

Use of AI in Code Reviews

Joe Osborn, BNL
April 22, 2026

Disclaimer

- I am not an expert
- These slides are based on personal experience with sPHENIX and Acts
- There are many tools out there that I don't have experience with (and likely the number is growing)

Code Rabbit

- Code rabbit is a free AI review tool that can be integrated into any repository
- Provides detailed review information of any PR. Jin Huang integrated into sPHENIX core software
- Pros:
 - Free
 - Easily integrated
 - Definitely finds bugs (from experience)
- Cons:
 - Extremely verbose. Have not determined a way to reduce the amount of text
 - Free version does have limitations

Code Rabbit: Example

- Example: 10 line commit that effectively adds a setter function
- Code rabbit provides a verbose summary
- For this particular PR, it has no comments

The image shows a side-by-side diff of two code files. The top window shows the diff for `offline/packages/trackreco/PHMicromegasTpcTrackMatching.cc`. The bottom window shows the diff for `offline/packages/trackreco/PHMicromegasTpcTrackMatching.h`. Both windows show a comparison between the original code (left) and the modified code (right). In the .cc file, line 892 shows the removal of a hard-coded string "TRKR_CLUSTER" and its replacement with a variable `_clustermap_name`. Line 897 shows the removal of a hard-coded error message and its replacement with a message that includes the variable `_clustermap_name`. In the .h file, line 45 shows the addition of a new setter function `set_clustermap_name`. Line 92 shows the addition of a default value for `_clustermap_name`.

```
@@ -889,12 +889,12 @@ int PHMicromegasTpcTrackMatching::GetNodes(PHCompositeNode* topNode)
889     }
890     else
891     {
892 -     _cluster_map = findNode::getClass<TrkrClusterContainer>(topNode,
892 +     _cluster_map = findNode::getClass<TrkrClusterContainer>(topNode,
      "TRKR_CLUSTER");
      _clustermap_name);
893     }
894
895     if (!_cluster_map)
896     {
897 -     std::cerr << PHWHERE << " ERROR: Can't find node TRKR_CLUSTER" <<
897 +     std::cerr << PHWHERE << " ERROR: Can't find node " << _clustermap_name <<
      std::endl;
      std::endl;
898     return Fun4AllReturnCodes::ABORTEVENT;
899     }
900
...
@@ -42,7 +42,7 @@ class PHMicromegasTpcTrackMatching : public SubsysReco
42     void set_pt_cut( const float pt) { _pt_cut = pt; }
43     void set_dphi_cut( const float dphi) { _dphi_cut = dphi; }
44     void SetIteration(int iter) { _n_iteration = iter; }
45 -
45 + void set_clustermap_name(const std::string& name) { _clustermap_name = name; }
46     void zeroField(const bool flag) { _zero_field = flag; }
47     int Init(PHCompositeNode* topNode) override;
48     int InitRun(PHCompositeNode* topNode) override;
...
@@ -89,6 +89,8 @@ class PHMicromegasTpcTrackMatching : public SubsysReco
89     TrackSeedContainer* _tpc_track_map{nullptr};
90     TrackSeedContainer* _si_track_map{nullptr};
91
92 +     std::string _clustermap_name = "TRKR_CLUSTER";
93 +
```

Code Rabbit: Example

- Example: 10 line commit that effectively adds a setter function
- Code rabbit provides a verbose summary
- For this particular PR, it has no comments

Summary

Motivation / Context

This PR adds configurability to the TPOT-TPC track matching code to enable selection of different cluster maps. Previously, the cluster map was hardcoded to "TRKR_CLUSTER", limiting the flexibility of track matching workflows. By exposing a setter method, downstream users can now select appropriate cluster maps for their analysis, ensuring all necessary information is available in the seed DST for track fitting and analysis.

Key Changes

- Added public setter method `set_clustermap_name(const std::string& name)` to allow runtime configuration of the cluster map name
- Modified cluster map lookup in `GetNodes()` to use the configurable `_clustermap_name` member variable instead of hardcoded "TRKR_CLUSTER" string
- Updated error messages to reflect the configured cluster map name
- Default value remains "TRKR_CLUSTER" for backward compatibility

Potential Risk Areas

- **Reconstruction behavior:** If users specify a different cluster map, the track matching results may differ from the default case. Ensure selection criteria and validation of available cluster maps are well-documented.
- **Configuration consistency:** No apparent validation that the specified cluster map actually exists in the DST; incorrect configuration could lead to silent failures or crashes. Consider adding validation or informative error handling.
- **Backward compatibility:** Default value preserves existing behavior, but users must explicitly configure non-default cluster maps.

Possible Future Improvements

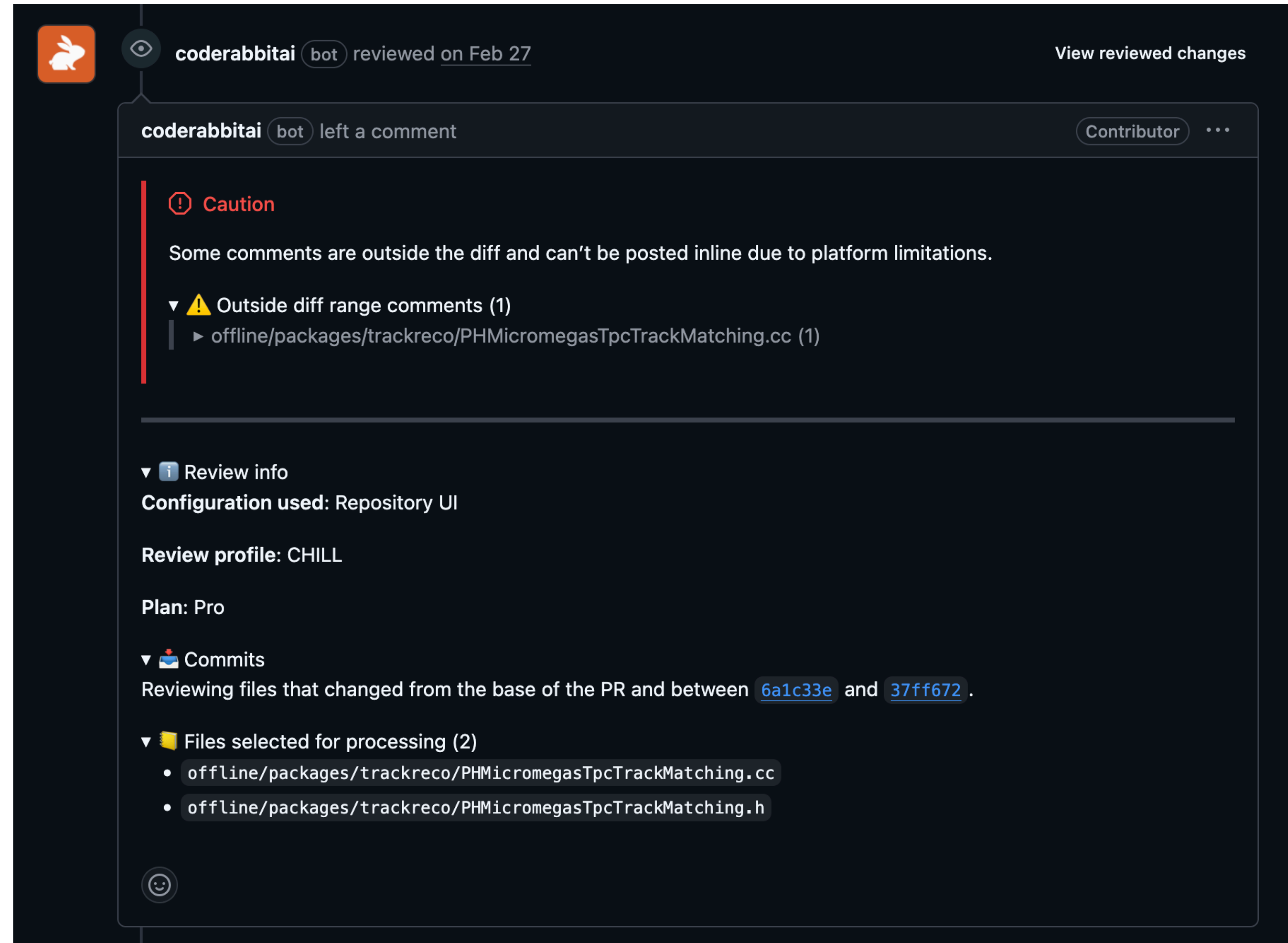
- Add validation to check that the configured cluster map exists in the input file
- Provide documentation or helper functions for discovering available cluster maps
- Consider logging/configuration validation to help users debug cluster map selection issues

Note: This summary is AI-assisted; please apply your best judgment when reviewing the actual code changes.



Code Rabbit: Example

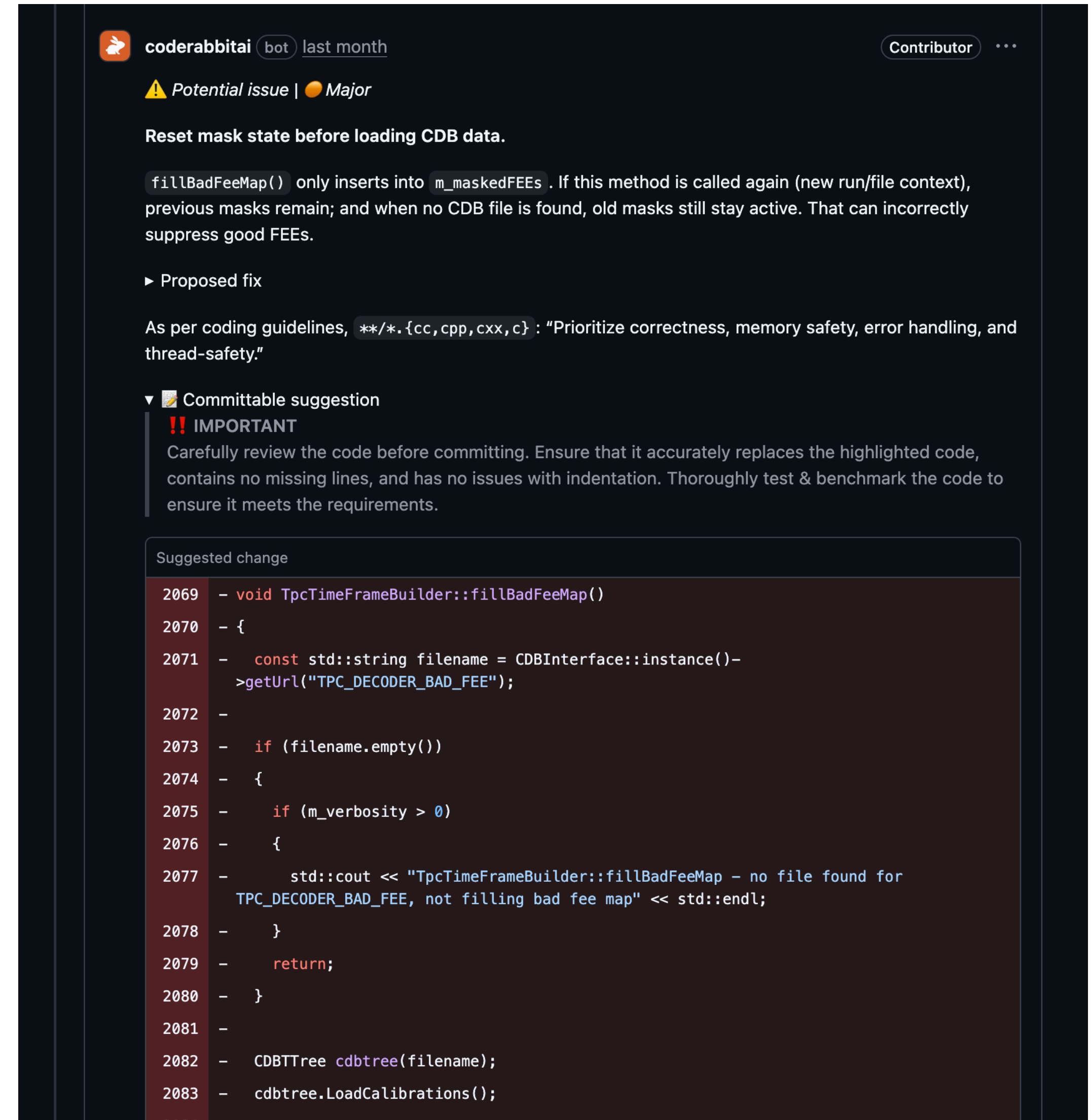
- Example: 10 line commit that effectively adds a setter function
- Code rabbit provides a verbose summary
- For this particular PR, it has no comments



The screenshot shows a Code Rabbit review interface. At the top, the Code Rabbit logo (a rabbit) is visible next to the text "coderabbitai bot reviewed on Feb 27". A link "View reviewed changes" is in the top right. Below this, a comment box shows "coderabbitai bot left a comment" with a "Contributor" button and a menu icon. The comment content includes a "Caution" icon and text: "Some comments are outside the diff and can't be posted inline due to platform limitations." Below this is a section for "Outside diff range comments (1)" with a sub-item: "offline/packages/trackreco/PHMicromegasTpcTrackMatching.cc (1)". Further down, there are sections for "Review info" (Configuration used: Repository UI, Review profile: CHILL, Plan: Pro), "Commits" (Reviewing files that changed from the base of the PR and between 6a1c33e and 37ff672), and "Files selected for processing (2)" with two file paths: "offline/packages/trackreco/PHMicromegasTpcTrackMatching.cc" and "offline/packages/trackreco/PHMicromegasTpcTrackMatching.h". A smiley face icon is at the bottom left of the comment box.

Code Rabbit: Bad Example

- In other cases, it will provide committable suggestions for how to improve the code
- Requires human interpretation - sometimes, the suggestions are unaware of important context
- In this example, the comment doesn't make sense given context of production infrastructure

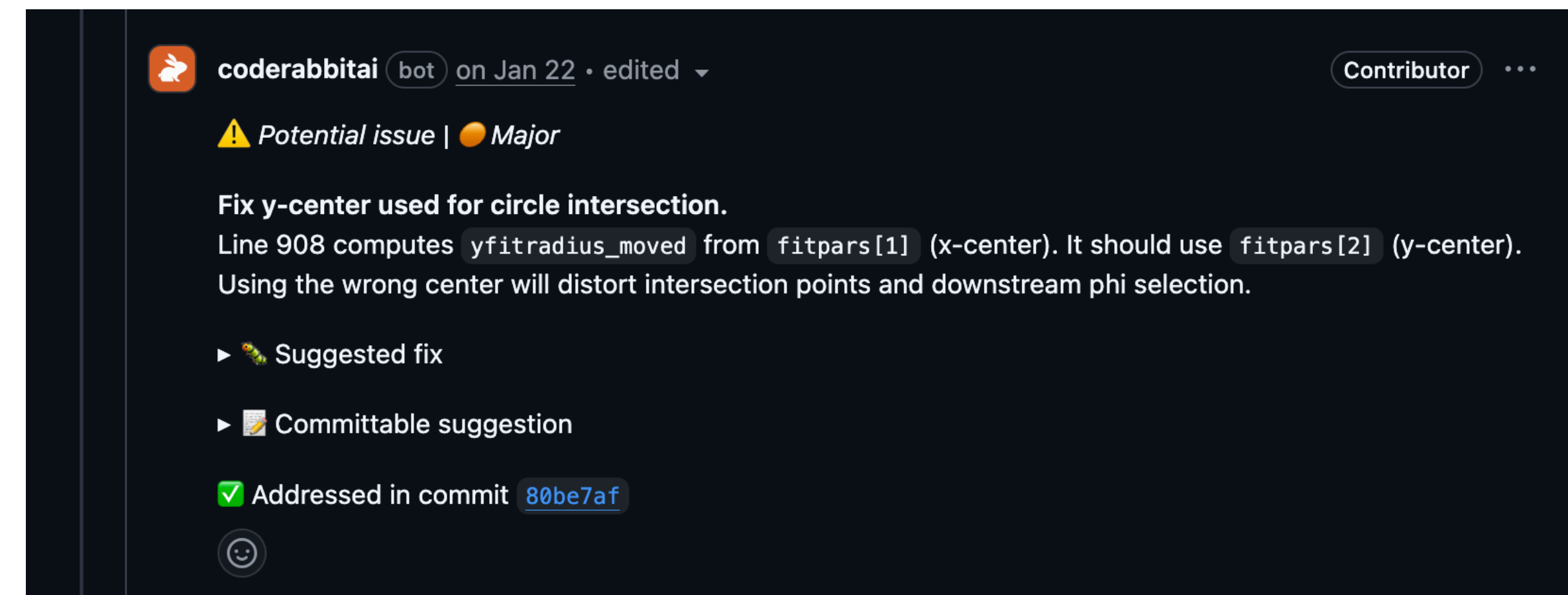


The screenshot shows a GitHub issue comment from the bot 'coderabbitai'. The comment is titled 'Reset mask state before loading CDB data.' and is marked as a 'Potential issue' and 'Major'. The text explains that the `fillBadFeeMap()` method only inserts into `m_maskedFEES` and that previous masks remain active if no CDB file is found. A 'Proposed fix' is provided, which is a code change to the `fillBadFeeMap()` method. The code change includes a comment that says: "Prioritize correctness, memory safety, error handling, and thread-safety." The code change is marked as 'Committable suggestion' and 'IMPORTANT'. The code change is as follows:

```
2069 - void TpcTimeFrameBuilder::fillBadFeeMap()
2070 - {
2071 -     const std::string filename = CDBInterface::instance()-
2072 -         >getUrl("TPC_DECODER_BAD_FEE");
2073 -
2074 -     if (filename.empty())
2075 -     {
2076 -         if (m_verbosity > 0)
2077 -         {
2078 -             std::cout << "TpcTimeFrameBuilder::fillBadFeeMap - no file found for
2079 -                 TPC_DECODER_BAD_FEE, not filling bad fee map" << std::endl;
2080 -         }
2081 -         return;
2082 -     }
2083 -     CDBTTree cdbtree(filename);
2084 -     cdbtree.LoadCalibrations();
```

Code Rabbit: Good Example

- In other cases, it finds actual bugs
- In this example, it found a real bug in someone's PR. Prompted them to go and fix it
- “Needle in the haystack” type of situation - generates a lot of text, but when digging through the text the rabbit does identify bugs when they are present



Code Rabbit: docstrings

- Code rabbit can be authorized to auto-generate documentation for a PR
 - Auto generated commits with comments only
- Commenting code can be a very contentious subject - some feel code should be overly commented, others feel code should comment itself
- Up for discussion - but the functionality exists, and it provides good (sometimes verbose) commented documentation

Code Rabbit: Experiences

- Experiences from implementing code rabbit in sPHENIX core reconstruction/simulation software package
 - PR submitters (may) use it to debug their code
 - Reviewers (may) use it as a tool, but in general the limited pool of reviewers is still looking at the code itself
 - Many PR submitters ignore the rabbit text entirely (not always a bad thing)
- Acts experience: code rabbit created more noise than was helpful. They also ran into limitations of free service

Conclusions

- AI reviews do not replace human reviews
- Nonetheless they can be a useful tool for debugging, both from a PR submitter and PR reviewer standpoint
 - In my experience, people do at least scan through code rabbits comments. Sometimes prompts changes to be made
 - Others with GithubPro accounts request copilot reviews
- They come with downsides - can be very verbose
 - Browsing around - some articles suggest they increase overall review time based on implementations in industry (more material for a reviewer to look through, which is sometimes not useful)
- There are other tools out there - but very few are free. I imagine paid tools have some better features
 - Tools can be used through personal accounts. For example, copilot GithubPro account, Claude, etc.