



# HARD TO CACHE

# Research Poster

A VISUALIZED SUMMARY OF OUR RESEARCH

## ADDRESSED PROBLEM

Despite advances in AI/HPC, >70% of computational studies fail replication due to undocumented dependencies, environment drift, and fragmented workflows. This erodes scientific trust and impedes progress.

Our research asks:

- How can we systematically quantify reproducibility risks in research artifacts?
- Can an automated framework bridge the gap between data collection, analysis, and public dissemination?

HardToCache contributes:

- A novel Scorecard methodology to audit reproducibility across 8 critical dimensions
- Integrated automation tools for scraping, scoring, and visualization

## GOALS

- A Quantifiable Reproducibility Framework
- Seamless Automation Pipeline
- Accessible Science Dissemination
- Open Collaboration Infrastructure

## RESOURCE NEEDS/LIST

Resources used:

- Google Colab - Python
- SGX3 Project Server - HTML/SSH
- Github
- Canva
- pandas, fitz, pathlib, matplotlib
- Google Sheets

## METHODOLOGY

1. Collect
  - Automated scraping of research papers
  - Extract code/data from publications
2. Score
  - Apply 6-key Scorecard
  - Validate with expert feedback
3. Share
  - Generate website + poster
  - Publish tools/scores on GitHub

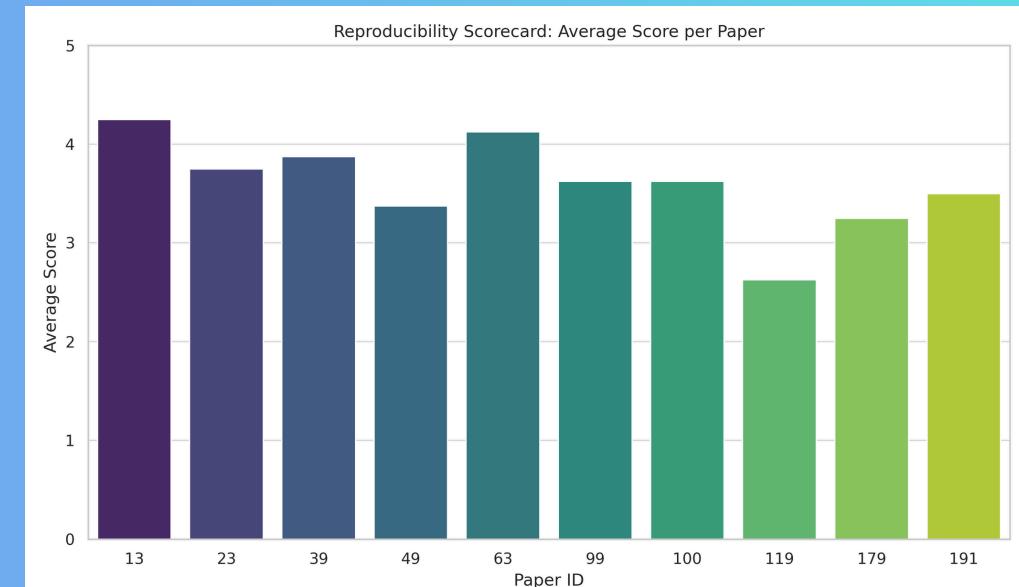
## EXAMPLE PAPER SCORING

- Only a few papers had fully available code and datasets that were able to be reproduced on any given environment. Most lacked full environment or hardware details
- This suggests that while some research is highly reproducible, many papers will still miss key details
- Each category was scored from 0 (not mentioned/accessible) to 5 (very clearly addressed) using keywords in the paper.

## POSSIBLE EXPANSIONS

- Scale Across Disciplines
  - Biomedical research, Climate modeling, Social science
- Real-Time Monitoring
  - Develop a browser extension that shows the reproducibility score and live ratings.
- Enhanced Automation:
  - Auto-update scores when papers are revised

## SCORECARD EXPLANATION



Canva Tip: Double-tap on the sample charts to change the data

## AUTHORS



Charli Brooks  
Elizabeth City  
State University  
cnbrooks04@gmail.com



Silas Erving  
Morehouse College  
silaserving815@gmail.com



Chante Ray  
Mississippi Valley  
State University  
chanteray3@gmail.com



Seth Austin Mack  
Morehouse  
College  
sethamack89@gmail.co  
m

# Team Members



**Charli Brooks**



**Silas Erving**



**Chante Ray**



**Seth Austin Mack**

# Technology Used

CO



```
Memory usage: 1%           IPv4 address for ens4: 10.0.0.2
Swap usage: 0%

10 updates can be applied immediately.
3 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Fri Jun 27 05:29:24 2025 from 84.198.244.196
cnbrooks04@jp-codinginstitute:~$ cd ~/SGX3-Project
source my-virtual-env/bin/activate
(my-virtual-env) cnbrooks04@jp-codinginstitute:~/SGX3-Project$ cs pdf_scrap
cs: command not found
((my-virtual-env) cnbrooks04@jp-codinginstitute:~/SGX3-Project$ cd pdf_scrap
((my-virtual-env) cnbrooks04@jp-codinginstitute:~/SGX3-Project/pdf_scrap$ cd ../Hard-2-Cache
((my-virtual-env) cnbrooks04@jp-codinginstitute:~/SGX3-Project/Hard-2-Cache$ python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a product
ion WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8056
 * Running on http://10.0.0.2:8056
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 654-440-886
84.198.244.196 - - [27/Jun/2025 06:27:58] "GET / HTTP/1.1" 200 -
84.198.244.196 - - [27/Jun/2025 06:27:58] "GET /static/css/style.css HTTP/1.1" 304 -
```

## Top 10 Python Libraries

 <b>Pandas</b> Data analysis and manipulation	 <b>NumPy</b> Mathematical functions
 <b>Matplotlib</b> Data visualisations	 <b>Seaborn</b> Data visualisations
 <b>Tensorflow</b> Machine Learning	 <b>Keras</b> Deep Learning
 <b>SciPy</b> Scientific computing	 <b>PyTorch</b> Machine Learning
 <b>Scrapy</b> Web crawling	 <b>SQLModel</b> Interact with SQL databases



# Scoring of Papers' Scorecard Explanation

## Scorecard Table

Paper ID	Paper	Code	Data	Comp.	GPU Req	Docs	Setup	Results	Avg Score	Notes
119	5	4	1	2	1	3	2	3	2.63	Missing datasets/hardware
49	5	4	3	3	3	3	2	4	3.38	High-quality artifacts, no container
39	5	5	4	4	2	4	3	4	3.88	Strongest overall, container missing
100	5	4	4	4	4	3	2	3	3.63	Code missing, but detailed results
191	5	4	4	3	4	4	2	2	3.50	No container, Joel parsing effort
179	5	1	5	4	1	4	3	3	3.25	Great data, weak automation
23	5	5	5	3	1	4	3	4	3.75	Needs Docker to improve
99	5	5	5	3	0	4	3	4	3.63	Mostly reproducible
13	5	5	5	4	4	4	3	4	4.25	Very strong, Docker would perfect it
63	5	5	5	4	4	4	2	4	4.13	Minor env tweaks needed

## Scoring Criteria

Each paper was scored on the following 8 metrics. All metrics are evaluated on a 0–5 scale, defined as follows:

Metric	0	1	2	3	4	5
Paper Availability	Not available	Paywalled	Partially open	Open, lacks DOI	Open w/ DOI	Fully open (e.g. arXiv/Zenodo)
Code & Software Availability	None	Link broken	Incomplete or undocumented	Complete but limited	Good docs, installable	Versioned, reproducible, containerized
Dataset Availability	None	Proprietary	Partial	Public but unclear	Licensed and usable	Fully open, well-described
Computer Requirements	Not mentioned	Vague	Basic specs	Moderate specs + OS	OS + RAM + tool versions	Full specs + env (e.g. Conda, Docker)
GPU Requirements	Not mentioned	Assumed only	GPU required, unspecified	GPU mentioned	GPU + RAM specs provided	All specs + setup scripts provided
Documentation Quality	None	Minimal README	Incomplete or unclear	Functional but sparse	Clear and complete	Rich docs + tutorials or examples
Ease of Setup	Not reproducible	Needs manual editing	Works with effort	Mostly works	Setup script available	One-click setup (Docker, VM)
Result Reproducibility	No results	Unverifiable	Partially match	Mostly match	Matches with effort	Fully reproduced

# Methodology

## Collect

- Automated scraping of research papers
- Extract code/data from publications

1

## Score

- Apply 8-key Scorecard:
- Validate with feedback

2

## Share

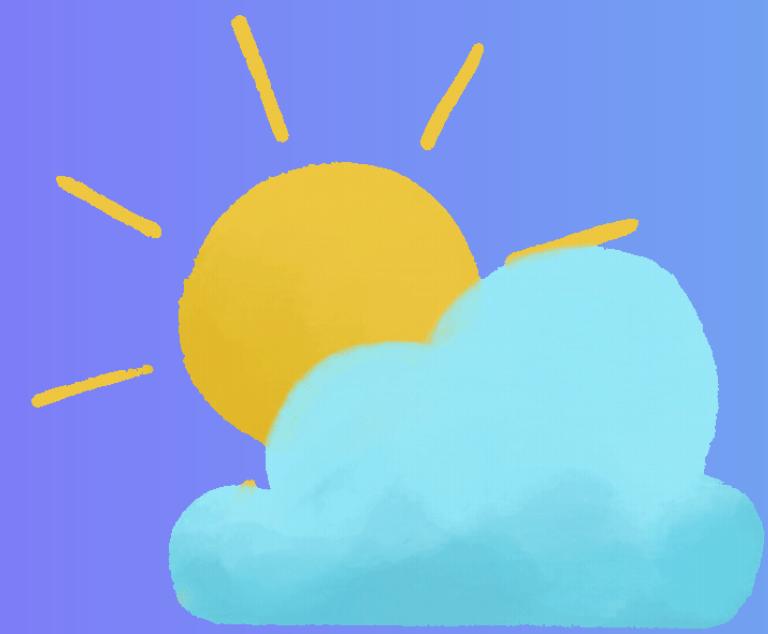
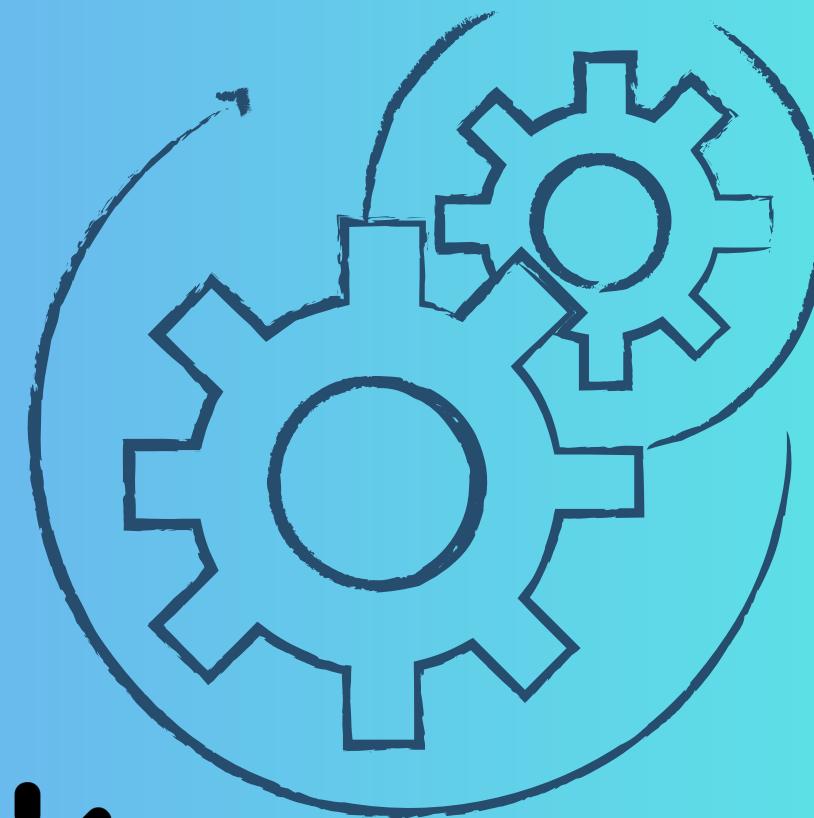
- Auto-generate website + poster
- Publish tools/scores on GitHub

3

# Possible Expansions

Building on our foundation, HardToCache could:

- Scale Across Disciplines
  - Adapt scoring framework for:
    - Biomedical research, Climate modeling, Social science
- Real-Time Monitoring
  - Develop a browser extension that shows:
    - !(Reproducibility Score) Live ratings for arXiv/PubMed papers
- Enhanced Automation:
  - Integrate with:
    - CI/CD platforms → Auto-update scores when papers revise
    - Container registries → One-click environment rebuilds



THANK  
YOU FOR  
LISTENING