Crystal Toolkit: A Web App Framework to Improve Usability and Accessibility of Materials Science Research Algorithms

Matthew Horton,[1,2,*] Jimmy-Xuan Shen,[1,2] Jordan Burns,[2,3] Orion Cohen,[1,4] François Chabbey,[1] Alex M. Ganose,[5] Rishabh Guha,[1] Patrick Huck,[1] Haoming Howard Li,[2,1] Matthew McDermott,[2,1] Joseph Montoya,[6] Guy Moore,[2,1] Jason Munro,[1] Cody O'Donnell,[1] Colin Ophus,[7] Guido Petretto,[8,9] Janosh Riebesell,[10,11] Steven Weitzner,[12] Brook Wander,[13] Donald Winston,[14] Ruoxi Yang,[1] Steven Zeltmann,[2] Anubhav Jain,[1] and Kristin A. Persson[11,2,†]

[1] *Materials Science Division, Lawrence Berkeley National Lab*
[2] *Department of Materials Science and Engineering, University of California, Berkeley*
[3] *Energy Storage and Distributed Resources Division, Lawrence Berkeley National Lab*
[4] *Department of Chemistry, University of California, Berkeley*
[5] *Department of Chemistry, Imperial College London*
[6] *Toyota Research Institute*
[7] *The National Center for Electron Microscopy, Molecular Foundry, Lawrence Berkeley National Lab*
[8] *Matgenix SRL*
[9] *Modelling Division, Institute of Condensed Matter and Nanosciences, Université catholique de Louvain*
[10] *Cavendish Laboratory, University of Cambridge*
[11] *Molecular Foundry, Lawrence Berkeley National Lab*
[12] *Lawrence Livermore National Laboratory*
[13] *Department of Chemical Engineering, Carnegie Mellon University*
[14] *Energy Technologies Area, Lawrence Berkeley National Lab*

Crystal Toolkit is an open source tool for viewing, analyzing and transforming crystal structures, molecules and other common forms of materials science data in an interactive way. It is intended to help beginners rapidly develop web-based apps to explore their own data or to help developers make their research algorithms accessible to a broader audience of scientists who might not have any training in computer programming and who would benefit from graphical interfaces. Crystal Toolkit comes with a library of ready-made components that can be assembled to make complex web apps: simulation of powder and single crystalline diffraction patterns, convex hull phase diagrams, Pourbaix diagrams, electronic band structures, analysis of local chemical environments and symmetry, and more. Crystal Toolkit is now powering the Materials Project website frontend, providing user-friendly access to its database of computed materials properties. In the future, it is hoped that new visualizations might be prototyped using Crystal Toolkit to help explore new forms of data being generated by the materials science community, and that this in turn can help new materials scientists develop intuition for how their data behaves and the insights that might be found within. Crystal Toolkit will remain a work-in-progress and is open to contributions from the community.

I. INTRODUCTION

Effective communication of scientific results is becoming ever more essential: an exponential increase in quantity of papers produced has lead to an "attention decay"[1], whereby it is becoming increasingly difficult for researchers to stay abreast of the published literature. Furthermore, new scientific techniques are generating a vast increase in the quantity of data: this, too, can present a problem, as insights are hidden in these large data sets, which are often difficult to explore. This is a challenge to our community and also an opportunity to find new solutions for better scientific communication. This paper describes improvement in this area by the introduction of a new tool for computational materials scientists to help them explore and share their data, and increase the impact of their work by making it easier to understand and build connections between data sets.

Materials science and chemistry have always benefited from a visual exploration of concepts, whereby visual representations have not only been used as a tool to accelerate learning but have actually enabled new scientific discoveries[2] (see Figure 1). These include Dalton's atoms at the advent of atomic theory, Kekulé's affinity units (initially a convenient orthographic device but that led to the discovery of the benzene ring), Hofmann's molecules built from croquet balls (introducing a strong sense of the spatial arrangement of molecules, and colour schemes for elements that are still in use today), van't Hoff's asymmetric carbon and the concept of stereoisomers, Barlow and the understanding of close packing in crystal structures, Hodgkin and the visualization of electron charge densities in real-space, and even Megaw's popularization of materials science through the Festival Pattern Group–the list goes on! These examples demonstrate how effective visualization has led to essential leaps throughout the history of materials science, and justifies close attention to this area – for education[3, 4] as well

* mkhorton@lbl.gov
† kapersson@lbl.gov

as research.

In the modern era, there has been an explosion of options for molecular visualizations enabled by advances in computation which support new computational simulation techniques. A full review of these would be a substantial effort, but include XCrysDen[5], OVITO[6], VESTA[7], Avogadro[8], VMD[9], Jmol[10], 3DMol[11], nglviewer[12], AtomEye[13]. Each of these tools has developed and refined what is possible in molecular visualization, and have, in turn, each contributed to scientific insights. It is reasonable therefore to consider that the problem of molecular visualization might be solved and ask what benefit yet another option might bring.

Crystal Toolkit has been developed to address new, specific requirements emerging from the computational materials science community. It provides not only new means of molecular and crystal visualization, but goes beyond these and facilitates the visualization of additional forms of data relevant to the field. These requirements come specifically as a result of the rise of high-throughput computational approaches and the vast quantities of data that they create, as well as from recent efforts which closely link computation with experiment to aid in the design of new materials.

II. DESIGN GOALS

Several key goals have governed the design of Crystal Toolkit, informed by historical and technological context, as well as the current needs of the computational materials science community. These include:

- **A human-centered approach.** Since the early study of "human-computer interaction"[14], people have framed the relationship between human and computer as a dialogue. Unfortunately, this dialogue is often too focused on the computer, and humans are forced to adapt to the machine rather than the inverse. In scientific research, this often manifests as poorly-documented codes which are difficult to use or compile, and has had a tremendous cost on scientific advance: in the best case, it delays progress of new users by months or years as they have to learn the quirks of these computational tools, while in the worst case it can alienate new scientists and drive them out of the field entirely, or even result in the publication of incorrect results due to misunderstanding and misinterpretation of a computer code or its output. It is critical that the next generation of tools that we develop as a community start with a human-centered approach to avoid these outcomes.

- **Strong mapping of visualizations to specific software objects.** Individual software objects (for example, a `Structure` object representing a crystal structure) are often difficult to introspect and understand what information they contain. If you are a proficient programmer you can inspect its attributes or read the relevant source code, but this is often inaccessible to new learners. Visualizations allow both novice and experienced users to leverage the brain's immense capacity for visual information processing[15]. Visual representations of data structures might include a formal representation such as a UML (Unified Modeling Language) diagram, but might more usefully be a 3D visualization or a graphical plot. Crystal Toolkit aims to provide understandable visualizations for the commonly used software objects used by the materials science community.

- **Faithful representations of software objects.** It is important that a visualization faithfully represents an underlying software object, and that this in turn is a faithful representation of the corresponding scientific abstraction (see Figure 2). These two degrees of separation–between concept and code, and code and visualization–are fraught, and can introduce problems. For a specific example, some molecular visualizations will automatically detect and insert bonds that are not present in the underlying software object. While useful, it impedes understanding of the object, and can introduce confusion for new developers. A better option is to only visualize the information actually present in the object *by default* with the option of including additional ornaments.

- **Composability of visualizations.** Object-orientated programming is a common pattern in modern programming, whereby complex objects can inherit from and be composed from simpler objects (see Figure 3). Where possible, if a complex object is built from a set of simpler objects, and these objects in turn already have visualizations available, then it should be possible that the visualization of the more complex object can be composed from these already existing visualizations. This allows for reuse and experimentation, and helps enforce the faithfulness of the representation.

- **Empowering users of all experience levels.** Many scientists do not receive formal training in programming, or might receive training in an informal and ad-hoc manner. Nevertheless, ability in programming is becoming increasingly essential to perform scientific analysis. Crystal Toolkit aims to tackle this issue in two ways, by (1) allowing complex web apps to be constructed such that the latest computational materials science algorithms can be used directly via a web app, without any additional programming required by the user, and (2) by providing a roadmap for scientists with some programming training to make their own web apps to explore their own data. To achieve (2), Crystal

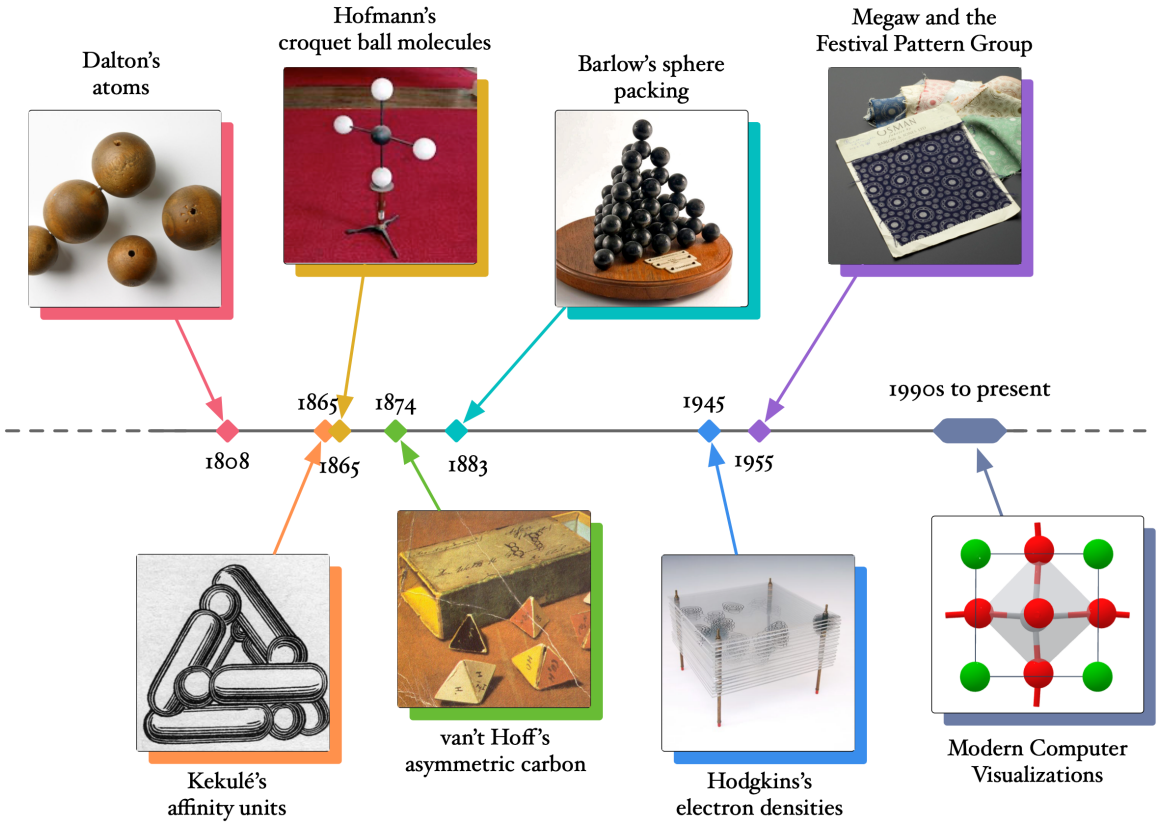

FIG. 1. A timeline of molecular and crystallographic visualizations with several key advances highlighted. Throughout the history of chemistry and materials science, effective visualization has proven an invaluable tool to help develop intuition and unlock key new insights that have advanced the field.

Toolkit should allow apps to be constructed and prototyped from just a few lines of Python code. Python was chosen since it is now the *de facto* standard programming language used for a wide range of scientific tasks, so a language that a new scientist is most likely to have experience with.

- **Portability of representations between different media.** Information can be lost when converting from one representation of a given object to another. For example, a scientist might use one form of plotting or visualization while performing analysis, use another for a publication, and a third for interactive use on a website (see Figure 4). Ideally, all of these views of the same object should be the same, within the limitations of the respective medium. This reduces the risk of loss of information and also reduces the effort required to share data in different places; for example, by making it easier to offer an interactive version of a visualization included in a publication.

- **An open, inclusive attitude to new contributors.** The Crystal Toolkit authors want the framework to live and develop over time, and hope new developers will join in this process. A Zenodo citation will also be provided to ensure new developers can be acknowledged for their contributions, and all users of Crystal Toolkit are encouraged to cite both this design manuscript and also the Zenodo citation.

- **A tool for learning and developing intuition.** In science, we are familiar with how to think about the correspondence between reality and our models of reality, where they agree and are helpful, and where the limits of our models can obscure truth (for example, imagining a bond as a spring). However, with the rise of computational science, we are less practiced in examining our simulations, their predictions and our visual or textual representations of them, whether in the form of graphics, animations or graphs. A good visualization tool should help develop intuition for the models we use, and act as an educational aid.

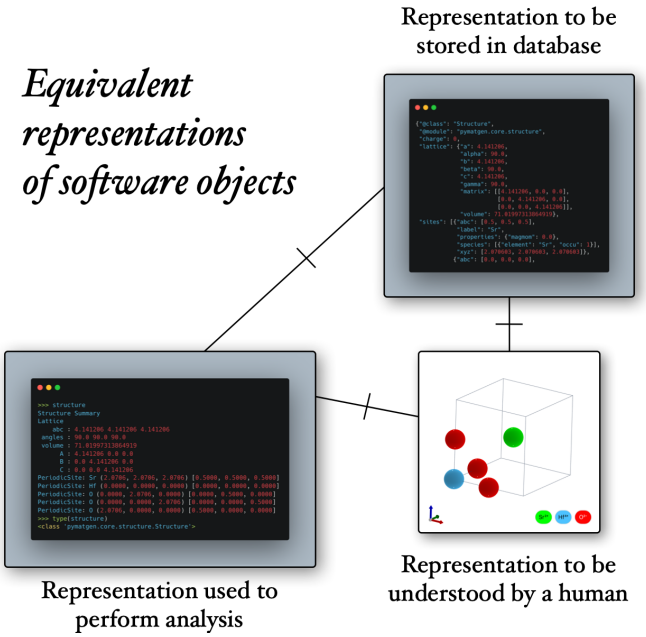

FIG. 2. A design goal of Crystal Toolkit is to provide human-centered representations of software objects, and specifically to ensure that the default representations are as faithful to the underlying software object as possible so as to help the user gain intuition about the use of these objects.

III. A LIBRARY OF READY-TO-USE COMPONENTS

Crystal Toolkit comes with several components included and ready to use. These components map to existing data types used in the materials science community, and mainly to specific classes available in the *pymatgen*[17] code. These components can be used individually, or can be linked together; for example, a "symmetry" component can be linked to a "crystal structure" component so that symmetry information is always shown for the same crystal structure being visualized.

The following is only an initial list of components, with components expected to evolve as development proceeds and new components are made available, so the current list should be considered a snapshot of Crystal Toolkit's capabilities at the time of publication. For an up-to-date list, please consult the Crystal Toolkit documentation[18]. See Figure 5, left panel, for illustrations of several of these components.

1. **Crystal Structures.** Visualize crystal structures with a variety of visualization options, including different colour schemes (including accessible colour schemes), choice of atom size and bonding algorithms[19]. The structure visualizer also allows transformation to different crystallographic settings and downloading of crystal structures to common file formats including CIF. Maps to `Structure` and `StructureGraph` objects.

2. **Molecules.** Visualize molecules in a similar way to crystal structures. Maps to `Molecule` and `MoleculeGraph` objects.

3. **Band Structures.** Visualize electronic or phonon band structures and densities of states. Interact with the band structure by zooming in to specific regions or, in future, by selecting different k-path conventions[20]. Maps to `BandStructure` and `DOS` objects.

4. **Pourbaix Diagrams.** Visualize Pourbaix (aqueous stability) diagrams built from experimental and theoretical data. Interact by changing ion concentrations, or show a heatmap of free energy for a specific material across voltage-pH space. Maps to `PourbaixDiagram` objects.

5. **X-ray Absorption Spectra.** Visualize X-ray absorption spectra, or other kinds of spectra, and interact by applying Gaussian broadening. Maps to `Spectrum` objects.

6. **Phase Diagrams.** Visualize phase diagrams generated with a convex hull construction. Supports 1-, 2-, 3- and 4-element phase diagrams, with 3-element phase diagrams viewable in both 2D and 3D with z-axis of formation enthalpy, and 4-element phase diagrams in 3D. Viewing in chemical potential space is also possible. Interaction includes transformation into a grand potential phase diagram by inclusion of open elements. Maps to `PhaseDiagram` objects.

7. **Symmetry.** Show symmetry information, including space group, point group and detected Wyckoff labels, for a given crystal structure, based on spglib[21]. Interaction includes being able to change tolerances for detection of symmetry, an important consideration for computationally-obtained crystal structures. Improvements include integration with Auguste[22] to show minimum strain values.

8. **Diffraction Patterns.** This component accepts a crystal structure and can generate either an X-ray powder diffraction pattern or a transmission electron microscope single crystal diffraction pattern. In both cases, the input crystal structure is first transformed into its conventional setting such that the peaks and diffraction spots respectively are annotated with the conventional lattice indices. Interaction includes the ability to apply Scherrer broadening to estimate finite size effects.

9. **Wulff Shapes.** Visualize a Wulff shape and interact to examine specific facets and surface energies. Maps to `WulffShape` objects. In an app, it is recommended to link this component to a crystal

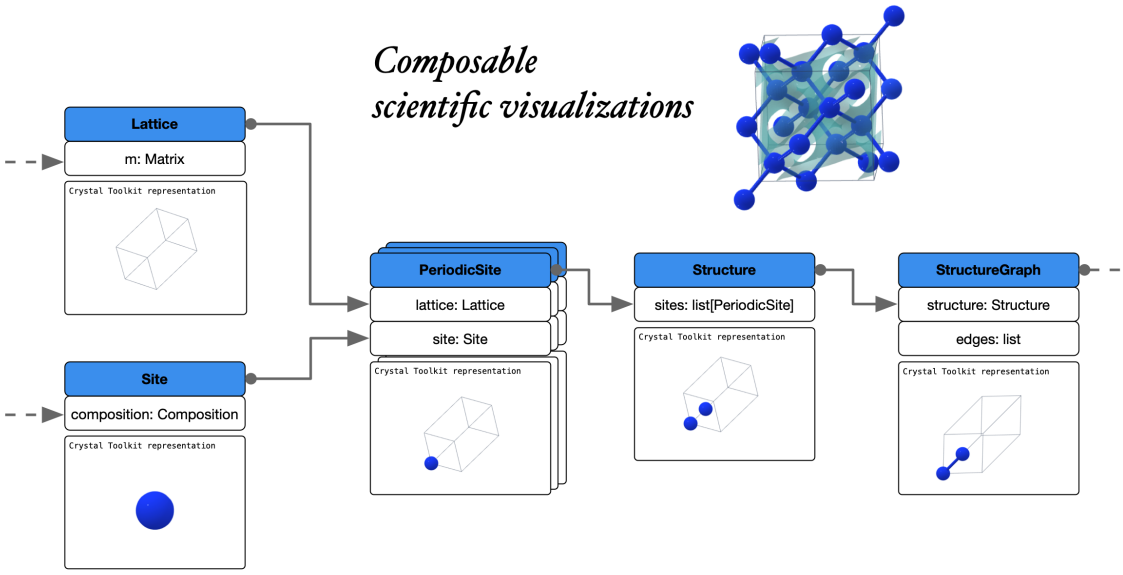

FIG. 3. In a similar way that object-orientated programming allows complex objects to be constructed and composed from simpler objects, visualizations in Crystal Toolkit are designed to map directly to the underlying software object and to be composable, so that complex visualizations can easily be built up. This example shows part of how a visualization for silicon is constructed, with the inset figure showing the final visualization after transformation to a conventional cell and addition of an electron charge density layer.

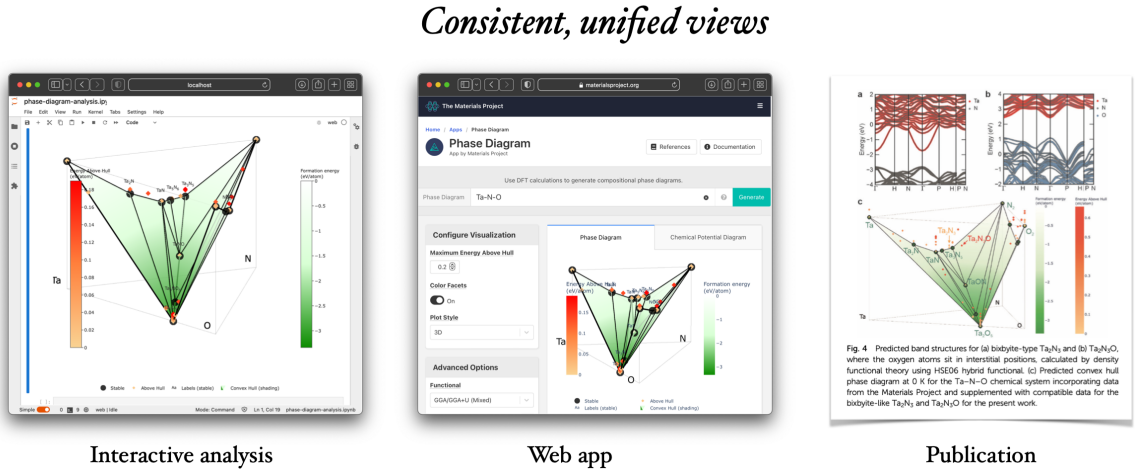

FIG. 4. A significant motivation for the design of Crystal Toolkit is to allow consistent views of the same information across journal publications, web apps, and also during initial data generation and analysis, so as to reduce the opportunity for loss of information or introduction of errors as the data moves from one location to another. Publication figure is taken from Jiang *et al.*[16]

structure component to allow corresponding surfaces to be visualized on interaction with a given facet.

10. **Molecular Graphs.** Molecular or crystal graphs, representing atoms as nodes and bonds (or other information) as edges, can be represented as interactive force-directed graphs. Maps to `StructureGraph` or `MoleculeGraph` objects.

11. **Periodic Table.** This component provides a means to allow selection of specific elements, to be used as an input for another component, and also supports displaying a heatmap for a set of data associated with a list of elements.

12. **Brillouin Zones.** Visualize Brillouin zones and k-paths in 3D. Improvements include linking to band structure components to allow a given high-symmetry line in a band structure diagram to be highlighted in the corresponding Brillouin zone.

Plotting of Fermi surfaces in the Brillouin zone is also possible using IFermi[23].

13. **Transformations Component.** This component accepts a crystal structure as input and provides a means to transform that crystal structure using *pymatgen*. Maps to `Structure` and `AbstractTransformation` objects.

IV. OPEN-SOURCE DEVELOPMENT MODEL

Crystal Toolkit is developed as Open Source Software under a BSD license[24], with the intent to encourage involvement of as many interested scientists as possible in its development. Discussions on features and bugs are conducted openly, currently on GitHub, and are open for participation from anyone.

This approach is motivated by the belief that scientific tools should be open to allow their results to be reproducible, and to try to avoid issues of older tools being abandoned as developers move on and then, over time, ceasing to function. It is essential that new tools are developed with sustainability in mind, so that they can continue to work into the future as the development team or lead maintainers change.

The technical implementation of Crystal Toolkit is expected to change over time as new technology becomes available. Thus, the current manuscript is primarily focused on the design goals and overall philosophy of the project.

At time of writing, the project uses both Python code and JavaScript code, and is built upon the Dash framework[25] by Plotly, with additional custom components written using React[26] and packaged as Dash components[27]. React is a JavaScript library for creating modern web apps that has now seen extensive use by the web development community.

Dash was chosen specifically because it allowed an interactive web app to be written in just a single file in the Python programming language yet provide common web developer convenience features such as hot-module reloading on file changes. This enables apps to be prototyped and developed more rapidly and helps catch bugs more quickly compared to having to write code "in the dark" without immediate feedback. All web components, including custom React components written in JavaScript, map back to automatically-generated and documented Python classes, such that they can be used by Python developers without prior web development experience. However, the ability to write new, custom React components empowers dedicated web developers to extend the functionality of this code in useful ways (e.g., the development of the Periodic Table component). Such a parallel development model enables web experts to extend the code while materials science domain experts can apply and use the code, without either blocking the other's work. Furthermore, since no client state is stored on the server by default, Dash readily scales horizontally and is appropriate even for web apps which experience a large amount of traffic like the Materials Project website.

These technologies were ultimately chosen to achieve the design goal of improving accessibility of the project to newcomers: the Python programming language has seen broad adoption in the scientific community and is now often the first programming language that a new scientist will learn.

Examples on how to get started with Crystal Toolkit to write your own app, including small, self-contained examples and installation instructions, are available in the public documentation[18].

V. CASE STUDY: POWERING THE MATERIALS PROJECT

The Materials Project[28] started as a result of the Materials Genome Initiative[29] with a mandate, among other goals, to develop "open web-based access to computed information on materials and analysis tools to design novel materials." To date, it is now in use by over a quarter of a million registered users from academia, industry and government, with a user base that is growing rapidly year-by-year. The Materials Project offers an encyclopedic view of most known, and many hypothetical, inorganic crystalline materials and their predicted properties, with a goal of accelerating the materials discovery process.

In 2022, Materials Project launched its first major iteration of its web platform (see screenshot in Figure 5, right panel) based upon the Crystal Toolkit framework. Adopting Crystal Toolkit has allowed more rapid development of new features and apps in Materials Project by opening up development to the broader materials science community including graduate students. For example, a new Catalysis Explorer app, making available information computed information on various adsorbates and crystallographic surfaces, and a new MOF Explorer app[30], making available information on over 20,000 metal–organic frameworks and coordination polymers, have both been made possible by the Crystal Toolkit framework, powered by data supplied by the Materials Project's "MPContribs" API.

Crystal Toolkit has also enabled the development of internal tools to examine the Materials Project database as it is being built to include new data, so as to more easily identify issues and perform quality control.

To demo the Crystal Toolkit framework, an app for the Materials Project website has also been developed. The Crystal Toolkit app allows users to upload their own crystal structures and perform transformations using *pymatgen* without having to write their own code. These transformations might include creating a surface slab, assigning oxidation states, or building grain boundaries. The Crystal Toolkit framework began development as an

effort to improve this app to include new transformation options, and as the benefit of the approach became evident, the technology grew to encompass the full Materials Project website frontend.

VI. CONCLUSION

Crystal Toolkit is a new tool intended for computational materials scientists to help visualize and share their data. It is designed to complement and build upon existing efforts, and offers a modest advance in a few key respects (see "Design Goals") by addressing the evolving needs of the materials science community.

Crystal Toolkit provides ways to visualize many common data types used in materials science, and provides interactive widgets that can be linked together and combined in a web app, as well as providing visualizations in analysis tools such as Jupyter notebooks. Crystal Toolkit is proven at scale, and is now powering Materials Project website, its apps, and visualizations included therein. As a demonstration, images of all 146,000 crystal structures available in Materials Project at time of writing have been generated using Crystal Toolkit, with the intent of releasing images under a Creative Commons license for download on Wikimedia.

In the future, we will look to continue to expand the visualizations available, and the means of interacting with them. Despite advances, in many ways modern computer visualizations are not that dissimilar to Hoffman's original ball-and-stick models. It is important that as new data becomes available that we also reassess the best ways of visualizing this data, and thereby develop new intuitions and find new ways of unlocking the secrets hidden within this data.

VII. AUTHOR CONTRIBUTIONS

Authors listed alphabetically with the exception of MKH, JXS, AJ, KAP. MKH initiated and led the project, prepared the manuscript and all figures, developed the code, and is lead maintainer of the code. JXS contributed initial support for Jupyter notebooks, Asymptote rendering, surface plotting, initial support for axes, and various bug fixes, and offered testing and feedback throughout. JB developed the reaction calculator example app. OC developed functionality related to periodic tiling. FC improved the JavaScript code, contributed features to 3D rendering and wrote the Periodic Table React component. AMG contributed a component for Fermi surfaces. RG improved rending of molecules. PH deployed the current Materials Project website based on Crystal Toolkit. HHL contributed functionality for rendering migration graph objects. MM contributed to the phase diagram, X-ray diffraction and X-ray absorption components. J Montoya contributed the Pourbaix component. GM improved functionality related to rendering of crystals with magnetic moments. J Munro contributed band structure component. CTO wrote several custom React components. SZ worked on electron diffraction component, supervised by CO. GP helped with X-ray diffraction component and with general bug fixes and improvements. JR helped with general repository maintenance, continuous integration and bug fixes. SW developed POV-Ray integration. BW helped with developing an example app for display of contributed data via MPContribs and improvements to documentation. DW helped with initial integration within the legacy Materials Project website. RY helped with bug fixes, display of calculation task documents, and absorption components. AJ provided helpful feedback. KAP provided helpful feedback, supervision and funding support.

VIII. ACKNOWLEDGEMENTS

Shyue Ping Ong is acknowledged for permission to adopt the "Crystal Toolkit" name. Shyam Dwaraknath is acknowledged for helpful conversations throughout and contributions to the design. Tyler Huntington is acknowledged for assisting DW in the initial integration of Crystal Toolkit with the Materials Project. Andrew Rosen is acknowledged for feedback and testing of Crystal Toolkit during development of an interactive web app for MOF data and for helpful comments on the manuscript. David Waroquiers is acknowledged for helpful discussions during the development of chemical environment components. Rachel Woods-Robinson is acknowledged for helpful comments on the manuscript. This work was supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division under Contract No. DE-AC02-05-CH11231 (Materials Project program KC23MP).

[1] P. D. B. Parolo, R. K. Pan, R. Ghosh, B. A. Huberman, K. Kaski, and S. Fortunato, Attention decay in science, J. Informetr. **9**, 734 (2015).

[2] C. Meine, Molecules and croquet balls, in *Models*, edited by S. de Chadarevian and N. Hopwood (Stanford University Press, Redwood City, 2022) pp. 242–275.

[3] R. Kobayashi, T. P. Goumans, N. O. Carstensen, T. M. Soini, N. Marzari, I. Timrov, S. Poncé, E. B. Linscott, C. J. Sewell, G. Pizzi, *et al.*, Virtual computational chemistry teaching laboratories—hands-on at a distance, Journal of Chemical Education **98**, 3163 (2021).

[4] S. Lehtola and A. J. Karttunen, Free and open source software for computational chemistry education, Wiley

FIG. 5. Left, a summary of just some of the components available for use in a Crystal Toolkit web app. Many components rely on underlying functionality available in the *pymatgen* code, and some functionality was specifically added to *pymatgen* to facilitate features in Crystal Toolkit. Right, an example of real-world usage of some of these components in a Materials Project "materials detail page", here showing the detail page for wurtzite GaN for this example, located at https://materialsproject.org/materials/mp-804.

Interdisciplinary Reviews: Computational Molecular Science **12**, e1610 (2022).
[5] A. Kokalj, XCrySDen—a new program for displaying crystalline structures and electron densities, Journal of Molecular Graphics and Modelling **17**, 176 (1999).
[6] A. Stukowski, Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool, Modelling and Simulation in Materials Science and Engineering **18**, 015012 (2010).
[7] K. Momma and F. Izumi, VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data, Journal of Applied Crystallography **44**, 1272 (2011).
[8] M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison, Avogadro: an advanced semantic chemical editor, visualization, and analysis platform, Journal of Cheminformatics **4**, 17 (2012).
[9] W. Humphrey, A. Dalke, and K. Schulten, VMD: Visual molecular dynamics, Journal of Molecular Graphics **14**, 33 (1996).
[10] R. M. Hanson, Jmol – a paradigm shift in crystallographic visualization, Journal of Applied Crystallography **43**, 1250 (2010).
[11] N. Rego and D. Koes, 3dmol.js: molecular visualization with WebGL, Bioinformatics **31**, 1322 (2015).
[12] A. S. Rose and P. W. Hildebrand, NGL Viewer: a web application for molecular visualization, Nucleic Acids Research **43**, W576 (2015).
[13] J. Li, AtomEye: an efficient atomistic configuration viewer, Modelling and Simulation in Materials Science and Engineering **11**, 173 (2003).
[14] S. K. Card, T. P. Moran, and A. Newell, eds., *Psychology of human/computer interaction* (Lawrence Erlbaum Associates, Mahwah, NJ, 1983).
[15] S. L. Franconeri, L. M. Padilla, P. Shah, J. M. Zacks, and J. Hullman, The science of visual data communication: What works, Psychological Science in the Public Interest **22**, 110 (2021), pMID: 34907835.
[16] C.-M. Jiang, L. I. Wagner, M. K. Horton, J. Eichhorn, T. Rieth, V. F. Kunzelmann, M. Kraut, Y. Li, K. A. Persson, and I. D. Sharp, Metastable ta_2n_3 with highly tunable electrical conductivity via oxygen incorporation, Materials horizons **8**, 1744 (2021).
[17] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, Python materials genomics (pymatgen): A robust, open-source python library for materials analysis, Computational Materials Science **68**, 314 (2013).
[18] Crystal Toolkit Development Team, Crystal Toolkit Documentation (2022).
[19] H. Pan, A. M. Ganose, M. Horton, M. Aykol, K. A. Persson, N. E. Zimmermann, and A. Jain, Benchmarking coordination number prediction algorithms on inorganic crystal structures, Inorganic chemistry **60**, 1590 (2021).
[20] J. M. Munro, K. Latimer, M. K. Horton, S. Dwaraknath, and K. A. Persson, An improved symmetry-based approach to reciprocal space path selection in band structure calculations, npj Computational Materials **6**, 112 (2020).
[21] A. Togo and I. Tanaka, Spglib: a software library for crystal symmetry search, arXiv preprint arXiv:1808.01590 (2018).
[22] P. M. Larsen, E. L. Pang, P. A. Parrilo, and K. W. Jacobsen, Minimum-strain symmetrization of bravais lattices, Physical Review Research **2**, 013077 (2020).
[23] A. M. Ganose, A. Searle, A. Jain, and S. M. Griffin, Ifermi: A python library for fermi surface generation and analysis, Journal of Open Source Software **6**, 3089 (2021).
[24] 3-Clause BSD License (2017).
[25] Plotly, Dash (2017).
[26] Materials Project, `mp-react-components` (2020).
[27] Materials Project, `dash-mp-components` (2019).
[28] Materials Project, Materials Project Website (2011).
[29] Materials Genome Initiative, Materials Genome Initiative Website (2011).
[30] A. S. Rosen, V. Fung, P. Huck, C. T. O'Donnell, M. K. Horton, D. G. Truhlar, K. A. Persson, J. M. Notestein, and R. Q. Snurr, High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration, npj Computational Materials **8**, 1 (2022).