Batteries & Supercaps

Research Article
doi.org/10.1002/batt.202300038

Chemistry Europe
European Chemical Societies Publishing

www.batteries-supercaps.org

GalvAnalyze: Streamlining Data Analysis of Galvanostatic Battery Cycling

Rory C. McNulty[+[b, c]] and Lukas Rier[+*[a]]

We present GalvAnalyze, a software tool developed in Python, that processes data collected using a variety of battery cyclers and creates a set of outputs that greatly reduce the inefficiencies associated with manual or semi-manual analysis of galvanostatic cycling data. An experiment is carried out by 10 participants with varying degrees of experience processing galvanostatic cycling data, enabling quantitative analysis of the processing time benefit that GalvAnalyze provides. The functionality of GalvAnalyze includes handling data where the applied current density varies, separating the data into individual charge-discharge cycle pairs, and producing hysteresis plots using a simple graphical user interface. The executable can be downloaded at http://www.thenamilab.com/ and is built to be accessible, with no prior coding expertise required. In the interest of transparency, and to allow future contributions to the functionality of GalvAnalyze by the wider community, the source code can be found on GitHub (https://github.com/LukasRier/GalvAnalyze).

Introduction

Batteries, and more broadly energy storage systems, have become an integral part of everyday life, from powering consumer electronics to the electrification of transport and supporting the operation of renewable energy technologies.[1–3] As the world electrifies, the demand for higher performance energy storage solutions maintains rapid growth, driving a concerted effort in academia and industry to discover appropriate solutions.[4–6] Galvanostatic charge-discharge cycling is critical to understanding the fundamental performance of these materials, used both in isolation and as a tool for exploring the ageing and degradation processes of battery chemistries with accompanying operando, *in-situ* and *ex-situ* characterisation techniques.[7–10] However, performing identical experiments on two different battery cyclers can require markedly different amounts of manual data processing, depending on hardware and software capabilities. Where manual or semi-manual data processing is employed to overcome the shortfalls of software, significant time and effort can be spent understanding the results of a given experiment. Although some research groups possess coding expertise that is sufficient to overcome the shortfalls of their specific instrumentation, those with limited coding experience are constrained by the solutions offered through instrument providers. This disparity in access to efficient data analysis tools creates unnecessary barriers to essential data processing.

Various open-source data analysis tools have been developed by the battery community to streamline specific parts of data analysis that are often deemed challenging.[11–17] Dahn *et al.* and, more recently, de Souza *et al.* developed tools that enable efficient differential voltage and incremental capacity analysis from high-quality data inputs.[11,12] Murbach *et al.* developed a tool for the analysis of battery impedance data.[13] Dubarry et al. and Herring et al. developed tools to evaluate and model battery cell performance,[14,15] and Lewis-Douglas et al. built a tool designed to enable development of a battery library/database to connect data collected by experimentalists to their modelling counterparts.[16] Topically, an article recently published by Ward et al. proposed the development of a 'Battery Data Genome', stating that the primary roadblock to a battery-data-science renaissance is the requirement for large amounts of high-quality data.[17] Although each of these open-source tools effectively serves the purpose of improving the efficiency of data processing of the specific target area, they have pre-requisites of either high-quality input data, specific experimental procedures, or prior coding expertise to install and operate the tool, presenting a barrier to adoption for many experimentalists.

Here we present GalvAnalyze, a Python-based tool that is designed to bridge the gap between research groups with coding expertise and those without, offering an open-source

[a] *Dr. L. Rier[+]*
Sir Peter Mansfield Imaging Centre
University of Nottingham
University Park
NG7 2RD
Nottingham (UK)
E-mail: Lukas.rier@nottingham.ac.uk
[b] *R. C. McNulty[+]*
Nottingham Applied Materials and Interfaces (NAMI) group
School of Chemistry
University of Nottingham
NG7 2TU
Nottingham (UK)
[c] *R. C. McNulty[+]*
The Faraday Institution
Quad One
Harwell Science and Innovation Campus
OX11 0RA
Didcot (UK)
[+] *These authors contributed equally to this work.*

Supporting information for this article is available on the WWW under https://doi.org/10.1002/batt.202300038

Research Article
doi.org/10.1002/batt.202300038

plug-and-play style executable, with a graphical user interface, for streamlining analysis of galvanostatic cycling data. All required libraries are built-in to a single downloadable file, enabling accessibility to experimentalists with no prior coding expertise, which is a critical motivation for this project. This publication acts as a guide to users, discussing the processes that are carried out from raw data input to the export of data in graphical and .csv (or .parquet) format, ready for further manipulation by the user. Furthermore, an experiment is carried out, wherein 10 participants analyse a dataset manually and with the aid of the GalvAnalyze executable, to quantify its time-saving effectiveness.

Results and Discussion

Discussion of software design

Regardless of chosen instrumentation, the same key outputs are commonly extracted from data acquired in galvanostatic cycling experiments: 1) a plot of the capacity as a function of potential – to understand the stages of intercalation and potential regions in which they occur,[18] 2) a plot of maximum charge and discharge capacity as a function of cycle number – to understand the specific capacity of the cell and how this varies over cycle life,[19] 3) the values of the coulombic efficiency – to understand the round-trip efficiency of the active species and highlight regions of inefficiency,[20] and 4) a plot of the first cycle hysteresis – which highlights the capacity loss and voltage hysteresis over a given cycle, i.e. lithium inventory loss due to solid-electrolyte-interphase (SEI) formation or loss/trapping of redox active materials.[21] As a result of this, the time taken to process data from a set of experiments conducted using "Instrument A" may be significantly different to that taken to process the same data from "Instrument B", depending on the accessibility of the output files. Furthermore, to cross-reference data from multiple instruments, these data must be filed in a uniform format. GalvAnalyze can handle data from a variety of battery cyclers, autonomously produce uniform outputs in the form of comma-separated value (.csv) or parquet files (to ensure scalability), and generate graphics that address each of the four key outputs mentioned above. The user inputs of time, applied current, and potential are used to clean and convert the data to a uniform format, derive further information from the data provided, export the cleaned data into files in a standardised format, generate graphical representations of the dataset, and offer additional functionality for further data processing.

Figure 1 shows the processing steps implemented in GalvAnalyze, from raw data file input to final outputs from the viewpoint of the user. Two manual inputs are required: the active mass value in mg, m_a, and a text file from the battery cycling software. If these inputs are accepted, the remainder of the data processing is done autonomously on a timescale of seconds to minutes depending on file size. The data are first split into positive (charge) current regions and negative (discharge) current regions through a combination of thresh-

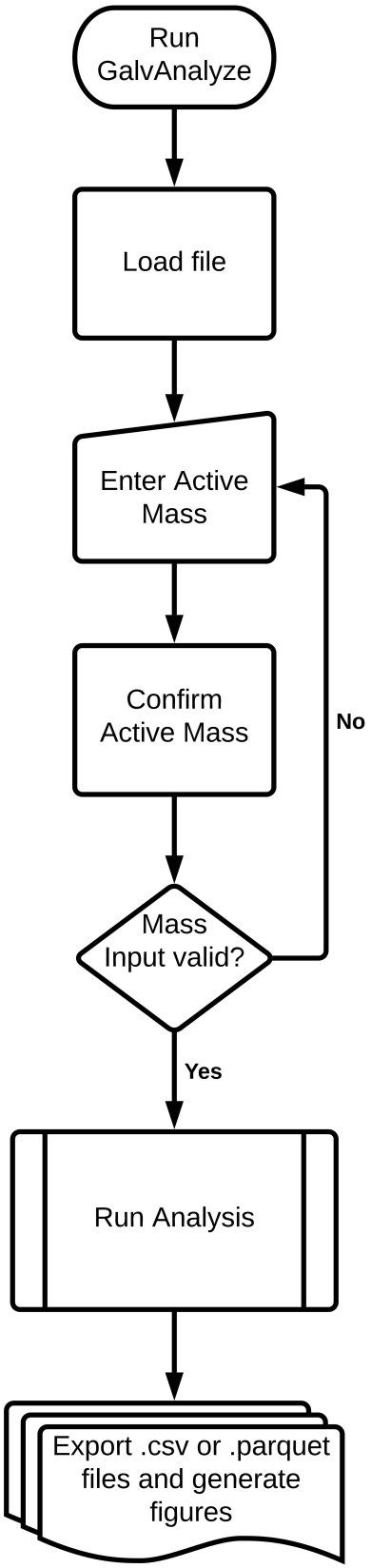

Figure 1. Flowchart for the main functionality of GalvAnalyze from running the executable to generating .csv files and figures. User inputs: Load file, Enter Active Mass, and decision: Input Valid are required for the script to carry out the remaining processing steps. Further descriptions can be found in the Supplementary information, including error handling and a full user guide.

Research Article
doi.org/10.1002/batt.202300038
Chemistry Europe
European Chemical Societies Publishing

olding and edge detection. Put simply, this identifies the start time point, t_0, and end time point, t_{final}, of each charge and discharge cycle. The concepts of thresholding and edge detection are described further in the supporting information (Figure S1). With each charge and discharge cycle separated into arrays of data points, between their respective t_0 and t_{final}, the elapsed cycle time, t_{cyc}, is calculated as the difference between the timestamp of each data point and the preceding t_0. Figure 2 shows a visual representation of the thresholding approach, where the red and blue lines indicate the positive and negative threshold, respectively, and the green and red shaded regions represent areas that have been identified as part of a charge and discharge process, respectively. This type of plot is provided each time GalvAnalyze is run to allow for transparent data processing and for users to verify that their dataset is of sufficiently high quality.

The capacity output from a battery cycler is given in its absolute form, as the product of the current passed and the time elapsed, commonly given in units of mAh or Ah. To understand the relevance of this absolute value to a specific experiment, this must be further converted into capacity per unit weight of active material, which enables comparison of datasets. GalvAnalyze carries out this conversion by calculating the capacity in mAh g^{-1} at each time point within a given cycle by using the user-provided active mass, m_a, the time points for the given cycle, t_{cyc}, and applied current, I using Equation (1):

$$\text{Capacity} = \frac{It_{cyc}}{3600m_a} \tag{1}$$

This generates the total capacity passed at every time point in all cycles that have been identified, providing all data required to generate plots that show the capacity as a function of potential throughout cycling and how the maximum charge and discharge capacity change as a function of cycle number. From the maximum charge and discharge capacity for each cycle, the coulombic efficiency can be calculated with Equation (2):

$$\text{Coulombic efficiency} = \frac{\text{max. cap. of second process}}{\text{max. cap. of first process}} \times 100 \tag{2}$$

These data are subsequently exported to a file. Furthermore, GalvAnalyze plots these data for the user using matplotlib,[22] giving immediate visualisation of their results.

The most basic functionality of the executable is the analysis of constant current charge-discharge cycling data, in which the same magnitude of current is applied throughout the entire experiment. Although this is suitable in many cases, experiments frequently involve one or more changes in the applied current density. Many long-term cycling experiments require the application of a low current density during SEI formation, after which a higher current density is applied for the remainder of the experiment.

To address more complex experiments and required outputs, GalvAnalyze offers the following options: 1) "Applied current varies" – where the autonomous cycle detection allows for changes in applied current density. 2) "Separate charge-discharge pairs to file" – where each cycle is collated into an individual file for ease of further analysis. 3) "Get hysteresis plot" – which allows the user to select one of the exported single-cycle files and produce a hysteresis plot of the selected cycle. A full breakdown of the functions present in the underlying code can be found in the Supporting Information under 'implementation and architecture', alongside a flowchart describing the full functionality of the GalvAnalyze executable (Figure S2), a comprehensive user guide, and a guide to error handling.

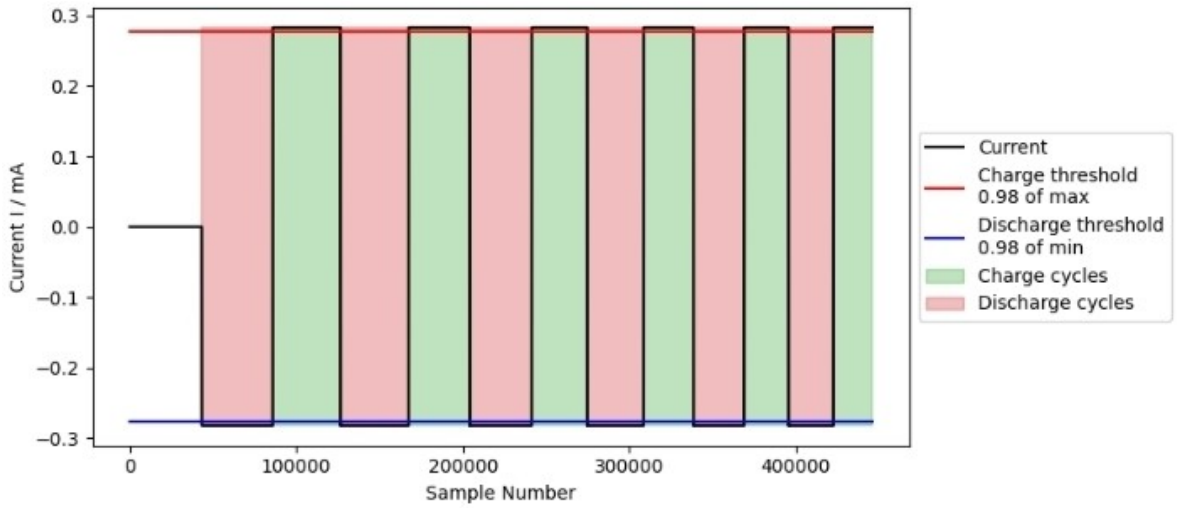

Figure 2. Example of a diagnostic plot produced when the GalvAnalyze executable is run. The red and blue lines represent the applied positive and negative thresholds, respectively, that are used to determine data points that belong to charge (green regions) and discharge (red regions).

Research Article
doi.org/10.1002/batt.202300038
Chemistry Europe
European Chemical Societies Publishing

Example application

GalvAnalyze was created to reduce the amount of manual processing required to go from a raw dataset to data that is ready to plot in the users' preferred figure plotting software. Figure 3 shows a common error found in raw datasets, wherein the final data points of a given charge process have been connected to the first datapoint of the preceeding discharge process. The same has occurred for each discharge. This results in several lines across the bottoms and top of the graphic which, if removed manually, would take considerable data manipulation effort, and might introduce variability in results produced by different researchers. The potential and capacity data for Figure 3 were exported directly from the cycler software and plotted without further manipulation.

Figure 4 explores the same dataset, now processed using the GalvAnalyze executable, using the raw time, potential, and current data exported directly from the cycler software. The erroneous data points have been removed and autonomous data processing has been carried out, producing graphical representations of the maximum capacities per cycle (Figure 4A), coulombic efficiency (Figure 4B), and first cycle hysteresis (Figure 4C). This reduces the inefficiencies associated with troublesome data points and provides the user with a variety of organised .csv or parquet files for further manipulation in their software of choice.

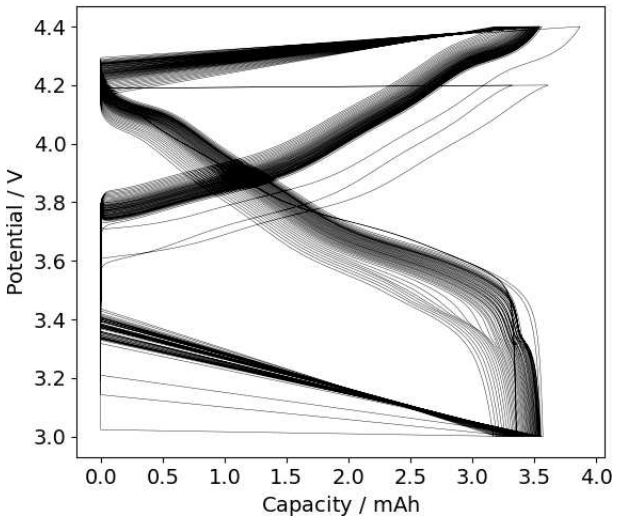

Figure 3. Plot showing example data from a 50-cycle dataset. A cell was cycled between 3.0–4.4 V. Raw potential and capacity data were exported using software included with a common commeial battery cycler and plotted without further manipulation.

Assessing the time-saving benefits of GalvAnalyze

A software solution can be expected to decrease the time taken to analyse a given dataset by removing the need for manual data manipulation. We quantified this effect to better understand the impact of GalvAnalyze on data analysis time.

An experiment was conducted on 10 participants with a range of battery cycling data analysis expertise, between 0–68 months, to determine the time taken to manually process a dataset with two different current protocols and a total of 12 cycles, compared to the time taken to process the same dataset using the GalvAnalyze executable. Written informed consent was obtained from all participants before conducting the experiment. The manual processing part of the experiment was split into three distinct time periods. T1 – where the data were given to participants in an 'unoptimised' format with raw time, current, and potential values. These had to be separated into individual charge and discharge pairs and used to calculate capacity values. T2 – where, once the data was separated, two common battery cycling graphics had to be plotted: 1) a double *y*-axis cycle number vs. maximum capacity and coulombic efficiency plot and, 2) a capacity vs. potential plot. T3 – where a third, less common, first cycle hysteresis plot was created. The time taken to complete the task while using GalvAnalyze included opening the executable, exporting data in the chosen file format from the battery cycling software, and processing these data into graphics.

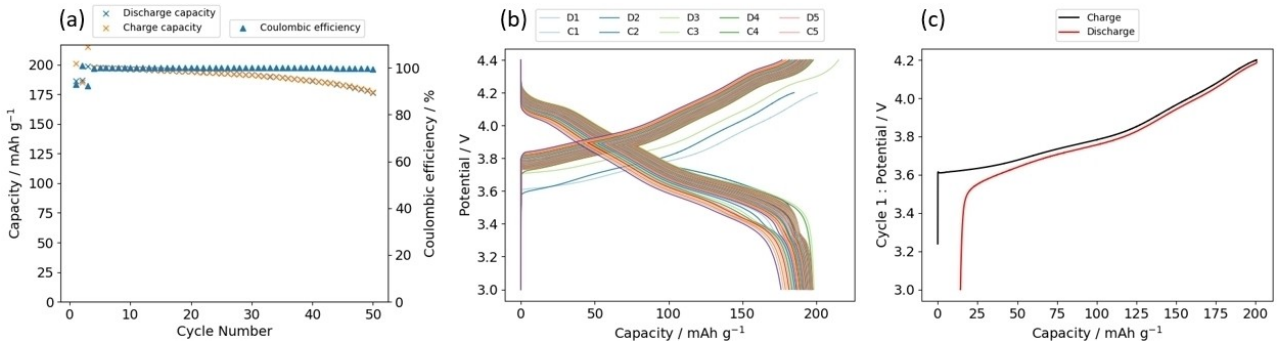

Figure 4. Shows the graphical outputs produced by GalvAnalyze from an example dataset, where time, potential and current values were exported using software included with a common commercial battery cycler. The data shown are from a 50-cycle dataset, cycled between 3.0–4.4 V and processed using GalvAnalyze to create graphics showing a) a double-Y plot of max capacity and coulombic efficiency per cycle, b) voltage profiles for each cycle, and c) a hysteresis plot of the first cycle.

Research Article
doi.org/10.1002/batt.202300038
Chemistry Europe
European Chemical Societies Publishing

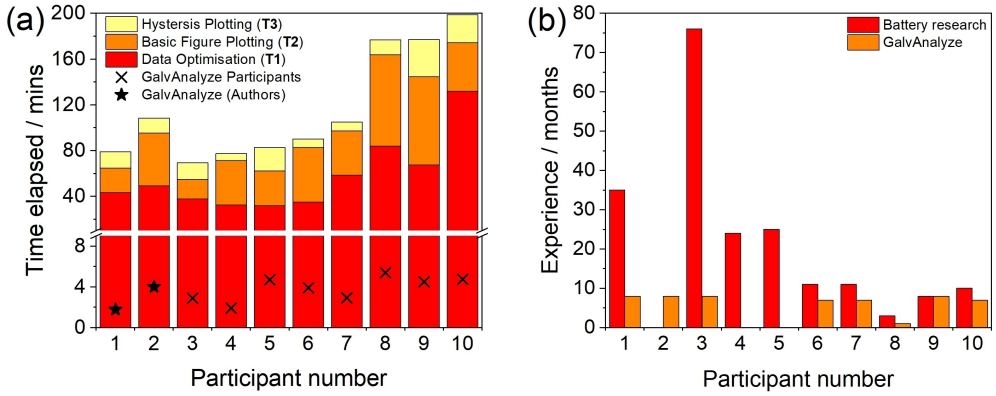

Figure 5. Plots showing the a) results from and b) battery and GalvAnalyze experience of the participants that took part in a data analysis experiment wherein a 12-cycle dataset was processed manually in three sections or with the GalvAnalyze executable (stars = authors, crosses = other participants). a) the manual part of the experiment was separated into three sections: T1 (organisation of data, red), T2 (basic figure plotting, orange), and T3 (hysteresis plotting, yellow).

Discussion of results

The experimental results are shown in Figure 5. The average time taken to analyse the dataset using GalvAnalyze was 3 min 41 s, with the maximum time being 5 min 24 s. It is worth noting that while the authors (who were included in the participant cohort) had significantly more experience using GalvAnalyze, processing times were comparable with those of less experienced users. A significant increase in processing time was found in the manual portion of the experiment, where an average time of 1 h 39 min and maximum of 3 h 18 min were recorded.

While the differences between manual and software-based analysis are striking, the processing of battery cycling data is not usually carried out from raw datasets containing only time, potential, and current inputs, and not every experiment will require the production of a hysteresis plot. A fairer comparison could therefore be made by contrasting the time spent using GalvAnalyze with T2, where an optimised dataset is taken from raw data to basic battery cycling graphics. For manual processing, a minimum and average T2 time of 17 min 17 s and 43 min 54 s were recorded, respectively. Comparing the maximum time taken to use GalvAnalyze with the minimum time taken to process the data manually (T2), we see a more than 3-fold increase in data processing efficiency using the GalvAnalyze executable. This highlights the benefits of using this tool for all users, from beginners to those with many years of data processing experience.

Conclusions

Here we have introduced GalvAnalyze, an accessible tool for processing and visualisation of galvanostatic charge-discharge cycling data, to the battery research community. Users can import data as a text file, enter experimental parameters, and obtain graphical representations of data and .csv files ready for further manipulation. The GalvAnalyze executable was shown to reduce data processing time to less than 6 min, where manual processing of the same dataset exported from commercial cycler software (T2) took an average of 43 min and 54 s across the 10 participants sampled. Comparison of the slowest GalvAnalyze run and the fastest manual T2 run revealed that GalvAnalyze decreased the required data processing time more than 3-fold, highlighting its practicality for all experience levels.

Supporting Information

Further information describing the functionality of the code, a full user guide, and error-handling scenarios can be found in the Supporting Information. Please submit any queries or requests for improvements online at http://www.thenamilab.com/about-4.

Acknowledgements

We gratefully acknowledge support of this research by the Faraday Institution's Degradation project (EP/S003053/1 FITG001, FIRG024).

Conflict of Interests

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords: Batteries · Lithium-ion · Data Processing · Python · Open-source

[1] B. E. Murdock, K. E. Toghill, N. Tapia-Ruiz, *Adv. Energy Mater.* **2021**, *11*, 2102028.
[2] I. M. Peters, C. Breyer, S. A. Jaffer, S. Kurtz, T. Reindl, R. Sinton, M. Vetter, *Joule* **2021**, *5*, 1353–1370.
[3] Y. Liang, C.-Z. Zhao, H. Yuan, Y. Chen, W. Zhang, J.-Q. Huang, D. Yu, Y. Liu, M.-M. Titirici, Y.-L. Chueh, H. Yu, Q. Zhang, *InfoMat* **2019**, *1*, 6–32.
[4] J. Ma, Y. Li, N. S. Grundish, J. B. Goodenough, Y. Chen, L. Guo, Z. Peng, X. Qi, F. Yang, L. Qie, C.-A. Wang, B. Huang, Z. Huang, L. Chen, D. Su, G. Wang, X. Peng, Z. Chen, J. Yang, S. He, X. Zhang, H. Yu, C. Fu, M. Jiang, W. Deng, C.-F. Sun, Q. Pan, Y. Tang, X. Li, X. Ji, F. Wan, Z. Niu, F. Lian, C. Wang, G. G. Wallace, M. Fan, Q. Meng, S. Xin, Y.-G. Guo, L.-J. Wan, *J. Phys. Appl. Phys.* **2021**, *54*, 183001.
[5] Y. Zhao, O. Pohl, A. I. Bhatt, G. E. Collis, P. J. Mahon, T. Rüther, A. F. Hollenkamp, *Sustain. Chem.* **2021**, *2*, 167–205.
[6] J. Deng, C. Bae, A. Denlinger, T. Miller, *Joule* **2020**, *4*, 511–515.
[7] K. Märker, C. Xu, C. P. Grey, *J. Am. Chem. Soc.* **2020**, *142*, 17447–17456.
[8] T. M. M. Heenan, A. Jnawali, M. D. R. Kok, T. G. Tranter, C. Tan, A. Dimitrijevic, R. Jervis, D. J. L. Brett, P. R. Shearing, *J. Electrochem. Soc.* **2020**, *167*, 140530.
[9] E. Björklund, C. Xu, W. M. Dose, C. G. Sole, P. K. Thakur, T.-L. Lee, M. F. L. De Volder, C. P. Grey, R. S. Weatherup, *Chem. Mater.* **2022**, *34*, 2034–2048.
[10] E. Miele, W. M. Dose, I. Manyakin, M. H. Frosz, Z. Ruff, M. F. L. De Volder, C. P. Grey, J. J. Baumberg, T. G. Euser, *Nat. Commun.* **2022**, *13*, 1651.
[11] H. M. Dahn, A. J. Smith, J. C. Burns, D. A. Stevens, J. R. Dahn, *J. Electrochem. Soc.* **2012**, *159*, A1405.
[12] M. L. de Souza, M. Duquesnoy, M. Morcrette, A. A. Franco, *Batteries & Supercaps* **2023**, *6*, e202200378.
[13] M. Murbach, B. Gerwe, N. Dawson-Elli, L. Tsui, *J. Open Source Softw.* **2020**, *5*, 2349.
[14] M. Dubarry, C. Truchot, B. Y. Liaw, *J. Power Sources* **2012**, *219*, 204–216.
[15] P. Herring, C. Balaji Gopal, M. Aykol, J. H. Montoya, A. Anapolsky, P. M. Attia, W. Gent, J. S. Hummelshøj, L. Hung, H.-K. Kwon, P. Moore, D. Schweigert, K. A. Severson, S. Suram, Z. Yang, R. D. Braatz, B. D. Storey, *SoftwareX* **2020**, *11*, 100506.
[16] A. Lewis-Douglas, L. Pitt, D. A. Howey, **2020**, arXiv preprint DOI: 10.48550/arXiv.2010.14959.
[17] L. Ward, S. Babinec, E. J. Dufek, D. A. Howey, V. Viswanathan, M. Aykol, D. A. C. Beck, B. Blaiszik, B.-R. Chen, G. Crabtree, S. Clark, V. D. Angelis, P. Dechent, M. Dubarry, E. E. Eggleton, D. P. Finegan, I. Foster, C. B. Gopal, P. K. Herring, V. W. Hu, N. H. Paulson, Y. Preger, D. Uwe-Sauer, K. Smith, S. W. Snyder, S. Sripad, T. R. Tanim, L. Teo, *Joule* **2022**, *6*, 2253–2271.
[18] D. Liu, W. Zhu, J. Trottier, C. Gagnon, F. Barray, A. Guerfi, A. Mauger, H. Groult, C. M. Julien, J. B. Goodenough, K. Zaghib, *RSC Adv.* **2013**, *4*, 154–167.
[19] J. E. Harlow, X. Ma, J. Li, E. Logan, Y. Liu, N. Zhang, L. Ma, S. L. Glazier, M. M. E. Cormier, M. Genovese, S. Buteau, A. Cameron, J. E. Stark, J. R. Dahn, *J. Electrochem. Soc.* **2019**, *166*, A3031.
[20] B. D. Adams, J. Zheng, X. Ren, W. Xu, J.-G. Zhang, *Adv. Energy Mater.* **2018**, *8*, 1702097.
[21] R. A. House, G. J. Rees, M. A. Pérez-Osorio, J.-J. Marie, E. Boivin, A. W. Robertson, A. Nag, M. Garcia-Fernandez, K.-J. Zhou, P. G. Bruce, *Nat. Energy* **2020**, *5*, 777–785.
[22] J. D. Hunter, *Comput. Sci. Eng.* **2007**, *9*, 90–95.

Manuscript received: February 2, 2023
Revised manuscript received: April 21, 2023
Accepted manuscript online: May 17, 2023
Version of record online: June 2, 2023