Readme of the replication package for "Comprehensive national accounting for CO_2 emissions"

Geir B. Asheim* Rintaro Yamaguchi†

October 24, 2025

Overview

Welcome! This replication package is composed of the following files:

- `JAERE_Readme.pdf` (this file) contains main instructions
- `JAERE_carbon_julia.txt` is a Julia code for computing country-level (national) marginal damages using MimiGIVE
- `scc_noag.py.txt` is a Python code for computing national SCC (NSCC) for the non-agricultural sector
- `scc_agri.py.txt` is a Python code for computing national SCC (NSCC) for the agricultural sector
- `country_SCC_JAERE.xlsx` is an MS Excel spreadsheet file that includes NSCC and other data to compute the full sustainability indicators (FSIs) and limited sustainability indicators (LSIs). In this spreadsheet, the cells you are expected to put data for replication purposes are indicated by blue, which are already filled in by the authors using APIs. This file also contains Figure 1 and Figure G in Appendix G.
- `JAERE_Fig2.py.txt` is a Python code for drawing Figure 2

*Department of Economics, University of Oslo, P.O. Box 1095 Blindern, 0317 Oslo, Norway. E-mail: `g.b.asheim@econ.uio.no` ORCiD: 0000-0003-0669-4632

†National Institute for Environmental Studies (NIES), 16-2 Onogawa, Tsukuba 305-8506, Japan. E-mail: `yamaguchi.rintaro.r41@kyoto-u.jp` ORCiD: 0000-0002-0470-1483

- `JAERE_Fig3.py.txt` is a Python code for drawing Figures 3 and 4
- `JAERE_Fig1_emission_global.png`, `JAERE_Fig1_emission_global_ch.png`, `JAERE_Fig2_GS_LSI.png`, `JAERE_Fig3_wb194_LSI1.png`, and `JAERE_Fig4_wb194_LSI2.png` are output figure files
- `JAERE_carbon_price.do` is the Stata Do file for the regression of carbon prices on GDP per capita
- `carbon_price.dta` and `carbon_price_estimate.dta` are raw and cleaned carbon prices dataset, respectively

Software requirements

We assume that the user has downloaded and installed Julia (Version 1.11.2), Python (Version 3.12.4), and Stata (17.0 MP).

Memory and Runtime requirements

Step 1 below is a time- and memory-intensive exercise. It takes at least several hours, and perhaps one day, depending on the replicator's computer, to complete a marginal damage computation by 10,000 Monte Carlo simulation (MCS) runs for one year (e.g., 2020, 2025 etc.). It takes several Gigabytes per one year NSCCs. If the purpose is not a strict replication of our paper but a trial, we recommend 1,000 MCS runs.

OK, let's get started.

1 Running the GIVE model to compute country-level marginal damages in Julia

Refer to the GIVE model repository for the details of the model.

Start Julia. Install Mimi and MimiGIVE if you have not done so. This can be done by entering] and

```
(@v1.10) pkg> add Mimi
(@v1.10) pkg> add MimiGIVE
```

Copy and paste the commands contained in the `JAERE_carbon_julia.txt` file to calculate country-level (national) marginal damages.

```
# Import Mimi and MimiGIVE
using Mimi
using MimiGIVE

# Define etas for the 8 consumption discount rates
eta1=1.01601
eta2=1.244458999
eta3=1.421158088
eta4=1.567899391
eta0=0

# Define deltas for the 8 consumption discount rates (CDRs)
# These are taken from the MimiGIVE repository https://github.com/rffscghg/MimiGIVE.jl
delta1=9.15002/100000
delta2= 0.001974588
delta3=0.004629467
delta4=0.007732453
delta5=0.015
delta6=0.02
delta7=0.025
delta8=0.03

# Define the 8 consumption discount rates that are employed in the Nature paper (Rennert et al. 2022)
discount_rates = [(label="Ramsey1.5", prtp=delta1, eta=eta2), (label="Ramsey2.0", prtp=delta2, eta=eta2), (label="Ramsey2.5", prtp=delta3, eta=eta3), (label="Ramsey3.0", prtp=delta4, eta=eta4), (label="Constant 1.5%", prtp=0.015, eta=0.) , (label="Constant 2%", prtp=0.02, eta=0.), (label="Constant 2.5%", prtp=0.025, eta=0.), (label="Constant 3%", prtp=0.03, eta=0.)]
```

Run the model to cumpute national marginal damages and global SCCs. The following is an example of SCC of 2020, based on 10,000 Monte Carlo simulation (MCS) runs. Note that this is quite memory-intensive exercise that involves at least several hours, so you might want to try only 1,000 runs, which is considered to be sufficient.

```
result2020 = MimiGIVE.compute_scc(year = 2020, discount_rates = discount_rates, compute_disaggregated_values=true, save_cpc=true, n = 10000)
```

After this MCS run,

```
# Check global scc (GSCC) of 2020 for the 8 consumption discount rates
scc2020 = result2020[:scc]
```

These results are documented in C191:G191 of the "2020" sheet in the Excel we mention in 4. Then move on to the calculation of consumption discount factors:

```
# Save net_cpc (consumption per capita net of damage) for the
    purpose of the computation of CDRs
net_cpc = result2020[:cpc][(region=:globe, sector=:total)]

# Define consumption discount factors (CDFs) for 8 CDRs
# They correspond to equation (6) of (Rennert et al. 2022)

consdf_ram15=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta1)*1/(1+delta1)^t
    for a in 1:10000, t in 0:280]
consdf_ram20=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta2)*1/(1+delta2)^t
    for a in 1:10000, t in 0:280]
consdf_ram25=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta3)*1/(1+delta3)^t
    for a in 1:10000, t in 0:280]
consdf_ram30=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta4)*1/(1+delta4)^t
    for a in 1:10000, t in 0:280]
consdf_const15=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta0)*1/(1+delta5)^
    t for a in 1:10000, t in 0:280]
consdf_const20=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta0)*1/(1+delta6)^
    t for a in 1:10000, t in 0:280]
consdf_const25=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta0)*1/(1+delta7)^
    t for a in 1:10000, t in 0:280]
consdf_const30=[(net_cpc[a,t+1]/net_cpc[a,1])^(-eta0)*1/(1+delta8)^
    t for a in 1:10000, t in 0:280]
```

MCS should have created new directories which contain marginal damage (md) Excel files, to which we now move. First, go to the folder of marginal damage at the country level with no agricultural damage. **In the following code, please update the file path to match the appropriate location on your local machine**:

```
cd("C:/Users/USERNAME/.julia/packages/MimiGIVE/3oCQU/output/mcs-SC/
    MCS 2025-03-14 16-02-51 MC10000/results/disaggregated_values/
    mds_country_no_ag")

# Save 8 CDFs in the same directory
using HDF5
h5write("consdf_ram15.h5", "consdf_ram15", consdf_ram15)
h5write("consdf_ram20.h5", "consdf_ram20", consdf_ram20)
h5write("consdf_ram25.h5", "consdf_ram25", consdf_ram25)
h5write("consdf_ram30.h5", "consdf_ram30", consdf_ram30)
h5write("consdf_const15.h5", "consdf_const15", consdf_const15)
h5write("consdf_const20.h5", "consdf_const20", consdf_const20)
h5write("consdf_const25.h5", "consdf_const25", consdf_const25)
h5write("consdf_const30.h5", "consdf_const30", consdf_const30)
```

After this, you should have all the country-level damage and consumption discount factor files in a directory that looks like this:

`C:\Users\USERNAME\.julia\packages\MimiGIVE\3oCQU\output`

Repeat this step for all the years for which we compute the NSCCs. In our JAERE paper (Asheim and Yamaguchi), we have computed SCCs for 2020, 2025, 2030, 2040, 2050, 2060, 2070, 2080, 2090, 2100, 2150, 2200, and 2250. For the 2025 NSCCs, for example, an MCS run command is substituted by:

```
result2025=MimiGIVE.compute_scc(year=2025, discount_rates =
    discount_rates, compute_disaggregated_values = true, save_cpc=
    true, n=10000)
```

and the computation of net consumption per capita by:

```
net_cpc = result2025[:cpc][(region=:globe, sector=:total)]
```

2 Computing the discounted non-agricultural marginal damages in Python

Go to a Python IDE and start a command prompt, e.g. Anaconda Powershell Prompt.

Move to the directory of undiscounted marginal damages you have created in step 1, first for the non-agricultural sector, and second for the agricultural sector.

The file path shown below is an example. The actual file path includes the user's name, a subfolder with a randomly generated identifier, and another subfolder based on the date and time.

```
(base) PS C:\Users\USERNAME> cd "C:\Users\USERNAME\.julia\packages\
    MimiGIVE\3oCQU\output\mcs-SC\MCS 2025-03-14 16-02-51 MC10000\
    results\disaggregated_values\mds_country_no_ag"
```

Put the `scc_noag.py.txt` file there.

Run the Python code to compute NSCCs for the non-agricultural sector:

```
(base) PS C:\Users\USERNAME\.julia\packages\MimiGIVE\3oCQU\output\
    mcs-SC\MCS 2025-03-14 16-02-51 MC10000\results\
    disaggregated_values\mds_country_no_ag> python scc_noag.py.txt
```

You should have now obtained the NSCCs for the non-agricultural sector in the MS Excel format in the same directory.

3 Computing the discounted agricultural marginal damages to obtain NSCCs in Python

Now, repeat the same exercise for the agricultural sector. Move to the Julia directory of undiscounted marginal damages for the agricultural sector:

```
(base) PS C:\Users\USERNAME> cd "C:\Users\USERNAME\.julia\packages\
    MimiGIVE\3oCQU\output\mcs-SC\MCS 2025-03-14 16-02-51 MC10000\
    results\disaggregated_values\mds_country_ag_only"
```

Put the `scc_agri.py.txt file` there.

Also, copy and paste the following 8 consumption discount factor files from the non-agricultural sector folder to the agricultural sector folder:

`consdf_const15.h5`
`consdf_const20.h5`
`consdf_const25.h5`
`consdf_const30.h5`
`consdf_ram15.h5`
`consdf_ram20.h5`
`consdf_ram25.h5`
`consdf_ram30.h5`

Run the Python code to compute NSCCs for the agricultural sector:

```
(base) PS C:\Users\USERNAME\.julia\packages\MimiGIVE\3oCQU\output\
    mcs-SC\MCS 2025-03-14 16-02-51 MC10000\results\
    disaggregated_values\mds_country_ag_only> python scc_agri.py.
    txt
```

You should have now obtained the NSCCs for the agricultural sector in the Excel format in the same directory.

4 Summarizing the NSCCs and other data in MS Excel

From here the replicator is expected to work on MS Excel. You can just use the prepared file `country_SCC_JAERE.xlsx` in the replication package, where all you need to do is to put your outputs in the blue cells. These cells are

- Future emissions: Median RFF-SPs CO_2 emissions for 2020–2300
- Future national emission share: Climate Action Tracker for 2019–2050

- Past emissions: Global Carbon Budget annual CO_2 emissions for 1850-2023 (of which only the recent past data for 2010-2019 is used)
- GDP per capita (current US$) from WDI for 2020-2023
- Genuine saving: Adjusted net savings excluding particulate emission damage (current US$) from WDI for 2010–2019
- GNI (current US$) from WDI for 2010–2019

However, for those who want to create the spreadsheet from scratch, the instructions are as follows.

4.1 Summarizing the NSCCs data

Create a new spreadsheet `country_SCC_JAERE.xlsx` that contain sheets titled E, Edot, FSI, LSI, WB, Fmi, MACa, MACb, CAT, Share, GS, CurrDam, R1.5, R2.0, R2.5, R3.0, C1.5, C2.0, C2.5, C3.0, 2020, 2025, 2030, 2040, 2050, 2060, 2070, 2080, 2090, 2100, 2150, 2200, and 2250.

Copy the NSCC values for both the non-agricultural and agricultural sectors in a "2020" sheet, and paste them in the `country_SCC_JAERE.xlsx` file.

Since we have 183 countries and the following 8 discount rates:

```
ram_1.5, ram_2.0, ram_2.5, ram_3.0,
const_1.5, const_2.0, const_2.5, and const_3.0,
```

paste the non-agricultural NSCCs in L3-S185 cells.

Likewise, paste the agricultural NSCCs in U3-AB185 cells.

Compute total NSCCs of 2020 by summing the non-agricultural and agricultural sectors in C3-J185 cells.

Finally, sum the NSCCs to compute the GSCCs in C187-J187 cells.

Now **Table F1–2** has been made.

Repeat this exercise for the years you want to study in separate sheets in the `country_SCC.xlsx` file. In our paper (Asheim and Yamaguchi JAERE), we have computed the following 13 years in the "2020", "2025", "2030", "2040", "2050", "2060", "2070", "2080", "2090", "2100", "2150", "2200", and "2250" sheets.

4.2 Importing genuine saving data from the World Bank API

Go to the World Development Indicators and download the following two series for all the studied countries for the recent past data in CSV or XLSX format:

- Adjusted net savings, excluding particulate emission damage (current US$)
- GNI (current US$)

In our paper (Asheim and Yamaguchi JAERE), we have used 2010-2019 data, but the most recent one might be sufficient.

Copy and paste the above CSV to the "GS" sheet in the `country_SCC_JAERE.xlsx` file.

Take the average of 2010-2019 in the column AG which is interpreted as the current genuine savings.

Copy GS data in C5:C187 in the sheets "R1.5", "R2.0", "R2.5", "R3.0", "C1.5", "C2.0", "C2.5", and "C3.0" that show the results in 8 different consumption discount rates (CDRs) in the `country_SCC_JAERE.xlsx` file.

4.3 Calculating the *Cost of current emissions* term in FSI and LSI

In the "CurrDam" sheet of the `country_SCC_JAERE.xlsx` file, write the 8 discount rates in column names in C2-J2 cells, and 183 country names as row names in B3-B185 .

Download the `co2-fossil-plus-land-use.xlsx` file from the Global Carbon Budget. Import past annual CO_2 emissions from fossil fuels and land-use change data of 1850-2023 in C55:FT55 in the "E" sheet.

Fill in the C1 cell by the current global annual emission, 40.3 $GtCO_2$ (billion tCO_2), as the average of CO_2 emissions from fossil fuels and land-use change in 2010-2019, copied from FG:FP55 of "E" sheet.

In C3-J185 cells, multiply 40.3 billion tCO_2 in the C1 cell and 2020 NSCCs in the "2020" sheet. Take the global sums in C187-J187 cells.

4.4 Importing future global emissions data

Download the median and mean RFF-SPs future CO_2 emissions data for 2020-2300 from the Zenodo repository.[1]

Copy and paste them in C2:JW2 in the "E" sheet in the `country_SCC_JAERE.xlsx` file. Only the median RFF-SPs future CO_2 emissions will be used, while the mean path is used only in Figure 1.

Since they are expressed in billion tC, multiply them by 44/12 to obtain the median RFF-SPs future CO_2 emissions in billion tCO_2 in C5:JW5.

4.5 Importing future country emissions of Climate Action Tracker data

Go to Climate Action Tracker and download "Insufficient" pathways for the studied countries for the years for 2019 to 2050.

Paste them in B3:AG41 in the "CAT" sheet. Sum the numbers in row 42 as "CATtotal".

4.6 Computing future country emission shares

In the "Share" sheet, compute the country shares by dividing country emissions by total emissions and multiplying with `C1`=0.8362371761 in `B3:JY42`. From 2051 to 2300, the country shares are constant and equal to the 2050 level. For example, the B3 cell for Argentina in 2019 looks like:

```
1    =CAT!B3/CAT!B$42*Share!$C$1
```

Compute the country emissions in C9:JW49 cells in the "E" sheet by multiplying the country shares in the "Share" sheet with the median RFF-SPs future CO_2 emissions in billion tCO_2 in C5:JW5 in the "E" sheet.

4.7 Computing the *Future emission increase* term in FSI and LSI

Take the time difference in the global emissions in the P3:KJ3 cells in "R1.5", "R2.0", "R2.5", "R3.0", "C1.5", "C2.0", "C2.5", and "C3.0" sheets.

Multiply them with corresponding NSCCs in P5:KJ187 cells.

[1] Kevin Rennert, Brian C. Prest, William A. Pizer, Richard G. Newell, David Anthoff, Cora Kingdon, Lisa Rennels, Roger Cooke, Adrian E. Raftery, Hana Ševčíková, & Frank Errickson. (2022). Resources for the Future Socioeconomic Projections (RFF-SPs) (Version 5) [Data set]. Zenodo.

Compute the *Future damage increase* term in I5:I187 by taking negative sums of each row of the P5:KJ187 cells.

4.8 Calculating the marginal mitigation costs and *Future mitigation increase* term in FSI

4.8.1 Emission change, $\dot{E}$

In the "Edot" sheet, simple time differences are taken from the D9:JW49 cells in "E" sheet, to the D9:JW49 cells in the "Edot" sheet.

4.8.2 Intercept of the marginal mitigation cost curve, $a_i(t)$

In "MACa", $a_i(t)$, intercept of the marginal mitigation cost curve, is computed by $a_i(t) = 5.133 + 0.476(\text{per-capita-GDP})_{it}$. See Section 6 below for estimating this value.

Note that there are exceptions to the calculation of $a_i(t)$, due to the lack of data, highlighted by yellow and blue cells. Yellow highlighted cells use actual price data. Blue cells are irregular estimates using closest year data.

Go to the World Development Indicators again and download the following series for all the studied countries for the recent past data in CSV or XLSX format:

- GDP per capita (current US$) from WDI for 2020-2023

These are now pasted in D47:G85 blue cells in the "MACa" sheet.

Regarding future GDP per capita, we assume per capita GDP growth rates to be 1.5% from 2024 to 2100, 1.2% from 2101 to 2150, 1.1% from 2151 to 2200,and 1.0% from 2201-2300, in light of RFF-SPs, as mentioned in Appendix E.

These are applied in G47:JX85 cells in the "MACa" sheet.

4.8.3 Slope of the marginal mitigation cost curve, $b_i(t)$

In "MACb", $b_i(t)$, slope of the marginal mitigation cost curve, is calibrated using $a_i(t) + b_i(t)E(t) = 800 * 0.985t$.

In C3:JW3 cells, this series is computed, starting from USD 800 in C3 in 2020, followed by the next year of D3=C3$*(1 - 0.015)$, and so on.

In column A, country labels are inserted, and in column B, write middle for BRA, CAN, IND, and RUS, and low for CHN.

If the backstop technology cost, $800 * 0.985t$ in C3:JW3 cells, is smaller than $a_i(t) = 5.133 + 0.476 \times (\text{per-capita-GDP})_{it}$ calculated in "MACa", then write zero in C3:JW42 cells. Otherwise, compute $(800 * 0.985t - a_i(t))/E(t)/2$ for middle-income countries, $(800 * 0.985t - a_i(t))/E(t)/3$ for low countries, and $(800 * 0.985t - a_i(t))/E(t)$ for the rest of the countries in C3:JW42 cells (See footnote 23 of the paper). For example, the C5 cell for ARG in 2020 reads:

```
=IF((C$3-MACa!D5)/E!C$5<0, "",IF($B5="middle", (C$3-MACa!D5)/E!
C$5/2, IF($B5="low", (C$3-MACa!D5)/E!C$5/3,(C$3-MACa!D5)/E!C$5)
))
```

4.8.4 Calculating *Future mitigation increase*

In the "Fmi" sheet, note that there are only constant discount rates

```
const_1.5, const_2.0, const_2.5, and const_3.0
```

for Fmi (*Future mitigation increase*), as noted in footnote 25.

In I4:KB42 of the "Fmi" sheet, the marginal mitigation costs for 38 countries for 2021–2300 are inserted, as long as the emission change in that specific year is negative. For example, the I5 cell for Argentina in 2021 reads:

```
=IFERROR(IF(Edot!D10>=0,0,Edot!D10*(MACa!E5+Edot!D10*MACb!D5))
,0)
```

Finally, the NPV of the marginal mitigation costs are calculated in B4:E42, looking at the four discount rates that are inserted in B3:E3. For example, B4 reads:

```
=NPV(B$3,$I4:$KB4)
```

4.9 Summing up the results

In each of the "R1.5", "R2.0", "R2.5", "R3.0", "C1.5", "C2.0", "C2.5", and "C3.0" sheets, let:

- C5:C187 (GS) refer to AG2:AG1065 in "GS" sheet;
- E5:E187 (Cost of current emissions) refer to C3:J185 cells in "CurrDam" sheet;
- G5:G187 (−Future mitigation increase) refer to B4:E42 cells in "Fmi" sheet;
- K5:K187 (Total) sum C, E, G, and I column;

- M5:M187 (GNI) vertically look up GNI (current US$) to the AG column in "GS" sheet.

- I5:I187 (−Future emission increase) is already computed in the above 4.7.

The results in "R2.0" constitute the outputs in **Tables 4**.

To make **Table 5**: Sensitivity of the full sustainability indicator FSI to various discount rates Unit: 2020 in billion USD, use the "FSI" sheet in which you collate GS and FSI values from "GS", "R1.5", "R2.0", "R2.5", "R3.0", "C1.5", "C2.0", "C2.5", and "C3.0" sheets.

Also, the results in "R1.5", "R2.0", "R2.5", "R3.0", "C1.5", "C2.0", "C2.5", and "C3.0" sheets for 183 countries are used in **Tables H1–16**.

In the "LSI" sheet, compute LSI for all the countries. Concretely, in E3:U187, only sum GS + Current damage + Future damage increase, skipping *Future mitigation increase*, from C, E, and I columns in each of the "R1.5", "R2.0", "R2.5", "R3.0", "C1.5", "C2.0", "C2.5", and "C3.0" sheets. For example, E5 reads:

```
=IFERROR(R1.5!$C5+R1.5!$E5+R1.5!$I5,"")
```

Also calculate LSI as shares of GNI in X3:AP187.

4.10 Comparing with WB indicators

To compare with the World Bank data, in S5:Y254 in the "WB" sheet, import country-level recent decade average annual CO_2 emissions from the file already downloaded from the Global Carbon Budget in 4.3. (In our file, Annual CO_2 emissions including land-use change, Annual CO_2 emissions from land-use change, and Annual CO_2 emissions in total, for 2019 are also shown in column V-X for just comparison purposes.)

- In column AE, GS is copied from "R1.5" sheet.

- In column AG, GS is subtracted by \$40/tCO_2 × recent decade average annual CO_2 emissions in billion tCO_2 in column Y.

- In column AI, GS is subtracted by \$194/tCO_2 × recent decade average annual CO_2 emissions in billion tCO_2 in column Y.

- In column AK, FSI is copied from "R2.0" sheet.

- In column AM, GNI is copied from "R1.5" sheet.

Now **Table 6** has been made for the central case of R2.0.

5 Drawing Figures

5.1 Figure 1

Figure 1 is taken from “E” and “Edot” sheets of the `country_SCC_JAERE.xlsx`.

5.2 Figure 2

Create a CSV file `GS_LSI.csv`.

Copy and paste X3:AP187 of “LSI” sheet in the `country_SCC_JAERE.xlsx`.

Start Python and run the Python code in `JAERE_Fig2.py.txt`:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("GS_LSI.csv")
plt.figure(figsize=(10,6))

#plt.scatter(df["GS"], df["ram_3.0"], color='lightblue', label='
    Ramsey_3.0', s=20)
#plt.scatter(df["GS"], df["ram_1.5"], color='green', label='
    Ramsey_1.5', s=20)
#plt.scatter(df["GS"], df["ram_2.5"], color='blue', label='Ramsey_2
    .5', s=20)
plt.scatter(df["GS"], df["ram_2.0"], color='lightgreen', label='
    Ramsey_2.0', s=20)

# 45-degree line
x_min, x_max = plt.xlim()
y_min, y_max = plt.ylim()
# plt.plot(df["GS"], df["GS"], 'k--', alpha=0.7, label='GS = LSI')
plt.plot(df["GS"], df["GS"], linestyle='--', color='gray', label='
    GS = LSI')

# Vertical and horizontal lines
plt.axhline(0, color="black", linewidth=0.8)
plt.axvline(0, color="black", linewidth=0.8)

highlight_countries = ["ABW","AFG","AGO","ALB","ARE","ARG","AUS","
    AUT","AZE","BDI","BEL","BEN","BFA","BGD","BGR","BHR","BHS","BIH
    ","BLR","BLZ","BOL","BRA","BRB","BRN","BTN","BWA","CAF","CAN","
    CHE","CHL","CHN","CIV","CMR","COD","COG","COL","COM","CPV","CRI
    ","CUB","CYP","CZE","DEU","DJI","DNK","DOM","DZA","ECU","EGY","
    ERI","ESP","EST","ETH","FIN","FJI","FRA","GAB","GBR","GEO","GHA
    ","GIN","GMB","GNB","GNQ","GRC","GTM","GUY","HKG","HND","HRV","
    HTI","HUN","IND","IDN","IRL","IRN","IRQ","ISL","ISR","ITA","JAM
    ","JOR","JPN","KAZ","KEN","KGZ","KHM","KOR","KWT","LAO","LBN","
```

```
    LBR","LBY","LCA","LKA","LSO","LTU","LUX","LVA","MAC","MAR","MDA
    ","MDG","MDV","MEX","MKD","MLI","MLT","MMR","MNE","MNG","MOZ","
    MRT","MUS","MWI","MYS","NAM","NCL","NER","NGA","NIC","NLD","NOR
    ","NPL","NZL","OMN","PAK","PAN","PER","PHL","PNG","POL","PRI","
    PRT","PRY","PSE","PYF","QAT","ROU","RUS","RWA","SAU","SDN","SEN
    ","SGP","SLB","SLE","SLV","SOM","SRB","STP","SUR","SVK","SVN","
    SWE","SWZ","SYR","TCD","TGO","THA","TJK","TKM","TLS","TON","TTO
    ","TUN","TUR","TWN","TZA","UGA","UKR","URY","USA","UZB","VCT","
    VEN","VNM","VUT","WSM","YEM","ZAF","ZMB","ZWE"]

for i, row in df.iterrows():
    if row["Country"] in highlight_countries:
        plt.text(row["GS"], row["ram_2.0"], row["Country"],
    fontsize=9, ha='right')

# Quadrant watermarks
plt.text(0.2, 0.2, "Quadrant I", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)
plt.text(-0.1, 0.2, "Quadrant II", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)
plt.text(-0.1, -0.1, "Quadrant III", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)
plt.text(0.2, -0.1, "Quadrant IV", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)

plt.xlim(-0.2, 0.4)

plt.xlabel("GS/GNI")
plt.ylabel("LSI/GNI")

plt.legend()
#plt.grid(True)
plt.show()
```

5.3 Figures 3 and 4

Create a CSV file `wb_compare_rate.csv`.

Copy and paste AC3:AM189 of the WB sheet in the `country_SCC_JAERE.xlsx`

Start Python and run the Python code `JAERE_Fig3.py.txt`:

```
import pandas as pd
import matplotlib.pyplot as plt

# Read and immediately write to CSV
# pd.read_excel("wb_compare_rate.xlsx").to_csv("wb_compare_rate.csv
    ", index=False)

```

```
df = pd.read_csv("wb_compare_rate.csv")
plt.figure(figsize=(10,6))

plt.scatter(df["WB194"], df["LSI"], color='green', s=20)
# plt.plot(df["WB194"], df["WB"], linestyle='--', color='gray',
    label='WB194 = LSI')
highlight_countries = ["ABW","AFG","AGO","ALB","ARE","ARG","AUS","
    AUT","AZE","BDI","BEL","BEN","BFA","BGD","BGR","BHR","BHS","BIH
    ","BLR","BLZ","BOL","BRA","BRB","BRN","BTN","BWA","CAF","CAN","
    CHE","CHL","CHN","CIV","CMR","COD","COG","COL","COM","CPV","CRI
    ","CUB","CYP","CZE","DEU","DJI","DNK","DOM","DZA","ECU","EGY","
    ERI","ESP","EST","ETH","FIN","FJI","FRA","GAB","GBR","GEO","GHA
    ","GIN","GMB","GNB","GNQ","GRC","GTM","GUY","HKG","HND","HRV","
    HTI","HUN","IND","IDN","IRL","IRN","IRQ","ISL","ISR","ITA","JAM
    ","JOR","JPN","KAZ","KEN","KGZ","KHM","KOR","KWT","LAO","LBN","
    LBR","LBY","LCA","LKA","LSO","LTU","LUX","LVA","MAC","MAR","MDA
    ","MDG","MDV","MEX","MKD","MLI","MLT","MMR","MNE","MNG","MOZ","
    MRT","MUS","MWI","MYS","NAM","NCL","NER","NGA","NIC","NLD","NOR
    ","NPL","NZL","OMN","PAK","PAN","PER","PHL","PNG","POL","PRI","
    PRT","PRY","PSE","PYF","QAT","ROU","RUS","RWA","SAU","SDN","SEN
    ","SGP","SLB","SLE","SLV","SOM","SRB","STP","SUR","SVK","SVN","
    SWE","SWZ","SYR","TCD","TGO","THA","TJK","TKM","TLS","TON","TTO
    ","TUN","TUR","TWN","TZA","UGA","UKR","URY","USA","UZB","VCT","
    VEN","VNM","VUT","WSM","YEM","ZAF","ZMB","ZWE"]

for i, row in df.iterrows():
    if row["Country"] in highlight_countries:
        plt.text(row["WB194"], row["LSI"], row["Country"], fontsize
    =8, ha='right')

# 45-degree line
x_min, x_max = plt.xlim()
y_min, y_max = plt.ylim()
plt.plot(df["WB"], df["WB194"], linestyle='--', color='gray', label
    ='WB194 = LSI')

plt.axhline(0, color="black", linewidth=0.8)
plt.axvline(0, color="black", linewidth=0.8)

plt.text(0.2, 0.2, "Quadrant I", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)
plt.text(-0.2, 0.2, "Quadrant II", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)
plt.text(-0.2, -0.1, "Quadrant III", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)
plt.text(0.2, -0.1, "Quadrant IV", fontsize=16, color="blue",
         ha="center", va="center", alpha=0.5, zorder=0)

```

```
plt.xlabel("WB194/GNI")
plt.ylabel("LSI/GNI")
plt.legend()
plt.show()
```

Figure 4 is just an enlarged version of Figure 3.

5.4 Figure G

Figure G (Top panel: Country-level emission path from 2020–2300. Bottom panel: Change in national emissions, 2020–2300. Unit: billion tCO_2) is taken from "E" and "Edot" sheets of the `country_SCC_JAERE.xlsx`.

6 Carbon price estimates for *Future mitigation increase*

In Section 4.8 we have used the parameters of the marginal mitigation costs as given. Here we describe how to replicate them, as explained in Section 6.2 of the paper and Appendix E.

First, go to the World Bank's State and Trends of Carbon Pricing Dashboard to obtain carbon prices (i.e., carbon taxes and tradable permit prices) for 42 countries and regions for 1991—2024. This raw data in `.xlsx` is converted and saved in the `carbon_price.dta` file.

Second, download GDP per capita (current US$) for 1991–2023 from World Development Indicators.

Third, BAU carbon prices in the market are regressed on GDP per capita. The Stata Do file `JAERE_carbon_price.do` starts from saving and cleaning the price data and ends with saving regression. **In the following code, please update the file path on lines 7 and 45 to match the appropriate location on your local machine**:

```
* carbon_price pasted from Excel carbon_price_wb (WB)
rename var1 year

* reshape to the long data
reshape long price_, i(year) j(country) string

save "C:\Users\USERNAME\Documents\carbon_price.dta"

* pcgdp pasted from Excel
rename var1 year
reshape long pcgdp_, i(year) j(country) string
```

```
12
13 sort country year
14
15 merge 1:1 year country using "C:\Users\USERNAME\Documents\
       carbon_price.dta"
16
17 * country names should be converted to country_id
18 encode country, generate(country_id)
19
20 * panel data recognized
21 xtset country_id year
22
23 replace pcgdp=pcgdp_/1000
24
25 * pooled OLS
26 qui reg price_ pcgdp
27 est sto reg1
28
29 qui reg price_ pcgdp year
30 est sto reg2
31
32 * panel data regression
33 qui xtreg price_ pcgdp, fe
34 est sto reg3
35
36 qui xtreg price_ pcgdp year, fe
37 est sto reg4
38
39 *random effects could be checked by
40 *xtreg price_ pcgdp year, re
41
42 * output to tex
43 ssc install estout, replace
44
45 esttab * using "C:\Users\USERNAME\Documents\stataresult.tex.txt",
       se r2 star(* 0.1 ** 0.05 *** 0.01) b(3) replace
```

This result is directly used in **Table E1**: Estimates for the linear coefficient $a_i(t)$ in the marginal mitigation cost. The final estimate dataset is contained in the `carbon_price.dta` file.

This is the end of the replication. Have a good day!