Data and Code for: Coordination dynamics between fuel cell and battery technologies in the transition to clean cars (Dugoua and Dumas, 2024)

Overview

This replication package contains all code for "Coordination dynamics between fuel cell and battery technologies in the transition to clean cars." The code in this replication package uses Python and Stata. One main file (`main.py`) runs all of the code to build the data for analysis, analyze it, and then output all the figures and tables used in the paper and appendix. Some of the raw data is confidential, so it is not included in this replication package. To facilitate replication, we have included intermediate datasets that enable the reproduction of the main results presented in the paper. We also include a series of "mock" files to illustrate the data structure of key datasets that we are not allowed to share.

The replication package contains three main directories:

1. `ipn` : this directory contains only code and, once the scripts have run, figures, tables, and other analysis files used in the paper. The files saved here are small and the directory's size should not exceed 50 MB.
2. `private_data` : this is a directory that should contain the confidential raw data that we are unable to share. Fully unzipped, PATSTAT Global 2022 takes up 340 GB.
3. `data_files` : this directory contains the raw data and intermediate files that we are able to share. Once all the scripts have run, this directory will also contain intermediate files created and saved by the cleaning and analysis scripts.

A replicator may choose to locate these different directories in different locations. To run the full pipeline, a replicator will need to acquire the confidential data and then specify the paths to these three directories. This readme contains more details on this and the architecture of the directories below.

Data Availability and Provenance Statements

Statement about Rights

We certify that the authors of the manuscript have legitimate access to and permission to use the data used in this manuscript.

We certify that the authors of the manuscript have documented permission to redistribute/publish the data contained within this replication package. Appropriate permission are documented in the LICENSE.md file.

License for Data

This replication package contains data that the authors constructed as well as secondary data from sources that allow reprinting. Data the authors constructed are licensed under a Creative Commons/CC-BY-NC license. See LICENSE.md for details. Further use of the data from external sources is governed by the relevant data providers. More details on those terms are contained below and in files associated with each data source.

Summary of Availability

Some data **cannot be made** publicly available.

Confidential data used in this paper and not provided as part of the public replication package will be preserved for 5 years after publication.

Note: If future researchers seek to extend this work using newer versions of the data rather than the versions described below, they may need to modify the code to account for changes in file names or contents.

Details on each Data Source

Name	Provided	Citation
“PATSTAT Global 2022 Spring Edition”	FALSE	European Patent Office (2022)
“Orbis IP Patstat Correspondence	FALSE	Bureau van Dijk
“Orbis”	FALSE	Bureau van Dijk
“MarkLines Automotive Industry Portal”	FALSE	MarkLines
“Factset Revere Supply Chain Relationships"	FALSE	Factset Revere
“IEA Data on RD&D Budgets”	FALSE	International Energy Agency (2019)
“Battery Prices”	FALSE	Ziegler MS, Trancik JE., 2021
“CPC-IPC Concordance”	TRUE	European Patent Office (2020)
“Clean Car National Strategies”	TRUE	Dugoua and Dumas (2024)
“Additional Data on RD&D Budgets”	TRUE	Dugoua and Dumas (2024)
“Clean, Grey, and Dirty Patent Codes”	TRUE	Dugoua and Dumas (2024)
“PPP conversion factors”	TRUE	World Bank
“US GDP deflator”	TRUE	World Bank

PATSTAT Global 2022 Spring Edition (European Patent Office, 2022)

- We purchased the 2022 Spring Edition of PATSTAT (Version 5.19) via the EPO website at https://www.epo.org/searching-for-patents/business/patstat.html. The current version of the PATSTAT database can be purchased at the same url.
- The unzipped `.csv` files should be placed under the directory `private_data/PATSTATGlobal2022/`. Fully unzipped, PATSTAT Global 2022 takes up 340 GB.
- The PATSTAT documentation provides detailed information about all PATSTAT variables. The documentation can be downloaded at https://www.epo.org/en/searching-for-patents/business/patstat.

Orbis IP Patstat Correspondence

- We constructed a correspondence between Orbis firm ids (bvdids) and PATSTAT patent ids (`Orbis_PATSTAT_updatePEM_anonymized.csv`) using Orbis Intellectual Property, which can be purchased at https://orbisintellectualproperty.bvdinfo.com (Bureau van Dijk).
- The correspondence file should be placed under `private_data/Patstat_orbis_correspondence/`.
- This replication package includes a small sample file (with random values) to illustrate the structure of the dataset.
- Variables:
  - "bvdid" is an anonymized firm identifier.
  - "appln_id" is the PATSTAT patent application identifier.
  - "docdb_family_id" is the PATSTAT patent family identifier.

Orbis

- We obtain information on BvD IDs, firm names and industry classification using Orbis, which can be purchased at https://login.bvdinfo.com/R0/Orbis (Bureau van Dijk). We've also extracted from Orbis the list of NAICS codes and subcodes as well as their label, of which a mock sample has been provided under `Other/NAICS_Codes_mock.csv`.

MarkLines Automotive Industry Portal

- We obtain data on car manufacturers' sales from Marklines. Access can be purchased at https://www.marklines.com/portal_top_en.html. We manually assembled a file called `OEMs.csv` using data from Marklines

and Orbis that contains information about corporate structure over time. To illustrate the structure of this file, we provided a mock example under `private_data/Marklines/OEMs_mock.csv`. For more information, see comments in `main.py`.

Factset Revere Supply Chain Relationships

- We obtain data on car manufacturers' suppliers from Factset Revere. Access can be purchased at https://www.factset.com/marketplace/catalog/product/factset-supply-chain-relationships.

IEA data on R&D Budgets (International Energy Agency, 2019)

- We accessed the data at https://www.iea.org/data-and-statistics/data-product/energy-technology-rd-and-d-budget-database-2.
- We downloaded the dataset called "Detailed Country RD&D Budgets" as a txt file, which should be placed under: `private_data/IEA/COUNTRY_BUDGETS.TXT`
- This replication package includes small sample files (with random values) to illustrate the structure of the datasets (all under `private_data/COUNTRY_BUDGETS_mock.TXT`).
- Variables:
  - `IEA_Detailed Country RD&D Budgets_ET_KA_17122019180935840.csv`.
    - "Country" contains the name of the country.
    - "Time" contains the year.
    - "Product" may contain for example "Total RD&D in Million USD (2017 prices and PPP)", "Total RD&D in Million NC (nominal)" or "Govt R&D in Million NC (nominal)".
    - "Flow" may contain categories such as: "GROUP 1: ENERGY EFFICIENCY", "23 CO2 capture and storage", "GROUP 3: RENEWABLE ENERGY SOURCES", "GROUP 4: NUCLEAR", "GROUP 5: HYDROGEN AND FUEL CELLS", "GROUP 6: OTHER POWER AND STORAGE TECHNOLOGIES", "GROUP 7: OTHER CROSS-CUTTING TECHS/RESEARCH", "GROUP 8: Unallocated", "21 Oil and gas", "22 Coal", "29 Unallocated fossil fuels"
    - "Value" contains the value corresponding to the particular product and flow. For the sample file, we provide a random value.
- For more information on variables, consult the IEA documentation.

Battery Prices: `private_data/Other/LiIon_price_series.csv`

- We obtained the data from Ziegler MS, Trancik JE., 2021, “Data series for lithium-ion battery technologies”, https://doi.org/10.7910/DVN/9FEJ7C, Trancik Lab Dataverse, V1, UNF:6:sVT2vBwWolbQL4BxsTSDUg== [fileUNF]
- Specifically, `LiIon_price_series.csv` contains the data shown in the tab "RepreSeries_Price_All_Cells" of the spreadsheet "LiIonDataSeries_represonly_withcover.xlsx".

CPC-IPC concordance (European Patent Office, 2020)

- The dataset `cpc-ipc-concordance.txt` can be downloaded from https://www.cooperativepatentclassification.org/cpcConcordances. It should be placed under `other_data_files/data/raw_data/green_patent_codes/`. The version we use and include here was downloaded on Oct 26, 2020.
- It contains a concordance between CPC and IPC codes. It is tab delimited.
- Variables:
  - Column 1 contains the CPC code, e.g., A01B1/028.
  - Column 2 contains the IPC code, e.g., A01B1/02.
  - Column 3 indicates the level of the CPC code. Codes of level 0 group codes of level 1, which group codes of level 2. For example, A01B1/02 (level 1) is embedded into A01B1/00 (level 0).

Clean, grey, and dirty patent codes: `GreenCPC_IPC_codes_fromliterature.xlsx`

- This file (comma delimited) is located under `other_data_files/data/raw_data/green_patent_codes/`. It contains the IPC and CPC codes related to clean, grey, and dirty technologies. We reviewed prior work in the literature to compile an initial list.

- The papers included are as follows:

- ADHMvR2016: Aghion, Philippe, Antoine Dechezleprêtre, David Hemous, Ralf Martin, and John Van Reenen. 2016 - Carbon Taxes, Path Dependency, and Directed Technical Change - Evidence from the Auto Industry
- DMM2019: Dechezleprêtre, Antoine, Ralf Martin, and Myra Mohnen. 2019. “Knowledge Spillovers from Clean and Dirty Technologies: A Patent Citation Analysis.”
- JHP2010: Johnstone, Nick, Ivan Haščič, and David Popp. 2010. “Renewable Energy Policies and Technological Innovation: Evidence Based on Patent Counts.” Environmental & Resource Economics 45 (1): 133–55.
- LVH2011: Lanzi, Elisa, Elena Verdolini, and Ivan Haščič. 2011. “Efficiency-Improving Fossil Fuel Technologies for Electricity Generation: Data Selection and Trends.” Energy Policy 39 (11): 7000–7014.
- PPH2020: Popp, David, Jacquelyn Pless, Ivan Hascic, and Nick Johnstone. 2020. “Innovation and Entrepreneurship in the Energy Sector.” In The Role of Innovation and Entrepreneurship in Economic Growth. University of Chicago Press.
- ENVTECH: OECD Environment Directorate Patent search strategies for the identification of selected environment-related technologies (ENV-TECH)

- We then reviewed the code one by one to verify the classification and amend it if needed. We then reviewed the code to decide which should be included or not in the analysis (which focuses on a subset of technologies). The column "Include" flags the categories that are included. The Python script `finding_all_CPCIPC_subgroups.py` reads in `CPC_IPC_codes_fromliterature.xlsx` to collect all the IPC and CPC subgroup-level codes that are embedded in higher-level codes shown on the spreadsheet. For example, Y02E10/10 is embedded in Y02E10. The script outputs the file `CPC_IPC_codes_allsubgroups.csv` (within `other_data_files`), which contains all the subgroups of the chosen IPC and CPC codes. It is a longer file (several thousand rows). N.B.: The two files have the variable "Code" in common.

- Variables:

  - The columns "Code" and "Description" contain the code and description of the code as reported in the paper.
  - The column "Include" indicates whether the code should be included or not: 0 means exclude, 1 means include in all analysis, 2 means include for some descriptive results but not for the main analysis.
  - The column "InclusionNotes" provides more information why a particular code is included or not if necessary.
  - "Scheme" indicates whether the code was found in either the CPC only, in both IPC and CPC, or in neither using the concordance file.
  - Type, Sector, and Subsector were assigned based both on what previous papers had done but with some modifications based on our own reassessment.
  - The column "Notes" provides more details when necessary. For example, when a code is reported in several papers but in slightly different ways, the descriptions were reviewed and the one judged most appropriate was chosen. In these cases, a note was added in the Notes column. Another example is that, when "type" is empty, a note will explain why (e.g., because the code on its own is not sufficient to infer its type).
  - "SupCode_DifferentType" records if there is a superior code in the hierarchy with a different type. e.g., F23 is dirty but F23B10 is grey. This column is useful when querying all the subgroups in the script `finding_all_CPCIPC_subgroups.py`. For example, the script `finding_all_CPCIPC_subgroups.py` will code all the codes embedded in F23 as dirty except those embedded in F23B10 which will be coded as grey.
  - "paper" indicates all the papers that contain that code. The Y code are by construction always included in ENVTECH but we don't add this under the variable "paper".
  - "Nbr_papers_present" indicates the number of papers that include the code though, for this, we don't consider ENVTECH as a paper.
  - "Code_group" contains everything before the "/". This may be at the group level, but also at the subclass and class level.

Additional Data on RD&D Budgets: `Additional_R&D_Data_manuallycollected.xls`

- This file is located under `data_files/raw_data/PolicyLandscaping/`. It contains additional information of country-level RD&D budgets for specific years which were collected from various sources as detailed on Appendix Table D.9.
- See LICENSE.md for license terms.

Clean Car National Strategies: `country_tech_strategy_timeline_table.csv`

- This file is located under `data_files/raw_data/PolicyLandscaping/`. It contains the final coding of strategic orientation of each country over time. Details are explained in Appendix Section D.1 "Strategic Orientation Policy Data".
- See LICENSE.md for license terms.

World Bank GDP deflator and PPP conversion

- The files `GDP_US_deflator_WB.csv` and `PPP_conversion_WB.csv` are located under `data_files/raw_data/Other/`.
- The GDP PPP conversion of local currency to US dollars were downloaded at https://data.worldbank.org/indicator/PA.NUS.PPP
- The GDP deflator series for the US were downloaded at https://data.worldbank.org/indicator/NY.GDP.DEFL.ZS.AD?view=chart&locations=US
- See LICENSE.md for license terms.

Intermediate Datasets

To facilitate replication, this package includes intermediate files under `data_files/provided_intermediates/`. Below is a description of the variables in these files.

`TimeSeries_FamilyCounts_in_ipc_cpc_transpo_from_OEM_and_Subsidiaries_inter.csv` :

- "earliest_filing_year": Year of filing
- "Count_Bat_excl": Count of battery patents filed by OEMs and their subsidiaries, using a definition of battery patent that classifies it exclusively as a battery patent (see Figure B.1)
- "Count_FC_excl": Count of fuel cell patents filed by OEMs and their subsidiaries, using a definition of fuel cell patent that classifies it exclusively as a fuel cell patent (see Figure B.1)

`Panel_OEMs_level1_with_subsi_FamInfo_inter.csv` :

- "OEM_Level1_ID": Unique ID for OEMs created internally to the project
- "name": Name of the OEM
- "earliest_filing_year": Year of filing
- "Count_Bat_excl": Count of battery patents filed by the OEM and its subsidiaries, using a definition of battery patent that classifies it exclusively as a battery patent (see Figure B.1)
- "Count_FC_excl": Count of fuel cell patents filed by by the OEM and its subsidiaries, using a definition of fuel cell patent that classifies it exclusively as a fuel cell patent (see Figure B.1)
- "shortname": Shorter OEM names used for figure

`firm_year_policy_exposure_inter.csv` :

- "OEM_Level1_ID": Unique ID for OEMs created internally to the project
- "Level1_MLName": Name of the OEM
- "Year"
- "strat_fc_exposure": Exposure of the OEM to the strategic orientation policies of governments towards fuel cell cars, from 0 to 1, where 1 indicates that there is a strategic orientation towards fuel cell cars in all export markets of the OEM; exposure is computed as a weighted average across export markets, using OEM-level 2004 export market shares
- "strat_bev_exposure": Exposure of the OEM to the strategic orientation policies of governments towards battery electric cars, from 0 to 1, where 1 indicates that there is a strategic orientation towards battery electric cars in all export markets of the OEM; exposure is computed as a weighted average across export markets, using OEM-level 2004 export market shares
- "fchydrogen_RDexposure": Exposure of the OEM to public RDD funding targeted at fuel cells and hydrogen, in 2018 Million USD; exposure is computed as a weighted average across export markets, using OEM-level 2004 export market shares
- "otherstorage_RDexposure": Exposure of the OEM to public RDD funding targeted at 'other power and storage technologies' (group 6 of IEA RDD categories), in 2018 Million USD; exposure is computed as a weighted average across export markets, using OEM-level 2004 export market shares

`Families_Bat_FC_naics_info_inter.csv` :

- "earliest_filing_year": Year of filing

- "pid": Patent family identifier created internally as replacement of Patstat's docdb_family_id, for purpose of sharing
- "fid": Firm identifier created internally as replacement of Orbis' bvdid, for purpose of sharing
- "OEMorSubsidiary": Dummy indicating whether patent was filed by an OEM or an OEM subsidiary
- "Sub-sector_exclusive": Categorical variable, either "battery" or "fuel cell"
- "MotorVehicle": Dummy for NAICS codes 3361, 3362, or 3363
- "Electronics": Dummy for NAICS codes 334 or 335
- 'MachineryChemical': Dummy for NAICS codes 333 or 325
- 'OtherTransport': Dummy for NAICS code 336, excluding 3361, 3362 and 3363

`Panel_suppliers_FamInfo_inter.csv` :

- "sid": Supplier firm identifier created internally as replacement of Factset's id, for purpose of sharing
- "earliest_filing_year": Year of filing
- "Active": Dummy if supplier has a supplier-buyer relationship with an OEM in that year
- "4digitNAICS": Primary NAICS code at 4-digit level (can be more than one)
- "oldguard": Dummy specifying if the supplier is part of the "old guard", defined as having had an observable active supplier-buyer relationship with an OEM prior to 2009
- "Count_Bat_excl": Count of battery patents filed by supplier, using a definition of battery patent that classifies it exclusively as a battery patent (see Figure B.1)
- "Count_FC_excl": Count of fuel cell patents filed by supplier, using a definition of fuel cell patent that classifies it exclusively as a fuel cell patent (see Figure B.1)

`Panel_Activesuppliers_Stocks_inter` :

- "OEM_Level1_ID": Unique ID for OEMs created internally to the project
- "Year"
- "Active": Dummy if supplier has a supplier-buyer relationship with an OEM in that year
- "Years_since_firstactive": Number of years since the supplier-buyer relationship became active
- "sid": Supplier firm identifier created internally as replacement of Factset's id, for purpose of sharing
- "Stock_Bat": The stock of battery patent families of the supplier, calculated as the cumulative discounted sum of families since 1980, discounted by 15% each year
- "Stock_Bat": The stock of fuel cell patent families of the supplier, calculated as the cumulative discounted sum of families since 1980, discounted by 15% each year

Mock Datasets

We also include a series of "mock" files to illustrate the data structure of key datasets that we are not allowed to share. These "mock" files do not contain any real data. Information about these specific files is included in comments in `main.py` next to the relevant scripts.

Computational requirements

Software Requirements

The code is written in multiple languages:

1. `.py` scripts were last executed using Python version 3.9.12. The scripts use the following packages:
   - Package `logging` : version 0.5.1.2
   - Package `pandas` : version 1.4.2
   - Package `platform` : version 1.0.8
   - Package `numpy` : version 1.21.5
   - Package `matplotlib` : version 3.5.2
   - Package `seaborn` : version 0.11.2
   - Package `sklearn` : version 1.0.1

- Package `textacy` : version 0.11.0
- Package `scipy` : version 1.7.1
- The scripts also use packages that are part of the Python Standard Library (`sys` , `os` , `subprocess` , `time` , `warnings` , `pathlib`).

1. `.do` scripts were last executed using Stata/SE version 14. The script code/setup.do includes a command that sets the version to 14, which should replicate the behavior of version 14 on Stata versions 14 and higher. The script `main.py` may need to be modified to replace instances of "stata-se" if using another edition of Stata such as Stata/MP. Stata code relies on several user-written packages, which are included in this repo in `code/ado` . They may require device-specific setup/compilation. The end of code/setup.do contains more details on the packages and their installation.

Controlled Randomness

No Pseudo random generator is used in the analysis described here.

Memory, Runtime, Storage Requirements

Summary

Approximate time needed to reproduce the analyses on a standard (2024) desktop machine: 3 days.

Approximate storage space needed: about 14 GB for files outputted under `data_files/outputted_data/` . Additional space will be needed for the raw data. Fully unzipped, PATSTAT Global 2022 takes up 340 GB.

Details

The code was last run on the following desktop:

- OS: "Ubuntu 20.04.6 LTS"
- Processor: Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz, 12 cores
- Memory available (RAM): 93 GB memory

Description of programs/code

High-Level Overview

- We provided a high-level overview of the program files and their purpose in the script `main.py` .
- For example, the code snippet below shows the line that will call and run the python script `a_figures_maintext.py` . We also include a comment explaining what the script does before the lines that call and execute the script.

```
# Outputs the key figures shown in the manuscript
from E_Analysis import a_figures_maintext
a_figures_maintext.main()
```

Scripts for Replication from Intermediates

- Some scripts are specifically dedicated to allow replication from intermediate files.
- `main_inter.py` is the main file that will run the analysis steps that can be run using the intermediate files provided. It outputs the figures used in the main paper.

Python Initializations

- The script `init.py` (executed in the preamble of `main.py` and `main_inter.py`) contains a series of initialization that are critical to the execution of the pipeline:
  - The function `set_log()` sets up the log environment, which will save log files under `output/log` .

- The function `set_paths()` sets up the paths. If no paths are specified for the replicator's platform, it will read the path from `dbpath.do`.
- The function `create_folderstructure()` creates the folder structure, i.e., creates all directories necessary to save intermediate data, figures and tables.

Other Notes

- The python script `a_figures_maintext.py` reproduces all figures provided in the paper. We have provided comments inside the script to indicate which part of the code replicates the different figures.

License for Code

The code is licensed under a MIT license. See LICENSE.md for details.

Instructions to Replicators

Replication from Raw Data

1) Download and unzip all data files referenced above, storing them where described above.

2) Edit `init.py` (line 53 to 66) to adjust the default paths to `data_files`, `private_data` but also to PATSTAT, Orbis and Factset. As mentioned above, the directories `data_files` and `private_data` can be located inside or outside this package's main directory for convenience due to file size. The code requires to set a separate paths for the data related to PATSTAT, Orbis and Factset as we assume these datasets would be located elsewhere (again, for convenience or due to other constraints).

3) Edit `dbpath.do` to adjust the default paths to `data_files`:

```
global dropbox "/ADD/PATH/TO/DATA_FILES/DIRECTORY/data_files"
```

4) Run `code/main.py` to run all steps in sequence, preferably by calling it from the command line within the working directory ipn. For example, this can be done as follows in the command line (code for Ubuntu):

```
$ cd ~/ipn/code     # to navigate to the directory where main.py is located
$ source activate py39   # to active the right Python environment (3.9 in this case)
$ nohup python main.py &> main.out&   # to run all steps in sequence.
```

Log files will be saved under `output/log`. The file `main.out` in the example above provides an additional wa

Replication from Intermediate Data

1) Editing `init.py` and `dbpath.do` as above.

2) Run `code/main_inter.py` to run all steps in sequence.

References

Bureau van Dijk. (n.d.). Orbis Intellectual Property. https://orbisintellectualproperty.bvdinfo.com.

Dugoua, Eugenie, and Marion Dumas. 2024. “Replication data for: Coordination dynamics between fuel cell and battery technologies in the transition to clean cars."

European Patent Office. 2020. CPC-IPC concordance. https://www.cooperativepatentclassification.org/cpcConcordances. Accessed October 26, 2020.

European Patent Office. 2022. PATSTAT Global 2022 - Single Edition (Spring). https://www.epo.org/searching-for-patents/business/patstat.html.

FactSet Revere. (n.d.). Supply Chain Relationships data. https://www.factset.com/marketplace/catalog/product/factset-supply-chain-relationships.

International Energy Agency. 2019. International Energy Agency Energy Technology Research and Development Database, 1974-2017. https://doi.org/10.5257/iea/et/2018-09.

MarkLines. (n.d.). Automotive Industry Portal. https://www.marklines.com/.

World Development Indicators database, World Bank. https://databank.worldbank.org/source/world-development-indicators. Accessed April 17, 2023.

Ziegler MS, Trancik JE., 2021, “Data series for lithium-ion battery technologies”, https://doi.org/10.7910/DVN/9FEJ7C, Trancik Lab Dataverse, V1, UNF:6:sVT2vBwWolbQL4BxsTSDUg== [fileUNF]

Acknowledgements

This README is based on the template README for social science replication packages.