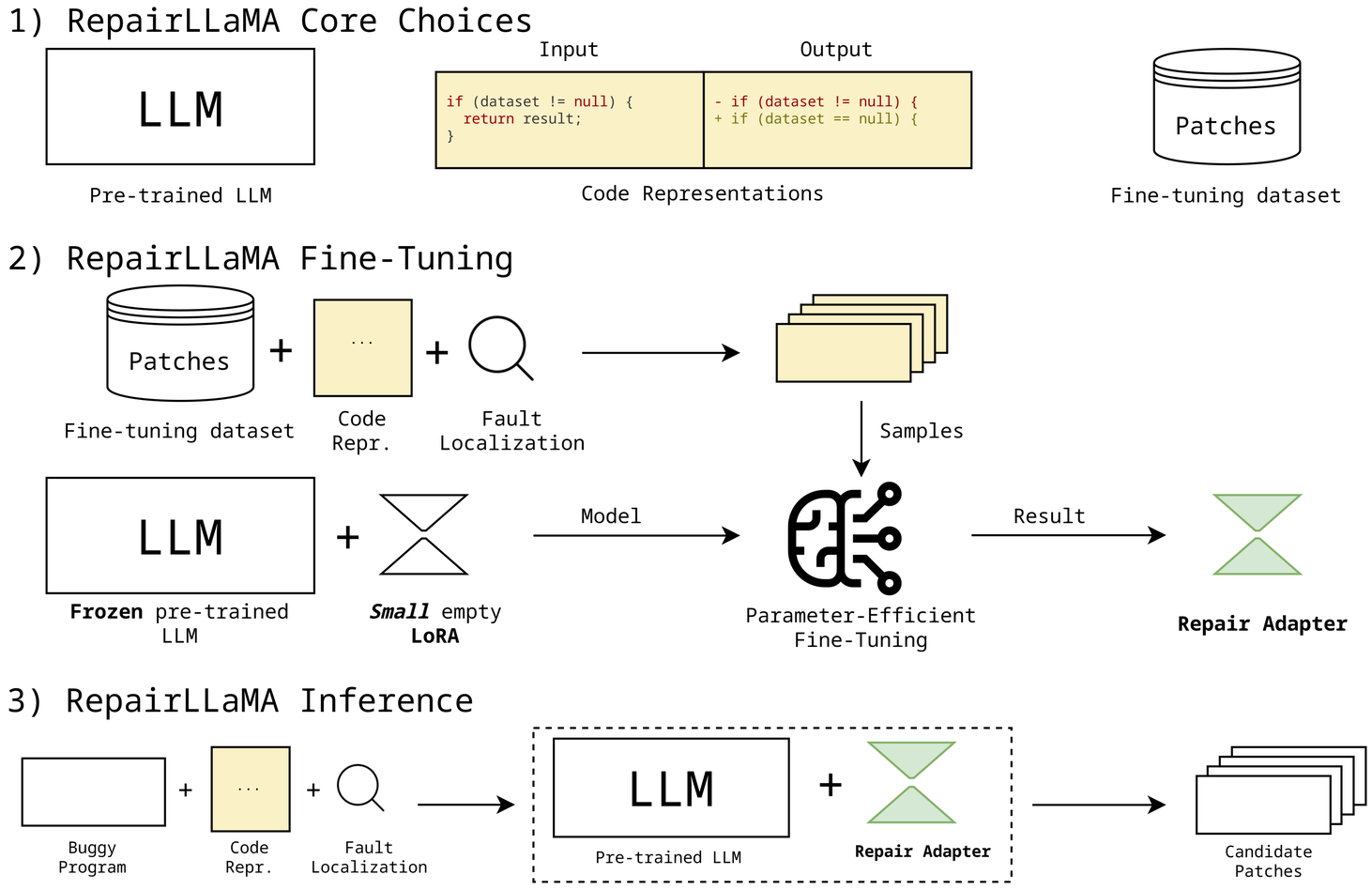
RepairLLaMA:Efficient Representations and Fine-Tuned Adapters for Program Repair

André Silva[1*], Sen Fang[1*], Martin Monperrus[1]
[1] KTH Royal Institute of Technology, Sweden

Scan to read more!

Motivation

- Fine-tuning LLMs for program repair [1] is a recent avenue of research.
- Existing works mostly fine-tune LLMs with naive code representations.
- Existing works do not scale to frontier LLMs due memory requirements.



Results

Model	Defects4J [5] (488 bugs)		
	AST Match	Semantic Match	Plausible
GPT-3.5	33	45	71
GPT-4	60	72	119
RepairLLaMA	**125**	**144**	**195**

RepairLLaMA's effectiveness compared with state-of-the-art ChatGPT-based APR techniques. RepairLLaMA is more effective in finding correct and plausible patches in Defects4J.

- The infilling scheme, supported by code language models, supplemented with the original buggy code, is the best input/output representation pair.
- PEFT outperforms full parameter fine-tuning, avoiding overfitting while optimizing ~1600x less parameters.
- RepairLLaMA outperforms state-of-the-art general purpose models, such as GPT-4, while being much smaller (7B parameters).

Objective

```
@@ -111,9 +111,6 @@
  final double[] lI = lTData[i];
- if (lTData[i][i] < absolutePositivityThreshold) {
-  throw new NotPositiveDefiniteMatrixException();
- }
  for (int j = i + 1; j < order; ++j) {
   final double[] lJ = lTData[j];
@@ -134,6 +131,9 @@
  final double[] ltI = lTData[i];
+ if (ltI[i] < absolutePositivityThreshold) {
+  throw new NotPositiveDefiniteMatrixException();
+ }
  ltI[i] = Math.sqrt(ltI[i]);
  final double inverse = 1.0 / ltI[i];
```

Example of a bug exclusively repaired by RepairLLaMA, Math-86 from Defects4J.

The main objective of RepairLLaMA is to fine-tune LLMs to produce high-quality bug/fix patches.

Methods

- We study two axis in fine-tuning LLMs:
  1) Code Representations
  2) Parameter-Efficient Fine-Tuning (PEFT)
- Fine-tune codellama-7b [2] with six different input-output code representation pairs, tailored for program repair.
- Use LoRA [3], instead of full parameter fine-tuning, reducing the memory requirements.
- Train on ~50k fine-tuning samples from Megadiff [4].

Conclusion

- RepairLLaMA combines PEFT with task-specific code representations.
- RepairLLaMA outperforms larger, state-of-the-art models through specialization in program repair.
- RepairLLaMA correctly repairs 144 real-world bugs from the Defects4J benchmark.

References

[1] Jiang, N., Liu, K., Lutellier, T., & Tan, L. (2023, May). Impact of code language models on automated program repair. In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE) (pp. 1430-1442). IEEE.
[2] Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., ... & Synnaeve, G. (2023). Code llama: Open foundation models for code. arXiv preprint arXiv:2308.12950.
[3] Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In International Conference on Learning Representations.
[4] Monperrus, M., Martinez, M., Ye, H., Madeiral, F., Durieux, T., & Yu, Z. (2021). Megadiff: A dataset of 600k java source code changes categorized by diff size. arXiv preprint arXiv:2108.04631.
[5] R. Just, D. Jalali, and M. D. Ernst, "Defects4j: A database of existing faults to enable controlled testing studies for java programs," in Proceedings of the 2014 International Symposium on Software Testing and Analysis, 2014, pp. 437–440.

KTH VETENSKAP OCH KONST

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM