessimistic* policy for concurrency control, Confucius' authors [70] treat the concurrency control problem as they would treat it in a face-to-face activity. A central node is needed for the collaboration to occur. A group is registered on this node, and the person responsible for registering the group is automatically assigned as the group moderator. The moderator is responsible for shift control, which is the definition of which group member is allowed to change the workflow at a given time. There is an algorithm for automatically granting and releasing the right to the shift, but the moderator can intervene by taking the right to the shift. Confucius also considers that workflow development can last for long periods in an asynchronous form and, in this scenario, workflow level locking may not be appropriate. Therefore, Confucius blocks smaller building blocks. Thus, several scientists can change the same workflow at the same time. Confucius establishes that the smallest building blocks are tasks and data channels, that in Taverna are called processors and data links, respectively. Confucius introduces the concept of synchronization area "that represents a conceptual area in a shared scientific workflow, which allows only one collaborator to work on it at a given time" [70]. When the user starts to modify a data link, the synchronization area is the data link itself. When the user locks a processor, the synchronization area is the processor and all the fan-out data links of the processor. Zhang et al. [68] also implement a pessimistic collaboration protocol based on Robert's Rules of Order. The protocol is fully described in [34, 69]. Mostaeen et al. [46] analyze the existing locking schemes in terms of concurrency control on the composition of workflows. The approach presents a pessimistic strategy of fine-grain locking in scientific workflows. The lock is done for a single user but at the attribute level, while other approaches use turns or module level locking. The main benefit here is to reduce the waiting time for a lock since smaller portions of the workflow are locked for each modification.

Only four approaches implement the *optimistic* policy for concurrency control. Ellkvist et al. [22] and VisTrails [26] present an optimistic lock approach that creates different branches in the version tree in the case of simultaneous changes. Although VisTrails presents a mechanism for merging, it merges two version trees of different files and not two branches of the same version tree. If the scientists want to keep both of the changes, they will have to use the diff functionality to better understand what has changed and to replay the changes manually. VisTrails also has a functionality called 'analogy' that could help on the process: given two versions of a workflow, VisTrails can automatically detect their differences and apply those differences to another workflow version. Ellkvist et al.'s proposal [22] is built on top of VisTrails, and although it adds support to real-time collaboration, it uses the same concurrency control approach of VisTrails. ProvDB [41] also works on the idea of immutable versions, in which any update will result in a new version. In Cocalc [13], the whole experiment environment is cloud-based. All changes are made directly in the cloud and synchronized with the online scientists' browser – there is no lock.

3.4 Sharing

Most of the approaches providing sharing features allow the sharing of *data and models*. That is the case of e-ScienceNet [12], ViroLab [7], myExperiment [29], CAMERA [4], Dataverse [35], OpenML [62], ProvDB [41], Zhang et al. [68], Ellkvist et al. [22], Confucius [70], Collaborative PL-Science [48], CoCalc [13] and Sumatra [17]. ProvDB [41], Zhang et al. [68], Ellkvist et al. [22], and Confucius [70] work with a centralized database for the experiment, which stores the provenance collected from the collaborative experiment and makes this information available to the involved scientists. ViroLab [7] addresses the issue of sharing code blocks for reuse. The approach also mentions the persistence and sharing of provenance but does not provide details on what kind of provenance information is stored and shared. Sumatra [17] provides a way of sharing a provenance database between multiple scientists.

Roure, Globe, and Stevens [19] argue that one of the barriers of workflow reuse is on how the knowl-

edge about the workflow could be transmitted to potential users. That challenge can be minimized by the distribution of other documentation data in addition to the workflow definition. Most of the approaches try to increase collaboration by adding the possibility of sharing *knowledge*. That is the case of e-ScienceNet [12], myExperiment [29], CAMERA [4], Dataverse [35], CoCalc [13], and Collaborative PL-Science [48]. Pereira et al. [48] propose the Collaborative PL-Science, an extension of PL-Science [14]. It aims to facilitate the reuse of components in the construction of scientific workflows, thus combining models and knowledge sharing. The idea is that adding information that helps to understand published artifacts facilitates reuse. The approach uses ontologies to enrich the information of shared objects. CoCalc [13] allows the sharing of a great variety of files, including scripts in multiple programming languages. It also allows the sharing of documentation that can help scientists to better understand what has been made on the experiment and help them to better use the shared data and scripts. e-ScienceNet [12] is another approach that allows both the sharing of data and models and also knowledge. It differs from other approaches because it presents a peer-to-peer solution for sharing the experiment results and models without the dependency of a central server.

Some publications explore the creation of portals for sharing data and reusable components in research, where it is common to share scientific workflows. Goble and Roure [29] propose myExperiment, a social network for scientists focused on workflow-related issues. It allows the sharing of the workflow itself as well as other metadata, such as provenance logs, besides enabling researchers to interact using the tool, commenting, and discussing the shared resources. CAMERA [4] also focuses on the sharing of scientific workflows and provenance logs. The tool works exclusively with Kepler [3] workflows and allows the execution of the experiments within the portal. OpenML [62] is focused on the machine learning community and provides a portal to share datasets, algorithm implementations, and workflows. It also presents a Web API, which allows users to interact with the portal in a programmatic form, and ways of sharing scientific tasks and receiving other scientists' collaboration. Dataverse [35] provides a Web infrastructure to share datasets related to scientific publications. The main idea is that sharing the datasets may increase the reproducibility of experiments, and, as a counterpart to the authors, it may increase the number of citations of the related publications [35].

RASA [42] is the only approach that focuses on sharing physical resources. The approach provides a framework for coordinating the use of scientific instruments. The idea is to provide a mechanism to dynamically modify workflows depending on the needs of the requester scientist and the particularities of the equipment, and also the knowledge of the equipment operator.

3.5 Provenance Support

As seen in Table 2, many of the tools do not collect provenance. Although ViroLab [7] provides some provenance support, it does not give details on what is stored. Dataverse [35], CAMERA [4], and myExperiment [19] provide support for storing and sharing provenance data collected by other tools. CoCalc [13] collects interaction provenance through the log of the activities executed by scientists, but this unstructured information is hard to query. VisTrails [26], Ellkvist et al. [22], and Sumatra [17] can capture data provenance from multiple users in their local stations and consolidate them in a single database, but those databases do not properly represent collaboration aspects of the research covered by Section 2, thus collaboration provenance is not included. Zhang et al. [68], Confucius [38, 61, 67, 70], and ProvDB [41] provide data and interaction provenance support, and use the collaboration-aware provenance models described in Section 2. The models proposed by Confucius and ProvDB need extended properties to represent some collaboration aspects, but the tools proposed by those papers are not able to capture these properties. Thus, there is a difference between the provenance types represented by the models and those supported by the tools.

4. DISCUSSION AND OPPORTUNITIES

Figure 6 shows a timeline that helps understand how research has progressed in this field. Some of the publications are highly related and represent the evolution of the same research. In such cases, we treat them in a consolidated manner, thus linking these publications in the figure and handling them as a single approach. This topic has received much attention in recent years, but there are still some gaps to be further explored. In this section, we classify the selected approaches, answer the research questions introduced in Section 1, discuss the gaps that still exist, and present opportunities derived from those gaps.

R1: How do existing tools store and collect provenance in a collaborative experiment?

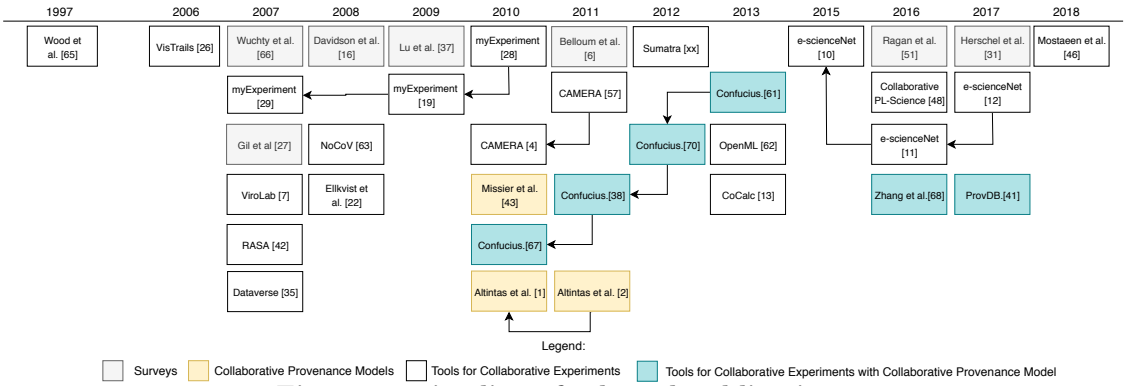


Figure 6: Timeline of selected publications

To answer this question, we analyzed the available models for storing provenance in collaborative environments. Although significant progress has been made with those models, all of them present limitations (they do not deal with different workflow formats, or do not deal with workflow evolution). Models that can represent all the aspects we analyzed do so by using extended properties, which makes them difficult to query.

Regarding the available tools and how they collect provenance: Some tools (Dataverse [35], CAMERA [4], and myExperiment [19]) just provide storage for provenance, but do not collect it. Other tools (VisTrails [26] , Ellkvist et al. [22], and Sumatra [17]) provide a way of consolidating the provenance collected from different users but lack support for other collaboration aspects. Finally, a few tools (Zhang et al. [68], Confucius [38, 61, 67, 70], and ProvDB [41]) use collaborative aware provenance models but still present some limitations.

R2: How do existing tools use provenance to make collaboration easier in scientific experiments? We conclude that the surveyed approaches fail to use the collected provenance to support the collaboration. Although Confucius [70], Zhang et al. [68], and ProvDB [41] are capable of collecting provenance of the collaboration process, they do not propose forms of using that valuable data to increase the efficiency and awareness of the process.

As illustrated in Table 2, most of the approaches support the composition phase of the experiment life cycle (especially the conception sub-phase). However, they are mostly based on Workflow Management Systems and ignore the fact that many scientists use scripts in their experiments [49]. The only approaches that support experiments represented as scripts are ViroLab [7], Sumatra [17], CoCalc [13], and ProvDB [41]. However, ViroLab only addresses the reuse sub-phase of the experiment composition. Sumatra fully delegates the script composition to Git and presents several limitations for the shared provenance storage, such as a possible data loss depending on the network connection. Despite being quite complete, CoCalc [13] demands the scientist to be online in order to work, and that she works on the browser, which can be a tough change in the workspace, tools, and IDEs that the scientist is used to. It is possible to run applications from the CoCalc portal, but this is not the same as running them from the scientist's machine. It also presents several limitations on free accounts. Another point worth mentioning is that it does not properly capture the provenance of the experiment. It presents features like "time travel" and "log" that let users see the history of the files and activity on the project, but it is very high level and may not be enough to guarantee the reproducibility of the experiment, for example. ProvDB uses Git to handle version management and a provenance ingestor framework to capture other provenance data, but it is highly specialized in data science problems and is not well prepared for a general-purpose experiment.

Although versioning tools handle several collaborative needs of script building, they are software development tools that do not address specific problems in scientific research. These tools will not provide provenance capture and analysis support by default. Provenance is not just related to the obtained results but also the input data, intermediate results, etc. Trying to deal with this complexity without the proper tooling support could take much effort from the scientists and steal the energy that should be spent on research. Although ProvDB considers these challenges, it depends on the scientist being able to access an external tool (Git), a specific OS (UNIX), and demands the creation of ingestors to capture some provenance aspects. ProvDB is also focused on a specialized type of experiment (data science analysis), and does not address awareness

during collaboration. Thus, ***we must investigate and design provenance-aware tools that can handle composition, execution, and analyses of generic script-based experiments collaboratively, increasing the awareness of users during the process at the same time***.

The execution phase also lacks support. We could find only one approach that supports collaboration in this phase of the experiment life cycle. RASA [42] supports the execution phase by controlling access to physical resources such as equipment. Providing provenance-aware support of the execution phase is crucial in collaborative experiments, since without it, important aspects of the collaboration may be lost. In fact, for reproducibility purposes, it is crucial to know which user executed each part of the experiment, where and under which conditions. Thus, the ***support for the collaborative execution of scientific experiments needs more investigation***.

Some approaches support the analysis phase of experiments. Most of them allow scientists to comment on the experiment structure or results. Some approaches [7, 26, 41, 70] provide provenance gathering of the collaborative experiment that could help the analysis of the experiment. However, they do not provide a clear way to collaborate throughout the analysis, so they were classified without this phase of the life cycle in Table 2.

Temporality is well explored, with several approaches supporting asynchronous or real-time interactions. However, some features could be improved. When conducting an experiment in groups, it is important to know what happened in the experiment while scientists were offline, who did what, and in which part of the experiment (interaction provenance). It is also important to know if there is anyone online and in which part of the experiment they are working at. Although some tools let the users query for some of that information, it would be desirable that such information would be automatically shown to users, depending on the context of the experiment. Thus, ***an interesting issue to examine will be ways of increasing the awareness of the scientists about the actions of their collaborators***.

As for concurrency control, most of the approaches use a pessimistic locking scheme. Pessimistic locking may work well in real-time scenarios, but it can be quite troublesome for asynchronous collaboration. VisTrails [26] and Ellkvist et al. [22] are the only solutions that work with an optimistic locking scheme, but they do not implement a merging mechanism capable of merging two workflow branches. Although VisTrails diff and analogy functionalities could help to merge two branches, they impose some additional steps for such a task and lack some basic merge functionalities like conflict resolution. Thus,***we need tools that work with optimistic locking and provide complete merge support in the composition of workflows***.

Also, in a collaborative environment, some collaboration tasks may perform better if treated with a pessimistic locking policy while others will benefit from an optimistic approach [50]. In experiments with files that are difficult to merge, scientists could opt to work with a pessimistic policy, while in others they may prefer to work with an optimistic one. Existing tools only implement one of the policies, so if scientists want to use this tool, they are forced to use the implemented policy. Scientists must have the flexibility needed to interact in a way that is more appropriate to the use case in hand. Thus, ***tools that allow scientists to choose the more appropriate lock policy are needed***.

Sharing is well covered in the literature with a wide range of available solutions. Solutions address centralized sharing as well as peer-to-peer sharing, besides providing mechanisms for commenting and enriching the shared artifacts, making them easier to use. We believe that, in this aspect, there is no clear gap in the available tools.

We end up finding that none of the available tools are capable of using provenance to make collaboration easier in scientific experiments (related to R2). So, ***there is a need to investigate how to use the captured provenance to make collaboration easier in scientific research***.

5. CONCLUSION

Scientific research is frequently collaborative and also conducted in silico. Although this is very positive for science, it brings several challenges. To better understand the challenges and evaluate the literature on the subject, this article presents a survey on collaboration in in silico scientific research. In this survey, we map the available tools and the state-of-art of research on collaborative experiments conducted in silico. We propose a taxonomy and use it to classify the existing tools and discuss opportunities based on the gaps we identified. We believe that a more systematic review process could find new articles and enrich the obtained results. However, we believe we cover a large part of the publications on the topic, and our findings at this stage can be useful and generate insights to researchers interested in this topic.

6. REFERENCES

[1] I. Altintas, M. K. Anand, D. Crawl, S. Bowers, A. Belloum, P. Missier, B. Ludäscher, C. A. Goble, and P. M. A. Sloot. Understanding collaborative studies through interoperable workflow provenance. In D. L. McGuinness, J. R. Michaelis, and L. Moreau, editors, *Provenance and Annotation of Data and Processes*, pages 42–58. Springer Berlin Heidelberg, 2010.

[2] I. Altintas, M. K. Anand, T. N. Vuong, S. Bowers, B. Ludäscher, and P. M. A. Sloot. A data model for analyzing user collaborations in workflow-driven escience. *International Journal of Computers and Their Applications*, 18:160–179, 2011.

[3] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. In *Scientific and Statistical Database Management*, pages 423–424, 2004.

[4] I. Altintas, A. W. Lin, J. Chen, C. Churas, M. Gujral, S. Sun, W. Li, R. Manansala, M. Sedova, J. S. Grethe, and M. Ellisman. Camera 2.0: A data-centric metagenomics community infrastructure driven by scientific workflows. In *World Congress on Services*, pages 352–359, 2010.

[5] M. K. Anand, S. Bowers, T. McPhillips, and B. Ludäscher. Exploring scientific workflow provenance using hybrid queries over nested data and lineage graphs. In M. Winslett, editor, *Scientific and Statistical Database Management*, Lecture Notes in Computer Science, pages 237–254. Springer Berlin Heidelberg, 2009.

[6] A. Belloum, M. A. Inda, D. Vasunin, V. Korkhov, Z. Zhao, H. Rauwerda, T. M. Breit, M. Bubak, and L. O. Hertzberger. Collaborative e-science experiments and scientific workflows. *IEEE Internet Computing*, 15(4):39–47, July 2011.

[7] M. Bubak, T. Gubala, M. Kasztelnik, M. Malawski, P. Nowakowski, and P. Sloot. Collaborative virtual laboratory for e-health. In *Expanding the Knowledge Economy: Issues, Applications, Case Studies, eChallenges*, pages 537–544, 2007.

[8] R. Caldwell and D. Lindberg. Participants in science behave scientifically. *Understanding Science.*, 2018. Available at `https://undsci.berkeley.edu/article/0_0_0/whatisscience_09`.

[9] S. Chacon and J. Long. Git. `https://git-scm.com/`. Accessed: 2018-06-09.

[10] T. Classe, R. Braga, F. Campos, and J. M. N. David. A semantic peer to peer network to support e-science. In *IEEE International Conference on e-Science*, pages 503–512, 2015.

[11] T. Classe, R. Braga, J. M. N. David, F. Campos, M. A. Araújo, and V. Ströele. A collaborative approach to support e-science activities. In *IEEE International Conference on Computer Supported Cooperative Work in Design*, pages 20–25. IEEE, 2016.

[12] T. Classe, R. Braga, J. M. N. David, F. Campos, and W. Arbex. A distributed infrastructure to support scientific experiments. *Journal of Grid Computing*, 15(4):475–500, 2017.

[13] Cocalc user manual documentation. `https://doc.cocalc.com/contents.html`, 2013. Accessed: 2019-12-05.

[14] G. C. B. Costa, R. Braga, J. M. N. David, and F. Campos. A scientific software product line for the bioinformatics domain. *Journal of Biomedical Informatics*, 56:239–264, 2015.

[15] C. J. Date. *An introduction to database systems*. Pearson/Addison Wesley, Boston, 2004.

[16] S. B. Davidson and J. Freire. Provenance and scientific workflows: Challenges and opportunities. In *ACM Special Interest Group on Management of Data*, pages 1345–1350. ACM, 2008.

[17] A. Davison. Automated capture of experiment context for easier reproducibility in computational research. *Computing in Science & Engineering*, 14(4):48–56, 2012.

[18] D. De Oliveira, E. Ogasawara, F. Baião, and M. Mattoso. Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows. In *International Conference on Cloud Computing*, pages 378–385, Washington, DC, USA, 2010.

[19] D. De Roure, C. Goble, and R. Stevens. The design and realisation of the myexperiment virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25(5):561–567, 2009.

[20] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: a framework for mapping complex scientific workflows onto

distributed systems. *Scientific Programming Journal*, 13(3):219–237, 2005.
[21] D. A. Duce and M. S. Sagar. skML a markup language for distributed collaborative visualization. In *Theory and Practice of Computer Graphics*, pages 171–178, 2005.
[22] T. Ellkvist, D. Koop, E. W. Anderson, J. Freire, and C. Silva. Using provenance to support real-time collaborative design of workflows. In *International Workshop on Provenance and Annotation (IPAW)*, pages 266–279. Springer, 2008.
[23] R. Elmasri and S. Navathe. *Fundamentals of database systems*. Addison-Wesley, 6 edition, Apr. 2010.
[24] D. Foulser. IRIS Explorer: a framework for investigation. *SIGGRAPH Computer Graphics*, 29(2):13–16, 1995.
[25] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21, 2008.
[26] J. Freire, C. T. Silva, S. P. Callahan, E. Santos, C. E. Scheidegger, and H. T. Vo. Managing rapidly-evolving scientific workflows. In L. Moreau and I. Foster, editors, *Provenance and Annotation of Data*, Lecture Notes in Computer Science, pages 10–18. Springer Berlin Heidelberg, 2006.
[27] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers. Examining the challenges of scientific workflows. *Computer*, 40(12):24–32, 2007.
[28] C. A. Goble, J. Bhagat, S. Aleksejevs, D. Cruickshank, D. Michaelides, D. Newman, M. Borkum, S. Bechhofer, M. Roos, P. Li, and D. De Roure. myexperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research*, 38(Web Server Issue):677–682, 2010.
[29] C. A. Goble and D. C. D. Roure. myexperiment: Social networking for workflow-using e-scientists. In *Workshop on Workflows in Support of Large-Scale Science*, pages 1–2. ACM, 2007.
[30] L. A. Goodman. Snowball sampling. *The Annals of Mathematical Statistics*, 32(1):148–170, 1961.
[31] M. Herschel, R. Diestelkämper, and H. B. Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906, 2017.
[32] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34(2):729–732, 2006.
[33] H. M. R. III, D. H. Honemann, T. J. Balch, D. E. Seabold, and S. Gerber. *Robert's rules of order newly revised*. PublicAffairs, 11 edition, 2011.
[34] Jia Zhang, C. Chang, and Jen-Yao Chung. Mediating electronic meetings. In *International Computer Software and Applications Conference*, pages 216–221, 2003.
[35] G. King. An introduction to the dataverse network as an infrastructure for data sharing, 2007.
[36] B. Lerner and E. Boose. Rdatatracker: collecting provenance in an interactive scripting environment. In *USENIX Workshop on the Theory and Practice of Provenance (TaPP)*, 2014.
[37] S. Lu and J. Zhang. Collaborative scientific workflows. In *IEEE International Conference on Web Services*, pages 527–534. IEEE, 2009.
[38] S. Lu and J. Zhang. Collaborative scientific workflows supporting collaborative science. *International Journal of Business Process Integration and Management*, page 185, 2011.
[39] M. Mattoso, C. Werner, G. H. Travassos, V. Braganholo, E. Ogasawara, D. Oliveira, S. Cruz, W. Martinho, and L. Murta. Towards supporting the life cycle of large scale scientific experiments. *International Journal of Business Process Integration and Management*, 5(1):79–92, 2010.
[40] Mercurial scm. `https://www.mercurial-scm.org/`. Accessed: 2019-04-23.
[41] H. Miao, A. Chavan, and A. Deshpande. Provdb: Lifecycle management of collaborative analysis workflows. In *Workshop on Human-In-the-Loop Data Analytics (HILDA)*, pages 7:1–7:6, New York, NY, USA, 2017. ACM.
[42] T. Miller, P. McBurney, J. McGinnis, and K. Stathis. First-class protocols for agent-based coordination of scientific instruments. In *IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 41–46, 2007.
[43] P. Missier, B. Ludascher, S. Bowers, S. Dey, A. Sarkar, B. Shrestha, I. Altintas, M. Anand, and C. Goble. Linking multiple workflow provenance traces for interoperable

collaborative science. In *Workshop on Workflows in Support of Large-Scale Science*, pages 1–8, 2010.

[44] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche. The open provenance model core specification (v1.1). *Future Generation Computer Systems*, 27(6):743–756, 2011.

[45] L. Moreau, P. Missier, K. Belhajjame, R. B'Far, J. Cheney, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, S. Miles, J. Myers, S. Sahoo, and C. Tilmes. PROV-DM: The PROV data model. W3C Recommendation. *W3C Recommendation*, 2013. Available at `http://www.w3.org/TR/2013/REC-prov-dm-20130430/`.

[46] G. Mostaeen, B. Roy, C. K. Roy, and K. A. Schneider. Fine-grained attribute level locking scheme for collaborative scientific workflow development. In *IEEE International Conference on Services Computing*, pages 273–277, 2018.

[47] L. Murta, V. Braganholo, F. Chirigati, D. Koop, and J. Freire. noworkflow: Capturing and analyzing provenance of scripts. In *International Workshop on Provenance Annotation (IPAW)*, pages 1–12, 2014.

[48] A. F. Pereira, J. M. N. David, R. Braga, and F. Campos. An architecture to enhance collaboration in scientific software product line. In *International Conference on System Sciences*, pages 338–347. IEEE, 2016.

[49] J. F. Pimentel, J. Freire, L. Murta, and V. Braganholo. A survey on collecting, managing, and analyzing provenance from scripts. *ACM Computing Surveys*, 52(3):47:1–47:38, 2019.

[50] J. Prudêncio, L. Murta, C. Werner, and R. Cepêda. To lock, or not to lock: That is the question. *Journal of Systems and Software*, 85(2):277–289, 2012.

[51] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, 2016.

[52] R. Ramakrishnan and J. Gehrke. *Database management systems*. McGraw-Hill, New York, third edition edition, 2003.

[53] M. C. Reddy, P. Dourish, and W. Pratt. Temporality in medical work: Time also matters. *Computer Supported Cooperative Work*, 15(1):29–53, 2006.

[54] D. H. Sonnenwald. Scientific collaboration. *Annual review of information science and technology*, 41(1):643–681, 2007.

[55] Apache subversion. `https://subversion.apache.org/`. Accessed: 2019-04-23.

[56] Sumatra 0.7.0 documentation. `https://pythonhosted.org/Sumatra/record_stores.html`. Accessed: 2019-12-03.

[57] S. Sun, J. Chen, W. Li, I. Altintas, A. Lin, S. Peltier, K. Stocks, E. E. Allen, M. Ellisman, J. Grethe, and J. Wooley. Community cyberinfrastructure for advanced microbial ecology research and analysis: the CAMERA resource. *Nucleic Acids Research*, 39:D546–551, 2011.

[58] A. S. Tanenbaum. *Modern operating systems*. Prentice Hall, Upper Saddle River, N.J, 3 edition edition, Dec. 2007.

[59] Gt4 globus toolkit web site. `http://toolkit.globus.org/toolkit/`. Accessed: 2019-04-23.

[60] G. H. Travassos and M. O. Barros. Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering. In *Workshop on Empirical Software Engineering the Future of Empirical Studies in Software Engineering*, pages 117–130, 2003.

[61] S. Vali and S. Sreerama. Multi-user tool for scientific work flow composition. *International Journal of Computer Trends & Technology*, 4, 2013.

[62] J. N. Van Rijn, B. Bischl, L. Torgo, B. Gao, V. Umaashankar, S. Fischer, P. Winter, B. Wiswedel, M. R. Berthold, and J. Vanschoren. Openml: A collaborative science platform. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 645–649. Springer, 2013.

[63] H. Wang, K. W. Brodlie, J. W. Handley, and J. D. Wood. Service-oriented approach to collaborative visualization. *Concurrency and Computation: Practice and Experience*, 20(11):1289–1301, 2008.

[64] M. Wilde, I. Foster, K. Iskra, P. Beckman, Z. Zhang, A. Espinosa, M. Hategan, B. Clifford, and I. Raicu. Parallel scripting for applications at the petascale and beyond.

Computer, 42(11):50–60, 2009.

[65] J. Wood, H. Wright, and K. Brodlie. Collaborative visualization. In *Conference on Visualization*, pages 253–259. IEEE Computer Society Press, 1997.

[66] S. Wuchty, B. F. Jones, and B. Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.

[67] J. Zhang. Co-taverna: A tool supporting collaborative scientific workflows. In *IEEE International Conference on Services Computing*, pages 41–48, 2010.

[68] J. Zhang, Q. Bao, X. Duan, S. Lu, L. Xue, R. Shi, and P. Tang. Collaborative scientific workflow composition as a service: An infrastructure supporting collaborative data analytics workflow design and management. In *IEEE International Conference on Collaboration and Internet Computing*, pages 219–228, 2016.

[69] J. Zhang, C. K. Chang, and J. Voas. A uniform meta-model for mediating formal electronic conferences. In *International Computer Software and Applications Conference*, pages 376–381. IEEE, 2004.

[70] J. Zhang, D. Kuc, and S. Lu. Confucius: A tool supporting collaborative scientific workflow composition. *IEEE Transactions on Services Computing*, 7(1), 2012.

Susan Davidson Speaks Out on Collaborating with Other Research Areas and Balancing Work and Family

Marianne Winslett and Vanessa Braganholo

Susan Davidson
https://www.cis.upenn.edu/~susan/

Welcome to ACM SIGMOD Records series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we're at the 2017 SIGMOD and PODS conference in Chicago. I have with me Susan Davidson, who's a professor at the University of Pennsylvania. Sue is an ACM Fellow, a Corresponding Fellow of the Royal Society of Edinburgh, and the recipient of the 2017 IEEE Technical Committee on Data Engineering Impact Award. Sue has served as the chair of her department, Deputy Dean of the School of Engineering and Applied Science, a member of the NSF Cise Advisory Committee, and as the chair of the Computing Research Association. Sue's Ph.D. is from Princeton University. So, Sue, welcome.

Thank you, Marianne.

There've been many decades of work on data provenance, but no one uses it! Scientists care about the provenance of their data, but they seem to be hand drilling their own solutions instead of using ours. Does that mean that we are solving the wrong problems?

I think that our solutions actually are beginning to be used, and can give a few examples. First of all, Zachary Ives' Orchestra System (Collaborative Data Sharing) is based on provenance. A fundamental aspect is the use of provenance tokens to evaluate trust. I've also heard that Boris Glavic has used provenance in his work with Oracle. And Laura Haas, in her keynote at ICDE 2017, showed how provenance was being captured and used in the context of the Accelerated Discovery Lab work.

So, I think there is impact. Where I don't think the impact is showing up is in a bench biologist's lab. And here I think the reason is a natural aversion to technology. Many people who go into biology don't really like technology or math. They really don't want to have to learn another workflow system that does automatic provenance capture. And they don't record provenance in the way that we think of it -- they typically use file names and notes to be able to record provenance.

That's cute that you started with biologists, but they may be the last to get on board.

They're late adopters sometimes.

What is the relationship between data provenance and data citation?

There's definitely a connection. Both of them are forms of annotation on data. Provenance as annotation is a big theme. Citation is annotation as well, but there's something additional in the snippets of information that you want to capture that may not be exactly provenance.

For example, you want to be able to record snippets of information that lets the reader of the citation know whether or not they want to look at the material being cited. Typically this involves something like authorship or title.

Can you give us an example?

Oh, yes. As an example, there's Einstein's famous paper on special relativity[1]. If you just give the reference to it, which is the journal in which it occurred, the volume, the number, the pages even, most people do not know that this is Einstein's paper on special relativity.

The same is true for the famous Watson and Crick paper on the helical structure of DNA[2]. Most people won't recognize the journal, the volume, the year. And so, additional information about the authorship and the title really gives people the intuition of whether they want to go look at that piece of work.

So, would you define data citation as provenance information plus a little marketing blurb?

Well, that's interesting, I hadn't thought of it that way. Certainly, it is provenance, but it may not be the "deep" provenance we think about, for example, in the provenance semirings work of Val Tannen. Data provenance typically starts from the very creation of the data as it was input to the database, and is tracked through queries, building up very complex provenance polynomials. The same happens in workflows, where data is tracked as the inputs and the outputs of processing steps that the data goes through.

In data citation, very often you just want to know where the data came from. Tracking back to see the influences of a previous work would be done through something like a citation graph, extending transitively past the immediate references. But we don't typically include that information in a citation.

And what makes data citation hard as a research problem?

I think what makes it hard is that the data is in a database rather than being published as an encyclopedia or a compendium of some sort. And the content in the database has been potentially contributed by many different individuals.

So, depending on the part of the database that you're interested in, the people who you should acknowledge are different. And there are a whole bunch of different parts of the database and a potentially infinite number of queries that would bring back a part of the database. So, you can't possibly attach an explicit citation to every possible query.

What you have to be able to do is to work with a small set of citations that the owners of the database attach to

[1] A. Einstein: On the Electrodynamics of Moving Bodies. Annalen der Physik 17: 891-921 (1905).

[2] J. D. Watson, F. H. C. Crick: Molecular Structure of Nucleic Acids: a structure for deoxyribose nucleic acid. Nature 171: 737-738 (1953).

pieces of the database, and use these to construct citations to general queries over the database. So, it's really the granularity of citations and number of different queries that makes this an interesting problem.

Being practical, can data provenance and data citation help solve the problem of fake news?

That's a really interesting question. Fundamentally, the problem with fake news is that you don't have a trustworthy origin for the news. So, if somebody claims that something is true without a reference to something that's trustworthy, then how do you know whether it's true or not?

If the person would give a reference, that is, they would provide the provenance for that remark, then you could go back and further evaluate whether you trust that as a source for that type of information.

So, I think it has something to do with provenance, and certainly, if there's provenance back to a trustworthy source, then it could be helpful. But in the absence of an endpoint which is trustworthy, there's really not much you can do about it.

[…] do you have a comfortable relationship with the person in that other field of whom you can ask stupid questions?

It seems like data researchers have gotten better over the years at picking problems to work on that are more likely to have an impact. Often that means looking at the data problems in a particular application area like bioinformatics in your case and recently air-conditioning, of all things in my own group. How do you do research that helps biologists or the air-conditioning industry when you aren't an expert on the subject matter?

Well, I've never worked in the air-conditioning industry, but I can talk about bioinformatics. I think that what it really boils down to is, do you have a comfortable relationship with the person in that other field of whom you can ask stupid questions? And do you have a basis – a vocabulary – with which to speak?

I like to think of this as a string of people holding hands, with different strengths and experiences. Somebody who can talk to the end-user who can also talk to somebody more on the systems building side. Maybe that person would talk to someone more on the theory side.

Not every member of the team needs to be an expert in all areas, in my case, in bioinformatics. And the biologist may not understand or even care about the technical solution. It's a team of people working together, talking – a lot of conversations have to happen. Frequently postdocs are really helpful, because they have the experience that a graduate student might not have, and they have the time that a professor doesn't have.

And how can you tell when you found the right team of collaborators?

I think it really is chemistry. A lot of this boils down to: Is this a group of people that you feel comfortable talking with, because you will inevitably display your ignorance. And if you're feeling nervous about somebody finding out what you don't know, then it's very difficult to ask the right questions and gain the experience that you need to come up with solutions.

How did you yourself become interested in bioinformatics?

I grew up in an academic family. My father was a professor of applied math, and my mother was a professor of plant science. So my mother was on the bio side, and my father was on the math side. When I went to school as an undergraduate at Cornell University, my sister, Jenny, was also there. She was three years older than I was, and she was studying biochemistry.

Jenny thought that we should take a course together since we were at the same university. And so I said, whatever you want to take is fine by me, and she said something really profound for the time (this was 1976). Jenny said that the future of biochemistry was computational and that we should, therefore, take a computing course. Computing courses were not popular back then, but she had the insight that it would be important. So, we took an introductory programming course – and I got hooked! I think our conversation gave me an appreciation for what computing could do to disciplines other than my own, and, of course, I have an affinity to biology because of my mother and my sister.

And you were a math major, right?

I started out as a piano performance major at Cornell, and realized by the end of the first semester that it wasn't a good major for me. I would walk into a piano lesson, sit down and start crying because I knew I was going to be crying by the end. So I focused on math,

which was my second major, and really enjoyed it -- right up until I took a very abstract course in topology, which I found very difficult to follow because I couldn't visualize it. One day, the professor (who had been scribbling madly on the board writing huge equations) walked over to the window, threw it open, and started barking at a dog. I didn't want to end up like that. So I thought I better choose another major. Computer Science seemed like a good one.

Was he barking at a dog or just barking like a dog?

There was a dog out there, and he was barking like a dog at a dog.

I would do that. I'm famous for doing that. I can work a dog into a frenzy with my bark.

Oh my gosh.

So, you switched to the wrong field, I think.

That's funny.

You've been involved with the Computing Research Association for a long time. What CRA accomplishments are you most proud of?

One of the reasons I've loved working on the Board of Directors of the CRA is that the people on the board are very service-oriented. They truly love computing research, and want to give back to their community. I like this type of person, and I like meeting them outside of my own field of databases. The other thing is that there's a disproportionately large number of women who serve on the board, and I enjoy being able to meet more women in computing.

One of the things that the Computing Research Association is well known for, of course, is the Taulbee Survey, which is widely used by departments for hiring and salary information. The Government Affairs Committee is also crucial, especially in these days when funding for science is becoming more difficult. The advocacy work that the Government Affairs Committee does is really important.

But the CRA also comes out with a number of statements and studies that can be used by the community. The most recent one that I was involved in was about the booming enrollments in computer science, a phenomena that has spread across the country -- the 2x-6x number of students in our courses, and the vast increase in the percentage of non-majors taking our courses. We did a survey, measured what the effect was, and tried to document how institutions were coping. As a result, we produced a report. That report[3] can be used by departments across the country that are trying to argue for more resources because of what they're facing in their enrollments. I think that this is really beneficial to the computer science community.

You had kids while you were still in graduate school. What advice can you offer for those who are trying to decide whether to start a family in grad school?

My advice for people is: start a family when you want to start a family, when it is the right time for you psychologically. The career issues will work themselves out. I chose to start in graduate school, which was risky because I was interviewing when I was pregnant. It was an awkward position to be in.

I started my first job as an assistant professor with a wee baby, which was extremely tiring. But I wanted to have children then, and I did, and I'm very glad that I did.

A lot of women wait until they get tenure, at which point fertility may be an issue; and you might never forgive yourself for not having had a child. So, for me, I preferred to take the risk with my career to regretting having waited.

My advice for people is: start a family when you want to start a family, when the time is right for you psychologically.

Our readers have requested tips from you on handling the balance between work and family life.

Always a difficult one. Interestingly, over the years I've had that question from as many men as women. I have always been very jealous of my nights and weekends, especially when my children were young. So I would work like a maniac during the day, and when I went home, I was with my family, with my children.

[3] Generation CS: CS Undergraduate Enrollments Surge Since 2006" by the CRA Enrollment Committee Institution Subgroup. Available at https://cra.org/data/generation-cs/

Sometimes I'd wake up in the middle of the night and start working, which my graduate students always enjoyed because 2:00 in the morning was when they were still up. So we were both up at 2:00 in the morning and could work on things together (remotely of course).

And weekends also. It is really important to be able to be at events for your children and do things with them. So I've always tried to be very efficient during the workday. I didn't spend a lot of time talking or lollygagging. I was quite focused on getting things done, and for me, that worked.

You've thought a lot about how to engage more women in computer science, including setting up such a program at Penn. What strategies have you found that seem to work well and others might want to use?

First of all, there are a lot of resources out there that we can use as departments, from the NCWIT, Women in Technology organization, to the Computer Research Association CRAW, a subcommittee of the CRA, to the annual Grace Hopper Celebration. There are also all sorts of resources that you can use to get students involved in undergraduate research, which I think is especially important for women.

The strategy that we've been using at Penn is to create a sense of community, so that women don't feel like they're the only ones dealing with the issues that they're struggling with. So we started a pre-orientation program for women coming into Penn so that they can come to campus ahead of time and get to know each other. We set up social meetings during the semester so that they can keep in touch. We provide options for them to be able to give back to the community by going to high schools, talking about computer science and how exciting it is.

We've also adopted strategies in how we teach computer science that seem to be more women-friendly. Peer programming and the ability to collaborate over homework assignments rather than working on them in isolation seems to very appealing to the women. And we've also tried to include in our courses as well as in our outreach events, an understanding of how computer science impacts everyday life. That computer science is not just a nerdish activity, but that it enables all sorts of good things, like discovery in medicine.

What is Dancing with the Professors?

At Penn, there's a Latin and Ballroom Dance student group that engages with faculty by having a competition each year, where they match up a faculty member with one of their student members. And you come up with a two to three-minute dance routine that you perform at the end of the semester.

It's just like Dancing with the Stars, but instead of a celebrity you've got a professor. I decided that I wanted to do it because I have always wanted to learn how to dance, and there's nothing like being given a deadline to force you to learn something.

So, I signed up for it. At the time, I was the Deputy Dean of the Engineering School, and when my Dean found out he was rather negative. He said, "Sue, it's very unprofessional, you know, dancing as a Deputy Dean." But I disagreed. I said, "I think it shows that I'm engaged with the students and that I want to be involved."

I saw this as a challenge, and really enjoyed it because it pushed me way past my comfort zone. I can get up and talk in front of hundreds of people, and it is not an issue for me. But memorizing a three-minute dance routine and performing it in front of 50 people was absolutely terrifying.

We've also adopted strategies in how we teach computer science that seem to be more women-friendly. Peer programming and the ability to collaborate over homework assignments rather than working on them in isolation seems to very appealing to the women.

What dance did you guys do?

We did a swing dance to "Shake a Tail Feather".

Aw, piece of cake, right?

You guys are good?

Do you know how I met my husband?

No.

Ballroom dancing.

Really?

Yeah.

Oh, that's great.

How does sports fit into your life?

When I was growing up, I didn't do any sports at all. But when I started graduate work at Princeton, I needed an escape valve for the pressure that I felt in pursuing my studies. So I took up running, and that has continued pretty much throughout my adult life.

It's always been some sport or other. It's either swimming, biking, running, yoga, strength building, or dancing. I've even taken up flying airplanes, which I don't think of a sport -- it was another crazy thing to try. But I think it's just to relieve some of the pressure that you feel when you're juggling so many different concerns between family and career.

Did you have a problem with injuries?

Only now that I've gotten older. Certainly not when I was younger. The warranty on my body parts expired when I turned 50!

I thought maybe that was why you switched from one to the other over time.

From one sport to the other?

Mm-hmm.

No, I think it's because I have a short attention span. Actually, my favorite sport was sprint triathlons because it's about a half-hour each (running, biking, swimming). You get to do something different every half hour, which is really good.

Do you have any words of advice for fledgling or mid-career database researchers?

The one bit of advice that my father gave me when I was young was: "don't think about it, just do it." And for me, that's been tremendously helpful. If I think about something for too long, very often I can convince myself that I shouldn't do it. Whereas if you go ahead optimistically and do your level best, very often you are successful. Fear of failure is common, especially with women, and prevents you from trying things. But sometimes, even failure is a good thing, and you can learn from it.

So, whether it's a paper that has been rejected from a conference, or a proposal that wasn't funded, or a student who decides they don't want to keep working with you, shake it off and keep going rather than getting depressed about it. I think that the benefit of age is that you've seen that in the past these things have worked out. The acceptance or rejection of a paper or proposal is a bit of a crapshoot. It's not necessarily an indication of the real worth of the idea or of as you as a person. You have to learn from failures as well as from successes.

Okay, so don't overthink it. Is that useful advice for daily life also? Job choice, shopping?

Shopping I can talk about.

Okay.

Usually, when you go shopping, you find something, and you instinctively like it or not. And then you think about it too much and end up walking away -- but then you return the next day to buy it. So I think that overthinking is something that we frequently fall prey to. I mean, look, we're computer scientists, we're analytical. We have to think about things, but overthinking is definitely a trap that we fall into.

Among all your past research, do you have a favorite piece of work?

I think that the work I did in workflow provenance with my postdoc Sarah Cohen-Boulakia is one of my favorites, because we started with real questions that people asking in the scientific community. We developed a beautiful formalism around it, which led to two Ph.D. theses, one by Zhuowei Bao and the other by Sudeepa Roy. Both had topics in their dissertation that were based on ideas from workflow provenance.

It was a very fertile field of work. One of my favorite papers (with Sarah) was on how to "zoom in and out" of provenance. How to abstract out from the details of provenance so that you can get an overview of it, and then how to dive in and see the details. This was a paper that was rejected from both SIGMOD and VLDB, and eventually published in ICDE. It's one of my favorite papers, and I think that it has had a lot of impact. This underscores my point of not taking failures too seriously. Have confidence in what you've done!

If you magically had enough extra time to do one additional thing at work that you're not doing now, what would it be?

If I had more time, I would like to spend time talking with more people across campus about the problems that they're facing related to information gathering, data

management[4], and data analysis. I'd like to understand real problems in areas like sociology, economics, history, law, public policy and all the rest. I'd love to be able to talk to more people but really, it's a question of bandwidth.

If you could change one thing about yourself as a computer science researcher, what would it be?

I would like to be more intellectually curious about other areas in computer science. I would like to be more up on the advances in technology.

But as Department Chair, didn't you have to know all that stuff?

You do. You have to be aware of what the contributions are that your faculty members have made. But I would really like to take the time to go back and deeply understand areas like statistics, machine learning and data mining.

I've always felt that I didn't had the cycles to do this. I know that many of my colleagues manage to make the time, and I think I need to start doing that as well.

Well, thanks very much for talking with us today.

It's been great. Thank you, Marianne.

[4] Editor's note: this is now widely known as "Data Science", but was not when this interview took place.

Report on the Second International Workshop on Semantic Web Meets Health Data Management (SWH 2019)

Haridimos Kondylakis
ICS-FORTH
Heraklion, Greece
kondylak@ics.forth.gr

Kostas Stefanidis
Tampere University
Tampere, Finland
kostas.stefanidis@uta.fi

Praveen Rao
Univ. of Missouri-Columbia
Columbia, USA
praveen.rao@missouri.edu

Dave Parry
Auckland University of Technology
New Zealand
dave.parry@aut.ac.nz

ABSTRACT

The advancements in health-care have brought to the foreground the need for flexible access to health-related information and created an ever-growing demand for efficient data management infrastructures. To this direction, many challenges must be first overcome, enabling seamless, effective and efficient access to several health data sets and novel methods for exploiting the existing information. The second international workshop on semantic Web technologies for health data management aimed at putting together an interdisciplinary audience that is interested in the fields of semantic web, data management and health informatics to discuss the challenges in health-care data management and to propose new solutions for the next generation data-driven health-care systems. In this article, we summarize the outcomes of the workshop, and we present a number of key observations and research directions that emerge.

1. INTRODUCTION

Precision medicine is the next frontier for research and innovation in healthcare. It deals with treatment and prevention of diseases by taking into account the genetic makeup, environmental and lifestyle factors of an individual [2]. As a result, medical professionals can precisely prevent and treat diseases rather than using a "one-size fits all" approach.

Key in achieving the vision of precision medicine as well as affordable, less intrusive and more personalized care, is to efficiently and effectively harness the value of healthcare data to gain meaningful insights. Ultimately this has the potential to improve patient outcomes, increase the quality of life of patients, and lower mortality. Another important benefit is the potential to lower healthcare costs and reduce medical errors. Electronic health records (EHR) of patients are rich and complex and contain hundreds of attributes [1]. An EHR contains data about a patient's medical history, demographics, diagnosis, medications, allergies, radiology images, lab test results, and other pertinent information. In addition, healthcare data exists in many different formats, from textual documents and web tables to well-defined relational data and APIs. Furthermore, they pertain to ambiguous semantics and quality standards resulted from different collection processes across sites. Data pertaining to healthcare can also be found on social media through healthcare conversations, in wearables and monitoring devices that continuously stream information about a person's fitness and health.

Much effort has been spent in developing interoperability standards for healthcare systems over the last few decades. HL7's Fast Healthcare Interoperability Resources (FHIR) [10] is emerging as a popular standard for healthcare data exchange and developing new applications. In fact, FHIR supports Semantic Web technologies such as the Resource Description Framework (RDF) and SPARQL. Thus, Semantic Web technologies can provide effective solutions for enabling interoperability and common language among healthcare systems, and can lead to the disambiguation of the information through the adoption of various terminologies and ontologies available. In addition, artificial intelligence (AI) and machine learning can enable data-driven decision making and extracting meaningful insights

from complex healthcare datasets. Thus, knowledge representation and reasoning on healthcare data become even more important. Semantic Web technologies have matured over the years and can provide these capabilities by design.

The goal of International Workshop on Semantic Web Meets Health Data Management (SWH) is to bring together researchers cross-cutting the fields of Semantic Web, data science, data management, and health informatics to discuss the challenges in healthcare data management and to propose novel and practical solutions for the next generation of data-driven healthcare systems. Developing optimal frameworks for integrating, curating and sharing large volumes of EHR data has the potential for a tremendous impact on healthcare, enabling better outcomes at a lower and affordable cost. The ultimate goal is to enable new innovations in Semantic Web, knowledge management, and data management for healthcare systems to move the needle to achieve the vision of precision medicine.

Next, we summarize the outcomes of the second workshop instance held in conjunction with the 18^{th} International Semantic Web Conference (ISWC 2019) in Auckland, New Zealand.[1]

2. INVITED TALKS

2.1 Semantic AI for Healthcare

Explainable Artificial Intelligence (XAI) aims at explaining the algorithmic decisions of AI solutions with non-technical terms in order to make these decision trusted and easily understandable by humans [3]. HORUS.AI [6] adopts XAI within the healthcare domain based on logical reasoning that supports the monitoring of users' behaviors and persuades them to follow healthy lifestyles.

Specifically, HORUS.AI is an AI-based system built upon the integration of semantic web technologies and persuasive techniques for motivating people to adopt healthy lifestyle or for supporting them to cope with the self-management of chronic diseases. The system collects data from users' devices, explicit users' inputs, or from the external environment (e.g., facts of the world) and interacts with users by using a goal-based metaphor. Interactive dialogues are used for proposing set of challenges to users that, through a mobile application, are able to provide the required information and to receive contextual motivational messages helping them to achieve the proposed goals.

[1]For a summary of the first instance of SWH, please refer to [15].

2.2 Personal Consent in Data Management

Semantic web technologies are inherently suitable to serve the role of providing the common shared vocabularies for data sharing intentions and agreements, together with the algorithmic machinery that is needed to process these agreements. Nowadays, there are several approaches that use knowledge graphs to express aspects of data sharing agreements, building on top of more general schemas used to describe persons, personal data, or even healthcare and medical imaging metadata.

These knowledge graphs are great steps towards a vision where users or parties encode their preferences and intentions of data usage in a machine process-able way and data processing algorithms automatically respect these preferences. In order to achieve this, the developed vocabularies have to be backed by the development of generic and re-applicable algorithms; possibly borrowing from data integration [4, 12] or ontology based query answering [14].

3. PAPER PRESENTATIONS

3.1 Dialogue Management in Healthcare

The development of methods that implement automated planning to manipulate human-machine dialogue is still in its early stages, but it has gained attention in recent years (e.g., [13]). [19] proposes a novel approach for supporting dialogues. The novelty of the approach has to do with the combination of reasoning and planning for supporting dialogues. These two techniques allow to dynamically update the behavior of conversational agents based on the data provided by users. The reasoner is responsible for inferring the most suitable status of a user (or patient). This activity is performed by exploiting not only the user data and the integrated conceptual model, but also the proper resources of the Linked Open Data cloud. On the other hand, the planner generates the interactions for supporting a multi-turn conversation with users in order to acquire the missing information enabling the classification of the users' status.

3.2 Self-Management of Diabetes Patients

The interest in designing smart platforms for supporting the self-management of chronic diseases significantly growths in the last years. One of the chronic diseases that most attracted the attention of the research community is diabetes. [7] presents the TreC-Diabetes system, a smart platform aiming to create a continuous link between clinicians

and patients for supporting the self-management of diabetes. The novelty of the approach lies in supporting real-time stream reasoning of information provided by patients (e.g. glycemic index, food intake, sport activities) to detect possible critical situations and to inform clinicians about them.

3.3 Education and Emotion Based Semantic Recommendations for Health

FairGRecs [18, 17] is a system focusing on recommending interesting health documents selected by health professionals, to groups of users, incorporating the notion of fairness [16], using a collaborative filtering approach. It is the first time such technologies are combined for health recommendations. The overall approach is based on a notion of semantic distance between documents and user profiles. The goal is to offer a list of recommendations to a caregiver who is responsible for a group of patients. The recommended documents need to be relevant to the patients profiles, i.e., to the patients personal health-care records (PHR). However recommendation algorithms so far ignore the fact that patients profiles are multifaceted. For example, recommending the proper document should not only focus on the patients relevant problems but also on their health literacy (namely, the ability to obtain, read, understand, and use health care information in order to make appropriate health decisions and follow instructions for treatment), educational level and psychoemotional status, as emotions can greatly affect the cognitive processes. [11] explores these dimensions, as well paving the way for a new system incorporating all aforementioned aspects.

3.4 From Chronic Diseases to Behavior Change

HeLiS [5] is an ontology aiming to provide in tandem a representation of both the food and physical activity domains. [8] presents two extensions of HeLiS modeling for the first time information about food risk levels and self-management barriers. As such, the first extension provides a conceptual model representing the risk level of the food categories already defined in HeLiS associated with the onset or worsening of the most common five chronic diseases (i.e., diabetes, kidney diseases, cardiovascular diseases, hypertension, and obesity). The second extension provides an abstract layer of a conceptual model representing the barriers that a user may encounter during the self-management of his/her lifestyle or of his/her chronic disease (e.g., knowledge representing why a diabetes patient is not able to check his/her glycemia constantly).

3.5 Modeling Context in Knowledge Graphs of Diagnostic Reports

Typically, the NLP-based informatics pipelines that target at converting free text to structured text, lack the ability to recover and convey implicit information, found in diagnostic reports. Such information is readily perceived and taken into account by a human reader. [9] develops a unique method in terms of modeling such contextual information for recovering implicit relationships among structurized diagnostic entities. This method enables structurization of contextual information into a cohesive and holistic representation of free text diagnostic reports. Specifically, for doing so, [9] models the context of a diagnostic report in relational triple resource description framework RDF-like format, which is the building block of the model's knowledge base. Triples that share subject or object induce a graph linked using the n-ary relation schema of the semantic web.

4. CONCLUSIONS

A number of key observations and research directions emerged in the discussions that we summarize below.

- Although recent technological advancements allow data collection from personal devices, off-the-shelf wearable sensors, and external sources, exploiting these data requires combining and reasoning on a considerable amount of knowledge from different domains (e.g. user attitudes, preferences and environmental conditions, etc.). Semantic technology is a key to this purpose. Besides structured data, semantic data integration should be generalized to unstructured information as well (e.g. discharge letters, pathology reports etc.) as still, such information is widely used in the healthcare domain capturing essential information.
- This semantic integration besides static should also include dynamic data, as multiple streams of data such as glycemic index, food intake, and performed sport activities, constantly arrive and their processing could highly benefit CDS systems.
- In order to generate effective personalized health recommendations, traditional recommendation approaches are not enough. Contextual, psychological and other information should be considered as well, motivating people to adopt healthy lifestyle and better management of their chronic conditions. To this direction persuasion techniques could also be exploited, whereas explainable AI and more

specifically explainable recommendations will pave the way for systems that end-users will actually trust and use daily.

- Another interesting direction in the health domain is dialogue management. Task-oriented dialogues can give advice to patients, offering guidance on the patient's treatment. However, in dialogue systems for the healthcare domain, making the right question at the right moment is a relevant challenge, whereas efficient and effective semantic reasoning are required for providing intelligent discussions.
- Finally, as the usage of personal data is key to in achieving the vision of precision medicine, methods are required to describe smart contracts of data usage in a formal, machine-processable language. Semantic Web technologies can have a central role in this approach by providing the formal tools and languages required.

This second instance of the Semantic Web Meets Health Data Management Workshop made clear that a lot of research work still needs to be done in the area of semantic health data management. Given the growing interest in industry and academia, the third version of the workshop will be held in Athens along with ISWC 2020 in Athens, Greece[2], with a renewed list of topics such as explainable AI in health through semantics, blockchain solutions etc.

5. REFERENCES

[1] What information does an electronic health record (EHR) contain?, 2019. https://www.healthit.gov/faq/what-information-does-electronic-health-record-ehr-contain.

[2] What is precision medicine?, 2019. `https://ghr.nlm.nih.gov/primer/precisionmedicine/definition`.

[3] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[4] V. Christophides, V. Efthymiou, and K. Stefanidis. *Entity Resolution in the Web of Data*. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers, 2015.

[5] M. Dragoni, T. Bailoni, R. Maimone, and C. Eccher. Helis: An ontology for supporting healthy lifestyles. In *ISWC*, 2018.

[6] M. Dragoni, T. Bailoni, R. Maimone, M. Marchesoni, and C. Eccher. HORUS.AI - A knowledge-based solution supporting health persuasive self-monitoring. In *ISWC*, 2018.

[7] M. Dragoni, C. Eccher, S. Forti, S. Puccini, B. Purin, and A. Valentini. Trec diabetes: A semantic platform for supporting the self-management of patients. In *SWH*, 2019.

[8] M. Dragoni and V. Tamma. Extending helis: From chronic diseases to behavior change. In *SWH*, 2019.

[9] P. S. Giannaris, C. Tang, O. Kholod, S. Hanson, C. Shyu, R. Hammer, D. Xu, and D. Shin. Modeling of contextual information in knowledge graphs of diagnostic reports. In *SWH*, 2019.

[10] HL7. Fast healthcare interoperability resources, 2019. `http://hl7.org/fhir`.

[11] H. Kondylakis and K. Stefanidis. Towards education and emotion based semantic group recommendations for health. In *SWH*, 2019.

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

[13] K. Lee, Y. S. Lee, and Y. Nam. A model of fsm-based planner and dialogue supporting system for emergency call services. *The Journal of Supercomputing*, 74(9):4603–4612, 2018.

[14] H. Pérez-Urbina, E. Rodríguez-Díaz, M. Grove, G. Konstantinidis, and E. Sirin. Evaluation of query rewriting approaches for OWL 2. In *SSWS+HPCSW*, 2012.

[15] K. Stefanidis, H. Kondylakis, and P. Rao, editors. *Proceedings of the First International Workshop on Semantic Web Technologies for Health Data Management, SWH 2018*, volume 2164 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

[16] M. Stratigi, H. Kondylakis, and K. Stefanidis. Fairness in group recommendations in the health domain. In *ICDE*, pages 1481–1488, 2017.

[17] M. Stratigi, H. Kondylakis, and K. Stefanidis. The fairgrecs dataset: A dataset for producing health-related recommendations. In *SWH*, 2018.

[18] M. Stratigi, H. Kondylakis, and K. Stefanidis. Fairgrecs: Fair group recommendations by exploiting personal health information. In *DEXA*, 2018.

[19] M. S. Teixeira, M. Dragoni, and C. Eccher. A planning strategy for dialogue management in healthcare. In *SWH*, 2019.

[2] `https://sites.google.com/view/swh2020`

ACM SIGMOD is committed to ensuring that all SIGMOD activities are carried out in an inclusive and diverse environment with zero tolerance for discrimination, harassment, or any other form of misconduct. DBCares, a database-community-wide initiative, co-started with the VLDB Endowment, is a realization of this principle:
https://sigmod.org/sigmod-policies/dbcares-policy

We cannot be satisfied with continuing the status quo. We must actively stand against discrimination. We will strive to find new ways to address the inequities that exist in our field and create an environment that is more welcoming, just, and equitable to all.

SIGMOD Executive Committee Position on Racism

We join ACM, CRA and VLDB Endowment in stating that we will continue to listen, to learn, to engage and to explore new ways to fight against and reject racism. Silence perpetuates, doubt reinforces, and rationalization of incident after incident only compounds the pain so many in our society continue to endure. This is especially important for the SIGMOD community, given that the computer science field has historically lacked diversity.

We know that racism:

- Is systemic and institutionalized.
- Continues to oppress people of color around the world – denying basic human rights, denying opportunity, and even more tragically denying many of their very lives.
- Is learned behavior that may be unlearned through education, compassion, empathy, and action.
- Drives a wedge between communities, and in doing so limits the quest for a society steeped in respect.
- Benefits the privileged from its existence, who must be willing to sacrifice to overcome it.

To stand against it, we:

- Acknowledge the existence of racism within our communities and commit to defeating it.
- Call out and reject rationalization of incidents and distortion of information.
- Educate ourselves and those around us to address racism in its many forms.
- Stand up against the status quo by using our voice and agency.
- Invite the community to discuss concrete steps and commit resources to create lasting change.

SIGMOD 2021 CALL FOR RESEARCH PAPERS

Xi'an, Shaanxi, China, June 20-25, 2021, https://2021.sigmod.org

The annual ACM SIGMOD conference is a leading international forum for data management researchers, practitioners, developers, and users to explore cutting-edge ideas and results, and to exchange techniques, tools, and experiences.

There are three paper categories in the research track in SIGMOD 2021:

- **Data Management**
  We invite the submission of original research contributions relating to all aspects of data management.

- **Data Science and Engineering**
  We invite the submission of original research in data science and engineering, inspired by real applications. Such papers are expected to focus on data-intensive components of data science pipelines; and solve problems in areas of interest to the community (e.g., data curation, optimization, performance, storage, systems).

- **Applications**
  We invite the submission of novel applications of data management systems and technologies from outside the core data management community (e.g., astronomy, computer graphics, computer networking, genomics).

HIGHLIGHTS

- Paper submission deadlines: Tue September 22, 2020 (Round 2)

- Submission website: https://cmt3.research.microsoft.com/SIGMOD2021 (open for submission starting September 8, 2020 for Round 2).

- Submissions must use the latest ACM format in the default 9pt font.

- Data Management submissions must be at most 12 pages plus unlimited number of pages for citations.

- Data Science and Engineering submissions must be at most 8 pages plus unlimited number of pages for citations.

- Applications submissions must be at most 4 pages plus unlimited number of pages for citations.