Taming open code LLMs for SQL generation and bug fixing

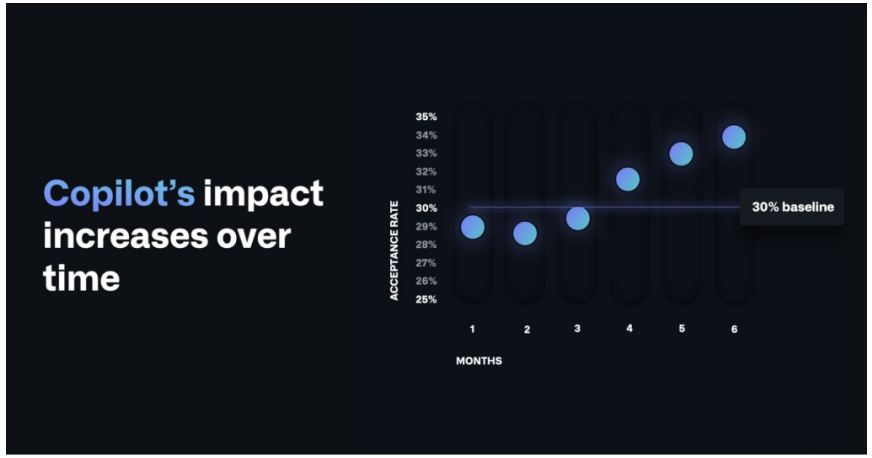
Asankhaya Sharma

CTO

Patched.Codes

Code LLMs

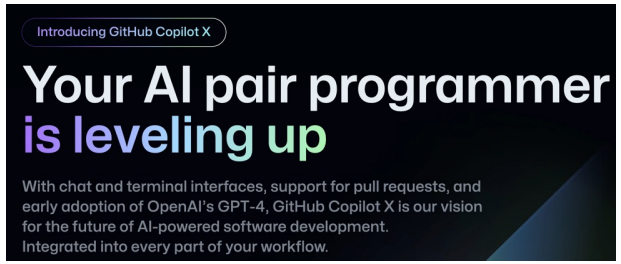
OpenAI



`codex` `code-davinci-002` `GPT-3.5-turbo` `GPT-4`

GitHub Copilot

Copilot for Business new



foss asia

Open-access Code LLMs

StarCoderBase is a 15B parameter decoder trained on 1T tokens of code in 80+ programming languages

Trained on additional 30B tokens of Python

StarCoder

STARCODER:
MAY THE SOURCE BE WITH YOU!

https://arxiv.org/abs/2305.06161

StarCoderBase

Different sizes

starcoderbase-1b
starcoderbase-3b
starcoderbase-7b

StarCoderPlus

Trained on additional 600B tokens of natural text from RefinedWeb and Wikipedia

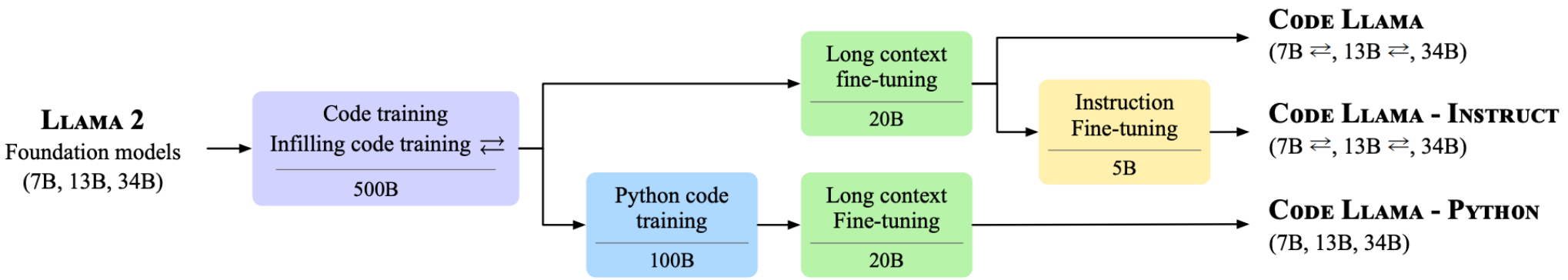
StarChat-Beta

fine-tuned StarCoderPlus with an "uncensored" variant of the openassistant-guanaco dataset

The Stack - a 6.4TB of source code in 358 programming languages from permissive licenses.

Open-access Dataset

Code Llama



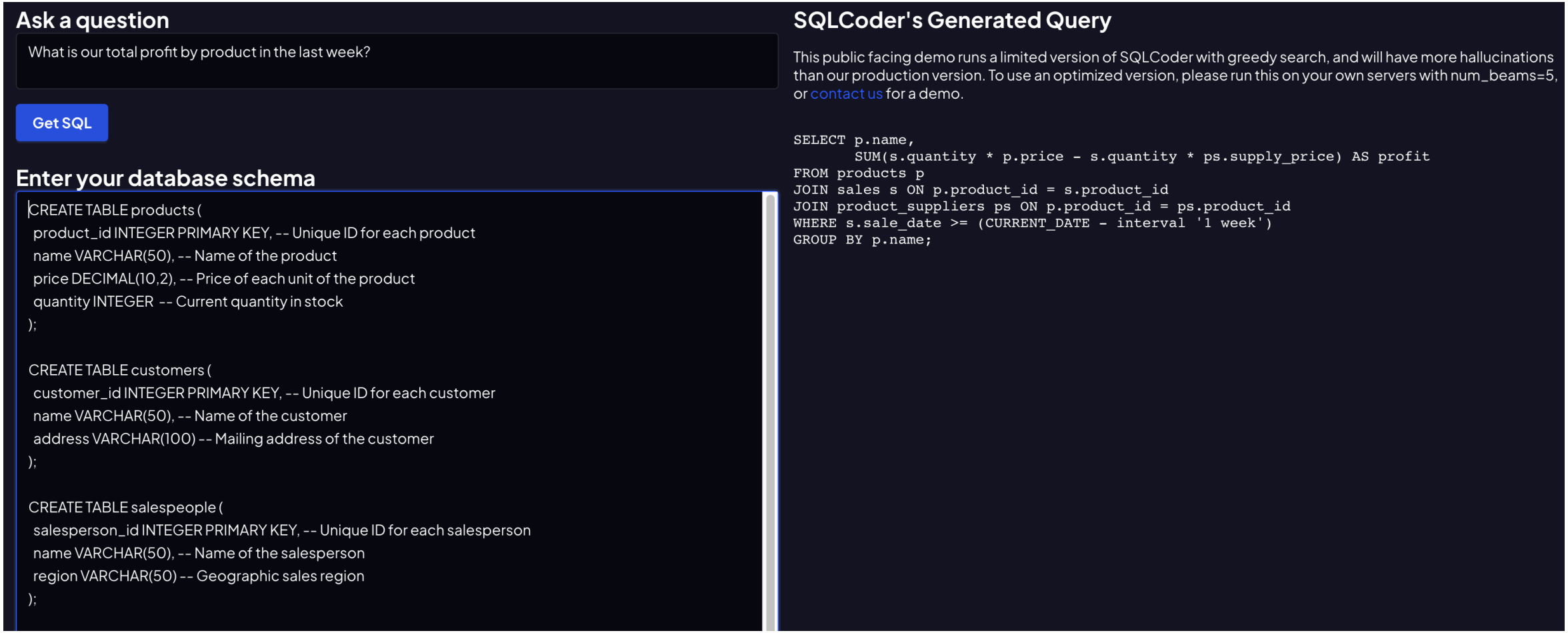
Code Llama: Open Foundation Models for Code

Baptiste Rozière[†], Jonas Gehring[†], Fabian Gloeckle[†,*], Sten Sootla[†], Itai Gat, Xiaoqing Ellen Tan, Yossi Adi[◇], Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, Gabriel Synnaeve[†]

Meta AI

https://arxiv.org/abs/2308.12950

Text-to-SQL Generation



https://defog.ai/sqlcoder-demo/

Fine-Tuning LLMs

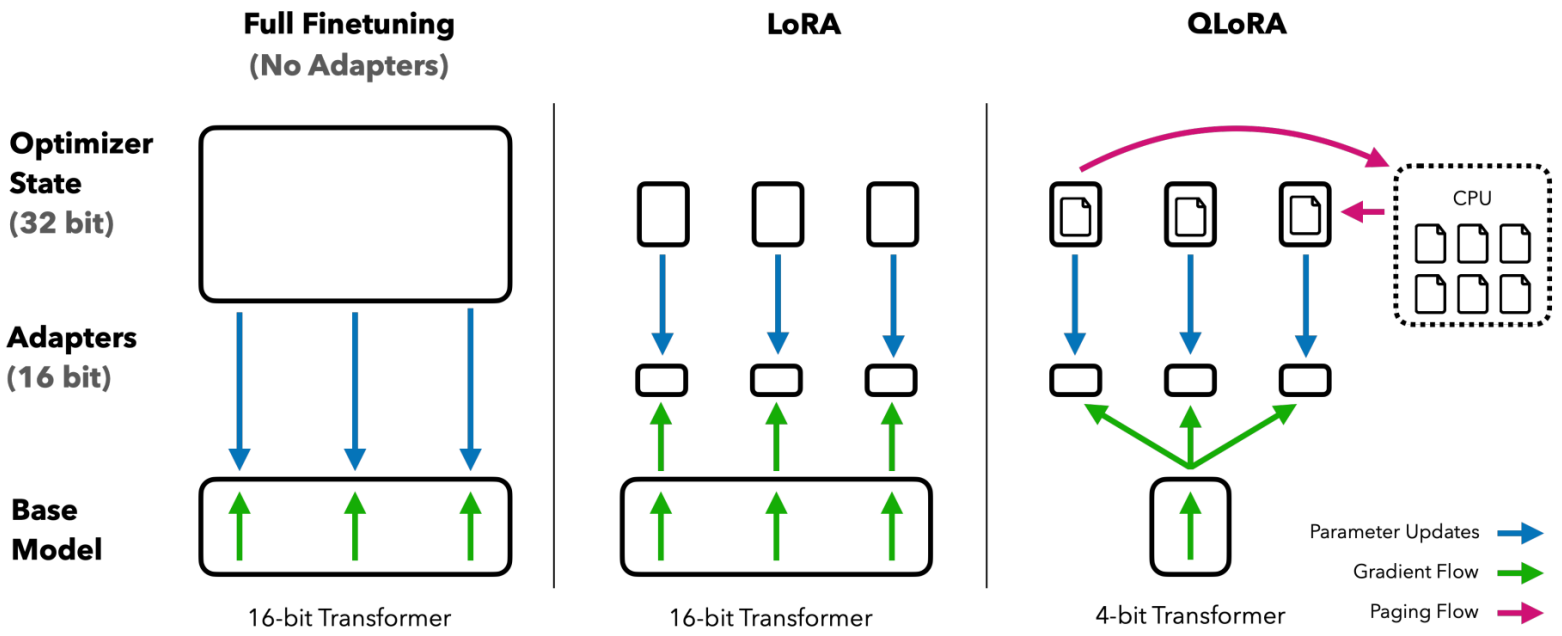


Figure 1: Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

QLoRA: Efficient Finetuning of Quantized LLMs

https://arxiv.org/abs/2305.14314

Google Colab Notebook

http://bit.ly/patched-codes

Bug Fixing

Fix bug in fibonacci

```
def fibonacci(n):
        if n == 0:
                return 0
        elif n == 1 or n == 2:
                return 1
        else:
                return fibonacci(n-1) - fibonacci(n-2)
```

Requires a LLM that can follow instructions (or is chatty)

```
def fibonacci(n):
        if n == 0:
                return 0
        elif n == 1 or n == 2:
                return 1
        else:
                return fibonacci(n-1) + fibonacci(n-2)
```

Are commits a good data source for instruction tuning code LLMs?

A Machine Learning Approach for Vulnerability Curation

Yang Chen
Veracode
ychen@veracode.com

Andrew E. Santosa
Veracode
asantosa@veracode.com

Ang Ming Yi
Veracode
mang@veracode.com

Abhishek Sharma
Veracode
absharma@veracode.com

Asankhaya Sharma
Veracode
asharma@veracode.com

David Lo
Singapore Management University
davidlo@smu.edu.sg

https://dl.acm.org/doi/10.1145/3379597.3387461

Code Before

```
import numpy as np
import matplotlib.pyplot as plt

# generate sample data
x_data = np.linspace(-5, 5, 20)
y_data = np.random.normal(0.0, 1.0, x_data.size)

plt.plot(x_data, y_data, 'o')
plt.show()
```

Commit Message

Change to sin() function with noise

Code After

```
import math
import numpy as np
import matplotlib.pyplot as plt

# generate sample data
x_data = np.linspace(-math.pi, math.pi, 30)
y_data = np.sin(x_data) + np.random.normal(0.0, 0.1, x_data.size)

plt.plot(x_data, y_data, 'o')
plt.show()
```

Google Colab Notebook

http://bit.ly/patched-codes

Thank You!

Questions?

Contact

asankhaya@patched.codes