# CitationMap: A Python Tool to Identify and Visualize Your Google Scholar Citations Around the World

Chen Liu[1]

[1]Yale University

August 07, 2024

**Abstract**

CitationMap is an open-sourced Python-based tool designed to identify and visualize the geographical distribution of Google Scholar citations. By retrieving citation data and mapping them onto the world map, CitationMap fresearchers an intuitive way to understand the global impact of their work. We first present the motivation, features, and implementation details, highlighting its unique capabilities compared to existing tools. Next, we demonstrate its performance as it processes profiles of six researchers with citation counts ranging from $10^0$ to $10^5$. Finally, we discuss its limitations and propose directions for further improvement.

# CitationMap: A Python Tool to Identify and Visualize Your Google Scholar Citations Around the World

**Chen Liu**
chen.liu.cl2482@yale.edu

## Abstract

`CitationMap` is an open-sourced Python-based tool designed to identify and visualize the geographical distribution of Google Scholar citations. By retrieving citation data and mapping them onto the world map, `CitationMap` offers researchers an intuitive way to understand the **global impact of their work**. We first present the motivation, features, and implementation details, highlighting its unique capabilities compared to existing tools. Next, we demonstrate its performance as it processes profiles of six researchers with citation counts ranging from $10^0$ to $10^5$. Finally, we discuss its limitations and propose directions for further improvement. The code is available at `https://github.com/ChenLiu-1996/CitationMap`.

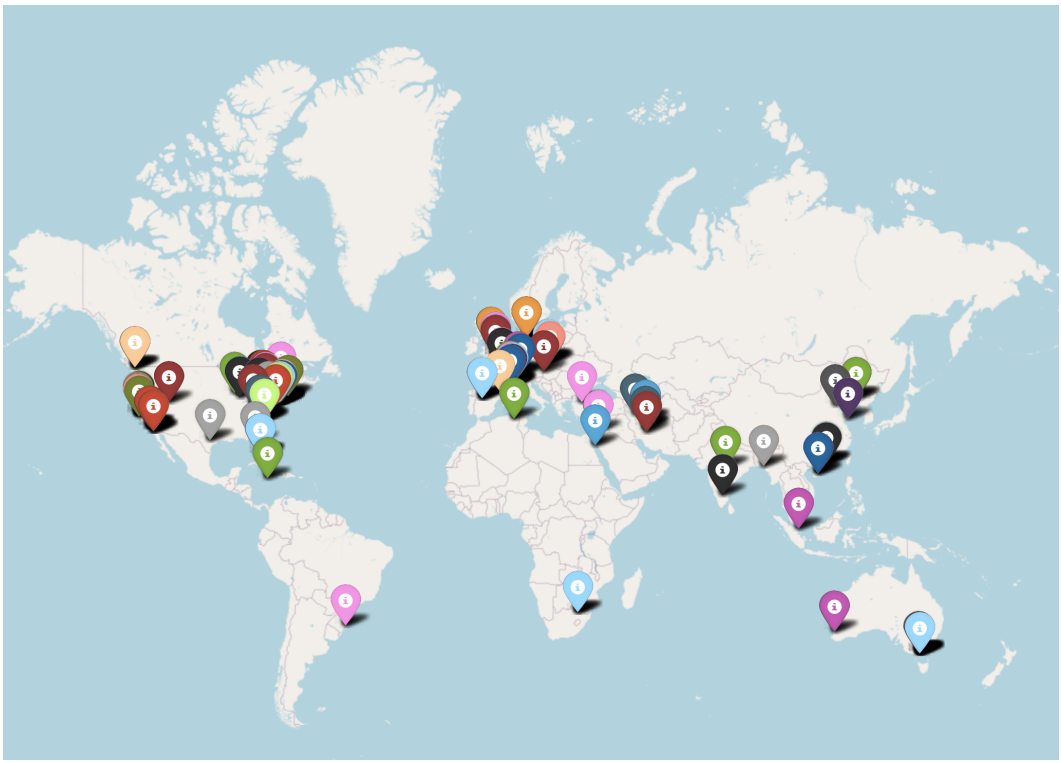

Figure 1: A sample Google Scholar Citation World Map generated by **CitationMap**.

Preprint. Under review.

## 1 Introduction

In the ever-expanding world of academic research, the impact and reach of scholarly works are often measured by citations. Citations not only reflect the influence of the research works, but also play a critical role in the academic reputation of researchers and institutions. Furthermore, the geographical distribution of the citations provides valuable information on the global impact of a researcher, or the patterns of collaboration among multiple researchers or institutions.

Among academic databases, Google Scholar [1] typically indexes the highest number of citations compared to other platforms such as ResearchGate [2] and Semantic Scholar [3]. However, despite its extensive coverage of citations, it lacks a method to visualize citations geographically. This shortfall is also common across other major academic databases. Citation analysis efforts have predominantly focused on constructing and analyzing citation networks or graphs (e.g., `www.connectedpapers.com`), with limited attempts to provide an interactive and intuitive means of visualizing citation origins worldwide. To our knowledge, Web of Science [4] is one of the few databases offering a geographic citation map, but it is not freely accessible and its citation coverage is significantly less comprehensive compared to Google Scholar. This gap underscores the need for a free and open-source tool that can retrieve Google Scholar citation data and present them in a visually engaging and informative way.

To address this need, we introduce `CitationMap`, a Python-based tool designed to identify and visualize Google Scholar citations around the world. `CitationMap` leverages the power of Python's data handling and visualization libraries to create an interactive map that showcases the geographical distribution of citations for any given person with a Google Scholar account. By integrating automated data retrieval with advanced mapping techniques, `CitationMap` offers a comprehensive solution for researchers, allowing them to gain deeper insight into their international impact.

In this paper, we describe `CitationMap`, providing a detailed overview of its implementation and capabilities (Section 2). We also analyze its performance scaling by running it on six researchers with citation counts over 5 degrees of magnitude (Section 3). Lastly, we discuss the limitations (Section 4) and propose directions for future improvement (Section 5). With `CitationMap`, we aim to empower researchers to better identify, analyze, and showcase the geographical reach of their academic contributions.

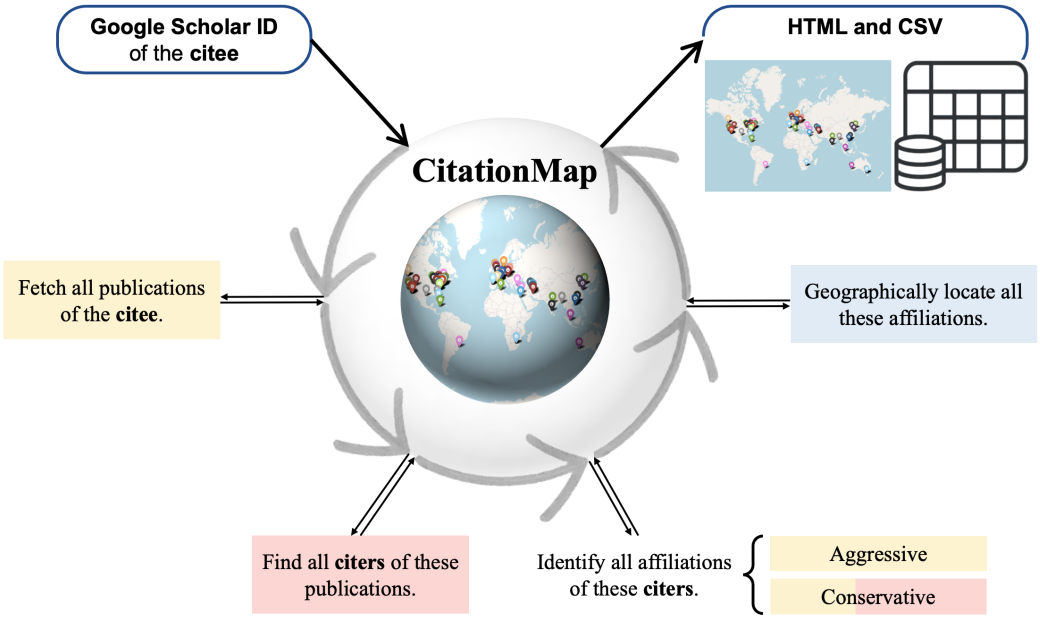


Figure 2: Workflow of **CitationMap**. Given a Google Scholar ID as an input, it returns an HTML file showing the citation world map and a CSV file recording the citation information. Intermediate steps are shown and the colors indicate the third-party services used. Yellow : Google Scholar parsing using scholarly. Red : Web scraping using BeautifulSoup. Blue : Geopositioning using geopy.

## 2 CitationMap

`CitationMap` is an open-source, freely-available Python tool to generate an HTML citation world map as well as rich information on the citation data from any given Google Scholar ID. In this section, we will describe the technical details of `CitationMap`.

### 2.1 Overview

As illustrated in the high-level workflow (Figure 2), `CitationMap` requires minimal effort from users. Users only need to find their own Google Scholar IDs, install the `CitationMap` package (`pip install citation-map`), and run a few lines of code with the template provided. During the process, `CitationMap` runs through five sequential steps and returns the outputs. We will describe the implementation details of these five steps in the following subsections (Section 2.2 to 2.6).

To facilitate understanding, we will use Yoshua Bengio, one of the pioneers in deep learning, as an example to illustrate the process. However, I would not recommend running `CitationMap` on him or any person with that many citations, because (1) it will take a long time, (2) it will likely get you blocked by Google Scholar after so many queries, and (3) it will not give you accurate results since one of our dependencies, `scholarly`, will only consider the first 2,000 citations for each publication.

### 2.2 Fetching publications

The input to `CitationMap` is the Google Scholar ID of a person, presumably the user. Let us call this person the "citee".

In the first step, we take the Google Scholar ID of the citee and find the list of all publications of the citee. The process is illustrated in Figure 3. Practically, this step is empowered by `scholarly` [5], a module that retrieves author and publication information from Google Scholar in a Pythonic way.

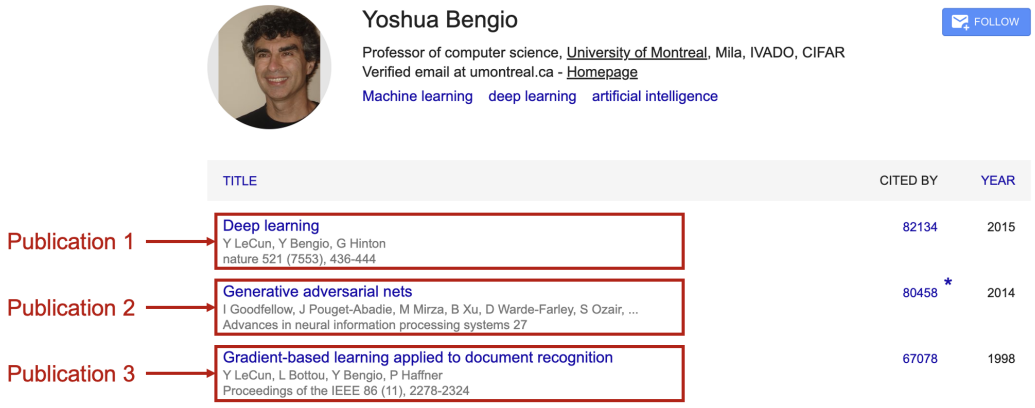


Figure 3: **CitationMap** Step 1. Fetching publications.

### 2.3 Finding citing authors

In the second step, we find all the authors who cite at least one of the publications of the citee, whom we call the "citers". The process is illustrated in Figure 4. Practically, this step is performed by web scraping with the help of the `BeautifulSoup` package. Essentially, we search on Google Scholar for each publication of the citee[1] and browse the information on the webpage for the citing papers. By web scraping, we can find the Google Scholar IDs for all authors in each citing paper.

In addition, we also keep track of the titles of the citing papers (authored by the citers) and the cited papers (authored by the citee) during the process, to facilitate the creation of the summary CSV file.

[1]Here we use the URL `https://scholar.google.com/scholar?hl=en&cites=` followed by the paper ID of the citee's publication.

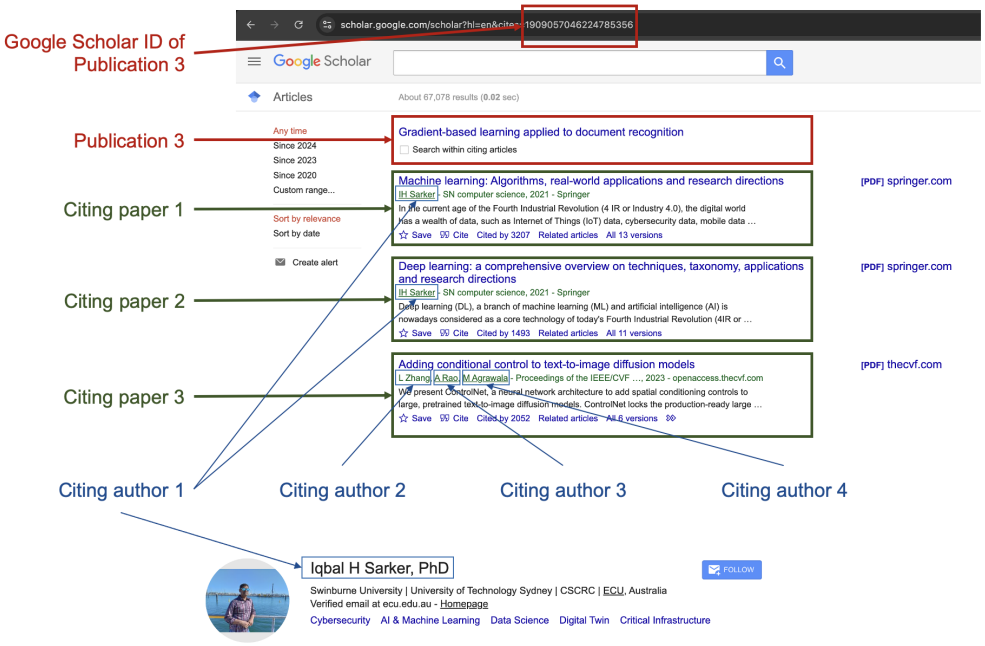


Figure 4: **CitationMap** Step 2. Finding citing authors.

### 2.4 Identifying citing affiliations

In the third step, we identify the affiliations of the citers in order to locate them. We primarily leverage the information in the "affiliation tab" of the citer's Google Scholar profile, as illustrated in Figure 5. One particular challenge is that the string in the affiliation tab is manually entered by the author and there is no particular constraint on the content, such as limiting the possible entries with a drop-down menu. As a result, we need to read and parse the affiliations from a diverse set of self-reported entries.

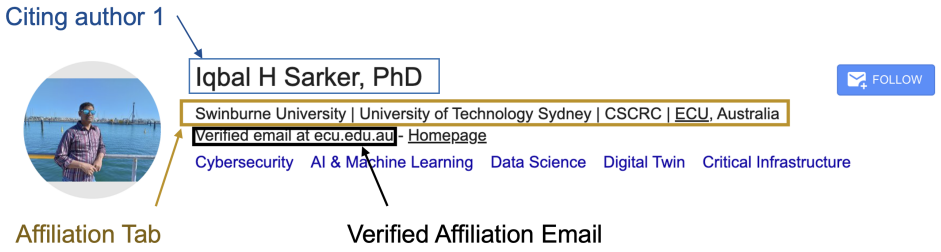


Figure 5: **CitationMap** Step 3. Identifying citing affiliations.

To that end, we provide two approaches, one more **aggressive** and one more **conservative**.

For the **aggressive** approach, we read the string from the citer's affiliation tab and parse the string to find **all** relevant affiliations. In the example shown in Figure 5, there would be four different affiliations, namely Swinburne University, University of Technology Sydney, CSCRC, and ECU, Australia. In the current implementation, the parsing logic is very simple: it involves regular expressions that (1) break strings into substrings by certain delimiters such as commas or semicolons, (2) filter substrings that are unrelated to an affiliation (such as "Professor" from "Professor, Stanford University"), and (3) gather all these likely affiliations. The aggressive approach is implemented by webscraping with `BeautifulSoup`.

For the **conservative** approach, we only consider the verified affiliation (the "organization" field in `scholarly`). This will give **at most one** relevant affiliation for each citer. It is possible for a citer to have no affiliation if one of the following happens: (1) the citer did not enter any affiliation into

the affiliation tab, (2) the citer's self-reported affiliation is not identified by Google Scholar (e.g., the company Meta), or (3) the citer did not verify the affiliation using an email address under that affiliation's domain. In the example shown in Figure 5, the only affiliation would be ECU, Australia. The conservative approach is implemented by web scraping with `BeautifulSoup` and identifying the verified affiliation with `scholarly`.

### 2.5 Locating citing affiliations

In the fourth step, we geographically locate the affiliations of all citers. This is implemented using the `geopy` service, where at each time we send a string representing an affiliation to `geopy` and listen to its response. If the communication is successful and the place is found in the database, we can obtain the *longitude*, *latitude*, *county*, *city*, *state*, and *country* of this affiliation.

### 2.6 Rendering and summarizing

In the final step, we return the results in two formats.

For the HTML file, we use `folium` [6] to render the citation world map (see Figure 1 for a demonstration of the visual result). Each affiliation is indicated by a location pin on the world map. The map is interactive, such that the user can drag and zoom in or out to better visualize a particular region. If a location pin is clicked by the cursor, the name of the corresponding affiliation as well as all affiliated citers will be displayed.

For the CSV file, we record a comprehensive table of information on each citation of the citee. For each citation, we store the name of the citer, citing paper title, cited paper title, citer affiliation, as well as the location (latitude, longitude, county, city, state, and country) of the affiliation.

## 3 Performance Analyses

We designed a series of experiments to demonstrate the performance of `CitationMap`. Empirical results were obtained using the profiles of six researchers with citation counts across 5 orders of magnitude ($10^0$ to $10^5$). The statistics of the researchers are summarized in Table 1. Papers without citation are quickly discarded early in the process and should minimally affect performance. All experiments were performed on a MacBook Pro with an Apple M2 Pro chip and 16 GB of RAM.

Table 1: Statistics of the citees used for analyses, with slight noise added to ensure anonymity.

| Index | Name | Number of Cited Papers | Number of Citations |
|---|---|---|---|
| 1 | Anonymous | 1 | 9 |
| 2 | Anonymous | 11 | 101 |
| 3 | Anonymous | 19 | 218 |
| 4 | Anonymous | 22 | 789 |
| 5 | Anonymous | 38 | 1,720 |
| 6 | Anonymous | 149 | 10,028 |

### 3.1 Runtime scaling

We first investigated the runtime scaling of `CitationMap`. For each user, we were able to run the tool in 16 minutes using the aggressive approach or in 8 minutes using the conservative approach to identify affiliations (see Section 2.4 if you are unfamiliar with these two approaches).

As illustrated in Figure 6, it appears that the runtime scales really well, as it shows a sublinear growth trend with respect to either number of cited papers or number of citations. Since most prospective users do not have more than 150 cited publications or more than 10,000 citations, we do not see the immediate necessity in further scaling up the x-axes.

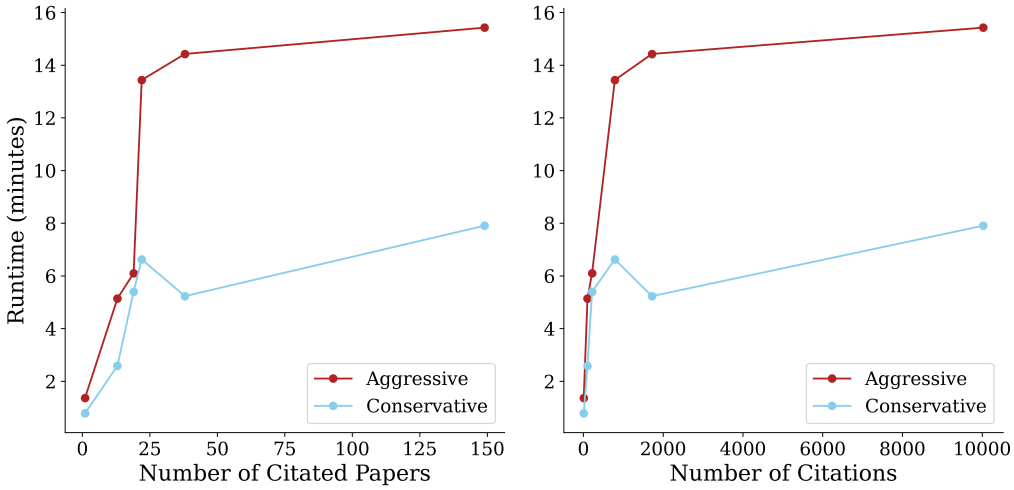


Figure 6: Runtime of **CitationMap**.

### 3.2 Affiliation identification accuracy

The second critical aspect is the accuracy of affiliation identification, which cannot be directly evaluated without extensive labeling. As a surrogate, we measured the ratio of the number of affiliations successfully located in the fourth step (Section 2.5) to the number of affiliations identified in the third step (Section 2.4). Similarly, we separately visualized the aggressive approach and the conservative approach (see Section 2.4 if you are unfamiliar with these two approaches).

As illustrated in Figure 7, adopting the aggressive approach generally results in identifying two to five times more affiliations compared to adopting the conservative approach. On the other hand, only around 50% of these identified affiliations can be located by `geopy` if we take the aggressive approach, compared to 90% for the conservative approach.

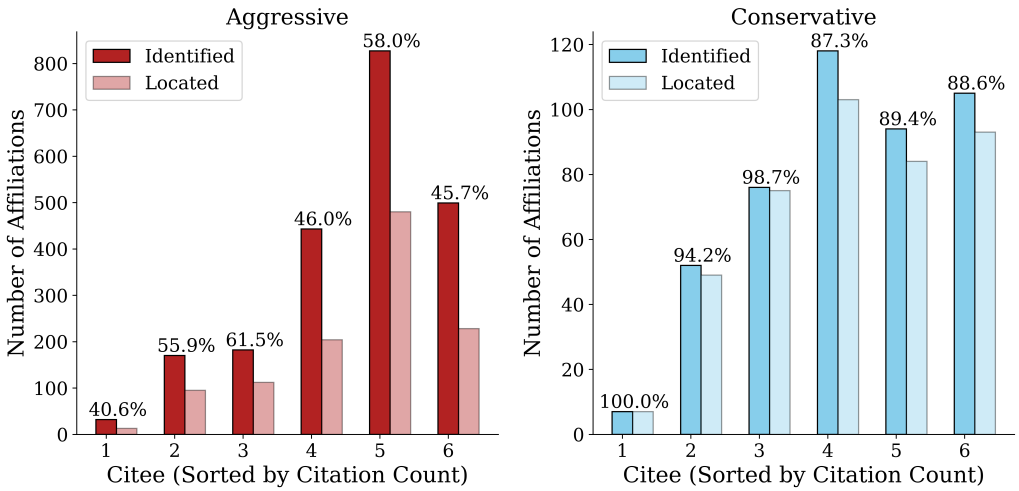


Figure 7: Geolocating success rate of **CitationMap**.

These observations indicate a trade-off between precision and recall for affiliation identification.

- The **aggressive** approach yields **lower** precision and **higher** recall.
- The **conservative** approach yields **higher** precision and **lower** recall.

### 3.3 Effects of affiliation identification approaches

Here, we qualitatively visualize the effects of whether we adopt the aggressive or conservative approach when we identify affiliations. For this purpose, we use a side-by-side comparison of citee 2.

In the left panel we display the citation world map using the aggressive approach (Figure 8) whereas in the right panel we display the counterpart using the conservative approach (Figure 9). It can be seen that the ratio of number of location pins is approximately two to one, which is consistent with the statistics in Figure 7.

It should be noted that **it is unclear which approach would give us the most accurate results of affiliation identification**, as there are many factors that can lead to underestimation and overestimation, as discussed in Section 4.3. Just as mentioned in the previous section, there is a trade-off between precision and recall. Additional quantitative analyses can provide valuable insights, but that would require a lot of labeling. Due to the prohibitively high work load, we would postpone more dedicated analyses to future investigations.

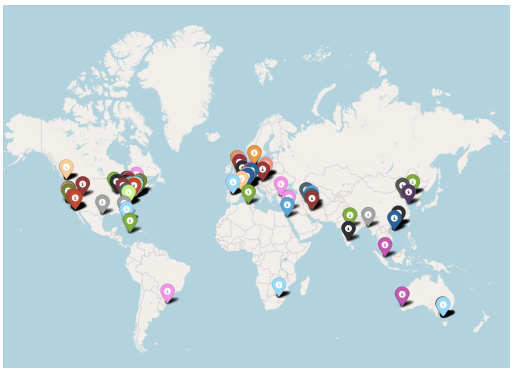

Figure 8: Aggressive affiliation identification.

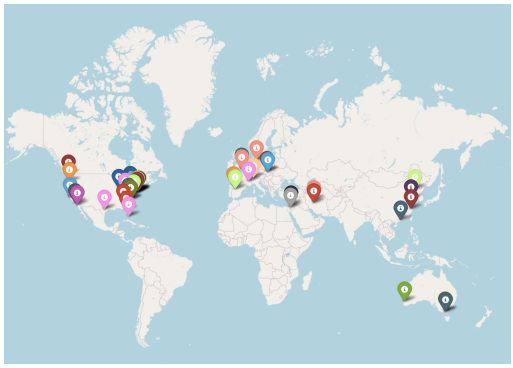

Figure 9: Conservative affiliation identification.

## 4 Limitations

Despite its effectiveness, `CitationMap` has several limitations.

### 4.1 Exclusively Google Scholar based

`CitationMap` is not using information outside Google Scholar. As a result, you are expected to have underestimations due to reasons such as the following.

1. Your Google Scholar profile is not up-to-date.
2. Some papers citing you are not indexed by Google Scholar.
3. Some authors citing you do not have Google Scholar profiles.
4. Some authors citing you do not report their affiliations.

### 4.2 Web scraping limitations

Web scraping is performed, and CAPTCHA or robot check can often get us, especially if we crawl frequently. This is more often seen in highly cited users.

Based on our experience, this is usually not a critical issue. Unless you are blocked by Google Scholar, at worst you will end up with missing several citers, which is not likely a huge deal for highly cited users regardless.

### 4.3 Affiliation identification and geolocating issues

This is a joint effect between affiliation identification and geolocating. The number of citing affiliations will be:

1. Underestimated if some affiliations are not found by `geopy.geocoders`.
2. Underestimated if we experience communication errors with `geopy.geocoders`.
3. (Aggressive approach only) Overestimated if non-affiliation phrases are incorrectly identified as locations by `geopy.geocoders`.
4. (Conservative approach only) Underestimated if the citer did not verify with an email address under a matching affiliation domain.
5. (Conservative approach only) Underestimated because all non-primary (with verified email address) affiliations are ignored.

## 5 Future Work

There are several aspects in which `CitationMap` can be further improved.

1. Handling CAPTCHA and robot checks.
   NOTE: Currently, we try our best not to impose too much extra work or burden on the users to ensure a good user experience, and therefore we are not considering paid services such as ScraperAPI.
2. Improving affiliation identification, possibly by named entity recognition.
   NOTE: Currently, we try our best not to impose too much extra work or burden on the users to ensure a good user experience, and therefore we are not considering paid services such as GPT API.
3. Considering citers without Google Scholar profiles. This would expand the coverage of citers by a significant margin, though it might be technically challenging.
4. Incorporating databases beyond Google Scholar, including but not limited to Semantic Scholar [3], ResearchGate [2], Web of Science [4], etc.

## 6 Acknowledgements

The authors thank **Zhijian Liu** (Research Scientist at NVIDIA, Incoming Assistant Professor at UCSD) for his inspiring suggestions and partial implementation that helped us introduce the conservative approach for affiliation identification.

The authors express their appreciation to **Yue Zhao** (Assistant Professor of Computer Science, University of Southern California) for promoting `CitationMap` on social media, which significantly enhanced its visibility.

## Frequently Asked Questions

1. **Q:** How do I use this tool?
   **A:** There is a comprehensive "User Guide" section on the GitHub repository (`https://github.com/ChenLiu-1996/CitationMap`). You can follow through the steps there.
2. **Q:** For each citing paper, do we consider the affiliations of only the first author or all authors?
   **A:** We count all authors. The purpose of `CitationMap` is to visualize the global impact of a researcher, and hence we would take all citing authors into account.
3. **Q:** What will happen if I have multiple versions of the same paper?
   **A:** As long as these versions are reflected on your Google Scholar profile, we will consider all of them. For example, some papers have multiple versions with either the same or different titles [7–11], and they should all be counted as long as you include them in your Google Scholar profile, whether or not they are merged into a single publication entry. In rare cases, a paper might have been assigned multiple Google Scholar IDs (e.g., the paper "Deep Learning" [12] published on *Nature*), but still, all versions will be considered.
4. **Q:** Where can I find help if I encounter errors?
   **A:** I recommend reviewing the "Debug" section on the GitHub repository (`https://github.com/ChenLiu-1996/CitationMap`). If your problem is not mentioned there, or if it cannot be fixed by those suggestions, you are welcome to submit a GitHub Issue.

## Popularity of CitationMap

`CitationMap` is the first GitHub project that received significant attention (see Figure 10) by the authors. It attracted around 300 stars in the first week. We thank everyone who used it, promoted it, or provided feedback. We are happy to iterate and improve this tool for the community.

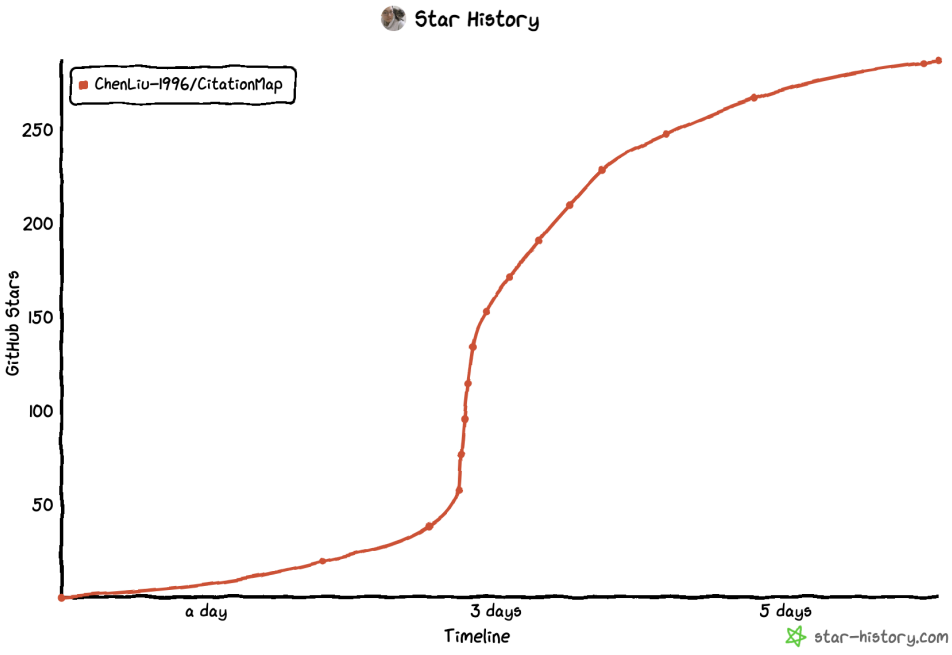


Figure 10: GitHub star history of **CitationMap**.

## References


[1] Alireza Noruzi. Google scholar: The new generation of citation indexes. 2005.

[2] David Nicholas, David Clark, and Eti Herman. Researchgate: reputation uncovered. *Learned Publishing*, 29(3):173–182, 2016.

[3] Suzanne Fricke. Semantic scholar. *Journal of the Medical Library Association: JMLA*, 106(1):145, 2018.

[4] Philippe Mongeon and Adèle Paul-Hus. The journal coverage of web of science and scopus: a comparative analysis. *Scientometrics*, 106:213–228, 2016.

[5] Steven A. Cholewiak, Panos Ipeirotis, Victor Silva, and Arun Kannawadi. SCHOLARLY: Simple access to Google Scholar authors and citation using Python, 2021.

[6] Martin Journois, Rob Story, James Gardiner, Halfdan Rump, Andrew Bird, Antonio Lima, Joshua Cano, Juliana Leonel, Tim Sampson, Jason Baker, et al. python-visualization/folium: v0. 11.0. *Zenodo*, 2022.

[7] Chen Liu, Matthew Amodio, Liangbo L Shen, Feng Gao, Arman Avesta, Sanjay Aneja, Jay C Wang, Lucian V Del Priore, and Smita Krishnaswamy. Cuts: A deep learning and topological framework for multigranular unsupervised medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2024.

[8] Danqi Liao, Chen Liu, Benjamin W Christensen, Alexander Tong, Guillaume Huguet, Guy Wolf, Maximilian Nickel, Ian Adelstein, and Smita Krishnaswamy. Assessing neural network representations during training using noise-resilient diffusion spectral entropy. In *2024 58th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2024.

[9] Chen Liu, Nanyan Zhu, Haoran Sun, Junhao Zhang, Xinyang Feng, Sabrina Gjerswold-Selleck, Dipika Sikka, Xuemin Zhu, Xueqing Liu, Tal Nuriel, et al. Deep learning of mri contrast enhancement for mapping cerebral blood volume from single-modal non-contrast scans of aging and alzheimer's disease brains. *Frontiers in Aging Neuroscience*, 14:923673, 2022.

[10] Nanyan Zhu, Chen Liu, Xinyang Feng, Dipika Sikka, Sabrina Gjerswold-Selleck, Scott A Small, and Jia Guo. Deep learning identifies neuroimaging signatures of alzheimer's disease using structural and synthesized functional mri data. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 216–220. IEEE, 2021.

[11] Chen Liu, Ke Xu, Liangbo L Shen, Guillaume Huguet, Zilong Wang, Alexander Tong, Danilo Bzdok, Jay Stewart, Jay C Wang, Lucian V Del Priore, et al. Imageflownet: Forecasting multiscale trajectories of disease progression with irregularly-sampled longitudinal medical images. *arXiv preprint arXiv:2406.14794*, 2024.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.