

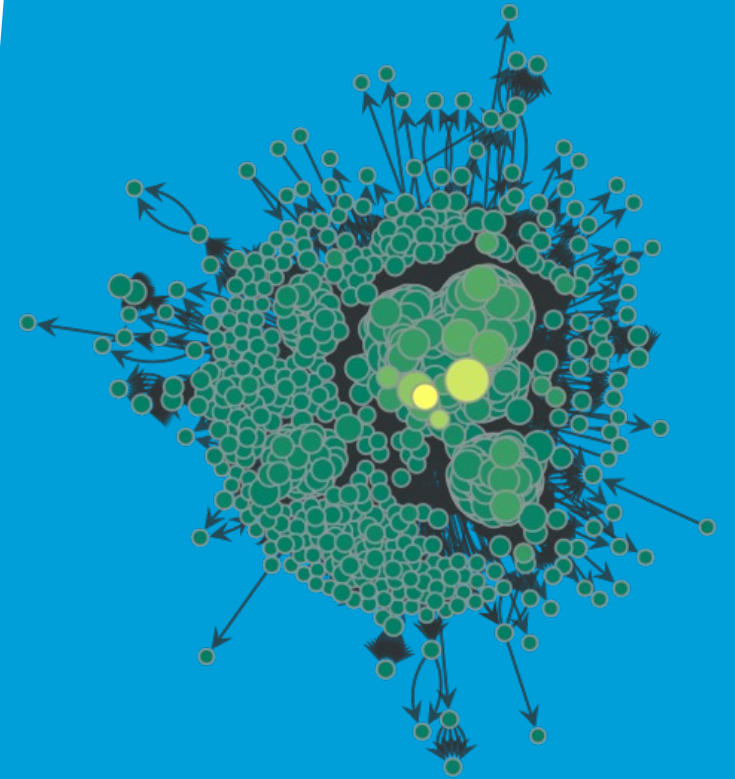
Socialz: Multi-Feature Social Fuzz Testing

Markus Wagner (Monash University)

In Collaboration with
Francisco Zanartu (University of Melbourne) and
Christoph Treude (Singapore Management University)

<https://arxiv.org/abs/2302.08664>
<https://github.com/fzanart/Socialz>

Supported by
Facebook/Meta



in Search Home My Network Jobs Messaging Notifications Me Tools Try Premium for free

5 Annoying Social Media Bugs

 **Charlotte Day**
Creative Director - Contentworks Agency #ContentMarketing Strategist [210 articles](#) [+ Follow](#)

May 19, 2015

As a Global Social Media Manager, I am on the social media platforms all day... every day! When we use social media for professional purposes there are often glitches, either bugs in the platform or intended limitations which can be SO annoying. Here are my top 5 Annoying Social Media Bugs that I really wish the platforms would fix!

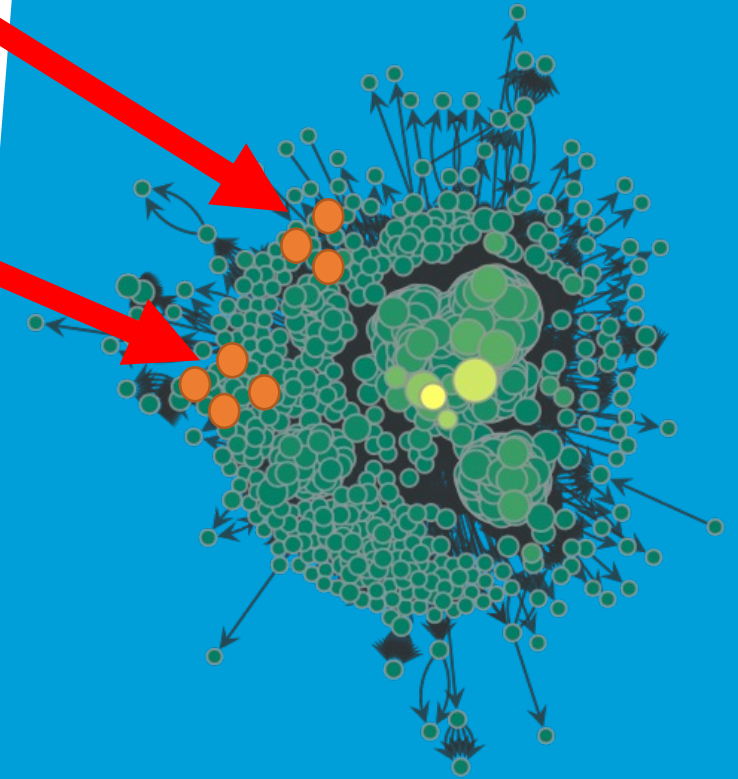
ars TECHNICA

TRUTHS AND RETRUTHS —

Trump's social app marred by bugs and apparent ban on Devin Nunes cow accounts

Trump's social network technically exists now, but good luck trying to use it.

JON BRODKIN - 2/24/2022, 7:49 AM



Trump's social app marred by bugs and apparent ban on Devin Nunes cow accounts

Trump's social network technically exists now, but good luck trying to use it.

JON BRODKIN - 2/24/2022, 7:49 AM

5 Annoying Social Media Bugs

Charlotte Day
Creative Director - Contentworks Agency #ContentMarketing Strategist
210 articles + Follow

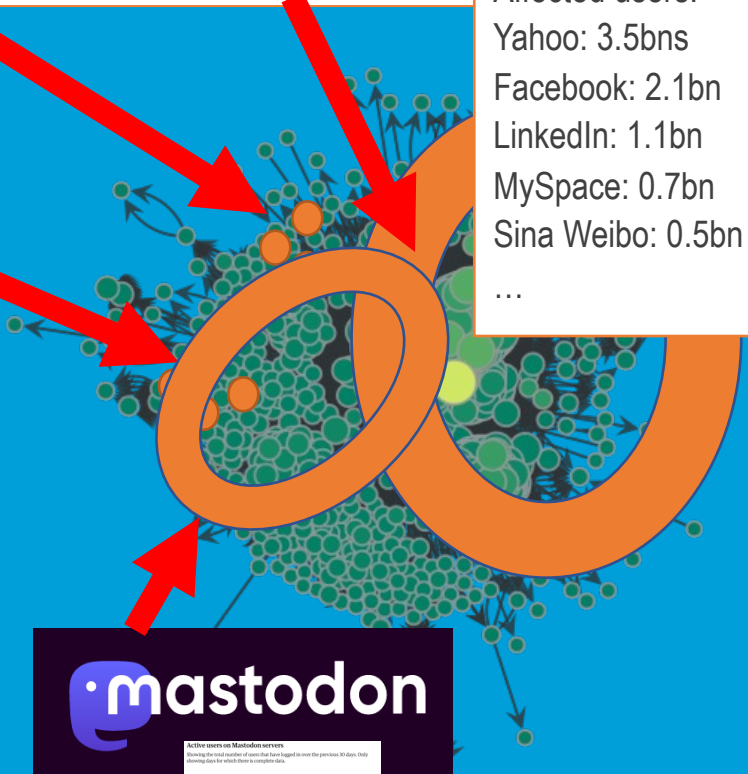
May 19, 2015

As a Global Social Media Manager, I am on the social media platforms all day... every day! When we use social media for professional purposes there are often glitches, either bugs in the platform or intended limitations which can be SO annoying. Here are my top 5 Annoying Social Media Bugs that I really wish the platforms would fix!

Data breaches have become a hazard of being on social media, but some websites are worse at handling our data than others.

In Ireland alone, the [Data Protection Commission](#) received notifications of 6,549 data breaches last year and issued a fine of €225m to Meta-owned WhatsApp over a range of compliance failures.

Affected users:
Yahoo: 3.5bns
Facebook: 2.1bn
LinkedIn: 1.1bn
MySpace: 0.7bn
Sina Weibo: 0.5bn
...



Affected users:
 Yahoo: 3.5bns
 Facebook: 2.1bn
 LinkedIn: 1.1bn
 MySpace: 0.7bn
 Sina Weibo: 0.5bn
 ...

ars TECHNICA

TRUTHS AND RETRUTHS —

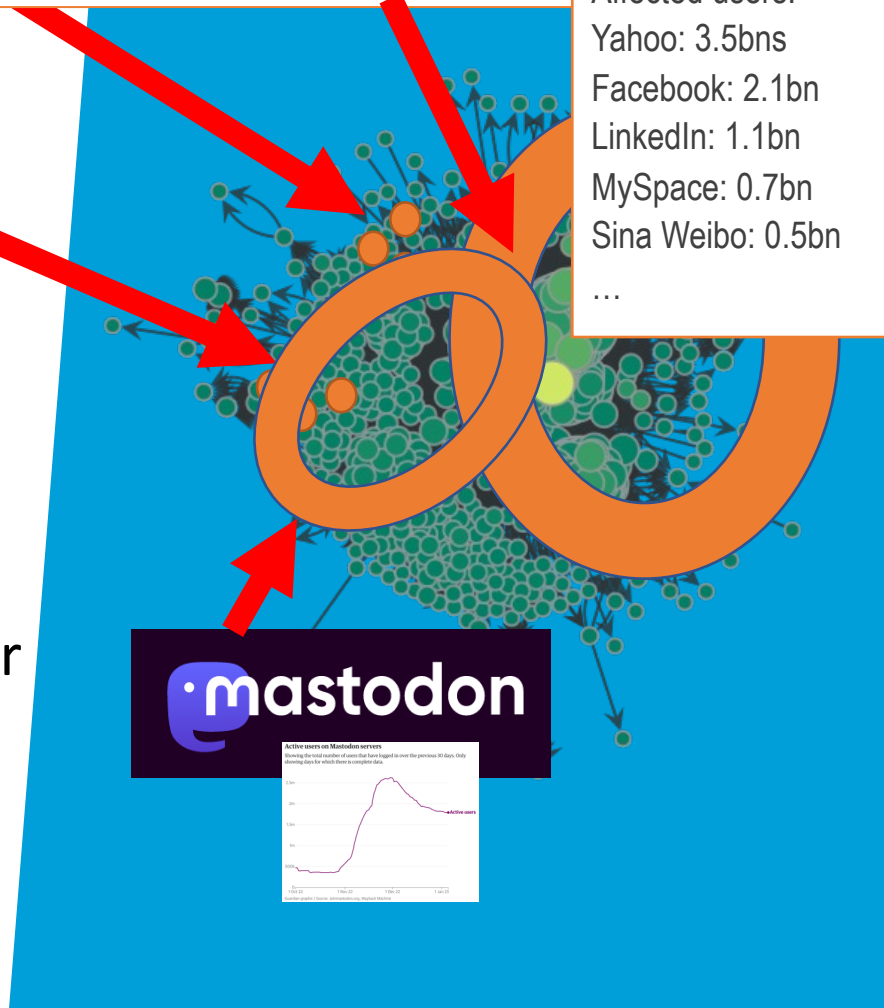
Trump's social app marred by bugs and apparent ban on Devin Nunes cow accounts

Trump's social network technically exists now, but good luck trying to use it.

JON BRODKIN - 2/24/2022, 7:49 AM

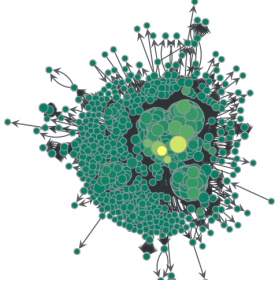
How to discover bugs (using testing)?

- Unit testing
- ...
- System testing: typically the 'journey' of a single user
- **NEW: social testing**, because increasingly, bugs are the result of complex interactions



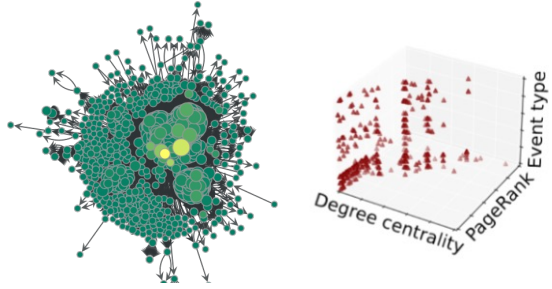
Diversification of Interaction

GitHub: COBOL, 2011-2016,
1523 users X 156 repositories



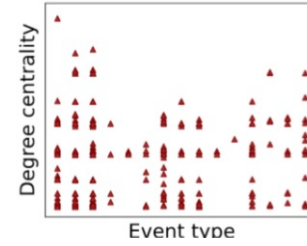
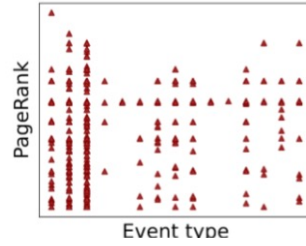
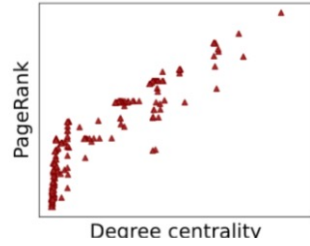
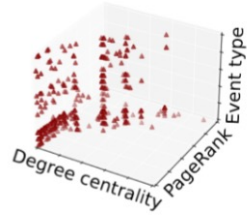
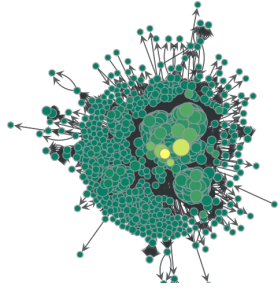
Diversification of Interaction

GitHub: COBOL, 2011-2016,
1523 users X 156 repositories



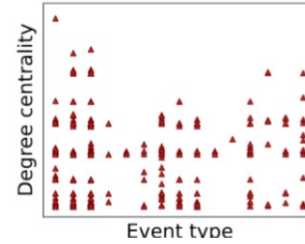
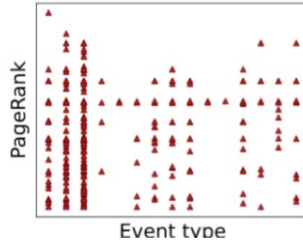
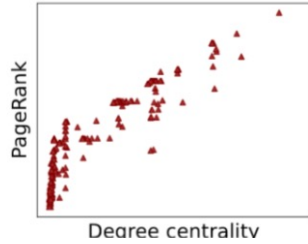
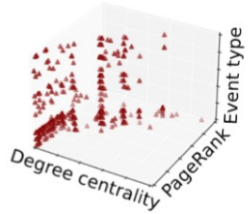
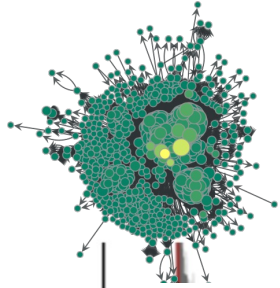
Diversification of Interaction

GitHub: COBOL, 2011-2016,
1523 users X 156 repositories

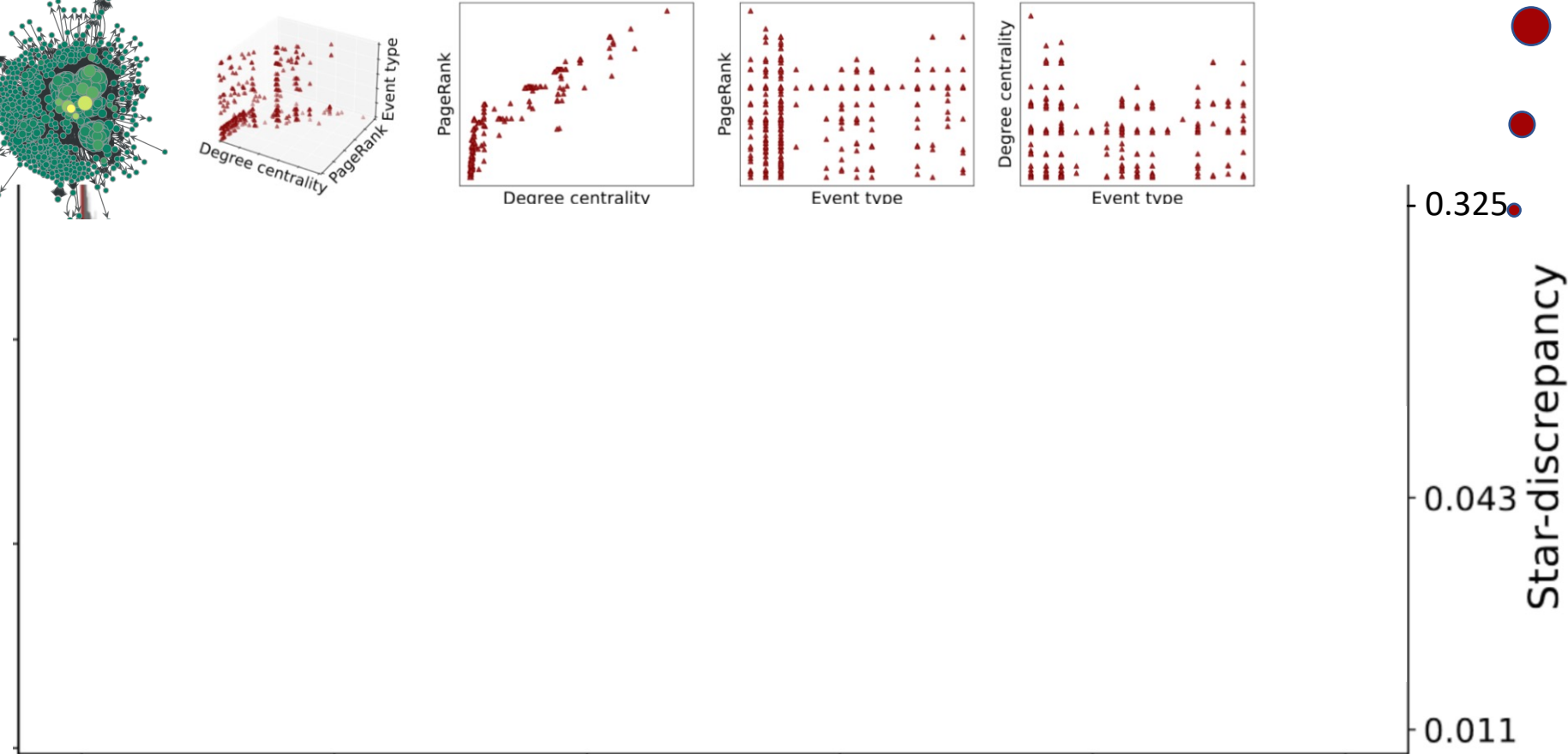
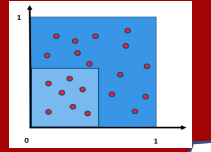


Diversification of Interaction

GitHub: COBOL, 2011-2016,
1523 users X 156 repositories



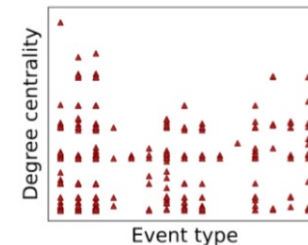
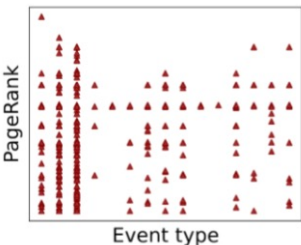
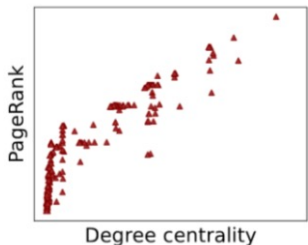
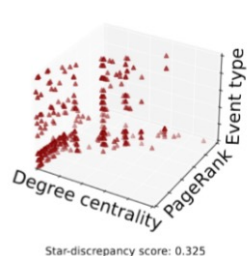
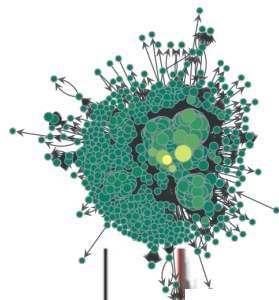
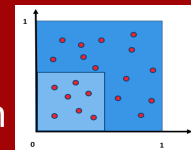
Star discrepancy: measures the regularity with which points are distributed in boxes anchored in the origin



Diversification of Interaction

GitHub: COBOL, 2011-2016,
1523 users X 156 repositories

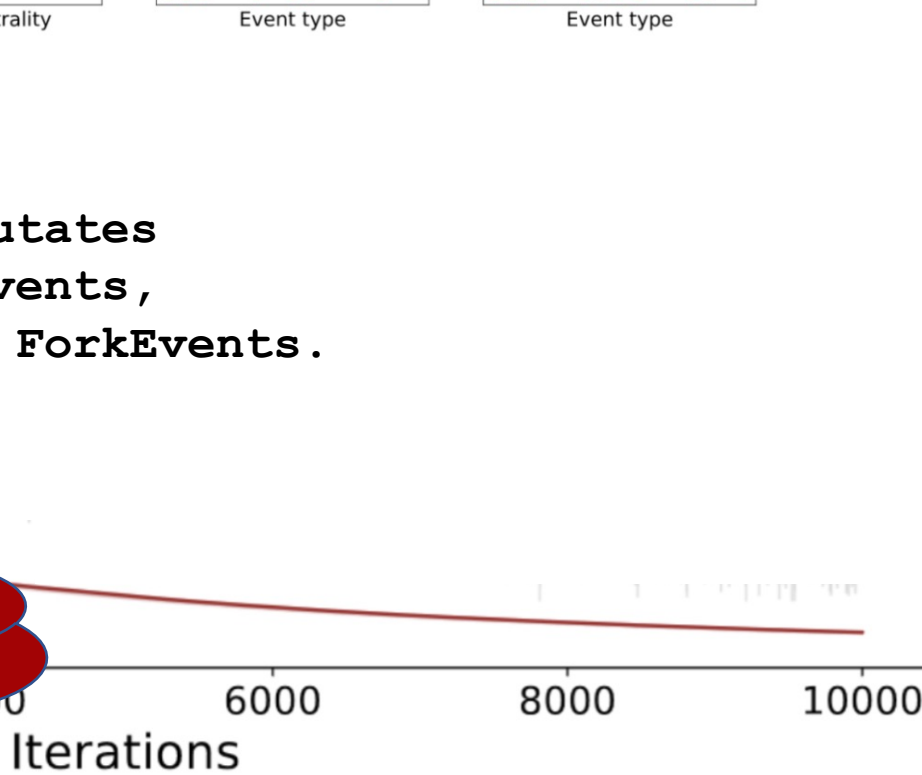
Star discrepancy: measures the regularity with which points are distributed in boxes anchored in the origin



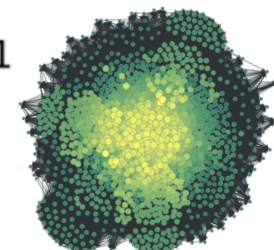
Simple (1+20)-EA mutates
PushEvents, WatchEvents,
PullRequestEvents, ForkEvents.



Mutate interactions in the graph
→ If diversity improves, then
keep the new graph for the next
iteration.



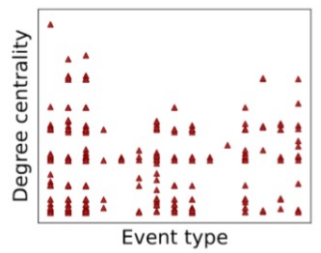
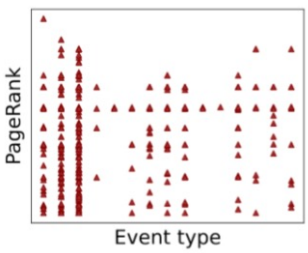
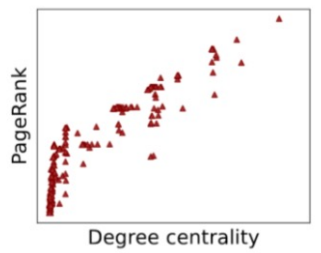
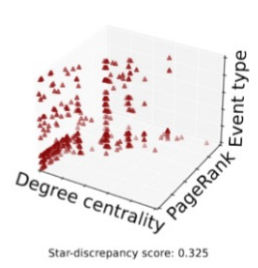
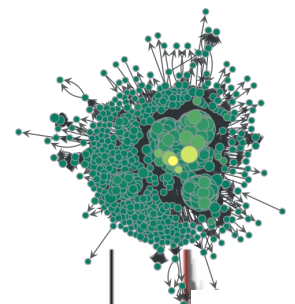
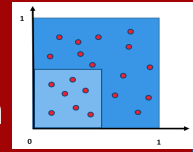
Star-discrepancy



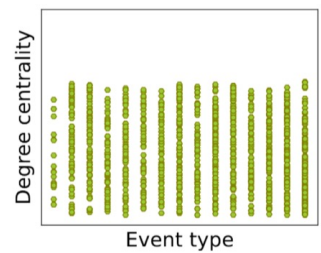
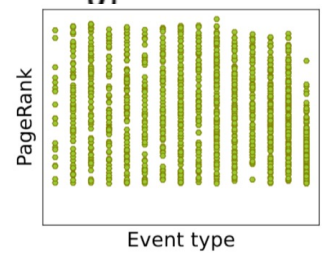
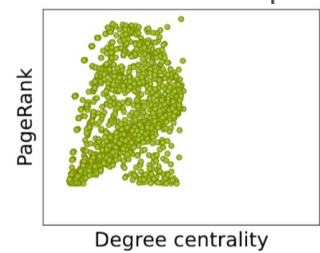
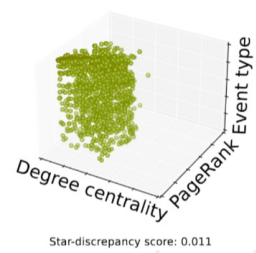
Diversification of Interaction

GitHub: COBOL, 2011-2016,
1523 users X 156 repositories

Star discrepancy: measures the regularity with which points are distributed in boxes anchored in the origin



Simple (1+20)-EA mutates
PushEvents, WatchEvents,
PullRequestEvents, ForkEvents.



0.325

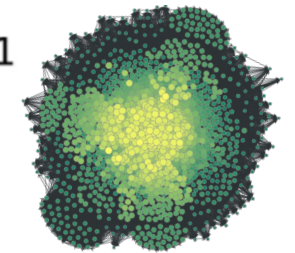
scrapancy

0.011

Iterations

6000 8000 10000

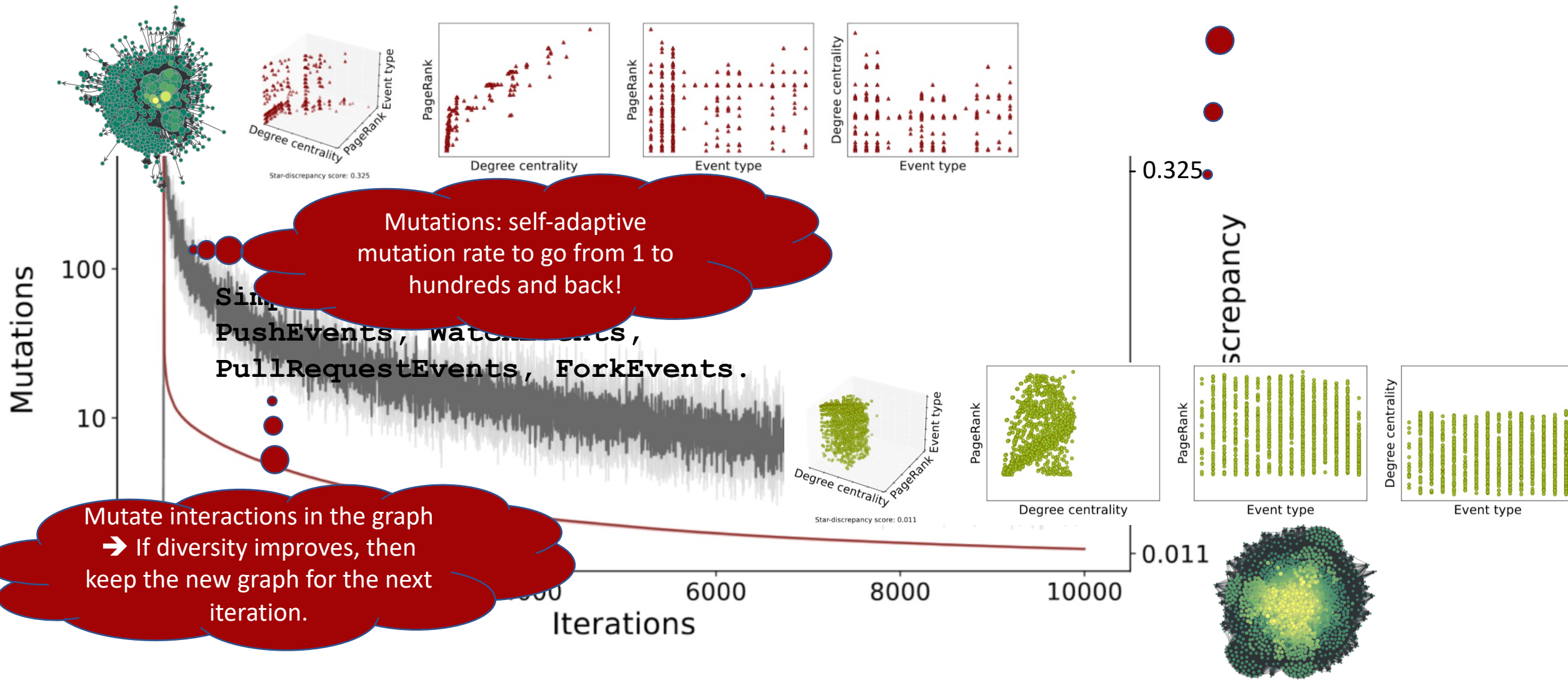
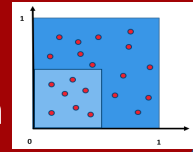
Mutate interactions in the graph
→ If diversity improves, then
keep the new graph for the next
iteration.



Diversification of Interaction

GitHub: COBOL, 2011-2016,
1523 users X 156 repositories

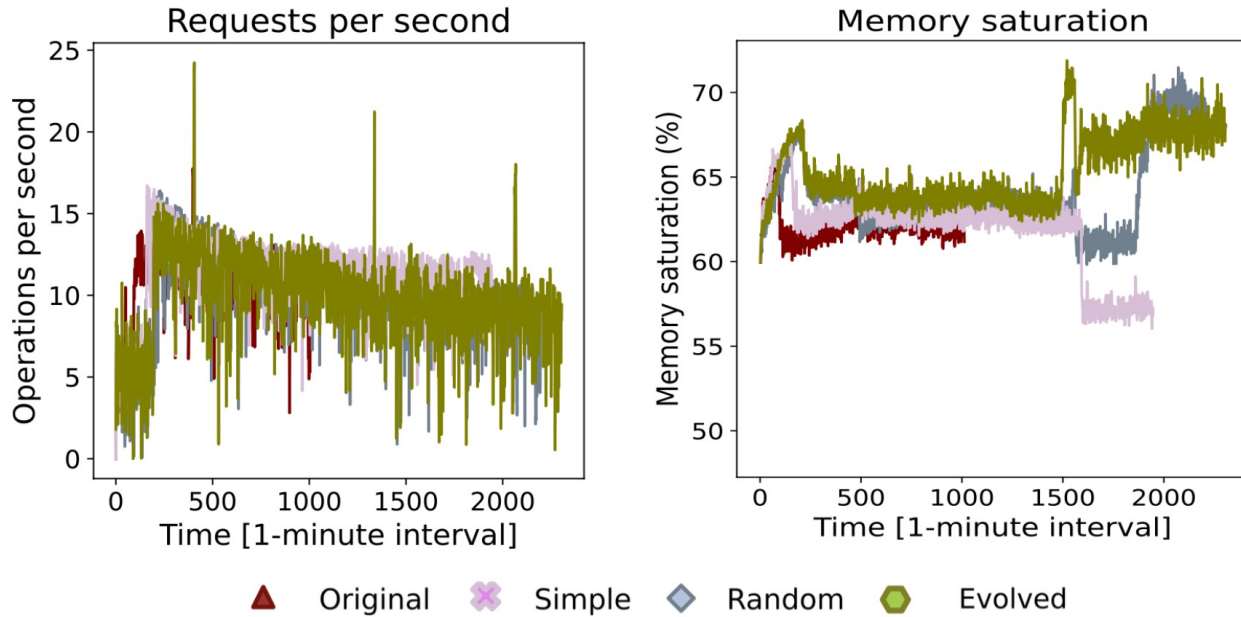
Star discrepancy: measures the regularity with which points are distributed in boxes anchored in the origin



Mutations: self-adaptive mutation rate to go from 1 to hundreds and back!

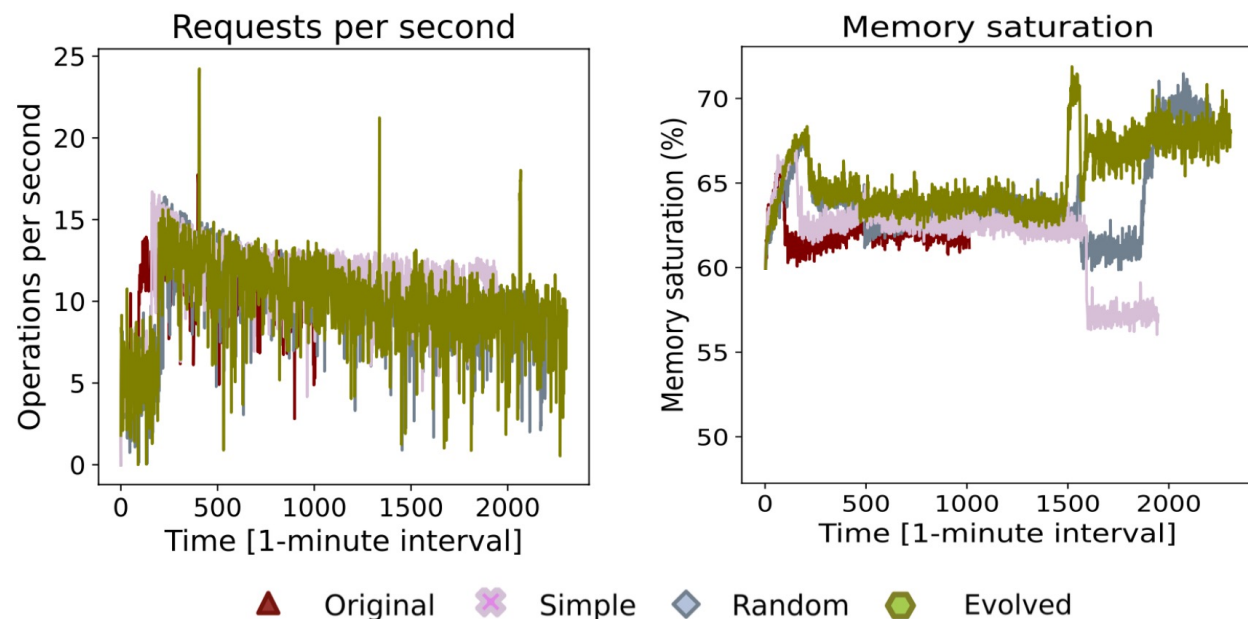
Mutate interactions in the graph
→ If diversity improves, then keep the new graph for the next iteration.

So what? 1/2



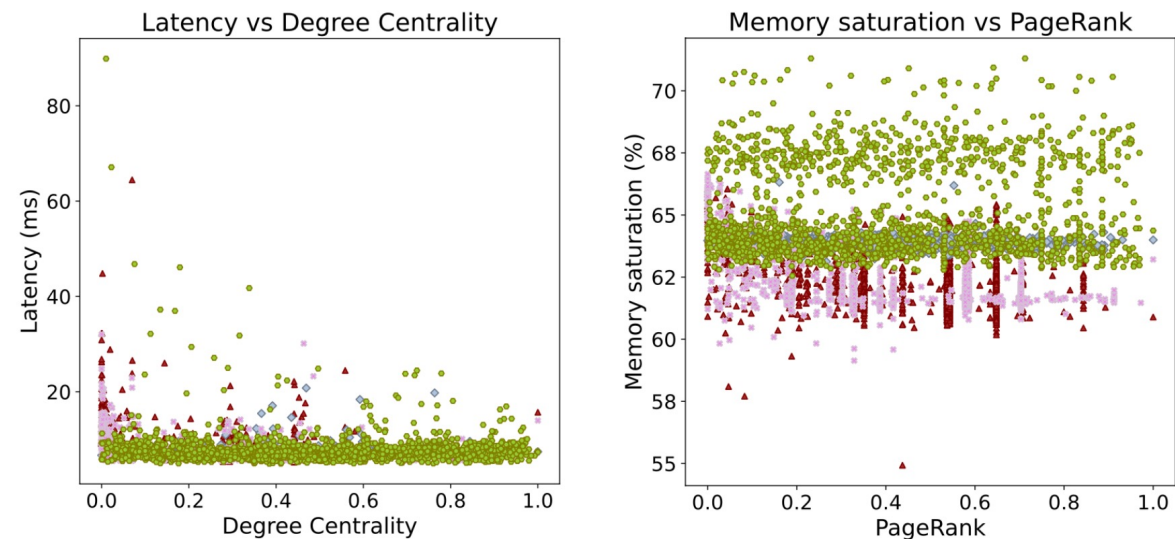
Processing of community interactions over time,
when loading the interaction graphs on our GitLab server
(we used copies of a clean setup as a common starting point)

So what? 1/2



Processing of community interactions over time, when loading the interaction graphs on our GitLab server (we used copies of a clean setup as a common starting point)

Correlation of user features and resource utilisation. Shown are the averages for each of the 1523 users



So what? 2/2

We ran this on a GitLab CE server and observed performance differences and a limitation*.

Fuzzing results. The 6,907 errors are the result of executing 1,696,386 events.

API return value	API explanation [16]	count	Our best explanation
304 Not Modified	The resource hasn't been modified since the last request.	1 (0.01%)	We have identified the offending user and their actions, but we have no explanation.
400 Bad Request	A required attribute of the API request is missing. For example, the title of an issue is not given.	3 (0.04%)	It is unclear why the three "issues" (PushEvents) fail, as we are enforcing minimum and maximum length on issue title, body, and message.
403 Forbidden	The request isn't allowed. For example, the user isn't allowed to delete a project.	172 (2.49%)	Because of a known issue: GitLab lag [28].
404 Not Found	A resource couldn't be accessed. For example, an ID for a resource couldn't be found, or the user isn't authorized to access the resource.	3660 (52.99%)	Because of a known issue: GitLab lag [28].
409 Conflict	A conflicting resource already exists. For example, creating a project with a name that already exists.	297 (4.30%)	Because of a known issue: GitLab lag [28].
NaN	<i>(not documented)</i>	2774 (40.16%)	We have no explanation.

So what? 2/2

We ran this on a GitLab CE server and observed performance differences and a limitation*.

Fuzzing results. The 6,907 errors are the result of executing 1,696,386 events.

API return value	API explanation [16]	count	Our best explanation
304 Not Modified	The resource hasn't been modified since the last request.	1 (0.01%)	We have identified the offending user and their actions, but we have no explanation.
400 Bad Request	A required attribute of the API request is missing. For example, the title of an issue is not given.	3 (0.04%)	It is unclear why the three "issues" (PushEvents) fail, as we are enforcing minimum and maximum length on issue title, body, and message.
403 Forbidden	The request isn't allowed. For example, the user isn't allowed to delete a project.	172 (2.49%)	Because of a known issue: GitLab lag [28].
404 Not Found	A resource couldn't be accessed. For example, an ID for a resource couldn't be found, or the user isn't authorized to access the resource.	3660 (52.99%)	Because of a known issue: GitLab lag [28].
409 Conflict	A conflicting resource already exists. For example, creating a project with a name that already exists.	297 (4.30%)	Because of a known issue: GitLab lag [28].
NaN	<i>(not documented)</i>	2774 (40.16%)	We have no explanation.

Final thoughts: how to evolve and evaluate? Slow speed (1 day), lots of side-effects on the virtual machine, lots of noise, "rollbacks" impossible or impractical, ...

Summary: we evolved community interactions that are substantially different from existing ones, and we found bugs that are beyond our debugging skills

