The background is a dark blue gradient. On the left, there is a large, semi-transparent circular inset showing a detailed view of a printed circuit board (PCB) with various electronic components. Overlaid on the top left of this circular inset are two overlapping triangles: a blue one in the foreground and a light green one behind it. In the top right corner, there is a faint, stylized pattern of interconnected lines and squares, resembling a circuit board or a data network.

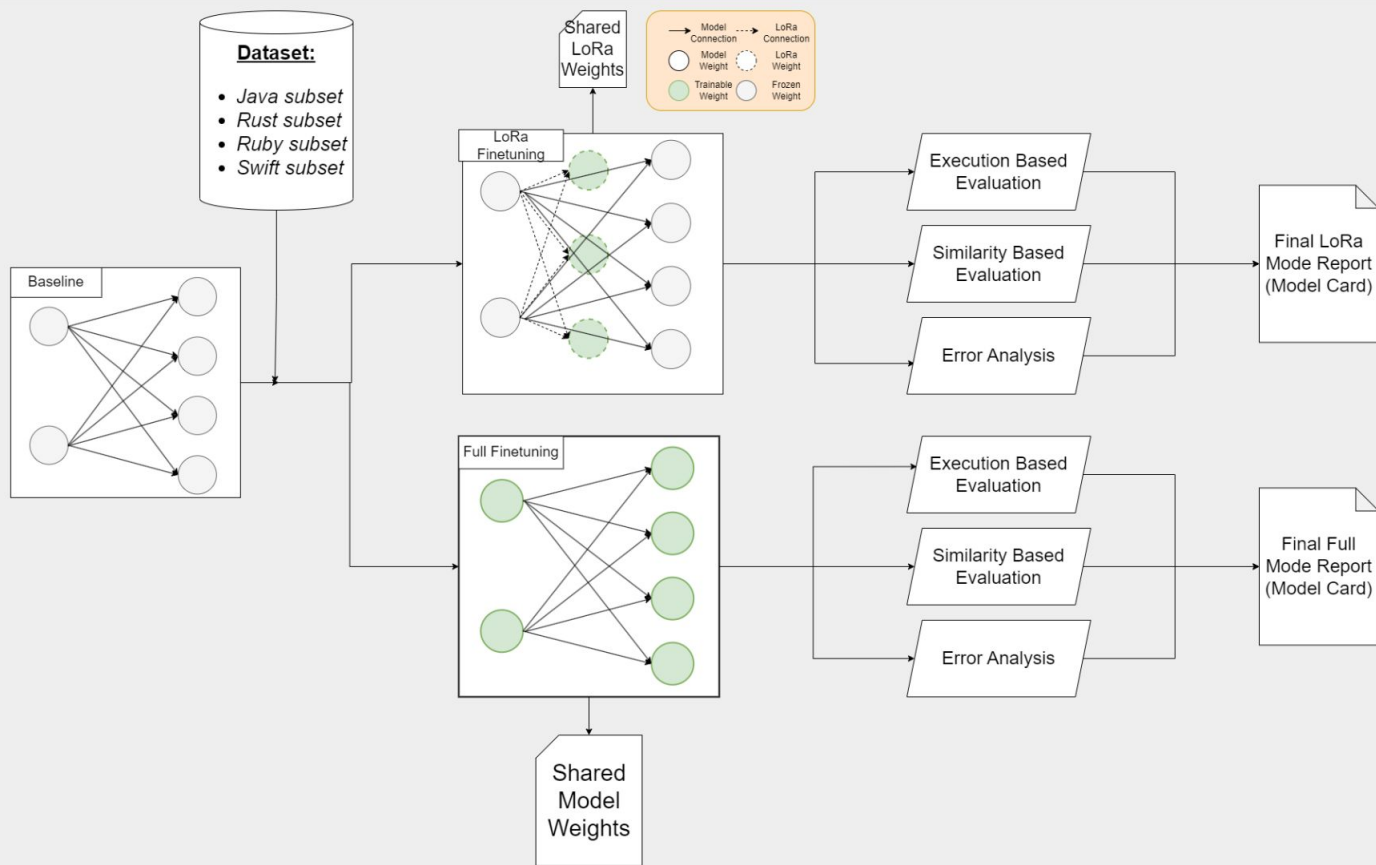
Scaling Down Multilingual Code Language Models



Objectives

- Sharing models weights is not enough to enable equitable access to Code LLMs because of the **Resource Limitations** and the **Knowledge Gap**.
- This project aims to:
 - a. Provide a set of small code language models that can perform code completion in multiple programming languages (**resource limitations**)
 - b. Provide a framework that practitioners can follow to fine-tune code LLMs to their own needs and according to their own constraints (**knowledge gap**)

Methodology



Baseline & Dataset

Dataset: 1 Million files of target programming languages, cleaned and filtered.

Baseline: CodeGen-350M-Mono Autoregressive Code LLM pretrained on Python.

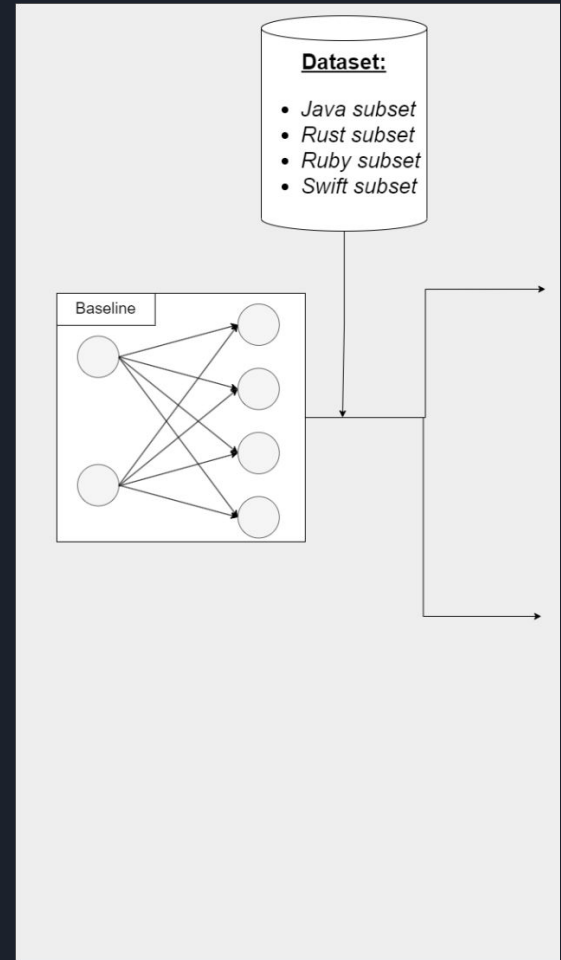
Task: Casual Language Modelling

$$L_{CLM}^{(x)} = -\frac{1}{|x|} \sum_{i=1}^{|x|} \log P(x_i / x_{<i})$$

where:

$x = \{x_1, x_2, x_3, \dots, x_{|x|}\}$ represents a sequence

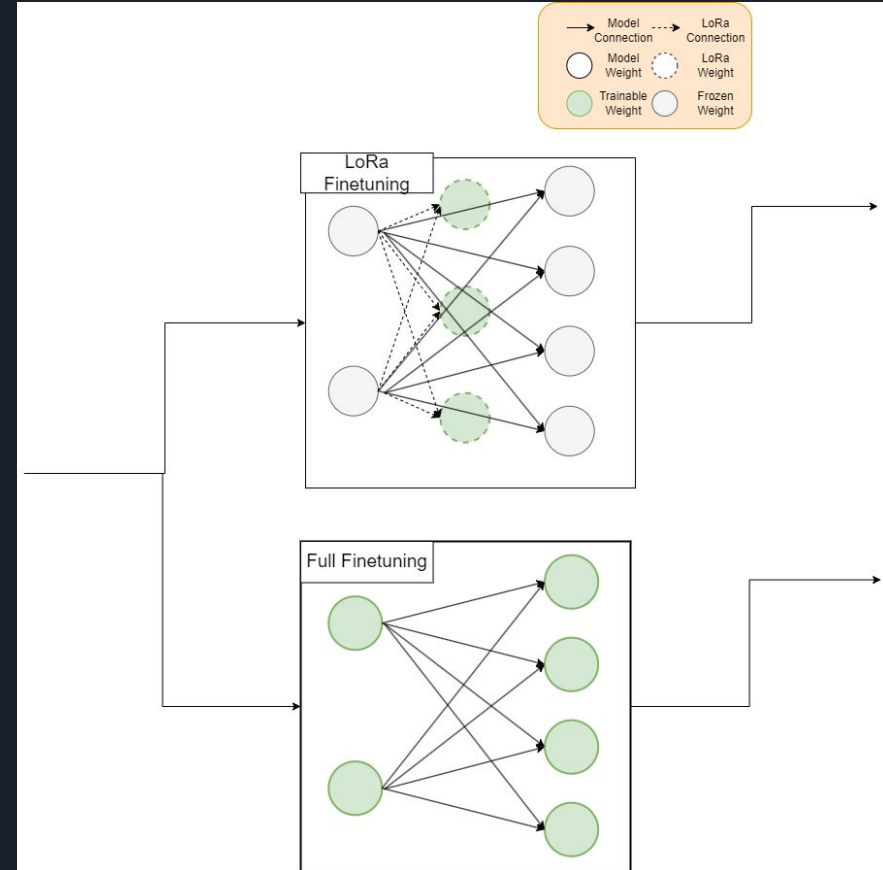
$x_{<i} = x_1, x_2, x_3, \dots, x_{i-1}$



Finetune Methods

Full: All the parameters of the model are trainable.

Low Rank Adaptation (LoRa): Small set of Parameters (inserted at specific layers) are trainable



Results

```
from typing import List

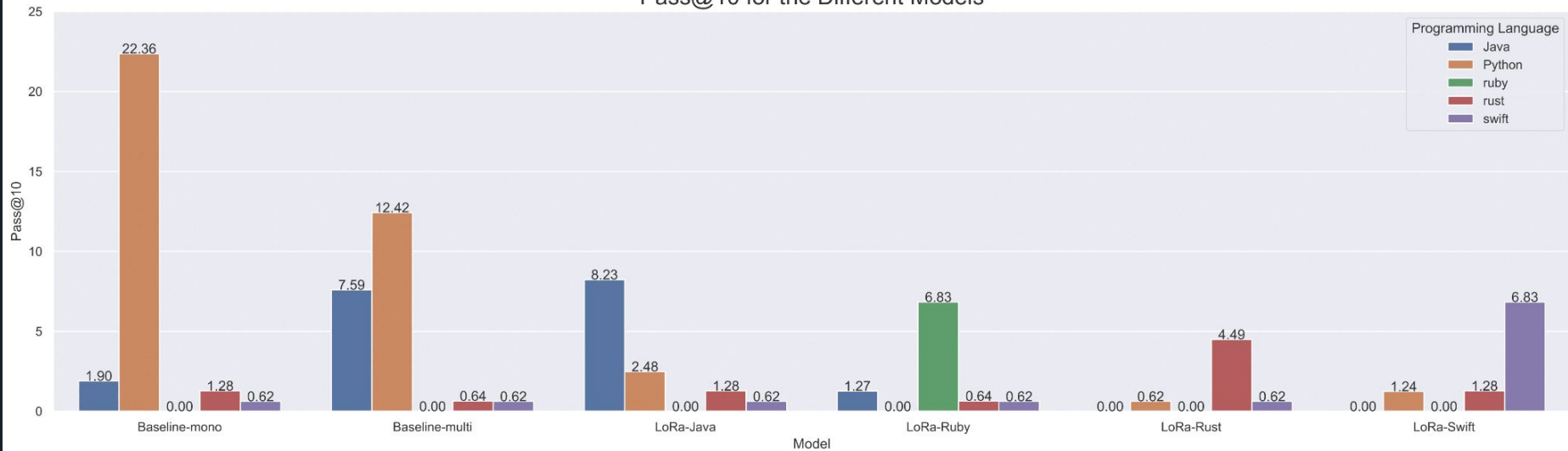
def median(l: List[int]) -> float:
    """Return median of elements in the list l.
    >>> median([3, 1, 2, 4, 5])
    3
    >>> median([-10, 4, 6, 1000, 10, 20])
    15.0
    """

def check(candidate):
    assert candidate([3, 1, 2, 4, 5]) == 3
    assert candidate([-10, 4, 6, 1000, 10, 20]) == 8.0
    assert candidate([5]) == 5
    assert candidate([6, 5]) == 5.5
    assert candidate([8, 1, 3, 9, 9, 2, 7]) == 7

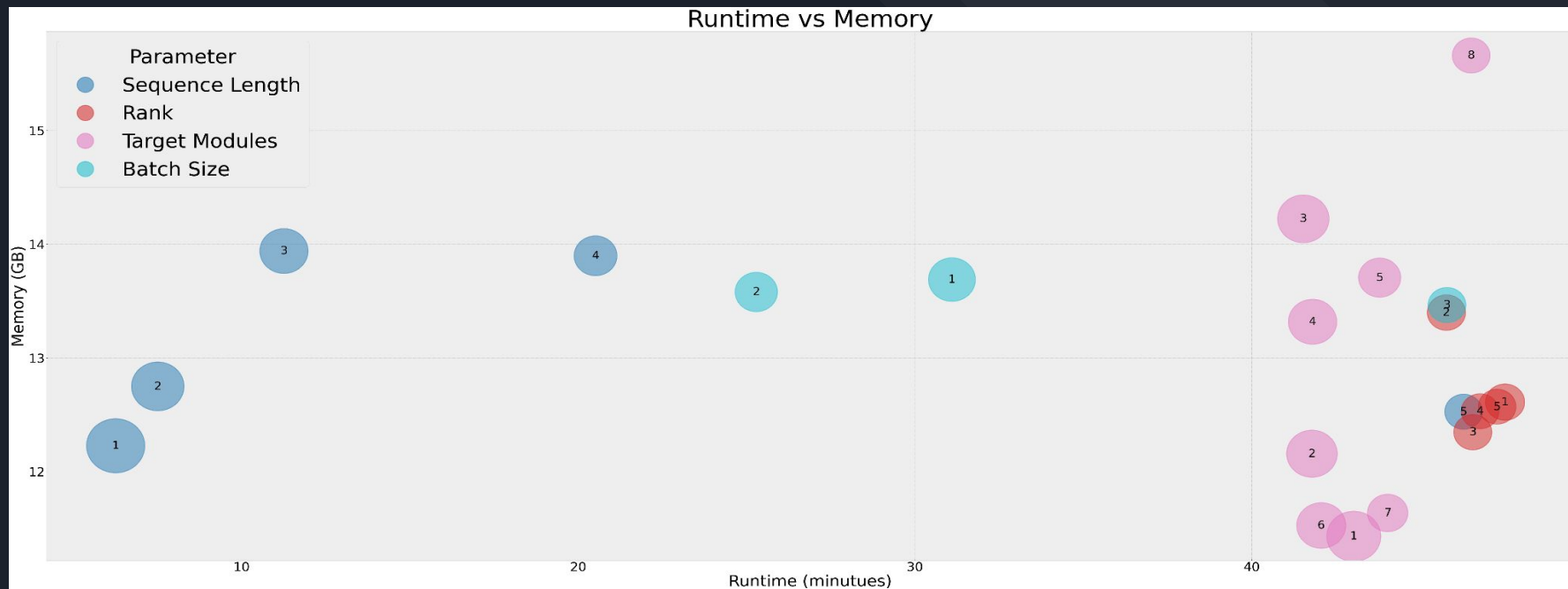
def test_check():
    check(median)

test_check()
```

Pass@10 for the Different Models



Trade-Off Analysis



Variable	Initial Value	Range
LoRa Rank	64	[8, 16, 32, 64, 128]
Batch Size	2	[1, 2, 4]
Sequence Length	1024	[128, 256, 512, 1024, 2048]
Learning Rate	5e-5	[5e-6, 1e-6, 5e-5, 5e-4]
Target Modules	Conf-6	[Attention Modules, Language Feature Modules, Fully Connected Module]

Open-Source (Inference Demo)

The screenshot shows the Hugging Face website's 'Models' page. The left sidebar contains navigation links: 'Tasks' (with a dropdown arrow), 'Libraries', 'Datasets', 'Languages', 'Licenses', and 'Other'. Below these are search filters: 'Filter Tasks by name' and 'Reset Tasks'. The 'Multimodal' section lists 'Feature Extraction', 'Text-to-Image', 'Image-to-Text', and 'Text-to-Video'. The 'Computer Vision' section lists 'Depth Estimation', 'Image Classification', 'Object Detection', 'Image Segmentation', 'Image-to-Image', and 'Unconditional Image Generation'. The main content area shows a list of models. The 'Models' tab is selected, and the filter 'java' is applied. The search results are sorted by 'Most Downloads'. The first model is 'microsoft/CodeGPT-small-java-adaptedGPT2', followed by 'microsoft/CodeGPT-small-java'. The third model, 'ammarnasr/codegen-350M-mono-java', is highlighted with a red box. Below it are 'digitous/Javalion-GPTJ', 'digitous/Javalion-R', and 'kdf/javascript-docstring-generation'.

Hugging Face

Search models, datasets, spaces, docs, solutions, pricing

Models 22

java

new Full-text search

Sort: Most Downloads

microsoft/CodeGPT-small-java-adaptedGPT2

Text Generation • Updated Jan 24 • 1.14k • 14

microsoft/CodeGPT-small-java

Text Generation • Updated Jan 24 • 621 • 11

ammarnasr/codegen-350M-mono-java

Text Generation • Updated 7 days ago • 118

digitous/Javalion-GPTJ

Text Generation • Updated Mar 1 • 23 • 1

digitous/Javalion-R

Text Generation • Updated Mar 2 • 16 • 5

kdf/javascript-docstring-generation

Text Generation • Updated Jul 29, 2022 • 10

The screenshot shows the Hugging Face website's 'Models' page, filtered by 'ruby'. The left sidebar is partially visible, showing the same navigation links as the previous screenshot. The main content area shows a list of models. The 'Models' tab is selected, and the filter 'ruby' is applied. The search results are sorted by 'Most Downloads'. The first model is 'ammarnasr/codegen-350M-mono-ruby', followed by 'stillerman/santacoder-ruby' and 'appvoid/ruby-002'.

Models 3

Ruby

new Full-text search

Sort: Most Downloads

ammarnasr/codegen-350M-mono-ruby

Text Generation • Updated 7 days ago • 48

stillerman/santacoder-ruby

Text Generation • Updated Feb 19 • 3

appvoid/ruby-002

Text Generation • Updated Jul 10 • 1



Thank You!

Questions ?

