# Patch Generation with Language Models : Feasibility and Scaling Behavior

Carnegie Mellon University

**How does scale impact a model's ability to generate patches?** We tested whether models ranging from 160M to 12B parameters could patch 40 programs implemented in both Java and Python, (each with a one-line bug).

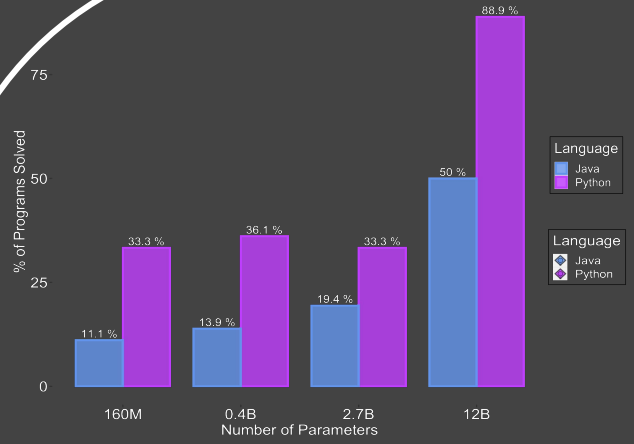
**Method:**

We prompt each model with code up to the buggy line, and then allow it to generate 100 candidate patches. We then check if the new program passes its test cases, and evaluate each model's sampling efficiency as well as the "naturalness" of its patches.

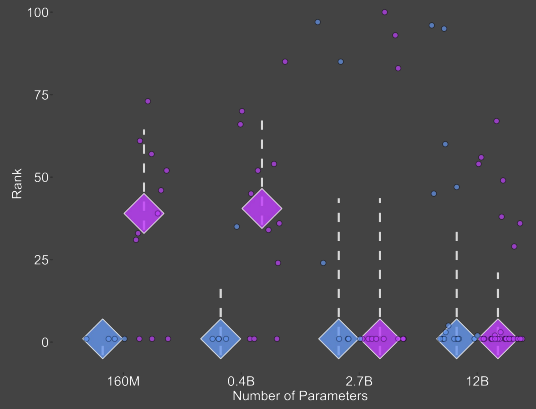
```
def sqrt(x, epsilon):
    approx = x / 2
    while abs(x - approx) > epsilon:
        approx = 0.5 * (approx + x / approx)
    return approx
```

Buggy line

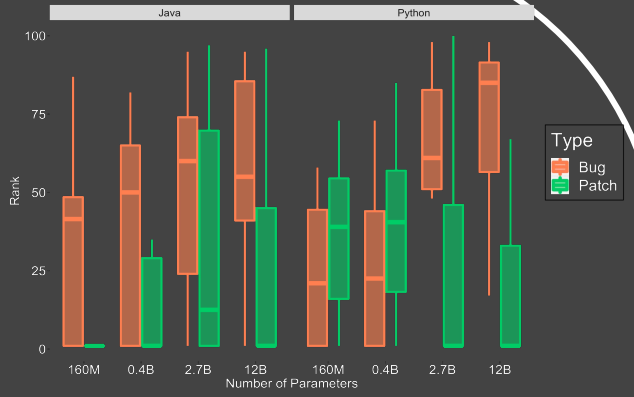
| Model size: | Pass | Candidate patch: |
|---|---|---|
| 160M | ✗ | `if approx > epsilon:` |
| 0.4B | ✗ | `if approx < epsilon:` |
| 2.7B | ✗ | `if abs(approx) < epsilon:` |
| 12B | ✔ | `while abs(approx ** 2 - x) > epsilon:` |




**Larger models patch more programs.**
A significant leap in performance was observed between 2.7B parameters (GPT-2) and 12B parameters (GPT-3). The largest and most successful model patched 89% of Python programs, and all models patched 2-3 times more programs in Python than Java.




**Larger models take fewer tries to generate a patch.** A significant decline in the median entropy of test-passing patches emerged between 0.4B (GPT-2) and 2.7B (GPT-2) in Python, demonstrating that sampling efficiency increases with model size.




**Larger models rank developer patches better than developer bugs.**
In Python, the larger two models (12B and 2.7B) assign true patches lower entropy than bugs, whereas the smaller two (0.4B and 160M) assign true bugs lower entropy (on average).

Sophia Kolak, Ruben Martins, Claire Le Goues, Vincent Hellendoorn