Exploring the Reproducibility for Visualization Figures in Climate Change Report

Lu Ying* Zhejiang University

Yingcai Wu† Zhejiang University

Jean-Daniel Fekete‡ Inria & Université Paris-Saclay

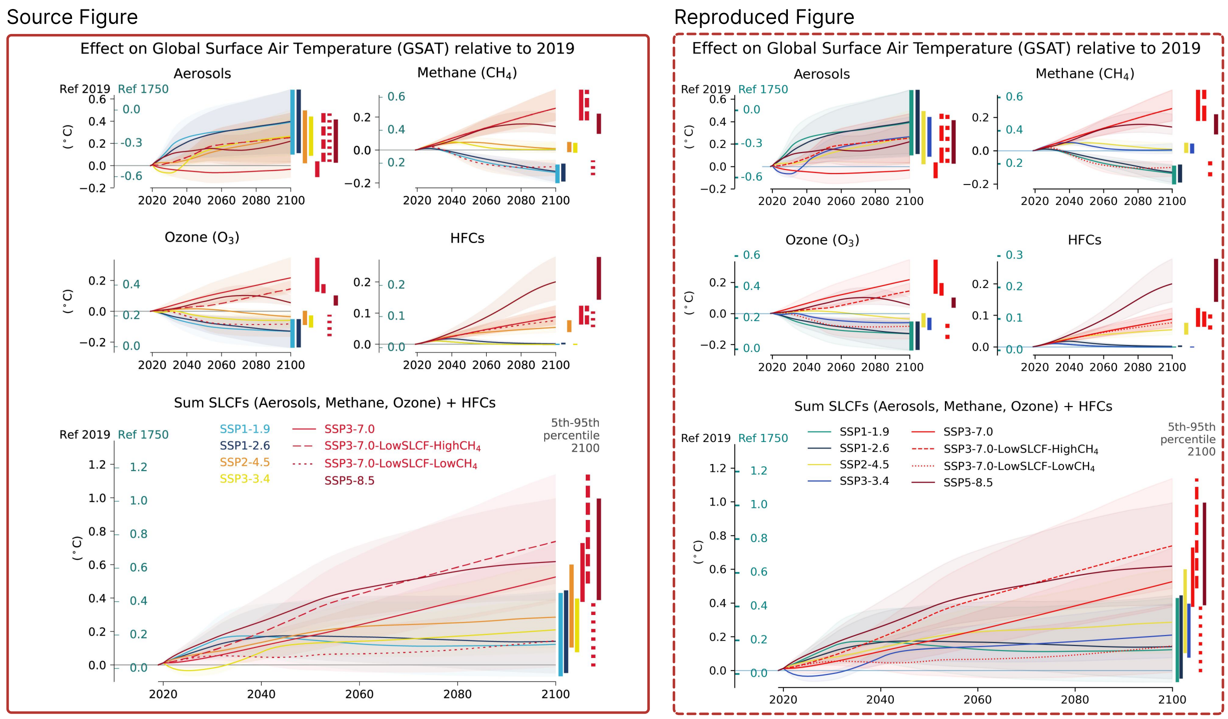

Figure 1: The source figure and reproduced figure for Chapter-6_Fig22.

Abstract

The Intergovernmental Panel on Climate Change (IPCC) plays a pivotal role in assessing and communicating climate science through its comprehensive reports. Despite the IPCC's efforts to provide source code and data for report figures, reproducing these figures is still challenging. This paper details our approach and the obstacles encountered in creating reproducible visualizations from the IPCC Working Group 1 data. Our work involved developing a set of front-end GitHub repositories that build upon the IPCC's original resources, incorporating reproducibility instructions and scripts to closely replicate the report's figures. By providing reproducible figures, we aim to enhance public engagement and contribution to climate change communication, ensuring accuracy and facilitating iterative improvements in figure presentation.

Index Terms: Reproducibility, climate change.

*e-mail: yingluu@zju.edu.cn
†e-mail: ycwu@zju.edu.cn
‡e-mail: jean-daniel.fekete@inria.fr

1 Introduction

The Intergovernmental Panel on Climate Change (IPCC)[1] is the United Nations body responsible for assessing the science related to climate change. The IPCC produces various reports on climate change, such as assessment and working group reports, that are approved and endorsed by most United Nations (UN) countries. The reports contain numerous charts that are often challenging for the public to understand, even for visualization experts.

To improve these charts reliably, it is crucial to have access to the source code and data, as pixel-based images are difficult to alter. For the Sixth Assessment Report (AR6, 2021–2023) Working Group 1 (WG1) [3], IPCC allegedly provides the source code and data for the charts in public repositories.

However, despite the fact that the IPCC mentions it is reproducible, we encountered difficulties reproducing the charts from the original IPCC WG1 report repositories. Therefore, this paper describes our efforts and challenges to create reproducible figures from IPCC data since reproducibility is an important issue for visualization [1]. This work was done in interaction with the Technical Support Unit of IPCC Group 1, which supported us and provided as much information as possible, given that AR6 had been finished since 2021 and all the participants were not easily reachable. On the one hand, the IPCC code and data to generate the figures have been endorsed by the UN countries through IPCC procedures and must

[1]https://www.ipcc.ch/

be used as a reference to claim conformity. On the other hand, generating the figures from IPCC repositories is never straightforward and often impossible. We have been able to create a collection of front-end GitHub repositories at github.com/repro-ipcc/, referencing the original IPCC ones and adding reproducibility instructions and scripts to regenerate the report figures as closely as possible. Figure 1 is one example. With reproducible figures, more people can become involved and contribute to communication on climate change by iteratively refining the figures while ensuring they use the correct data and calculations.

2 Relate Work

Reproducibility is a critical concern in research, particularly when data is involved. Freire et al. [2] emphasize the importance of reproducibility within data-driven experiments across various scientific domains, highlighting its significance within the realm of e-science. In specialized fields like geoscience, Koukouraki and Kray [4] point out that clear documentation and standardized practices are essential to ensure reproducibility in the preparation and presentation of geoscientific data.

In the domain of data visualization, the challenges of achieving reproducibility have been examined by Fekete and Freire [1]. They discuss the methodologies needed to address these challenges, emphasizing the importance of reproducibility in visualizations. Additionally, The Computer Graphics community has created the Graphics Replicability Stamp Initiative (GRSI[2]), offering a formal mechanism to verify the replicability of visualizations to promote better practices in research and publication.

In our situation, we focus on data visualizations in the climate change field. The IPCC reports are foundational documents that guide global understanding and policy-making regarding climate change. Ensuring that the data visualizations within these reports are reproducible is essential for maintaining their credibility and utility in scientific discourse and policy-making.

3 Structure

We will first introduce the structure of the repository. The complete organization of figures in Working Group 1 can be found on GitHub[3]. These repositories are divided into two main tables: one for the official report, including Chapters 1-12, the Chapter Atlas, the Technical Summary, and the Summary for Policymakers (SPM), and another for the Boxes, Cross-Chapter Boxes (CCBOX), Frequently Asked Questions (FAQs), and supplementary materials (SM).

Each part comprises numerous repositories. Due to the figures being generated by different authors, the structure varies across different chapters. While most repositories contain only a single figure, some, such as Chapters 7, 9, 11, and 12, include all figures for the entire chapter. The code used for generating these figures is written in various languages, including Python, R, Matlab, and NCL.

Given the existence of 137 repositories with hundreds of figures, in this paper, we focus on Python—a mature language with a large community—to demonstrate our proposed working pipeline.

4 Reproducible Pipeline

In this section, we detail our pipeline for reproducing the figures. The entire process is divided into four parts: setting up the environment, preparing the data, patching then running the code, and finally giving complete instructions for the following users. All reproductions are performed on a Linux machine with an *anaconda* environment.

[2] replicabilitystamp.org
[3] github.com/IPCC-WG1/

There are some preparations before we start. We first created a new organization in Github named "repro-IPCC"[4]. For each repository in the original IPCC organization, we create a new repository with the prefix "repro-", such as "repro-Chapter-6_Fig22". In this new repository, we create a git submodule that links to the original IPCC repository.

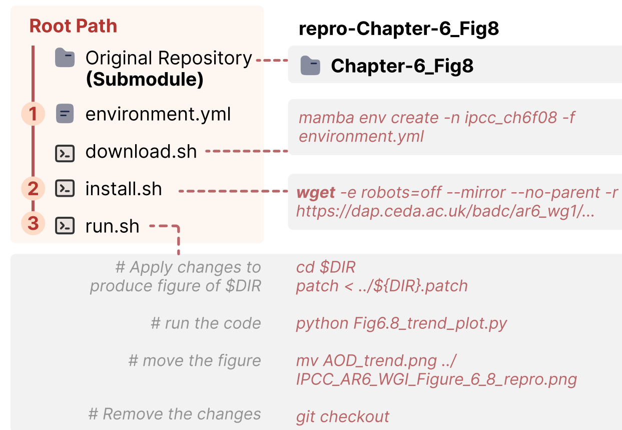

Figure 2: The reproduced file structure after four steps.

4.1 Setting up the Environment

We categorized the environment setup into three scenarios, ranging from optimal to challenging:

Optimal Condition: The author provides a file with a complete list of dependencies, including specific versions for each library. We use the provided file to set up the environment in this scenario.

Moderate Condition: The author provides a file listing all dependencies but without specifying versions. Here, we use Python version 3.7, the version available when the original repository was created, to set up the environment. If issues arise with version 3.7, we attempt nearby versions like 3.6 and 3.8 based on the error messages encountered. We create a setup file to create the anaconda environment named *"environment.yml"* in the root directory.

Challenging Condition: The author does not provide any environment setup instructions. In this case, we examine the Python file used to generate the figure and identify the required libraries. We then create a new setup file, similar to the second scenario, and test the Python environment accordingly.

For this step, we create a shell script named *"install.sh"* for users to do a quick setup.

4.2 Preparing the Data

After setting up the environment, we run the code to check for missing files. Since the IPCC mandates traceability for each number in the charts, all numbers are calculated from the original datasets, which can be quite large. As a result, authors typically do not include these large datasets in the repository. Instead, they provide instructions for locating the data and running the code.

However, this is not always the case. In most cases, the necessary data can be found in the CEDA Archive[5], the official repository for IPCC data. To facilitate the data retrieval process, we create a shell script named *"download.sh"* that uses *wget* to fetch the data from CEDA.

In the worst-case scenario where the data cannot be found, we currently email the author, introduce ourselves, describe the data issues we encountered, and request their assistance. Based on our

[4] github.com/repro-IPCC/
[5] ipcc.ch/report/ar6/wg1/resources/data-access/

experience, the authors have been willing to help and provide the missing data. We directly include the missing data in the root path for the repositories we have successfully reproduced.

4.3 Patching then Running the Code

We then run the code following the instructions in *"readme.md"* in the original repository, if available. If the *"readme.md"* does not contain instructions, we identify which Python file is the main one used to generate the figure. In the best-case scenario, the code runs successfully. If not, we attempt to resolve the issues ourselves. This typically involves correcting file locations and names for the data downloaded from the CEDA Archive. We also encounter problems related to different library versions. Once we resolve these issues, we generate the figure, usually in PDF and PNG format. Since we focus on Python repositories, the figures are typically created using the Matplotlib library. We can easily modify the saving function to output the figure in SVG format.

To maintain the integrity of the original repository, we create a patch file instead of altering and committing to the original submodule directly. This patch file, named according to the original repository (e.g., *"Chapter-2_Fig31.patch"*), is saved in the root path.

Since a significant amount of work is involved in rerunning the code, we create another shell script named *"run.sh"* to assist future users in rerunning the process more easily. For the shell script, we first navigate to the submodule folder and execute the main Python file. If the main file is a Jupyter notebook, we first convert the notebook to a Python script to enable automated execution rather than an interactive experience. Next, we apply the patch file to incorporate previous changes to ensure the code runs successfully. After that, we rename and move the figures in various formats from their respective locations to the root path. We follow the naming convention from the IPCC website, appending the postfix "_repro". Finally, we remove our changes to the original repository, using *"git checkout"*.

4.4 Instructions for Future Users

Since we invested significant effort in exploring the reproduction process, we aim to make it easier for future users to reproduce the figures without facing the same challenges. Therefore, we have written a detailed *"readme.md"* to guide users through the reproduction process.

In this *"readme.md"*, we provide specific commands for users to install Miniforge, initialize submodules, set up the conda environment, and ultimately reproduce the figure. The file structure after reproducing is shown in Figure 2.

5 Label the Reproducibility

Building on Peng's work [5], we assessed reproducibility in the subsequent phase using five key dimensions: Input, Execution, Output, Quality, and Post-process. These dimensions are primarily intended to help us track the advancements in our work.

- **Input:** Input completeness is assessed based on three sub-dimensions: *Data*, *Dependence*, and *Code*. Each sub-dimension can be classified as *Complete*, *Partial*, or *Missing* based on the reproduced results.
- **Running**: This dimension evaluates whether the repository can be executed. It is classified as *Running* or *Not Running*. If *Not Running*, the following three dimensions are omitted.
- **Output**: This dimension measures the completeness of the output. We calculate the proportion of full, partial, and missing components out of the total number of components. Components can include subfigures, legends, and other elements, depending on how the authors create the figure. For instance, if a figure has three subfigures (a, b, c) and only one is complete, it is classified as *Full 1/3* and *Missing 2/3* (Figure 3). If there are no full components, this dimension is omitted for clarity.
- **Quality:** This dimension evaluates the quality of the fully reproduced components from the previous dimension. It uses three categories. *Same* means the reproduced subfigure is exactly the same as the one in the report. *Similar* means the reproduced subfigure needs only minor adjustments, such as adding a box or changing the color, to match the one in the report. *Different* means the reproduced subfigure is significantly different from the original, requiring substantial modifications to match the report.
- **Post-process:** This dimension indicates the effort required to generate a figure that is exactly the same as the one in the report. It uses three labels. *Auto* means the figure can be reproduced with code-level modifications. *Manual* means the figure requires manual adjustments to match the original, such as combining two sub-figures into one figure. *Self-contained* means the figure is already identical to the original without any modifications.

Chapter 2 Figure 03

Input	Running	Output	Quality	Post-process
Code Complete Dependence Complete Data Complete	Running	1/3 Full 0/3 Partial 2/3 Missing	0/1 Same 1/1 Similar 0/1 Different	Auto Manual

Chapter 5 Figure 04

Input	Running	Output	Quality	Post-process
Code Complete Dependence Complete Data Complete	Running	1/1 Full	1/1 Same	Self contained

Figure 3: The labels in the readme.md files.

6 Findings and Discussions

In this section, we report our findings and discussions throughout the process.

6.1 Reproduced Charts Need Post-processing

We observed several issues with reproduced figures.

First, the colors are almost always slightly different between the generated figure and the IPCC-provided PNG figure. Despite using similar color schemes, they are not identical. This may be due to variations in printing or compiling methods.

Second, many generated charts are actually components of larger figures rather than complete ones. This includes sub-figures (e.g., (a), (b), (c)) in most conditions and also includes some other conditions where charts and legends are generated separately and then combined. Example can be seen from Figure 4(a).

Third, there are minor style differences compared to the original figures, such as the presence of outer boxes around legends, variations in color usage, and the inclusion or exclusion of legends or axis labels. These repositories likely contain versions close to, but not exactly, the final printed versions.

6.2 Difficulty in Ensuring Identical Reproduction

It is challenging to confirm that the reproduced charts are identical to those in the original report, and it is also difficult to pinpoint the differences. Currently, we rely on visual comparison, similar to playing a "spot the differences" game.

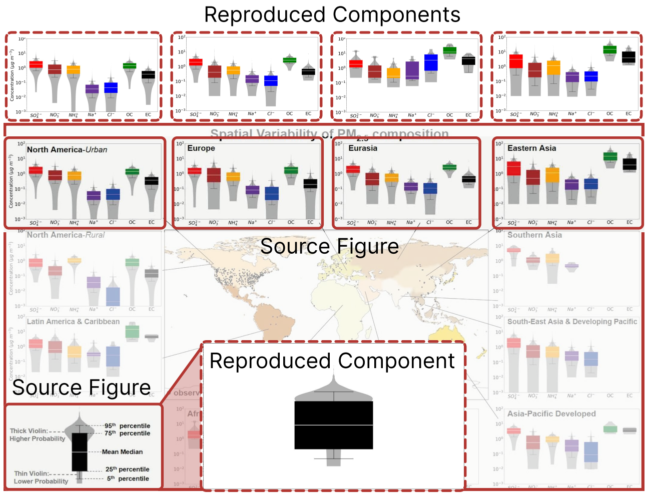

Figure 4: Chapter-6_Fig7 and some reproduced components (enclosed by a dotted box).

We have attempted some automated methods. Initially, we calculate correlation scores between two images, yielding results between 0 and 1, with 1 indicating identical images. However, the scores often do not align with visual differences. High-scoring images sometimes exhibit more differences than lower-scoring ones. Then we tried some pre-processing techniques, such as adding blur, but these methods did not improve the correlation much. We also utilized Large Language Models (LLMs) like GPT-4 to identify differences. While these models provided useful information for reference, their results were inconsistent. The model sometimes changed its answers for the same figure pairs and identified nonexistent differences. Therefore, automated methods currently do not surpass manual inspection in accurately identifying differences.

6.3 Enhancing Figures for Broader Impact

Reproducing IPCC charts offers significant benefits for both the visualization community and the public.

For researchers, having access to source figures is crucial for detailed analysis and design refinement, as original charts may not always be optimized for every application. Since climate change impacts everyone worldwide, it's important that these figures are made accessible and understandable to a wider audience, including students and non-specialists. Using reproducible figures, researchers can enhance these visualizations to communicate vital information more effectively to diverse groups. This could involve improving the design from a visualization perspective or creating alternative formats such as data comics [6] and data videos [7].

Improved figures are crucial for the public, especially young people who will face the long-term impacts of climate change and need to be informed. Enhanced visualizations can serve as powerful educational tools, helping students and young individuals grasp complex climate concepts more easily. This can promote environmental awareness, encourage proactive behaviors, and inspire a new generation of informed climate advocates. Making these charts more accessible and engaging for the general audience empowers it to understand better and address present and future climate challenges.

6.4 Limitations and Future work

While our study has made strides in exploring the reproducibility of figures in IPCC reports, several limitations exist that open avenues for future work. First, our current approach is limited to figures generated using Python. However, many figures in the IPCC reports are produced using other programming languages (e.g., R, NCL) and tools (e.g., Matlab). Expanding our methodology to support a broader range of languages will be crucial for enhancing the generalizability and applicability of our work. Second, the dimensions we used to record the reproduction process are currently tailored to our specific needs and are manually labeled. This approach, while functional, is labor-intensive and prone to inaccuracies. Future work should focus on automating the labeling process and refining the dimensions to make them more universally applicable, thereby improving both efficiency and accuracy. Third, the sheer volume of figures across multiple IPCC reports presents a significant challenge. Our capacity to handle the reproduction process for hundreds of figures is limited. Thus, we document our methods and results comprehensively in this paper, encouraging broader community involvement. By engaging more researchers and practitioners, we hope to scale our efforts, making a larger number of figures reproducible. Furthermore, our approach could be extended beyond IPCC reports to other domains, such as WHO reports.

Acknowledgments

Thanks to Clotilde Péan and Lina Sitz from the IPCC Working Group 1 Technical Support Unit for their support.

References

[1] J. Fekete and J. Freire. Exploring reproducibility in visualization. *IEEE Computer Graphics and Applications*, 40(5):108–119, 2020. doi: 10.1109/MCG.2020.3006412 1, 2

[2] J. Freire, N. Fuhr, and A. Rauber. Reproducibility of Data-Oriented experiments in e-Science (dagstuhl seminar 16041). *Dagstuhl Reports*, 6(1):108–159, 2016. doi: 10.4230/DAGREP.6.1.108 2

[3] IPCC. *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, vol. In Press. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021. doi: 10.1017/9781009157896 1

[4] E. Koukouraki and C. Kray. Map Reproducibility in Geoscientific Publications: An Exploratory Study. In *International Conference on Geographic Information Science, GIScience*, vol. 277 of *LIPIcs*, pp. 6:1–6:16, 2023. doi: 10.4230/LIPICS.GISCIENCE.2023.6 2

[5] R. D. Peng. Reproducible Research in Computational Science. *Science (New York, N.y.)*, 334(6060), 2011. doi: 10.1126/science.1213847 3

[6] Z. Wang, H. Romat, F. Chevalier, N. H. Riche, D. Murray-Rust, and B. Bach. Interactive data comics. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):944–954, 2022. doi: 10.1109/TVCG.2021.3114849 4

[7] L. Ying, Y. Wang, H. Li, S. Dou, H. Zhang, X. Jiang, H. Qu, and Y. Wu. Reviving static charts into live charts. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–16, 2024. doi: 10.1109/TVCG.2024.3397004 4