Supplementary Information—
The metrics of battery leadership: Patenting patterns since 2000

Philipp Metzger[1,*], **Sandro Mendonça**[3], **José A. Silva**[2], **and Bruno Damásio**[1]

[1]NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Lisbon, Portugal
[2]Instituto Dom Luiz, Faculdade de Ciências Universidade de Lisboa, Lisbon, Portugal
[3]ISCTE Business School, Business Research Unit (BRU-IUL), Lisbon, Portugal; UECE/REM – ISEG/ University of Lisbon, Portugal; SPRU, University of Sussex, UK
[*]philipp.metzger.bat.pat@gmail.com

Battery IPF intensities for each continent

Figure S1 presents the development of the number of battery IPFs per 1M workers (battery IPF intensities) for each continent. In terms of battery IPF intensities, Europe and North America outperform Asia. Asia contributed approximately 60% to the global labor force in the timeframe of 2000-2019 (Europe and North America contributed approximately 9% and 8%, respectively). This imbalance explains why Asia's battery patenting activity is lower in the perspective of this representation.

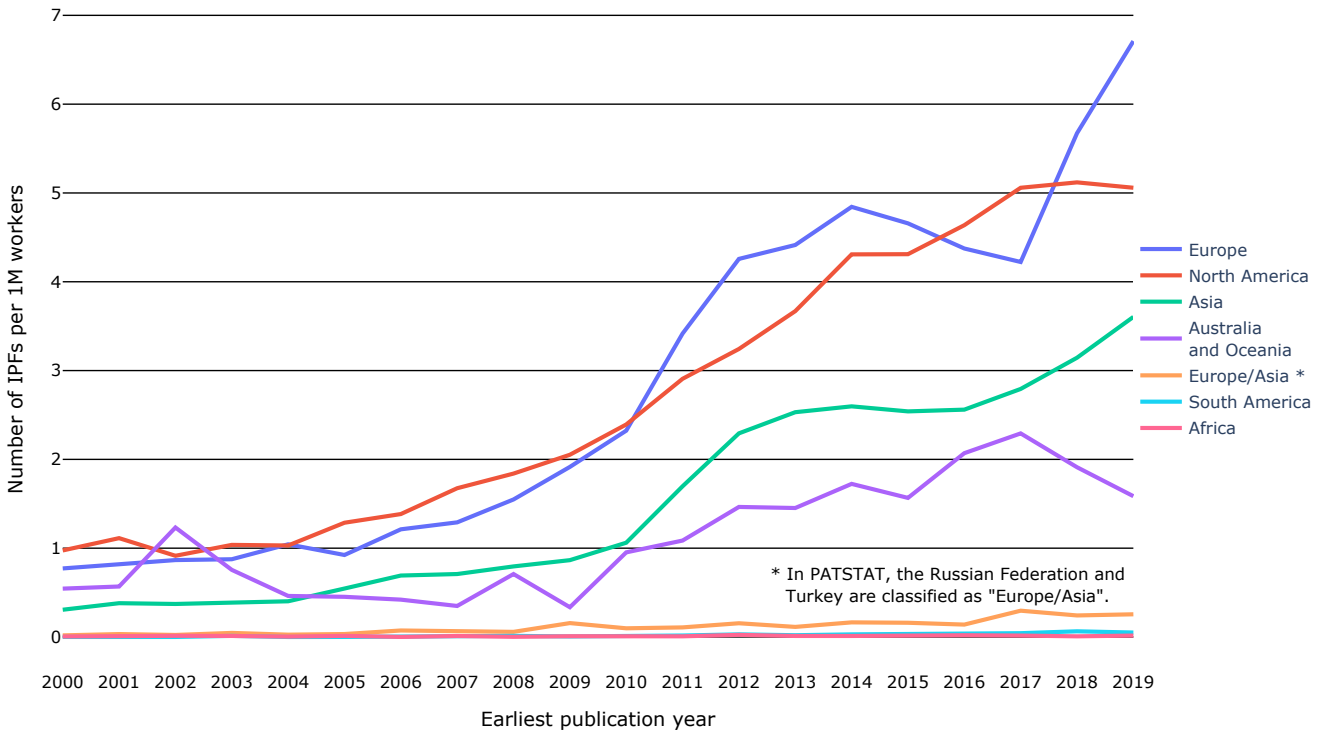

Figure S1. Battery IPFs per 1M workers by inventors' continents of origin, 2000-2019. In terms of battery IPF intensities, Europe and North America outperform Asia.

Data and Methods

The raw data

The foundation of this study is the PATSTAT database[1] provided by the European Patent Office, more precisely the Autumn 2021 edition of PATSTAT Online. Transact-SQL or T-SQL is the language used for querying it. The query designed for

selecting and downloading the data used for this study is defined in the text file "PATSTAT_Online_query.txt", which is included in the GitHub repository associated with this work, which can be found by following this link:

https://github.com/ph1001/battery_patents.git.

The patents that were downloaded from PATSTAT and that make up the raw dataset for this study were all patent applications (including ungranted) that are part of patent families whose intra-family value for the feature "earliest publication year" lies in the timeframe of 1999-2019 (the timeframe was later reduced to 2000-2019) and which contain at least one IPC entry matching one of the following codes: **H01M...** (processes or means, e.g. batteries, for the direct conversion of chemical energy into electrical energy), **H02J 3/32** (circuit arrangements for AC mains or AC distribution networks using batteries with converting means), **H02J 7...** (circuit arrangements for charging or depolarising batteries or for supplying loads from batteries), or **B60L 53...** (methods of charging batteries, specially adapted for electric vehicles; charging stations or on-board charging equipment therefor; exchange of energy storage elements in electric vehicles).

PATSTAT Online has the restriction that all SQL queries must begin with a "SELECT" statement, a fact that makes analyses of a higher complexity impossible to achieve inside PATSTAT Online itself. Consequently, data has to be queried, downloaded, and then processed in a different environment. The programming language used for all steps after querying the database and downloading the data was Python[2] (Version 3.9.7), more specifically the web application Jupyter Notebook[3] (Version 6.4.3), the data processing libraries pandas[4] (Version 1.3.3) and Numpy[5] (Version 1.20.3), the visualisation tools Plotly[6] (Version 5.1.0) and Seaborn[7] (Version 0.11.2), the text mining suite Natural Language Toolkit (NLTK)[8] (Version 3.6.5) and the analytics toolboxes Scikit-learn[9] (Version 0.24.2) and SciPy[10] (Version 1.7.1).

Ancillary sources were used. The labor force counts used for scaling were downloaded from the World Bank's website[11] and for the specific case of Taiwan from the website of "National Statistics: Republic of China (Taiwan)".[12]

Preprocessing and data reduction

Preprocessing and data reduction steps undertaken in order to obtain the final dataset from the raw data downloaded from PATSTAT are defined in the Jupyter Notebook "01_create_dataset.ipynb", which is included in the GitHub repository linked above. The following paragraphs contain a summary of these preprocessing steps.

First, the raw data downloaded from PATSTAT Online was loaded and checked for its integrity. Then each patent family's earliest intra-family values for the features "earliest publication date" and "earliest publication year" were determined and added as new columns to every row of the dataset (i.e. they were harmonized on patent family level). Like this, patent families can easily be assigned to their respective year later during the analyses. Next, all patent families were classified and tagged as either "IPF", "singleton", or "neither". The resulting tags are stored in the newly created column "tag". Next, more tags for further data selection were created. This process took place in five steps, which are described in the following:

- First, every patent family was scanned for the IPC codes related to non-active battery parts, electrodes, or secondary cells (IPC codes H01M 2..., H01M 50..., H01M 4..., and H01M 10...). Patent families that contained any of these codes were added in their entirety, except if they contained any of the IPC codes H01M 6..., H01M 8..., H01M 12..., H01M 14..., or H01M 16..., which are related to primary cells, fuel cells, hybrid cells, electrochemical current or voltage generators not provided for in groups H01M 6/00-H01M 12/00, and structural combinations of different types of electrochemical generators, which were hereby explicitly excluded from the analysis. The patent families passing this stage were tagged as "non-active parts, electrodes, secondary cells".

- In a second step, every patent family was scanned for the IPC codes related to "circuit arrangements for ac mains or ac distribution networks using batteries with converting means" (H02J 3/32), "circuit arrangements for charging or depolarising batteries or for supplying loads from batteries" (H02J 7...), "methods of charging batteries, specially adapted for electric vehicles" (B60L 53...), or "secondary cells; methods for charging or discharging" (H01M 10/44). Patent families that contained any of these codes were added in their entirety, except if they contained any of the IPC codes listed for exception in the above step or any of the codes B60L 53/54, B60L 53/55, or B60L 53/56 that refer to charging stations using fuel cells, capacitors, or mechanical storage means, respectively. Patent families that passed this stage were tagged as "charging".

- As a third step, in order to identify affiliations of the resulting patent families to a set of technological categories, each patent family's titles and abstracts were scanned using individual sets of regular expressions for each technology. These regular expressions are defined in the Jupyter notebook "01_create_dataset.ipynb". Titles and abstracts of all languages were considered and a patent family was selected in its entirety if any substring of its titles or abstracts matched any of the respective regular expressions. Please note that—in order to decrease the risk of false positives—before scanning abstracts for these regular expressions, they were cut off at the beginning of any appearance of the string "independent claims are also included for". The selected patent families were assigned the value 1 in the newly created columns with

the column name "is x", with $x \in$ {Lead-acid, Lithium-air, Lithium-ion, Lithium-sulfur, Other Lithium, Magnesium-ion, Nickel-cadmium, Nickel-iron, Nickel-zinc, Nickel-metal hydride, Rechargeable alkaline, Sodium-sulfur, Sodium-ion, Solid-state, Aluminium-ion, Calcium(-ion), Organic radical} being the name of the respective technology. Please note that due to the considerable overlap of the concept of solid-state batteries with other technologies, especially lithium-ion batteries, all patent families that were classified as patents related to solid-state batteries were untagged in any other category in which they acquired tags through the process described here. To be very clear: This especially means that the lithium-ion battery category does not contain any patent families that are tagged as solid-state battery inventions.

- The fourth step's purpose was to add patent data related to redox flow and nickel–hydrogen batteries to the dataset. For this purpose, a combination of IPC classes queries and text queries was deployed. The reason for this separate step is that redox flow and nickel–hydrogen batteries are closely related to fuel cells and, consequently, patents associated with them are often included in IPC classes that were excluded by the above steps. Analogous to the above steps, the IPC classes qualifying for potential inclusion were H01M 2..., H01M 50..., H01M 4..., H01M 8..., and H01M 10... and the IPC classes demanding exclusion were H01M 6..., H01M 12..., H01M 14..., and H01M 16.... Analogous to the above step, these patent families' titles and abstracts were then scanned using one set of regular expressions for redox flow and another for nickel–hydrogen batteries. These regular expressions can be reviewed in the Jupyter notebook "01_create_dataset.ipynb". All patent families that passed this stage were assigned the value 1 in the newly created columns with the names "is Redox flow" or "is Nickel–hydrogen", respectively.

- As the last step, another additional column was computed: The dataset column "technologies one hot sum" contains the sum across each row's "is <technology name>" values. This sum is needed in the rare cases where technology classifications overlap. The share of patent families that had more than one technology associated with them was 0.61% in the final dataset. The counts resulting from these overlapping technologies were not counted multiple times but, using the respective "technologies one hot sum" value, distributed as equal fractions across the overlapping classes.

The tags created in the above steps were used for selecting the appropriate data for each analysis. All patent families not having the "IPF" tag were filtered out before all analyses. That the rest was kept in the unfiltered dataset was only for completeness, having potential future analyses with a broader scope in mind. The data selection method that was applied before each analysis that is based on the labels whose creation was described above is presented in Fig. S2:

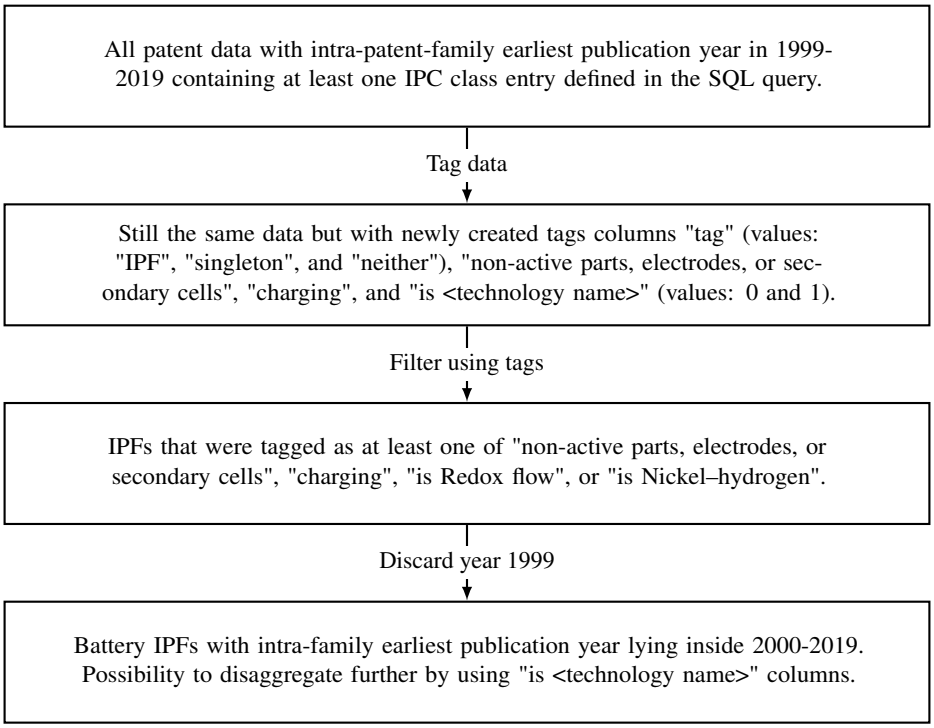

Figure S2. Flow chart depicting the data selection process for this study. The entire raw dataset was labeled using newly created columns. Before each analysis, the final dataset was acquired by filtering, using labels and timestamp columns.

Counting patents

As already mentioned in the Introduction, the methodological setup of this study roughly follows the framework defined in the report by IEA and EPO[13]. This means that all dates in this study refer to the earliest publication date within the respective IPF and that the geographic distributions were calculated based on the geographic information assigned to the respective inventors in PATSTAT and where multiple inventors are indicated, each inventor was assigned an equal fraction of the respective count. We believe there is a limitation to this approach, which is described in the following: For identifying the inventors, their PATSTAT name attribute "psn_name" is used. The harmonization of this feature, which was carried out by PATSTAT, is not complete. For example, pairs of entries like "SHIH, I-FEN" and "Shih, I-Fen" exist, which in reality correspond to the same inventor, but consequently are treated as two different individuals. This shifts the fractions of countries of origin in these entries' patent families in favor of the country of the unharmonized name.

The code used for counting patents by countries is contained in the Jupyter Notebook "02_counts_technologies_clustering.ipynb", which is part of the GitHub repository linked at the beginning of this section.

Methods: Battery technologies

Different from the report by IEA and EPO[13], in the current study fractional counting was also applied when breaking down counts by technological categories. Whenever an IPF was classified to belong to more than one category, each technology was assigned an equal fraction of the respective count. This only happened in a very small minority of the cases since only 0.61% of all IPFs were assigned to more than one technology. The code used for counting patents by technologies is contained in the Jupyter Notebook "02_counts_technologies_clustering.ipynb", which is part of the GitHub repository linked at the beginning of this section.

Methods: Clustering

The metric R^2 applied for comparing the performance of several clustering algorithms using varying numbers of clusters can be characterized as follows:

$$R^2 = \frac{SSB}{SST} = \frac{SST - SSW}{SST} = 1 - \frac{SSW}{SST} \in [0,1] \tag{1}$$

where

$$SSB = \sum_{i=1}^{p} n_i (\overline{X}_i - \overline{X})^2 = sum\ of\ squared\ differences\ between\ groups \tag{2}$$

and

$$SSW = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X_i})^2 = sum\ of\ squared\ differences\ within\ groups \tag{3}$$

and

$$SST = \sum_{i=1}^{p} \sum_{j=1}^{n_i} (X_{ij} - \overline{X})^2 = total\ sum\ of\ squared\ differences \tag{4}$$

with

$p = number\ of\ clusters,$

$n_i = number\ of\ elements\ in\ cluster\ i,$

$\overline{X_i} = centroid\ of\ cluster\ i,$

$\overline{X} = center\ of\ whole\ dataset,\ and$

$X_{ij} = jth\ element\ of\ cluster\ i.$

Relations 1 are true if and only if $SST = SSW + SSB$, which is the case because:

$$SST = \sum_{i=1}^{p}\sum_{j=1}^{n_i}(X_{ij}-\overline{X})^2 = \sum_{i=1}^{p}\sum_{j=1}^{n_i}(X_{ij}\underbrace{-\overline{X_i}+\overline{X_i}}_{=0}-\overline{X})^2 = \sum_{i=1}^{p}\sum_{j=1}^{n_i}(X_{ij}-\overline{X_i})^2 + \sum_{i=1}^{p}\sum_{j=1}^{n_i}(\overline{X_i}-\overline{X})^2 + 2\sum_{i=1}^{p}\sum_{j=1}^{n_i}(X_{ij}-\overline{X_i})(\overline{X_i}-\overline{X})$$

$$= \sum_{i=1}^{p}\sum_{j=1}^{n_i}(X_{ij}-\overline{X_i})^2 + \sum_{i=1}^{p}\sum_{j=1}^{n_i}(\overline{X_i}-\overline{X})^2 + 2\sum_{i=1}^{p}(\overline{X_i}-\overline{X})\underbrace{\overbrace{\sum_{j=1}^{n_i}(X_{ij}-\overline{X_i})}^{(5)}}_{=0} = \sum_{i=1}^{p}\sum_{j=1}^{n_i}(X_{ij}-\overline{X_i})^2 + \sum_{i=1}^{p}\sum_{j=1}^{n_i}(\overline{X_i}-\overline{X})^2$$

$$= \overbrace{\sum_{i=1}^{p}\sum_{j=1}^{n_i}(X_{ij}-\overline{X_i})^2}^{(3)} + \overbrace{\sum_{i=1}^{p} n_i(\overline{X_i}-\overline{X})^2}^{(2)} = SSW + SSB$$

with

$$\sum_{j=1}^{n_i}(X_{ij}-\overline{X_i}) = \sum_{j=1}^{n_i}X_{ij} - \sum_{j=1}^{n_i}\overline{X_i} = \frac{n_i}{n_i}\sum_{j=1}^{n_i}X_{ij} - n_i\overline{X_i} = n_i\overline{X_i} - n_i\overline{X_i} = 0 \tag{5}$$

Given a non-varying dataset and a fixed number of clusters, a higher R^2 value indicates a better clustering solution. Clustering algorithms that were compared are k-means and hierarchical agglomerative clustering using complete, average, single, and Ward linkage and the numbers of clusters that were tested ranged from two to nine.

The decision to use only the five dimensions "Lead-acid", "Redox flow", "Solid-state", "Sodium-ion", and "Lithium-sulfur" resulted from extensive testing of other configurations, especially those that included "Lithium-ion", "Other lithium" or a joint category of "Lithium-ion and other lithium". These tests were not found to be satisfying, since it was observed that the lithium-related IPFs were overshadowing the other categories due to their sheer amount, resulting in clustering solutions that lacked the clear interpretability of the solution presented in this work. Lithium-air batteries, another battery technology that has received increased attention in recent years,[14] was considered as a candidate feature for this analysis but was discarded due to its still very low yearly IPF counts. The code used for clustering countries based on their technology distribution is contained in the Jupyter Notebook "02_counts_technologies_clustering.ipynb", which is part of the GitHub repository linked at the beginning of this section.

Methods: Title and abstract mining

Unigrams, bigrams, and trigrams were extracted from cleaned abstracts and titles from which meaningless words and phrases had been removed and in which certain synonyms and anomalies had been treated. The n-gram counts method simply counts occurrences and displays them as yearly sums whilst the n-gram intensities method does the same with the difference that its resulting values are scaled using each years' numbers of abstracts or titles, respectively. Three ways for presenting the identified n-grams were designed for this study:

- Method 1a: Sorted in descending order of increase over the given timeframe of 2000-2019 with the measure used for sorting being $m_1 = count_{last} - count_{first}$.
- Method 1b: Sorted in ascending order of increase over the given timeframe of 2000-2019 with the measure used for sorting being m_1. This method's purpose is to show n-grams that exhibit a negative increase, i.e. have decreased over the given time period.
- Method 2: Sorted in descending order with the measure used for sorting being $m_2 = \sum abs(year-to-year\,difference_{i,i+1})$. This method's purpose is to show n-grams whose count or intensity changed the most (in absolute terms) between all adjacent years.

The results displayed in the tables that are presented in this study were obtained using method 1a, patent abstracts, and trigrams. The code for computing these results is contained in the Jupyter Notebook "03_title_and_abstract_mining.ipynb", which is part of the GitHub repository linked at the beginning of this section. The results obtained by using the methods and data combinations not presented in this paper can best be viewed by opening the HTML file "03_title_and_abstract_mining.html", which is also available in the same folder. The combinations for which results were computed can be characterized by the Cartesian product $c = \{n=1,\ n=2,\ n=3\} \times \{n-gram\ counts,\ n-gram\ intensities\} \times \{method\ 1a,\ method\ 1b,\ method\ 2\} \times \{titles,\ abstracts\}$.

References

1. De Rassenfosse, G., Dernis, H. & Boedt, G. An introduction to the patstat database with example queries. *Aust. Econ. Rev.* **47**, DOI: 10.1111/1467-8462.12073 (2014).
2. Van Rossum, G. & Drake, F. L. *Python 3 Reference Manual* (CreateSpace, Scotts Valley, CA, 2009).
3. Kluyver, T. *et al.* Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. & Schmidt, B. (eds.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87 – 90 (IOS Press, 2016).
4. McKinney, W. Data structures for statistical computing in python. In Stéfan van der Walt & Jarrod Millman (eds.) *Proceedings of the 9th Python in Science Conference*, 56 – 61, DOI: 10.25080/Majora-92bf1922-00a (2010).
5. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362, DOI: 10.1038/s41586-020-2649-2 (2020).
6. Plotly Technologies Inc. Collaborative data science (2015).
7. Waskom, M. *et al.* mwaskom/seaborn: v0.11.2 (august 2021), DOI: 10.5281/zenodo.5205191 (2021).
8. Bird, S., Klein, E. & Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit* (" O'Reilly Media, Inc.", 2009).
9. Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
10. Virtanen, P. *et al.* Scipy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods* **17**, 261–272, DOI: 10.1038/s41592-019-0686-2 (2020).
11. The World Bank: Labor force, total. https://data.worldbank.org/indicator/SL.TLF.TOTL.IN (2022). Accessed: 10. Jan. 2022, 12:07.
12. National Statistics, Republic of China (Taiwan): Labor force, total. https://eng.stat.gov.tw/ct.asp?xItem=42761&ctNode=1609&mp=5 (2022). Accessed: 10. Jan. 2022, 12:49.
13. IEA. Innovation in batteries and electricity storage, a global analysis based on patent data (2020).
14. Aaldering, L. J. & Song, C. H. Tracing the technological development trajectory in post-lithium-ion battery technologies: A patent-based approach. *J. Clean. Prod.* **241**, 118343, DOI: https://doi.org/10.1016/j.jclepro.2019.118343 (2019).

Figure legends

Copy-paste all figure legends here in the end