FormulaCode: Evaluating Agentic Superoptimization on Large Codebases

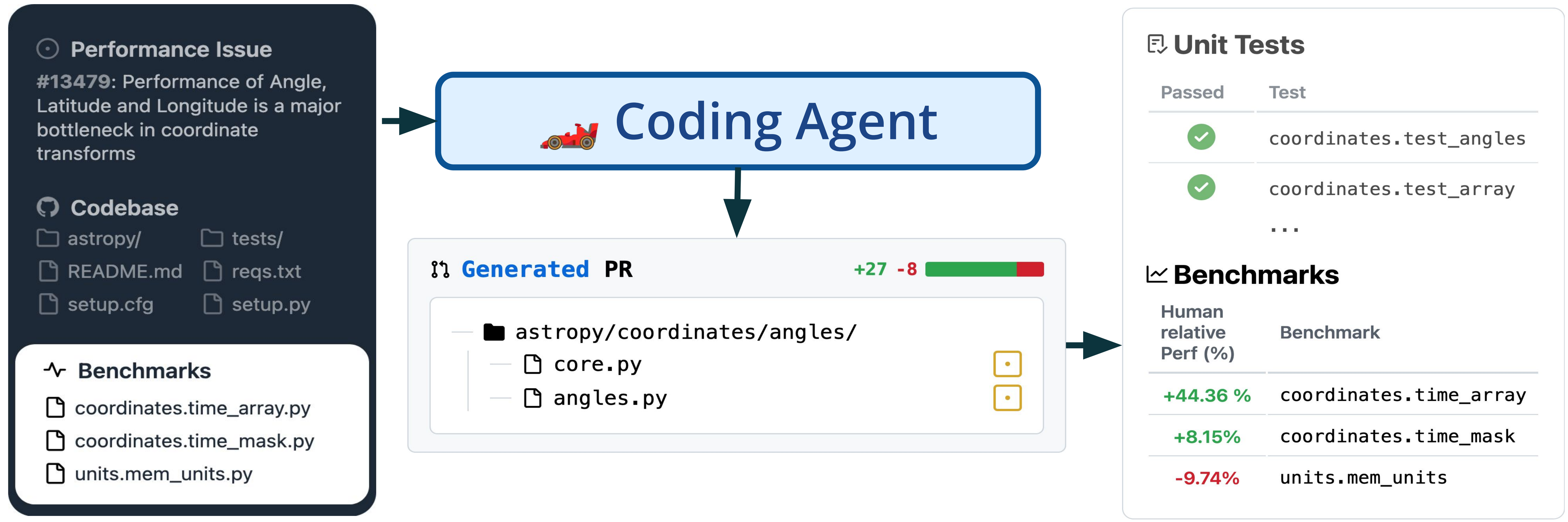
Atharva Sehgal[1*], **James Hou[3*]**, Swarat Chaudhuri[1], Jennifer J. Sun[2], Yisong Yue[3]
[1]UT Austin [2]Cornell University [3]Caltech, *Equal Contribution

https://github.com/formula-code

Problem

Can coding agents optimize software performance as well as humans can?

Motivation



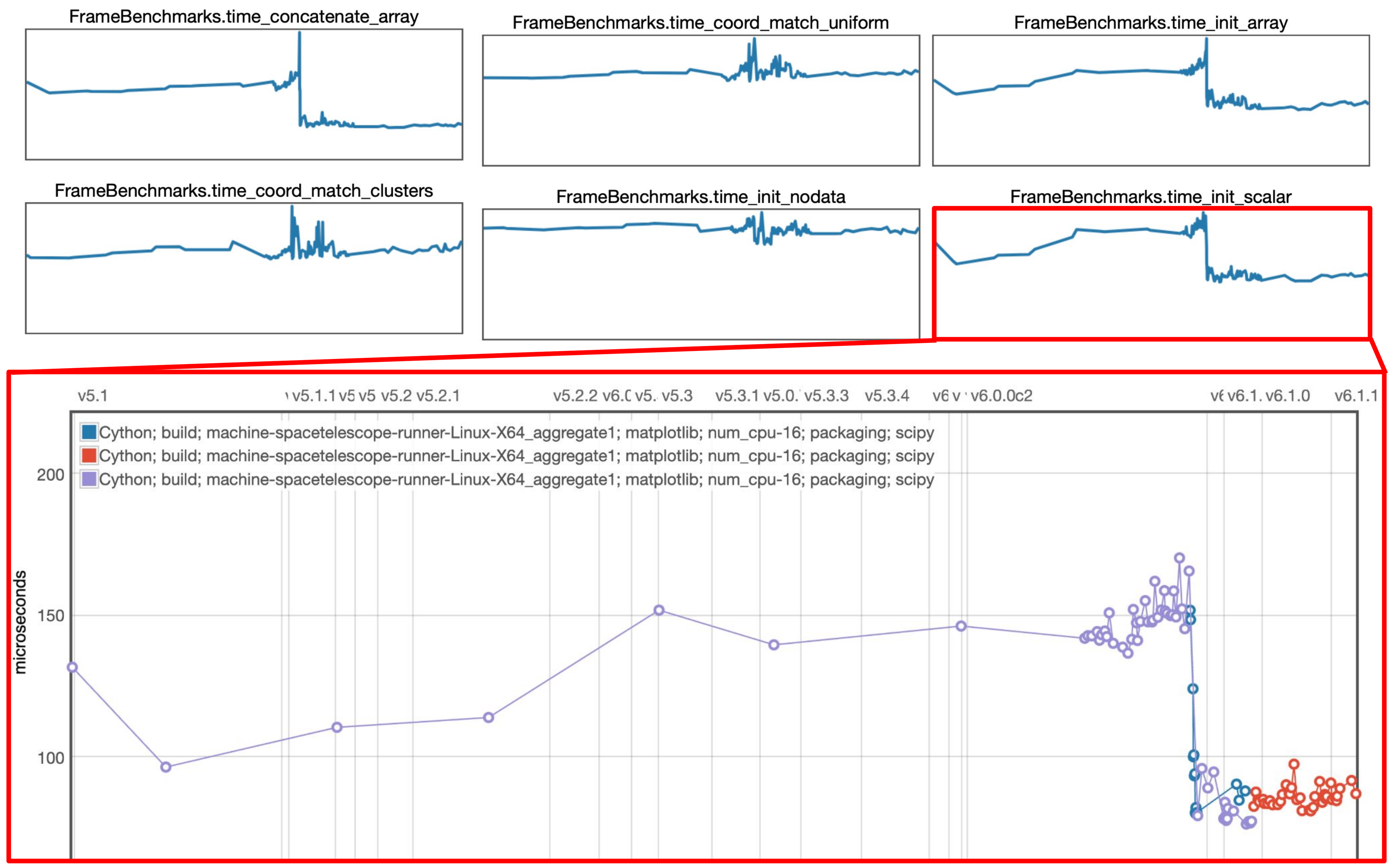
FormulaCode is a continuously updating benchmark that complements SWE-Bench in evaluating optimization agents (like AlphaEvolve)

Benchmark	# Tasks	Data Source	Evaluation Modality	Search space	Synthesis Scope	Live Updates	Data Leakage Helps?
FormulaCode	440^{++}	**Github**	**Performance Benchmarks**	**Large**	**Repo**	**Yes**	**No, human relative perf.**
SWE-Bench	2292	Github	Unit Tests	Small	Repo	No	Yes, hidden test set needed.
LiveCodeBench	300^{++}	Competitive Programming			File	Yes	Yes, continual updating needed.
CruxEval	800^{++}	Autogenerated			File	No	No, synthetic tasks

Current coding benchmarks present an incomplete picture of coding performance.

Benchmark Construction

Sample human improvement on asv benchmark



FormulaCode composition

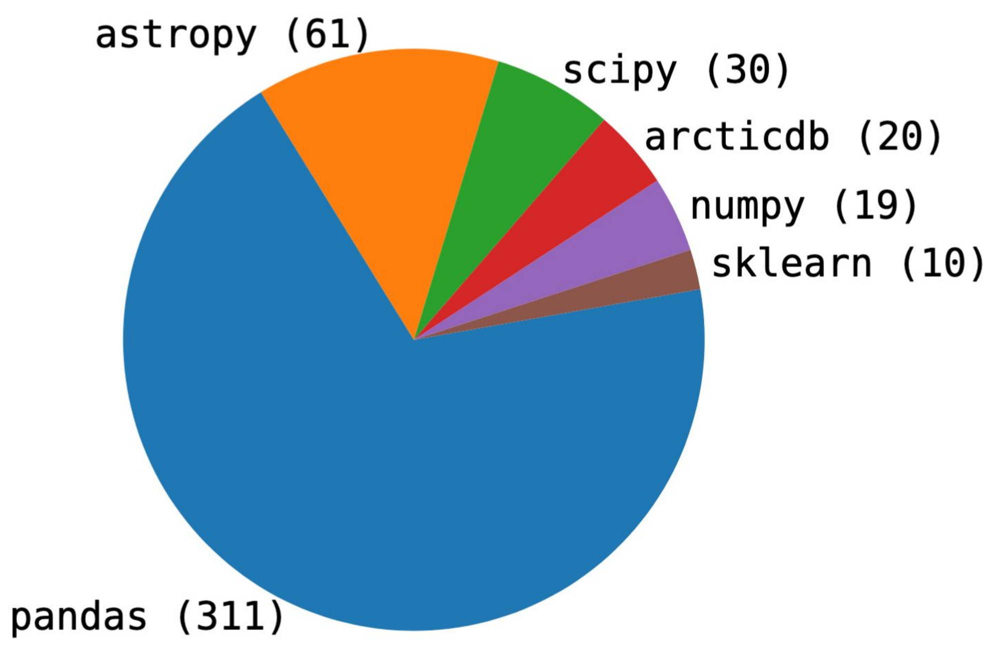


Figure 2: Distribution of FormulaCode tasks across five open source GitHub repositories. These repositories have a combined 157,000+ GitHub stars and 200,000+ academic citations and each repository uses *Airspeed Velocity* for regression testing. We collect 451 filtered tasks for our preliminary dataset consisting of 500,000+ measurements.

Benchmark	Human	GPT-4o	Sonnet 3.7	OpenEvolve	Composition
objective benchmark	46.91	59.39	-0.61	44.36	**70.91**
coordinates.FrameBenchmarks	16.60	18.61	9.80	8.15	**23.75**
coordinates.RepresentationBenchmarks	**17.71**	17.63	23.96	3.88	9.04
coordinates.SkyCoordBenchmarks	**21.28**	13.37	3.40	13.95	16.05
coordinates (core)	2.92	**22.91**	-5.75	9.74	-4.51
imports	-0.25	0.00	**0.25**	-0.25	**0.25**
Mean Improvement Percentage	**16.77**	15.88	5.24	10.72	14.71

Benchmark Suite	# Instances	GPT-4o		Sonnet 3.7		GPT-4o Oracle		Sonnet 3.7 Oracle	
		Δ%	#Valid	Δ%	#Valid	Δ%	#Valid	Δ%	#Valid
coordinates	15	-32.11	8	**8.91**	12	-36.68	11	5.18	12
imports	10	-9.26	5	13.87	5	2.18	4	**14.94**	3
io_ascii	7	0.13	2	-23.96	3	-4.37	4	**15.01**	4
io_fits	3	-2.37	1	-56.86	1	**21.18**	1	–	0
modeling	7	-3.08	3	**18.15**	6	-20.58	5	0.68	4
stats	2	-10.20	2	-2.21	2	-1.29	1	**-1.09**	2
table	7	3.53	3	**23.58**	5	-5.31	3	-1.98	6
units	10	-11.87	6	**13.61**	8	-10.54	5	-4.46	8
Overall	61	-13.19	30	**9.02**	**42**	-16.58	34	3.08	39

Agent / Model FormulaCode Evaluations

Takeaways

- LLMs can beat humans on targeted eval, but real-world optimization is multi-objective—local gains often harm global performance (low MIP).
- Human baselines help anchor evaluations and reduce data leakage.
- Benchmark functions provide dense, informative reward signals for learning agents.
- Human and agent patches target different areas—combining them can amplify gains.