CRUX: Crowdsourced Materials Science Resource and Workflow Exploration

Mengying Wang
Case Western Reserve University
Cleveland, Ohio, USA
mxw767@case.edu

Hanchao Ma
Case Western Reserve University
Cleveland, Ohio, USA
hxm382@case.edu

Abhishek Daundkar
Case Western Reserve University
Cleveland, Ohio, USA
aad157@case.edu

Sheng Guan
Case Western Reserve University
Cleveland, Ohio, USA
sxg967@case.edu

Yiyang Bian
Case Western Reserve University
Cleveland, Ohio, USA
yxb227@case.edu

Alp Sehirlioglu
Yinghui Wu
axs461@case.edu
yxw1650@case.edu
Case Western Reserve University
Cleveland, Ohio, USA

ABSTRACT

Modern multidisciplinary materials science routinely processes scientific workflows that integrate different data resources (*e.g.,* X-ray data, scripts, analytical results). Most of such data resources are isolated in research labs, created ad-hocly, and remain underutilized. We demonstrate **CRUX**, a **C**rowdsourced platform for materials data **R**eso**U**rces and workflow e**X**ploration. CRUX is empowered by coherent data-workflow modeling, knowledge-based resource assembly for workflow search, and data provenance to support workflow exploration. CRUX allows users to declare parameterized workflows as graph patterns, and automatically recommends crowdsourced resources with quality guarantees. We demonstrate the ease-of-use and the performance of CRUX with three categories of queries: data search, workflow recommendation, and resource exploration. We make case of CRUX for peak finding in X-ray Diffraction (XRD) data, a cornerstone task in materials research. We show that CRUX enables new interactive paradigms to explore and design workflows for data analysts in general.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Data management systems**.

KEYWORDS

Knowledge Graph, Graph Search, Scientific Workflows

ACM Reference Format:
Mengying Wang, Hanchao Ma, Abhishek Daundkar, Sheng Guan, Yiyang Bian, Alp Sehirlioglu, and Yinghui Wu. 2022. CRUX: Crowdsourced Materials Science Resource and Workflow Exploration. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557194

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557194

1 INTRODUCTION

Data-driven Materials discovery has been on the forefront of materials science as well as other basic science fields such as chemistry and physics, highlighted by the Materials Genome Initiative that started in 2011[20]. Materials scientific workflows has been accelerated by machine learning approaches focused on a specific material space, feature, property or application [5, 13, 19, 21]. The success to materials discovery strongly depends on the input data for training and testing models [8]. Despite the need for data-driven materials research, many high-value materials datasets where the targeted structure was not attained or the results were not reported are often unpublished and never available for public use.

Challenges. Nevertheless, to utilize these "failed good data" and scientific scripts that routinely access them, there are several challenges that existing data infrastructure[7] cannot address.

Isolated Data and Workflows. (Unpublished) materials data and scripts are mostly kept in "silos", created ad-hocly for individual workflows, and are often stored separately from the workflows that accessed them. Many data platforms like AFLOW[6] and OQMD[18] collected massive datasets from various materials compounds. However, data material research requires holistic methods to discover and integrate useful data resources and streamline them to analytical pipelines. This calls for coherent, domain-specific data models to link data, scripts and analytics results for workflow completion.

Data discovery for Scientific Workflows. Materials analysis often requests proper data for experimental pipelines. For example, a query "*What is a proper value for deposition of PLD thin film in the deposition process of this microstructure-explicit simulation?*" requests measurement data that reports proper deposition values. Existing data platforms require structured query languages (e.g., SQL) to specify such data, which is hard to write explicitly, or keywords, like the Material Project[10], which can be ambiguous and lead to irrelevant results. These call for expressive and user-friendly search mechanism to channel data directly to actionable workflows.

Explore Workflow design space. Materials community also ask to clarify the analytical results with follow-up "Why", "What" and

"How" questions. For example, the above user may further ask "*What if I need to introduce a dopant?*" Conventional "query-response" mechanism does not support an iterative exploratory process to explore search results and the design space of multi-phased scientific workflows[12]. Recently, some materials workflow management tools are sprouted (e.g., Aiida[9]), but for all we know, none of them analyze the resulting workflow.

CRUX. Motived by these, and inspired by our prior work in ML-based material science [4], knowledge based search [17] and provenance [15, 16], we developed CRUX, a crowdsourced materials data infrastructure to curate and recommend high-value unpublished materials data for the need of the materials community. CRUX has the following unique components.

Coherent Data-Workflow Modeling. CRUX is empowered by a materials knowledge graph CRUX-KB, a three-layered network of materials-relevant entities and their semantic relationships, along with an information extraction engine (CRUX-IE) to profile and extracts facts from the shared raw data resources and metadata. CRUX-KB involves three types of entities: source (data contributors), resources (dataset, scientific scripts, tasks), and facts (materials properties, calculated properties). This interlinks isolated data resources with unified representation and makes them discoverable by workflows with specified tasks.

"Workflow-centric" Query Answering. CRUX provides a query engine CRUX-Q to process queries with "workflow-centric" dataset search. Unlike existing services that stop at returning a list of datasets [3], CRUX provides (1) data resources along with the context of matching scientific scripts, tasks and historical analysis results, (2) a recommendation of integrated view of dataset, scripts and tests that best fit a user-defined workflow description. None of existing platform supports such workflow-centric search.

Exploratory Search. Beyond data search, CRUX supports a class of "Why" and "What-if" queries, all expressed by CRUX-Q query semantics. The "Why"-analysis tracks the data resources, scripts and tests that are responsible for the occurrence of relevant analytical results of materials-specific workflows, and "What-if" analysis discovers new relevant data resources upon changes of workflow specification. This makes CRUX unique in exploring workflow design space beyond data search.

Visual Interfaces. CRUX provides user-friendly Web interfaces to support data upload and sharing, visual query and workflow construction, and visual result exploration.

CRUX community. Currently, CRUX is specialized to XRD data, scripts and XRD-based workflows. These resources are contributed by a group of CRUX community of 6 collaborative institutions. CRUX community also includes International Centre for Diffraction Data (ICDD) and JADE database (with active users from 53 countries). CRUX platform will readily be supporting other materials data (*e.g.,* images, videos, etc) as the community grows.

We next introduce the general framework (Section 2) and the system architecture(Section 3) of CRUX. We also demonstrated scenarios in Section 4 to show how the key components work.

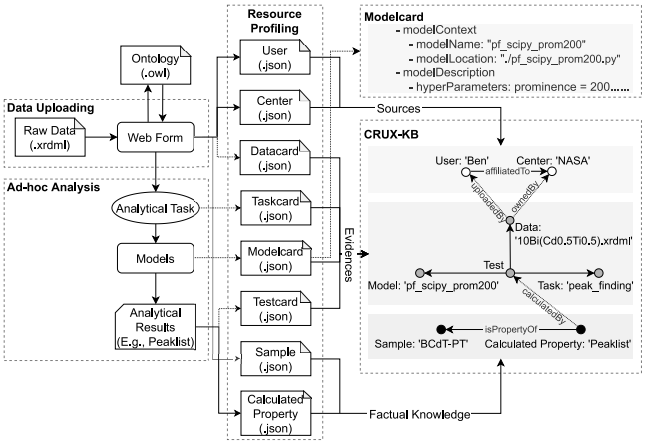

Figure 1: CRUX: "Materials Data-to-Knowledge" Framework (specifying XRD-based peak finding)

2 FRAMEWORK OVERVIEW

2.1 From Raw Data to Factual Knowledge

We start with an overview of an end-to-end "data to facts" framework of CRUX. A framework featuring XRD and peak analysis is illustrated in Fig 1. (1) CRUX collects raw materials data resources via Web forms or from materials database export services (such as JADE database, not shown). Users upload raw materials data resources such as (a) raw (XRD) data files (in *e.g.,* ".xrdml" or ".raw" format), (2) Python scripts that processes XRD-based analysis (*e.g.,* peak finding tools), and (3) auxiliary analytical results, such as peaks, generated from any "ad-hoc" routinely performed analysis. (2) CRUX-IE automatically performs entity extraction and relation inference, recognizing 8 types of entities and relations among them (see CRUX-IE). These data resources are profiled uniformly as JSON objects, validated by a built-in materials ontology CRUX-Onto, featurized to attributed entities and triples, and integrated to be a fraction of CRUX-KB (see CRUX-KB). (3) CRUX-KB will be consulted for query evaluation (see CRUX-Q).

We will demonstrate the following unique components of CRUX.

Multi-layered Materials Knowledge Graph. CRUX-KB is a *three-tier knowledge graph model*, which is a hierarchical network representation of the following three types of entities: materials (*e.g.,* elements, atoms, components), resources (*e.g.,* datasets, analytics files, ML (Python) scripts), and *source* (*e.g.,* experimentalist, universities, organizations, companies). CRUX-KB consists of a set of *materials statements* that describe factual knowledge from materials analysis. (2) A support relation connects a statement or a materials entity and a supporting *resource* entity. A resource entity may refer to diffraction dataset, or a variety of analysis result based on the analysis obtained from *e.g.,* JADE software or Rietveld analysis. (3) For each resource entity, CRUX-KB also records its sources or ownership entities whenever available (*e.g.,*, authors, contributors or websites), connected by *e.g.,* "uploadedBy" or "ownedBy" relation.

"Data-to-facts" Flow. X-ray Diffraction (XRD) data come with widely adopted format such as ".raw" or ".xrdml". CRUX-IE implements an automated flow that directly transform XRD data, along with analytical results of "ad-hoc", routinely performed analysis such as peak finding, into data objects and triples to enrich

CRUX-KB. (1) It profiles raw files into 8 types of "cards": source, resource (data, task, model, test), and factual cards (*e.g.,* calculated property). Each card is a JSON object with key-value pairs. An example of a model card for "pf_scipy_prom200" is illustrated in Fig. 1. (2) CRUX-IE then constructs and infers relations among the card entities. This is validated by a built in materials ontology CRUX-Onto. CRUX-Onto integrates manually designed domain-aware materials ontologies, integrating fractions of existing ones [11]. This pipeline is implemented by the built in API library.

Example 2.1. A user uploaded a data file "$10Bi(Cd0.5TI0.5)$.xrdml" from an experiment. This file conforms to XML syntax, with user-defined elements specifying experimental settings (*e.g.,* temperature, processing method). Additional metadata are specified via online forms (see "Interface"). CRUX-IE invokes built-in APIs to parse the file and Web form inputs into attributed entities, and creates card entities (JSON objects) accordingly. It stores the raw files and JSON objects on GCP, and extracts facts among the cards. For example, it encodes statements such as "*Ben from NASA contributed data file '10Bi(Cd0.5TI0.5).xrdml'* ", or "*ML script 'pf_scipy_prom200' is used to perform 'peak finding' task over dataset '10Bi(Cd0.5TI0.5).xrdml' with result file 'Peaklist'*". This generates a graph of 8 entities and 8 edges to be integrated into CRUX-KB, as shown in Fig. 1.

2.2 From Factual Knowledge to Workflows

CRUX **Queries**. CRUX adopts a class of *graph pattern queries* as native queries. A CRUX query Q is a connected graph (V_Q, E_Q, L_Q, T_Q), where V_Q (resp. $E_Q \subseteq V_Q \times V_Q$) is a set of pattern nodes (resp. pattern edges). Each pattern node $u \in V_Q$ (resp. pattern edge $e \in V_E$) has a class label $L_Q(u)$ (resp. relation $L_Q(e)$). For each node $u \in V_Q$, $T_Q(u)$ is a set of literals. A literal is in the form of $u.A$ op c, where op is from $\{>, >=, =, <=, <\}$, and c is a constant. A pattern node can be a designated "output node", clarifying the entities to be returned as matches via graph pattern matching.

Workflow-centric Querying. To help users find data resources without writing complex queries, CRUX-Q, the query engine of CRUX, supports a "workflow-centric" search in two modes.

(1) For novice users who are familiar with keyword search only, CRUX-Q will directly transform a keyword query into a subgraph of CRUX-KB. Built on our prior study [14, 23], CRUX learns a conditional random field (CRF) model to (a) transform keyword terms to a set of card entities, (b) derives a set of edges that can best connect them, which conform to CRUX-Onto to ensure the semantic relevancy, and (c) invokes a minimum spanning tree algorithm to expand the answer to include closely relevant entities including tasks, models, source and factual information, to help users understand the context for workflow design.

(2) CRUX provides a built-in YAML workflow syntax for professional users to declare workflow template with variables. Users can declare a YAML file with "placeholders". CRUX-Q (a) parses the YAML file into CRUX graph queries (as a directed acyclic graph), and (b) nontrivially extends top-k subgraph search *e.g.,* [14, 22] to return k instantiated workflows by replacing all the variable with matching data resources or scripts. Here the answer quality is determined by semantic closeness derived from CRUX-Onto, or learned from the features of model and data cards (node embeddings) [23].

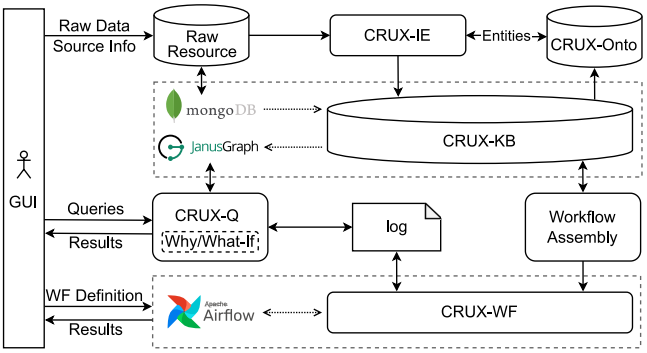

Figure 2: CRUX Architecture

Better still, when the scripts are available, CRUX automate the workflow assembly process by interacting with existing workflow tools such as Airflow to assemble and execute the workflow to directly generates the results of the constructed workflow.

Workflow Exploration. CRUX-Q supports interactive search sessions to allow users explore query results, by posing a class of "Why" and "What-if" queries. (a) Users can specify a "Why"-question by selecting a particular entity v in the query result, which asks "Why this entity (unexpectedly) occurs in my query result?" CRUX-Q will highlight a set of entities and relations from CRUX-KB that are responsible for the occurrence of the entity. In a nutshell, CRUX-Q computes a minimal set of triples, such that if removed from CRUX-KB, v is gone from the result of the same query. (b) For "What-if" analysis, users can either suggest a specific entity (*e.g.,* a newly uploaded dataset) or change the workflow declaration (query), stating "What if the workflow setting is changed?". CRUX-Q addresses the former by identifying the CRUX-KB entities and triples that are to be investigated to include the the entity in the query result; for the latter, it updates the search result to suggest a new set of entities to be investigated.

3 SYSTEM ARCHITECTURE

The architecture of CRUX is depicted in Fig. 2.

(1) At the core of the platform is the curated materials knowledge graph CRUX-KB. CRUX-KB is maintained as property graph over JanusGraph. The entities are profiled and separately stored over MongoDB as JSON objects, where their location are stored as URI in the cards entities in CRUX-KB. This is to reduce the overhead of loading content-heavy raw data objects, and provide "lightweighted" access via card objects by default. The loading of the original data file or script is only triggered when the answer of a CRUX query (*e.g.,* an instantiated workflow) is requested to be executed.

(2) CRUX-IE transforms the shared data resources to curated knowledge graph CRUX-KB, and consult CRUX-Onto for validation.

(3) Underlying CRUX-Q is (a) a query parser that transform keywords or workflow declaration to native CRUX query, and (b) a wrapper that invokes JanusGraph and Gremlin to produce results for CRUX graph pattern query. The parser and wrapper are supported by scripts from the built in CRUX API library (not shown).

(4) The workflow tier of CRUX (CRUX-WF) is responsible for workflow declaration and execution. (a) CRUX-WF interacts with workflow assembly that instantiate the workflow query results into

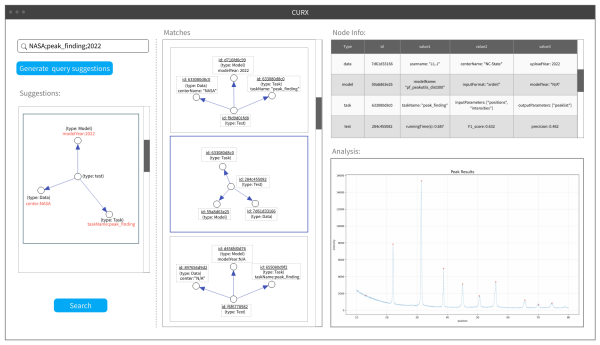

Figure 3: CRUX Interface: Materials Data Search (Keywords)

executable workflows. (b) For the instantiated workflow, it invokes Apache Airflow, a workflow execution tool, to generate results (*e.g.*, peak files) with available scripts and datasets.

(5) CRUX stores large raw data files in Google Cloud Platform. The query and workflows are tracked by the logging system for query processing and optimization.

4 DEMOSTRATION OVERVIEW

The demonstration consists of the following. (1) We demonstrate the "one-click" upload and transformation of raw XRD files. We show the visualized curated knowledge graph CRUX-KB and CRUX-Onto to illustrate the coherent data resource representation. (2) We illustrate CRUX query processing, for both novice users, with keyword search, and professional users with YAML coding. (3) We illustrate the "Why" and "What-if" search with interactive sessions, where users can track responsible resources and explore new data resources, specified for finding the peaks in XRD-based analysis.

Environment. The prototype CRUX [2] is implemented in Java and Python. We demonstrate CRUX in a cluster of Linux servers, equipped with Intel Core i7 processor with 2.6 GHz and 16G memory. An online interface is available [1].

"One-click" Upload. We will start a walkthrough of the end-to-end "data-to-knowledge" framework. We start by clarifying X-ray Diffraction Data, data format of XRDML, and their common analytical tasks and roles in materials research. We then introduce the upload interface (not shown). Users can share a raw XRDML file with a simple "one-click" manner, with the option to fill in additional metadata including temperature, atmosphere, among other conditions. The interface of CRUX has built-in term suggestion function, based on crowedsourced upload history and log files.

"Data-to-Knowledge". We will then illustrate the visualized materials ontology CRUX-Onto, including its components that are used to validate source, resource and factual statements. In accordance, we will illustrate the design of the 8 types of cards of CRUX-KB, and demonstrate their interactions. We finally execute the built in pipeline to show how CRUX transforms a raw XRDML file into a corresponding fraction of knowledge graphs via "one-click".

Data Discovery. We invite the audience to experience CRUX query interface with both modes. In the novice user mode, users can input a set of keyword terms and view the suggested graph pattern query. The professional users can directly use YAML-based workflow templates, and the suggested workflow queries as DAGs.

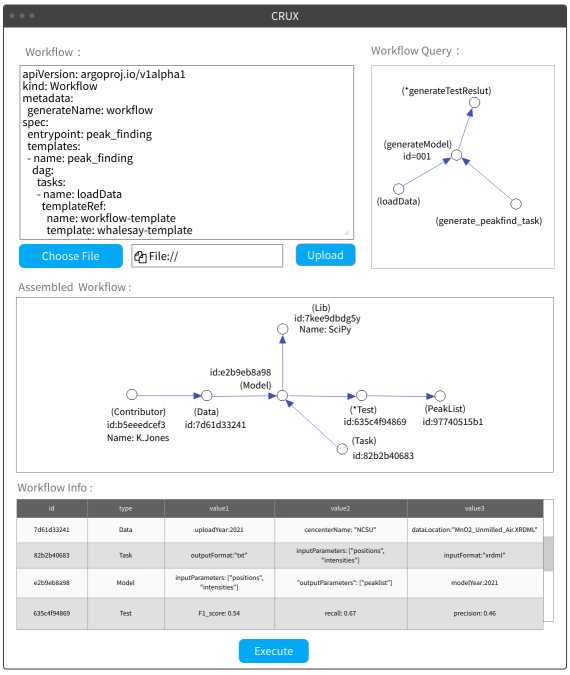

Figure 4: CRUX Interface: Workflow Queries (with YAML)

Example 4.1. As illustrated in Fig. 3, a list of matched subgraphs are returned for a keyword query "*NASA, peak_finding, 2022*". CRUX interprets the query as to "*find the tests that use scripts (model) uploaded in 2022 that perform peak finding over XRD data shared by contributors from NASA*". Three subgraphs are returned. (a) The first is returned due to its closest description given the keywords. (b) The second discovers a 'test' with models that has no "year" information, yet still serves as a proper match as it accessed a dataset that is uploaded in 2022 with additional contributor's information. (c) The third has lowest relevancy due to more missing information.

Fig. 4 illustrates an example of a workflow template, with a visualized CRUX query shown on its right. The matched result is an instantiated workflow as illustrated in Fig. 4. for user to browse.

Workflow Exploration. We will demonstrate the interactive search sessions to allow users investigate the changes of the search results upon the specified entities for "Why" or "What-if" analysis. Continuing with the above example, a user may choose a model script and specify a "What-if" question with a dataset ("*What if my dataset is changed to a new one*"), to explore more shared datasets and scripts.

Example 4.2. We demonstrate an scenario to recommend pretrained models (scripts) for new XRD data without retraining or inference. A workflow query is suggested that (1) load datasets that are pre-processed by GSAS-II similar to the uploaded one, (2) recommend k (k=20) models that are used to process the datasets for the task "peak finding", with hightest accuracy. With a retesting of the pretrained models, CRUX is able to achieve at least 70% in terms of both precision@k and recall@k, without retesting models.

ACKNOWLEDGMENTS

This work is supported by NSF under CNS-1932574, OIA-1937143, ECCS-1933279, CNS-2028748 and OAC-2104007.

REFERENCES

[1] 2022. CRUX Online Demo. https://CRUX.hcma.repl.co.

[2] 2022. CRUX Source Code. https://github.com/crux-project.

[3] 2022. Materials Data Hub. https://materialhub.org/.

[4] Prasanna V Balachandran, Benjamin Kowalski, Alp Sehirlioglu, and Turab Lookman. 2018. Experimental search for high-temperature ferroelectric perovskites guided by two-step machine learning. *Nature communications* 9, 1 (2018), 1–9.

[5] Chi Chen, Yunxing Zuo, Weike Ye, Xiangguo Li, Zhi Deng, and Shyue Ping Ong. 2020. A critical review of machine learning of energy materials. *Advanced Energy Materials* 10, 8 (2020), 1903242.

[6] Stefano Curtarolo, Wahyu Setyawan, Gus LW Hart, Michal Jahnatek, Roman V Chepulskii, Richard H Taylor, Shidong Wang, Junkai Xue, Kesong Yang, Ohad Levy, et al. 2012. AFLOW: An automatic framework for high-throughput materials discovery. *Computational Materials Science* 58 (2012), 218–226.

[7] Lauri Himanen, Amber Geurts, Adam Stuart Foster, and Patrick Rinke. 2019. Data-driven materials science: status, challenges, and perspectives. *Advanced Science* 6, 21 (2019).

[8] Yang Hong, Bo Hou, Hengle Jiang, and Jingchao Zhang. 2020. Machine learning and artificial neural network accelerated computational discoveries in materials science. *Wiley Interdisciplinary Reviews: Computational Molecular Science* 10, 3 (2020), e1450.

[9] Sebastiaan P Huber, Spyros Zoupanos, Martin Uhrin, Leopold Talirz, Leonid Kahle, Rico Häuselmann, Dominik Gresch, Tiziano Müller, Aliaksandr V Yakutovich, Casper W Andersen, et al. 2020. AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Scientific data* 7, 1 (2020), 1–18.

[10] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. 2013. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *Apl Materials* 1, 1 (2013), 011002.

[11] Huanyu Li, Rickard Armiento, and Patrick Lambrix. 2020. An ontology for the materials design domain. In *ISWC*.

[12] Ji Liu, Esther Pacitti, Patrick Valduriez, and Marta Mattoso. 2015. A survey of data-intensive scientific workflow management. *Journal of Grid Computing* 13, 4 (2015), 457–493.

[13] Yingli Liu, Chen Niu, Zhuo Wang, Yong Gan, Yan Zhu, Shuhong Sun, and Tao Shen. 2020. Machine learning in materials genome initiative: A review. *Journal of Materials Science & Technology* (2020).

[14] Hanchao Ma, Morteza Alipourlangouri, Yinghui Wu, Fei Chiang, and Jiaxing Pi. 2019. Ontology-based entity matching in attributed graphs. *Proceedings of the VLDB Endowment* 12, 10 (2019), 1195–1207.

[15] Mohammad Hossein Namaki, Qi Song, and Yinghui Wu. 2019. NAVIGATE: Explainable Visual Graph Exploration by Examples. In *Proceedings of the 2019 International Conference on Management of Data*.

[16] Mohammad Hossein Namaki, Qi Song, Yinghui Wu, and Shengqi Yang. 2019. Answering why-questions by exemplars in attributed graphs. In *Proceedings of the 2019 International Conference on Management of Data*.

[17] Mohammad Hossein Namaki, Xin Zhang, Sukhjinder Singh, Armen Ahmed, Armina Foroutan, Yinghui Wu, Anurag Srivastava, and Anton Kocheturov. 2020. kronos: lightweight knowledge-based event analysis in cyber-physical data streams. In *Proceedings of the 2020 International Conference on Management of Data*.

[18] James E Saal, Scott Kirklin, Muratahan Aykol, Bryce Meredig, and Christopher Wolverton. 2013. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). *Jom* 65, 11 (2013), 1501–1509.

[19] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. 2019. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials* 5, 1 (2019), 1–36.

[20] National Science and Technology Council (US). 2011. *Materials genome initiative for global competitiveness*. Executive Office of the President, National Science and Technology Council.

[21] Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei, and Ming Lei. 2019. Machine learning in materials science. *InfoMat* 1, 3 (2019), 338–358.

[22] Yinghui Wu, Shengqi Yang, and Xifeng Yan. 2013. Ontology-based subgraph querying. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. 697–708.

[23] Shengqi Yang, Yanan Xie, Yinghui Wu, Tianyu Wu, Huan Sun, Jian Wu, and Xifeng Yan. 2014. SLQ: A user-friendly graph querying system. In *SIGMOD*.