

# Taming open code LLMs for SQL generation and bug fixing

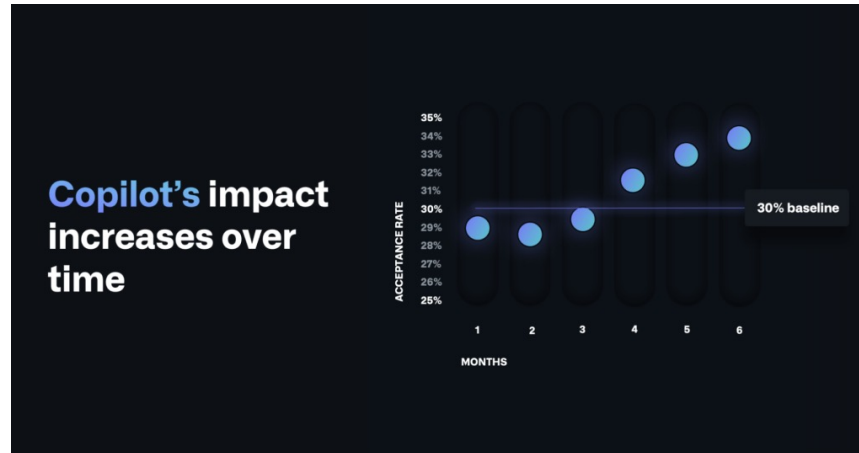
Asankhaya Sharma

CTO

[Patched.Codes](#)

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
selection at the end -add  
ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly  
-- OPERATOR CLASSES ----  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
context):  
context.active_object is not
```

# Code LLMs



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



Copilot for Business **new**

Introducing GitHub Copilot X

**Your AI pair programmer is leveling up**

With chat and terminal interfaces, support for pull requests, and early adoption of OpenAI's GPT-4, GitHub Copilot X is our vision for the future of AI-powered software development. Integrated into every part of your workflow.

**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

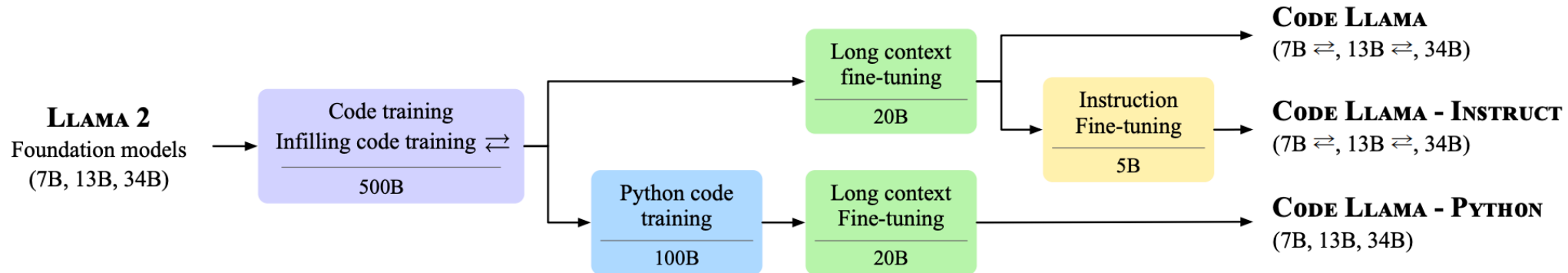
fine-tuned StarCoderPlus with an "uncensored" variant of the openassistant-guanaco dataset



**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<sup>†</sup>, Jonas Gehring<sup>†</sup>, Fabian Gloeckle<sup>†,\*</sup>, Sten Sootla<sup>†</sup>, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi<sup>°</sup>, 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<sup>†</sup>

Meta AI

<https://arxiv.org/abs/2308.12950>

# Text-to-SQL Generation

## Ask a question

What is our total profit by product in the last week?

Get SQL

## Enter your database schema

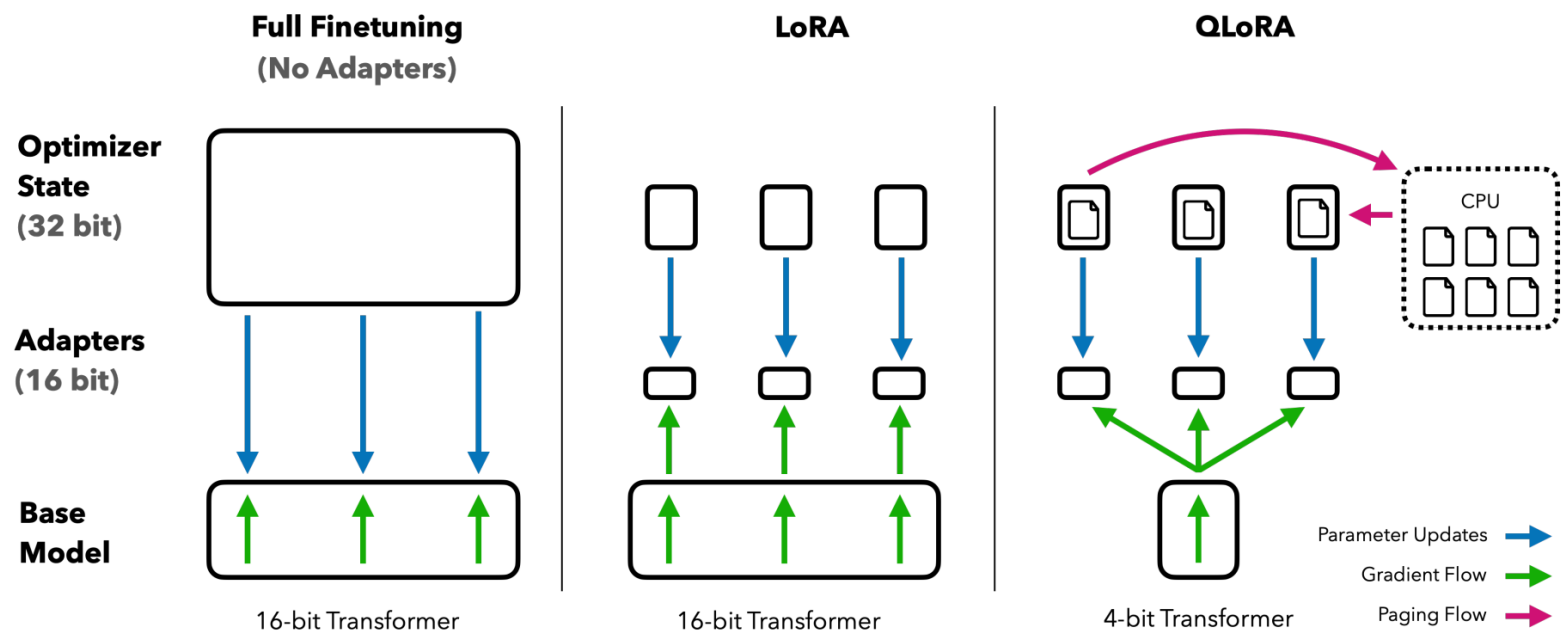
```
CREATE TABLE products (  
  product_id INTEGER PRIMARY KEY, -- Unique ID for each product  
  name VARCHAR(50), -- Name of the product  
  price DECIMAL(10,2), -- Price of each unit of the product  
  quantity INTEGER -- Current quantity in stock  
);  
  
CREATE TABLE customers (  
  customer_id INTEGER PRIMARY KEY, -- Unique ID for each customer  
  name VARCHAR(50), -- Name of the customer  
  address VARCHAR(100) -- Mailing address of the customer  
);  
  
CREATE TABLE salespeople (  
  salesperson_id INTEGER PRIMARY KEY, -- Unique ID for each salesperson  
  name VARCHAR(50), -- Name of the salesperson  
  region VARCHAR(50) -- Geographic sales region  
);
```

## SQLCoder's Generated Query

This public facing demo runs a limited version of SQLCoder with greedy search, and will have more hallucinations than our production version. To use an optimized version, please run this on your own servers with num\_beams=5, or [contact us](#) for a demo.

```
SELECT p.name,  
       SUM(s.quantity * p.price - s.quantity * ps.supply_price) AS profit  
FROM products p  
JOIN sales s ON p.product_id = s.product_id  
JOIN product_suppliers ps ON p.product_id = ps.product_id  
WHERE s.sale_date >= (CURRENT_DATE - interval '1 week')  
GROUP BY p.name;
```

# Fine-Tuning LLMs



**QLoRA: Efficient  
Finetuning of  
Quantized LLMs**

<https://arxiv.org/abs/2305.14314>

**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.

# 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>

```
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()
```

**Code Before**

Change to sin() function with noise

**Commit  
Message**

```
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()
```

**Code After**

# Google Colab Notebook

<http://bit.ly/patched-codes>





# Thank You!

---



**Questions?**



**Contact**

`asankhaya@patched.codes`