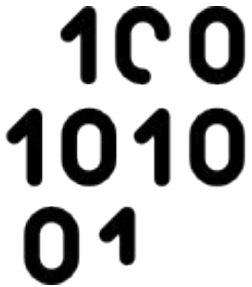
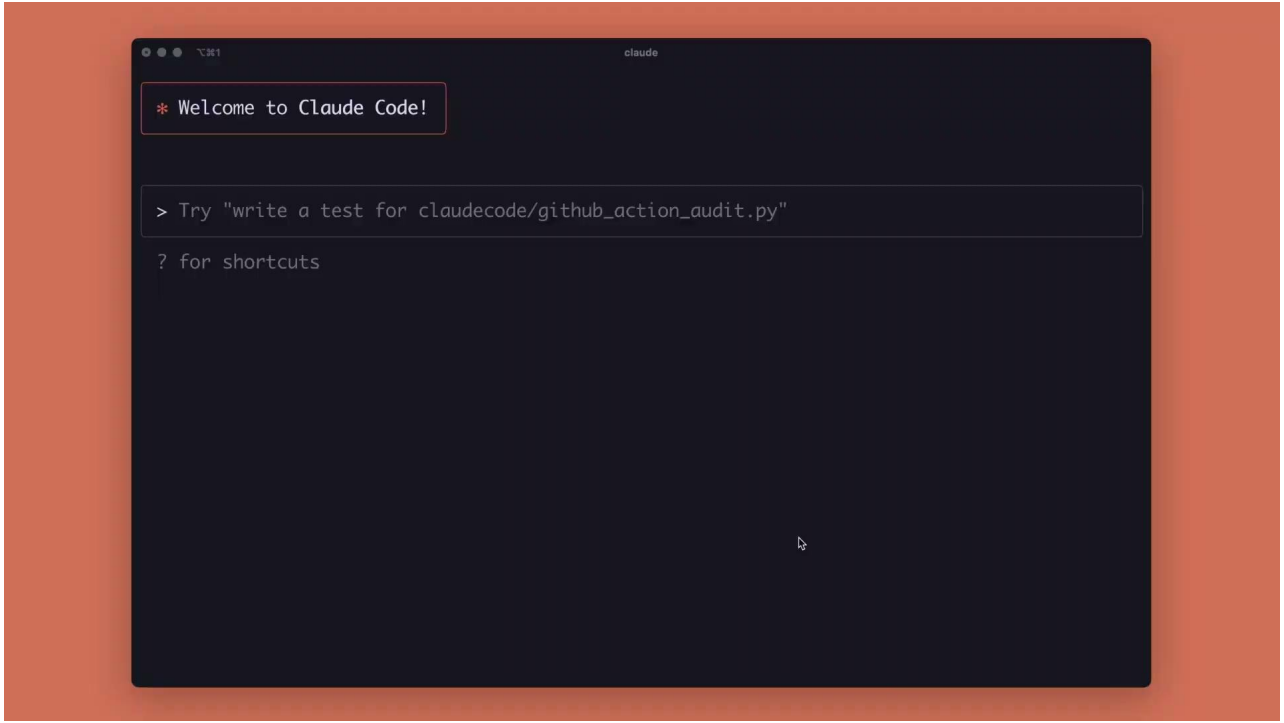
AI Coding Agents

Large Language Model Applications



Zora Wang

Coding Agents Acting On Par with Software Engineers



3 Years Ago - Coding Models Barely Work

Best coding models then

- Fail on using libraries

Should use `client.send` instead!

```
# sending http headers to `client`
client.get('http://localhost:5000/failed_examples')
```

GitHub Copilot

- Unaware of execution errors

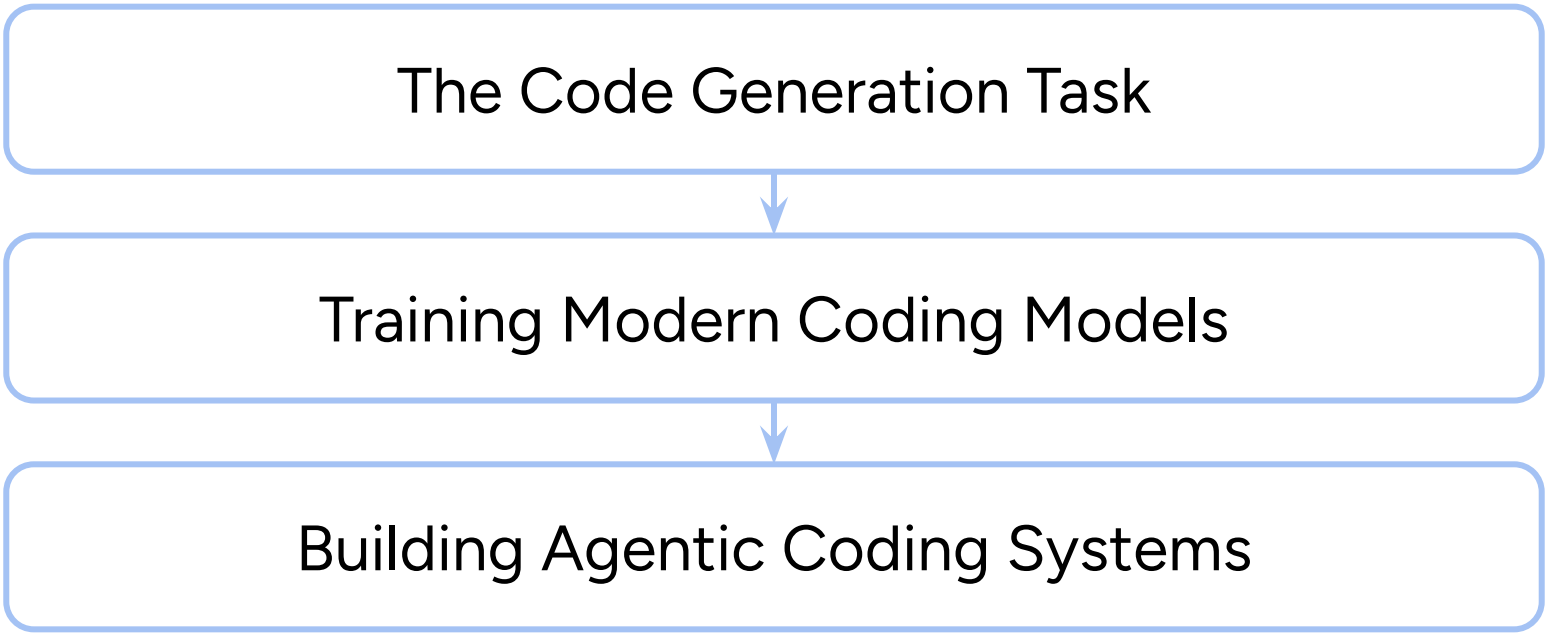
```
2  # convert hex string '470FC614' to a float number
3
4  import struct
5
6  hex_string = '470FC614'
7  float_number = struct.unpack('f', hex_string.decode('hex'))[0]
```

GitHub Copilot

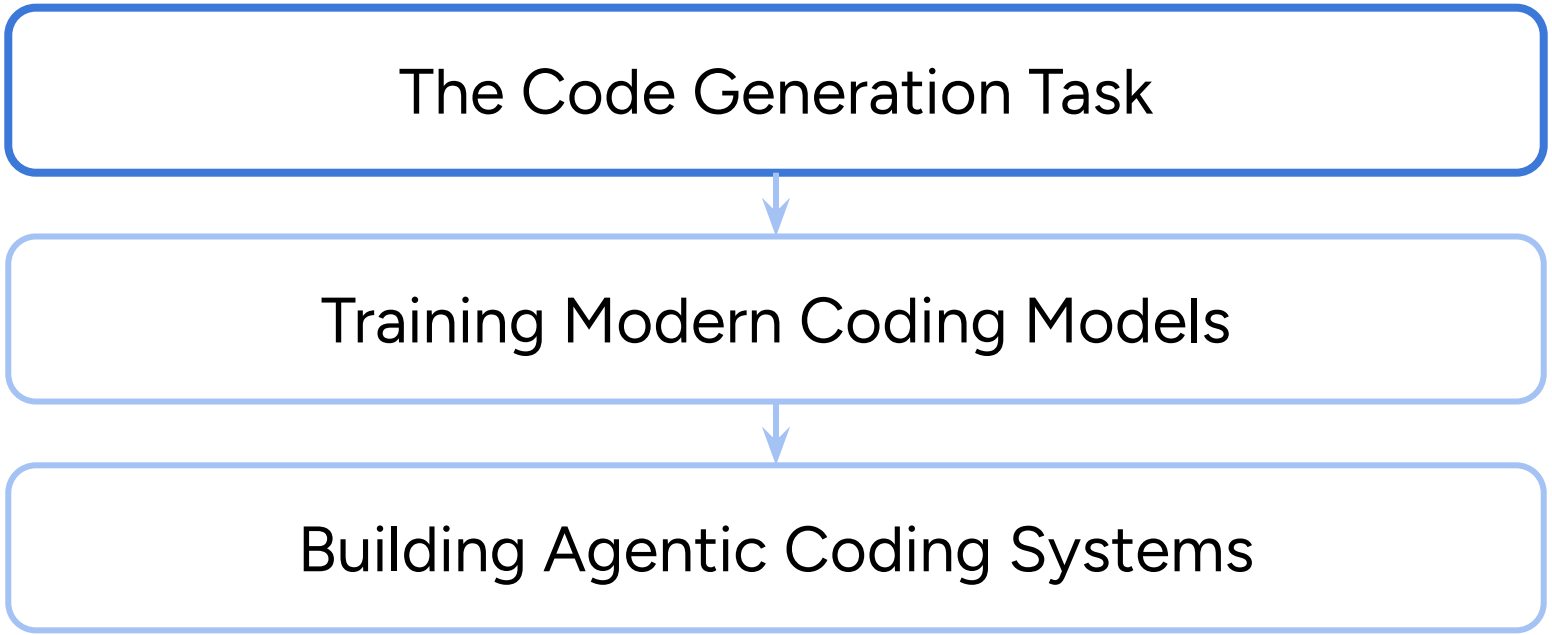
Exception has occurred: AttributeError ×

'str' object has no attribute 'decode'

How Did We Get Here?

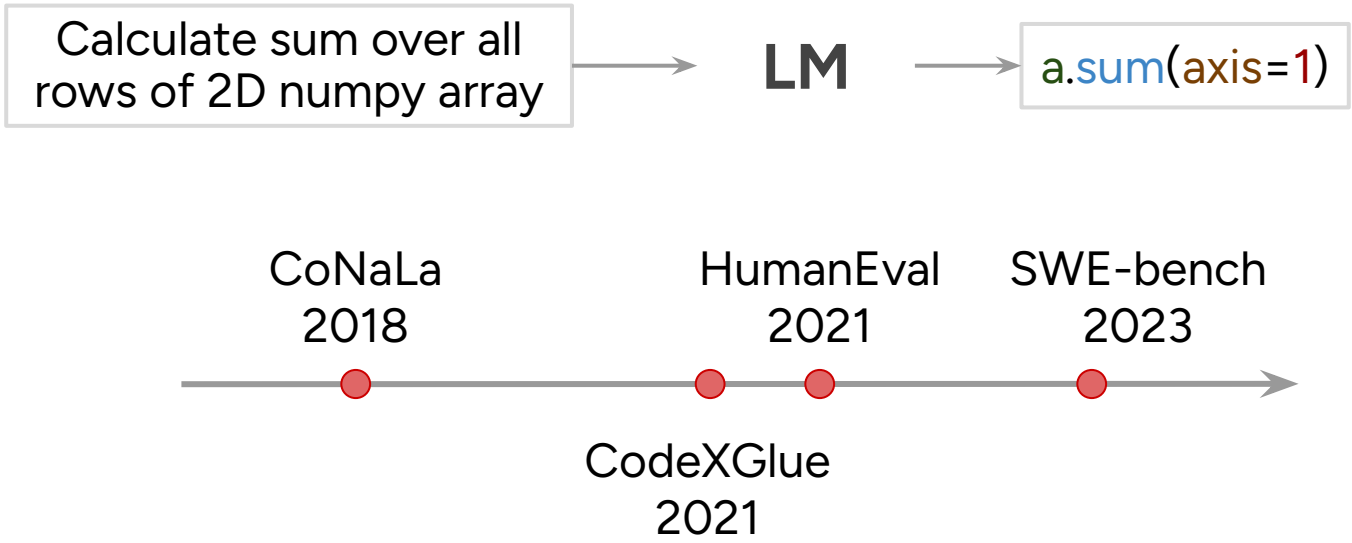


How Did We Get Here?



NL2Code: Natural Language to Code Generation

Given a natural language instruction *Q*, generate code implementation *C*.



CoNaLa: Poor on Single-Line Code Generation

I_4: *Python: take the content of a list and append it to another list*
URL: https://stackoverflow.com/questions/8177079/

Top Predictions:

S_1 `list2.append(list1)` ✗

S_2 `list2.extend(list1)` ✓

S_3 `list1.extend(mylog)` ✓

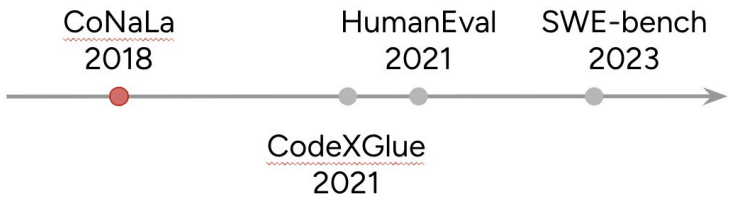
I_5: *Converting integer to string in Python?*
URL: https://stackoverflow.com/questions/961632/

Top Predictions:

S_1 `int('10')` ✗

S_2 `str(10); int('10')` ✗

S_3 `a.__str__()` ✓



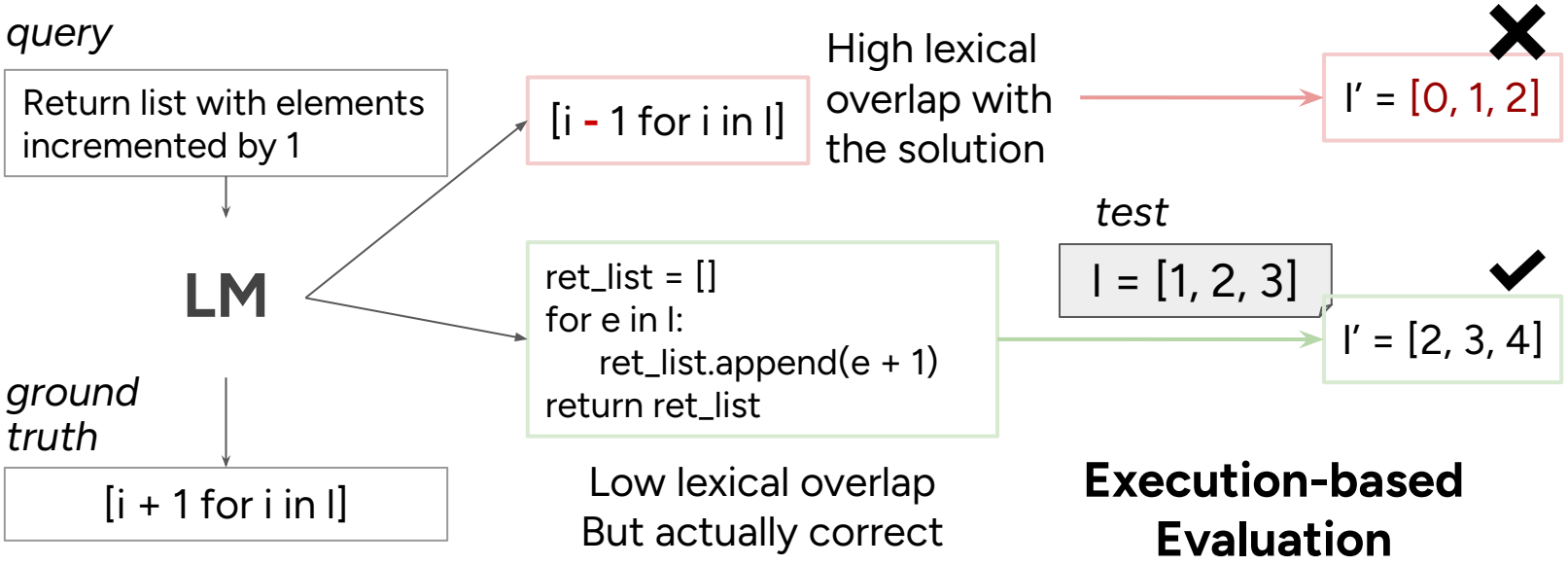
CodeXGLUE: Different Code-Related Tasks

Category	Task	Dataset Name	Language	Train/Dev/Test Size	Baselines
Code-Code	Clone Detection	BigCloneBench [71]	Java	900K/416K/416K	CodeBERT
		POJ-104 [52]	C/C++	32K/8K/12K	
	Defect Detection	Devign [99]	C	21K/2.7K/2.7K	
	Cloze Test	CT-all	Python,Java,PHP, JavaScript,Ruby,Go	-/-/176K	
		CT-max/min [18]	Python,Java,PHP, JavaScript,Ruby,Go	-/-/2.6K	
	Code Completion	PY150 [62]	Python	100K/5K/50K	CodeGPT
		Github Java Corpus[4]	Java	13K/7K/8K	
	Code Repair	Bugs2Fix [75]	Java	98K/12K/12K	Encoder-Decoder
	Code Translation	CodeTrans	Java-C#	10K/0.5K/1K	
Text-Code	NL Code Search	CodeSearchNet [35], AdvTest	Python	251K/9.6K/19K	CodeBERT
		CodeSearchNet [35], WebQueryTest	Python	251K/9.6K/1K	
	Text-to-Code Generation	CONCODE [38]	Java	100K/2K/2K	CodeGPT
Code-Text	Code Summarization	CodeSearchNet [35]	Python,Java,PHP, JavaScript,Ruby,Go	908K/45K/53K	Encoder-Decoder
Text-Text	Documentation Translation	Microsoft Docs	English-Latvian/Danish /Norwegian/Chinese	156K/4K/4K	

~110M params

Optimizing ‘False’ Code Correctness

Common metrics then: BLEU, CodeBLEU



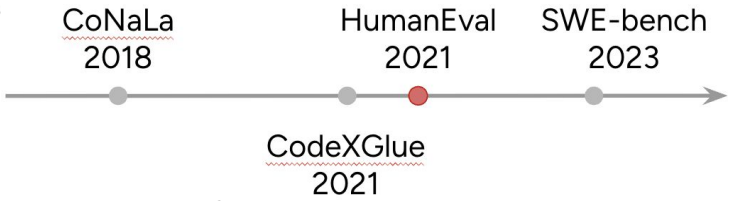
HumanEval: Evaluating Functional Correctness

```python
def incr_list(l: list):
    """Return list with elements incremented by 1.
    >>> incr_list([1, 2, 3])
    [2, 3, 4]
    >>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])
    [6, 4, 6, 3, 4, 4, 10, 1, 124]
    """
    return [i + 1 for i in l]
```

```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

NL instruction

Executable test cases

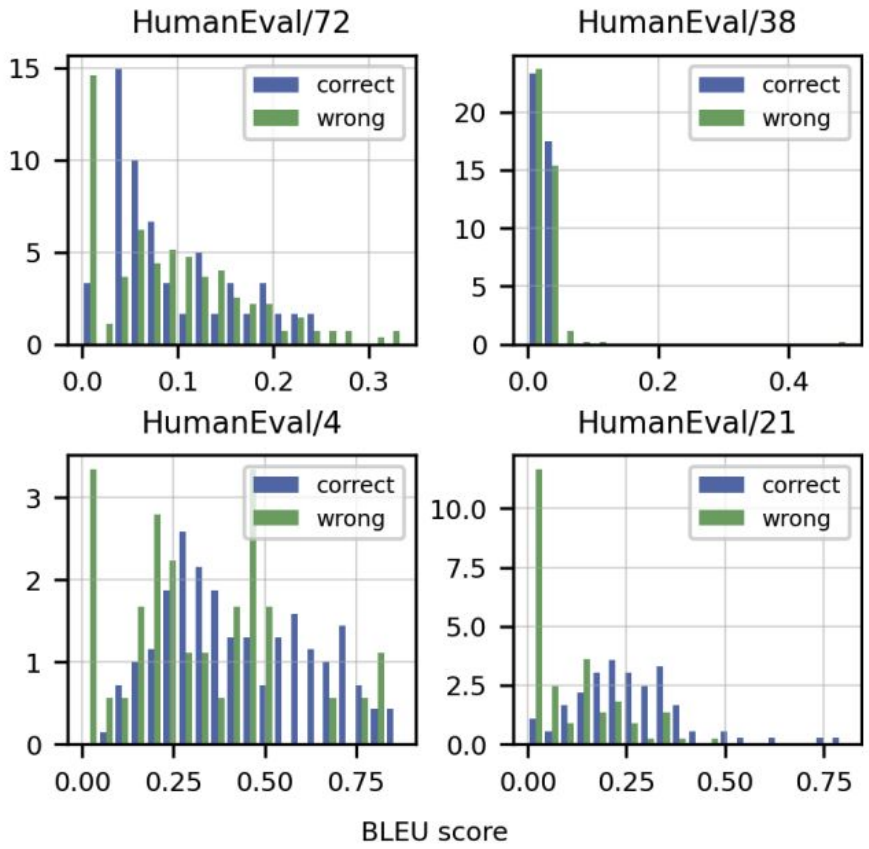


HumanEval: Human-Written Test Cases

164 hand-written examples

- Prevent evaluation contamination
- Safety: sandbox for code execution

Optimizing BLEU != Functional Correctne



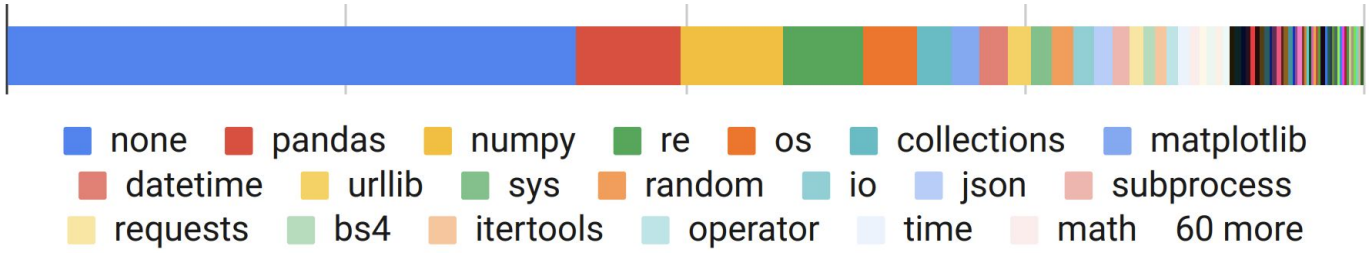
Increasing Complexity of Code Generation

Various domains:

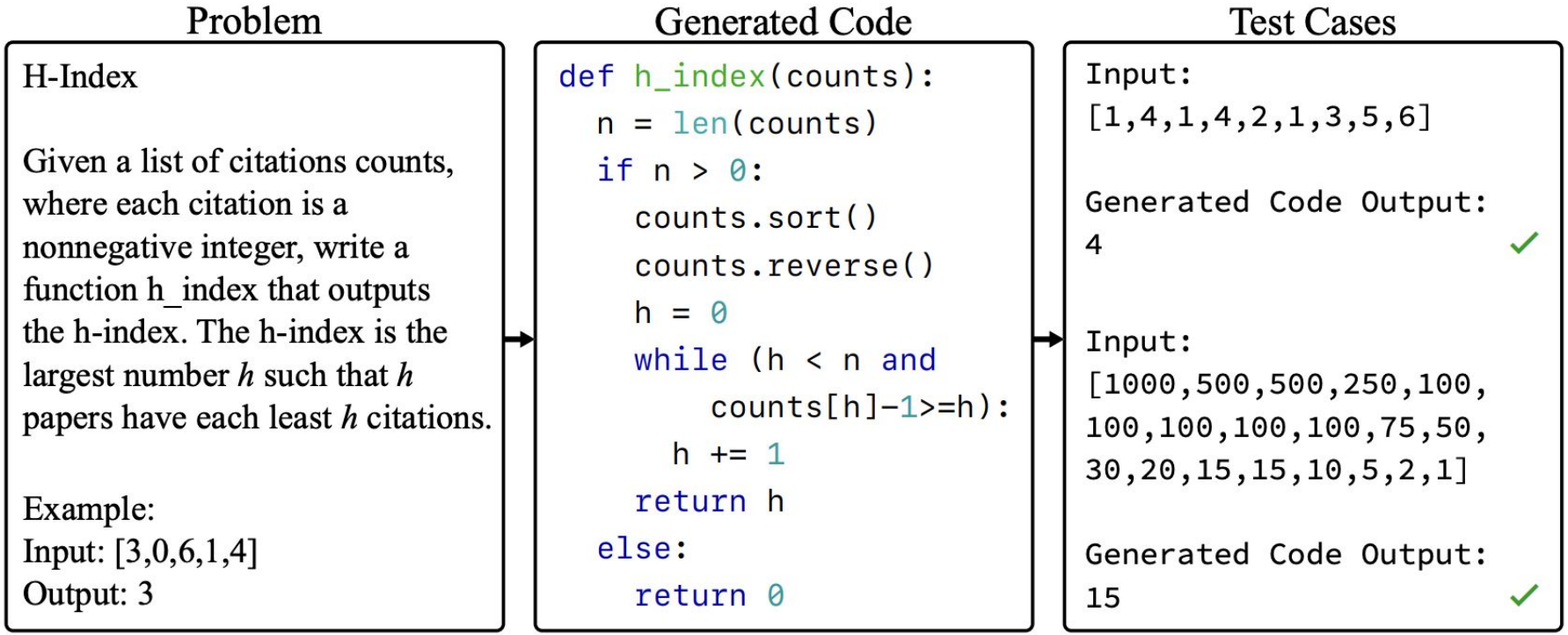
- data science: DS-1000, ARCADE
- common Python libraries: ODEX

```
[1] import pandas as pd

c1  df = pd.read_csv('dataset/Gamepass_Games_v1.csv')
```

```
[2] u1 Extract min and max hours as two columns
def get_avg(x):
  try: return float(x[0]) , float(x[1])
  except: return 0, 0

c2  df['min'], df['max'] = zip(*df['TIME'].str.replace(
  ' hours','').str.split("-").apply(get_avg))
```



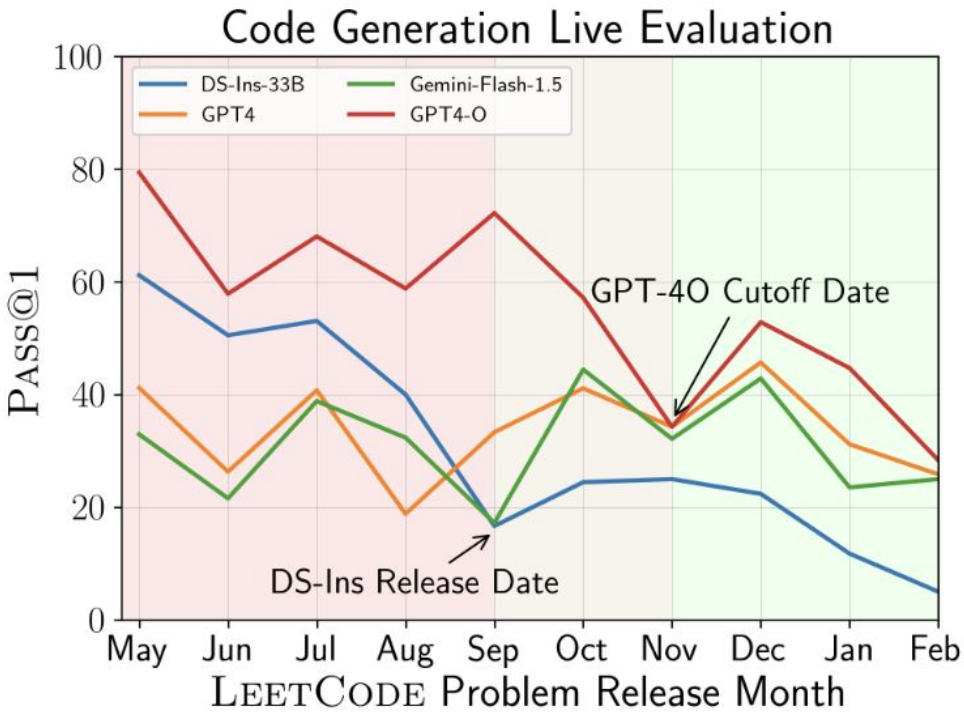
Increasing Complexity of Code Generation



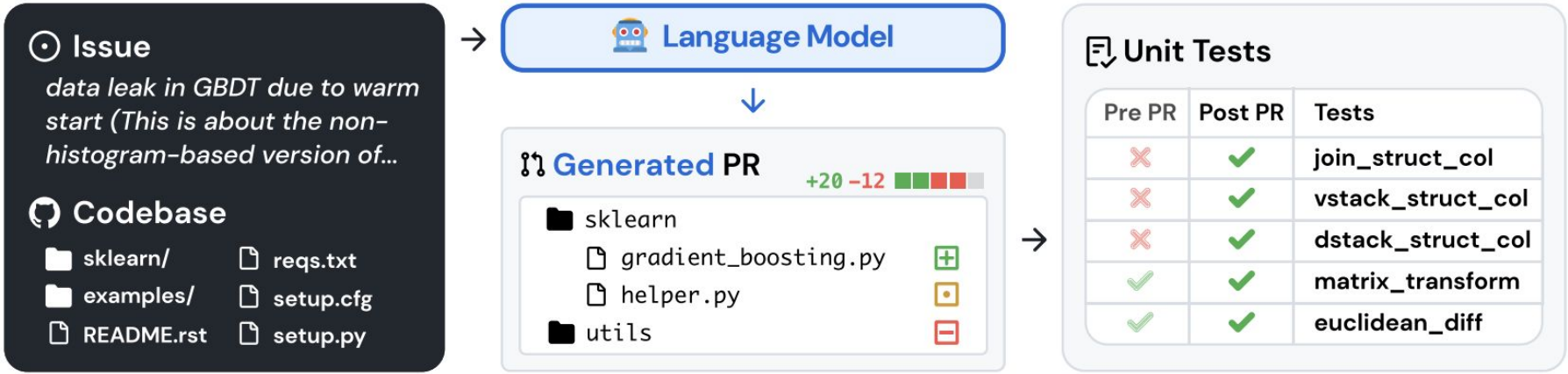
LiveCodeBench: Contamination-Free Evaluation

Performance drop around

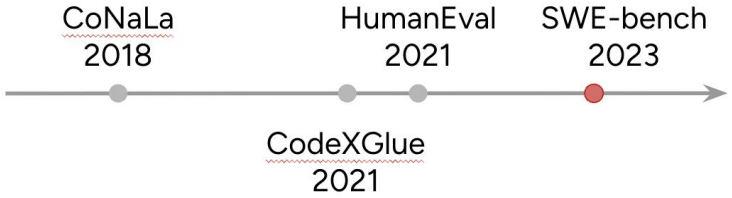
model training cutoff data



SWE-bench: Resolve Real-World GitHub Issues



- More complex coding contexts: single file → large codebase
- More complex coding output: function completion → script edits
- Realistic: high-quality GitHub issues & tests



Quiz

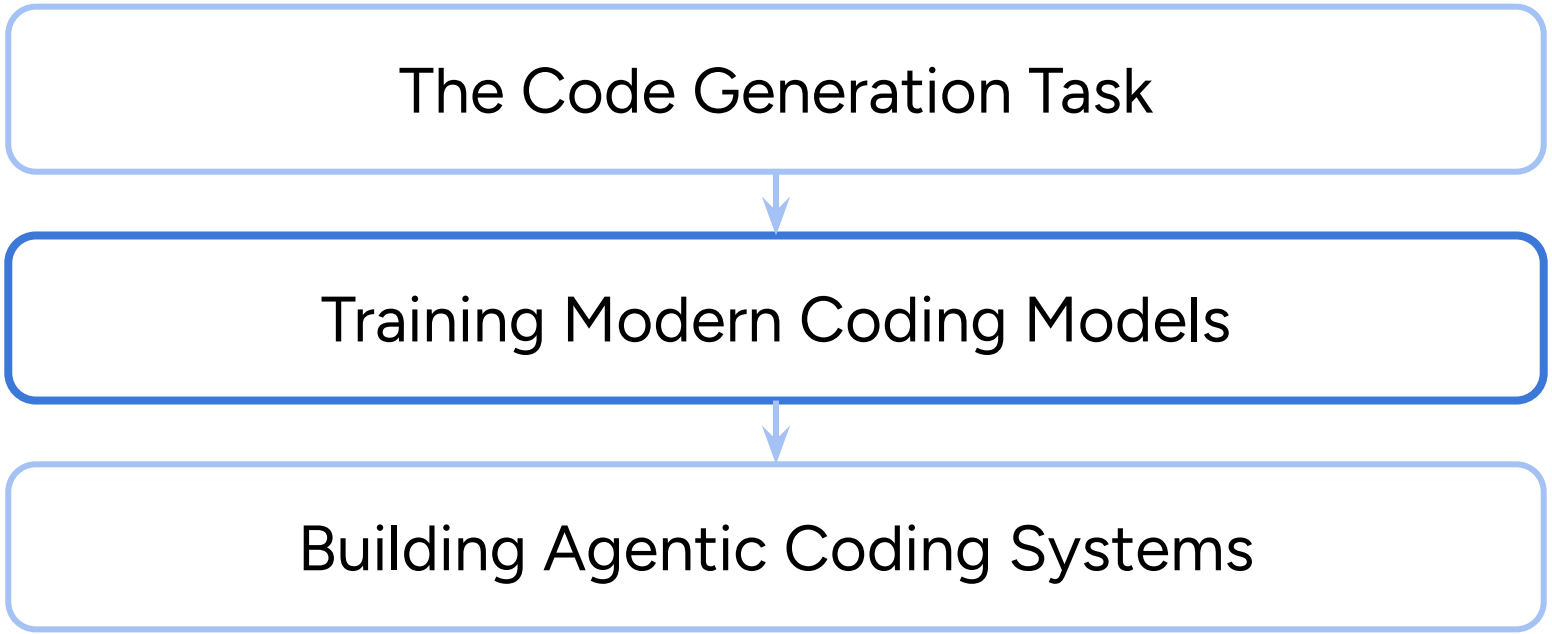
What is the biggest difference between HumanEval and previous datasets?

A. Manually annotated
B. Python-only
C. Leetcode-style problems
D. Execution-based evaluation

Why do people avoid execution-based evaluation before 2021?

A. Not necessary to for evaluating model correctness
B. The coding tasks cannot support execution
C. Environment setup is hard
D. Others

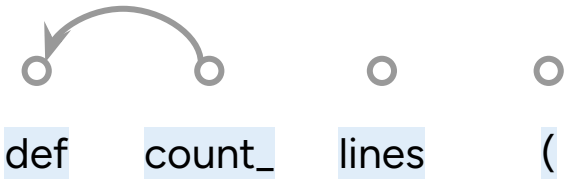
How Did We Get Here?



Unidirectional vs. Bidirectional Transformers

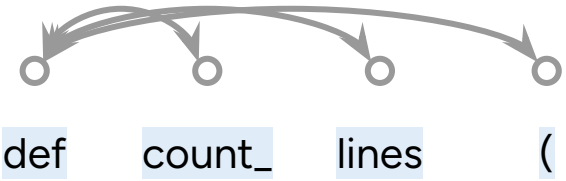
- **Uni**-directional Transformers

Each token has information about its **previous** tokens



- **Bi**-directional Transformers

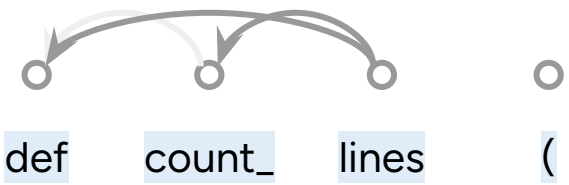
Each token has information about its **all** tokens



Unidirectional vs. Bidirectional Transformers

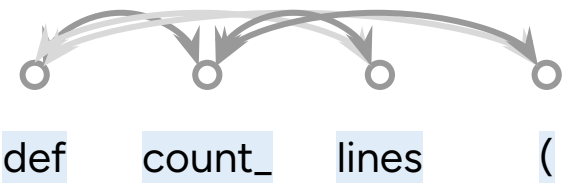
- **Uni**-directional Transformers

Each token has information about its **previous** tokens



- **Bi**-directional Transformers

Each token has information about its **all** tokens

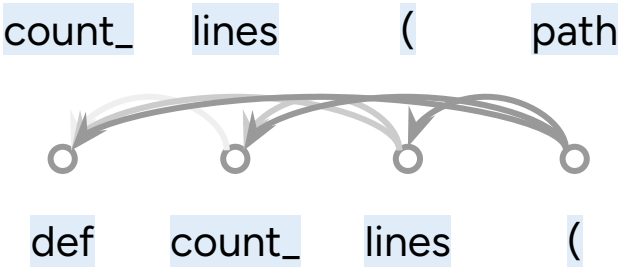


Training Objective

Autoregressive Language Modeling

~ **Uni**directional

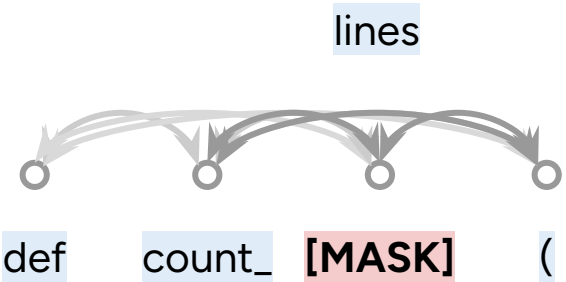
$$P(X) = \prod_{i=1}^{|X|} P(x_i | x_1, \ldots, x_{i-1})$$



Masked Language Modeling

~ **Bi**directional

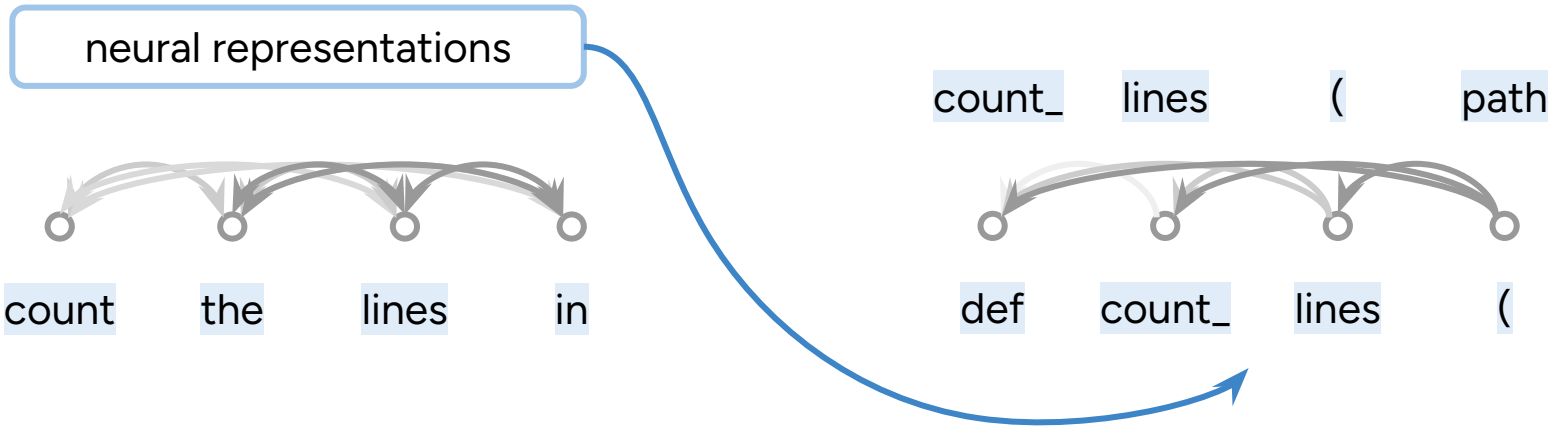
$$P(X) \neq \prod_{i=1}^{|X|} P(x_i | x_{\neq i})$$



Training Objective: Sequence-to-Sequence (seq2seq)

Encoder: Bidirectional Transformers

Decoder: Unidirectional Transformers



CodeBERT: Example of Masked Language Modeling

- 125M model parameters
- Train on
  - 2B text+code data
    - CodeSearchNet
  - 6B code-only data
    - GitHub

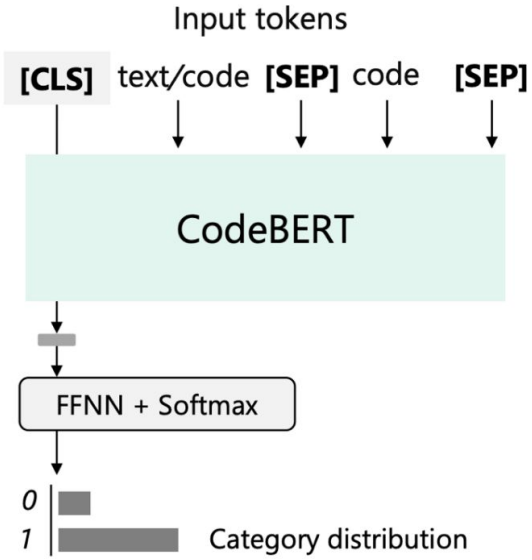
Training Data	*bimodal* data	*unimodal* Codes
Go	319,256	726,768
Java	500,754	1,569,889
JavaScript	143,252	1,857,835
PHP	662,907	977,821
Python	458,219	1,156,085
Ruby	52,905	164,048
All	2,137,293	6,452,446

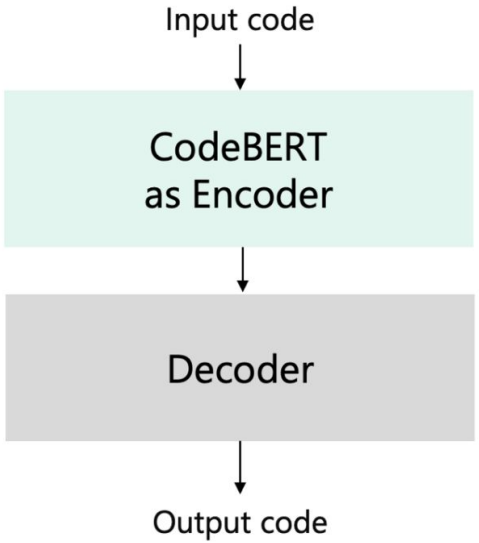
CodeBERT: Example of Masked Language Modeling

- 125M model parameters
- Train on
  - 2B text+code data
  - 6B code-only data

Task-specific Fine-tuning

- Classification: code search
- Generation





CodeT5: Example of Seq2Seq

- General seq2seq: masked span prediction task
- Code-specific: identifier tagging to learn code semantics

60M and 220M params

Trained on 6.4M instances

- CodeSearchNet
- BigQuery

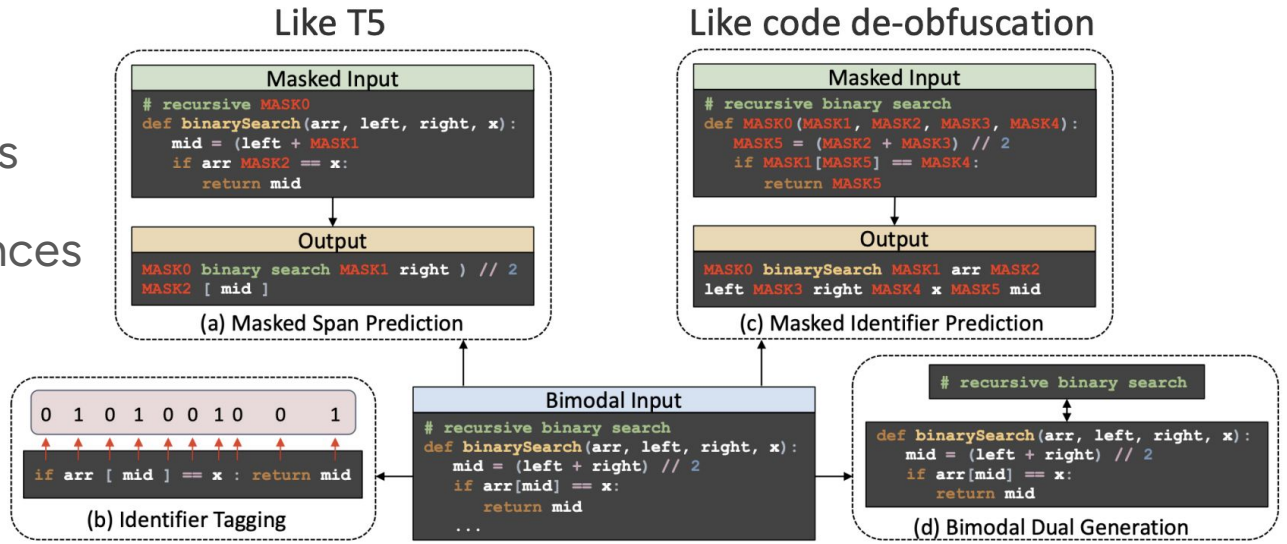
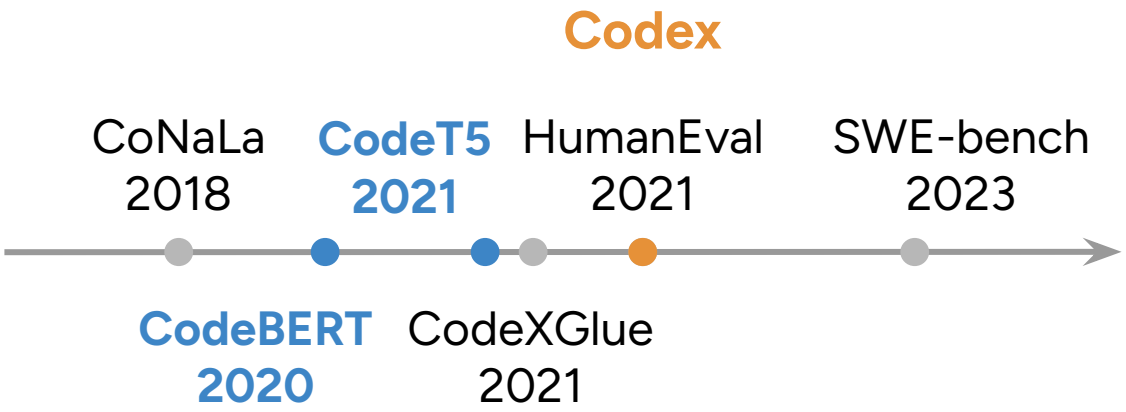


Figure 2: Pre-training tasks of CodeT5. We first alternately train span prediction, identifier prediction, and identifier tagging on both unimodal and bimodal data, and then leverage the bimodal data for dual generation training.

Timeline


Codex
CoNaLa
2018
CodeT5
2021
HumanEval
2021
SWE-bench
2023
CodeBERT
2020
CodeXGlue
2021

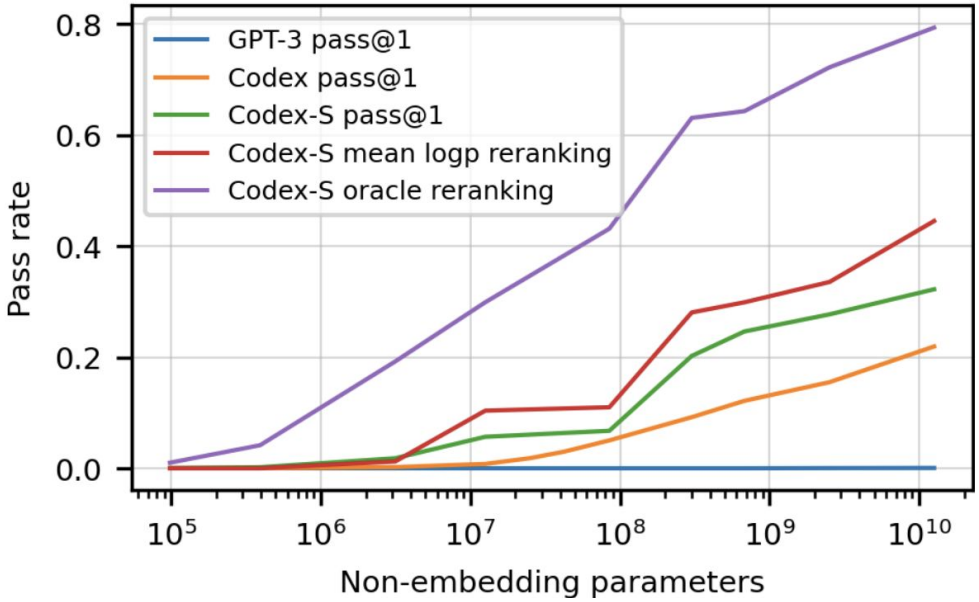
OpenAI Codex: Example of Autoregressive LM

12B model parameters

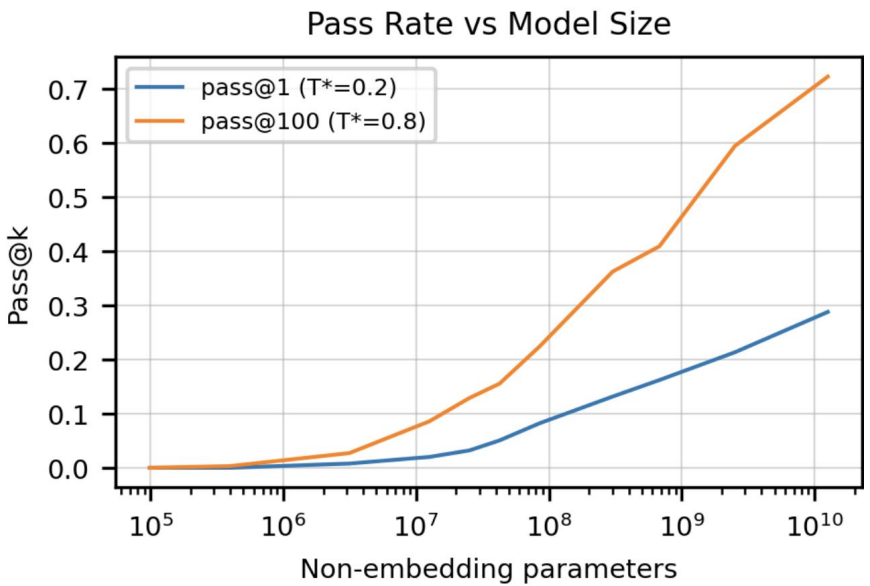
Train GPT-3 with 160GB of Python data from GitHub

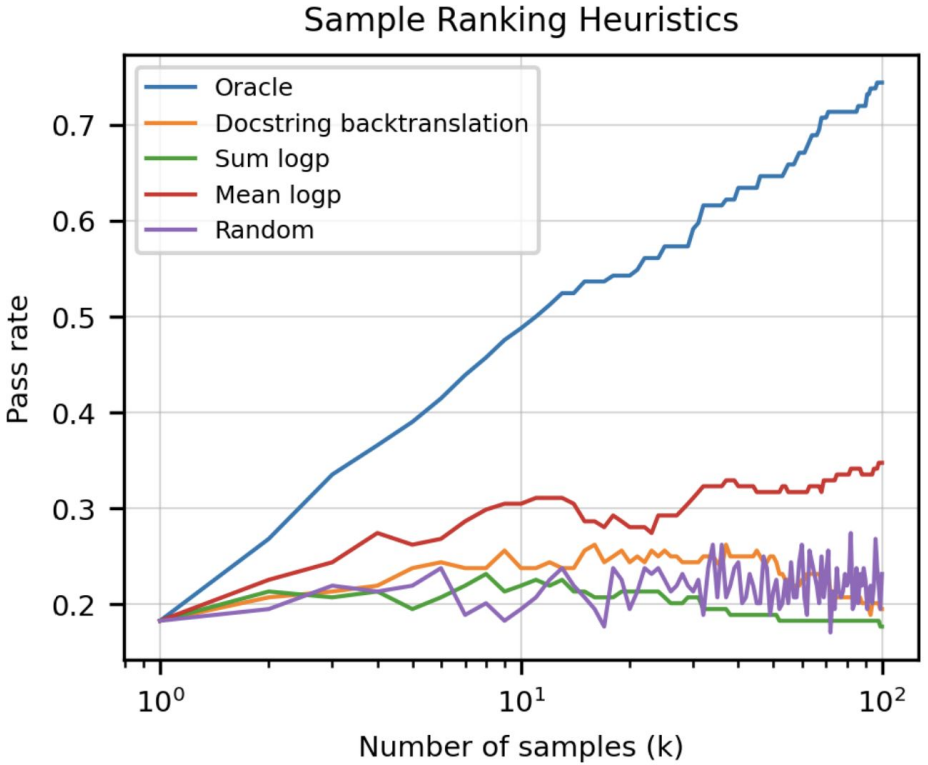
Codex: finetuned on code

Codex-S: + correct functions only



OpenAI Codex: Sampling-Based Generation





Exemplar Training Pipelines

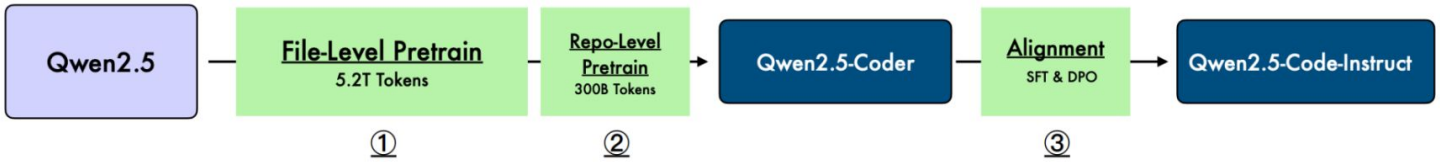


Figure 2: The three-stage training pipeline for Qwen2.5-Coder.

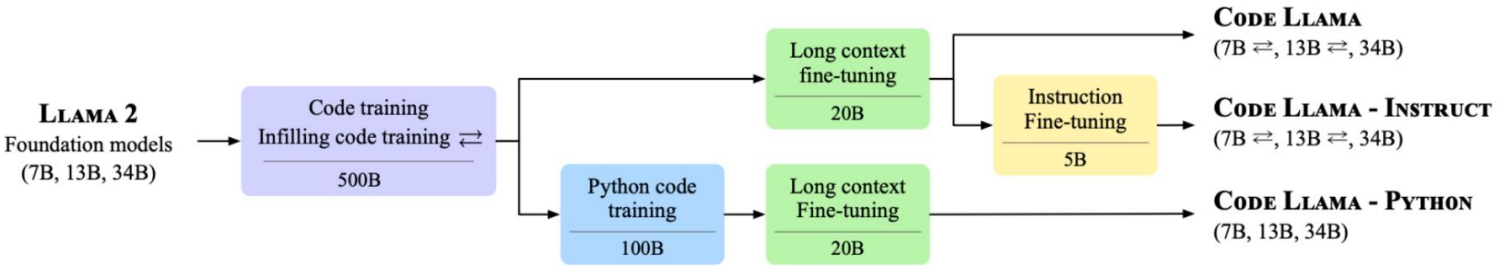
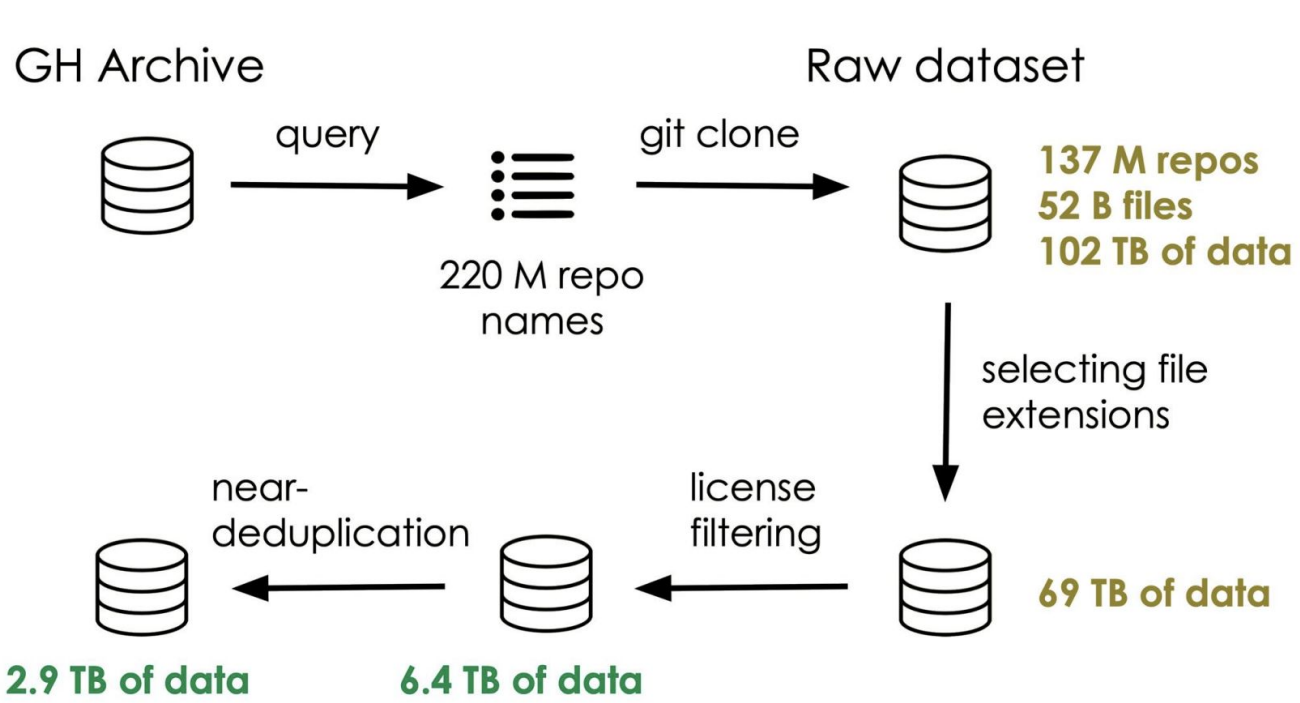
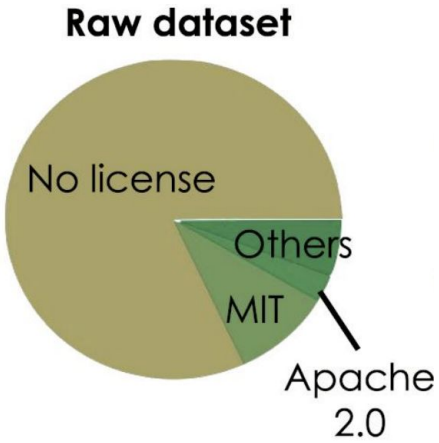


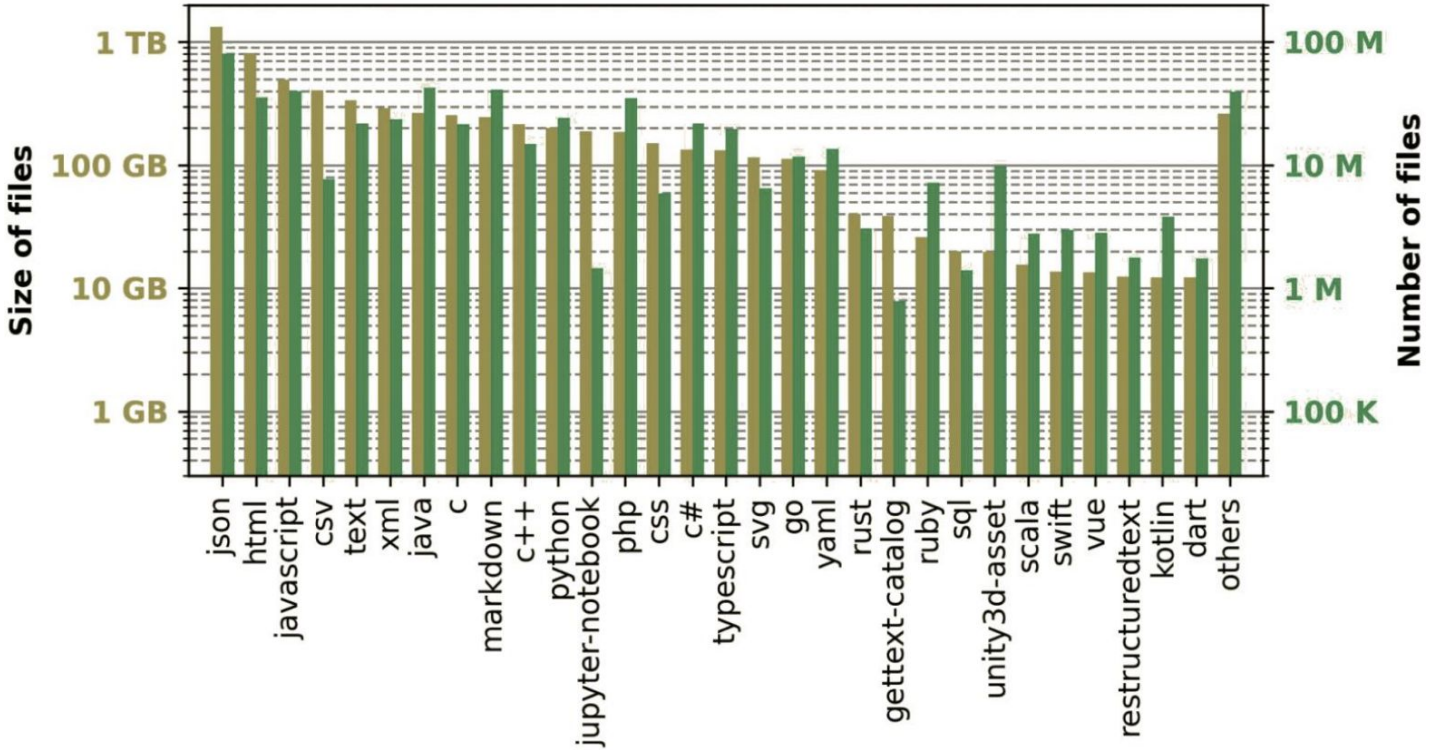
Figure 2: **The Code Llama specialization pipeline.** The different stages of fine-tuning annotated with the number of tokens seen during training. Infilling-capable models are marked with the ⇄ symbol.

Data: The Stack





Data: The Stack



Data: The Stack

- Possible to match Codex performance on HumanEval
- License filtering hurts; deduplication always improve performance

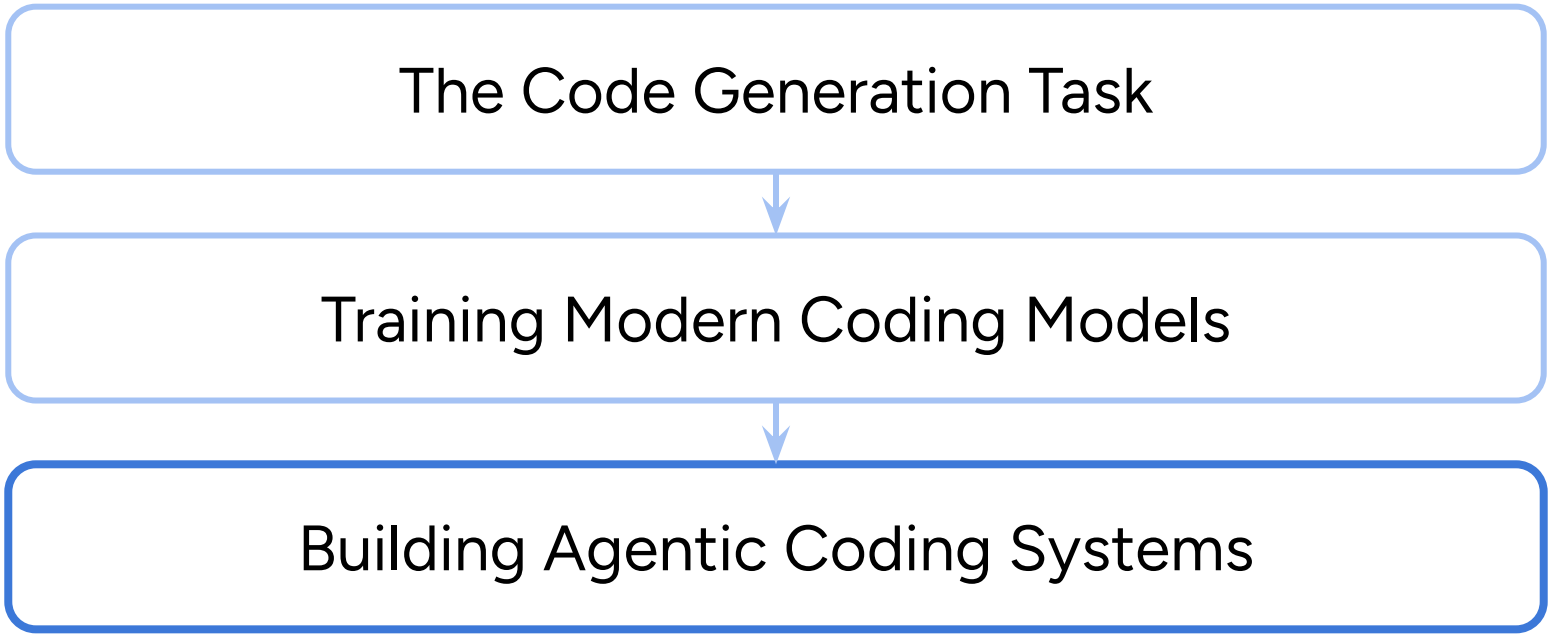
Dataset	Filtering	pass@1	pass@10	pass@100	Python Data
Codex (300M)	Exact-dedup?	13.17	20.17	36.27	180 GB
CodeGen (350M)	unknown	12.76	23.11	35.19	
Python all-license	None	13.11	21.77	36.67	740 GB
	Near-dedup	**17.34**	**27.64**	**45.52**	
Python permissive-license	None	10.99	15.94	27.21	191 GB
	Near-dedup	**12.89**	**22.26**	**36.01**	80 GB

Quiz

What do you think is the most important factor behind Codex’s power? Why

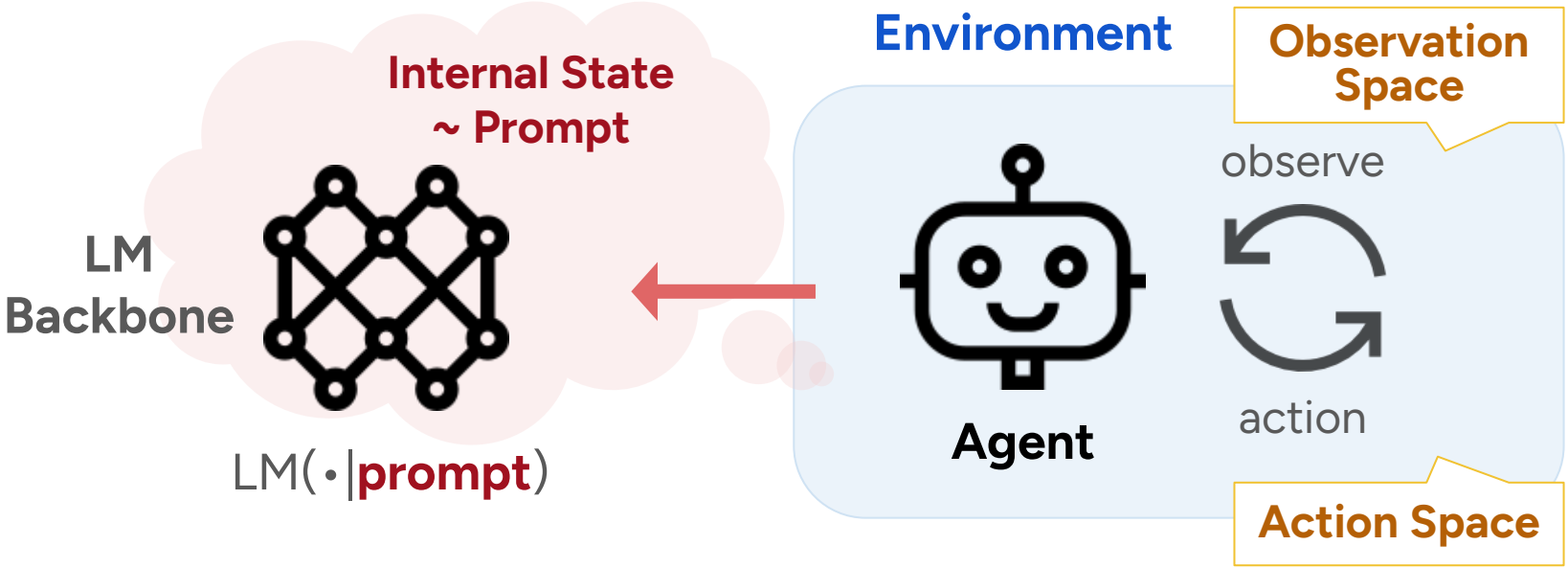
A. Model architecture
B. Model size (i.e., the number of parameters)
C. Data scale
D. Data quality
E. Others

How Did We Get Here?



What Is A Coding Agent?

"An intelligent agent perceives its environment via sensors and acts rationally upon that environment with its effectors."



SWE-agent

Agent-computer interface (ACI) design affects performance

Issue

data leak in GBDT due to warm start (This is about the non-histogram-based version of...

Codebase

sklearn/ reqs.txt
examples/ setup.cfg
README.rst setup.py

Observation (File Viewer)

```
[File: solvers/diophantine.py (3093 lines total)]
(400 lines above)
401:    elif total_degree == 3:
402:        diop_type = "cubic_thue"
403:
404:    elif (total_degree > 3):
405:        if all(coeff[k] == 1):
406:            diop_type = 'general_sum_even'
407:
408:    if diop_type is not None:
409:        return var, coeff, diop_type
(2684 lines below)
```

Response (Thought + Edit Action)

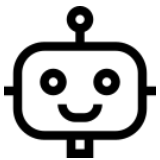
We need to add a check to ensure that all powers are the same. Let's edit the code to include this check.

```
edit 404:407
    elif (total_degree > 3 and
          len(set(k.exp[k.is_Pow])) == 1):
        if all(coeff[k] == 1):
            diop_type = 'general_sum_even'
end_of_edit
```

(a) Observation from the file viewer.

(b) Action using the edit interface.

observe



action

John Yang, Carlos E. Jimenez, et al.. "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering."

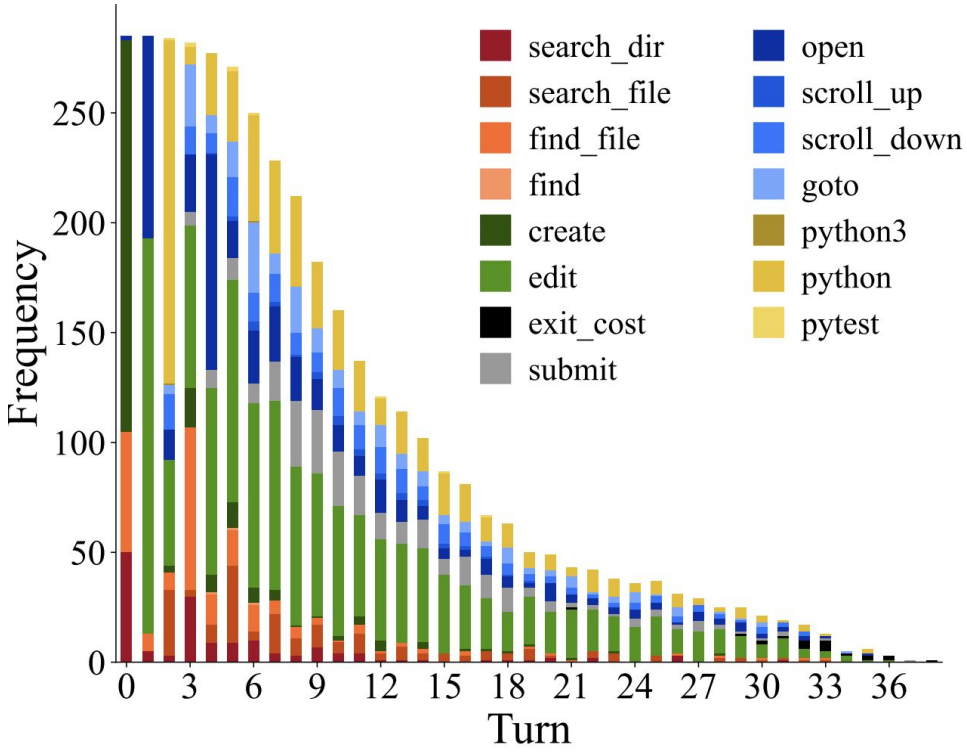
SWE-agent: Analysis of Agent Behavior

Within a session

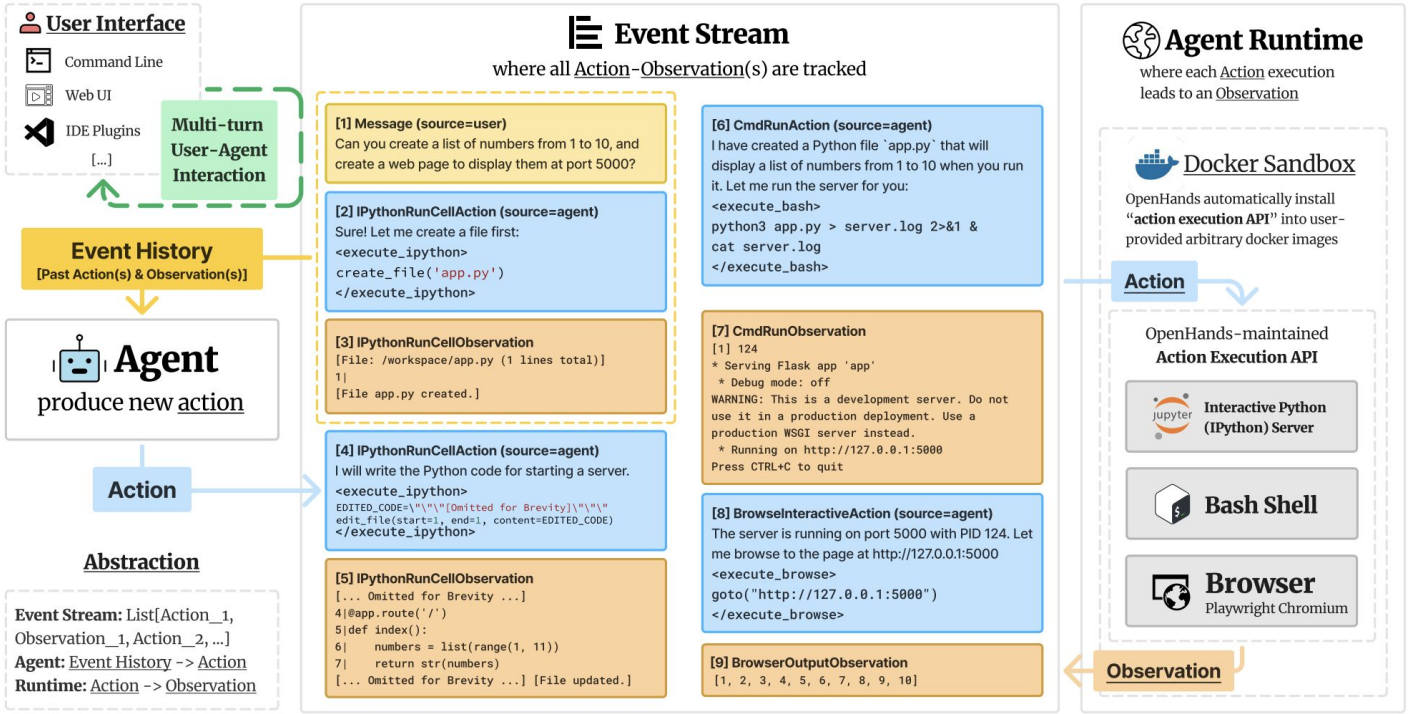
- reproduction/localization is the first step
- remaining: "edit then execute" loops

Overall

- agents succeed quickly and fail slowly
- most failures are incorrect implementations



OpenHands (OpenDevin)



Different

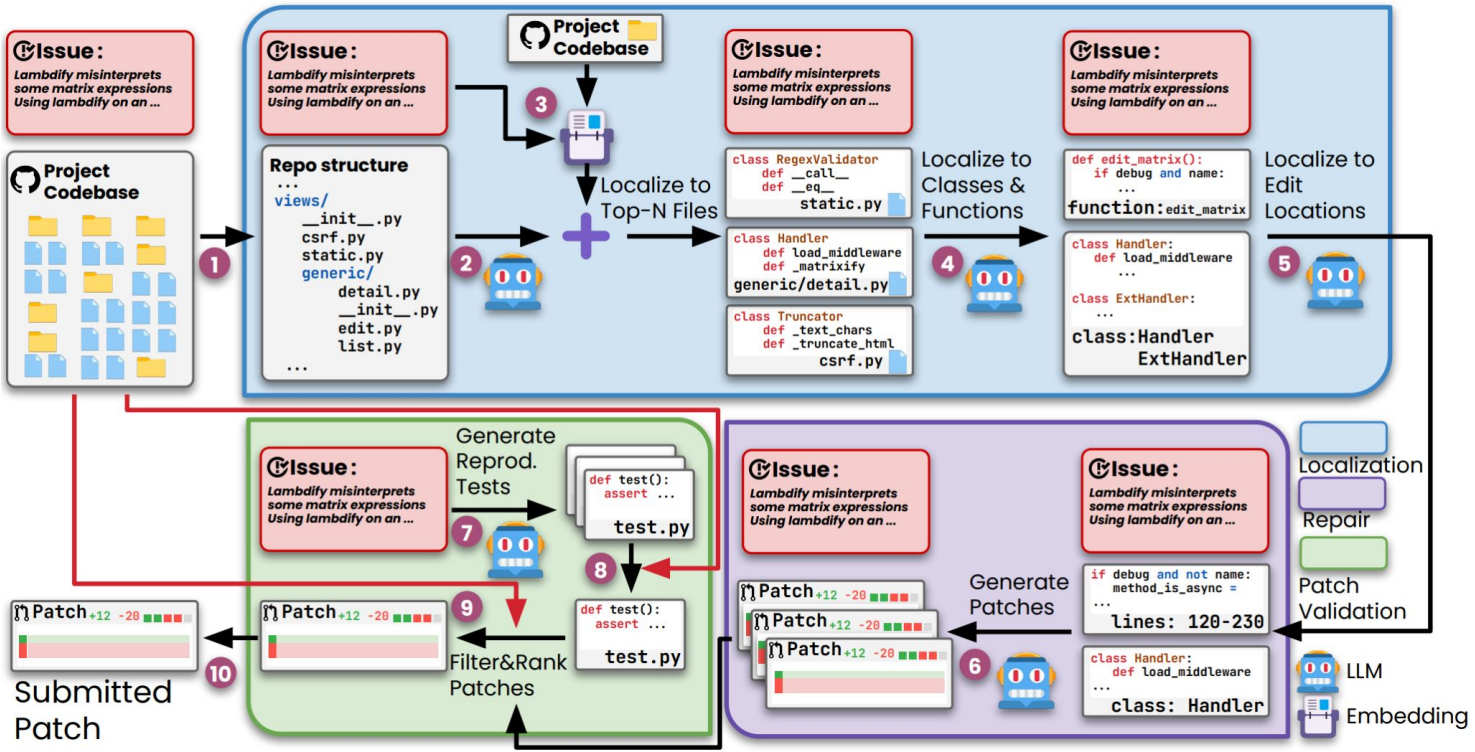
- Actions (a.k.a.tools)
- Observation

Still loops of

- **Read file**
- **Edit + exec code files**

Xingyao Wang, et al.. "OpenHands: An Open Platform for AI Software Developers as Generalist Agents"

Agentless



Chunqiu Steven Xia, et al.. "Agentless: Demystifying LLM-based Software Engineering Agents"

Performance (2024.07)

Fixed two-stage pipeline >>better than>>agents with iterative loops

Tool	LLM	% Resolved	Avg. $ Cost	Avg. # Tokens	% Correct Location Line	Function	File
AutoCodeRover-v2 [6]	GPT-4o	92 (30.67%)	-	-	35.0%	52.3%	69.3%
RepoGraph [19]	GPT-4o	89 (29.67%)	-	-	36.7%	51.3%	71.0%
Moatless [15]	Claude 3.5 S	80 (26.67%)	$0.17	-	38.7%	54.7%	78.7%
	GPT-4o	74 (24.67%)	$0.14	-	36.0%	52.0%	73.0%
OpenDevin+CodeAct v1.8 [17]	Claude 3.5 S	80 (26.67%)	$1.14	-	38.0%	49.7%	67.3%
Aider [37]	GPT-4o+Claude 3.5 S	79 (26.33%)	-	-	35.3%	50.0%	69.7%
SWE-agent [101]	Claude 3.5 S	69 (23.00%)	$1.62	521,208	40.7%	54.3%	72.0%
	GPT-4o	55 (18.33%)	$2.53	498,346	29.3%	42.3%	58.3%
	GPT-4	54 (18.00%)	$2.51	245,008	30.7%	45.3%	61.0%
AppMap Navie [5]	GPT-4o	65 (21.67%)	-	-	29.7%	44.7%	59.7%
AutoCodeRover [108]	GPT-4	57 (19.00%)	$0.45	38,663	29.0%	42.3%	62.3%
RAG [101]	Claude 3 Opus	13 (4.33%)	$0.25	-	22.0%	30.0%	57.0%
	GPT-4	8 (2.67%)	$0.13	-	12.7%	23.3%	47.3%
	Claude-2	9 (3.00%)	-	-	16.7%	24.3%	46.7%
	GPT-3.5	1 (0.33%)	-	-	6.3%	11.3%	27.3%
AGENTLESS	GPT-4o	96 (32.00%)	$0.70	78,166	35.3%	52.0%	69.7%

Fixed Pipeline Better Than Iterative Agents?

Core issue

- Backbone LM too weak (back in 2023): GPT-4, claude-3.5
- To correctly use observation (read-file) and action (edit file) tools

Agents need to

- Graduate from wraps around static LMs
- To training LM parameters

Major Limitations: each task instance needs

- Specialized environment (e.g., install packages)
- Execution-based verifiers (i.e., test cases)

SWE-Gym: Training SE Agents + Verifiers

Collect more task instances following the SWE-bench approach

Category	Metric	SWE-Gym	SWE-Gym Lite
Size	# Instances	2,438 (2,294)	230 (300)
	# Repos	11 (12)	11 (12)
Issue Text	Length by Words	239.8 (195.1)	186.2 (175.9)
Codebase	# Non-test Files	971.2 (2944.2)	818.8 (2988.5)
	# Non-test Lines	340675.0 (363728.4)	340626.2 (377562.4)
Gold Patch	# Lines edited	69.8 (32.8)	10.6 (10.1)
	# Files edited	2.5 (1.7)	1.0 (1.0)
	# Func. edited	4.1 (3.0)	1.4 (1.34)
Tests	# Fail to Pass	10.0 (9.0)	2.04 (3.5)
	# Total	760.8 (132.5)	99.9 (85.2)

e 2: Statistics comparing SWE-Gym with the SWE-Bench test split (in parenthesis).
ept for size metrics, we report the average value across instances.

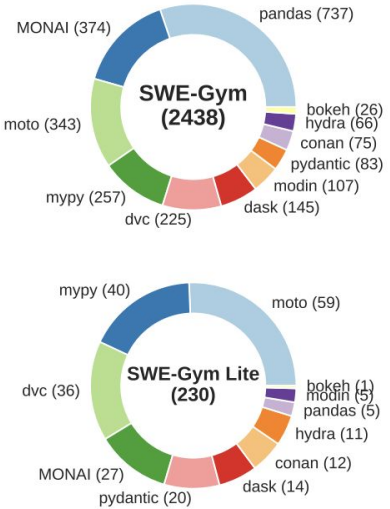


Figure 2: Repository distribution of SWE-Gym instances.

Jiayi Pan, et al.. "Training Software Engineering Agents and Verifiers with SWE-Gym"

SWE-Gym: Training & Test-Time Scaling

Instance and repository diversity is not yet a bottleneck

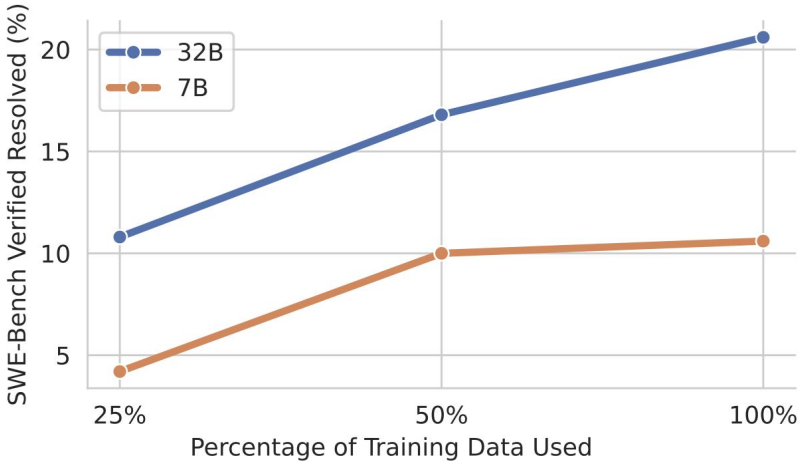
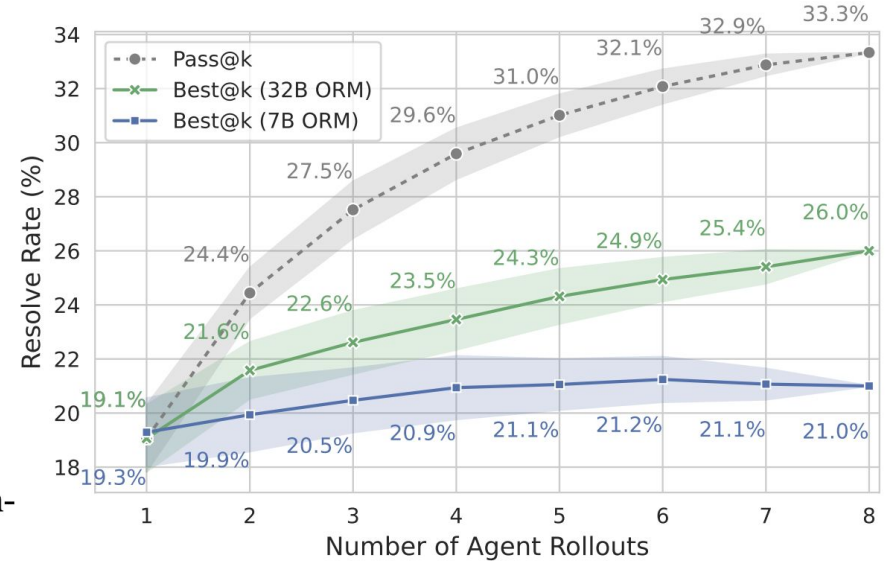


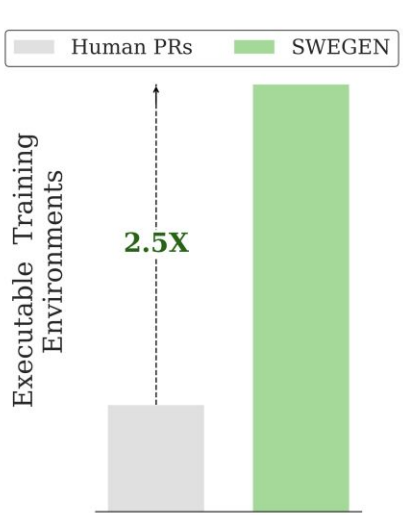
Figure 5: Scaling effects of increasing the number of randomly sampled trajectories for training.

Trained verifiers help solution selection at inference time

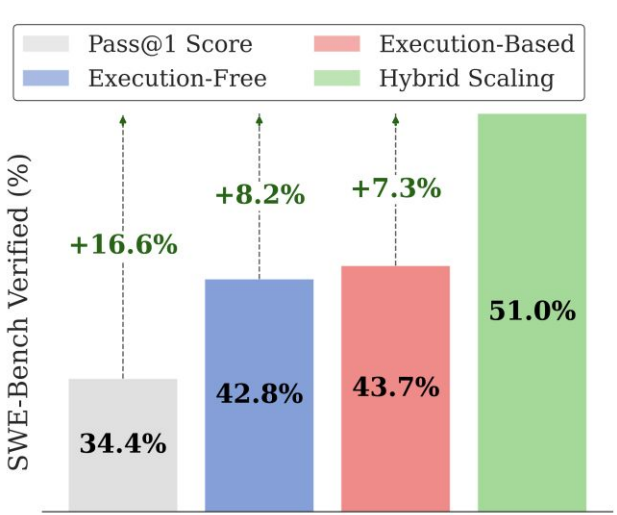


R2E-Gym: Synthesized Envs + Dual Verifiers

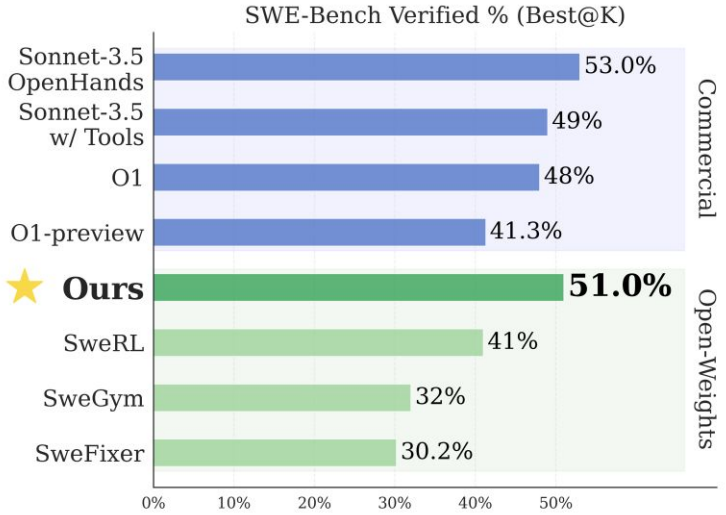
- synthesize data: executable environments from GitHub commits
- execution-based + execution-free verifiers



(a) Synthetic Data



(b) Hybrid Test-time Scaling



(c) Open-weights SOTA Performance

Naman Jain, et al.. "Training Software Engineering Agents and Verifiers with SWE-Gym"

SWE-Smith

Consider adding:

- SWE-smith
- SWE-RL: Advancing LLM Reasoning via Reinforcement Learning on Open Software Evolution
- CWM: An Open-Weights LLM for Research on Code Generation with World Models

Summary: AI for Code Today

Constructing more difficult benchmarks

- SWE-bench*
- MLE-bench, TerminalBench, ...

Building sophisticated agent frameworks

- SWE-agent, OpenHands, ...

Scalable training with effective verification

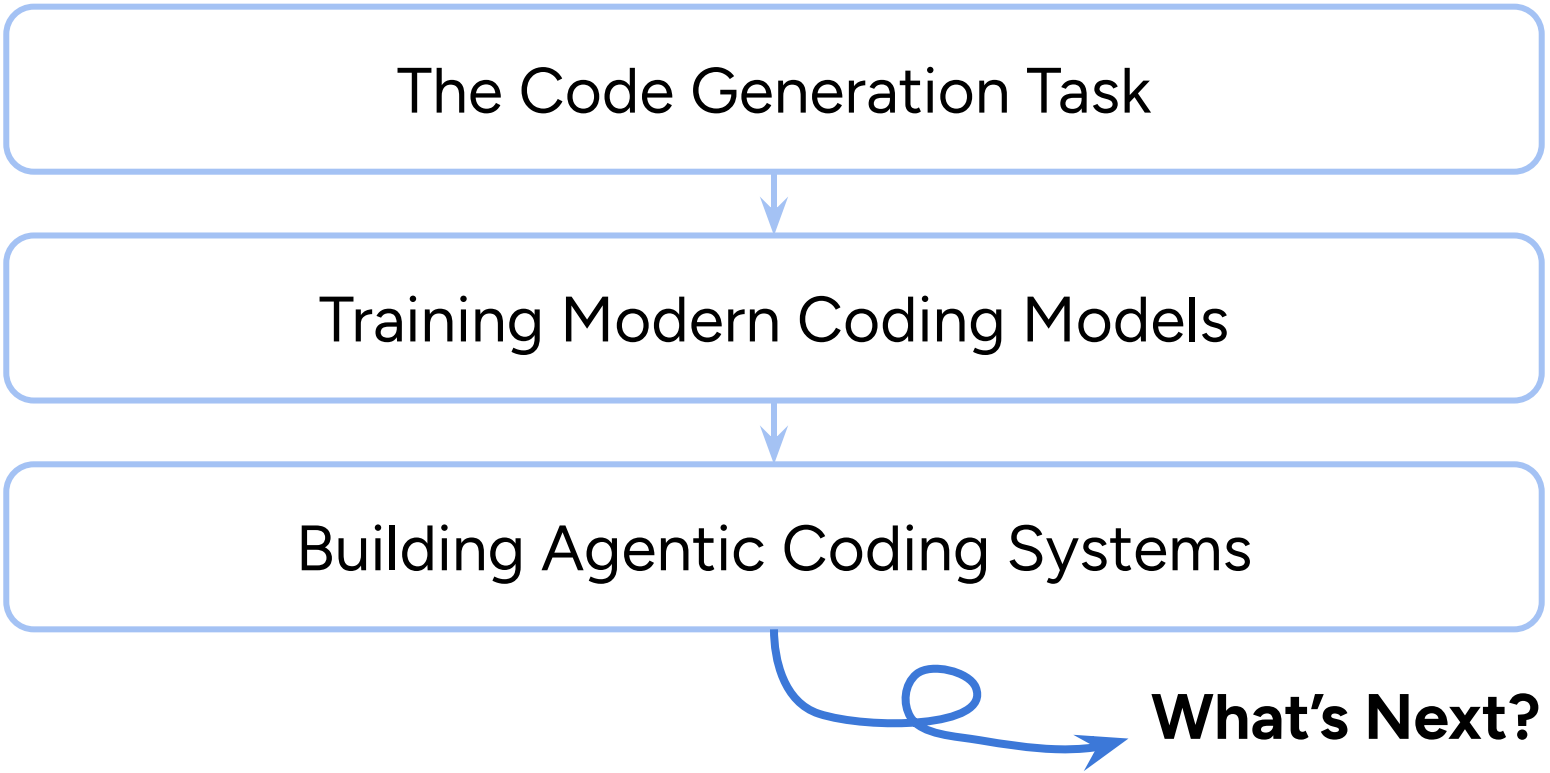
- SWE-gym, R2E-Gym

Quiz

What is needed for a better coding agent? Besides backbone code LM. Why?

A. Coding environment
B. Agent design (observation/action space, prompt, …)
C. Others

How Did We Get Here?

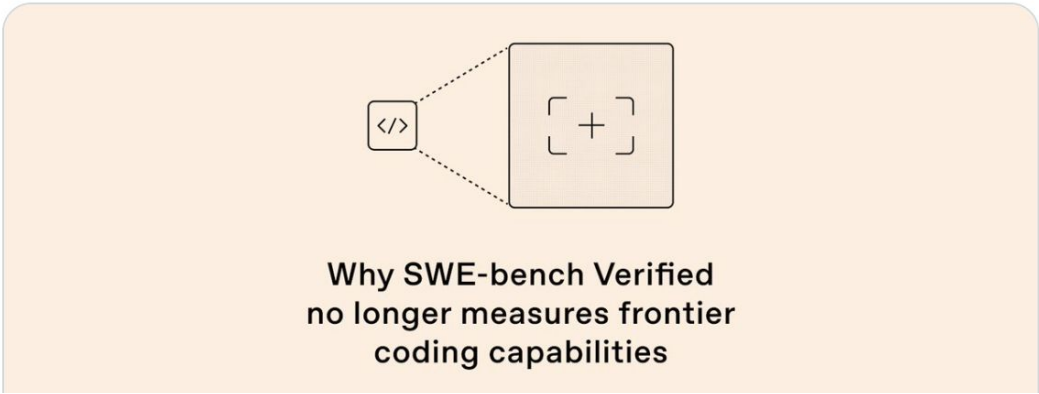


We’re Saturating SWE-Bench

OpenAI Developers
@OpenAIDevs

The standard for frontier coding evals is changing with model maturity.

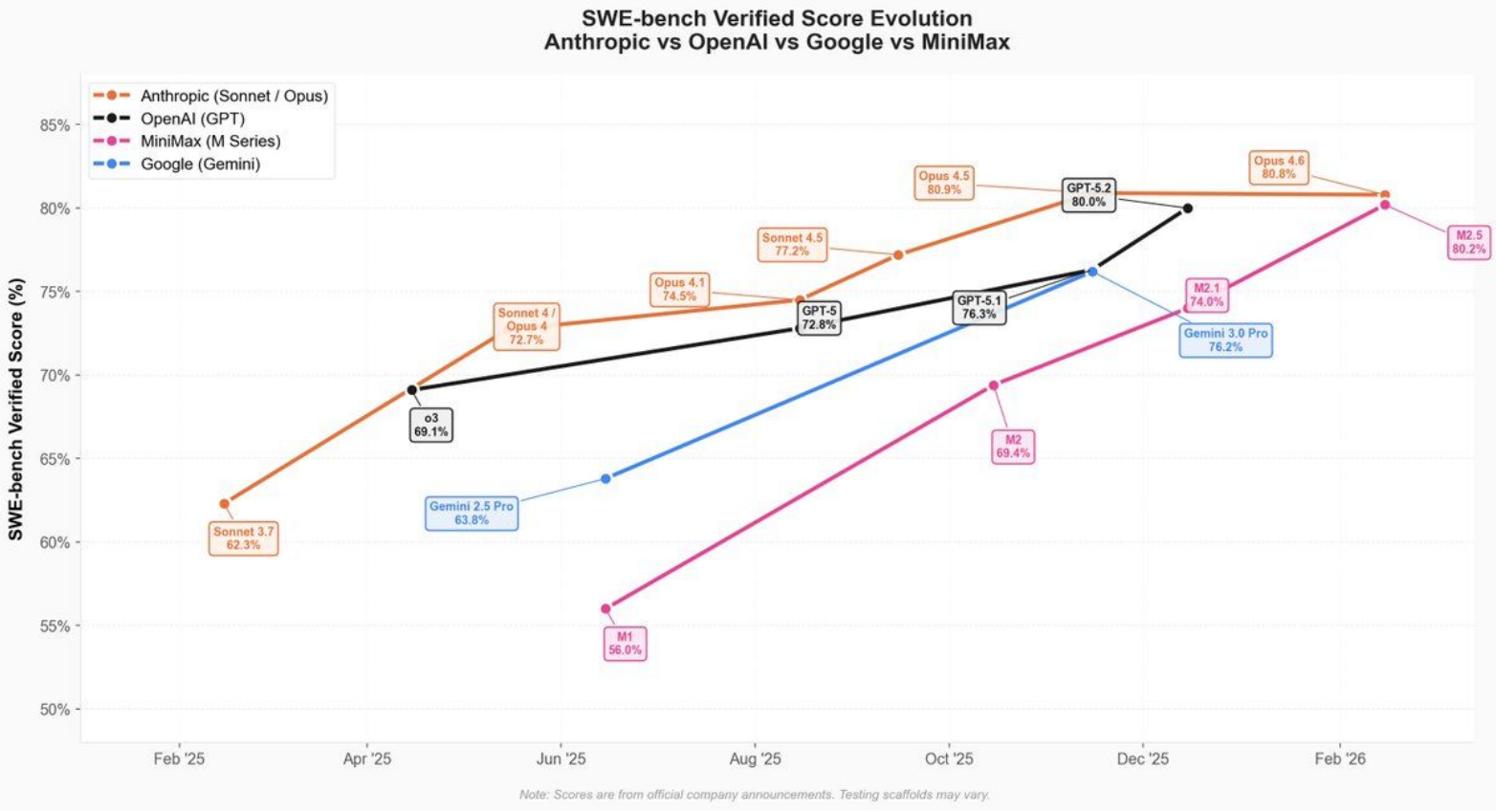
SWE-bench Verified was a strong benchmark, but we’ve found evidence it is now saturated due to test-design issues and contamination from public repositories.



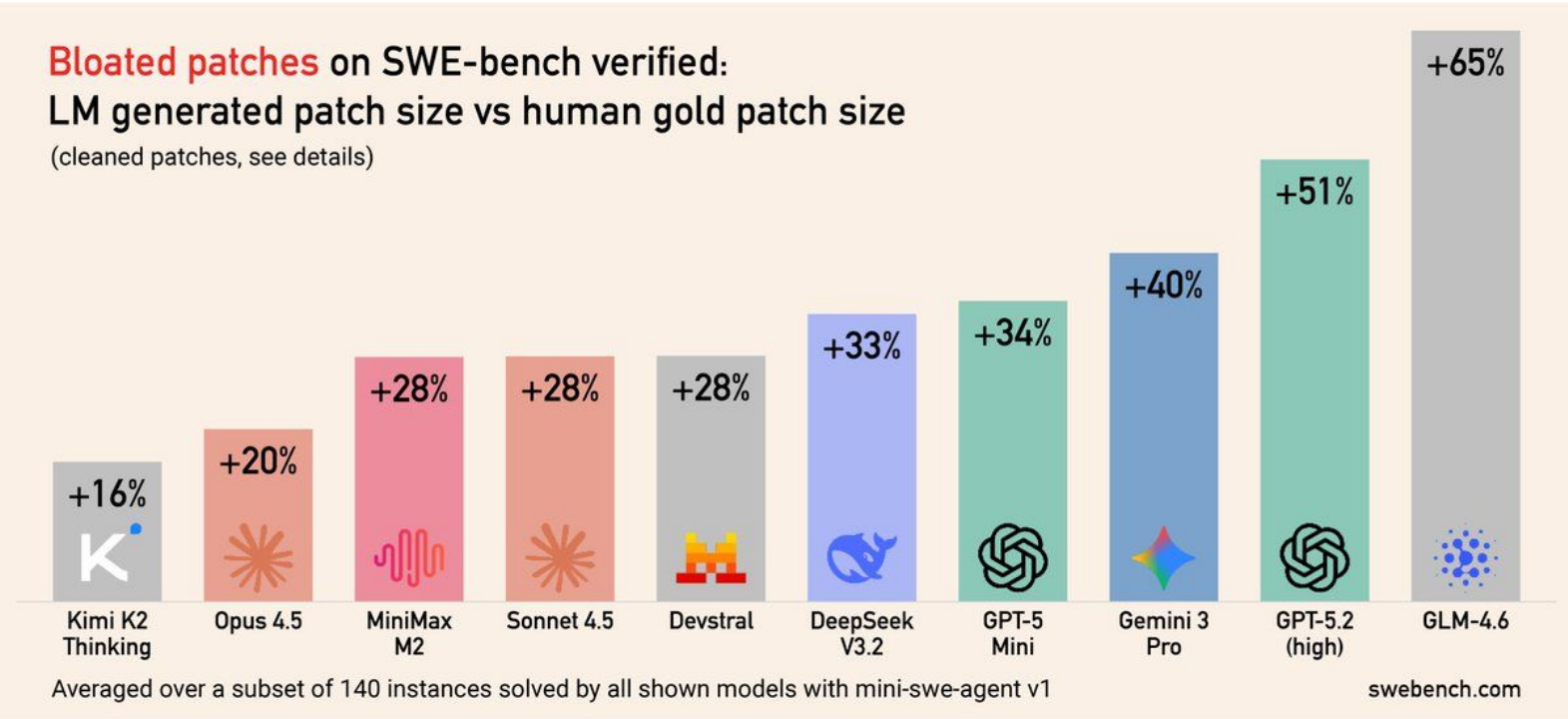
Feb 23, 2026

"Why SWE-bench Verified no longer measures frontier coding capabilities".

We’re Saturating SWE-Bench



Issues in Current Agents



https://x.com/KLieret/status/2029219763423986030?s=20

Issues in Current Agents

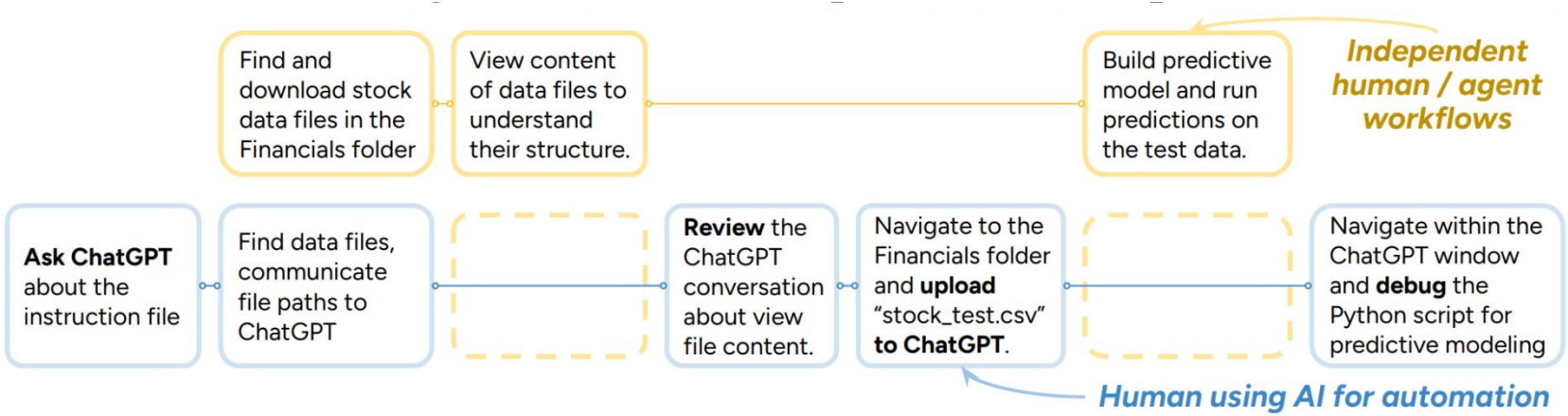
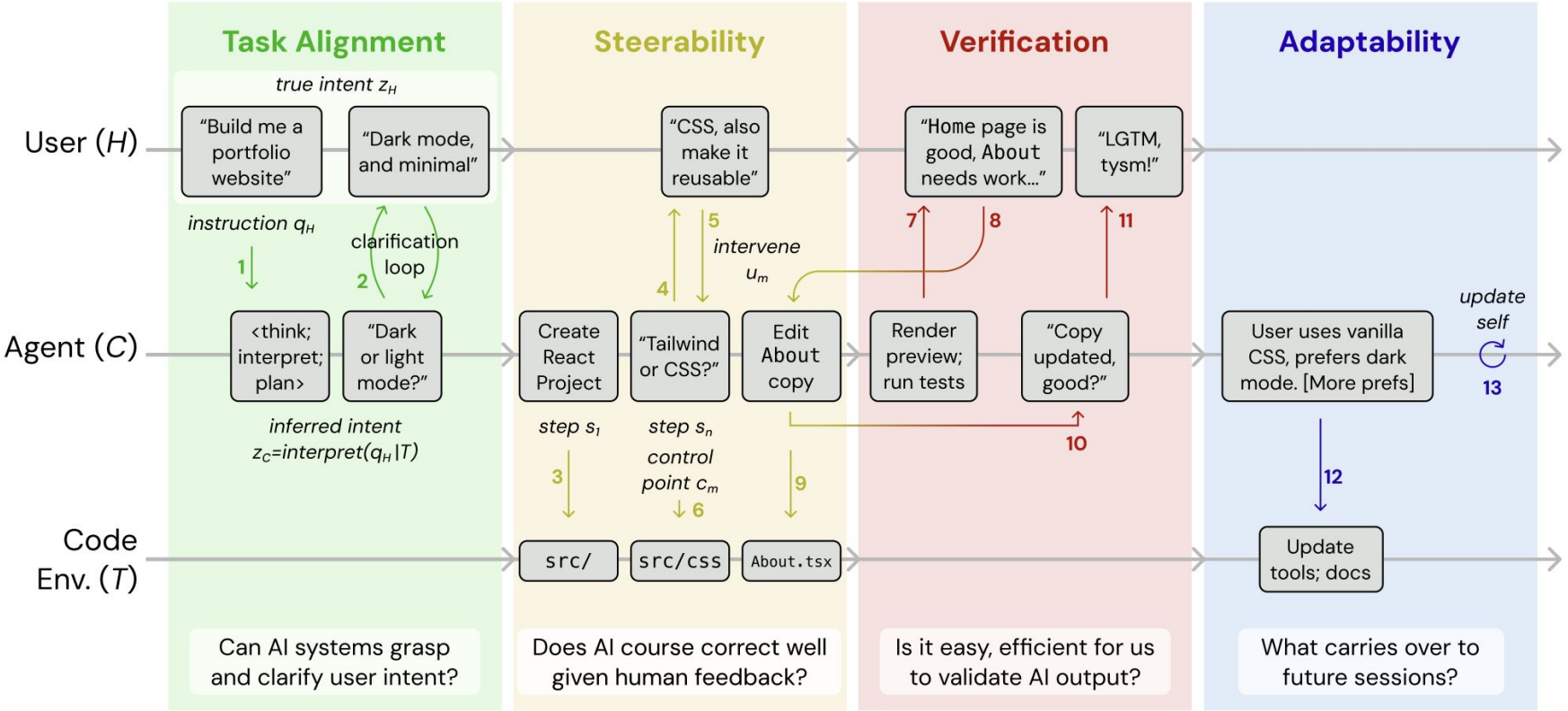


Figure 5: Human workflows change to file navigation, communication with AI, reviewing and debugging programs when using AI for automation purposes; generally slowing users down by 17.7%, as opposed to the 24.3% work acceleration when using AI for augmentation.

including error propagation, unpredictable and unproductive agent loop behavior, and the need for clear communication to mitigate the layered transparency issues. Early adopters' perspectives about the role of transparency underscored its importance as a way to build

Zora Wang, et al., "How Do AI Agents Do Human Work? Comparing AI and Human Workflows Across Diverse Occupations". 2025.

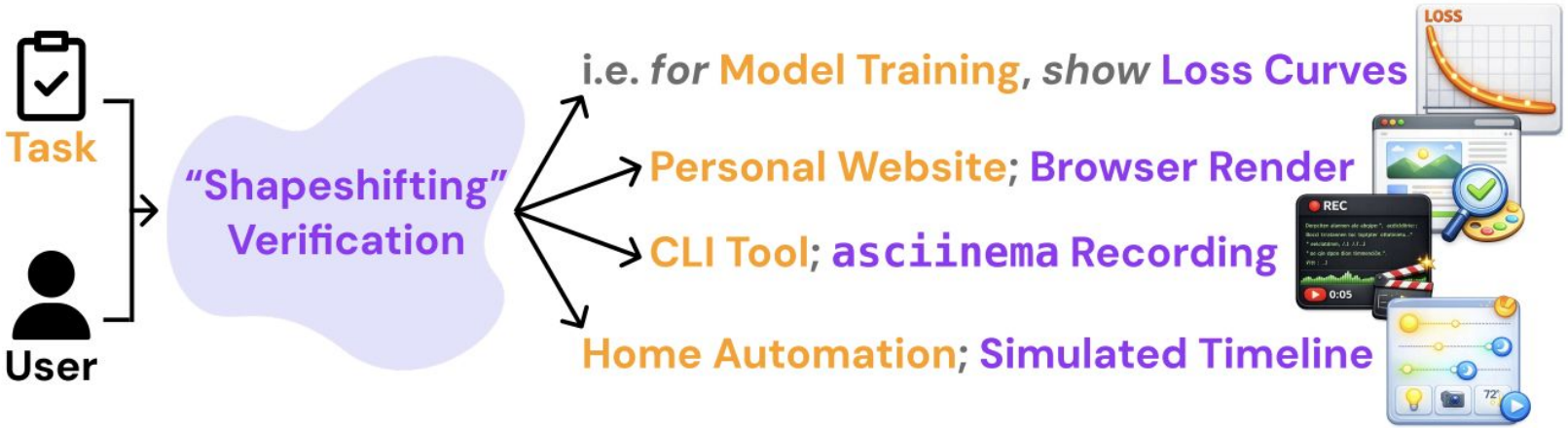
Next: Building Human-Centered Coding Agents



Zora Wang, John Yang, Kilian Lieret, et al., "Position: Humans are Missing from AI Coding Agent Research".

How to Close the Gap?

- Scale human modeling
- Enable efficient oversight
- Define measures for interaction
- Go beyond software engineering



Beyond Software Engineering

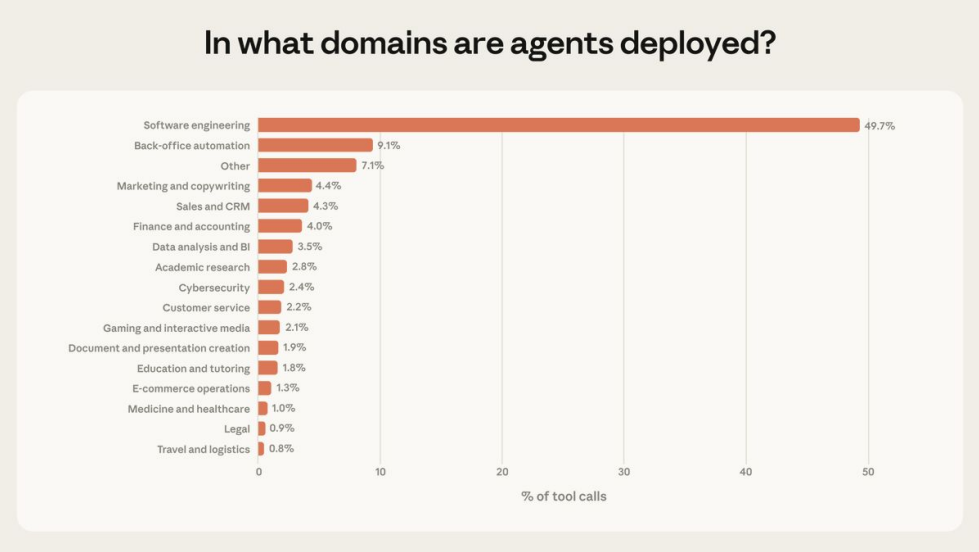
Coding agents have the potential to be

“Generalist Agents”

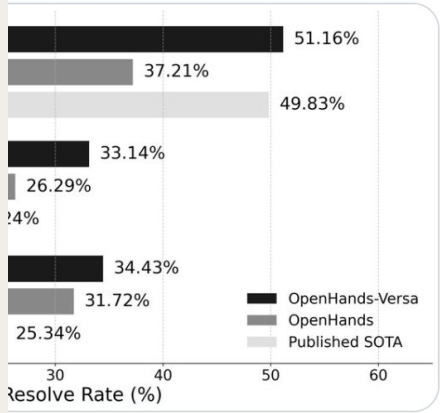
Can we design AI Agents that achieve generalizability across diverse task domains?

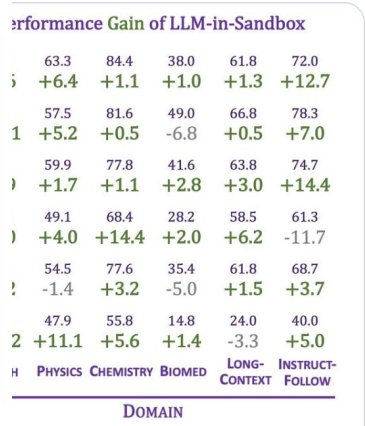
New preprint: "LLM-in-Sandbox Elicits General Agentic Intelligence"

andbox unlocks emergent



-Versa, a generalist agent with
g agent benchmarks, ranking #1
ent Company leaderboards 🚀



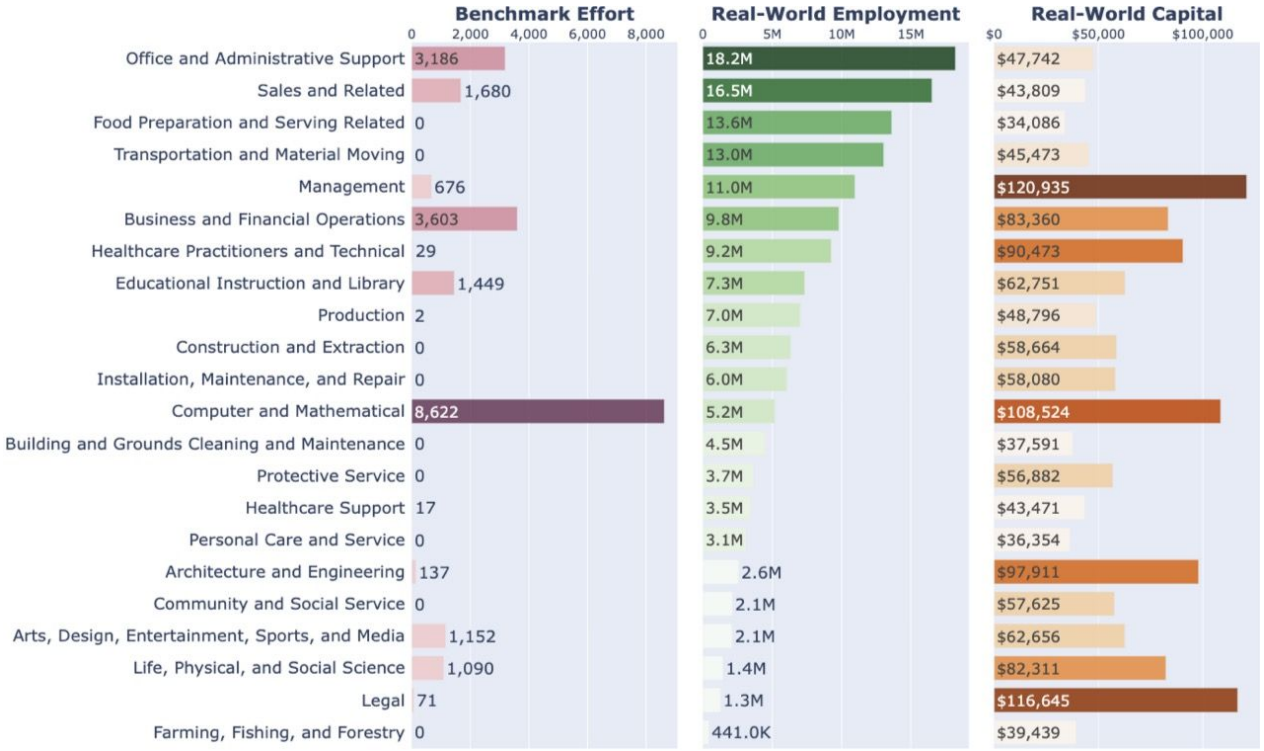


Beyond Software Engineering Agents

SE only captures 5% human employment!

Many other domains

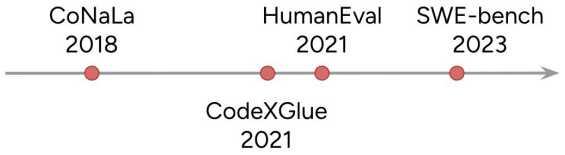
- Rarely studied
- Huge human employment
- Huge capital centralized to



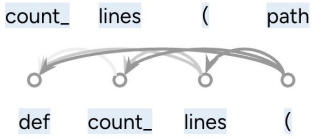
Zora Wang, et al., "How Well Does AI Agent Development Reflect Real-World Work?"

Summary

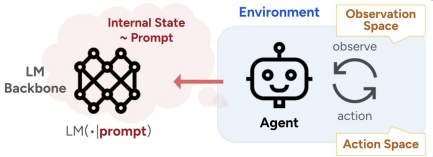
The Code Generation Task



Training Modern Coding Models



Building Agentic Coding Systems



Human-Centered