

How to Cook a Paper: from Recipes to Dishes

Chen Yan

2021.11.12

How do you cook?

1. Find recipes
2. Read & learn recipes
3. Design your recipe
4. Get your hands dirty
5. Cook!
6. Dish up
7. Serve
8. Wash the dishes



How do you **cook** a paper?



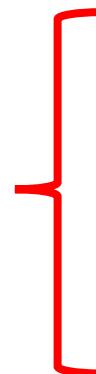
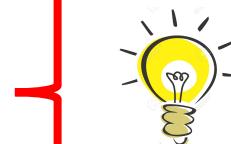
- | | |
|-------------------------|-----------------------------------|
| 1. Find recipes | 1. Find papers |
| 2. Read & learn recipes | 2. Read papers |
| 3. Design your recipe | 3. Have a research idea |
| 4. Get your hands dirty | 4. Experiment |
| 5. Cook! | 5. Write your paper |
| 6. Dish up | 6. Submit your paper |
| 7. Serve | 7. Rebuttal & revision |
| 8. Wash the dishes | 8. Followups |

How do you **cook** a paper?



Reimplementing
other people's
work is a good
starting point.

Practice,
practice,
and practice!



- 1. Find papers**
- 2. Read papers**
- 3. Have a research idea**
- 4. Experiment**
- 5. Write your paper**
- 6. Submit your paper**
- 7. Rebuttal & revision**
- 8. Followups**

1. Find papers – for purpose



You want to learn a research topic

/make sure nobody has done it
/find reference

- Read survey/sok papers (lucky)
- Google Scholar with keywords, find 3-5 recent papers
- Read the related work sections
- Find shared citations and repeated author names
→ **popular papers and researchers**
- Go to their websites and see where they've published recently → ***top conferences***

1. Find papers – as a habit



But you also need to keep a good habit:

Always be keen to new and good papers!

- Regularly watch for the good source:
 - The big-4: IEEE S&P, Usenix Security, CCS, NDSS
 - Top conferences/journals in related fields
 - Best/distinguished papers
 - Science/Nature papers
 - Website/google scholar of good researchers/teams
 - Know their rating: CCF推荐排名
 - Know their timeline: sec-deadline, confsearch
- Learn from **good papers** that you enjoy reading

CCF推荐会议/期刊目录

中国计算机学会 China Computer Federation 为计算领域的专业人士服务

CNCC2021 加入CCF

CNCC YEF NOI FCES

关于CCF 新闻 会员 活动 出版物 数字图书馆 分支机构 服务机构 奖励 计算机历史 合作 学会党建

您的位置：首页 > 学术评价 > ● 网络与信息安全

CCF推荐国际学术刊物目录 CCF推荐中文学术刊物目录

●简介 ●计算机体系结构/并行与分布计算/存储系统 ●计算机网络 ●**网络与信息安全** ●软件工程/系统软件/程序设计语言 ●数据库/数据挖掘/内容检索
●计算机科学理论 ●计算机图形学与多媒体 ●人工智能 ●人机交互与普适计算 ●交叉/综合/新兴 [联系我们](#)

**中国计算机学会推荐国际学术刊物
(● 网络与信息安全)**

A类

序号	刊物名称	刊物全称	出版社	地址
1	TDSC	IEEE Transactions on Dependable and Secure Computing	IEEE	http://dblp.uni-trier.de/db/journals/tdsc/

《2020-2021中国计算机科学技术发展报告》发布



https://www.ccf.org.cn/Academic_Evaluation/By_category/

Google Scholar profile

≡ Google Scholar



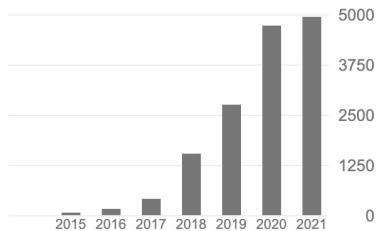
Nicholas Carlini
Google Brain
Verified email at google.com - [Homepage](#)

[FOLLOW](#) [GET MY OWN PROFILE](#)

TITLE	CITED BY	YEAR
Towards evaluating the robustness of neural networks N Carlini, D Wagner 2017 IEEE Symposium on Security and Privacy (SP), 39-57	4356	2017
Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples A Athalye, N Carlini, D Wagner ICML 2018	1804	2018
Adversarial examples are not easily detected: Bypassing ten detection methods N Carlini, D Wagner Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security ...	1185	2017
Mixmatch: A holistic approach to semi-supervised learning D Berthelot, N Carlini, I Goodfellow, N Papernot, A Oliver, CA Raffel Advances in Neural Information Processing Systems, 5050-5060	907	2019
Audio adversarial examples: Targeted attacks on speech-to-text N Carlini, D Wagner 2018 IEEE Security and Privacy Workshops (SPW), 1-7	697	2018
Hidden Voice Commands. N Carlini, P Mishra, T Vaidya, Y Zhang, M Sherr, C Shields, D Wagner, ... USENIX Security Symposium, 513-530	502	2016

Cited by

	All	Since 2016
Citations	14907	14700
h-index	31	31
i10-index	36	36



Public access [VIEW ALL](#)

0 articles	12 articles
not available	available

Based on funding mandates

Sec-deadline

Security and Privacy Conference Deadlines

 Tweet

 Star

Countdowns to top Security and Privacy conference deadlines.

To add/update a conference, send in a [pull request](#).

Last update: 5 November 2021

Security Privacy Crypto

S&P (Oakland) 2022

IEEE Symposium on Security and Privacy

May 22 – 26 2022 // San Francisco, California, USA

22 days 04h 03m 25s

Deadline: 3 Dec 2021, 7:59:59 pm

Three deadlines

ICICS 2022

The 24th International Conference on Information and Communications Security

Sept 05 – 08 2022 // University of Kent, Canterbury, UK

131 days 04h 03m 25s

Deadline: 22 Mar 2022, 7:59:59 pm

AsiaCCS 2022

ACM Asia Conference on Computer and Communications Security

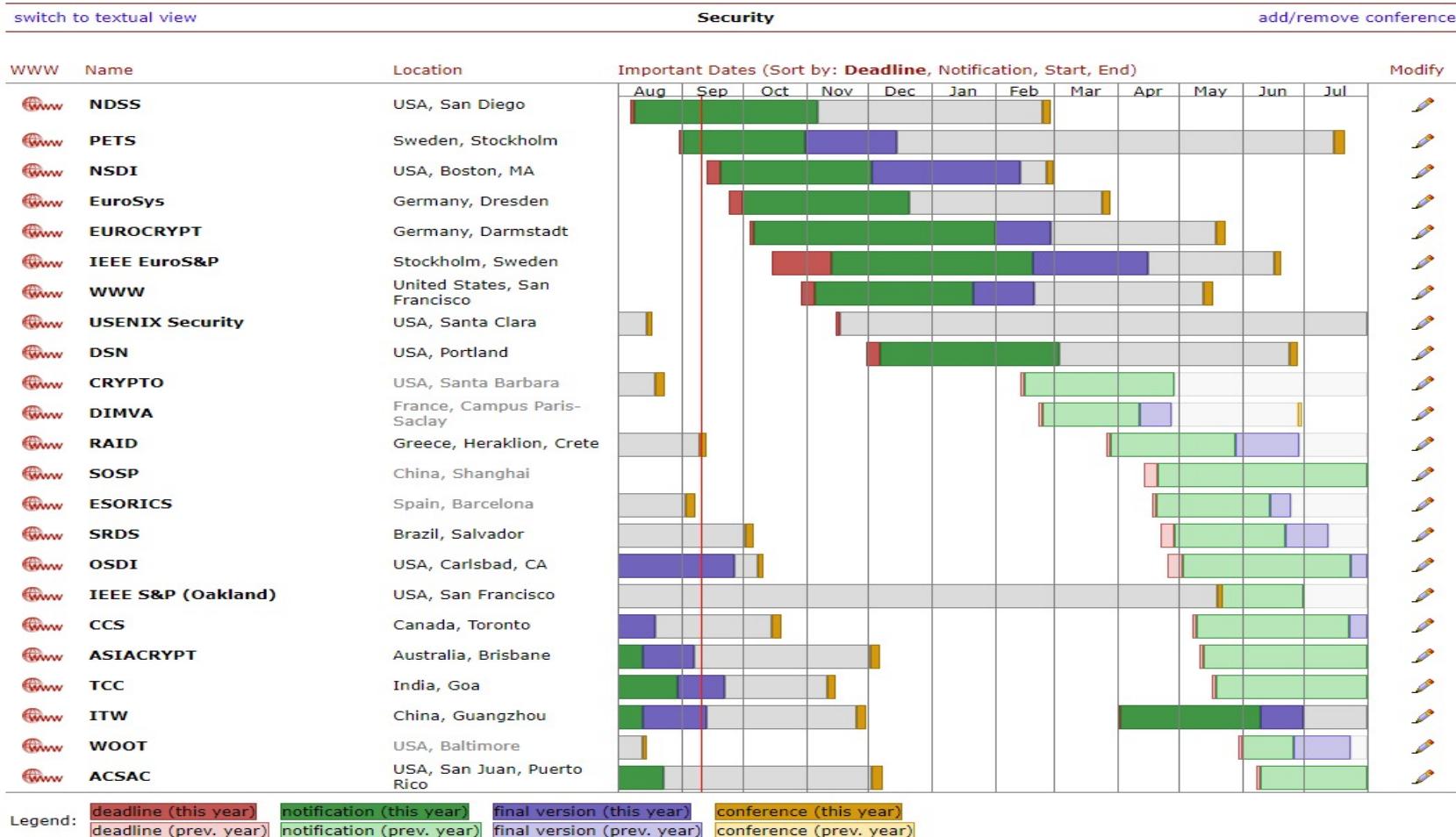
May 30 – June 3 // Nagasaki, Japan

09 days 04h 03m 25s

Deadline: 20 Nov 2021, 7:59:59 pm

2 review cycles

Confsearch



This data is provided by www.confsearch.org, a search tool for computer science conferences.

Paper management

Manage your papers with reference managers

- Zotero/Mendeley/Endnote ...

The screenshot shows the Zotero application window. On the left is a sidebar with a tree view of 'My Library' containing sections like 'Colonial Medicine', 'Teaching', and 'My Publications'. Below this are 'Group Libraries' and a 'To Read' section with various topics. The main area is a table view of references. A specific entry for 'Circulation of Medicine in the Early Modern Atlantic World' by Cook and Walker (2013) is selected and expanded, showing its full text PDF and a list of related items. To the right of the table is a detailed 'Info' panel displaying metadata such as Item Type (Journal Article), Title (Circulation of Medicine in the Early Modern Atlantic World), Authors (Cook, Harold J. and Walker, Timothy D.), Abstract (The search for powerful drugs has caused people and commodities to move around the globe for many centuries, as it still does...), Publication (Social History of Medicine), Volume (26), Issue (3), Pages (337-351), Date (2013/08/01), Series (Series Title and Series Text), Journal Abbr (Soc Hist Med), Language (en), DOI (10.1093/shm/hkt013), ISSN (0951-631X), Short Title (URL https://academic.oup.com/shm/article/26/3...), Accessed (1/24/2018, 10:17:12 AM), and Loc. in Archive (Archive). There are also tabs for 'Notes', 'Tags', and 'Related'.

And most importantly,



always share!

With the lab or your group/team members

2. Read papers



A typical researcher will likely spend hundreds of hours every year reading papers.

You read other papers so that

- You are learning what papers are like
- You are currently in the field
- You maybe writing a survey (literature review)
- You want to find what to compare with
- You review papers for a conference

We follow a **3 pass reading approach**

Borrowed from Keshav at Waterloo and Jon Crowcroft at Cambridge

2. The three-pass approach



The three pass approach:

- The *1st* pass gives you a **general idea** about the paper
- The *2nd* pass lets you grasp the paper's **content**, but **not its details**
- The *3rd* pass helps you understand the paper **in depth**

Do not drown in details before getting a bird's-eye-view!

2. Read paper – Pass 1



Structural overview of paper

- Carefully read the title, abstract, and introduction
- Read the section and subsection headings, but ignore everything else
- Read the conclusions
- Scan references noting ones you know

2. Read paper – Pass 1 output



- You can now answer the *five Cs*
 - Category: What type of paper is this (category defines methodology)
 - Context: What other papers/projects relate to this?
 - Correctness: Are the assumptions valid?
 - Contributions: What are the **key novel scientific** contributions?
 - Clarity: Is the paper well written?
- Takes about 5 minutes
- Adequate for papers that aren't in your research area, but may someday prove relevant

2. Read paper – Pass 2



- Check integrity of paper
 - Look carefully at figures, diagrams and other illustrations
 - Note unfamiliar references for further reading
 - Do not check proofs/experimental details yet
- Takes ~ 1 hour
- You should be able to summarize the paper, with supporting evidence, to someone else now
- Appropriate for a paper in which you are interested, but does not lie in your research specialty

2. Read paper – Pass 3



- Virtually **re-implement** the paper
 - *Think as the author*
 - Challenge every assumption in every statement
 - Think how you would present a particular idea
 - Think adversarially about experiments, proofs, simulation scenarios
 - Ideas for future work
- Takes 4-5 hours for beginners, 1 hour for experienced readers
- You should be able to reconstruct the entire paper from memory and identify its strong and weak points



3. Have a research idea

Ideas are cheap. Execution is expensive. 😊

Ideas are 1%
execution is 99%



Great ideas
are rare. 😞

No idea 😞

need good idea



0.1% 2%

IQ score 55 70

34% 34%

68%

95%

14%

2% 0.1%



You need to start
with a strong idea

PhD 1-2 Y

PhD 2-4 Y

PhD 4+ Y



3. Have a research idea

Methods to have an idea:

- **Read extensively and find the unsolved problems** 梳理法
 - Example: liveness detection

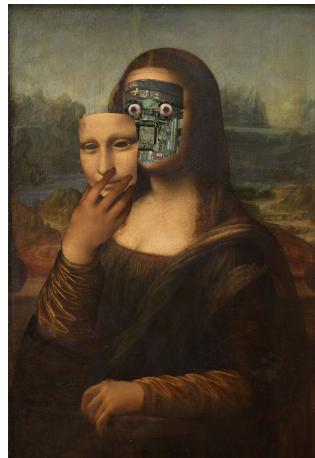




3. Have a research idea

Methods to have an idea:

- **Read extensively and find the unsolved problems** 梳理法
 - Example: liveness detection
- **Read in-depth and explore new directions** 发散法
 - Example: lightcommand, poltergeist, eararray, ultracom





3. Have a research idea

Methods to have an idea:

- **Read extensively and find the unsolved problems** 梳理法
 - Example: liveness detection
- **Read in-depth and explore new directions** 发散法
 - Example: lightcommand, poltergeist, eararray, ultracom
- **Interesting phenomenon in life** 好奇法
 - Example: bluenote, ghost touch, capspeaker
- **Find and solve real world problems** 现实法
 - Example: voiceprint, fakewake, dewicam, dehirec
- **Just try, maybe things do not turn out as expected** 尝试法
 - Example: dolphinattack, AE, tesla, rolling shutter
- **Ideas just come to you** 天才法

Advice 1: Relax, open mind, think, brainstorm, and discuss

Advice 2: Remember to write down your idea at any time!

4. Experiment

A few tips:

- It is definitely worthwhile to know your tools first.
- Learn quick with practice. Only learn systematically when you have time.
- Focus on your goal. Do not chase after butterflies.
- Keep a good track of your time and progress.
- Do not give up easily.
- Actively ask for advice.
- Be devoted. Achieve your goal at any cost!

5. Write your paper

How to write a good paper?

Tools:

- Write in LaTeX, no matter what
- Manage your code and paper with Git
- Remotely host your code and paper and collaborate with others on Github

5. Write your paper - LaTeX

- Typeset your paper correctly and beautifully
- You do need to spend some time learning LaTeX,
but the typesetting results are visibly better

No ligature



efficient

efficient

Correct ligature



LATEX

5. Write your paper - LaTeX

- Get started
 - Books: *LaTeX Tutorials*, *The Not So Short Introduction to LaTeX 2e*
 - Advice: <https://github.com/dspinellis/latex-advice>
 - Package install: MiKTeX, proTeXt, Texlive, or MacTeX
 - Local editors: TeXstudio, WinEdt, TeXmaker, TeXworks, or editors (e.g., in Sublime Text)
 - Online editor: overleaf
- Tips
 - Put different sections into different .tex files and input them in the main .tex file
 - Put figures, .tex files, templates into different folders
 - Include all temporary files and your paper.pdf into .gitignore

5. Write your paper – Git(hub)

- Version control is important!
- Git repositories help you to track all histories and collaborate with others
- Difference between Git and Github
- Learn Git
 - Books: *Pro Git* (<https://git-scm.com/book/en/v2>)
 - Tutorials: too many online
- Use git with Github – a hands-on guide

About version control

- What is “version control”, and why should you care?
- Get crazy when you can’t find earlier texts after saving and exiting word?
- Tired of endless file.v1 .v1.1 .v1.1.1 v1.2 v1.2.1?
- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

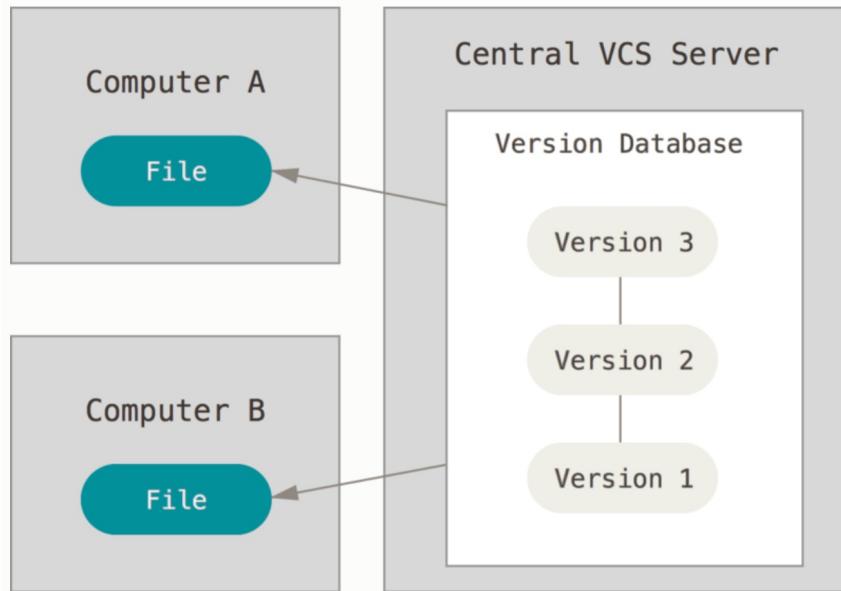
ONE DOES NOT SIMPLY

UNDERSTAND GIT

MemesHappen

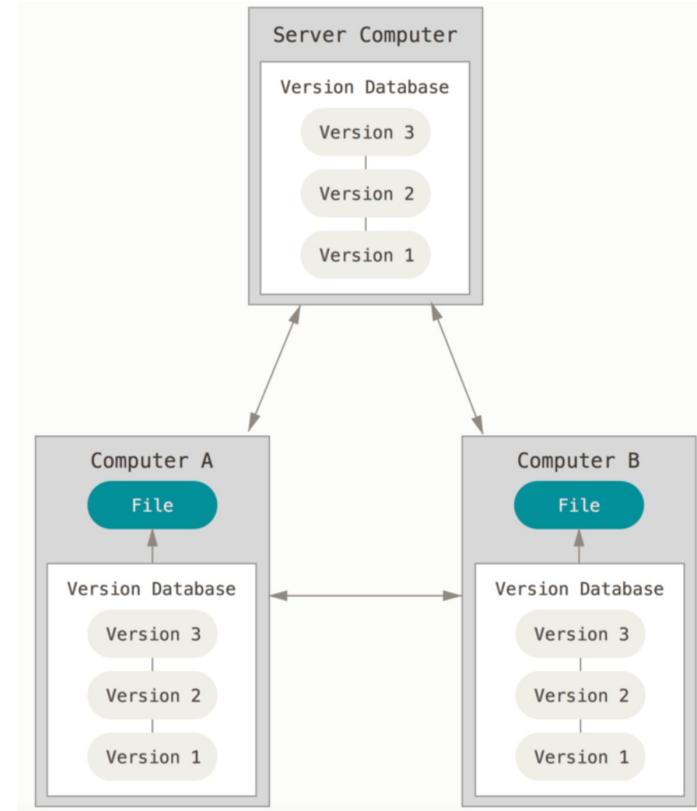
Centralized vs Distributed

Centralized version control



CVS, Subversion, and Perforce

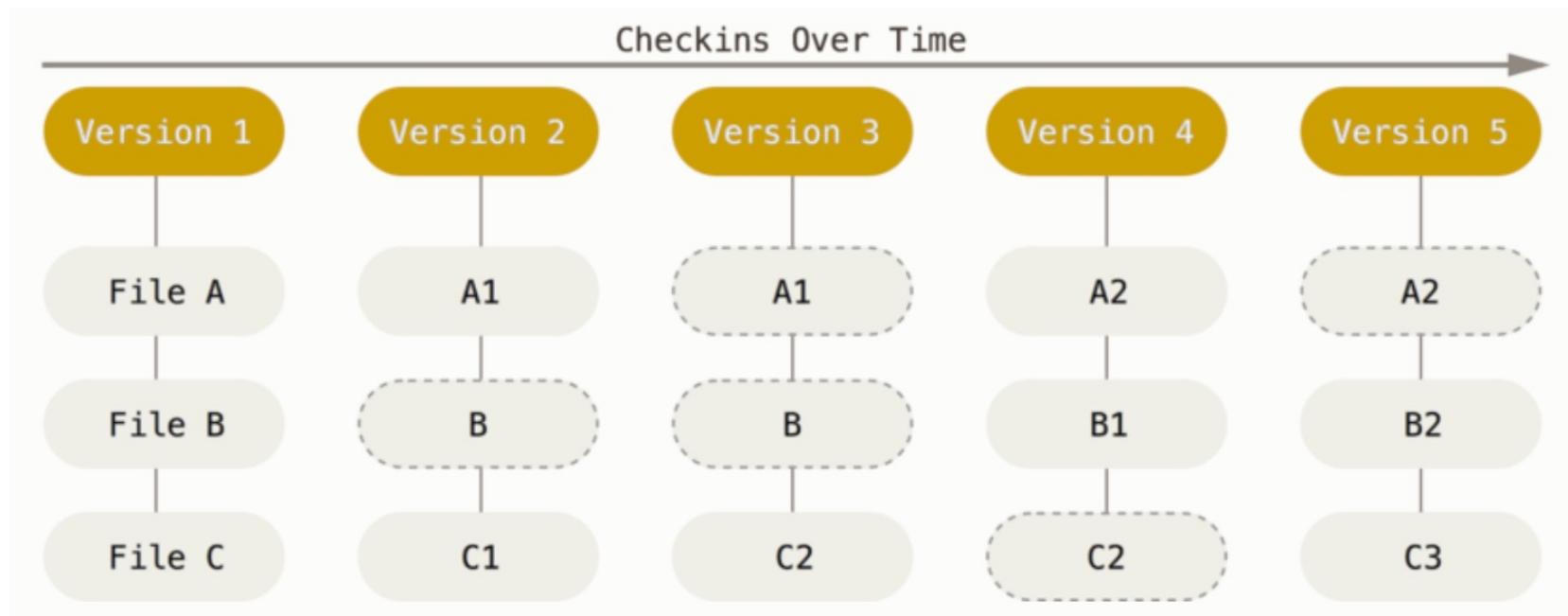
Distributed version control



Git, Mercurial, Bazaar or Darcs

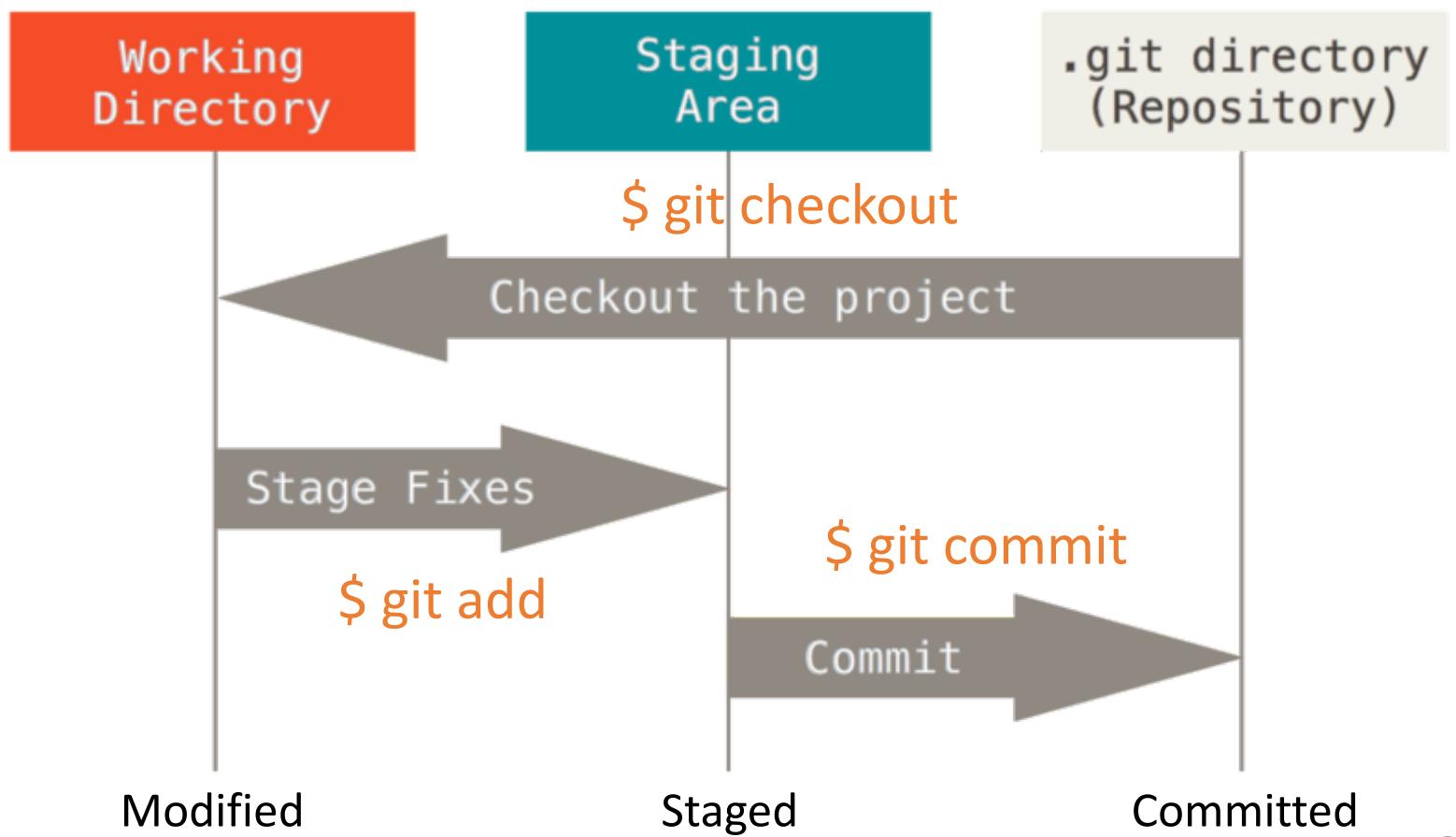
What is Git?

Git thinks about its data more like a **stream of snapshots**.

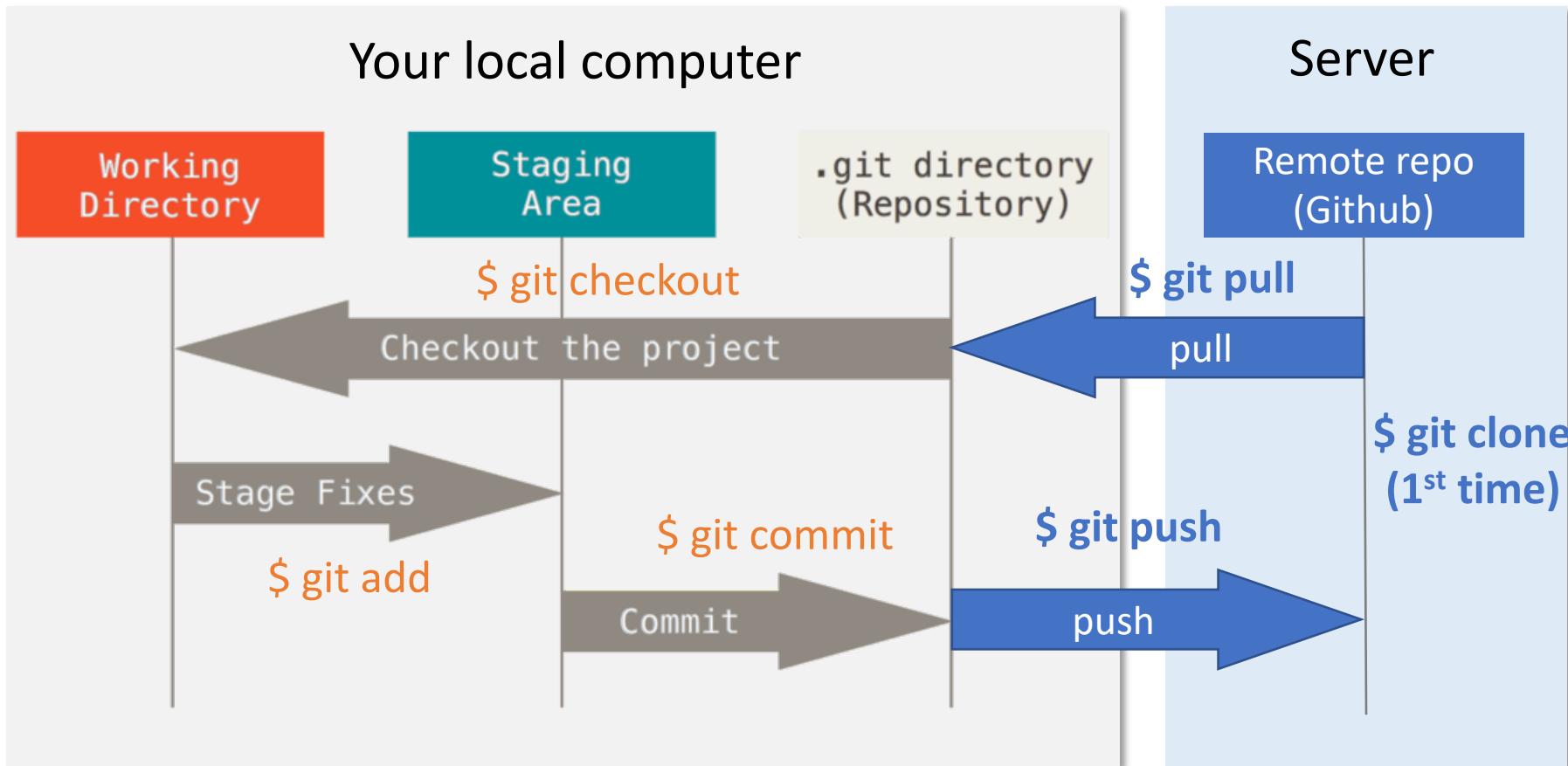


The three sections of Git

1. Working tree
2. Staging area
3. Git directory (local)



Working with remote repository



5. Write your paper – Git(hub)

- Install git
 - Windows: <https://git-scm.com/download/win>
 - Linux: \$ sudo apt install git-all
 - macOS: \$ git --version or \$ brew install git
- Git interface
 - Command line: Git for windows (<https://gitforwindows.org/>)
 - Graphical: Github Desktop (<https://desktop.github.com/>)
 - **Only use graphical after knowing how to use command line!**
- Apply for a student account on Github
 - <https://education.github.com/students>
 - To have free Github Pro
 - Supply 英文在读证明

Create a new private repository

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Cyan27 ▾

Repository name

/ example ✓

Great repository names are short and memorable. Need inspiration? How about [sturdy-telegram](#).

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: [None ▾](#)

Add a license: [None ▾](#)



Create repository

Add your collaborators

The screenshot shows the GitHub repository settings page for a private repository named "Cyan27 / example". The "Settings" tab is selected. On the left, a sidebar menu includes "Options", "Collaborators" (which is currently selected), "Branches", "Webhooks", "Integrations & services", and "Deploy keys". The main content area is titled "Collaborators" and contains the message: "This repository doesn't have any collaborators yet. Use the form below to add a collaborator." Below this is a search bar with the placeholder "Search by username, full name or email address" and a note: "You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead." At the bottom right of the search bar is a "Add collaborator" button.

Clone into your local folder

```
$ git clone https://github.com/yourname/example.git
```

The screenshot shows a GitHub repository page for the user 'Cyan27' named 'example'. The repository is private, as indicated by the 'Private' button. The main navigation bar includes links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. Below the navigation bar, there is a note: 'No description, website, or topics provided.' with an 'Edit' button. A 'Add topics' link is also present. Key statistics are displayed: 1 commit, 1 branch, 0 releases, and 1 contributor. The 'Branch: master' dropdown is set to 'master'. There is a 'New pull request' button. On the right side, there are buttons for 'Create new file', 'Upload files', 'Find file', and a prominent green 'Clone or download' button. A tooltip for the 'Clone with HTTPS' option shows the URL <https://github.com/Cyan27/example.git>. Below the repository name, there are three file entries: 'Create README.md' (with a 'Create' button), 'README.md', and another 'README.md'. At the bottom, there is a large, bolded 'example' text.

Put your paper into the git folder

- Add .gitignore
 - Include a .gitignore file before you mess up everything (<https://github.com/github/gitignore/blob/master/TeX.gitignore>)
- Add your paper's TeX files
- Often check your git repo status with
 - \$ git status
- Select the files you want to stage
 - \$ git add <filename> or \$ git add -- all
- Commit your changes to the files
 - \$ git commit -m "A summarize of your edits"
- Download updates from Github
 - \$ git pull
- Fix conflicts if any (manually fix conflicts and commit again)
- Upload to Github
 - \$ git push

How to Write a Git Commit Message

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Git Commit Message (bad)

```
$ git log
```

e5f4b49 Re-adding ConfigurationPostProcessorTests after its brief removal in r814. @Ignore-ing the testCglibClassesAreLoadedJustInTimeForEnhancement() method as it turns out this was one of the culprits in the recent build breakage. The classloader hacking causes subtle downstream effects, breaking unrelated tests. The test method is still useful, but should only be run on a manual basis to ensure CGLIB is not prematurely classloaded, and should not be run as part of the automated build.

2db0f12 fixed two build-breaking issues: + reverted ClassMetadataReadingVisitor to revision 794 + eliminated ConfigurationPostProcessorTests until further investigation determines why it causes downstream tests to fail (such as the seemingly unrelated ClassPathXmlApplicationContextTests)

147709f Tweaks to package-info.java files

22b25e0 Consolidated Util and MutableAnnotationUtils classes into existing AsmUtils

7f96f57 polishing

Git Commit Message (good)

```
$ git log
```

5ba3db6 Fix failing CompositePropertySourceTests

84564a0 Rework @PropertySource early parsing logic

e142fd1 Add tests for ImportSelector meta-data

887815f Update docbook dependency and generate epub

ac8326d Polish Mockito usage

The 7 rules of a great Git commit message

<https://chris.beams.io/posts/git-commit/>

Join the USSLAB group

Send Wenjun your Github username!

The screenshot shows the GitHub organization page for 'Ubiquitous System Security Laboratory' (USSLAB) at Zhejiang University. The page features a navigation bar with tabs for 'Repositories' (3), 'People' (8), 'Teams' (0), 'Projects' (0), and 'Settings'. Below the navigation is a search bar and filters for 'Type: All' and 'Language: All'. A green 'New' button is visible. The main content area displays three repository cards:

- DolphinAttack**: Inaudible Voice Commands. Languages: security, speech-recognition, ultrasound, audible-voice-commands. Stars: 14, Forks: 5. Updated on 12 Mar.
- CAN_OMEGA**: Forked from zjlywjh001/CAN_OMEGA. Ultimate CAN Bus tool for Car hacking. Full Featured hardware & software. Languages: Java. Stars: 6, Forks: 6. MIT license. Updated on 13 Feb 2017.
- CAN_OMEGA_Tools**: Forked from zjlywjh001/CAN_OMEGA_Tools. A Powerful cross-platform GUI tool for CAN Omega Hardware. Languages: Java. Stars: 2, Forks: 2. MIT license. Updated on 13 Feb 2017.

On the right side, there are two boxes: 'Top languages' (Java) and 'People' (8). The 'People' box shows profile icons for several team members and a 'Invite someone' button.

Advice on research habits

我的博士收获与心得

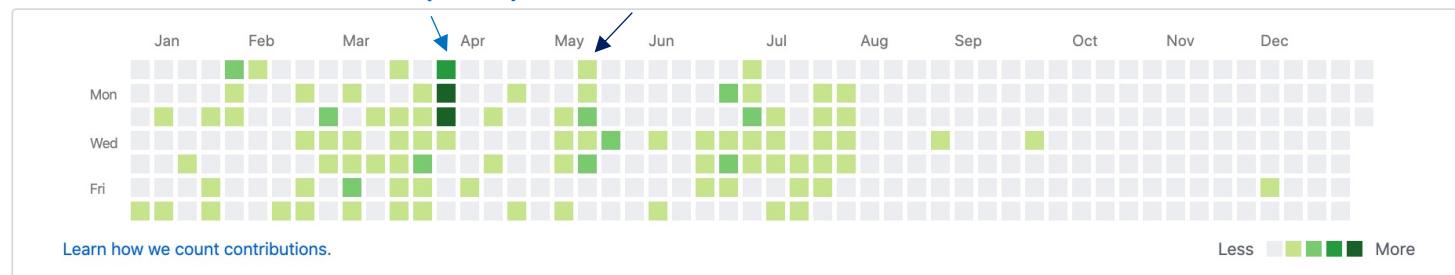
- 多读、多做、多写、多改、多投（熟能生巧、分散风险）
- 多想（多问为什么）
- 高效合作需要好的方式和方法
- 细节很重要，但要顾全大局，做好时间、精力把控
- 实现比idea更重要，完成比完美更重要
- 持续产出需要保持连惯性投入

297 contributions in 2019

SoK (end) Cafield

Contribution settings ▾

2019



276 contributions in 2018

Cuba

Infocom

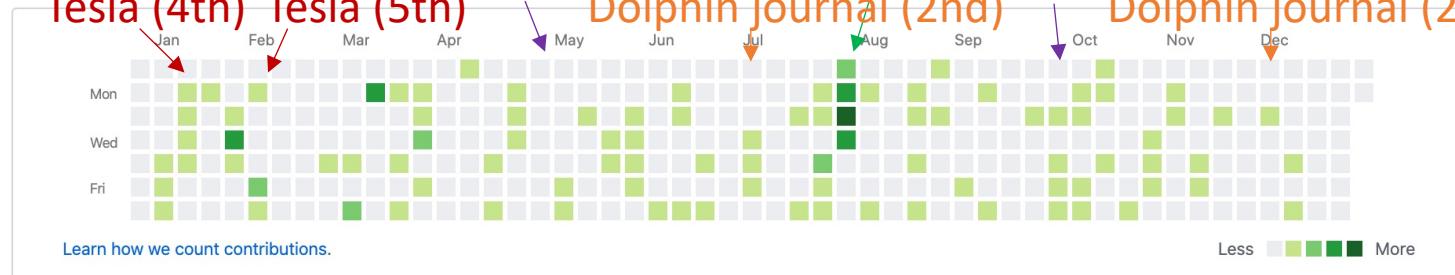
Cuba

Tesla (4th) Tesla (5th)

Dolphin journal (2nd)

Dolphin journal (2nd)

2018



401 contributions in 2017

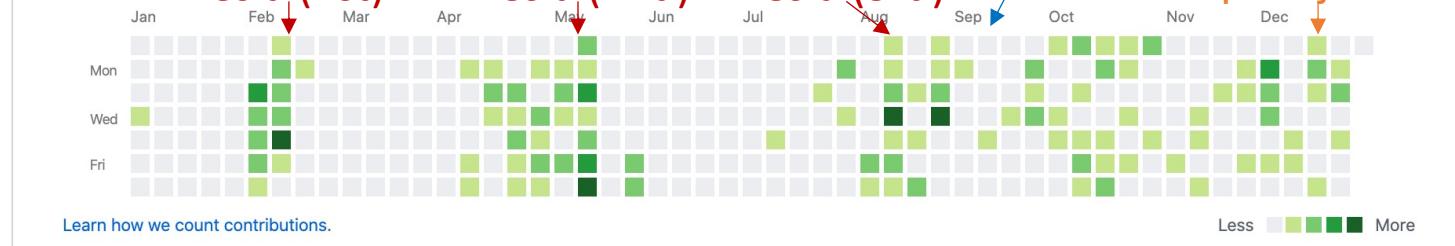
Tesla (1st)

Dolphin
Tesla (2nd)

SoK (start)

Dolphin journal (1st)

2017

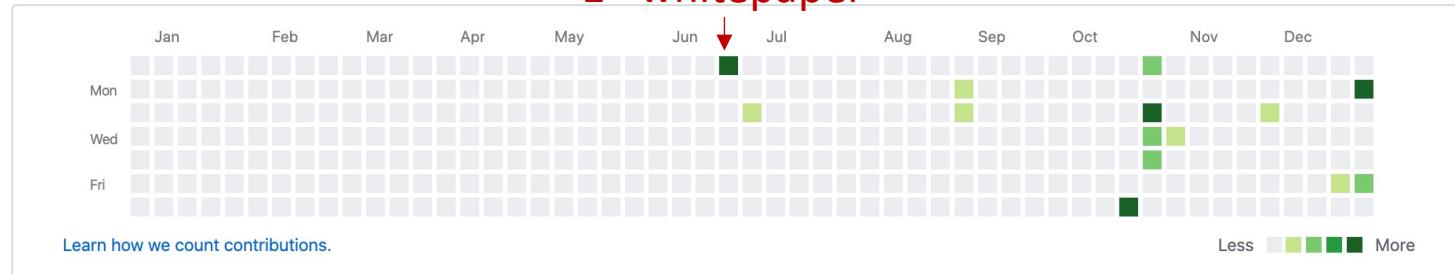


59 contributions in 2016

1st whitepaper

Contribution settings ▾

2016



Questions or Suggestions?

