

AI-Assisted Code Review

Alexey Rybalchenko

Software Development for Experiments (SDE) group,
GSI Helmholtz Centre for Heavy Ion Research

HSF Seminar - AI-assisted software tools

CERN, June 4, 2025



Who we are

GSI Darmstadt, FAIR, SDE group



GSI Helmholtz Centre for Heavy Ion Research

Accelerator facility for research purposes (physics, biology)
in Darmstadt, Germany, since 1969



Facility for Antiproton and Ion Research (FAIR)

International accelerator facility for the research with antiprotons and ions,
under construction since 2017

Software Development for Experiments (SDE) Group

The SDE group (7 people) develops and maintains common scientific software for the physics experiments in close collaboration with the experiment groups and High Energy and Nuclear Physics community.



<https://github.com/FairRootGroup>

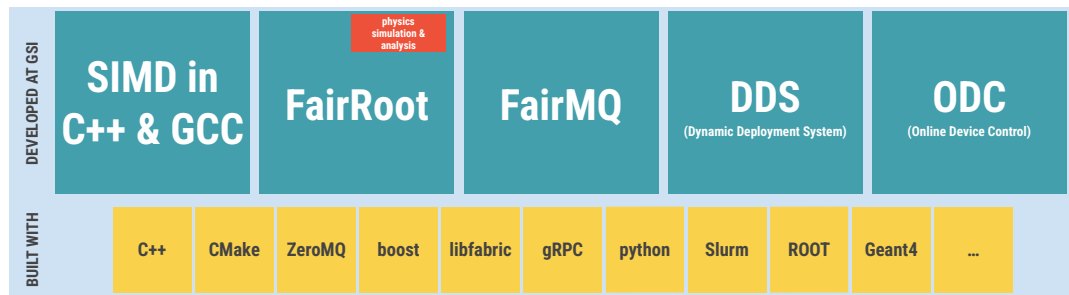


<https://github.com/GSI-HPC>

AI-Assisted Code Review

in the GSI SDE group

- Using **CodeRabbit AI** code review (based on Large language Models (LLMs)) since ~1.5 years to assist with code review on several production projects.
- In parallel, since ~1 year, developing an open source LLM code review tool **Pearbot** for use with open-weights models & to gain experience with LLM tooling.



Software Development for Experiments (SDE) Group

The SDE group (7 people) develops and maintains common scientific software for the physics experiments in close collaboration with the experiment groups and High Energy and Nuclear Physics community.



<https://github.com/FairRootGroup>



<https://github.com/GSI-HPC>

CodeRabbit

Overview

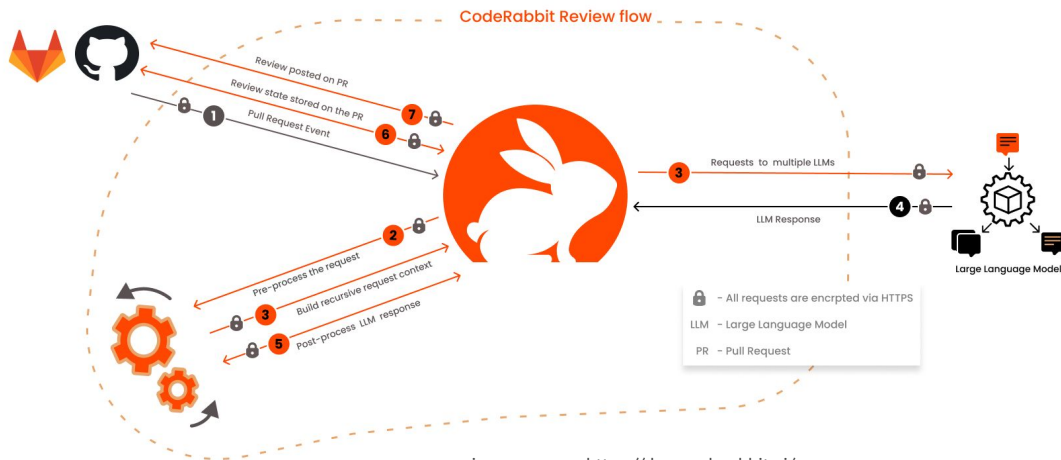
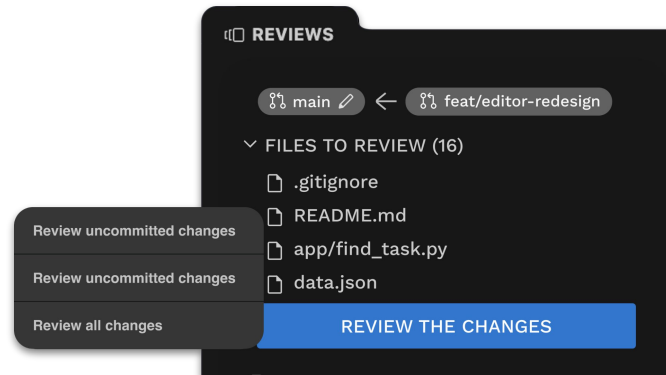


image source: <https://docs.coderabbit.ai/>



<https://coderabbit.ai/>

- LLM-based Pull Request reviewer via a Github/Gitlab App.
- Previously open source, now a closed project.
- Combination of commercial models.
- Free to use for open source projects.
- New (May 14, 2025): VS Code plugin to provide reviews before code goes to repository.




CodeRabbit


Pre-processing & post-processing

Review quality can be significantly improved by providing additional contextual information, relevant to the submitted changes. Among obvious things are PR text, commit messages, comments, relevant coding guidelines, parts of relevant code base, etc.


Details about what exactly is used and how are not revealed by CodeRabbit.

Assess Linked Issues 


Generate an assessment of how well the changes address the linked issues in the walkthrough.

Related Issues 


Include possibly related issues in the walkthrough.

Related PRs 

Include possibly related pull requests in the walkthrough.

Web Search 


Enable the web search integration.

Path Instructions  Path Instructions

Provide specific additional guidelines for code review based on file paths.

Learnings

By opting in, CodeRabbit will utilize and store insights from your interactions to enhance its learning over time. This process allows CodeRabbit to deliver increasingly refined and personalized assistance. Below, you'll find learnings generated across various repositories.

Similarity Search Top K 

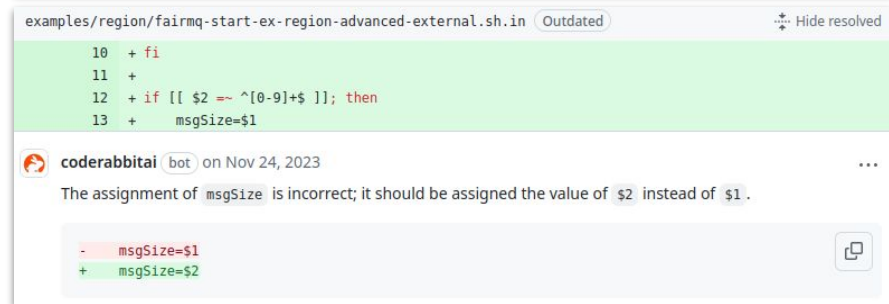
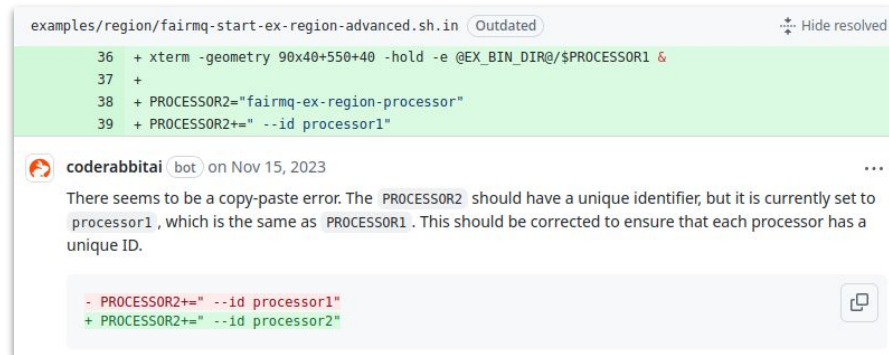
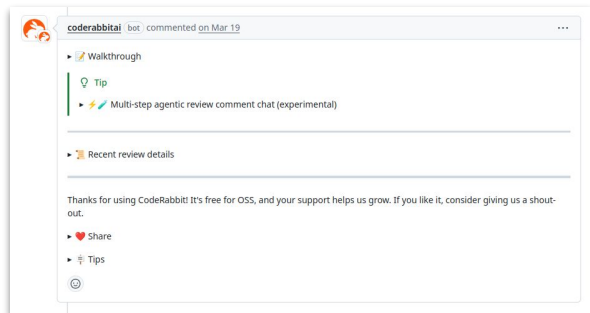
- The `getContainer` method in `FairModule` needs further investigation regarding its handling of potential null return values.
- `Data` in `FairVolumeList` can only contain `FairVolume` objects, eliminating the need for type casting error handling.
- `PACKAGE_PROJECT_CMAKEMOD_DIR` is defined by the call to `configure_package_config_file()` in the CMakeLists.txt file.
- The `SetCurrentTrack` method in `FairGenericStack` centralizes the responsibility of setting the current track, promoting consistency and maintainability.

and more...

CodeRabbit

... and LLM-based review in general. Our experience

- + Good at identifying logical errors early, that other automated tools or even human review may overlook.
- + Unbiased, 24/7, scalable, multi-lingual, customizable, with a broad knowledge base, can be kept up-to date with new data.
- + Can learn from: related issues, related PRs, previous interactions, web search.
- May produce unnecessary output, when no actionable changes are necessary or such are not deduced by the model.
- Limited understanding of complex projects.
- Potential for false positives, flagging issues that aren't actually problematic.
- Commercial project, may include some promotional output:

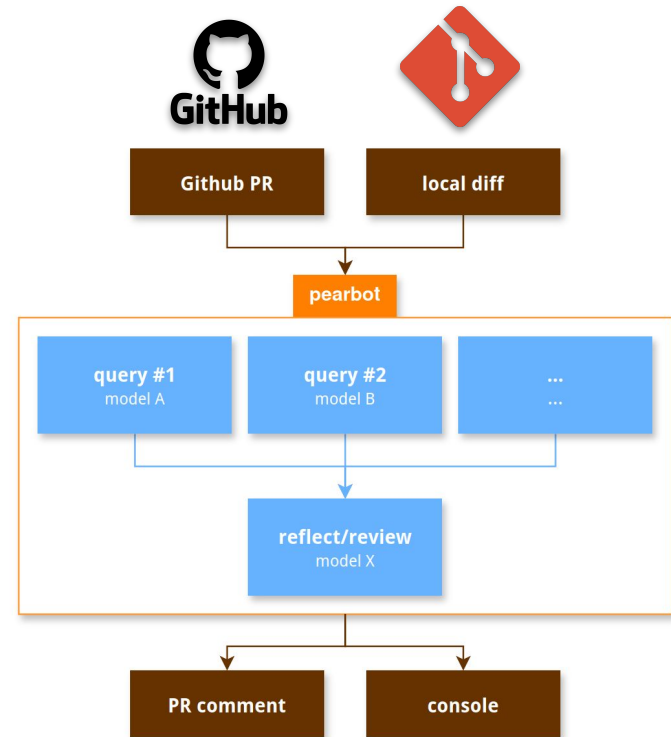


Pearbot

Overview

 <https://github.com/GSI-HPC/pearbot>

- GitHub App for reviewing Pull Requests.
- Local execution mode for diffs or annotated commits.
- LLM ensemble approach for improved results.
- Customizable model(s).
- Execution on low-end hardware and/or without GPU.
- Customizable prompt(s).



Pearbot

Usage

As a GitHub App:

```
python pearbot.py --server
```

Analyze a local diff file:

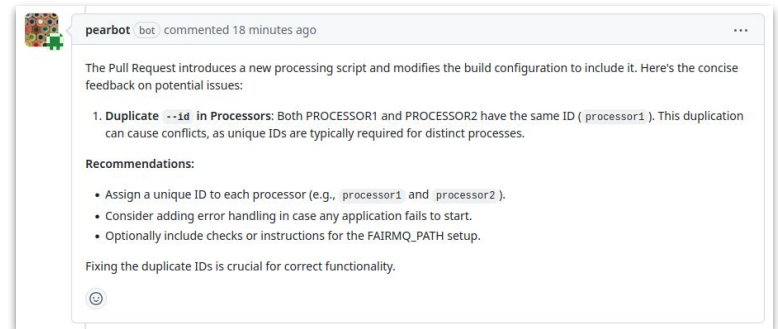
```
python pearbot.py --diff path/to/your/diff/file
```

Or pipe a diff directly:

```
git diff | python pearbot.py
```

Generate detailed output with commit messages, e.g.:

```
git format-patch HEAD~3..HEAD --stdout | python pearbot.py
```



Pearbot

Backend

ollama: open-source LLM server, written in Go, backed by **llama.cpp** (C++)

- Efficient serving of large language models
- CPU/GPU/CPU+GPU hybrid inference to partially accelerate models larger than the total VRAM capacity
- Supports many model architectures: deepseek2, llama, gemma2, qwen2, ...
- Support for multitude of model quantization techniques for faster inference and reduced memory use
- Usage Metrics

```
Model: deepseek-r1:70b
Family: llama, Format: gguf
Parameter Size: 70.6B, Quantization: Q4_K_M
Context Length: 131072
Prompt tokens: 325
Tokens generated: 209
Total tokens: 534
Speed: 18.53 tokens/second
Generation time: 11.28 seconds
Total duration: 11.57 seconds
```



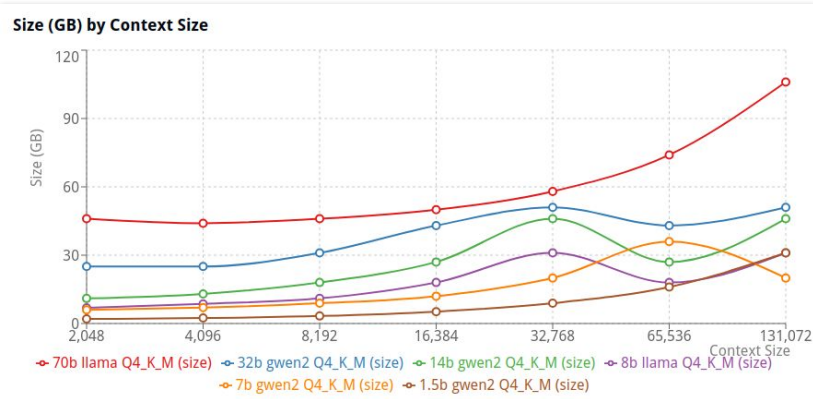
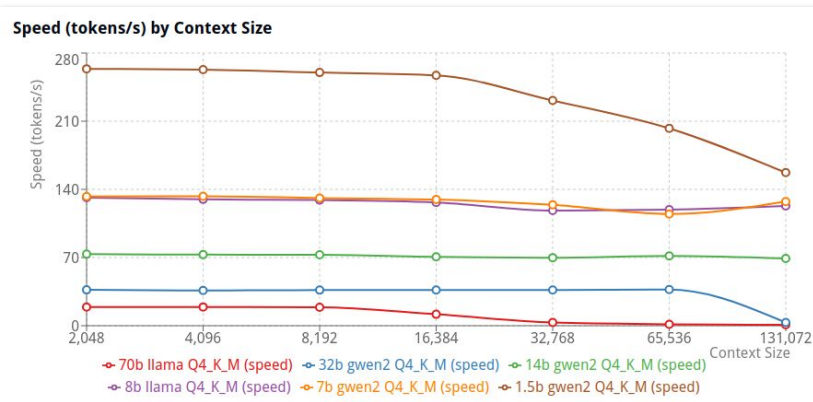
ollama has a very minimal feature set when it comes to things like:

- queue visibility
- additional info (how many tokens does this prompt take?)
- multi-server setups with “big” GPUs

Alternatives are being exploring, e.g. **vLLM**

ollama

context size performance impact



Common gotcha with ollama is the default context length setting of 2048 tokens.

Comparing the performance of different distillations of DeepSeek-R1 with varying context size:

- generation speed
- model size in memory
- GPU usage

On Nvidia RTX 6000 Ada Generation, 48GB VRAM

Matrix View: GPU Usage (%)

Model	2,048	4,096	8,192	16,384	32,768	65,536	131,072
70b llama	100	100	100	98	86	67	47
32b gwen2	100	100	100	100	100	100	72
14b gwen2	100	100	100	100	100	100	100
8b llama	100	100	100	100	100	100	100
7b gwen2	100	100	100	100	100	100	100
1.5b gwen2	100	100	100	100	100	100	100

Pearbot

Quality improvements over the base model

1. Multi-Model Initial Reviews:

- Multiple LLMs (ensemble) generate initial code reviews
 - “generate independent thoughts” that may touch different aspects of the problem.

2. Reflection^{[1][2]} by a “decider” model:

- A separate, potentially more advanced model analyzes the initial reviews.
- Refines the generated feedback, rejects potentially less impactful comments.
- It prioritizes the most important issues and suggestions (prompt-dependent).

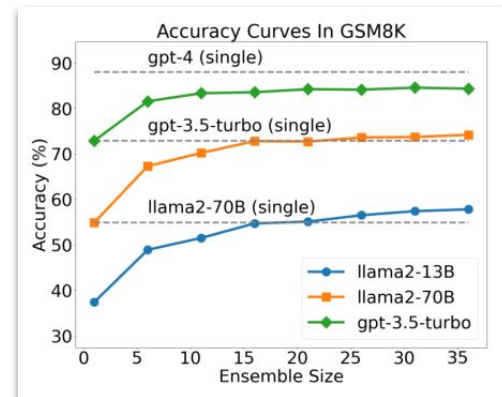
3. Prompt improvements:

- Specific & useful code review examples.
- Examples include Chain-of-Thought^[3] type of reviews, that include some reasoning why the suggestions would be good.

[1] Madaan, Aman, et al. “Self-refine: Iterative refinement with self-feedback.” *Advances in Neural Information Processing Systems* 36 (2024). <https://doi.org/10.48550/arXiv.2303.17651>

[2] Shinn, Noah, et al. “Reflection: Language agents with verbal reinforcement learning.” *Advances in Neural Information Processing Systems* 36 (2024). <https://doi.org/10.48550/arXiv.2303.11366>

[3] Wei, Jason, et al. “Chain-of-thought prompting elicits reasoning in large language models.” *Advances in neural information processing systems* 35 (2022). <https://doi.org/10.48550/arXiv.2201.11903>



Accuracy of multi-agent approach Grade School Math 8K problems.

image from: Li, Junyou, et al. “More agents is all you need.” *arXiv preprint arXiv:2402.05120* (2024).

→ **Good results with smaller PRs (even with small quantized models). Focus suffers on larger PRs. Issues when hitting context size limits.**

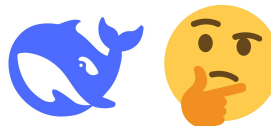
→ **DeepSeek-R1’s approach mostly overshadows these optimizations**

Trained to generate “thinking tokens” – reflecting on the problem from different perspectives.

→ **Similar to:**

OpenAI o1 or newer (commercial, thought tokens hidden)

Claude Sonnet 3.7 or newer (commercial, thought tokens visible & configurable)



Pearbot

TODO list & general improvement ideas

- Inline comments
- Improve handling with large PRs/commits: context size limits
- Additional context:
 - related issues
 - code history
 - experience from past interactions

Balance additional context with keeping the focus on the task - avoid distractions.

- Rejection of useless output, e.g. for automated reviews no found issues should produce no comments, but only a green GitHub checkmark.
- Balance num_ctx & max_tokens
- Balance larger & smaller models for different tasks → cost/efficiency optimization
- Deployment with larger models



<https://www.greptile.com/blog/make-llms-shut-up>

How to Make LLMs Shut Up

Written by **Daksh Gupta** · December 18, 2024

✘ prompting

✘ LLM-as-a-judge

✔ clustering in a vector database

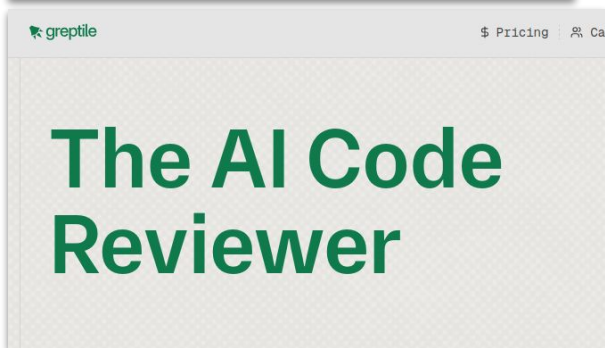
AI Code Review

Related projects



<https://www.greptile.com>

closed source



<https://www.getpanto.ai/>

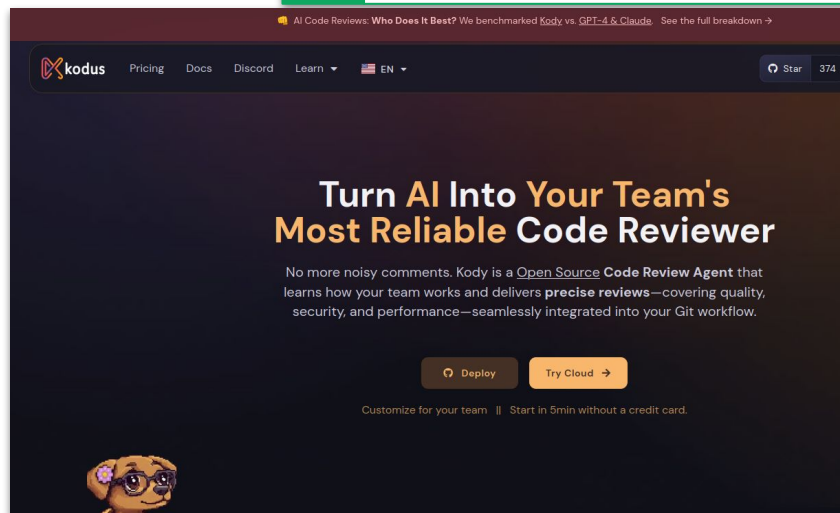
closed source

Category	Panto AI	CodeRabbit
Critical Bugs	12	10
Refactoring	14	8
Performance Optimization	5	1
Validation	0	8
Nitpick	3	13
False Positive	4	2
Total	38	42

open source,
since April 2025



<https://kodus.io/>



AI Code Review

Conclusion

- LLMs are improving fast
- AI reviews are helpful, with (decreasing number of) downsides
- Tools development & experience ► benchmarks
- Open-weights LLMs are already viable
- Biggest challenge: providing enough context for the most helpful review, *while* keeping the focus
- Can be very resource-hungry
- With on-premises execution, balance where possible/needed:
 1. quantization
 2. smaller models (fewer parameters)
 3. context size