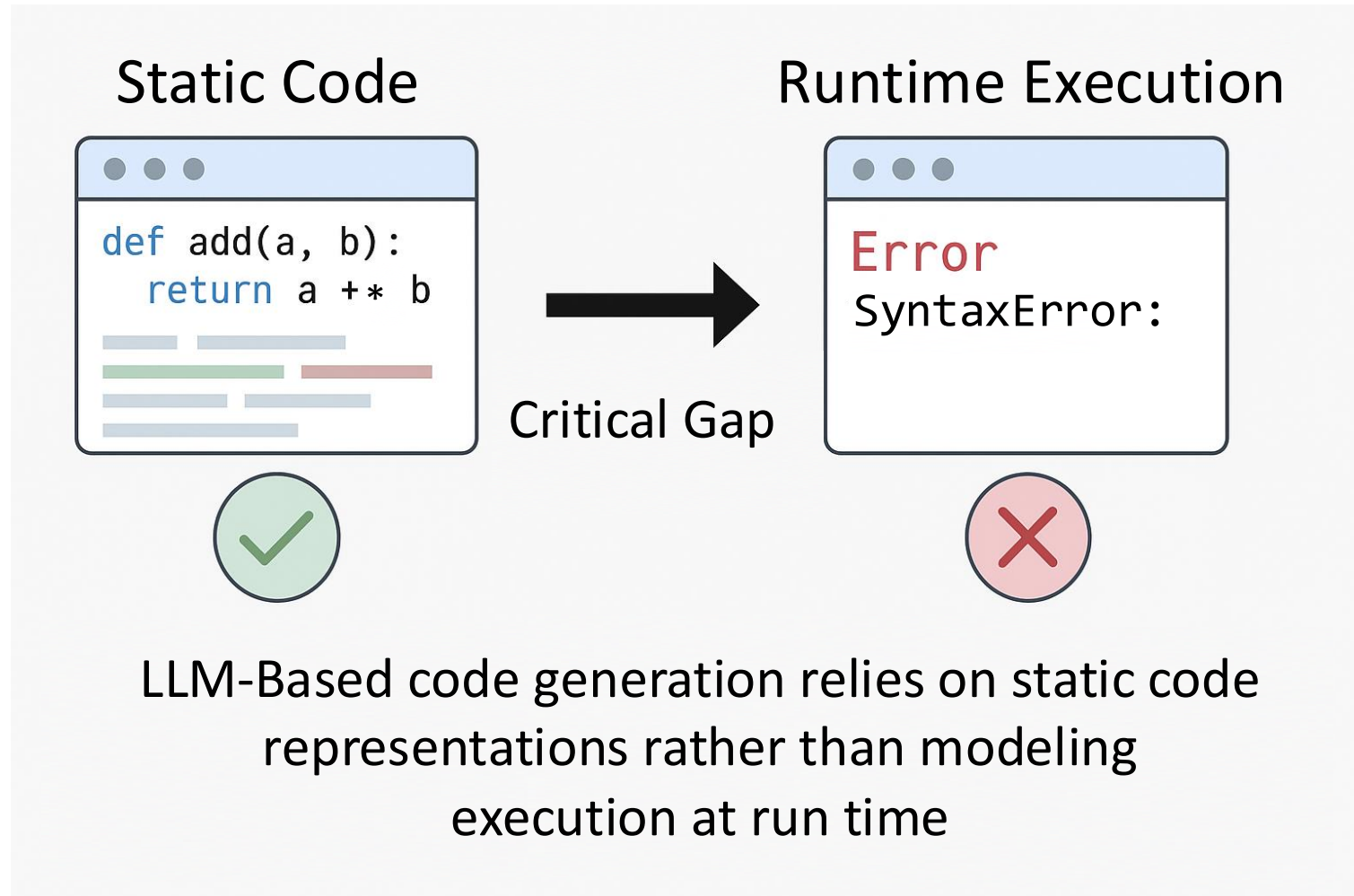# Execution-Guided Line-by-Line Code Generation
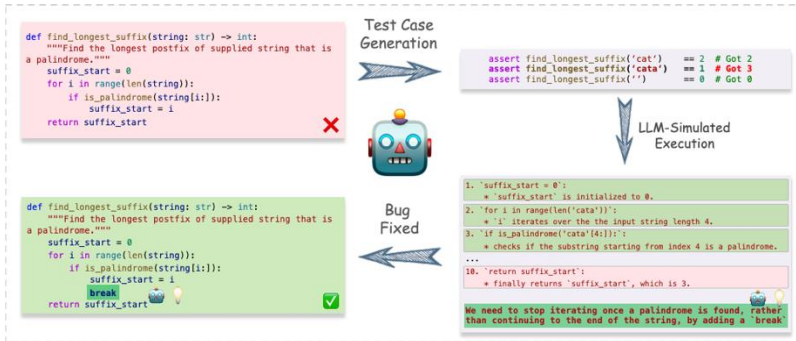## EG-CFG

Boaz Lavon, Shachar Katz, Lior Wolf
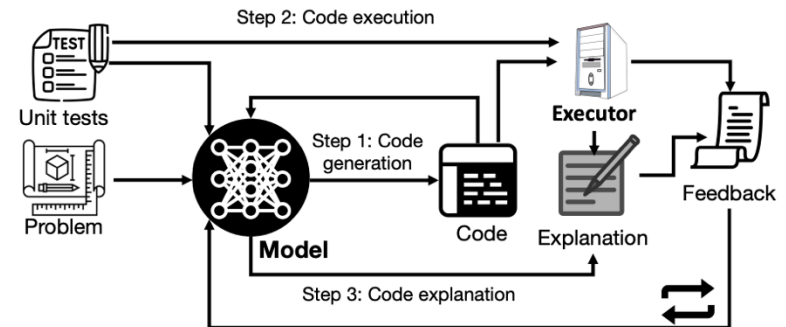Tel Aviv University

# Static Code Generation

Sequential Iterative Refinement

Yuling Shi, Songsong Wang, Chengcheng Wan, and Xiaodong Gu. From code to correctness: *Closing the last mile of code generation with hierarchical debugging*. arXiv preprint arXiv:2410.01215, 2024

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. *Teaching large language models to self-debug*. arXiv preprint arXiv:2304.05128, 2023.
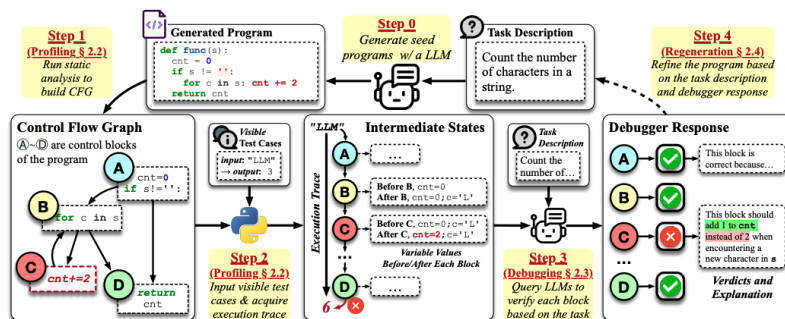

Li Zhong Zilong Wang† Jingbo Shang. Debug like a Human: *A Large Language Model Debugger via Verifying Runtime Execution Step by Step* . arXiv preprint https://arxiv.org/abs/2402.16906
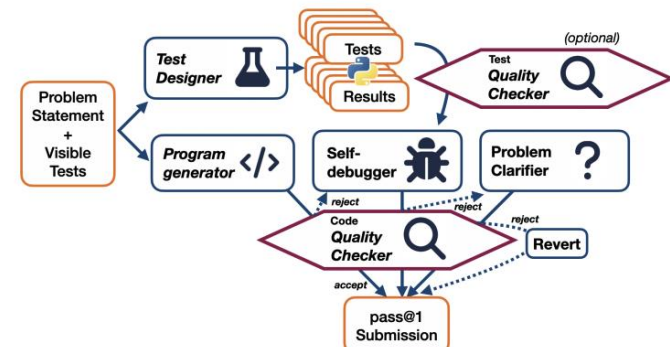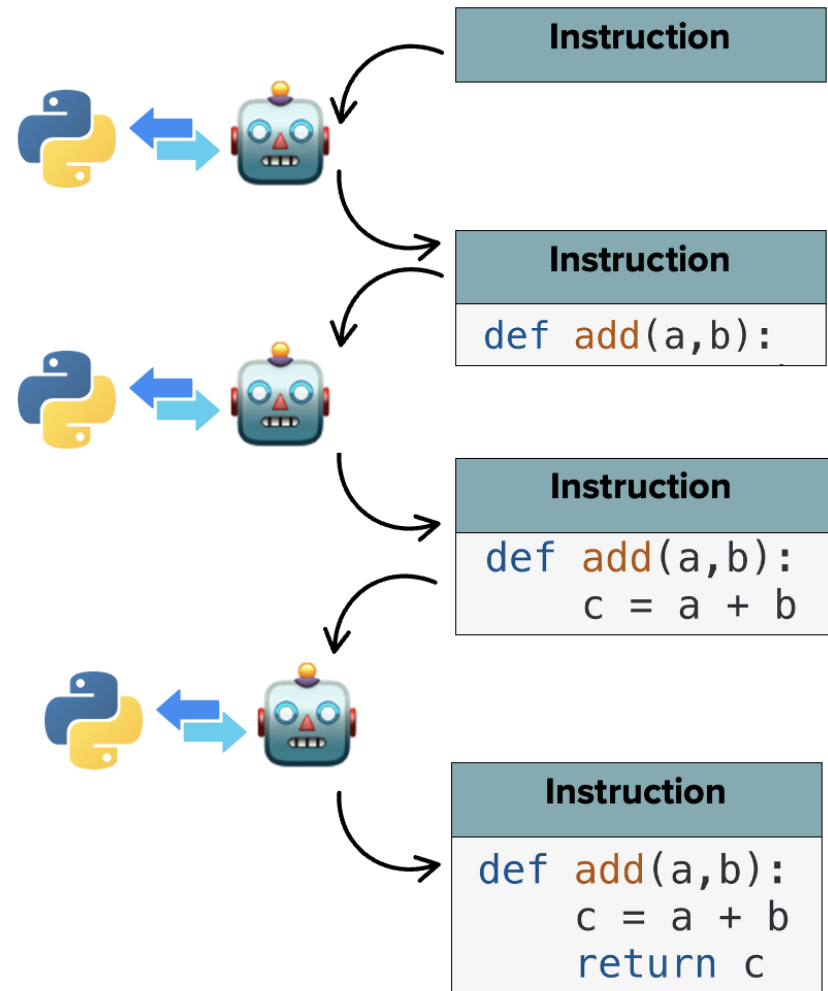

Yaojie Hu, Qiang Zhou, Qihong Chen, Xiaopeng Li, Linbo Liu, Dejiao Zhang, Amit Kachroo, Talha Oz,and Omer Tripp. Qualityflow: *An agentic workflow for program synthesis controlled by llm quality checks*. arXiv preprint arXiv:2501.17167, 2025.
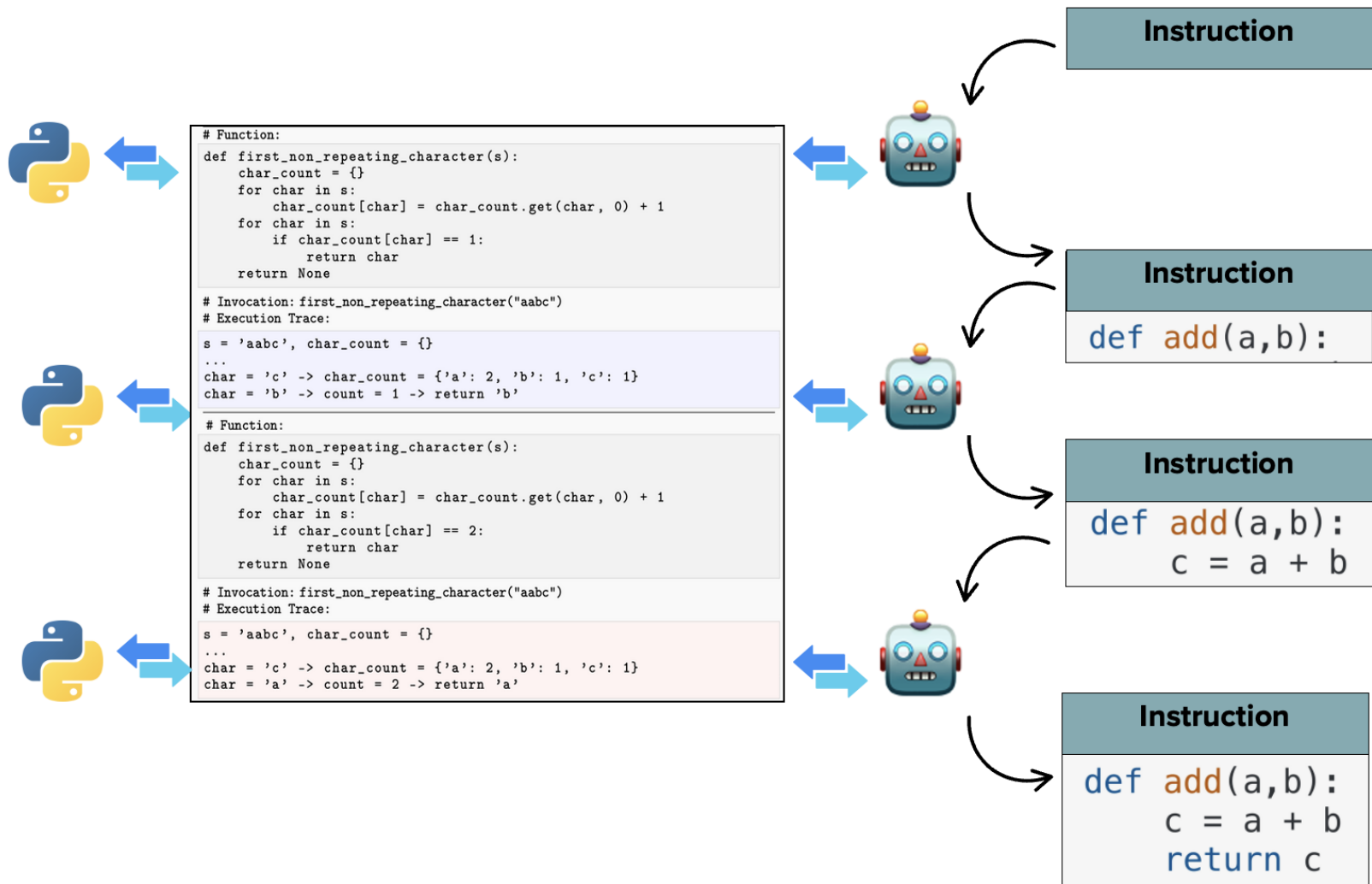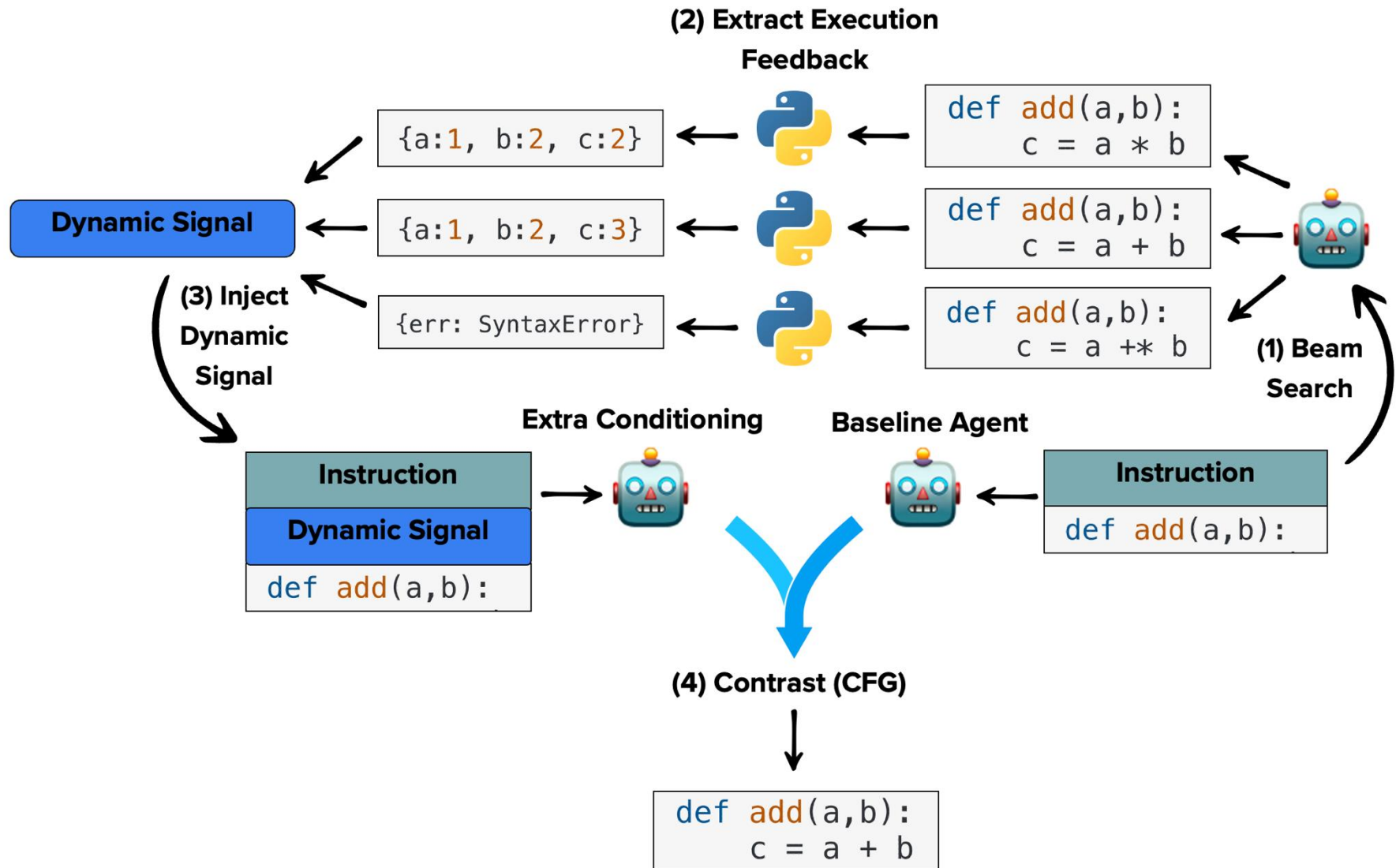
Sequential Iterative Refinement (prev. slide)
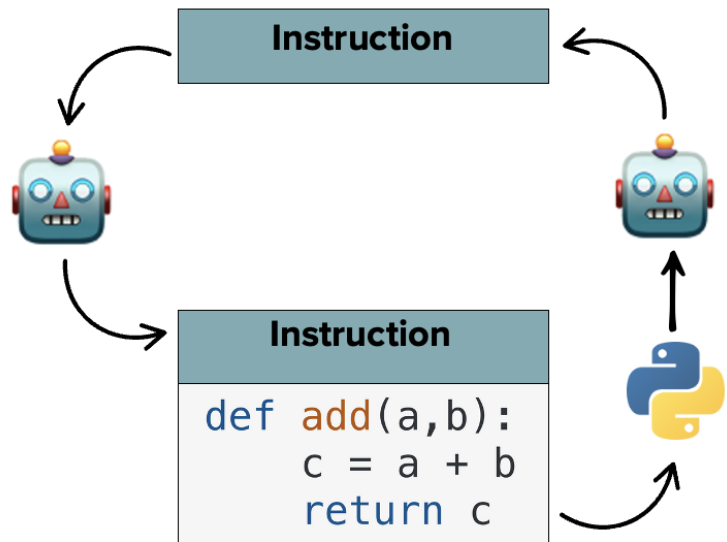
EG-CFG: Incremental Line-by-Line Generation.

Rich Execution Feedback
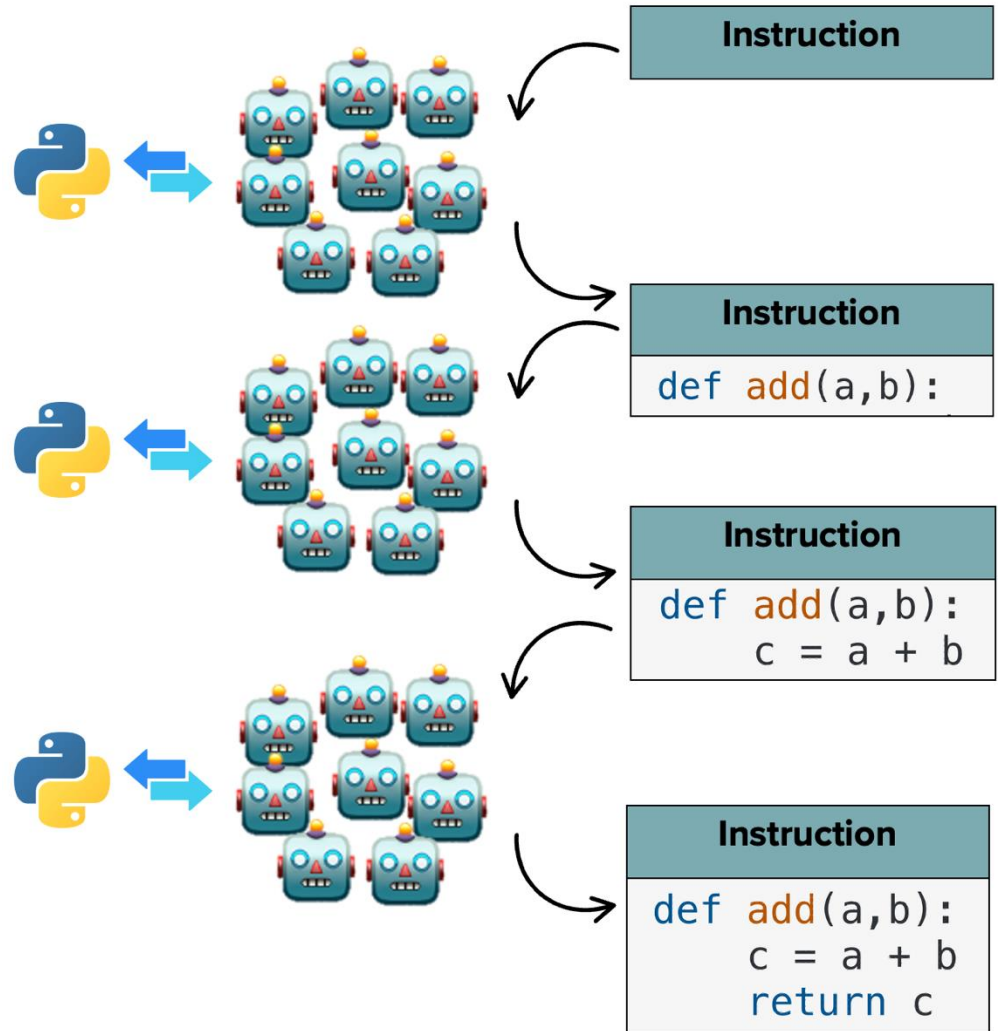
Classifier Free Guidance (CFG)

Native Parallelism

Multiple agents are running in parallel

Each agent uses a different configuration, exploring diverse reasoning paths

Together, creating a diverse set of candidate solutions

Instruction

Instruction
```python
def add(a,b):
```

Instruction
```python
def add(a,b):
    c = a + b
```

Instruction
```python
def add(a,b):
    c = a + b
    return c
```

SOTA Code Generation Results

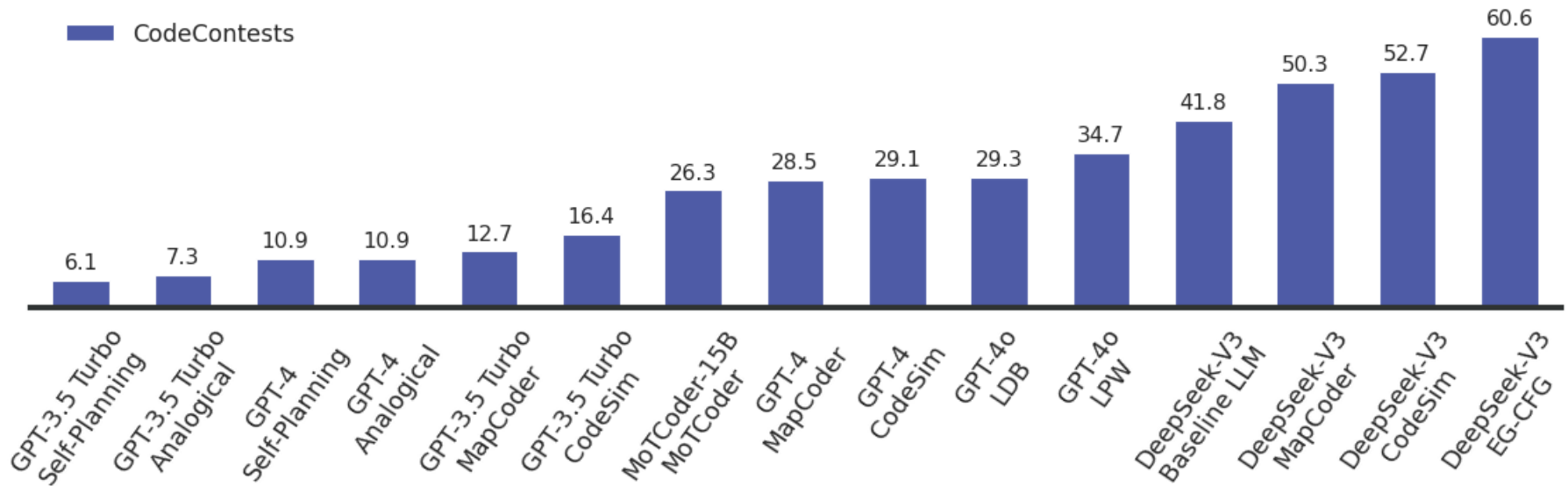MBPP and the much more challenging Extended Testing MBPP-ET
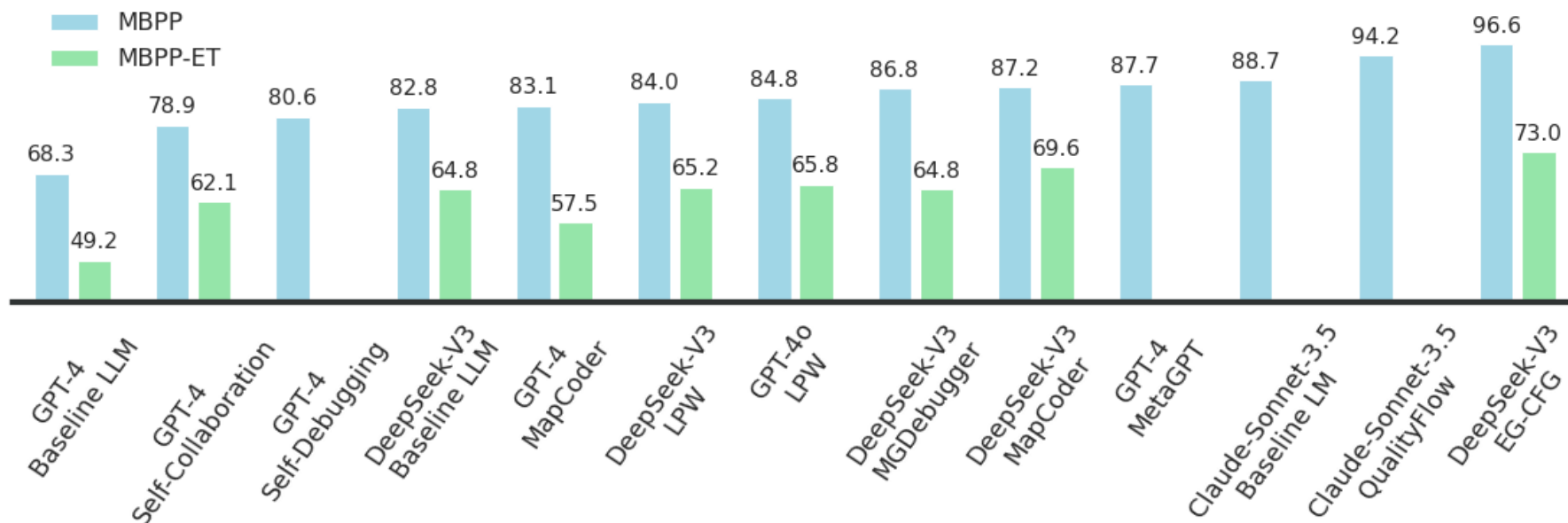
SOTA Code Generation Results

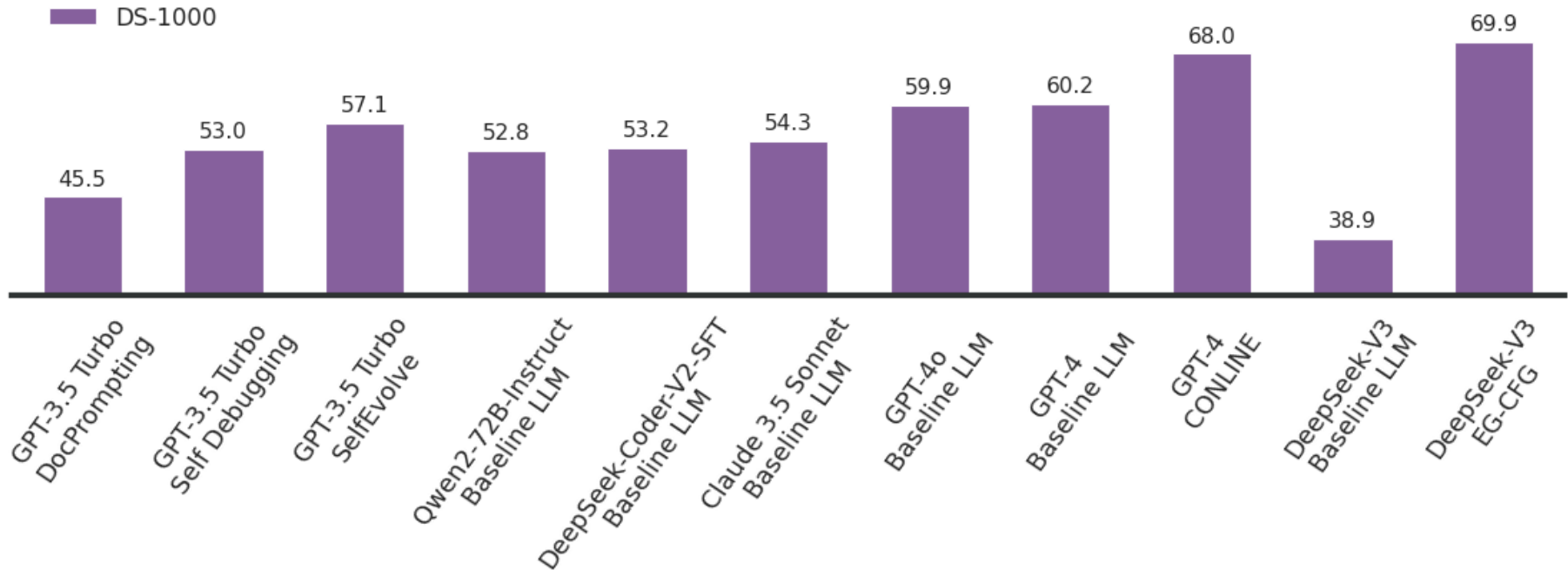HumanEval and the much more challenging Extended Testing HumanEval-ET
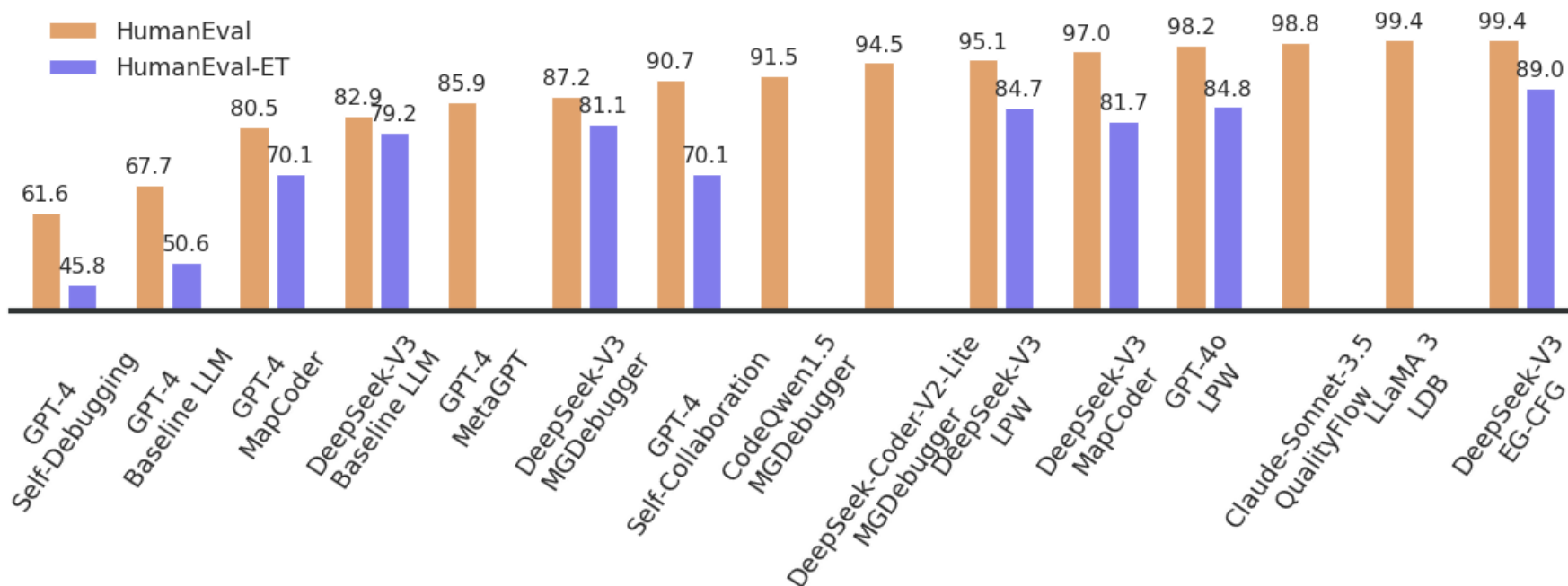
SOTA Code Generation Results

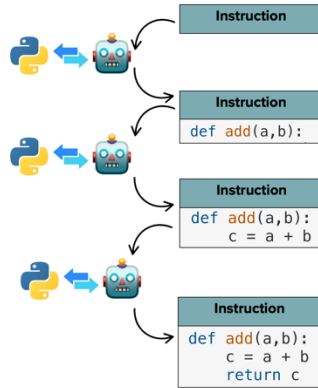CodeContests: Challenging Competitive Programming Problems

SOTA Code Generation Results
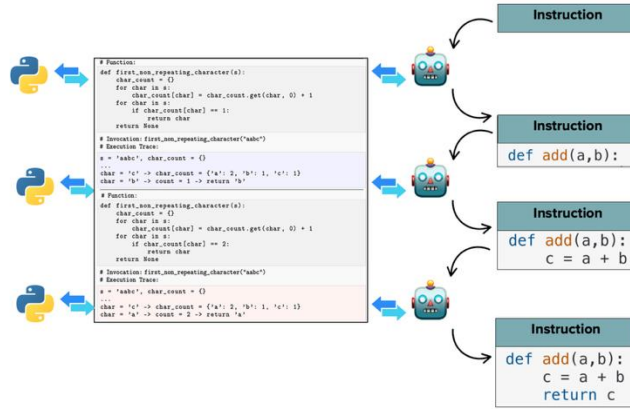
DS-1000: Challenging Data Science Problems

Thank You!

Line-by-Line Generation

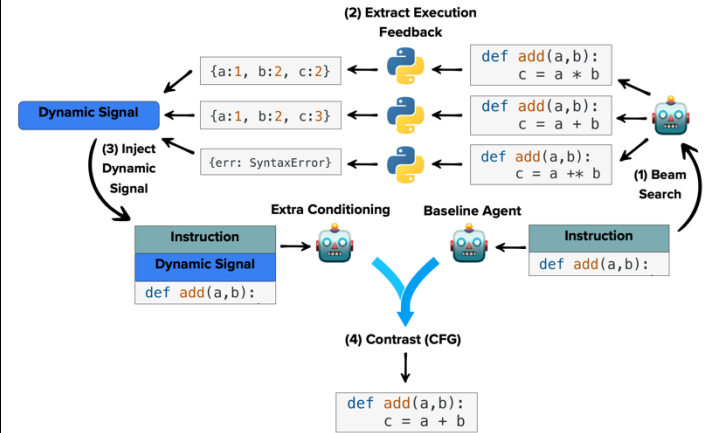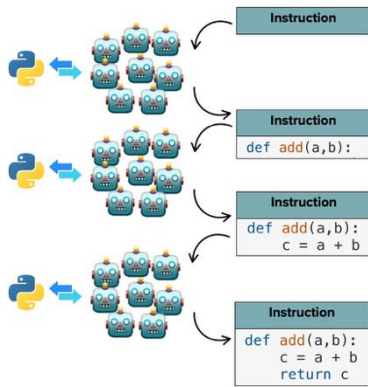## Rich Execution Feedback

Classifier Free Guidance (CFG)

Native Parallelism

SOTA Code Generation Results