
Pull requests

Key takeaways

1. A pull request proposes the commits on a feature branch for a target branch, with evidence and discussion that support a merge or decline.
2. For tm-compute, compare the frozen base with the final head using your authored cases, justified truth, and the same measurement conditions.
3. A real review anchors technical feedback to code, receives a response, and ends with an evidence-backed merge or decline.

Common misunderstandings of pull requests

Which claim would you defend right now? Choose before the evidence.

“A pull request is a copy of some files; branches are just Git jargon.”

“Better accuracy on one rack proves the final version is an improvement.”

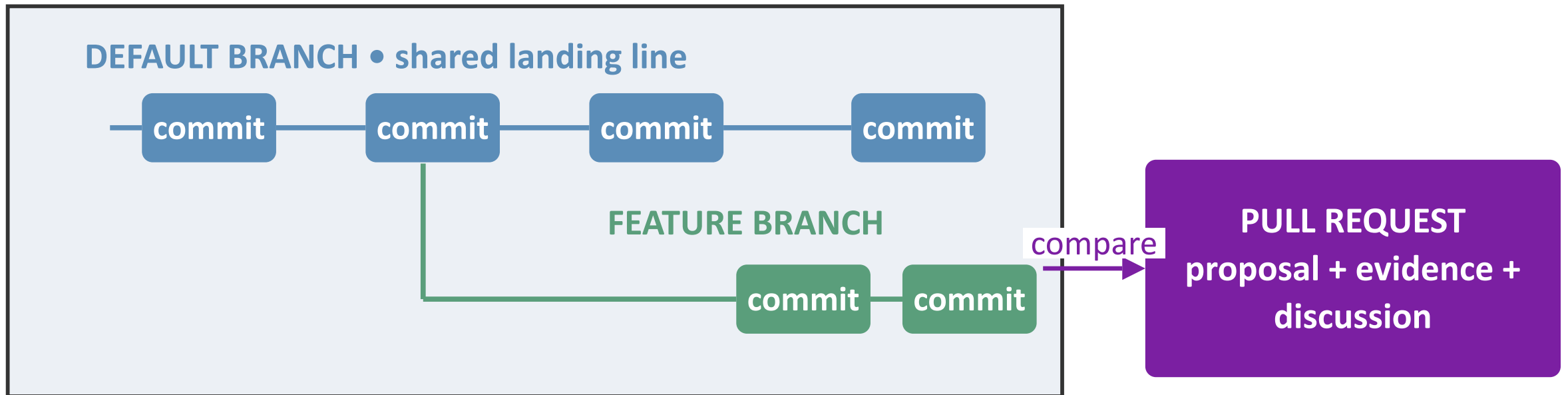
“A top-level ‘looks good’ comment counts as a technical GitHub review.”

Keep your choice. Look for the mechanism or counterexample that would make it fail.

The GitHub vocabulary fits one picture

A pull request compares commits; it does not copy a folder

REPOSITORY



BASE = target branch state • **HEAD** = final feature-branch state

Choose the target before you open the PR

GitHub calls the receiving branch base and your proposal compare

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: my-patch-1 ✓ **Able to merge.** These branches can be automatically merged.

Choose a head ref

my

my-patch-1

myarb-patch-1

my

Update CONT...

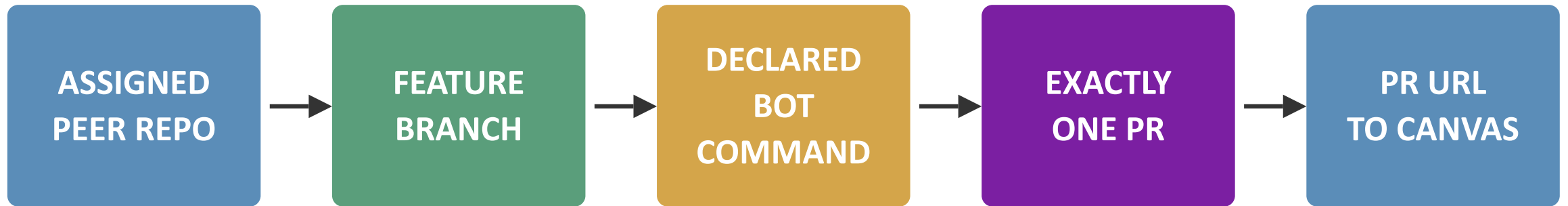
Write Prev

Leave a comment

In this GitHub form: BASE receives the change; COMPARE contains your work.

The assignment fixes the path to the PR

Work in the repository and branch that the round assigns



Never push to the peer repository's default branch.

Target the WRITEUP.md line: Bot command:

One unit leaves one durable pull-request record

The URL identifies the code, frozen commits, evidence, and discussion

Solo or pair

Open exactly one PR per unit. A pair does not open one PR per partner.

Canvas submission

Submit the PR URL, not the repository URL. Both partners submit the same PR URL.

Keep the record

Keep the PR and branch. Do not rewrite history or delete either until grades are released.

Use “tm-compute: protocol & deliverable” and PR_ROUND.md for the no-late-window rule, entry checks, timing, and staff exceptions.

One GitHub page holds the decision

New commits update the same page instead of creating a second PR

Adding sparkles #45834

Open

octocat wants to merge 6 commits into `main` from `octocats-branch`

Conversation34

Commits6

Checks57

Files changed2

+32-4

Changes from all commits

File filter

Conversations

0 / 2 files viewed

Review in codespace

Review changes

Area	What it records
Conversation	Claim, evidence, responses, and final verdict
Commits	How the feature branch developed
Checks	Automated results for the proposed head
Files changed	Exact differences from the base

source: Screenshot: GitHub pull-request page. Orange boxes identify the review tabs.

Give the reviewer one claim to verify

A focused weakness connects the code change to a measurable outcome

Weakness

Name one generalization or performance problem in the peer bot.

Direction

Explain how the change fits that bot's ranking method and resource policy.

Evidence

Show whether the same authored cases support the claim from base to head.

A reviewer should be able to finish: "This PR claims that ..."

The description is a reproducible map

Write for the author who did not watch your work

Section	Record before requesting review
Target and hypothesis	Input condition, frozen-bot diagnosis, and design fit
Baseline evidence	Authored cases, justified truth, command, environment, repetitions, and base result
Direction log	Distinct idea, rationale, commits, measurement, and result
Final proposal	Final change, supporting evidence, limit checks, and remaining risk

Keep reverted directions in the log and link their commits.

Base and head are fixed comparison points

The round keeps the start stable while your branch develops

Frozen base

The retained peer-bot commit at the round's start. Every direction compares with this same state.

Working commits

Experiments may succeed, fail, or be reverted. Their commits preserve what you tried.

Final head

The feature-branch commit the course and author evaluate at the end.

The authoritative comparison is frozen base versus final PR head.

Instapoll

Instapoll · Review · Sep 24 B

A classmate's pull request says its new allocation policy ranks racks better. The diff is small, you follow every line, and all checks are green. What has still not been checked?

- A. Nothing — small, readable, and green is what an approval means
- B. Whether the two versions were measured on the same racks and the same host
- C. Whether the branch targets the right base, which a green check never confirms
- D. Whether the commit history is clean enough to stay readable later

A fair comparison changes one thing

Base and head must face the same evidence conditions

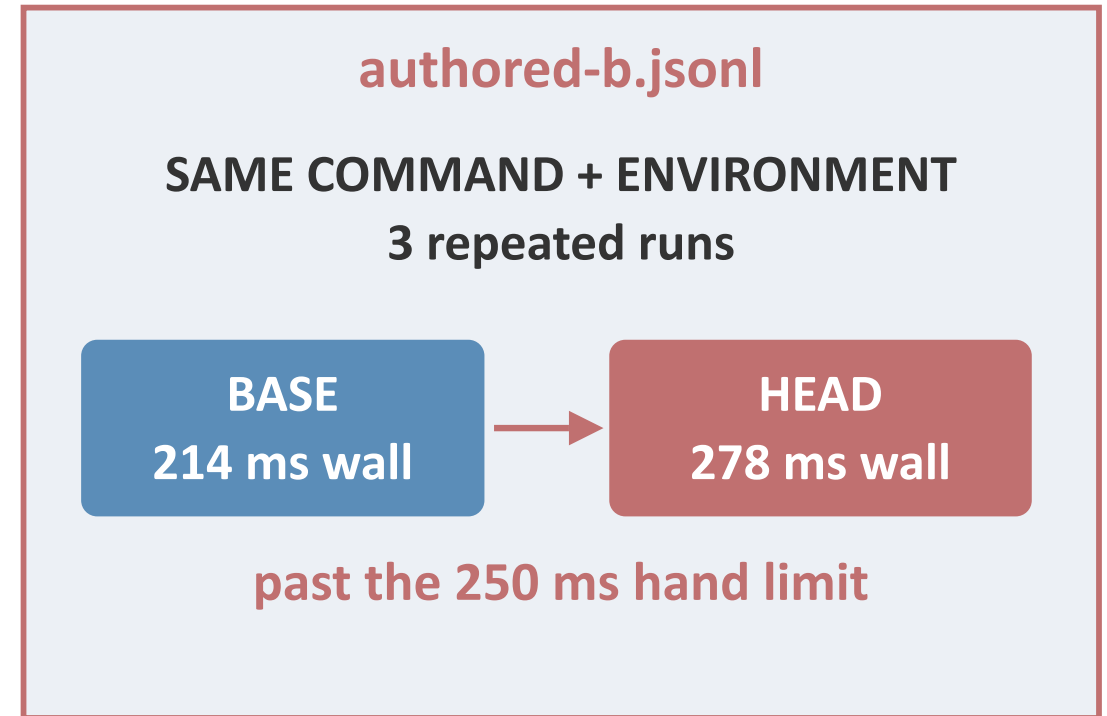
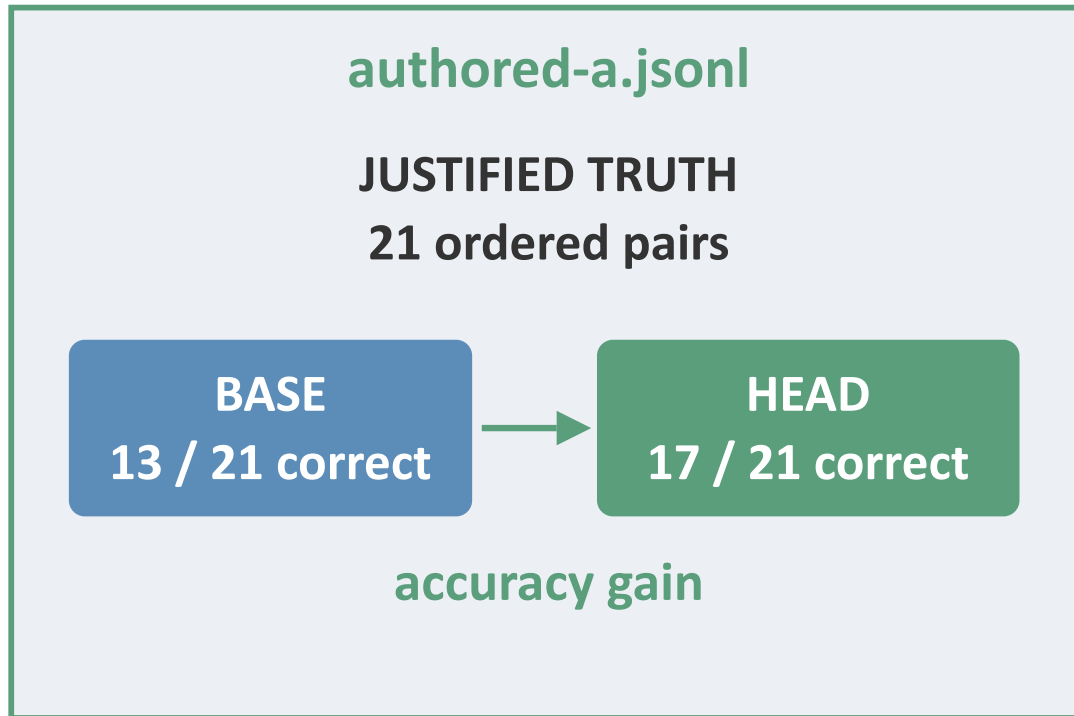
Hold constant	What to record
Authored cases	The same rack files for base and head
Expected result	How you justified the correct order
Bot launch	The command declared by the peer WRITEUP.md
Environment	The same host and relevant setup
Repetitions	The same stated repetition policy for both versions

Green checks show that checks ran; they do not establish a fair comparison.

One gain can hide one disqualifying regression

Read correctness and the rack-wide wall limit together

ILLUSTRATIVE AUTHORED EVIDENCE — NOT COURSE RESULTS



This head ranks more pairs correctly on one rack but is not a safe final head.

Two contributor paths can earn the outcome

Both paths require implemented work and fair recorded measurements

Verified improvement

The final head improves over the frozen base and remains conformant and within the assignment limits.

Two unsuccessful directions

Implement and measure two technically distinct, substantial ideas against the same frozen base.

The second path is evidence of real engineering work, not a prose fallback.

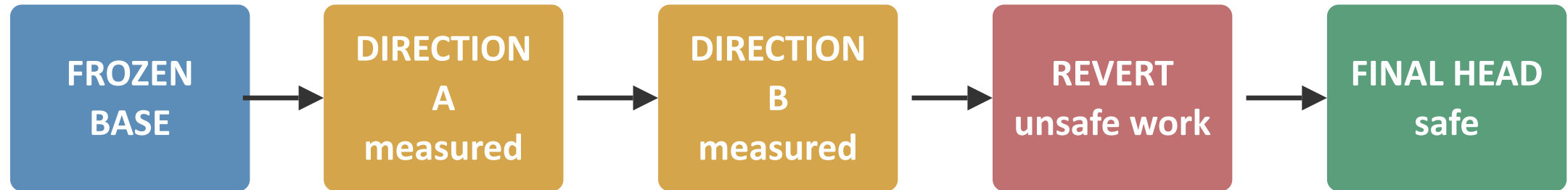
A direction changes behavior, not just a number

Two attempts must be technically distinct

Substantial direction	Does not become a second direction
Change the ranking method	Nudge another constant in the same method
Change resource allocation	Repeat the same implementation
Change behavior on meaningful inputs	Cosmetic refactor or comments
Implement, commit, and measure it	Describe an idea without code

Preserve failed work; repair the final head

History proves the attempts while the endpoint stays safe



The PR keeps commits and measurements for both attempts.

The final head still runs, speaks tmc.v1, stays within limits, and performs no worse than the frozen base.

Review the claim before the lines

Each pass answers a different decision question

Pass	Action	Decision
1. Understand	Read the claim and evidence	Is the goal specific?
2. Inspect	Read the final combined diff	Does code match the idea?
3. Verify	Repeat the key comparison	Does evidence support it?
4. Respond	Submit one GitHub review	Accept or request action?

Check correctness, resource limits, and ranking risk before style preferences.

Files changed is the final combined proposal

Read file by file and re-check a file after later commits change it



- Use the file tree and filters to move through the diff.
- Mark a finished file Viewed; GitHub clears that state after a new edit.
- Inspect base-to-head behavior, not only the newest commit.

Anchor technical feedback where it applies

Use an inline comment or name the exact code location

2

.github/workflows/codeql-analysis.yml

@@ -13,7 +13,7 @@ name: "CodeQL"

1313

1414on:

1515push:

1616- branches: [main, add-emoji, add-read-file-1, update-readme-5]

1616+ branches: [main, add-emoji, branch-, branch-

1717pull_request:

1818# The branches below must be a subset of the

1919branches: [main]

Weak	Reviewable
Too slow	bot.py:84 raises per-card work, but the limit covers the whole rack. Please repeat the wall-time comparison.
Looks wrong	sim.py:51 drops head position from the repeat key. Please test a justified case where only the head position differs.

source: Screenshot: GitHub inline-comment control. The highlighted plus anchors a comment to the diff.

A technical comment connects three parts

Ask for evidence when evidence can settle the disagreement

Observation

Name the exact line, behavior, or missing comparison you found.

Impact

Explain the correctness, performance, budget, or ranking risk.

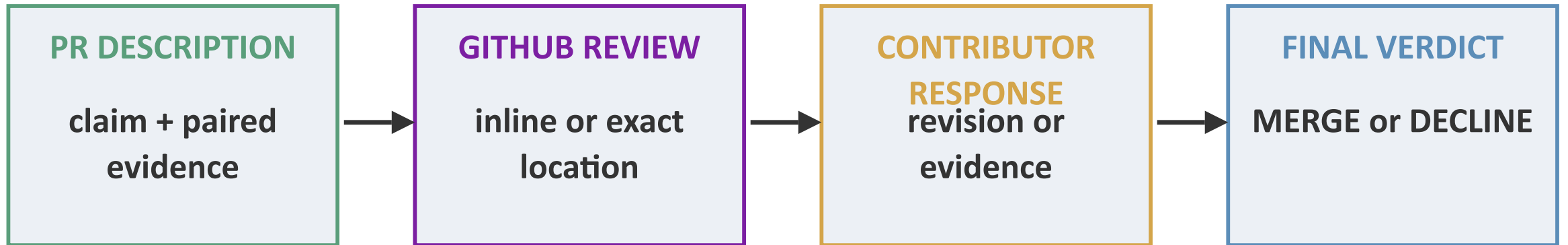
Action

Request a specific revision or measurement that can resolve the risk.

A generic approval or objection supplies no technical decision evidence.

The conversation must reach a verdict

Each role adds the artifact it controls



Every arrow is visible on the same pull-request page.

Merge and decline are both valid maintenance decisions

The author follows the final evidence, not social pressure

Final evidence	Author action
Safe head; claim survives review	MERGE and name the evidence that supports it
Safe head; benefit is not established	DECLINE and explain the uncertainty or lack of benefit
Head is unsafe or worse than base	DECLINE and point to the failed check or measurement

A supported decline can earn full author credit. The author never has to merge a distrusted change.

Key Takeaways

1. Work in a feature branch of the assigned peer repository, never its default branch. Open exactly one PR per unit against the bot command declared in WRITEUP.md.
2. Compare frozen base with final head on authored cases whose expected order you justify. Keep cases, command, environment, and repetitions the same, and read accuracy together with the 250 ms hand limit.
3. Preserve failed directions and discussion, keep the final head safe, and complete the visible chain: anchored review, contributor response, then an evidence-backed merge or decline.