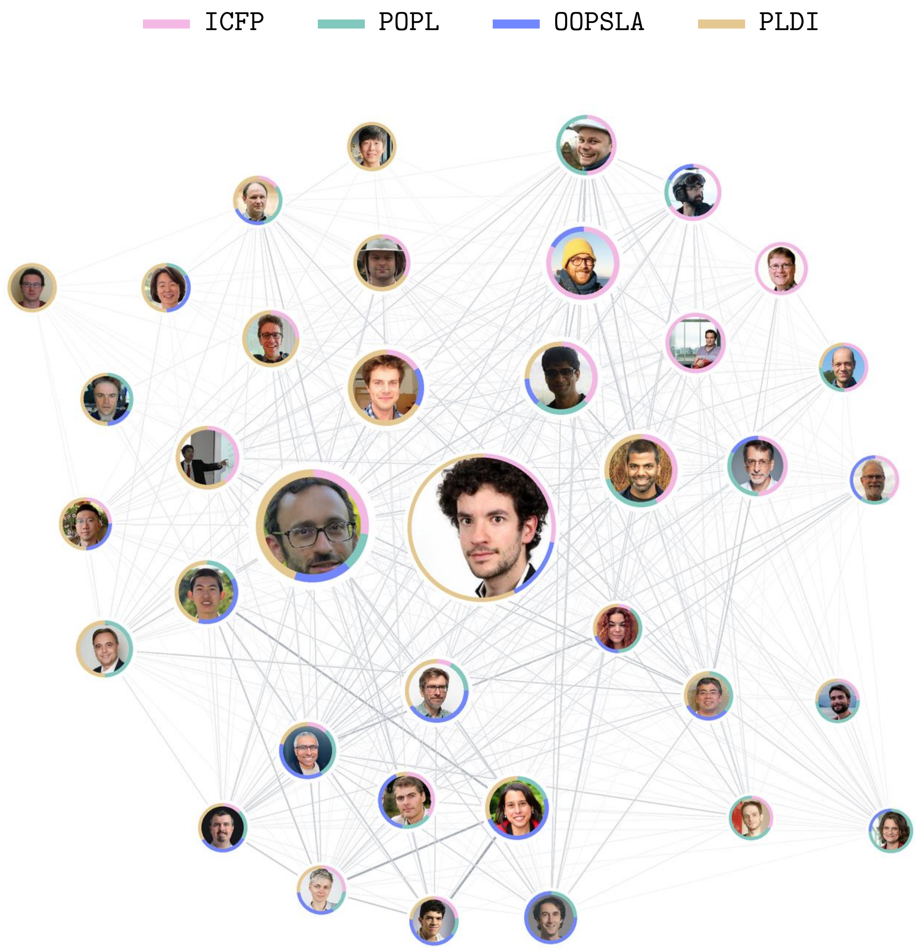
Are Program Committee Members Cited Differently ?

Citation Patterns around PC Service in Programming Languages Conferences



Optional Master Research Project Report, Spring 2026

Author
Ender Sari
ender.sari@epfl.ch

Supervisor
Clément Pit-Claudel
clement.pit-claudel@epfl.ch

EPFL

Systems and Formalisms Laboratory

Detailed Outline

1. Introduction and Research Questions

- Motivates the main question: are PC members cited differently?
- RQ1 asks whether accepted papers cite PC members' earlier work more often.
- RQ2 asks whether citations shift toward the serving conference in PC years.

2. Empirical Design

- Explains why PC members cannot be compared directly with researchers not on the PC.
- Introduces the counterfactual, the panel design, and the main confounding concerns.

3. Data Construction and Validation

- Describes the three data construction steps: PC records, references, and the panel data.
- Explains the validation checks used before building the panel data.

4. Descriptive Event Study Analysis

- Answers RQ1 by comparing citations around first observed PC service.
- Presents selected examples of PC members.

5. Citation Source Analysis

- Answers RQ2 by comparing conference citation sources around PC service.
- Presents selected examples of PC members.

6. Discussion

- Interprets RQ1 and RQ2 results as descriptive, not causal.
- Summarizes the main design, data, and generalizability limitations.
- Describes the future work.

7. Timeline and Reflection

- Compares the planned timeline with what actually happened.
- Explains what took longer than expected and why.

References

Appendices

- Appendix A: PC source and history checks.
- Appendix B: Accepted paper and OpenAlex citation checks.
- Appendix C: Final panel construction and author matching checks.
- Appendix D: Analysis methods, detailed figures, and selected examples.

1 Introduction and Research Questions

This project studies whether public program committee (PC) visibility is associated with citation behavior. The motivating idea is simple: when a research paper PC list is publicly available before or around submission, authors may observe which researchers are listed on it. If that visibility matters, accepted papers in that conference year may cite listed PC members' earlier work differently.

Therefore, the main research question is:

Are PC members cited differently?

I study this question in programming languages conferences. The analysis covers ICFP, POPL, OOPSLA, and PLDI from 2017 to 2025. Rather than measuring a researcher's total citation count, I measure citations from the same conference: citations from accepted papers in conference c, year y, to earlier work by researcher r.

I use two empirical versions of the research question:

RQ1. Citation level: Do accepted papers in that conference year cite PC members' earlier work more often?

RQ2. Citation source concentration: Do PC members' citations shift toward the conference where they serve?

Because PC service is not randomly assigned, I interpret both analyses descriptively rather than causally. The next section explains the resulting selection problem and motivates the empirical design.

2 Empirical Design

A simple comparison between PC members and researchers not on the PC is not enough. The same factors that make researchers more likely to appear on a PC may also make their earlier work more likely to be cited.

Figure 1 summarizes this confounding problem with a plausible directed acyclic graph (DAG).

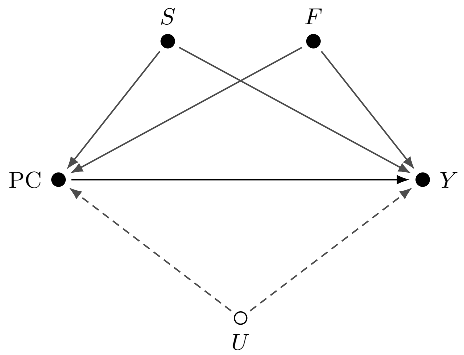


Figure 1: Why comparing PC members with researchers not on the PC is confounded.

In the DAG, PC is being listed on a conference's research paper program committee, and Y is citations from that same conference to the researcher's earlier work. S denotes seniority, such as career age, publication history, or h-index; F denotes fit to the conference, such as whether the researcher's past work matches the conference's topics that year; and U denotes additional, unobserved confounders, such as informal professional connections to organizers or network relationships not captured in the data. Each of S, F, and U can affect both being listed on the PC and citations from the same conference, so a simple comparison is confounded.[1]

Given this confounding problem, estimating a causal effect of PC listing would require a counterfactual comparison: for the same researcher, in the same time period and conference context, how would citations from that same conference to their earlier work have differed if they had not been publicly listed on the PC page? The ideal would be to observe the same researcher in the same conference year in both states, publicly listed and not listed, and compare the two outcomes. That difference would be the causal effect.[2] I do not observe this setup in the data. Therefore, I construct a panel based on researcher, conference, and year and use event study analyses.

The panel design improves on a simple comparison between PC members and other researchers by comparing the same researcher within the same conference across years. For each researcher and conference pair, I compare citations before, during, and after the PC service year. This reduces stable differences between researchers, such as long run seniority, reputation, and broad research area, while keeping the citing conference fixed.

However, the panel does not remove all confounding. It reduces stable differences between researchers, but factors that change over time can still remain, such as recent citation momentum, a highly visible paper, or temporary topic fit with the conference. These factors can affect both PC selection and citations. So, I interpret the event study analyses descriptively.

3 Data Construction and Validation

To create the panel, I follow the pipeline in Figure 2. I first collect PC data from PC lists[3], recording who served on each research paper PC in each conference year. I then use OpenAlex[4] as the main bibliographic source for paper, reference, and author data, queried through the Python client `pyalex`.[5] Finally, I combine these sources into a panel by researcher, conference, and year. The panel links PC status to citations from each conference year.

[1] This is one plausible structure. Other variables and arrows are possible, but the key point here is that S, F, and U are common causes of both PC and Y.

[2] This causal interpretation also requires a timing assumption: PC list visibility must come before the relevant citation decisions. In my data, I only observe final accepted papers, not submission and revision bibliographies, so I cannot tell whether a reference was already present before review or added later.

[3] See, e.g., the PC list for POPL 2021: https://popl21.sigplan.org/committee/POPL-2021-research-papers-program-committee.

[4] https://openalex.org/.

[5] https://pypi.org/project/pyalex/.

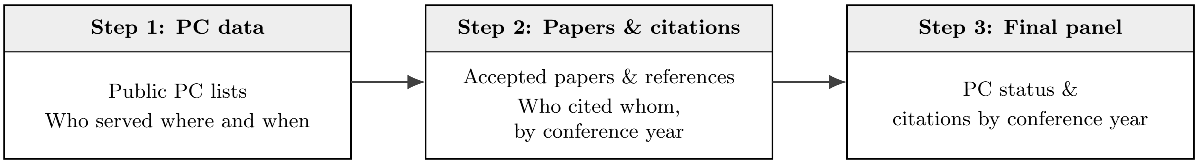


Figure 2: From PC records and citations to the panel data.

The final panel contains one observation for each researcher, conference, and year. Each observation records whether the researcher served on the PC in that conference year and how many citations the researcher's prior work received from papers in that same conference year. The rest of this section summarizes the main data decisions and observations.

Step 1: PC Service Data

Throughout this report, *PC member* means a researcher listed on the PC/review page for research papers in a conference year.[6]

Many researchers appear more than once in the data, which suggests that PC service is concentrated among a recurring group of researchers rather than being evenly distributed across researchers.[7] This matters for the study because repeated PC service may reflect seniority, visibility, and prior involvement in the community, rather than only a single conference year role. These repeated service patterns are shown in Figure 3; service across conferences patterns are shown in Appendix A, Figure 13.

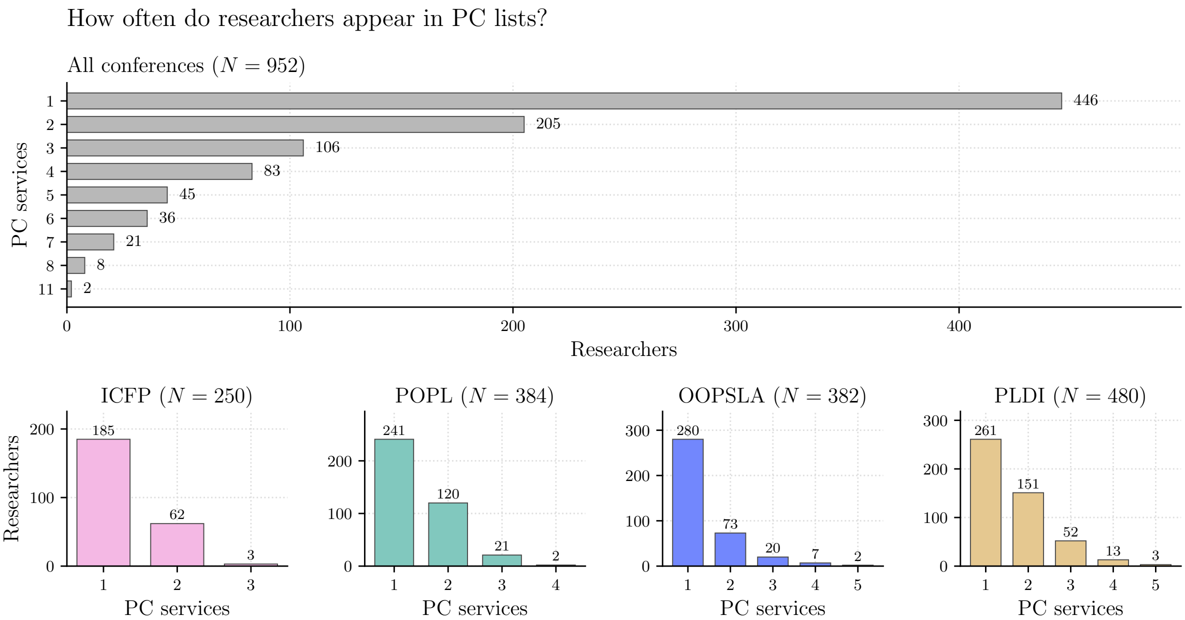


Figure 3: How often researchers appear in the PC data.

[6]Therefore, I do not include other committee groups, such as student research competitions, workshops, tutorials, or artifact evaluation.

[7]After scraping the PC pages, the PC data contain 2,180 records from ICFP, POPL, OOPSLA, and PLDI between 2017 and 2025. There are 952 unique researchers.

To validate the PC data, I check it against HotCRP reviewer pages,[8] EPC/ERC pages (External Program Committee / External Review Committee)[9]. These checks help me assess whether the PC records are comparable across conferences and years.[10]

Figure 4 shows the PC counts separately for the main PC pages, EPC/ERC pages when available, and HotCRP reviewer lists. My main observation is that the source structure is not the same across conferences and years. POPL and ICFP are the most comparable cases, since their PC pages and external reviewer pages are close to the available HotCRP reviewer lists. OOPSLA becomes harder to compare after 2021 because OOPSLA 2022 introduced two separate rounds of reviewing.[11] PLDI also changes around 2021: before 2021, the PC and the external reviewer group were listed separately; from 2021 onward, both appeared on the PC page. As a result, PC page visibility may not represent the same type of service across the two PLDI periods.

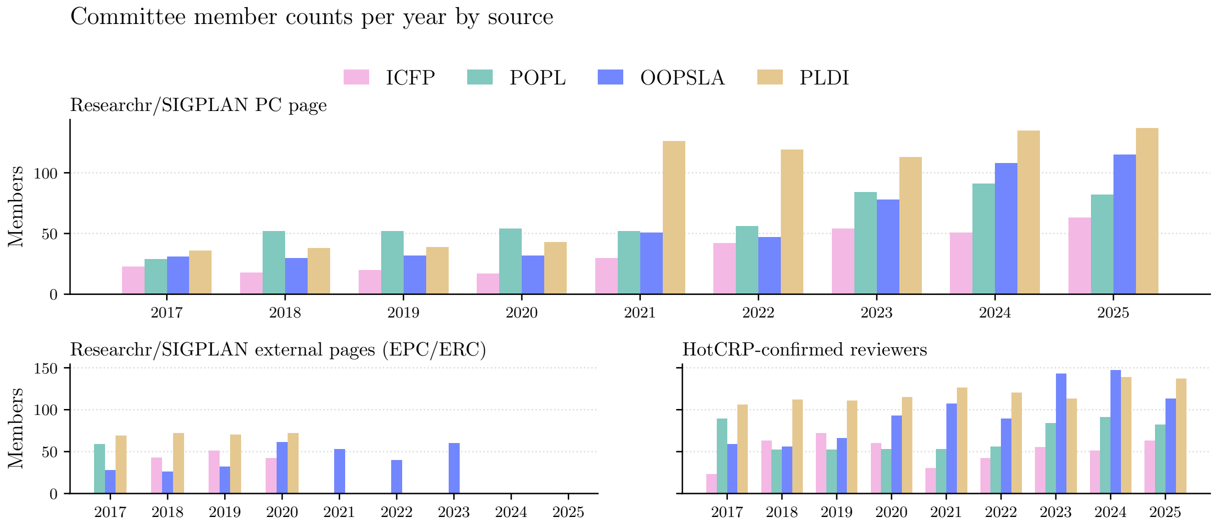


Figure 4: PC member counts by source across conferences and years.

Step 2: Paper and Citation Coverage

I build the accepted paper data for each conference year using PACMPL records and DOI matching between DBLP and OpenAlex for PLDI years that are not covered by PACMPL.[12]

After filtering to ICFP, POPL, OOPSLA, and PLDI between 2017 and 2025, the combined paper data contain 2,528 unique accepted papers across 39 conference year combinations. Figure 5 shows the number of accepted papers by conference year.[13]

[8] See, e.g., the HotCRP PC list for PLDI 2024: https://pldi2024.hotcrp.com/users/pc.

[9] See, e.g., the ERC list for PLDI 2019: https://pldi19.sigplan.org/committee/pldi-2019-papers-external-review-committee.

[10] I use them to qualify the sample rather than pooling all conference years mechanically. Appendix A gives the detailed PC checks and observations.

[11] https://2022.splashcon.org/track/splash-2022-oopsla#Call-for-Papers

[12] https://blog.sigplan.org/2022/08/08/pldi-will-join-pacmpl/

[13] I check the OpenAlex/DBLP paper list against ACM front matter PDFs. This check helps verify that the filtered paper data are not missing included accepted papers.

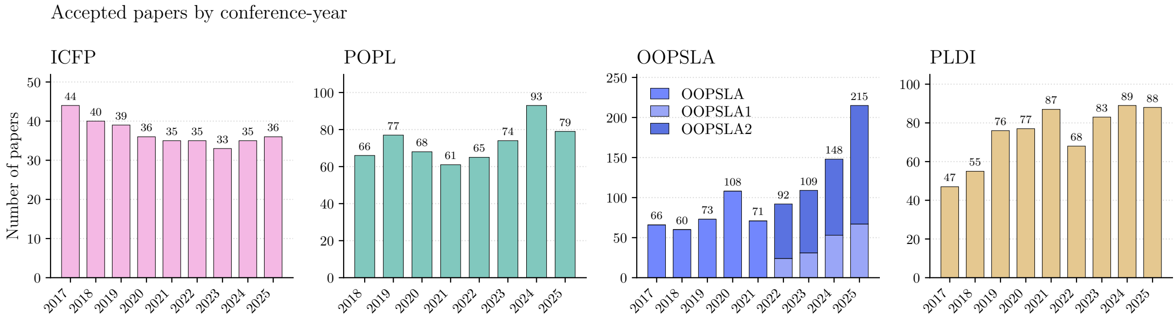


Figure 5: Accepted paper counts by conference year.

I use OpenAlex reference lists to connect each accepted paper to cited works and cited authors. This gives the citation graph needed for the panel: who cited whom, and from which conference year.[14]

For the citation outcome, the OpenAlex reference lists need to be representative enough to support citation counting.[15] Figure 6 shows the reference count distribution overall and by conference year.

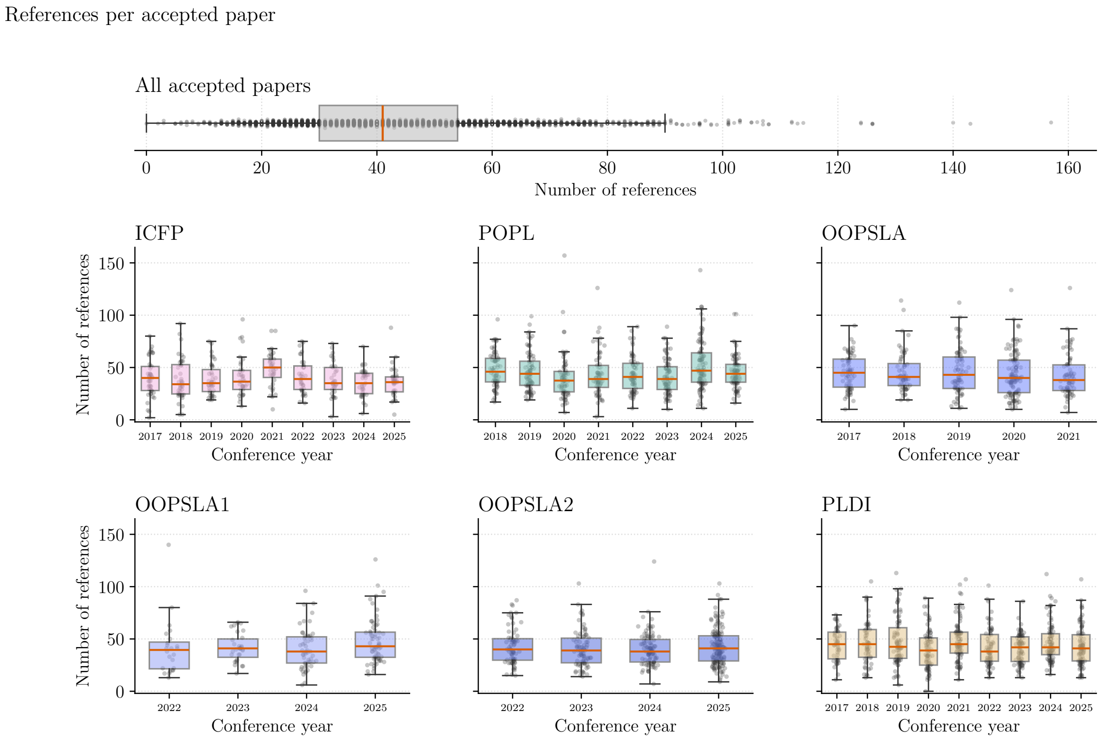


Figure 6: OpenAlex reference counts per accepted paper.

Based on these checks, I think OpenAlex is suitable for this study, while still subject to the metadata limitations reported in Appendix B and discussed in Section 6.

[14] OpenAlex metadata changes over time; the data and plots in this report use OpenAlex citation metadata cached on June 11, 2026.

[15] In the filtered paper data, papers have an average of 43.6 OpenAlex references and a median of 41 references. The 25th and 75th percentiles are 30 and 54 references per paper, which suggests that the reference coverage is generally dense enough for constructing citation outcomes by conference year.

Step 3: Author Matching and Final Panel

The final step is linking names on PC pages to OpenAlex author records.[16] After additional name and manual checks, citation outcomes can be constructed for 942 researchers (98.9%).[17]

Table 1 shows the structure schematically. The same researcher can appear in multiple rows, one for each conference year cell in the panel.[18]

Table 1: Schematic structure of the final panel.

Researcher	Conference	Year	PC status	Citations
r_1	ICFP	2021	0	2
r_1	ICFP	2022	1	7
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_2	POPL	2021	0	3
r_2	POPL	2022	0	4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_i	c_j	t	p_{ijt}	y_{ijt}

4 Descriptive Event Study Analysis

RQ1 asks whether accepted papers in a conference year cite PC members' earlier work more often. I answer this descriptively by aligning each researcher and conference pair so that the first observed PC year in that same conference is $t = 0$. I then examine the event window $t \in \{-2, -1, 0, +1, +2\}$.[19]

The outcome is $\log_{10}(\text{citations} + 1)$, where citations are references from accepted papers in the same conference year to the researcher's earlier work. I use the log scale because citation counts from the same conference are sparse and skewed to the right.[20]

For each included event unit i and event time t, I compute

$$x_{i,t} = \log_{10}(\text{citations}_{i,t} + 1), \qquad d_{i,t} = x_{i,t} - x_{i,-2}.$$

Therefore, the plotted values are changes relative to two years before PC service, not raw

[16]This is not fully automatic because PC pages are name based, while OpenAlex uses author identities and cited author names. Exact name matching to a single OpenAlex author ID covers 793 of 952 PC researchers (83.3%).

[17]Appendix C reports the identity, ORCID, and manual review checks.

[18]The final output is a panel with 37,128 rows, indexed by researcher, conference, and year. It covers 952 researchers across 39 conference year combinations.

[19]I keep only researcher and conference pairs with a complete window and drop cases with another PC service in the same conference at $t = +1$ or $t = +2$. This keeps the years after PC service from being mixed with another PC service year.

[20](see Appendix D, Figures 21 and 22): many observations have zero or few citations, while a small number have much larger citations.

citation levels.[21] I show 95% bootstrap confidence intervals as error bars.[22]

For the main analysis, I focus on ICFP, POPL, and PLDI 2021 onward, together with their pooled aggregate.[23] I show two service history samples in the main Figure 7. The first panel (a) includes all event units that pass the window filters. The second (b) is the no earlier service subset: event units where the available public history checks found no earlier review service evidence in the same conference before the observed PC year.[24] Figure 7 gives the main answer to RQ1.[25] In panel (a), the aggregate change in the PC year is small: 0.034 on the $\log_{10}(\text{citations} + 1)$ scale[26], which corresponds to about 8.2% higher citations + 1 relative to $t = -2$. However, the confidence interval includes zero.

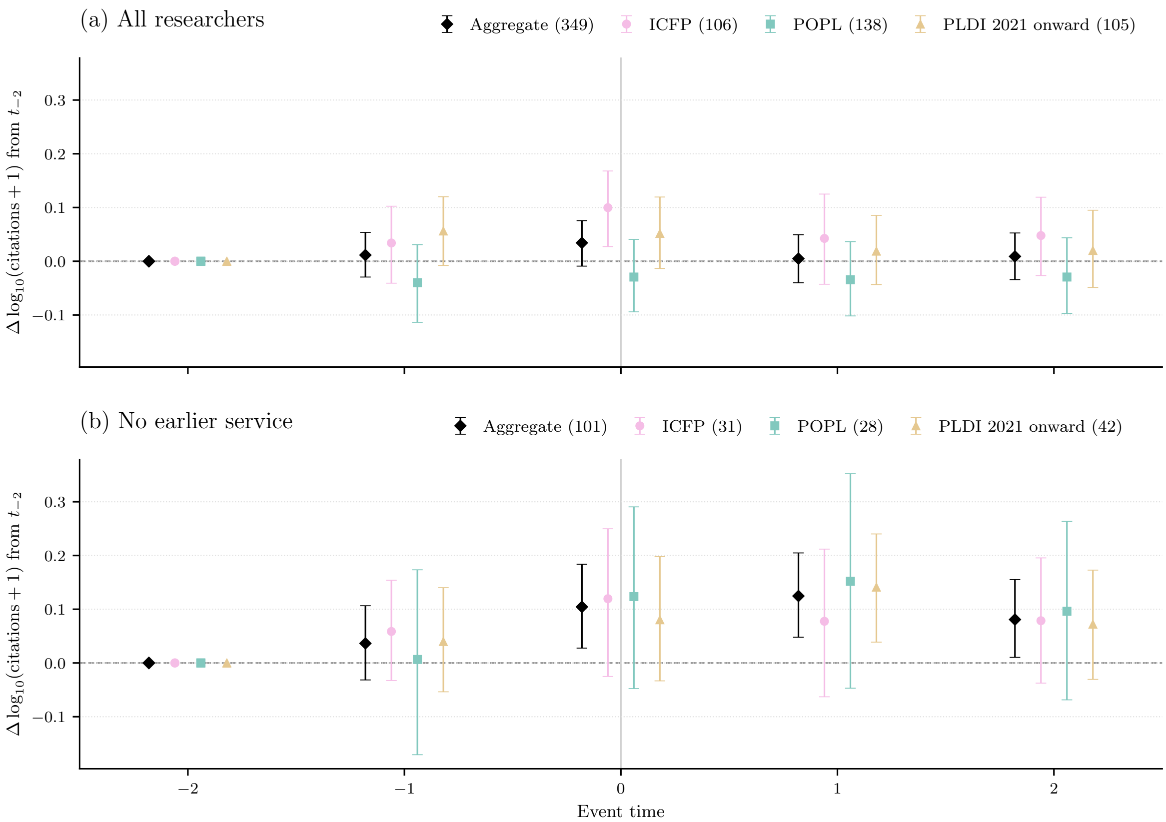


Figure 7: Log scale event study around PC service, relative to $t = -2$.

[21] A log scale change δ corresponds to a multiplicative factor of 10^{δ} in citations + 1, or $100(10^{\delta} - 1)$% higher when $\delta > 0$. Since I average log changes across event units, this percentage should be read on the geometric mean citations + 1 scale, not as a change in average raw citation counts.

[22] See the event study algorithm in Appendix D. A short robustness check using the average of $t = -2$ and $t = -1$ before PC service is reported in Figure 32.

[23] I exclude OOPSLA and PLDI before 2021 from the main analysis. OOPSLA follows the later rolling OOPSLA1/OOPSLA2 structure, and PLDI before 2021 uses a different PC and external reviewer source structure. I report the corresponding OOPSLA and PLDI before 2021 checks in Appendix D (Figures 29a and 29b and 31a and 31b)

[24] Here, public history checks mean the Researchr histories, ACM front matter PDFs, HotCRP reviewer pages, EPC/ERC pages, and selected CVs used to look for earlier service. Review service evidence means main PC or related EPC/ERC, chair, steering, or organizing roles found in those public records. This restriction is based on public records, restriction, not proof that the researcher never served earlier.

[25] The plotted values are changes in $\log_{10}(\text{citations} + 1)$ relative to $t = -2$, and the vertical bars are 95% bootstrap intervals; the raw count version is reported in Appendix D (Figure 23).

[26] (95% CI $= [-0.009, 0.076], N = 349$)

Panel (a) also does not show a uniform pattern across conferences.[27] In ICFP, the pattern looks more like a jump in the PC year followed by a partial fade, while POPL and PLDI 2021 onward are flatter or less regular.[28] In panel (b), the aggregate change in the PC year is larger: 0.104 on the log scale[29]. This corresponds to about 27.2% higher citations + 1 relative to $t = -2$.

Figure 8 shows that the no earlier service subset tends to come from researchers earlier in career stage and with lower measured visibility: it has lower median career age, lower median h-index, and lower raw citation levels before PC service than the excluded complement. Therefore, I consider panel (b) as a cleaner descriptive check, not as the same sample with only prior service removed.[30]

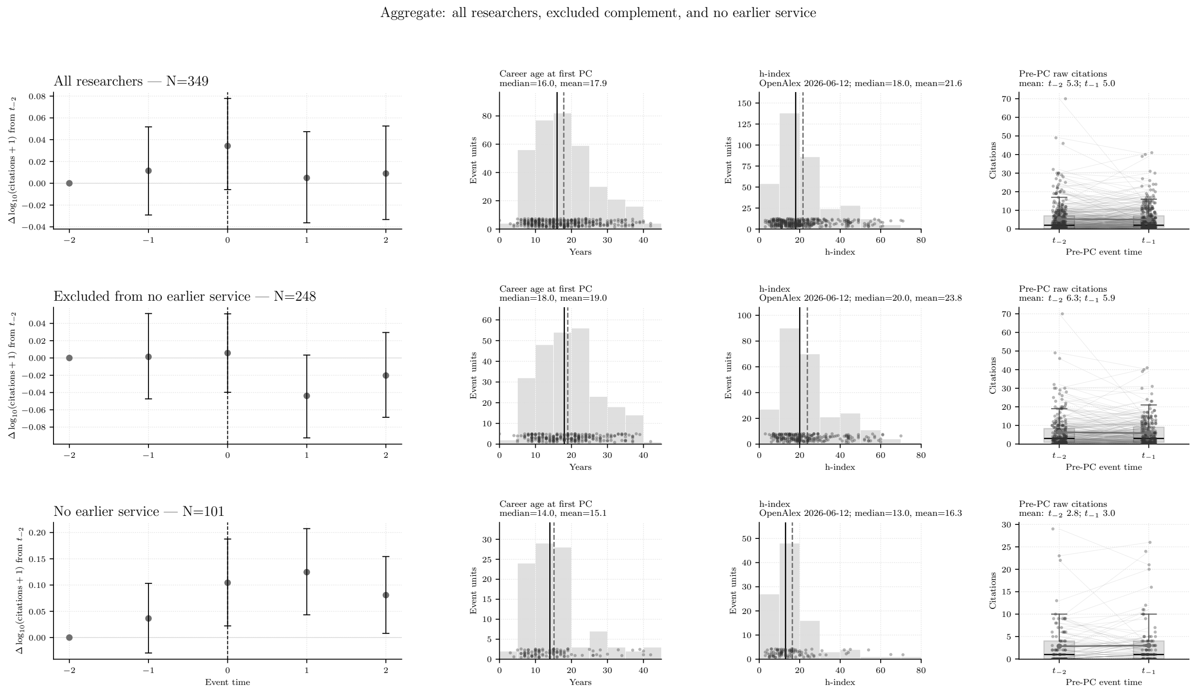


Figure 8: Career age, h-index, and citation levels before PC service in the samples.

[27] With the panel design and event study setup, I try to reduce stable differences between researchers by comparing the same researcher and conference pair over time rather than comparing different researchers to each other. This helps reduce fixed factors such as long run seniority and broad reputation. However, factors that change over time can still remain, such as citation momentum, a recent highly visible paper, or temporary topic fit with the conference. If such factors were generating a broad upward citation trajectory around PC service, I would expect the years after PC service, especially $t = +1$ and $t = +2$, to remain more clearly above the baseline before PC service. Panel (a) does not show that uniformly.

[28] This makes the service history definition important. In the 2017 to 2025 panel data, $t = 0$ is the first observed PC service year in the same conference, not necessarily the researcher's first actual service year. So, I also report the no earlier service panel (b). This is not proof that the researcher never served earlier, but it gives a cleaner subset for interpretation.

[29] (95% CI = $[0.028, 0.184]$, $N = 101$)

[30] The career age and h-index panels in Figure 8 and Figures 25 and 26 are composition checks. They show that the event study pattern is sensitive to how the sample is filtered and sliced. I use them to understand sample sensitivity, not to estimate separate career stage or conference specific effects. The selected examples below make this temporary elevation pattern more concrete and discuss possible mechanisms.

Table 2 adds one useful observation as well. In the no earlier service sample, the main movement occurs in the PC year. The change from $t = -1$ to $t = 0$ is +0.07 log units, or about +0.65 raw citations on average. The level remains above the baseline before PC service at $t = +1$ and $t = +2$, but it does not continue growing. As a result, I read the pattern as a temporary elevation around the PC year rather than as a steadily accumulating long run increase.

Table 2: No earlier service aggregate event study ($N = 101$).

event time	$\Delta \log_{10}$	95% CI (log)	% change in (citations + 1)	Δ raw cites	95% CI (raw)
$t = -2$ (ref)	0.000	n/a	0.0%	0.00	n/a
$t = -1$	+0.036	[−0.029, 0.102]	+8.7%	+0.18	[−0.56, 0.95]
$t = 0$	+0.104*	[0.025, 0.183]	+27.2%*	+0.83	[−0.05, 1.74]
$t = +1$	+0.125*	[0.045, 0.201]	+33.2%*	+0.89	[0.00, 1.83]
$t = +2$	+0.081*	[0.010, 0.151]	+20.5%*	+0.78	[−0.07, 1.69]
Year to year change:					
before PC → PC year	+0.068	n/a	+16.9%	+0.65	n/a
PC year → +1	+0.020	n/a	4.8%	+0.06	n/a
+1 → +2	−0.044	n/a	−9.6%	−0.11	n/a

Note: Stars mark log scale estimates whose 95% bootstrap intervals exclude zero. Year to year rows are descriptive point changes and are not separately bootstrapped.

Selected PC Member Examples

The aggregate event study answers RQ1 at the sample level, but it can hide individual variation. Therefore, I inspect a small set of selected PC service cases. These examples are illustrative and should not be read as representative of all PC members. Figure 9 summarizes these selected cases using the same method as the main event study.

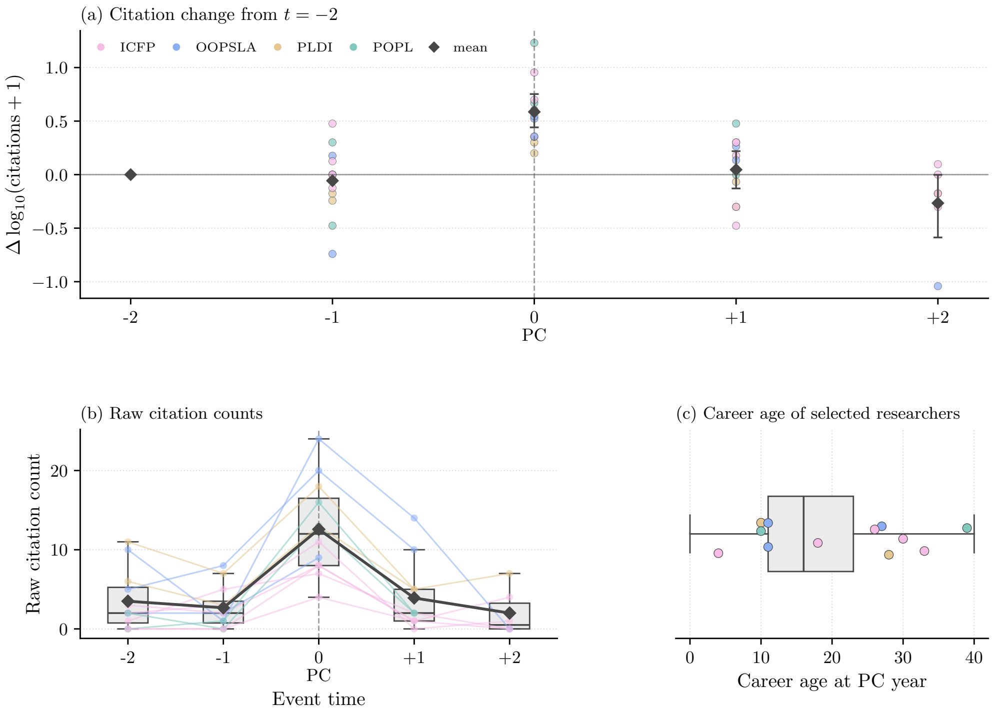


Figure 9: Selected PC member event study examples.

Figure 10 shows the same pattern separately for each selected researcher.

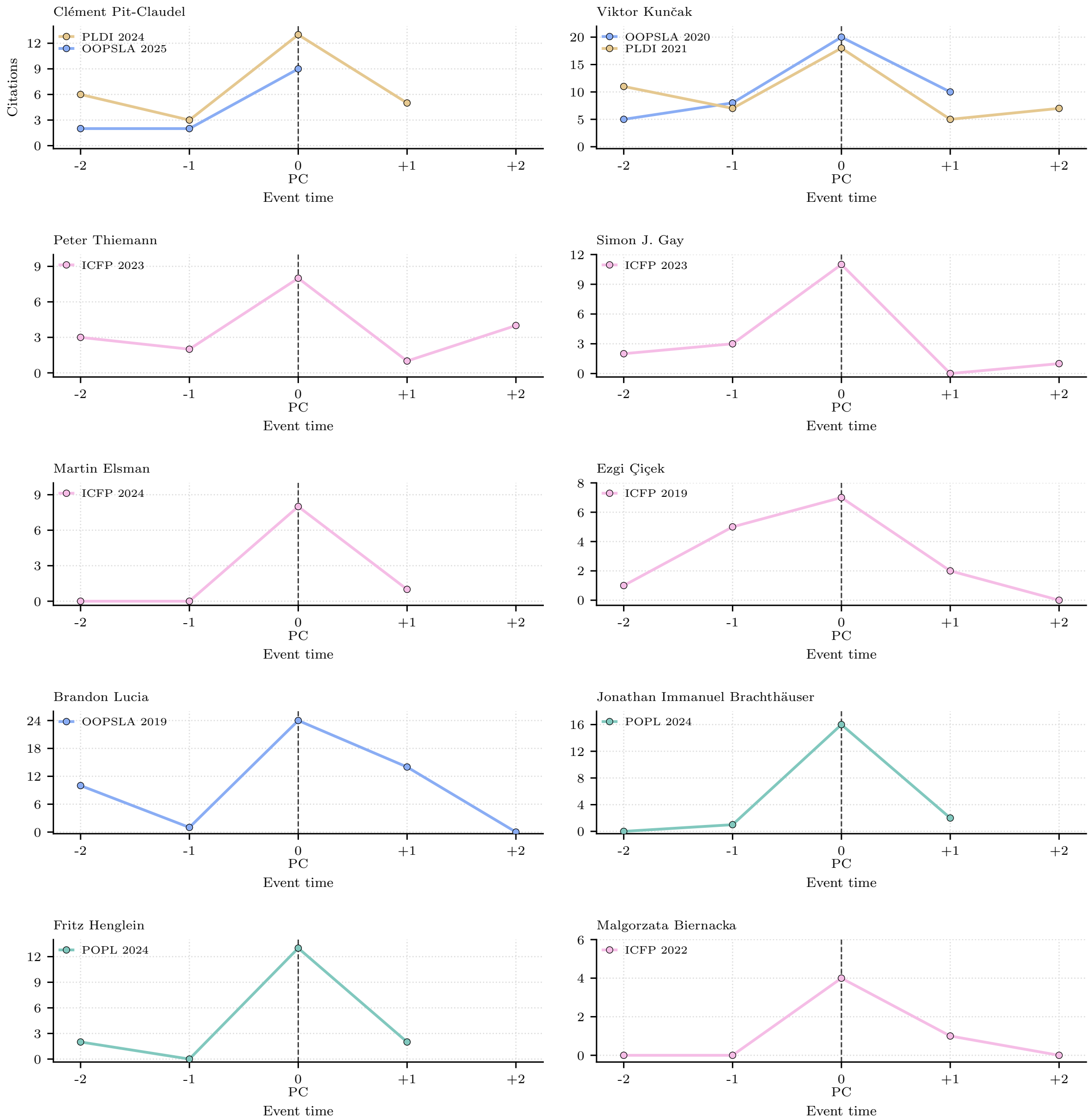


Figure 10: Selected PC member raw citation examples.

The selected examples help me interpret possible mechanisms. Authors may notice PC membership while making citation decisions; citations may also reflect recent research momentum, topic fit, or citation suggestions during review. Also, recent work discusses reviewer self citation requests and coercive citation (Barnett, 2025). My manual checks at the paper level in Appendix D also show that some increases in the PC year come from only a few citing papers, and sometimes include self citation. I use these cases as illustrative examples, not as a general pattern.

5 Citation Source Analysis

RQ2 asks: among the citations a researcher receives from the conferences, does a larger share come from the conference where the researcher is serving on the PC? To answer this question, I use the same event window of five years as before, but I make three changes.[31]

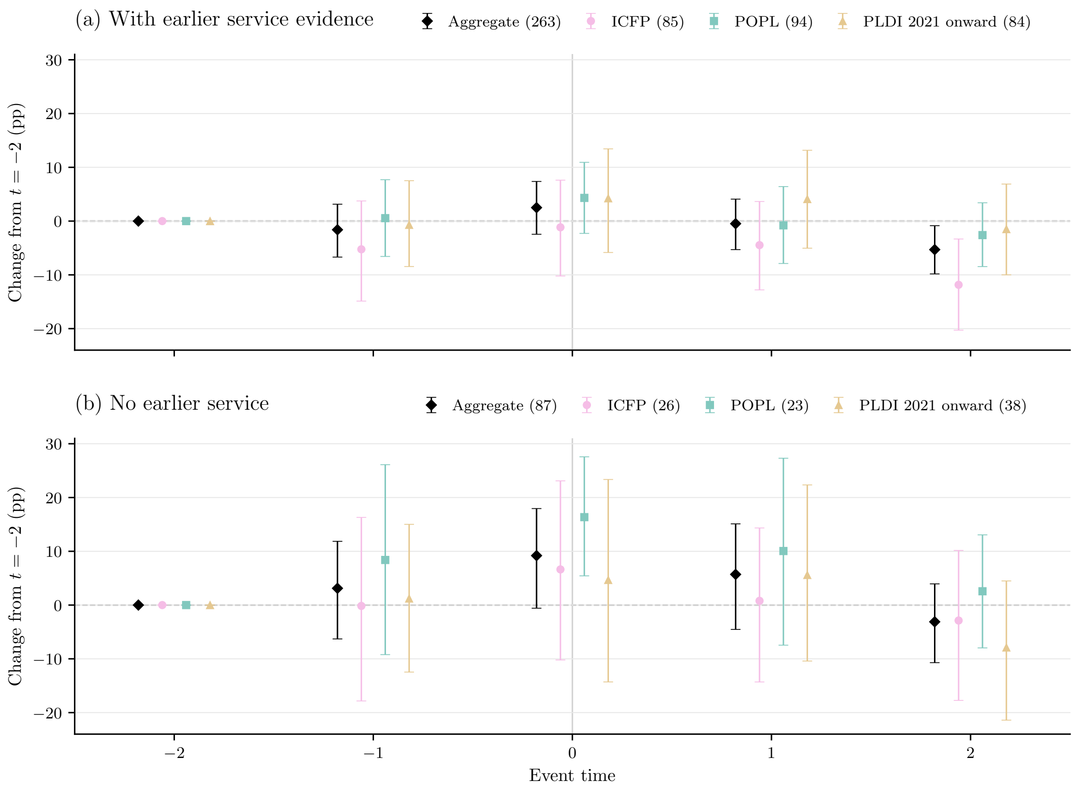


Figure 11: Citation source shift around PC service.

Figure 11 reports the resulting citation source shift for ICFP, POPL, and PLDI 2021 onward, together with their aggregate.[32] It gives the main answer to RQ2. In the panel (a), the aggregate shift in the PC year is small: +2.5 percentage points.[33] In the no earlier service panel (b), the aggregate shift in the PC year is larger: **+9.2 percentage points.**[34]

[31]First, I keep only event units with exactly one observed PC service year in the same conference. This makes the event window easier to interpret, because the observed change in citation share can be associated with one PC year. Second, I replace the citation count outcome with the share of the researcher's conference citations that comes from the event conference. Third, I measure the change in that share relative to $t = -2$, so the final outcome is a percentage point change rather than a citation count level.

[32]The remaining checks by conference are reported in Appendix D. I compute confidence intervals by bootstrapping event units; see the citation source algorithm in Appendix D.

[33](95% CI $= [-2.5, 7.4]$) across 263 event units, with 220 defined source share changes in the PC year.

[34](95% CI $= [-0.6, 17.9]$) across 87 event units, with 62 defined source share changes in the PC year. In both panels, the largest percentage point increase appears at $t = 0$, and the path softens again at $t = +1$ and $t = +2$. This suggests a temporary shift in citation share toward the conference around the PC year than a persistent increase after it. However, I cannot interpret this pattern causally. The outcome is compositional, so an increase in the event conference's share must come with a decrease in the share of other conferences. In addition, PC service is not randomly assigned, and confounders such as recent visibility or citation momentum, can also generate a temporary shift in citation share.

Figure 12 also shows two selected illustrative examples at the researcher level. Each mini chart breaks the researcher's citations into source conferences across the event window. These examples show how the event conference can take a larger share of the researcher's conference citations around the PC service year. The remaining selected researcher cases are reported in Appendix D (Figure 39).

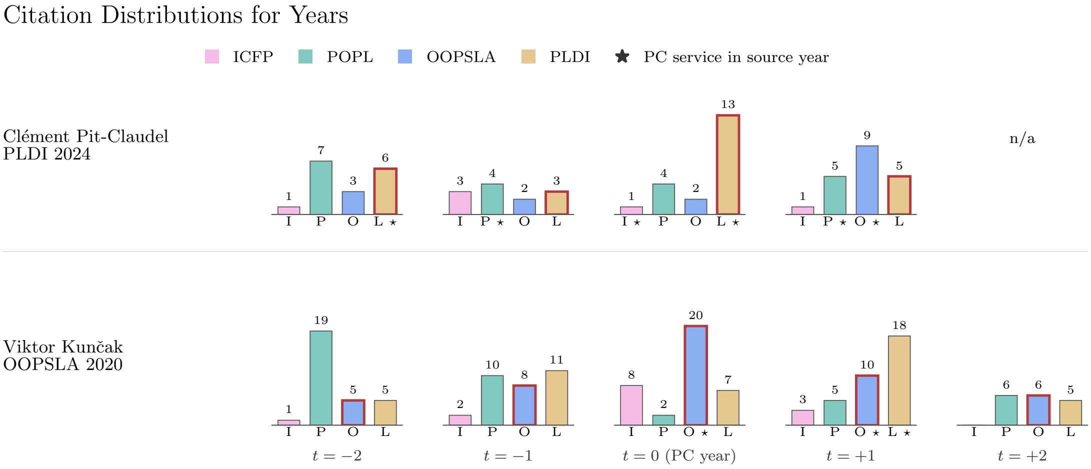


Figure 12: Selected citation source concentration examples around PC service.

6 Discussion

Understanding whether public PC membership shapes citation decisions matters for the fairness and transparency of scientific communication. Ideally, authors should cite prior work because it is relevant to their research, not because its author is temporarily more visible on a PC list. In the programming languages conferences I study, PC lists are often public before or during submission and review, so authors may see PC members' names while citation decisions are still being made. This motivates the question of whether accepted papers cite PC members' prior work differently around years of PC service.

Results: In the less restricted samples, citation changes around years of PC service are small and uneven. However, the pattern is more visible in the subset with no earlier service evidence in the same conference, which also tends to contain researchers earlier in their careers and with lower measured visibility. In that subset, papers accepted at the conference cite PC members' earlier work more around the PC year, and the serving conference's citation share increases by about 9.2 percentage points relative to two years before PC service. The citation source shift softens after the PC year, and the citation count pattern partly fades by $t = +2$. Since PC service is not randomly assigned and several mechanisms could produce the same pattern, I interpret these findings as descriptive rather than causal.

Limitations

Design limitations: I cannot observe the ideal counterfactual: how the same researcher would have been cited in the same conference year without PC service. PC service is also not random: researchers are invited because of visibility, expertise, topic fit, and community involvement, which can also affect citations. The panel design reduces fixed differences between researchers, but it does not remove changing factors such as citation momentum, recent papers, or temporary topic fit. Also, I only observe final accepted papers, so I cannot tell whether a reference was already in the submission or added during review or revision after conditional acceptance. Therefore, with this design, I cannot tell whether citation changes come from PC list visibility, citation suggestions during review, recent research momentum, or topic fit.

Data limitations: The first observed year of PC service in my data may not be the researcher's first actual PC service. I measured this limitation partially. I study four programming languages conferences between 2017 and 2025, so I can identify the first PC service year observed inside this panel. I also check available public service histories.[35]

For OpenAlex, I observe enough reference information to construct the citation outcome, but there are two separate limitations. The first is reference recall: I do not estimate how many true bibliography entries OpenAlex captures across the full sample. I only check this indirectly.[36] The second limitation is cited author metadata among the references that OpenAlex does capture.[37] When this happens, the reference cannot contribute to the panel citation outcome, which can lead to undercounting.

At the unique cited work level, there is missing data for cited author names and IDs.[38] This missingness varies across conference years, but I do not observe a systematic conference year pattern. I also observe duplicate OpenAlex work IDs.[39] This can make citation counts larger than counts based on grouping records with the same cited title.

Author matching adds another measured limitation. The strictest case is exact name matching to one OpenAlex author ID, which covers 793 of 952 PC researchers (83.3%).[40] ORCID helps validate some matches, but ORCID.org name searches return multiple candidates for 338 of 952 researchers (35.5%).[41]

---

[35] Across 1,837 researcher and conference pairs, 480 pairs (26.1%) have earlier PC evidence in the same conference before the first PC year observed in my panel, and 1,181 pairs (64.3%) have earlier review service evidence in the same conference.Review service evidence means main PC or related EPC/ERC, chair, steering, or organizing roles found in the public records. However, these checks depend on available public records, so absence of earlier evidence is not proof that the researcher never served earlier. Duplicate source profiles are a small issue in this dataset; the manual merge list is reported in Appendix A, Table 4.

[36] Accepted papers have a median of 41 OpenAlex references (IQR 30 to 54).

[37] At the reference edge level, 7,588 of 110,167 edges (6.9%) have no cited author name, and 11,652 of 110,167 edges (10.6%) have no cited author ID.

[38] 3,560 of 44,334 cited works (8.0%) have no cited author name, and 5,674 of 44,334 cited works (12.8%) have no cited author ID. Across conference years, the cited author ID missingness ranges from 5.1% to 15.6%. Figures 18 and 19 show this missingness by conference year.

[39] In the cited author data, 3,111 of 37,283 cited title groups (8.3%) map to multiple OpenAlex work IDs.

[40] After manual and name checks, citation outcomes can be constructed for 942 of 952 researchers (98.9%). Appendix C, Table 8 shows how citation counts are assigned; Table 9 lists the 10 unresolved cases; and Figure 20 summarizes the ORCID checks.

[41] For 153 of 952 researchers (16.1%), OpenAlex gives no ORCID in the author records I use for matching. Therefore, ambiguous names, missing ORCIDs, manual name maps, and unresolved cases remain part of the measurement error.

Citation counts are sparse and skewed to the right.[42] Career age and h-index are proxies.[43]

Generalizability limitations: I study four programming languages conferences between 2017 and 2025, so the results could be considered in the scope about this setting, not as a general claim about all scientific fields or all peer review systems. These conferences have public PC pages, strong community overlap, and reviewing structures that are specific to the field. Other fields may have different committee visibility, citation norms, publication timelines, and reviewer assignment practices.

The no earlier service sample is not random either: it tends to include researchers earlier in their careers and with lower measured visibility. The patterns may therefore change under other sample definitions or in other communities.

Future Work

The immediate next step would be to use the prepared panel data in a matched or weighted observational design. The unit would be an observation by researcher, conference, and year (i, c, t). The treatment would be whether researcher i serves on conference c's PC in year t. The outcome would be the number of citations from papers in conference c, year t, to researcher i's papers published before year t. The comparison group would be researchers not yet on the PC from the same conference year who looked similar before the PC year.

One option is propensity score matching or weighting, where PC service observations are compared with observations for researchers not yet on the PC that had similar estimated probabilities of PC service before the PC year. The matching variables should all be measured before PC service, such as prior citation history, productivity, career age, topic fit, and service history. The design should then report covariate balance, outlier sensitivity, and checks using fake PC years or years before PC service. Because unobserved professional networks may still affect both PC selection and citations, Rosenbaum's Γ sensitivity analysis would be a useful way to quantify how strong hidden bias would need to be to change the conclusion (Rosenbaum, 2020).[44]

Future work would also apply this framework to other research communities. For another field, the same work would require the same basic checks: defining program committee service, validating PC data and citation metadata, and linking PC names to bibliographic author records. Applying this framework to fields such as machine learning, human computer interaction, computational social science, or psychology would make it possible to test whether the patterns I observe are specific to programming languages conferences or also appear elsewhere. Comparing fields would show when visibility from conference service matters most.

[42] As shown in Appendix D, Figures 21 and 22. A few papers or researchers can therefore shape the raw count patterns.

[43] Appendix D, Table 13 reports the career age checks for the selected PC member examples.

[44] Even then, reference timing remains difficult because the outcome uses final accepted papers, but we do not observe whether a reference was already present at submission or added during revision. This matters because different mechanisms can produce the same final citation count: authors may have already known the work, the cited researcher may have been selected for the PC because the topic was already important for that conference, or references may have been added during review. A stronger future design would compare submitted and final reference lists when paper versions are available.

7 Timeline and Reflection

The project timeline changed from my proposal as follows.

Weeks 1 to 3: Data construction: I planned to build the first PACMPL/OpenAlex pipeline, starting with ICFP and POPL. This part was mostly on track: I scraped the first PC lists, fetched the accepted paper data, expanded the OpenAlex references, and started the first metadata checks. However, this stage took longer than I expected. One reason was that I did not freeze the OpenAlex data immediately; because OpenAlex metadata changes over time, I kept updating the data while the analysis was still evolving. Another reason was that validation became more involved than I had planned: I checked author identities not only by name, but also with OpenAlex author IDs and ORCID evidence.

Weeks 4 to 6: Dataset and confounders: I planned to work on confounders and variables. In practice, I first had to make the researcher matches more reliable. I checked PC page names against OpenAlex author IDs, ORCID evidence, Researchr DOI evidence, and manual cases. This took longer than expected, but it was necessary because the citation counts depend on linking the right PC member to the right OpenAlex author record. During the same period, I also started moving from simple citation summaries to an event time analysis around first observed PC service.

Weeks 7 to 9: Main analysis: I planned to run the main analysis within researchers. I did this, but the result became more descriptive than I first expected. Although the initial plots came together faster than the data validation, interpreting them took longer. I also created a small website to explore the event study results visually.[45] This was not in my plan, but it helped me compare slices of the data and see that pooled results can hide different service history patterns. The first ICFP and POPL plots also showed that some researchers looked like first time PC members inside my 2017 to 2025 panel even though they may have served earlier. This pushed me to validate service history before making stronger claims.

Weeks 10 to 13: Extension and validation: I planned to either extend the analysis or validate the results. In practice, I did both within the programming languages setting, so this stage became broader than I expected. I added PLDI, reran the pipeline, and refreshed the validation checks. The event study results also made me look more carefully at earlier PC service, so I checked Researchr profiles, HotCRP pages, ACM front matter PDFs, and conference websites. I also split the event study samples by career age, h-index, and citation levels before PC service to see how sensitive the results were to sample composition. This helped me see that OOPSLA's rolling OOPSLA1/OOPSLA2 structure and PLDI's source change before and after 2021 had to be handled explicitly in the analysis.

Weeks 13 to 14: Final presentation and documentation: I planned to prepare the final presentation and report. This stage took longer than I expected because the report had to document not only the results, but also the data decisions behind them. I cleaned the notebooks, regenerated figures and tables from scratch, organized the appendix and my repository, and wrote the reproducibility notes.[46]

[45] https://sariender.github.io/citemeifyoucan/

[46] The code repository is available at https://gitlab.epfl.ch/systemf/student-projects/2026-02-citations-vs-pc-memberships.

References

[1] Barnett, A. (2025). Are peer reviewers influenced by their work being cited? *eLife*, 14, RP108748. https://doi.org/10.7554/eLife.108748.4

[2] Rosenbaum, P. R. (2020). *Design of Observational Studies* (2nd ed.). Springer Nature. https://doi.org/10.1007/978-3-030-46405-9

Appendix A: Data, Step 1: PC service data

Figures

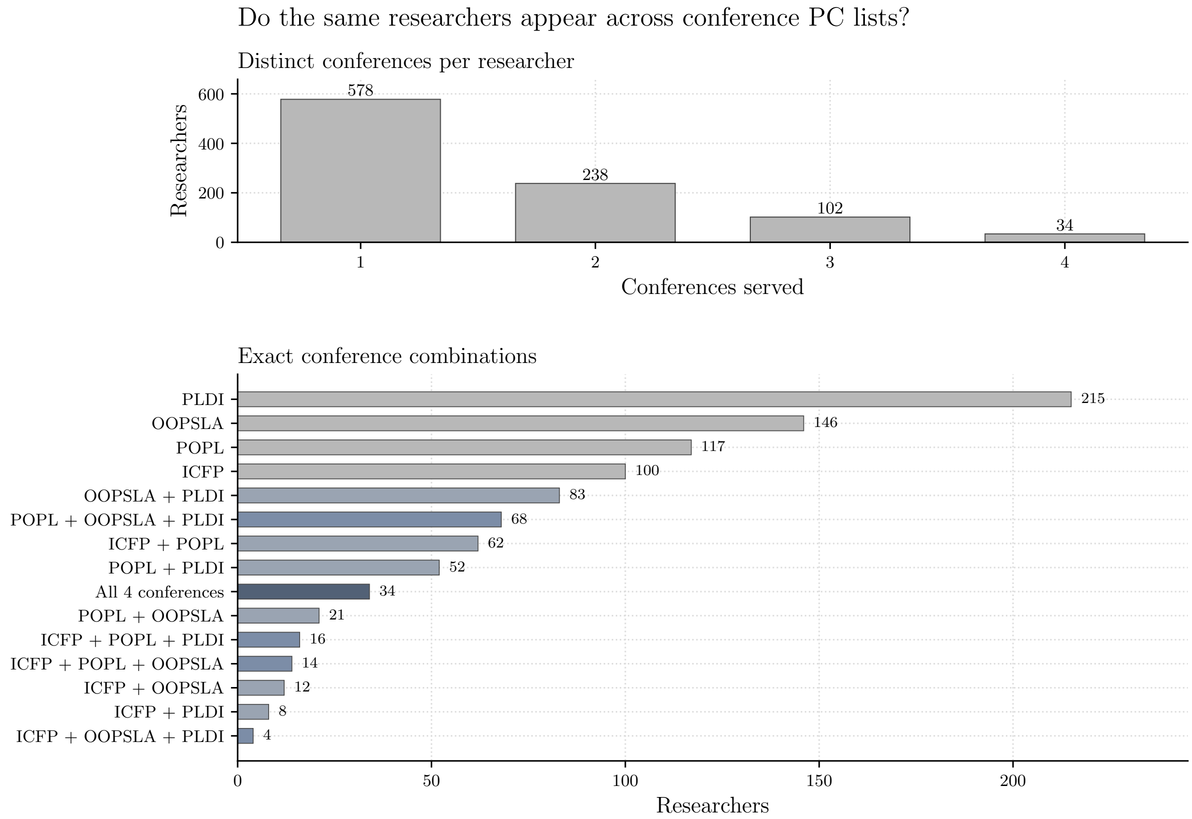


Figure 13: Conference overlap in the PC data.

Tables

Source type	Scope	Link
PC/review page	In scope	PLDI 2025 papers review page
Side track page	Out of scope	PLDI 2025 SRC page
HotCRP reviewer list	Validation	ICFP 2019 HotCRP PC page
EPC/ERC page	Validation	PLDI 2018 EPC page
ACM DL proceedings	History	ACM DL PLDI proceedings
ACM PACMPL issue	History / titles	ACM PACMPL 2019 issue archive
ACM front matter PDF	History / titles	ACM front matter PDF example

Table 3: Step 1 source link samples.

Researcher	Merged source profile ID(s)
Guoqing Harry Xu	`guoqingharryxu`
Vikash K. Mansinghka	`vikashmansinghka1`
Corina S. Pasareanu	`corinapasareanu`; `corinaspasareanu1`
Görel Hedin	`gorelhedin1`
Michelle Mills Strout	`michellestrout`

Table 4: Manual duplicate profile merges.

Researcher	Services	Conferences	Conference set	Years
Nadia Polikarpova	11	4	ICFP, OOPSLA, PLDI, POPL	2018 to 2025
Sukyoung Ryu	11	3	OOPSLA, PLDI, POPL	2018, 2020 to 2025
Adam Chlipala	8	3	ICFP, POPL, PLDI	2017, 2018, 2020, 2021, 2023, 2025
Arjun Guha	8	4	ICFP, OOPSLA, PLDI, POPL	2018, 2020, 2021, 2023 to 2025
Colin Gordon	8	3	OOPSLA, PLDI, POPL	2018, 2020 to 2022, 2024, 2025
Hakjoo Oh	8	4	ICFP, OOPSLA, PLDI, POPL	2018, 2021, 2022, 2024, 2025
Jan Hoffmann	8	3	ICFP, POPL, PLDI	2017, 2019 to 2023, 2025
Neel Krishnaswami	8	4	ICFP, OOPSLA, PLDI, POPL	2018, 2019, 2022 to 2024
Ranjit Jhala	8	4	ICFP, OOPSLA, PLDI, POPL	2017, 2021 to 2025
Ronald Garcia	8	4	ICFP, OOPSLA, PLDI, POPL	2017, 2019, 2021, 2022, 2024, 2025

Table 5: Researchers with the most repeated PC service records.

Researcher	Linked service records in the PC member data
Adam Chlipala	ICFP 2017 (PC Member); POPL 2017 (PC Member); PLDI 2018 (PC Member); ICFP 2020 (PC Chair); PLDI 2021 (PC Member); POPL 2021 (PC Member); PLDI 2023 (PC Member); POPL 2025 (PC Member). Additional check: CV.
Clément Pit-Claudel	PLDI 2022 (PC Member); POPL 2023 (PC Member); ICFP 2024 (PC Member); PLDI 2024 (PC Member); OOPSLA 2025 (PC Member); POPL 2025 (PC Member). Additional check: CV.

Table 6: Example linked PC service histories.

Appendix B: Data, Step 2: Papers & citations

Figures

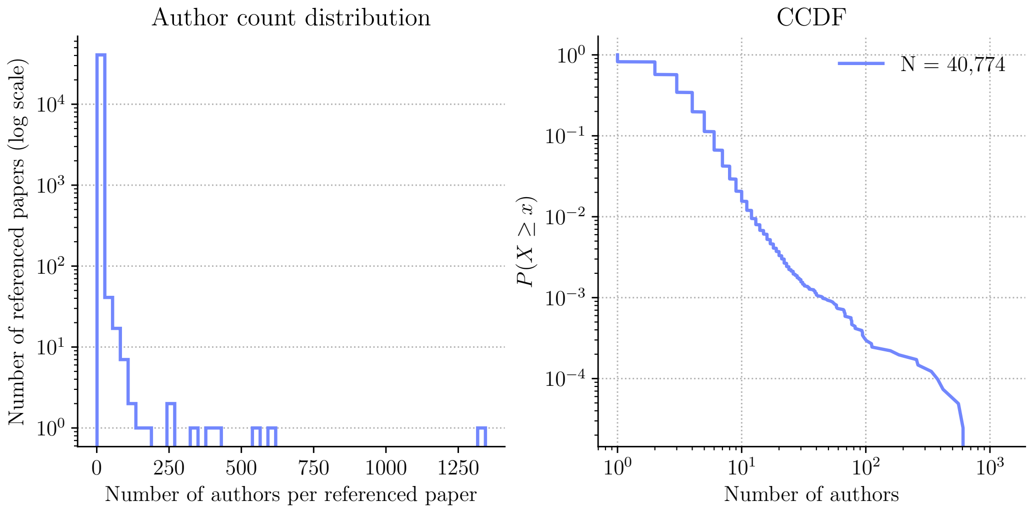


Figure 14: Distinct cited author names per referenced work.

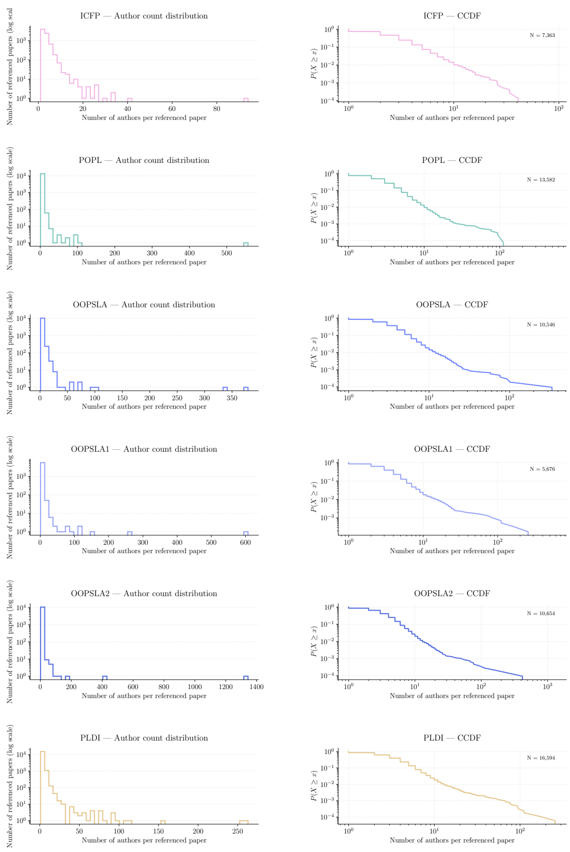


Figure 15: Distinct cited author names per referenced work, by citing conference.

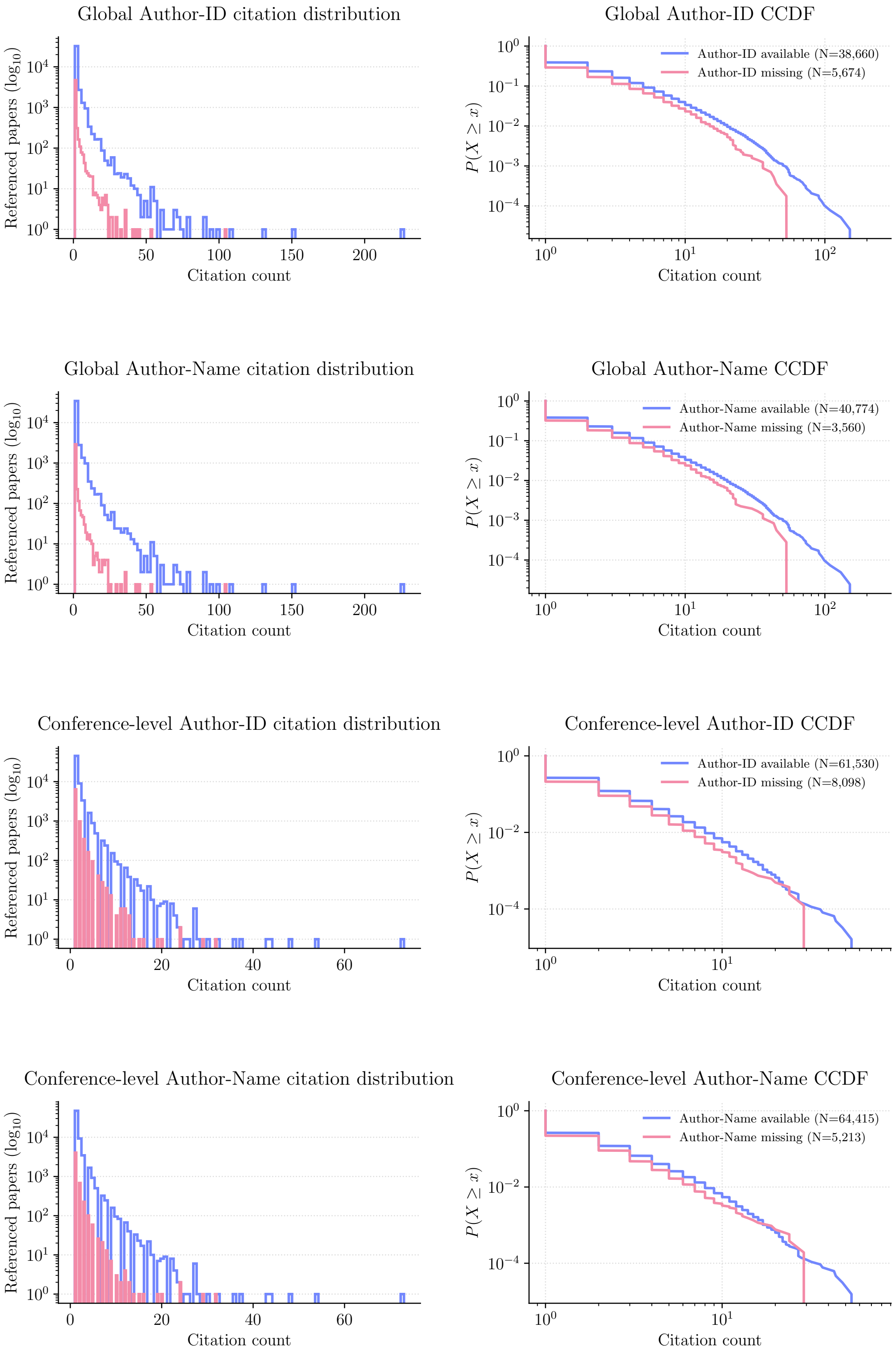


Figure 16: Referenced work citation distributions by cited author metadata availability.

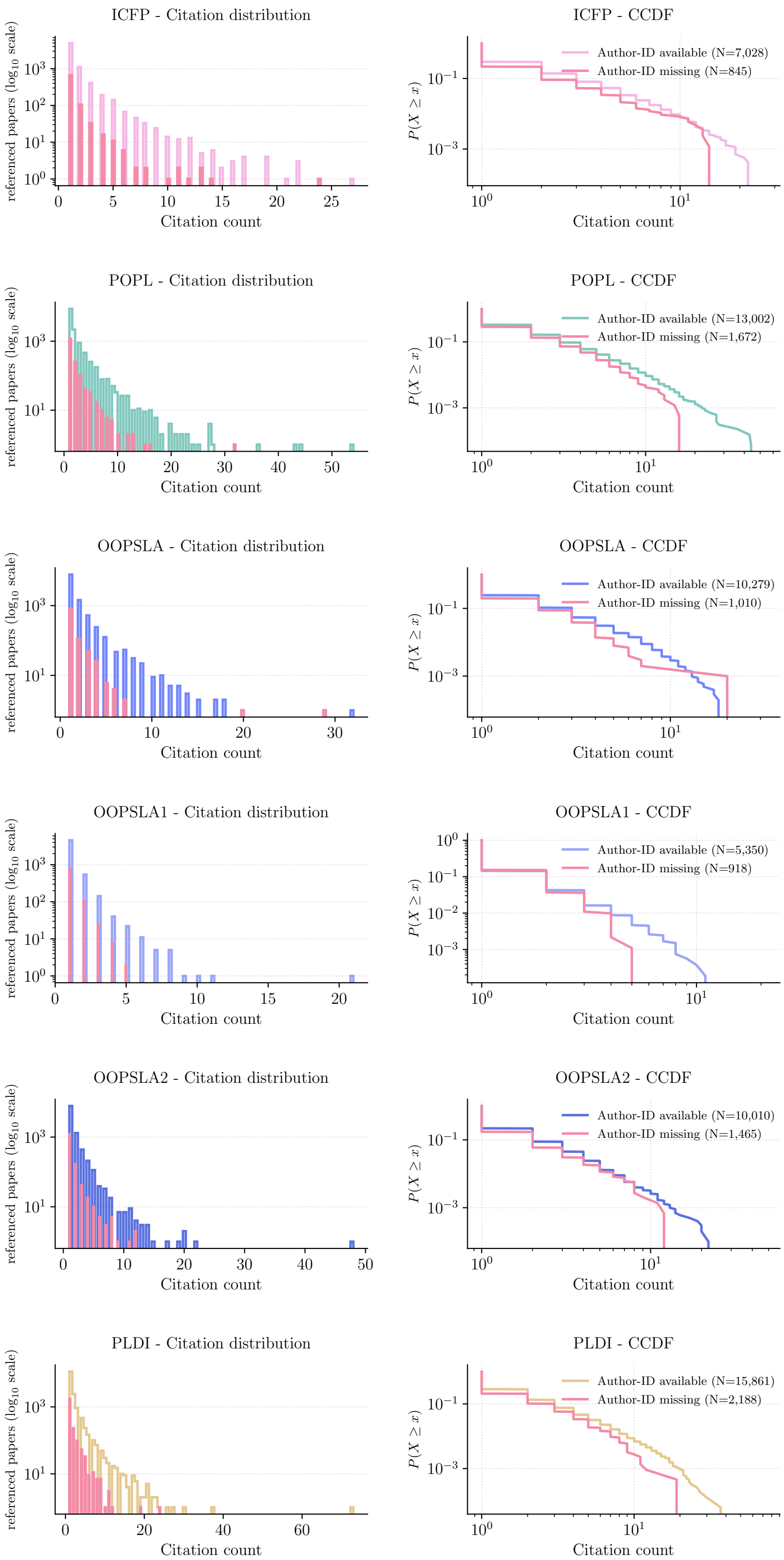


Figure 17: Referenced work citation distributions by conference.

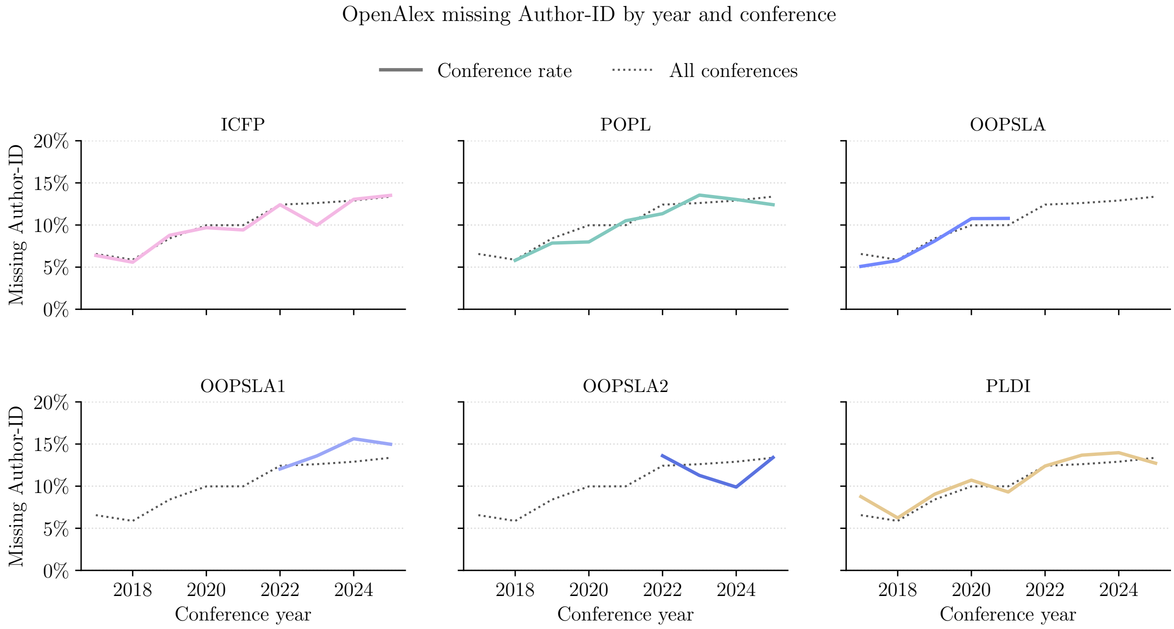


Figure 18: OpenAlex cited author ID missingness by conference year.

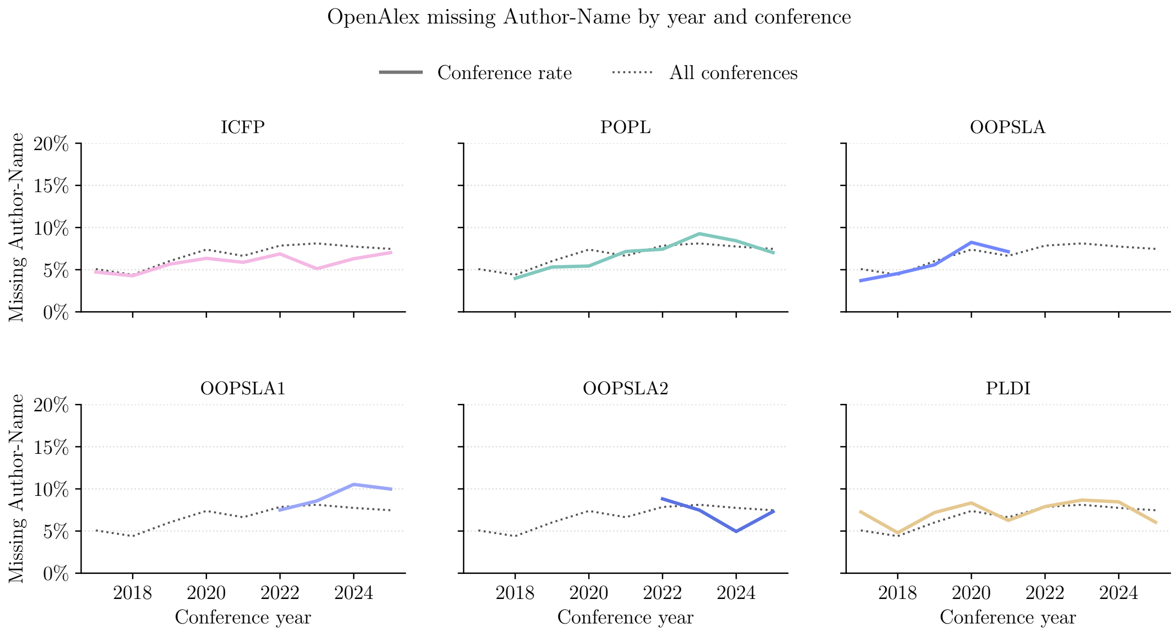


Figure 19: Missing OpenAlex cited author names by conference year.

Tables

Work ID	Referenced work	Authors	Year	Citations	Threshold
`W1480909796`	Z3: An Efficient SMT Solver	Leonardo de Moura, Nikolaj Bjørner	2008	227	200+
`W2901454403`	Iris from the ground up: A modular foundation for higher-order concurrent separation logic	Ralf Jung, Robbert Krebbers, Jacques-Henri Jourdan, Aleš Bizjak, et al.	2018	151	150+
`W2043100293`	Abstract interpretation	Patrick Cousot, Radhia Cousot	1977	130	100+
`W2137628566`	Separation logic: a logic for shared mutable data structures	John Reynolds	2003	109	100+
`W4285719527`	`W4285719527` (OpenAlex work lookup returned `null`)	Missing in direct OpenAlex lookup	n/a	105	100+

Table 7: Referenced works cited more than 100 times in the corpus.

Appendix C: Step 3: Final Panel Construction

Identity Checks

Match basis	N	Share	Use in citation counting
Single OpenAlex author ID	793	83.3%	The mapped PC name appears in the cited author data and points to one OpenAlex author ID.
Manual alternate name accepted	51	5.4%	Manual review accepts a different cited author name as the same researcher and uses that accepted match for citation counting.
Ambiguous name match	84	8.8%	The PC name appears in the cited author data, but it points to multiple OpenAlex author IDs, so citation counts are kept with an ambiguity flag.
Name only match	14	1.5%	The researcher appears in cited author names, but there is no stable OpenAlex author ID behind the match.
Unresolved	10	1.1%	No reliable OpenAlex evidence is accepted, so these researchers stay in the panel with missing citation outcomes.

Table 8: How citation counts are assigned for the 952 PC researchers.
Note: The rows sum to all 952 PC researchers. Only the unresolved cases remain in the panel without citation outcomes.

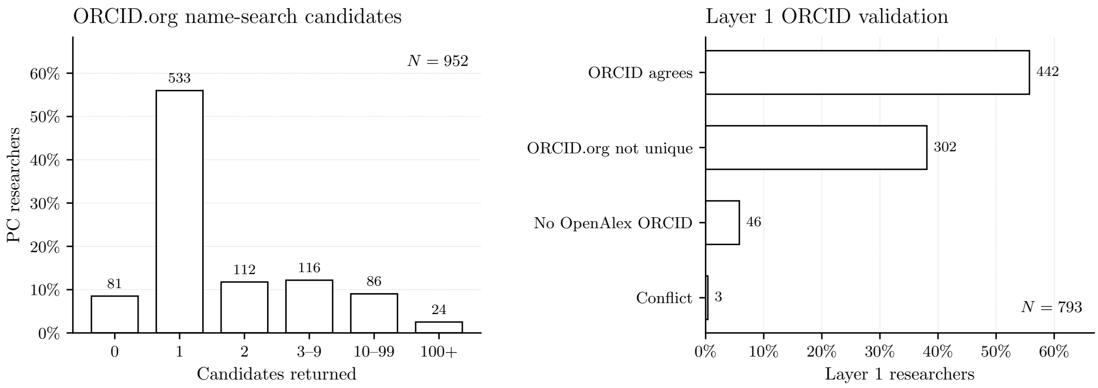


Figure 20: ORCID.org validation checks for PC researchers.

Table 9: Unresolved PC researchers kept in the panel with missing citation outcomes.

PC name
Alexey Loginov, Angelica Moreira, Dalal Alrajeh, Frank Fu, Julien Verlaguet, Kristóf Marussy, Marie Kerjean, Neha Agarwal, Suparna Bhattacharya, Vaishnavi Sundararajan

Appendix D: Analysis Details

Outcome Checks

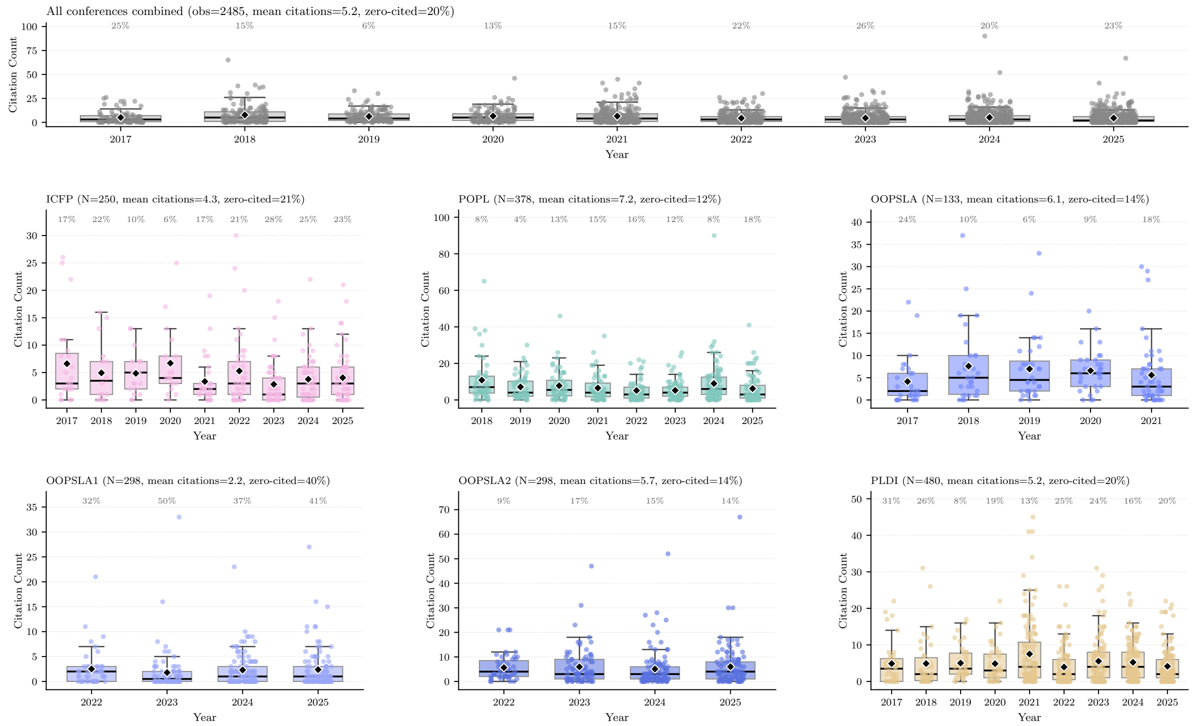


Figure 21: Raw citation count distribution around PC service.

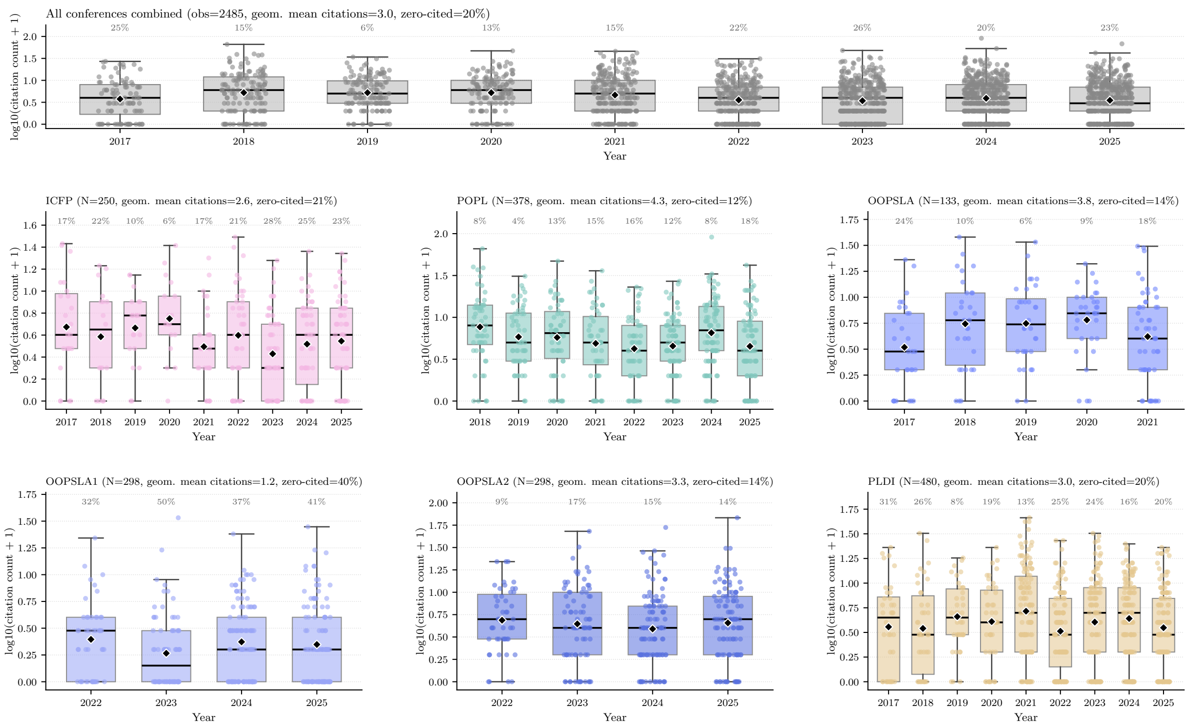


Figure 22: Log scale citation count distribution around PC service.

Conference	Citation obs.	Missing	Zero obs.	Zero researchers	Zero share	Mean	Median
ICFP	317	1	67	60	21.1%	4.33	3.0
OOPSLA	174	2	24	23	13.8%	6.13	4.0
OOPSLA1	344	4	139	130	40.4%	2.24	1.0
OOPSLA2	344	4	49	48	14.2%	5.65	3.0
PLDI	783	3	154	140	19.7%	5.20	3.0
POPL	523	0	62	55	11.9%	7.18	4.0

Table 10: PC service observations with zero citations in the conference year.

Detailed $t = -2$ Event Study Construction & Figures

For RQ1, I use the panel with one row per researcher, conference, and year. In a panel row (r, c, y), $P_{r,c,y}$ records whether researcher r served on conference c's research paper PC in year y, and $Y_{r,c,y}$ records citations from accepted papers in conference c, year y, to researcher r's earlier work.

For RQ1, I ask whether $Y_{r,c,y}$ changes around PC service. I first restrict to researcher and conference pairs with usable author identity information and career age metadata, because the event study figures also report career age and h-index composition checks.

For each researcher and conference pair (r, c):

1. Look across the panel rows for (r, c), and find the years in which $P_{r,c,y} = 1$. These are the observed years in which researcher r is listed on conference c's research paper PC.

2. If researcher r is never observed on conference c's PC, drop the pair. Otherwise, let y_0 be the first observed PC year for this researcher and conference pair, and set this year as event time $t = 0$.

3. Use the event window
$$t \in \{-2, -1, 0, +1, +2\}.$$
For each event time t, the corresponding calendar year is $y_0 + t$.

4. Keep the pair only if all five calendar years $y_0 - 2, \ldots, y_0 + 2$ are observed in the panel.

5. Drop the pair if the researcher is listed again on conference c's PC at $t = +1$ or $t = +2$. This prevents the window after PC service from being mixed with another PC service year for the same researcher and conference. Repeated service outside these two years after PC service is allowed.

6. Define the service history sample. For the all researcher sample, keep every pair that passes the filters above. For the no earlier service sample, keep only pairs where the available public history checks found no earlier review service evidence in the same conference before y_0. This filter is based on public records, not proof that the researcher never served earlier.

7. For each remaining pair and event time, take the citation count from the panel:
$$Y_{r,c,y_0+t}.$$

This is the number of citations from accepted papers in conference c, in year $y_0 + t$, to researcher r's earlier work.

8. Transform the citation count:

$$x_{r,c,t} = \log_{10}(Y_{r,c,y_0+t} + 1).$$

I use +1 so that observations with zero citations remain in the analysis.

9. Use $t = -2$ as the reference year. For each event time, compute the change within each pair:

$$d_{r,c,t} = x_{r,c,t} - x_{r,c,-2}.$$

10. Average $d_{r,c,t}$ across the included researcher and conference pairs, separately for each event time.

11. Compute 95% bootstrap intervals by resampling researcher and conference pairs with replacement. In each bootstrap repetition, keep each selected pair's full trajectory across the five event times together and recompute the average change at each event time. The interval is the 2.5th and 97.5th percentiles of the 2,000 bootstrap averages.

The base panel does not contain the no earlier service filter. I construct this filter after building the event window rows, using the public service history fields constructed from the Step 1 source checks.

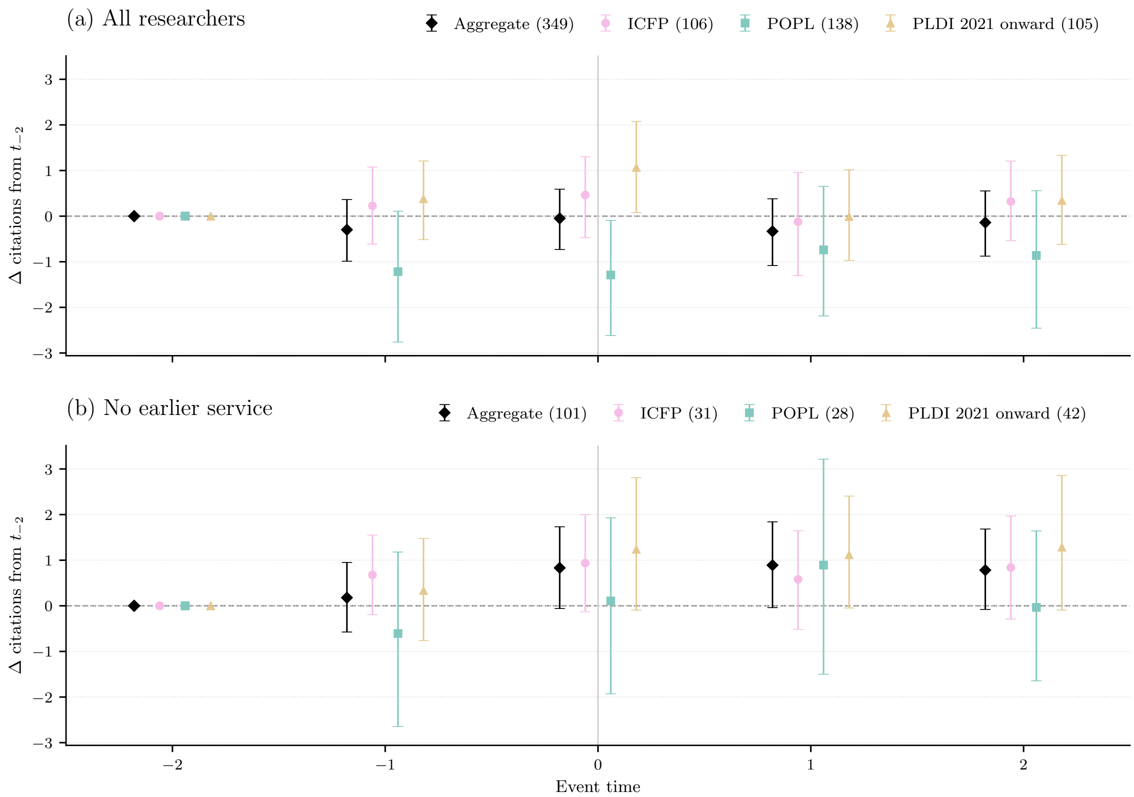


Figure 23: Raw count version of the main text event study figure.

Table 11: Event study sample counts.

Sample	N
Panel rows	37,128
Researchers in panel	952
Event study pairs with a full window (all conferences)	403
No earlier service event study pairs (all conferences)	113
Main text event study pairs (ICFP, POPL, PLDI 2021 onward)	349
Main text no earlier service pairs	101
Citation source pairs (one observed PC year in the same conference)	263
Citation source pairs with defined source share change in the PC year	220
Citation source no earlier service pairs	87
Citation source no earlier service pairs with defined source share change in the PC year	62

Note: An event unit is one researcher and conference pair. "No earlier service" means that the history checks found no earlier review service evidence in the same conference for that pair. For the citation source analysis, the event unit count includes the full balanced window sample; the source share estimate at $t = 0$ uses only event units with a defined $t = -2$ baseline and a defined source share in the PC year.

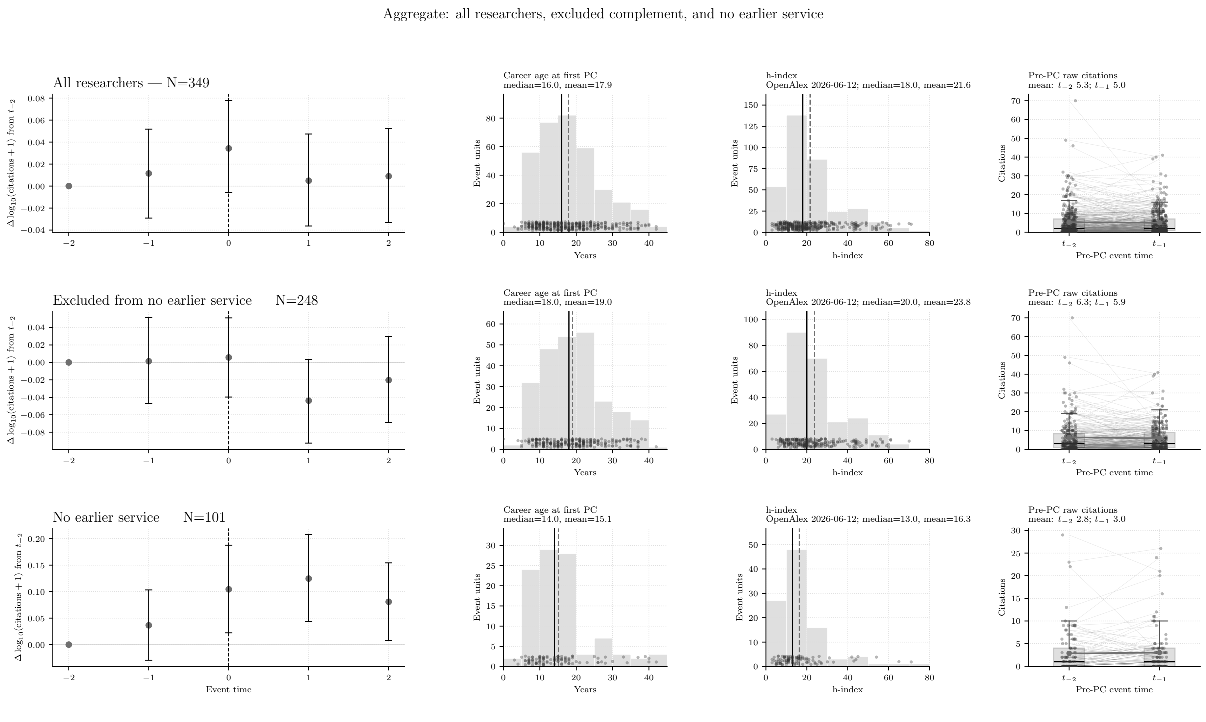


Figure 24: Aggregate sample comparison for ICFP, POPL, and PLDI 2021 onward.

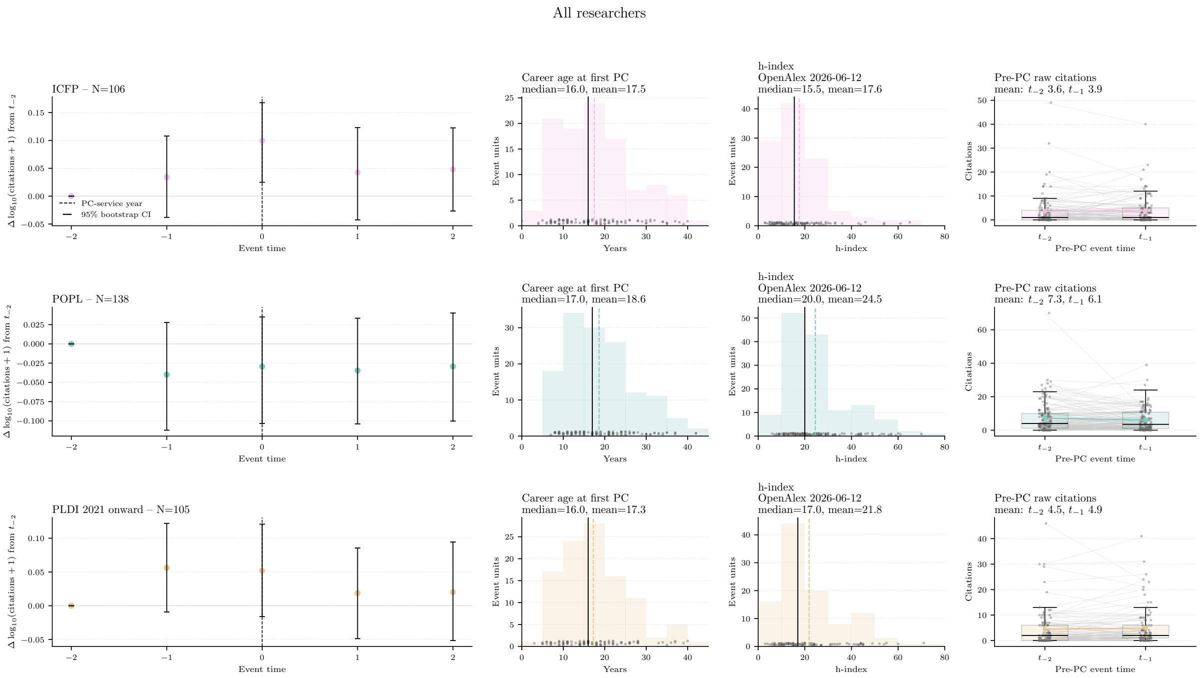


Figure 25: Conference comparison for all researchers in ICFP, POPL, and PLDI 2021 onward.

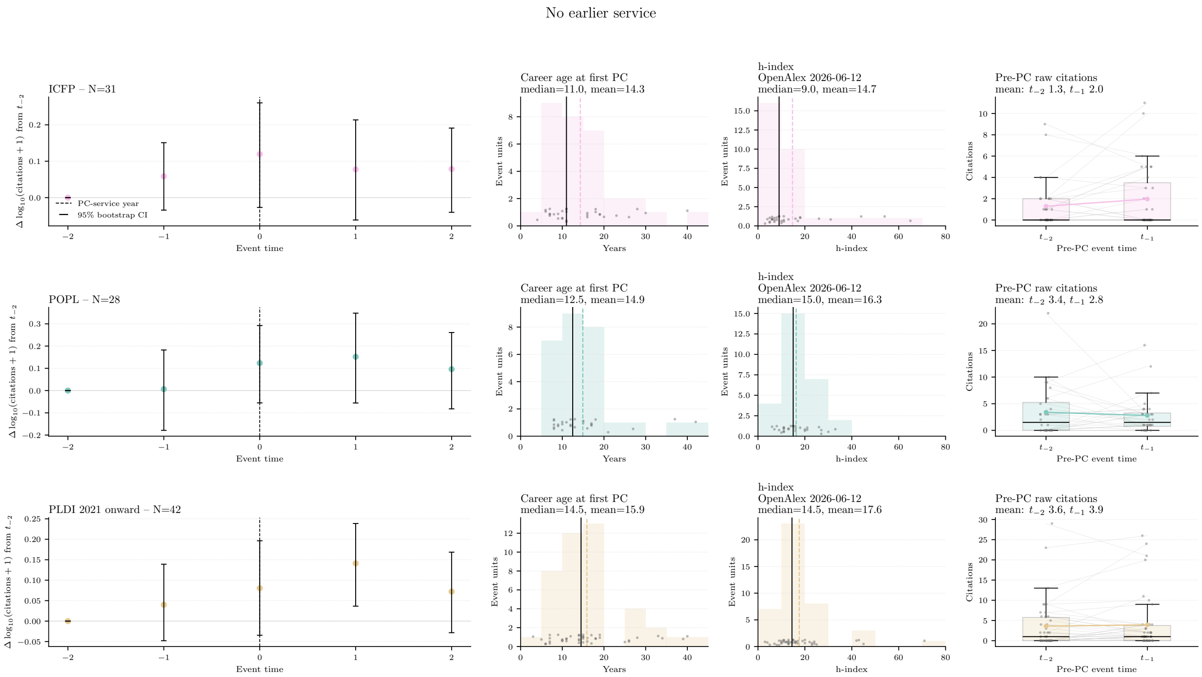
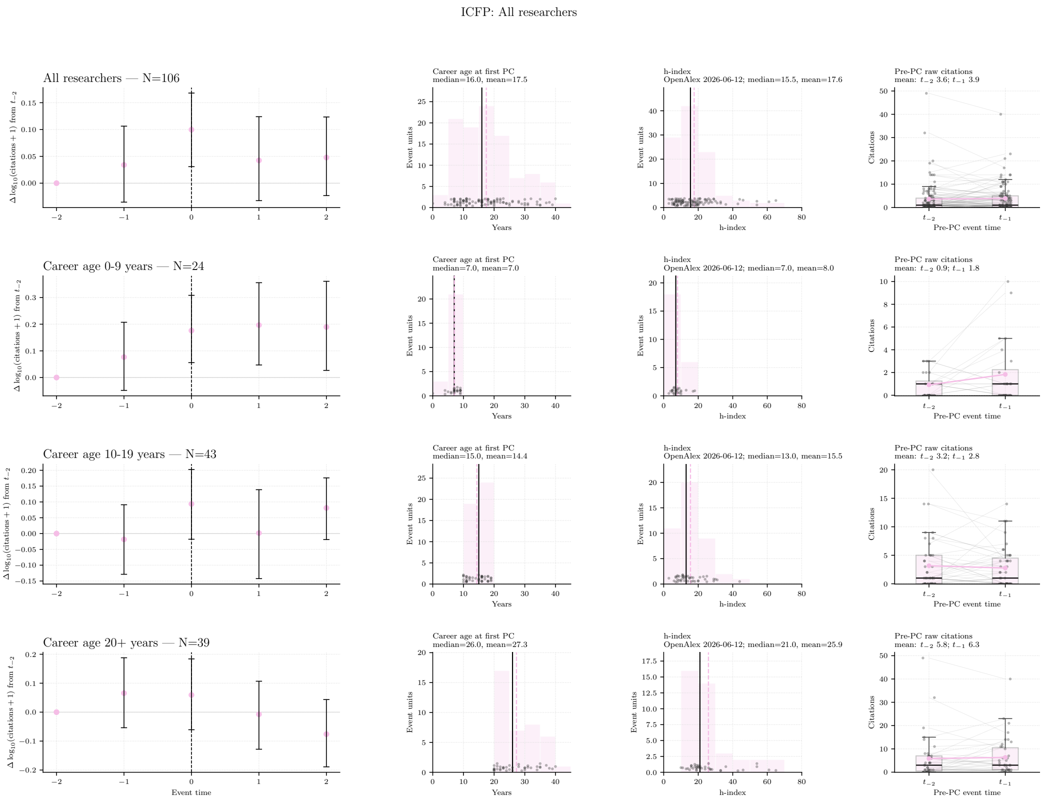
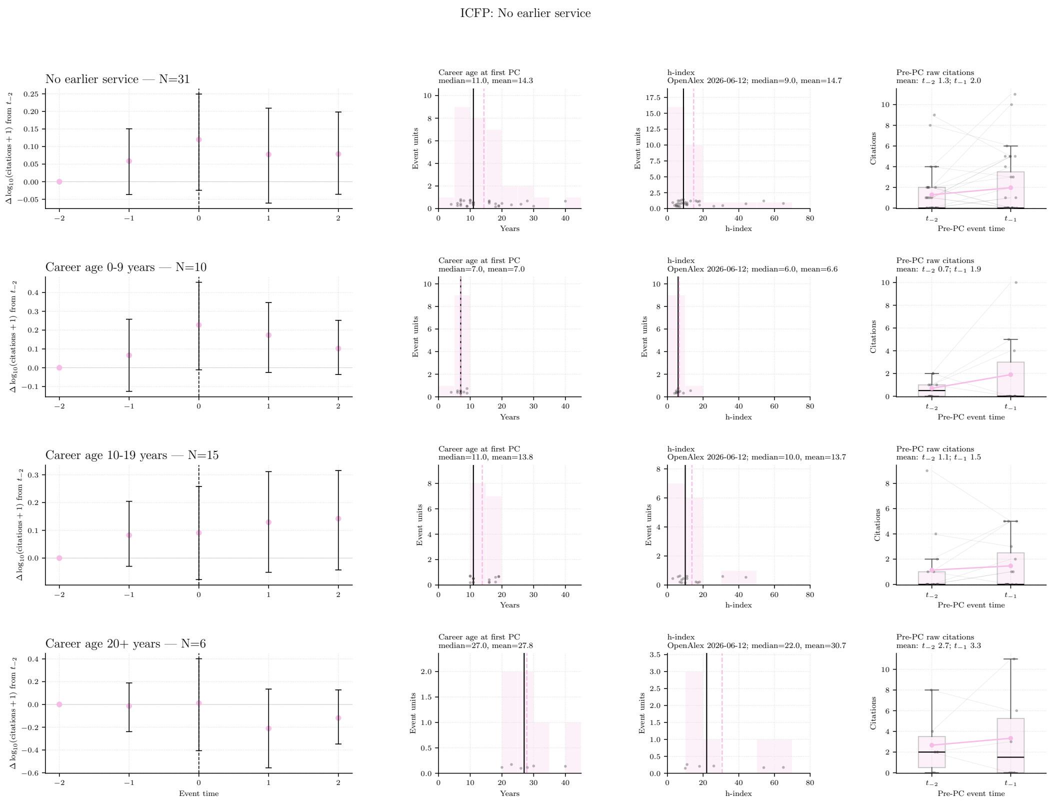


Figure 26: No earlier service comparison for ICFP, POPL, and PLDI 2021 onward.

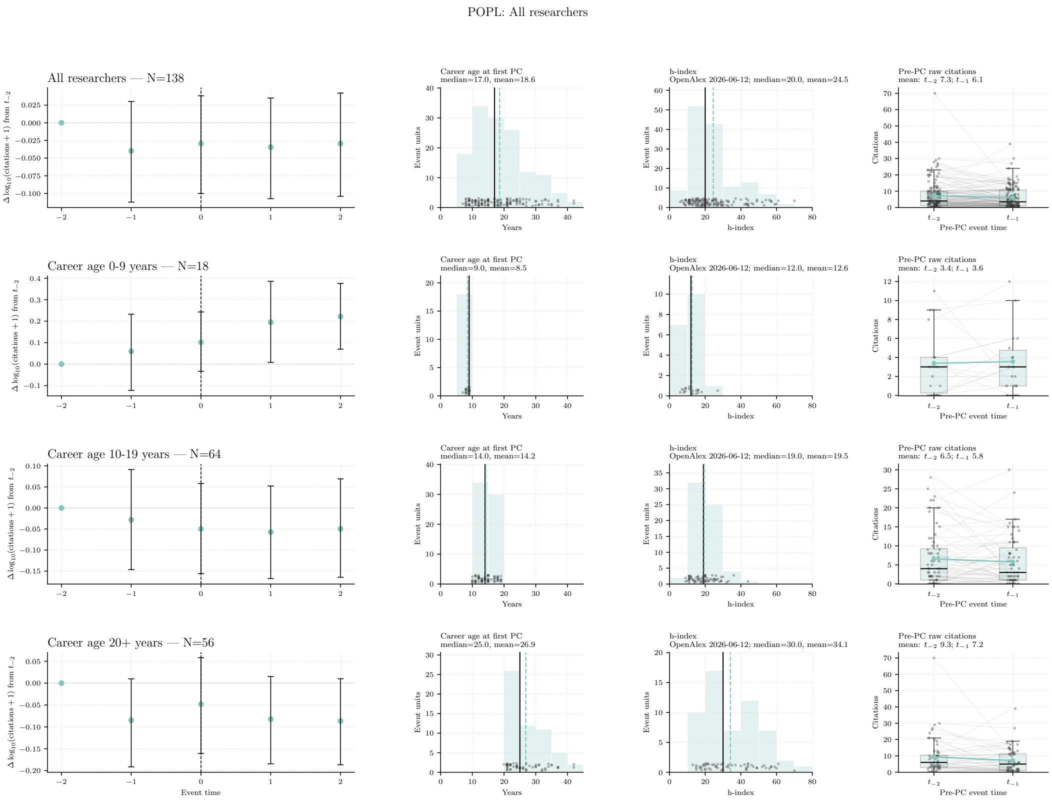


(a) All researchers.

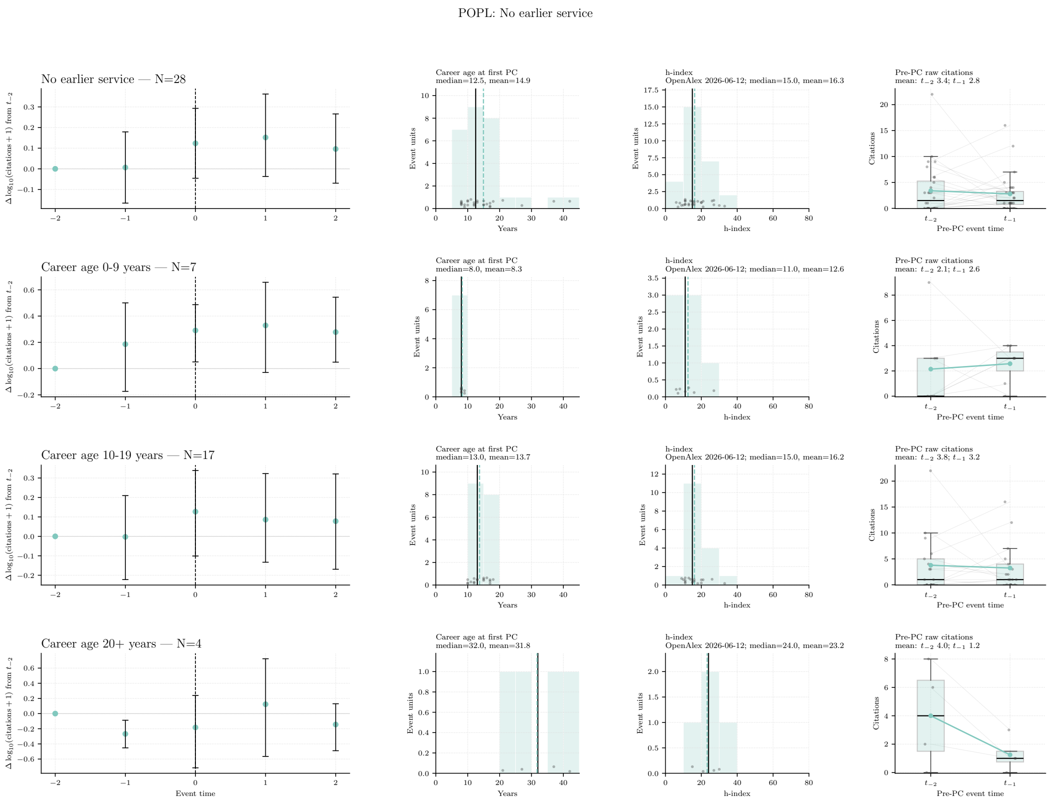


(b) No earlier service.

Figure 27: Detailed ICFP $t = -2$ event study figures.

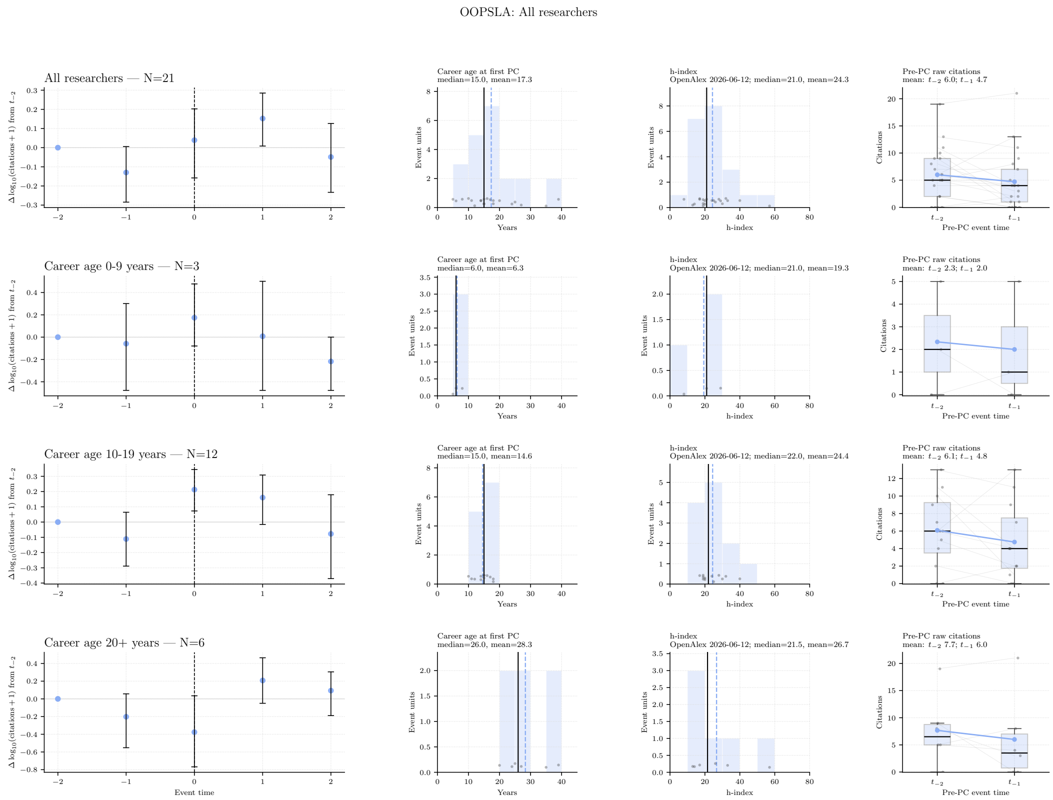


(a) All researchers.

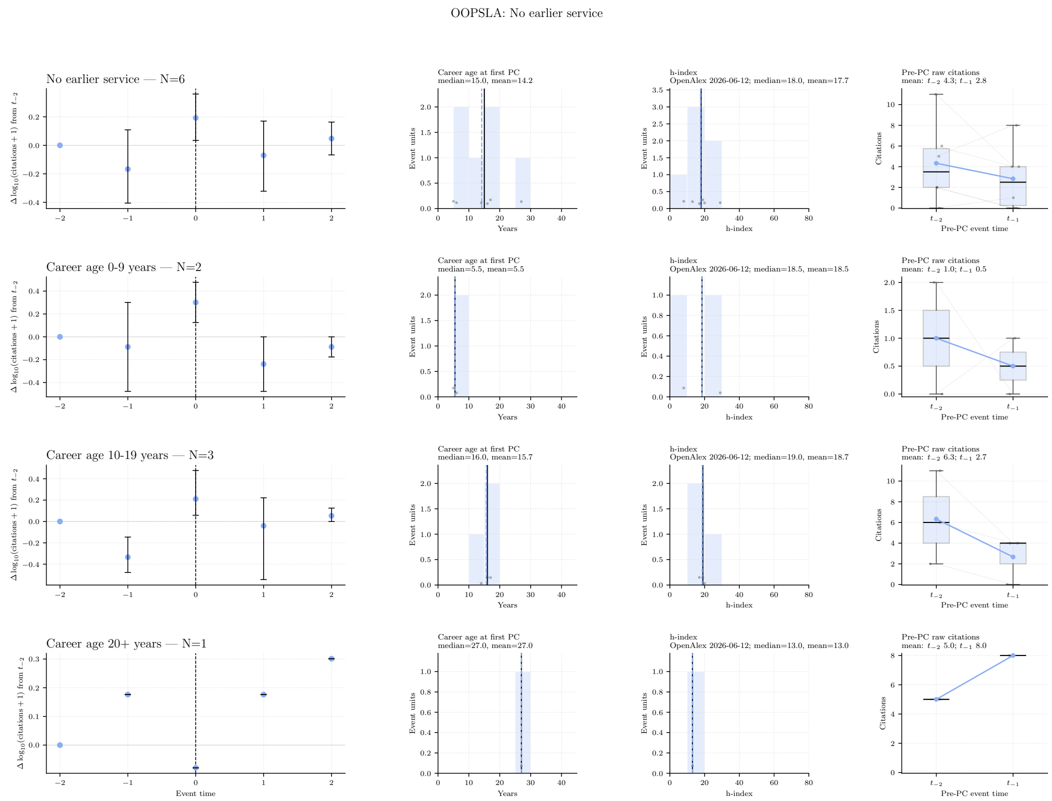


(b) No earlier service.

Figure 28: Detailed POPL $t = -2$ event study figures.

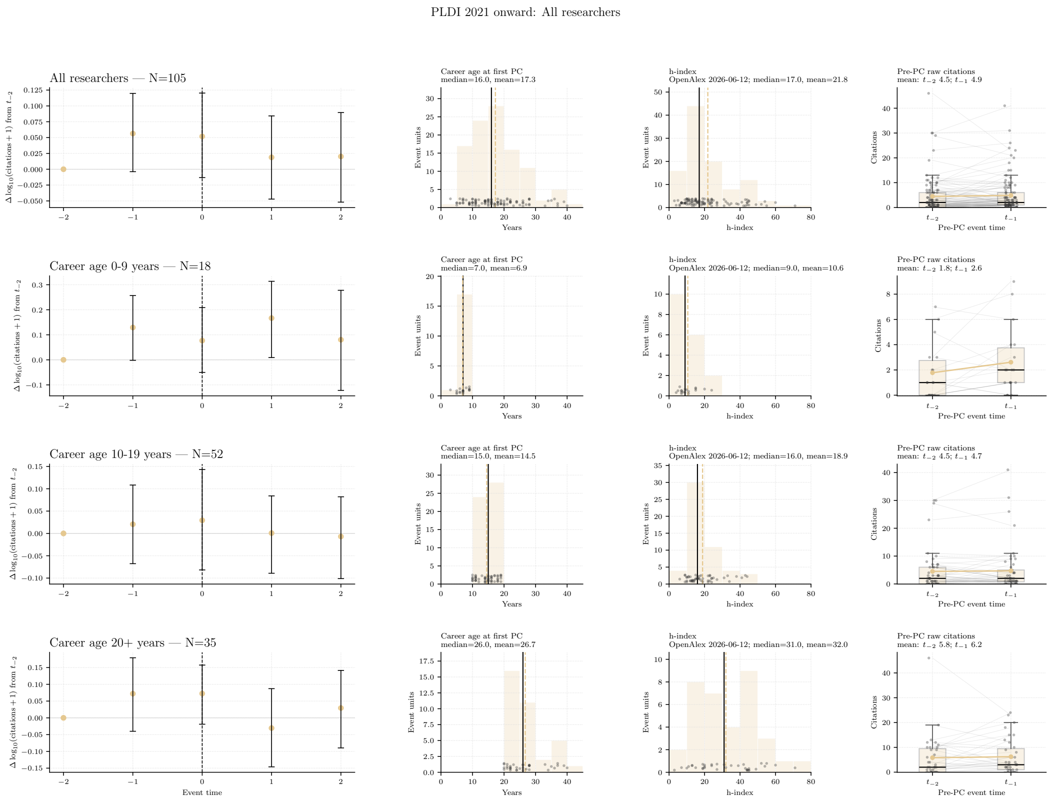


(a) All researchers.

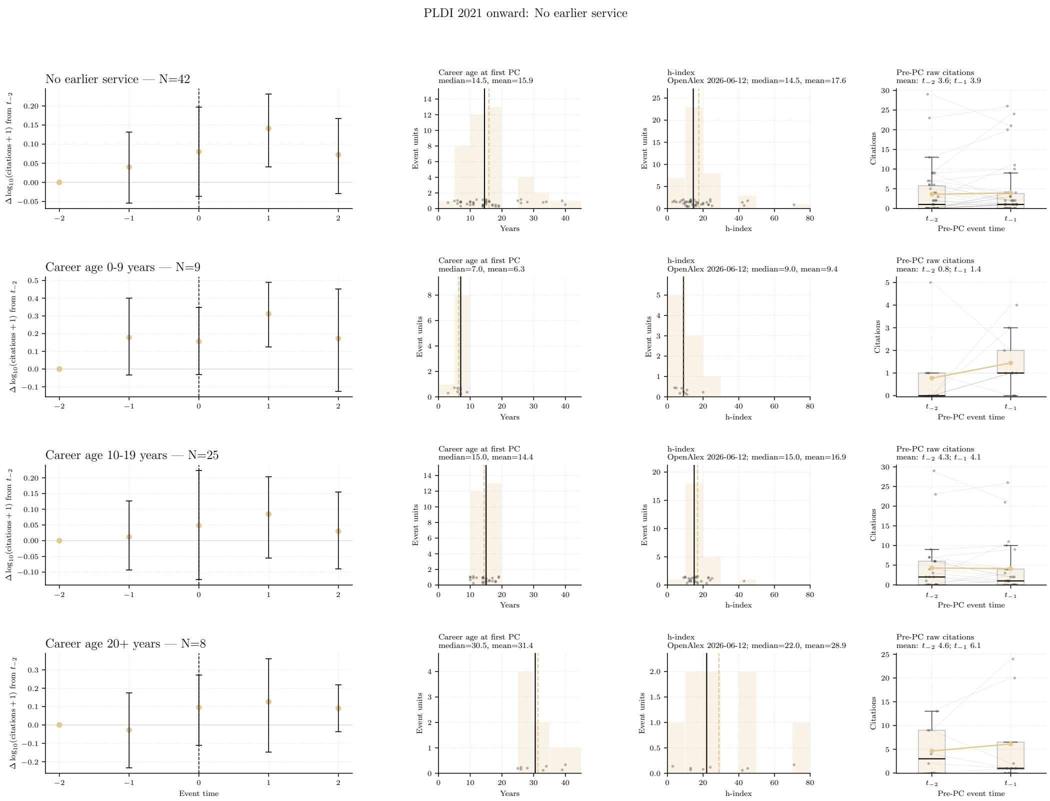


(b) No earlier service.

Figure 29: Detailed OOPSLA $t = -2$ event study figures.

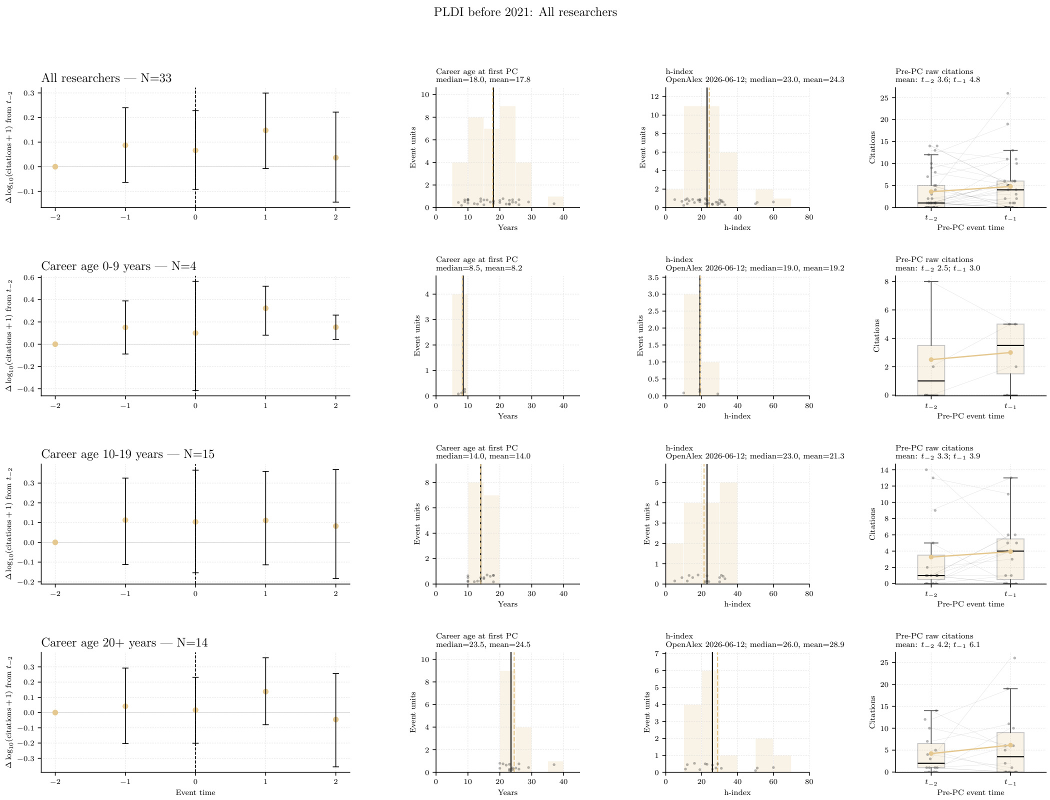


(a) All researchers.

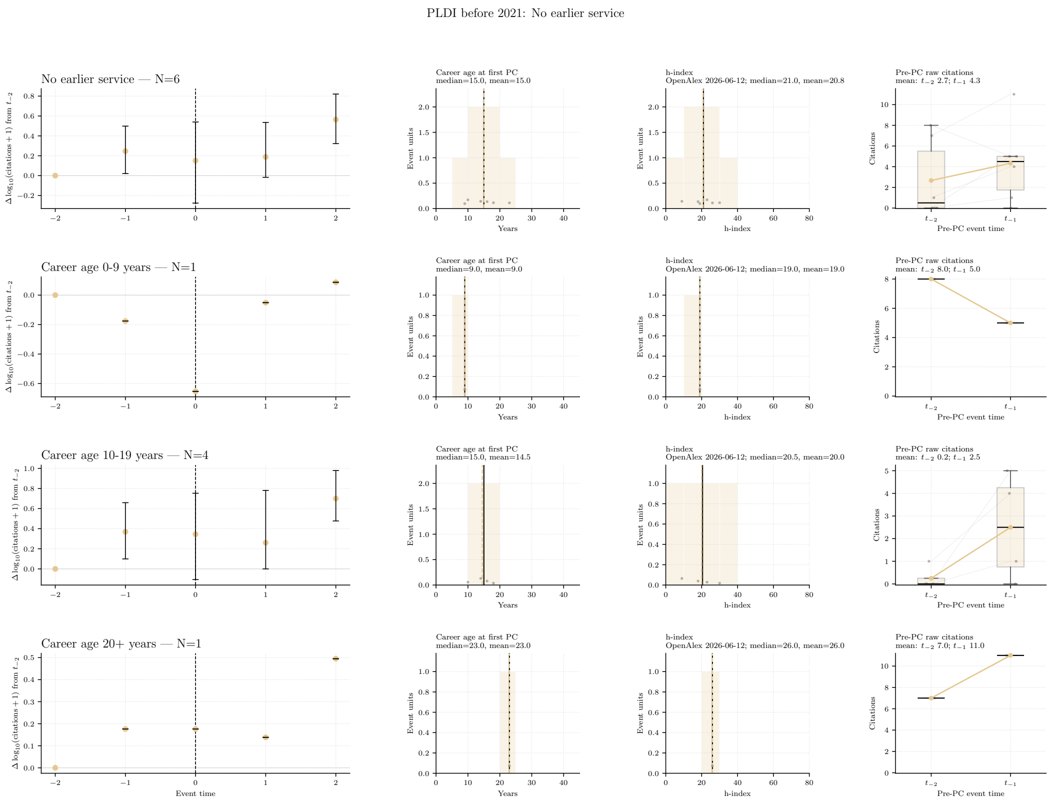


(b) No earlier service.

Figure 30: Detailed PLDI 2021 onward $t = -2$ event study figures.



(a) All researchers.



(b) No earlier service.

Figure 31: Detailed PLDI before 2021 $t = -2$ event study figures.

Alternative Baseline Check

I also check a different baseline before PC service. Instead of using $t = -2$ alone, I use the average of $t = -2$ and $t = -1$ for the same sample. This shows whether the result depends on the baseline definition.

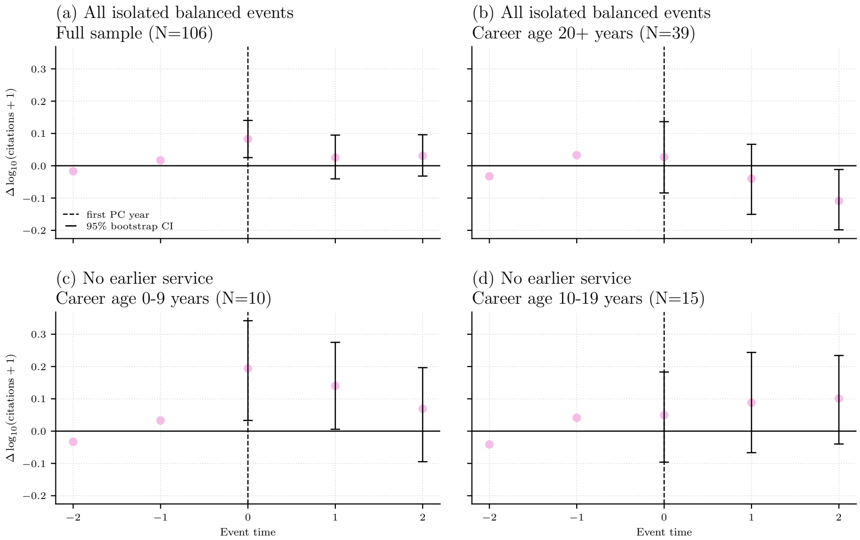


Figure 32: ICFP robustness check with an average baseline before PC service.

Detailed Citation Source Construction & Figures

This analysis uses the same panel with one row per researcher, conference, and year, but replaces the main citation count outcome with citation counts broken down by citing conference. In a panel row (r, c, y), $P_{r,c,y}$ records whether researcher r served on conference c's research paper PC in year y. For any citing conference s, $Y_{r,s,y}$ records citations from accepted papers in conference s, year y, to researcher r's earlier work.

For each researcher and conference pair (r, c):

1. Look across the panel rows for (r, c), and find the years in which $P_{r,c,y} = 1$. These are the observed years in which researcher r is listed on conference c's research paper PC.

2. Keep the pair only if researcher r is observed on conference c's PC exactly once. If the researcher is never observed on conference c's PC, or is observed in multiple PC years in the same conference, drop the pair. Otherwise, let y_0 denote the unique PC service year, and set this year as event time $t = 0$.

3. Use the event window
$$t \in \{-2, -1, 0, +1, +2\}.$$
For each event time t, the corresponding calendar year is $y_0 + t$.

4. Keep the pair only if all five calendar years
$$y_0 - 2, \ldots, y_0 + 2$$
are observed in the panel.

5. Define the service history sample. For the all researcher sample, keep every pair that passes the filters above. For the no earlier service sample, keep only pairs where the available public history checks found no earlier review service evidence in the same conference before y_0. This filter is applied after the requirement of one PC year.

6. For each remaining pair and event time, compute the number of citations coming from the event conference:
$$n_{r,c,t} = Y_{r,c,y_0+t}.$$
This is the number of citations from accepted papers in conference c, in year $y_0 + t$, to researcher r's earlier work.

7. Let $\mathcal{C}$ be the set of conferences in the analysis, and let s index one conference in this set. Compute the total number of citations from all conferences in the analysis:
$$m_{r,c,t} = \sum_{s \in \mathcal{C}} Y_{r,s,y_0+t}.$$

8. Define the event conference citation share:
$$q_{r,c,t} = \frac{n_{r,c,t}}{m_{r,c,t}}$$
whenever $m_{r,c,t} > 0$. If $m_{r,c,t} = 0$, set $q_{r,c,t}$ to missing.

9. If the baseline share $q_{r,c,-2}$ is missing, the event unit is still counted in the sample size, but its source share changes are undefined. This happens when the researcher received no citations from any conference in the analysis in the reference year. I keep these units in the bootstrap sampling frame because they are part of the eligible event window sample. If such a unit is drawn, it contributes no value to the mean for that event time.

10. Use $t = -2$ as the reference year. For each event time, compute the change in citation share within the pair:

$$d_{r,c,t} = 100\,(q_{r,c,t} - q_{r,c,-2}).$$

    This is measured in percentage points. If $q_{r,c,t}$ is missing, then $d_{r,c,t}$ is also missing.

11. Average $d_{r,c,t}$ across the included researcher and conference pairs, separately for each event time:

$$\hat{\delta}_t = \text{mean}\{d_{r,c,t} : d_{r,c,t} \text{ is defined}\}.$$

    By construction, $\hat{\delta}_{-2} = 0$.

12. Compute 95% bootstrap intervals separately at each event time by resampling eligible researcher and conference event units with replacement. If a drawn event unit has undefined $d_{r,c,t}$, it contributes no value to that draw's mean for that event time. The interval is the 2.5th and 97.5th percentiles of the 2,000 bootstrap averages.

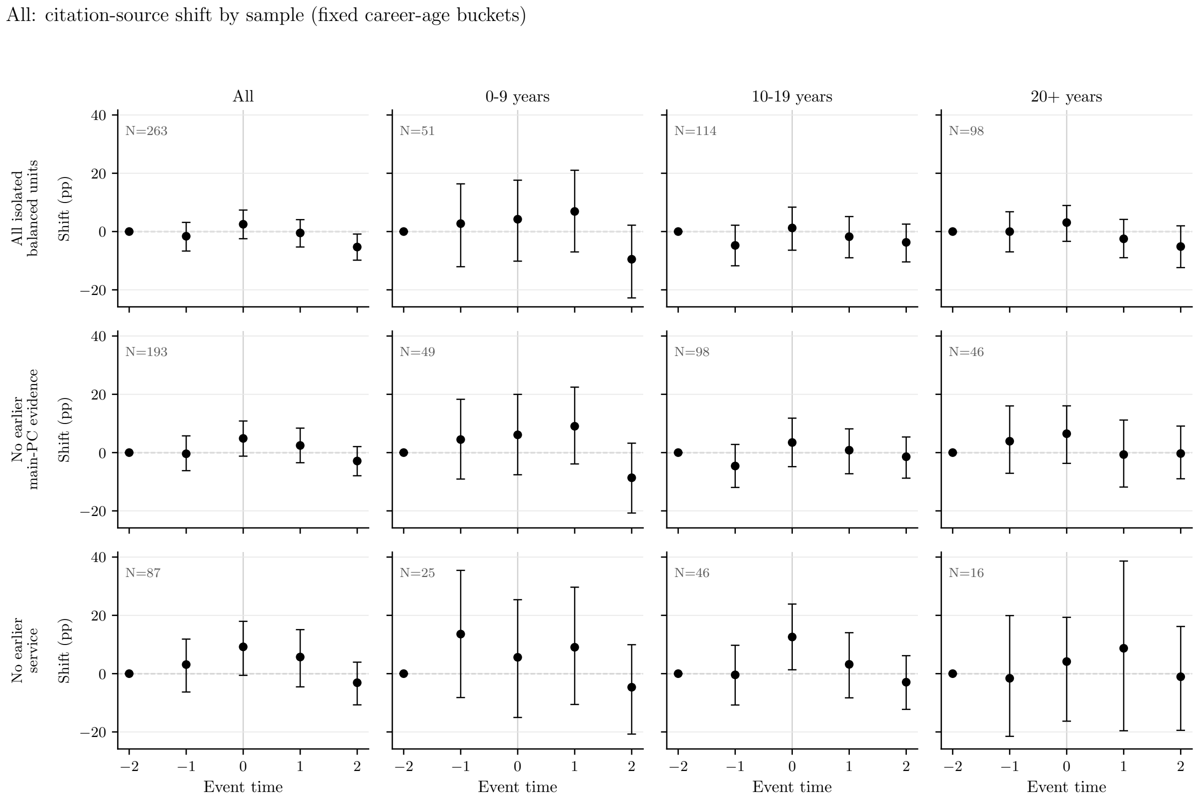


Figure 33: Detailed aggregate citation source figure.

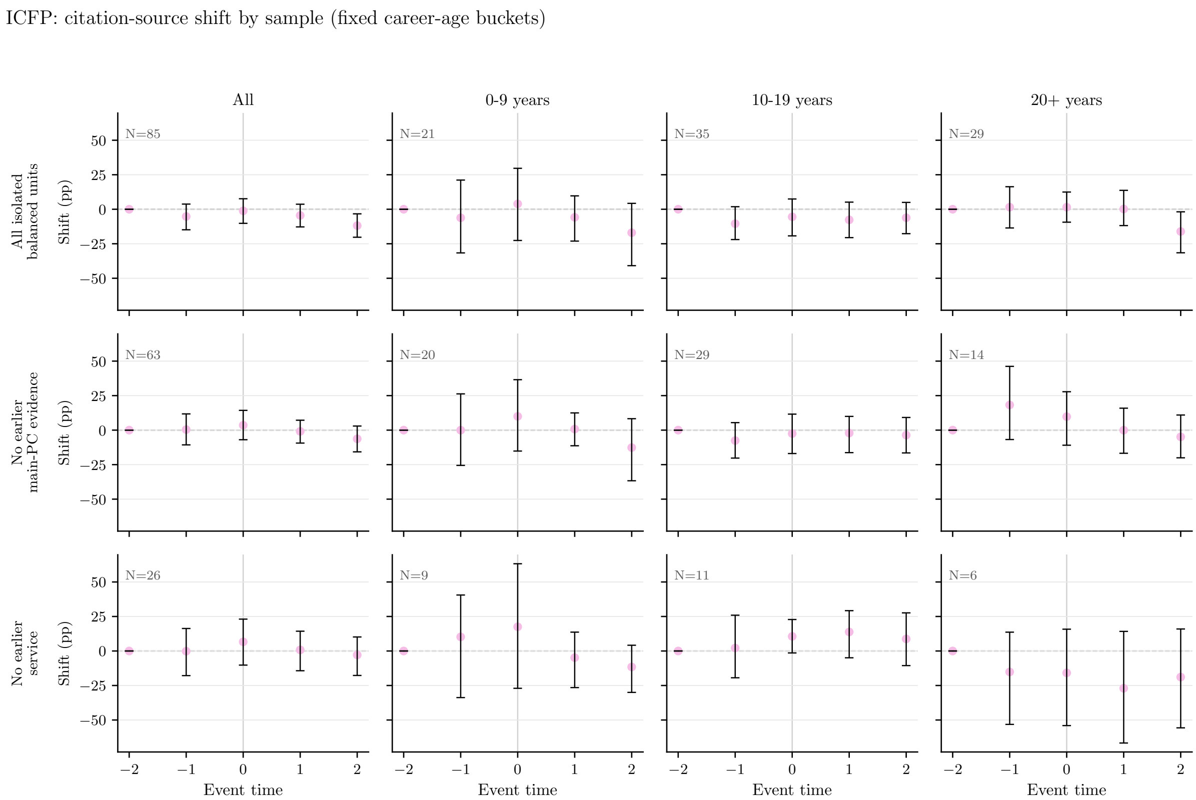


Figure 34: Detailed ICFP citation source figure.

POPL: citation-source shift by sample (fixed career-age buckets)

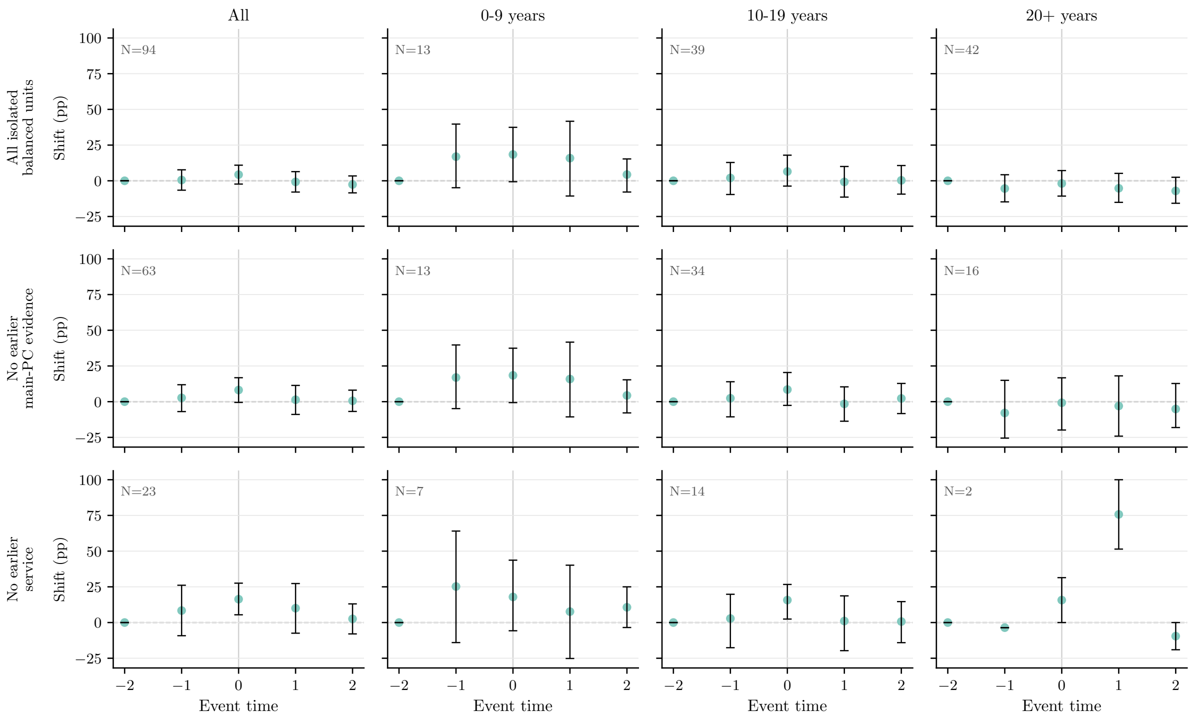


Figure 35: Detailed POPL citation source figure.

OOPSLA: citation-source shift by sample (fixed career-age buckets)

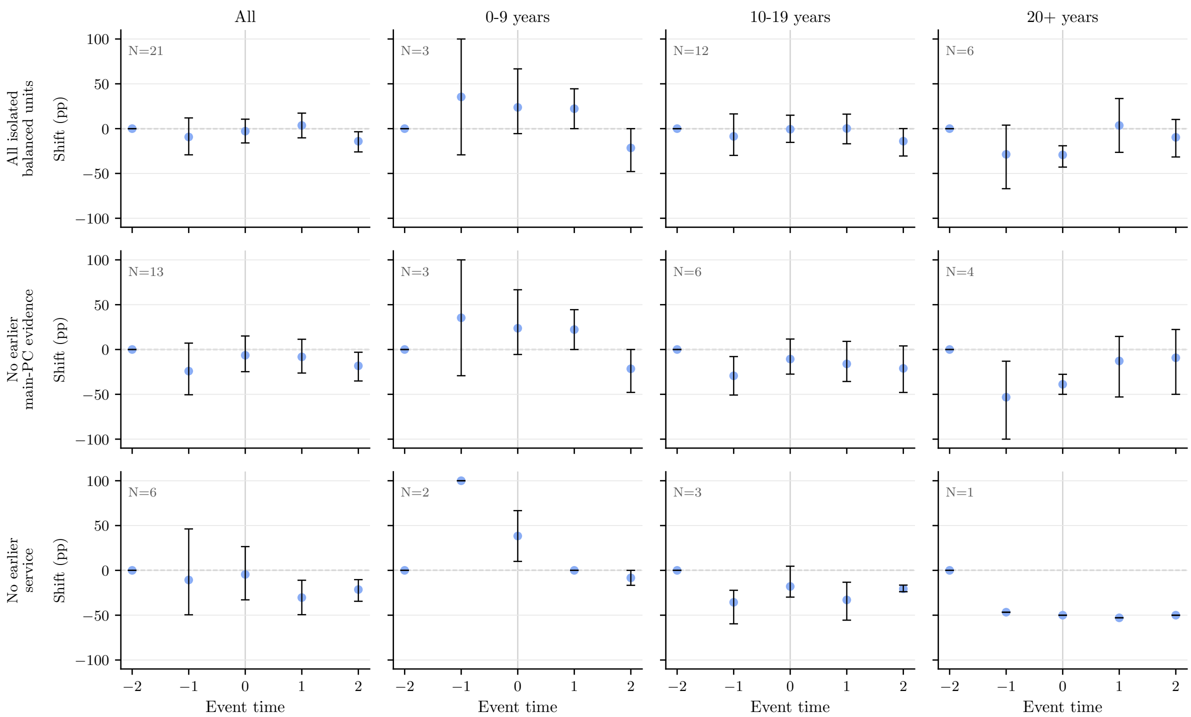


Figure 36: Detailed OOPSLA citation source figure.

PLDI 2021 onward: citation-source shift by sample (fixed career-age buckets)

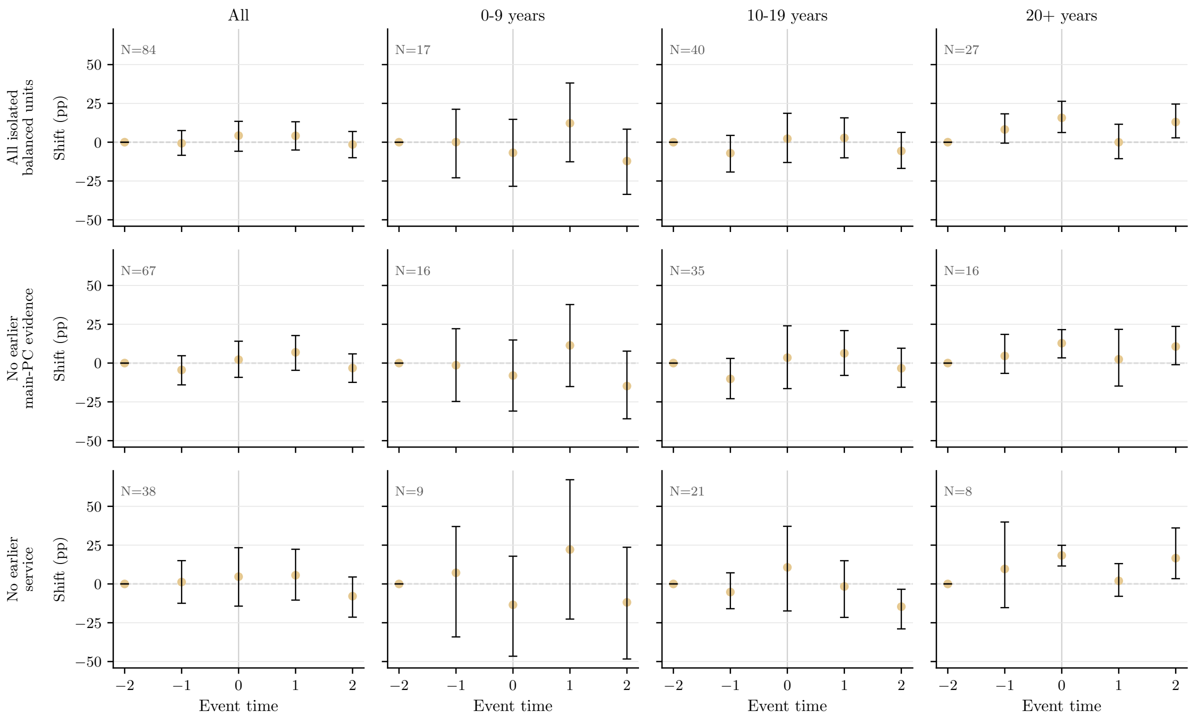


Figure 37: Detailed PLDI 2021 onward citation source figure.

PLDI before 2021: citation-source shift by sample (fixed career-age buckets)

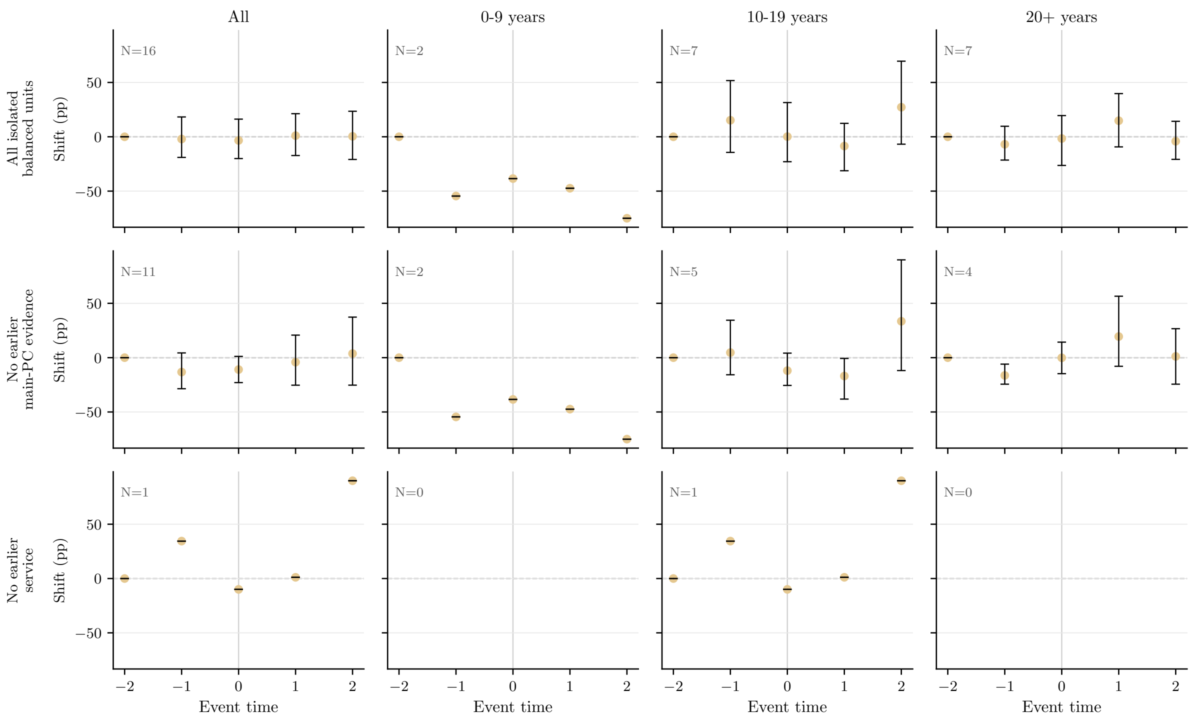


Figure 38: Detailed PLDI before 2021 citation source figure.

Citation Distributions for Years: Additional Cases

Figure 39: Selected citation source concentration examples.

Selected PC Member Examples

Table 12 lists the accepted papers behind the PC-year citations for each selected PC service case. Each block is one selected case, and the rows under it are accepted papers from that conference year that cite the researcher's earlier work. In the summary counts, R is the total number of OpenAlex reference links, T is the same count after combining duplicate cited titles, S is the number of self citation links, and C is the number of links from papers that include a coauthor of the selected researcher.

Table 12: Accepted papers behind the selected PC-year citations.

Researcher	PC	Citing paper	Citing authors	R	T	S	C	Cited prior work(s)
Clément Pit-Claudel, PLDI 2024: R=13, T=13, citing papers=9, S=2, C=7								
Clément Pit-Claudel	PLDI 2024	A Verified Compiler for a Functional Tensor Language	Amanda Liu; Gilbert Bernstein; Adam Chlipala; Jonathan Ragan-Kelley	3	3	0	3	Extensible Extraction of Efficient Imperative Programs with Foreign Functions, Manually Managed Memory, and Proofs (2020; E=1); Fiat (2014; E=1); Relational compilation for performance-critical applications: extensible proof-producing translation of functional models into low-level code (2022; E=1)
Clément Pit-Claudel	PLDI 2024	Daedalus: Safer Document Parsing	Iavor S. Diatchki; Mike Dodds; Harrison Goldstein; Bill Harris; David A. Holland; Benoit Razet; Cole Schlesinger; Simon Winwood	1	1	0	0	Narcissus: correct-by-construction derivation of decoders and encoders from binary formats (2019; E=1)
Clément Pit-Claudel	PLDI 2024	Foundational Integration Verification of a Cryptographic Server	Andres Erbsen; Jade Philipoom; Dustin Jamner; Ashley Lin; Samuel Gruetter; Clement Pit-Claudel; Adam Chlipala	2	2	2	2	Extensible Extraction of Efficient Imperative Programs with Foreign Functions, Manually Managed Memory, and Proofs (2020; E=1); Relational compilation for performance-critical applications: extensible proof-producing translation of functional models into low-level code (2022; E=1)
Clément Pit-Claudel	PLDI 2024	IsoPredict: Dynamic Predictive Analysis for Detecting Unserializable Behaviors in Weakly Isolated Data Store Applications	Chujun Geng; Spyros Blanas; Michael D. Bond; Yang Wang	1	1	0	0	Trigger Selection Strategies to Stabilize Program Verifiers (2016; E=1)
Clément Pit-Claudel	PLDI 2024	Live Verification in an Interactive Proof Assistant	Samuel Gruetter; Viktor Fukala; Adam Chlipala	2	2	0	2	Fiat (2014; E=1); Relational compilation for performance-critical applications: extensible proof-producing translation of functional models into low-level code (2022; E=1)
Clément Pit-Claudel	PLDI 2024	Modular Hardware Design of Pipelined Circuits with Hazards	Minseong Jang; J. G. Rhee; Woojin Lee; Shuangshuang Zhao; Jeehoon Kang	1	1	0	0	The essence of Bluespec: a core language for rule-based hardware design (2020; E=1)
Clément Pit-Claudel	PLDI 2024	The Functional Essence of Imperative Binary Search Trees	Anton Lorenzen; Daan Leijen; Wouter Swierstra; Sam Lindley	1	1	0	0	Relational compilation for performance-critical applications: extensible proof-producing translation of functional models into low-level code (2022; E=1)
Clément Pit-Claudel	PLDI 2024	Verified Extraction from Coq to OCaml	Yannick Forster; Matthieu Sozeau; Nicolas Tabareau	1	1	0	0	Relational compilation for performance-critical applications: extensible proof-producing translation of functional models into low-level code (2022; E=1)
Clément Pit-Claudel	PLDI 2024	Wavefront Threading Enables Effective High-Level Synthesis	Blake Pelton; Adam Sapek; Ken Eguro; Daniel Lo; Alessandro Forin; Matt Humphrey; Jin-wen Xi; D. J. Cox; Rajas H. Karandikar; Johannes de Fine Licht; Evgeny Babin; Adrian M. Caulfield; Doug Burger	1	1	0	0	The essence of Bluespec: a core language for rule-based hardware design (2020; E=1)
Clément Pit-Claudel, OOPSLA 2025: R=9, T=9, citing papers=8, S=0, C=1								
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA1] A Mechanized Semantics for Dataflow Circuits	Tony Law; Delphine Demange; Sandrine Blazy	1	1	0	0	The essence of Bluespec: a core language for rule-based hardware design (2020; E=1)

Continued on next page

Table 12: Accepted papers behind the selected PC-year citations, continued.

Researcher	PC	Citing paper	Citing authors	R	T	S	C	Cited prior work(s)
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA1] Efficient Algorithms for the Uniform Tokenization Problem	Wu Angela Li; Konstantinos Mamouras	1	1	0	0	Linear Matching of JavaScript Regular Expressions (2024; E=1)
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA1] QED in Context: An Observation Study of Proof Assistant Users	Jessica Shi; Cassia Torczon; Harrison Goldstein; Benjamin C. Pierce; Andrew Head	1	1	0	0	Untangling mechanized proofs (2020; E=1)
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA2] Certified Decision Procedures for Width-Independent Bitvector Predicates	Siddharth Bhat; Leo Stefanesco; Chris Hughes; Tobias Grosser	1	1	0	0	The essence of Bluespec: a core language for rule-based hardware design (2020; E=1)
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA2] Incremental Certified Programming	Tomas Diaz; Kenji Maillard; Nicolas Tabareau; Eric Tanter	2	2	0	0	Fiat (2014; E=1); Incremental Proof Development in Dafny with Module-Based Induction (2024; E=1)
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA2] Interactive Bitvector Reasoning using Verified Bit-Blasting	Henrik Boving; Siddharth Bhat; Luisa Cicolini; Alex Keizer; Leon Frenot; Abdalrhman Mohamed; Leo Stefanesco; Harun Khan; Joshua Clune; Clark Barrett; Tobias Grosser	1	1	0	0	The essence of Bluespec: a core language for rule-based hardware design (2020; E=1)
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA2] On the Impact of Formal Verification on Software Development	Eric Mugnier; Yuanyuan Zhou; Ranjit Jhala; Michael Coblenz	1	1	0	0	Formally Verified Cloud-Scale Authorization (2025; E=1)
Clément Pit-Claudel	OOPSLA 2025	[OOPSLA2] We've Got You Covered: Type-Guided Repair of Incomplete Input Generators	Patrick LaFontaine; Z. Zhou; Ashish Mishra; Suresh Jagannathan; Benjamin Delaware	1	1	0	1	Fiat (2014; E=1)
Viktor Kunčak, OOPSLA 2020: R=20, T=19, citing papers=11, S=0, C=1								
Viktor Kunčak	OOPSLA 2020	Digging for fold: synthesis-aided API discovery for Haskell	Michael B. James; Zheng Guo; Ziteng Wang; Shivani Doshi; Hila Peleg; Ranjit Jhala; Nadia Polikarpova	1	1	0	0	Complete completion using types and weights (2013; E=1)
Viktor Kunčak	OOPSLA 2020	Enabling accuracy-aware Quantum compilers using symbolic resource estimation	Giulia Meuli; Mathias Soeken; Martin Roetteler; Thomas Haner	1	1	0	0	Sound compilation of reals (2014; E=1)
Viktor Kunčak	OOPSLA 2020	Feedback-driven semi-supervised synthesis of program transformations	Xiang Gao; Shraddha Barke; Arjun Radhakrishna; Gustavo Soares; Sumit Gulwani; Alan Leung; Nachiappan Nagappan; Ashish Tiwari	2	2	0	0	Complete completion using types and weights (2013; E=1); Counterexample-Guided Quantifier Instantiation for Synthesis in SMT (2015; E=1)
Viktor Kunčak	OOPSLA 2020	Formulog: Datalog for SMT-based static analysis	Aaron Bembenek; Michael Greenberg; Stephen Chong	1	1	0	0	Scala to the Power of Z3: Integrating SMT and Programming (2011; E=1)
Viktor Kunčak	OOPSLA 2020	Gradual verification of recursive heap data structures	Jenna Wise; Johannes Bader; Cameron Wong; Jonathan Aldrich; Eric Tanter; Joshua Sunshine	1	1	0	0	Runtime Checking for Separation Logic (2008; E=1)
Viktor Kunčak	OOPSLA 2020	Guiding dynamic programing via structural probability for accelerating programming by example	Ruyi Ji; Yican Sun; Yingfei Xiong; Zhenjiang Hu	2	2	0	0	Counterexample-Guided Quantifier Instantiation for Synthesis in SMT (2015; E=1); Refutation-based synthesis in SMT (2017; E=1)
Viktor Kunčak	OOPSLA 2020	Just-in-time learning for bottom-up enumerative synthesis	Shraddha Barke; Hila Peleg; Nadia Polikarpova	4	3	0	0	Complete completion using types and weights (2013; E=1); On Repair with Probabilistic Attribute Grammars (2017; E=1); Synthesis modulo recursive functions (2013; E=2)
Viktor Kunčak	OOPSLA 2020	On the unusual effectiveness of type-aware operator mutations for testing SMT solvers	Dominik Winterer; Chengyu Zhang; Zhendong Su	2	2	0	0	Counterexample-Guided Quantifier Instantiation for Synthesis in SMT (2015; E=1); On Counterexample Guided Quantifier Instantiation for Synthesis in CVC4 (2015; E=1)
Viktor Kunčak	OOPSLA 2020	Programming with a read-eval-synth loop	Hila Peleg; Roi Gabay; Shachar Itzhaky; Eran Yahav	2	2	0	0	Bidirectional evaluation with direct manipulation (2018; E=1); Deductive Program Repair (2015; E=1)

Continued on next page

Table 12: Accepted papers behind the selected PC-year citations, continued.

Researcher	PC	Citing paper	Citing authors	R	T	S	C	Cited prior work(s)
Viktor Kunčak	OOPSLA 2020	Statically verified refinements for multiparty protocols	Fangyi Zhou; Francisco Ferreira; Raymond Hu; Rumyana Neykova; Nobuko Yoshida	1	1	0	0	SMT-based checking of predicate-qualified types for Scala (2016; E=1)
Viktor Kunčak	OOPSLA 2020	Unifying execution of imperative generators and declarative specifications	Pengyu Nie; Marinela Parovic; Zhiqiang Zang; Sarfraz Khurshid; Aleksandar Milicevic; Milos Gligoric	3	3	0	1	Executing Specifications Using Synthesis and Constraint Solving (2013; E=1); Programming with enumerable sets of structures (2015; E=1); Test generation through programming in UDITA (2010; E=1)
Viktor Kunčak, PLDI 2021: R=18, T=14, citing papers=6, S=0, C=0								
Viktor Kunčak	PLDI 2021	Beyond the elementary representations of program invariants over algebraic data types	Yurii Kostyukov; Dmitry Mordvinov; Grigory Fedyukovich	2	1	0	0	Decision procedures for algebraic data types with abstractions (2010; E=2)
Viktor Kunčak	PLDI 2021	CoStar: a verified ALL(*) parser	Sam Lasser; Chris Casinghino; Kathleen Fisher; Cody Roux	2	2	0	0	LL(1) Parsing with Derivatives and Zippers (2019; E=1); Zippy LL(1) parsing with derivatives (2020; E=1)
Viktor Kunčak	PLDI 2021	Cyclic program synthesis	Shachar Itzhaky; Hila Peleg; Nadia Polikarpova; Reuben N. S. Rowe; Ilya Sergey	7	5	0	0	Complete functional synthesis (2010; E=2); Counterexample-Guided Quantifier Instantiation for Synthesis in SMT (2015; E=1); Deductive Program Repair (2015; E=1); Refutation-based synthesis in SMT (2017; E=1); Synthesis modulo recursive functions (2013; E=2)
Viktor Kunčak	PLDI 2021	Example-guided synthesis of relational queries	Aalok Thakkar; Aaditya Naik; Nathaniel Sands; Rajeev Alur; Mayur Naik; Mukund Raghothaman	1	1	0	0	Counterexample-Guided Quantifier Instantiation for Synthesis in SMT (2015; E=1)
Viktor Kunčak	PLDI 2021	Specification synthesis with constrained Horn clauses	Sumanth Prabhu; Grigory Fedyukovich; Kumar Madhukar; Deepak D'Souza	1	1	0	0	A Verification Toolkit for Numerical Transition Systems (2012; E=1)
Viktor Kunčak	PLDI 2021	Synthesizing data structure refinements from integrity constraints	Shankara Pailoor; Yuepeng Wang; Xinyu Wang; Isil Dillig	5	4	0	0	Decision procedures for algebraic data types with abstractions (2010; E=2); Generalized Typestate Checking for Data Structure Consistency (2005; E=1); Hob: A Tool for Verifying Data Structure Consistency (2005; E=1); On Repair with Probabilistic Attribute Grammars (2017; E=1)
Peter Thiemann, ICFP 2023: R=8, T=8, citing papers=3, S=4, C=4								
Peter Thiemann	ICFP 2023	Combinator-Based Fixpoint Algorithms for Big-Step Abstract Interpreters	Sven Keidel; Sebastian Erdweg; Tobias Hombucher	1	1	0	0	Type Analysis for JavaScript (2009; E=1)
Peter Thiemann	ICFP 2023	Dependent Session Protocols in Separation Logic from First Principles (Functional Pearl)	Jules Jacobs; Jonas Kastberg Hinrichsen; Robbert Krebbers	3	3	0	0	Duality of Session Types: The Final Cut (2020; E=1); Polymorphic Typestate for Session Types (2022; E=1); Relating Functional and Imperative Session Types (2022; E=1)
Peter Thiemann	ICFP 2023	Intrinsically Typed Sessions with Callbacks (Functional Pearl)	Peter Thiemann	4	4	4	4	An Implementation of Session Types (2004; E=1); Combinators for program generation (1999; E=1); Context-free session types (2016; E=1); Polymorphic lambda calculus with context-free session types (2022; E=1)
Simon J. Gay, ICFP 2023: R=11, T=11, citing papers=4, S=2, C=2								
Simon J. Gay	ICFP 2023	Dependent Session Protocols in Separation Logic from First Principles (Functional Pearl)	Jules Jacobs; Jonas Kastberg Hinrichsen; Robbert Krebbers	3	3	0	0	Duality of Session Types: The Final Cut (2020; E=1); Linear type theory for asynchronous session types (2009; E=1); Subtyping for session types in the pi calculus (2005; E=1)
Simon J. Gay	ICFP 2023	HasChor: Functional Choreographic Programming for All (Functional Pearl)	Gan Shen; Shun Kashiwa; Lindsey Kuper	1	1	0	0	Behavioral Types in Programming Languages (2016; E=1)
Simon J. Gay	ICFP 2023	Intrinsically Typed Sessions with Callbacks (Functional Pearl)	Peter Thiemann	5	5	0	0	Behavioral Types in Programming Languages (2016; E=1); Linear type theory for asynchronous session types (2009; E=1); Modular session types for objects (2015; E=1); Type checking a multithreaded functional language with session types (2006; E=1); Typechecking protocols with Mungo and StMungo: A session type toolchain for Java (2017; E=1)

Continued on next page

Table 12: Accepted papers behind the selected PC-year citations, continued.

Researcher	PC	Citing paper	Citing authors	R	T	S	C	Cited prior work(s)
Simon J. Gay	ICFP 2023	Special Delivery: Programming with Mailbox Types	Simon Fowler; Duncan Paul Attard; Franciszek Sowul; Simon J. Gay; Phil Trinder	2	2	2	2	Behavioral Types in Programming Languages (2016; E=1); Linear type theory for asynchronous session types (2009; E=1)
Martin Elsman, ICFP 2024: R=8, T=8, citing papers=4, S=4, C=4								
Martin Elsman	ICFP 2024	A Safe Low-Level Language for Computer Algebra and Its Formally Verified Compiler	Guillaume Melquiond; Josue Moreau	1	1	0	0	Towards size-dependent types for array programming (2021; E=1)
Martin Elsman	ICFP 2024	Double-Ended Bit-Stealing for Algebraic Data Types	Martin Elsman	4	4	4	4	Combining region inference and garbage collection (2002; E=1); Integrating region memory management and tag-free generational garbage collection (2021; E=1); Polymorphic equality – No tags required (1998; E=1); Static interpretation of modules (1999; E=1)
Martin Elsman	ICFP 2024	Oxidizing OCaml with Modal Memory Management	Anton Lorenzen; Leo White; Stephen Dolan; Richard A. Eisenberg; Sam Lindley	2	2	0	0	A Retrospective on Region-Based Memory Management (2004; E=1); On the Effects of Integrating Region-Based Memory Management and Generational Garbage Collection in ML (2020; E=1)
Martin Elsman	ICFP 2024	Staged Compilation with Module Functors	Tsung-Ju Chiang; Jeremy Yallop; Leo White; Ningning Xie	1	1	0	0	Static interpretation of modules (1999; E=1)
Ezgi Çiçek, ICFP 2019: R=7, T=5, citing papers=2, S=0, C=5								
Ezgi Çiçek	ICFP 2019	Closure conversion is safe for space	Zoe Paraskevopoulou; Andrew W. Appel	3	3	0	1	A type theory for incremental computational complexity with control flow changes (2016; E=1); Refinement Types for Incremental Computational Complexity (2015; E=1); Relational cost analysis (2016; E=1)
Ezgi Çiçek	ICFP 2019	Relational cost analysis for functional-imperative programs	Weihao Qu; Marco Gaboardi; Deepak Garg	4	2	0	4	A type theory for incremental computational complexity with control flow changes (2016; E=2); Relational cost analysis (2016, 2017; E=2)
Brandon Lucia, OOPSLA 2019: R=24, T=23, citing papers=5, S=13, C=14								
Brandon Lucia	OOPSLA 2019	CLOTHO: directed test generation for weakly consistent database systems	Kia Rahmani; Kartik Nagar; Benjamin Delaware; Suresh Jagannathan	2	2	0	0	ColorSafe (2010; E=1); Concurrency debugging with differential schedule projections (2015; E=1)
Brandon Lucia	OOPSLA 2019	Dependence-aware, unbounded sound predictive race detection	Kaan Genc; Jake Roemer; XU Yu-fan; Michael D. Bond	3	3	0	1	Conflict exceptions (2010; E=1); IFRit (2012; E=1); Valor: efficient, software-only region conflict exceptions (2015; E=1)
Brandon Lucia	OOPSLA 2019	Efficient lock-free durable sets	Yoav Zuriel; Michal Friedman; Gali Sheffi; Nachshon Cohen; Erez Petrank	5	5	0	0	Adaptive dynamic checkpointing for safe efficient intermittent computing (2018; E=1); Alpaca: intermittent execution without checkpoints (2017; E=1); Chain: tasks and channels for reliable intermittent programs (2016; E=1); Intermittent Computing: Challenges and Opportunities (2017; E=1); Transactional concurrency control for intermittent, energy-harvesting computing systems (2019; E=1)
Brandon Lucia	OOPSLA 2019	I/O dependent idempotence bugs in intermittent systems	Milijana Surbatovich; Limin Jia; Brandon Lucia	13	12	13	13	A Reconfigurable Energy Storage Architecture for Energy-harvesting Devices (2018; E=1); A simpler, safer programming and execution model for intermittent systems (2015; E=1); Alpaca: intermittent execution without checkpoints (2017; E=1); An Energy-interference-free Hardware-Software Debugger for Intermittent Energy-harvesting Systems (2016; E=1); Chain: tasks and channels for reliable intermittent programs (2016; E=1); ColorSafe (2010; E=1); Finding concurrency bugs with context-aware communication graphs (2009; E=1); Isolating and understanding concurrency errors using reconstructed execution fragments (2011; E=2); Production-guided concurrency debugging (2016; E=1); Supporting peripherals in intermittent systems with just-in-time checkpoints (2019; E=1); Termination checking and task decomposition for task-based intermittent programs (2018; E=1); Transactional concurrency control for intermittent, energy-harvesting computing systems (2019; E=1)

Continued on next page

Table 12: Accepted papers behind the selected PC-year citations, continued.

Researcher	PC	Citing paper	Citing authors	R	T	S	C	Cited prior work(s)
Brandon Lucia	OOPSLA 2019	Trace aware random testing for distributed systems	Burcu Kulahcioglu Ozkan; Rupak Majumdar; Simin Oraee	1	1	0	0	Finding concurrency bugs with context-aware communication graphs (2009; E=1)
Jonathan Immanuel Brachthäuser, POPL 2024: R=16, T=16, citing papers=8, S=0, C=1								
Jonathan Immanuel Brachthäuser	POPL 2024	Algebraic Effects Meet Hoare Logic in Cubical Agda	Donnacha Oisin Kidney; Zhixuan Yang; Nicolas Wu	3	3	0	0	A typed continuation-passing translation for lexical effect handlers (2022; E=1); Compiling effect handlers in capability-passing style (2020; E=1); Effect handlers for the masses (2018; E=1)
Jonathan Immanuel Brachthäuser	POPL 2024	Effectful Software Contracts	Cameron Moy; Christos Dimoulas; Matthias Felleisen	2	2	0	0	Effect handlers, evidently (2020; E=1); Effects, capabilities, and boxes: from scope-based reasoning to type-based reasoning and back (2022; E=1)
Jonathan Immanuel Brachthäuser	POPL 2024	Explicit Effects and Effect Constraints in ReML	Martin Elsman	4	4	0	0	Effects as capabilities: effect handlers and lightweight effect polymorphism (2020; E=1); Effects, capabilities, and boxes: from scope-based reasoning to type-based reasoning and back (2022; E=1); Effekt: Capability-passing style for type- and effect-safe, extensible effect handlers in Scala (2020; E=1); With or Without You: Programming with Effect Exclusion (2023; E=1)
Jonathan Immanuel Brachthäuser	POPL 2024	Polymorphic Reachability Types: Tracking Freshness, Aliasing, and Separation in Higher-Order Generic Programs	Guannan Wei; Oliver Bracevac; Songlin Jia; Yuyan Bao; Tiark Rompf	3	3	0	0	Effects as capabilities: effect handlers and lightweight effect polymorphism (2020; E=1); Effects, capabilities, and boxes: from scope-based reasoning to type-based reasoning and back (2022; E=1); Safer exceptions for Scala (2021; E=1)
Jonathan Immanuel Brachthäuser	POPL 2024	Securing Verified IO Programs Against Unverified Code in F*	Cezar-Constantin Andrici; Stefan Ciobaca; Catalin Hritcu; Guido Martinez; Exequiel Rivas; Eric Tanter; Theo Winterhalter	1	1	0	0	Effects as capabilities: effect handlers and lightweight effect polymorphism (2020; E=1)
Jonathan Immanuel Brachthäuser	POPL 2024	Soundly Handling Linearity	Wenhao Tang; Daniel Hillerstrom; Sam Lindley; J. Garrett Morris	1	1	0	0	Effects as capabilities: effect handlers and lightweight effect polymorphism (2020; E=1)
Jonathan Immanuel Brachthäuser	POPL 2024	Total Type Error Localization and Recovery with Holes	Eric Zhao; Raef Maroof; A Ambedkar Dukkipati; Andrew Blinn; Zhiyi Pan; Cyrus Omar	1	1	0	0	Getting into the Flow: Towards Better Type Error Messages for Constraint-Based Type Inference (2023; E=1)
Jonathan Immanuel Brachthäuser	POPL 2024	When Subtyping Constraints Liberate: A Novel Type Inference Approach for First-Class Polymorphism	Lionel Parreaux; Aleksander Boruch-Gruszecki; Andong Fan; Chun Yin Chau	1	1	0	1	Getting into the Flow: Towards Better Type Error Messages for Constraint-Based Type Inference (2023; E=1)
Fritz Henglein, POPL 2024: R=13, T=13, citing papers=8, S=0, C=0								
Fritz Henglein	POPL 2024	Efficient CHAD	Tom Smeding; Matthijs Vakar	2	2	0	0	Combinatory Adjoints and Differentiation (2022; E=1); Futhark: purely functional GPU-programming with nested parallelism and in-place array updates (2017; E=1)
Fritz Henglein	POPL 2024	Efficient Matching of Regular Expressions with Lookaround Assertions	Konstantinos Mamouras; Agnishom Chattopadhyay	4	4	0	0	Bit-coded Regular Expression Parsing (2011; E=1); Kleenex: compiling nondeterministic transducers to deterministic streaming transducers (2016; E=1); Optimally Streaming Greedy Regular Expression Parsing (2014; E=1); Two-Pass Greedy Regular Expression Parsing (2013; E=1)
Fritz Henglein	POPL 2024	Explicit Effects and Effect Constraints in ReML	Martin Elsman	1	1	0	0	A direct approach to control-flow sensitive region-based memory management (2001; E=1)
Fritz Henglein	POPL 2024	On-the-Fly Static Analysis via Dynamic Bidirected Dyck Reachability	Shankara Narayanan Krishna; A. K. Lal; Andreas Pavlogiannis; Omkar Tuppe	2	2	0	0	Efficient type inference for higher-order binding-time analysis (1991; E=1); Global tagging optimization by type inference (1992; E=1)
Fritz Henglein	POPL 2024	Parametric Subtyping for Structural Parametric Polymorphism	Henry DeYoung; Andreia Mordido; Frank Pfenning; Ankush Das	1	1	0	0	Coinductive Axiomatization of Recursive Type Equality and Subtyping (1998; E=1)
Fritz Henglein	POPL 2024	Polymorphic Type Inference for Dynamic Languages	Giuseppe Castagna; Mickael Laurent; Kim Nguyen	1	1	0	0	Type inference with polymorphic recursion (1993; E=1)

Continued on next page

Table 12: Accepted papers behind the selected PC-year citations, continued.

Researcher	PC	Citing paper	Citing authors	R	T	S	C	Cited prior work(s)
Fritz Henglein	POPL 2024	Type-Based Gradual Typing Performance Optimization	John Peter Campora; Mohammad Wahiduzzaman Khan; Sheng Chen	1	1	0	0	Dynamic typing: syntax and proof theory (1994; E=1)
Fritz Henglein	POPL 2024	Validation of Modern JSON Schema: Formalization and Complexity	Lyes Attouche; Mohamed-Amine Baazizi; Dario Colazzo; Giorgio Ghelli; Carlo Sartiani; Stefanie Scherzinger	1	1	0	0	The complexity of type inference for higher-order typed lambda calculi (1994; E=1)
Małgorzata Biernacka, ICFP 2022: R=4, T=4, citing papers=1, S=4, C=4								
Małgorzata Biernacka	ICFP 2022	A simple and efficient implementation of strong call by need by an abstract machine	Malgorzata Biernacka; Witold Charatonik; Tomasz Drab	4	4	4	4	A Derived Reasonable Abstract Machine for Strong Call by Value (2021; E=1); A syntactic correspondence between context-sensitive calculi and abstract machines (2007; E=1); Deriving an Abstract Machine for Strong Call by Need (2019; E=1); Generalized Refocusing: From Hybrid Strategies to Abstract Machines (2017; E=1)

Researcher	Selected PC service	OpenAlex career age	Google Scholar career age	$t=-2$	$t=-1$	$t=0$	$t=+1$	$t=+2$
Clément Pit-Claudel	PLDI 2024	10	8	6	3	13	5	n/a
Clément Pit-Claudel	OOPSLA 2025	11	9	2	2	9	n/a	n/a
Viktor Kunčak	OOPSLA 2020	27	17	5	8	20	10	n/a
Viktor Kunčak	PLDI 2021	28	18	11	7	18	5	7
Peter Thiemann	ICFP 2023	33	28	3	2	8	1	4
Simon J. Gay	ICFP 2023	30	29	2	3	11	0	1
Martin Elsman	ICFP 2024	26	27	0	0	8	1	n/a
Ezgi Çiçek	ICFP 2019	4	5	1	5	7	2	0
Brandon Lucia	OOPSLA 2019	11	10	10	1	24	14	0
Jonathan Immanuel Brachthäuser	POPL 2024	10	10	0	1	16	2	n/a
Fritz Henglein	POPL 2024	39	33	2	0	13	2	n/a
Małgorzata Biernacka	ICFP 2022	18	18	0	0	4	1	0

Table 13: Raw citation counts for the selected PC member examples.
Note: OpenAlex career age is the selected PC year minus the first OpenAlex CS publication year used for that researcher. Google Scholar career age is the selected PC year minus the first visible year in the researcher's Google Scholar citation plot by year.

Researcher	PC service	Raw E	Title E	Dup.	Citing papers	Self E	Co. E
Clément Pit-Claudel	PLDI 2024	13	13	0	9	2	7
Clément Pit-Claudel	OOPSLA 2025	9	9	0	8	0	1
Viktor Kunčak	OOPSLA 2020	20	19	1	11	0	1
Viktor Kunčak	PLDI 2021	18	14	4	6	0	0
Peter Thiemann	ICFP 2023	8	8	0	3	4	4
Simon J. Gay	ICFP 2023	11	11	0	4	2	2
Martin Elsman	ICFP 2024	8	8	0	4	4	4
Ezgi Çiçek	ICFP 2019	7	5	2	2	0	5
Brandon Lucia	OOPSLA 2019	24	23	1	5	13	14
Jonathan Immanuel Brachthäuser	POPL 2024	16	16	0	8	0	1
Fritz Henglein	POPL 2024	13	13	0	8	0	0
Małgorzata Biernacka	ICFP 2022	4	4	0	1	4	4

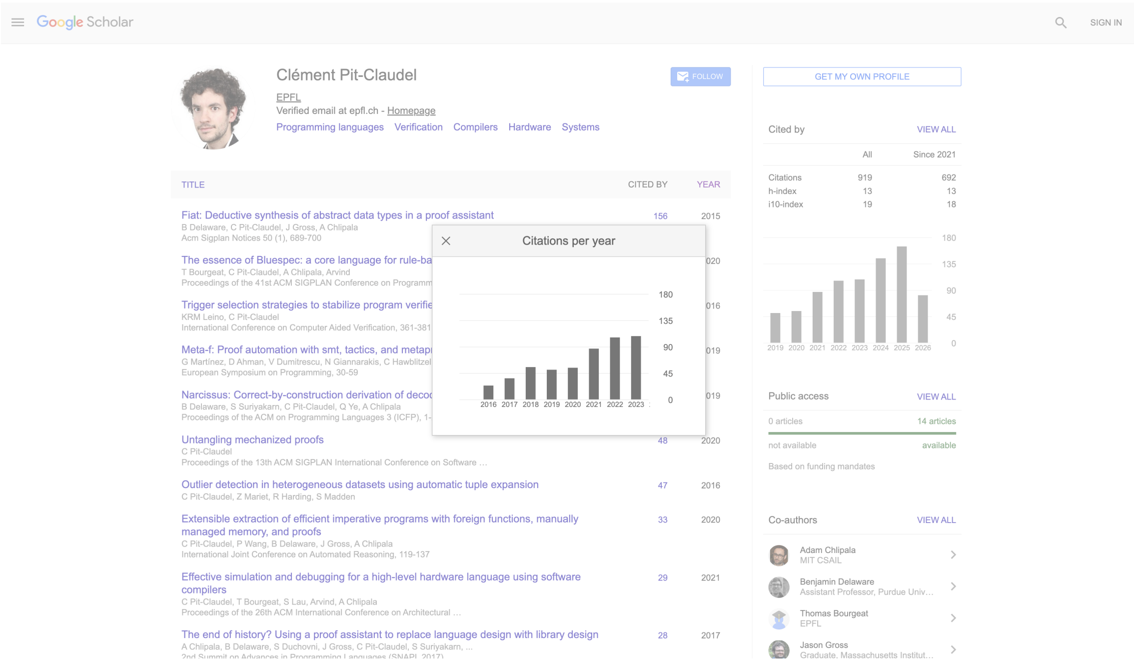
Table 14: Citation source summary for the selected PC service cases.

Table 15: Manual bibliography check examples for selected PC service cases.

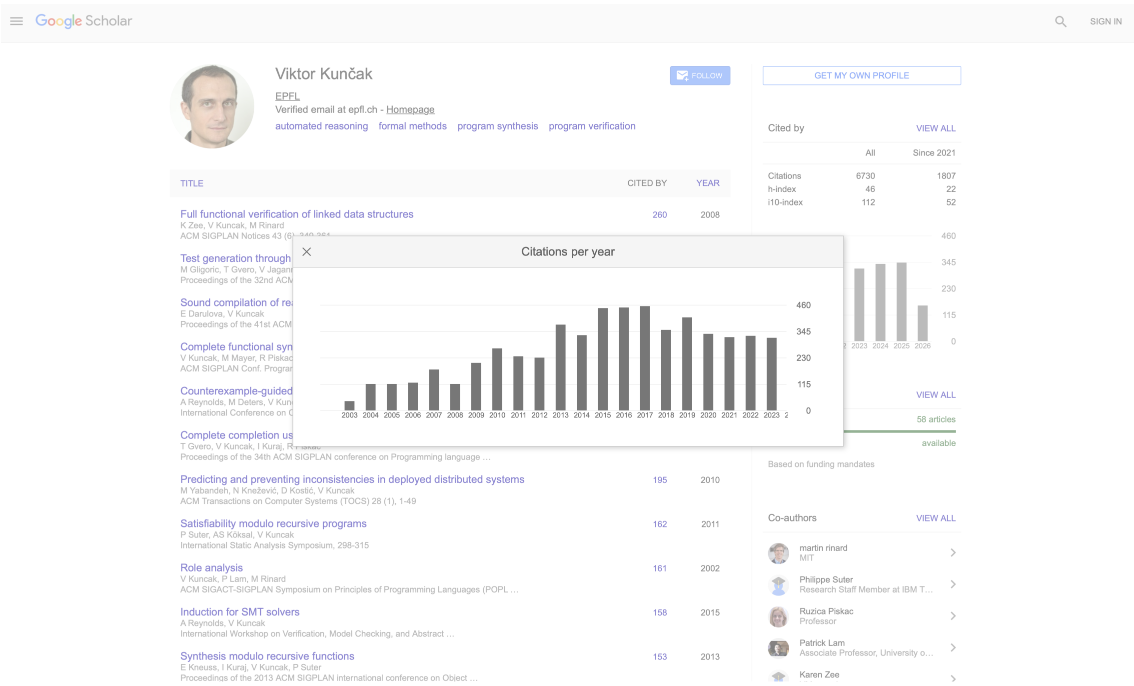
Citing paper or context	Finding	Reference checked
Clément Pit-Claudel		
Live Verification in an Interactive Proof Assistant	Missing	*An Experience Report on Writing Usable DSLs in Coq* (2021)
QED in Context: An Observation Study of Proof Assistant Users	Missing	*Diagrammatic Notations for Interactive Theorem Proving* (2023)
Viktor Kunčak		
Just-in-time learning for bottom-up enumerative synthesis	Missing	*An Update on Deductive Synthesis and Repair in the Leon Tool* (SYNT@CAV 2016)
Beyond the elementary representations of program invariants over algebraic data types	Missing	*Induction for SMT Solvers* (2015)
On the unusual effectiveness of type-aware operator mutations for testing SMT solvers	Extra	*Counterexample-Guided Quantifier Instantiation for Synthesis in SMT*
CoStar: a verified ALL(*) parser	Extra	*LL(1) Parsing with Derivatives and Zippers*
Peter Thiemann		
Dependent Session Protocols in Separation Logic from First Principles	Missing	*Intrinsically-Typed Mechanized Semantics for Session Types* (2019)
Dependent Session Protocols in Separation Logic from First Principles	Missing	*Label-dependent session types* (2020)
Intrinsically Typed Sessions with Callbacks	Missing	*Combining behavioural types with security analysis* (2015)
Intrinsically Typed Sessions with Callbacks	Missing	*Label-Dependent Session Types* (2020)
Simon J. Gay		
Special Delivery: Programming with Mailbox Types	Missing	*Multiparty Session Types for Safe Runtime Adaptation in an Actor Language* (2021)
Martin Elsman		
Double-Ended Bit-Stealing for Algebraic Data Types	Missing	*A Framework for Cut-Off Incremental Recompilation and Inter-Module Optimization* (2008)
Double-Ended Bit-Stealing for Algebraic Data Types	Missing	*Programming with Regions in the MLKit (Revised for Version 4.7.2)* (2022)
Oxidizing OCaml with Modal Memory Management	Missing	*Programming with regions in the ML Kit (for version 4)* (2002)
Ezgi Çiçek		
Relational cost analysis for functional-imperative programs	Missing	*Bidirectional type checking for relational properties* (2019)
Jonathan Immanuel Brachthäuser		
Explicit Effects and Effect Constraints in ReML	Missing	*Tracking Captured Variables in Types* (2021)
Explicit Effects and Effect Constraints in ReML	Missing	*Scoped Capabilities for Polymorphic Effects* (2022)
Polymorphic Reachability Types: Tracking Freshness, Aliasing, and Separation in Higher-Order Generic Programs	Missing	*Tracking Captured Variables in Types* (2021)
Polymorphic Reachability Types: Tracking Freshness, Aliasing, and Separation in Higher-Order Generic Programs	Missing	*Scoped Capabilities for Polymorphic Effects* (2022)
Soundly Handling Linearity	Missing	*Qualified Effect Types – Taming Control-Flow through Linear Effect Handlers* (2023)
Małgorzata Biernacka		
A simple and efficient implementation of strong call by need by an abstract machine	Missing	*An Abstract Machine for Strong Call by Value* (2020)

Researcher	Checked PC service case	R	Miss.	Extra	Net	Interpretation
Clément Pit-Claudel	PLDI 2024; OOPSLA 2025	22	2	0	+2	OpenAlex misses some checked references.
Viktor Kunčak	OOPSLA 2020; PLDI 2021	38	2	2	0	Missing and extra edges offset.
Peter Thiemann	ICFP 2023	8	4	0	+4	Some missing edges are self citations.
Simon J. Gay	ICFP 2023	11	1	0	+1	The missing edge is a self citation.
Martin Elsman	ICFP 2024	8	3	0	+3	Two missing edges are self citations.
Ezgi Çiçek	ICFP 2019	7	1	0	+1	The missing edge is coauthor mediated.
Jonathan Immanuel Brachthäuser	POPL 2024	16	5	0	+5	Missing edges are not self citations.
Małgorzata Biernacka	ICFP 2022	4	1	0	+1	One missing prior work reference.
Total	Selected manual checks	114	19	2	+17	Small net difference across selected checks.

Table 16: Manual bibliography check summary for selected PC service cases.



Clément Pit-Claudel



Viktor Kunčak

Figure 40: Representative Google Scholar screenshots used to check career age.