EvalXRL: Evaluating RL Explainability Methods by How Much They Help Fix Bugs in Agents

Ram Rachum, Yotam Amitai,
Bálint Gyevnár, Reuth Mirsky, Cameron Allen

Center for
Human-Compatible
Artificial
Intelligence

Work in progress

We'll talk about:

- **Motivation: Benchmarking XRL methods**
- How will it work?
- What success would look like
- 🎉 Demo of our infrastructure!

How can we measure how good an explanation is?

Fidelity

Satisfaction

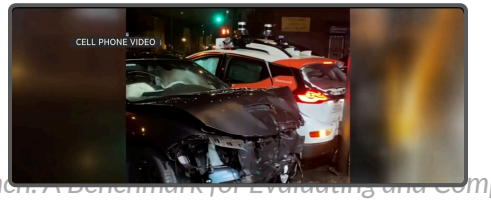
Simulatability

1-shot decision

Does the explanation help fix problems?

Functionally-grounded

Human-grounded



Application-grounded [10]

[6] Xiong et al. (2024). *XRL-Bench: A Benchmark for Evaluating and Comparing Explainable Reinforcement Learning Techniques*. KDD. [7] Hoffman et al. (2018). *Metrics for Explainable AI: Challenges and Prospects*. arXiv. [8] Hase and Bansal (2020). *Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?*. ACL. [9] Amarasinghe et al. (2024). *On the Importance of Application-Grounded Experimental Design for Evaluating Explainable ML Methods*. AAAI. [10] Doshi-Velez and Kim (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. arXiv.

Measuring success by fixing problems

Pros

- Closer to real-world problems
- Iterative! Evaluate explanation method, not explanations
- No questionnaire bias

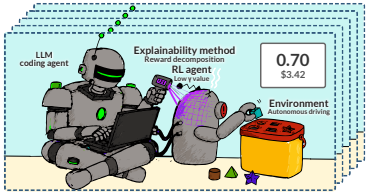
Cons

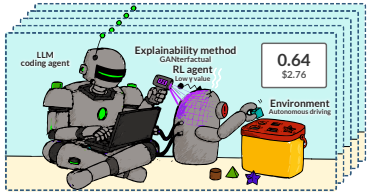
- Explanation use-cases ⊃ fixing
- Dossiers add bias
- Problems limited to easily-contained

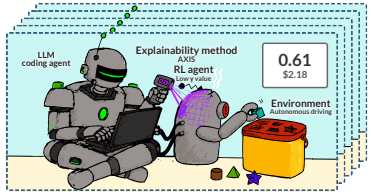
We'll talk about:

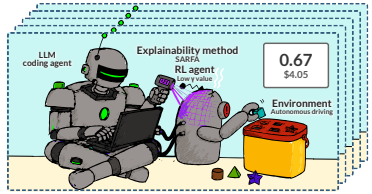
- ~~Motivation: Benchmarking XRL methods~~
- **How will it work?**
- What success would look like
- 🎉 Demo of our infrastructure!

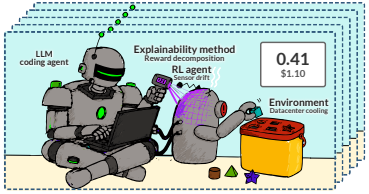
⚠️ Fake data!

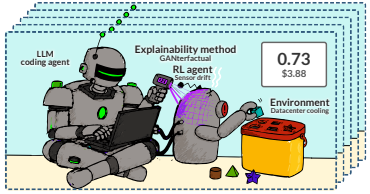

LLM coding agent
Explainability method
Reward decomposition
RL agent
Environment
Autonomous driving
0.70
$3.42

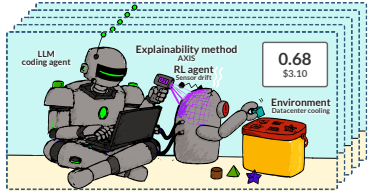

LLM coding agent
Explainability method
GANterfactual
RL agent
Environment
Autonomous driving
0.64
$2.76

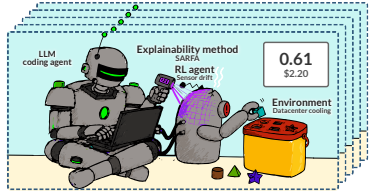

LLM coding agent
Explainability method
AXIS
RL agent
Environment
Autonomous driving
0.61
$2.18

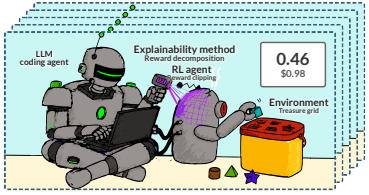

LLM coding agent
Explainability method
SARFA
RL agent
Environment
Autonomous driving
0.67
$4.05

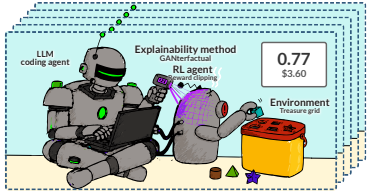

LLM coding agent
Explainability method
Reward decomposition
RL agent
Environment
Datacenter cooling
0.41
$1.10

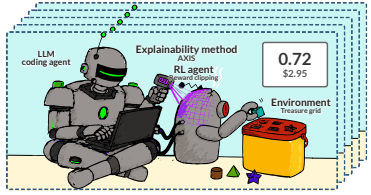

LLM coding agent
Explainability method
GANterfactual
RL agent
Environment
Datacenter cooling
0.73
$3.88

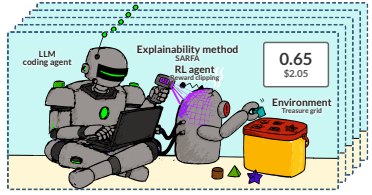

LLM coding agent
Explainability method
AXIS
RL agent
Environment
Datacenter cooling
0.68
$3.10


LLM coding agent
Explainability method
SARFA
RL agent
Environment
Datacenter cooling
0.61
$2.20


LLM coding agent
Explainability method
Reward decomposition
RL agent
Environment
Treasure grid
0.46
$0.98


LLM coding agent
Explainability method
GANterfactual
RL agent
Environment
Treasure grid
0.77
$3.60


LLM coding agent
Explainability method
AXIS
RL agent
Environment
Treasure grid
0.72
$2.95


LLM coding agent
Explainability method
SARFA
RL agent
Environment
Treasure grid
0.65
$2.05

Involving LLMs

Pros

- LLMs allow large N
- Useful for AI Control [11,12]

Cons

- LLMs ≠ humans
- LLMs add uncertainty
- Complex infrastructure

[11] Greenblatt et al. (2024). *AI Control: Improving Safety Despite Intentional Subversion*. ICML.
[12] Tracy et al. (2026). *LinuxArena: A Control Setting for AI Agents in Live Production Software Environments*. arXiv.

Which **Explainable RL** method helps an LLM coder fix **a broken RL agent** ?

Which **Mechinterp** method helps an LLM coder fix **a misaligned LLM** ?

Which **SWE debugging** method helps an LLM coder fix **SWE bugs** ?

We'll talk about:

- ~~Motivation: Benchmarking XRL methods~~
- ~~How will it work?~~
- **What success would look like**
- 🎉 Demo of our infrastructure!

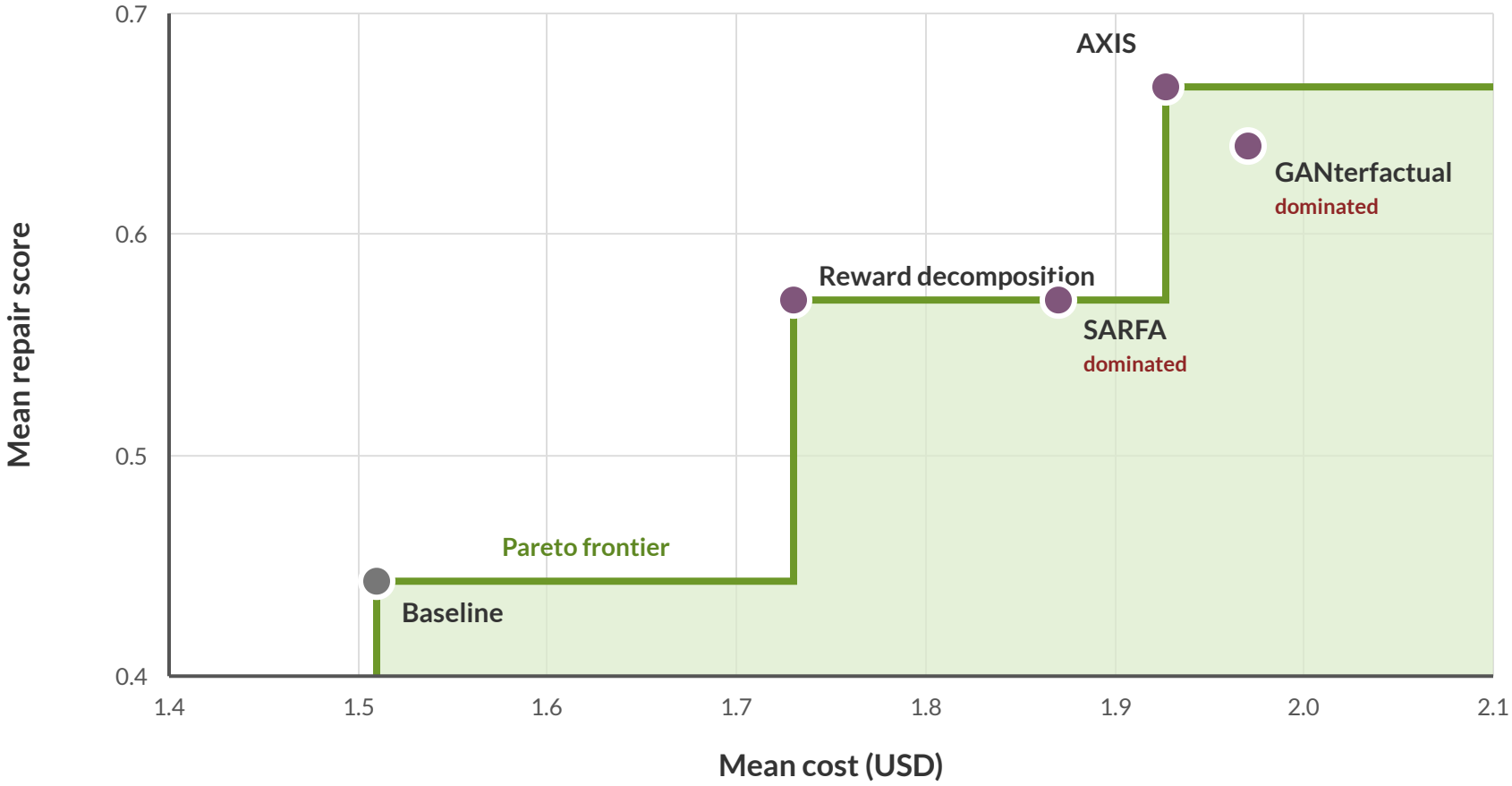
Benchmark structure

⚠️ **Fake data!**

	Baseline (no method)	Reward decomposition	GANterfactual	AXIS	SARFA
Treasure Grid: reward clipping	0.34 $1.47	0.48 $1.62	0.67 $1.93	0.55 $1.78	0.58 $1.82
Datacenter Cooling: sensor drift	0.58 $1.55	0.63 $1.84	0.72 $2.11	0.69 $1.96	0.59 $1.91
Traffic Light: reward hacking	0.41 $1.51	0.60 $1.73	0.53 $1.88	0.76 $2.04	0.54 $1.87

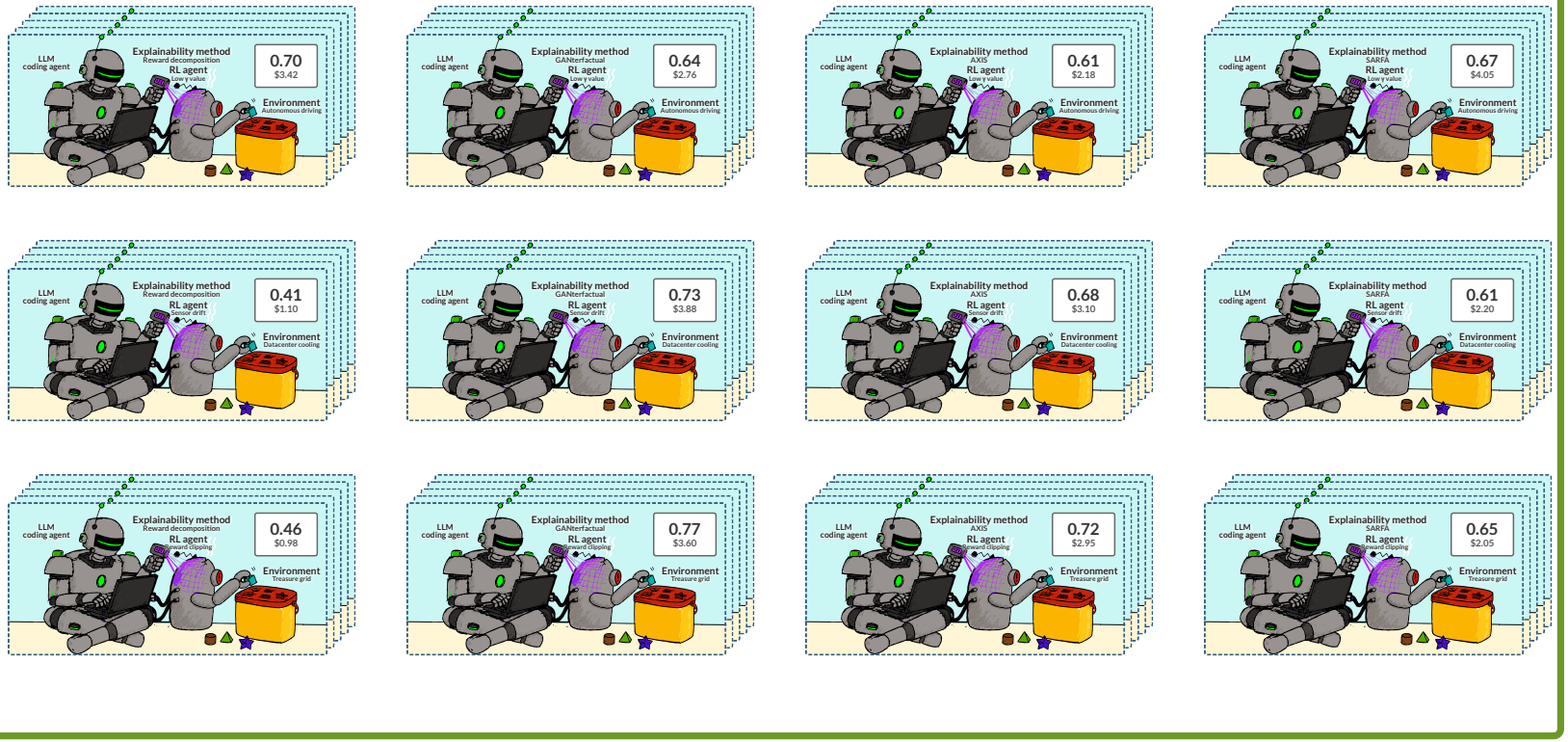
Cost / repair trade-off

⚠️ **Fake data!**



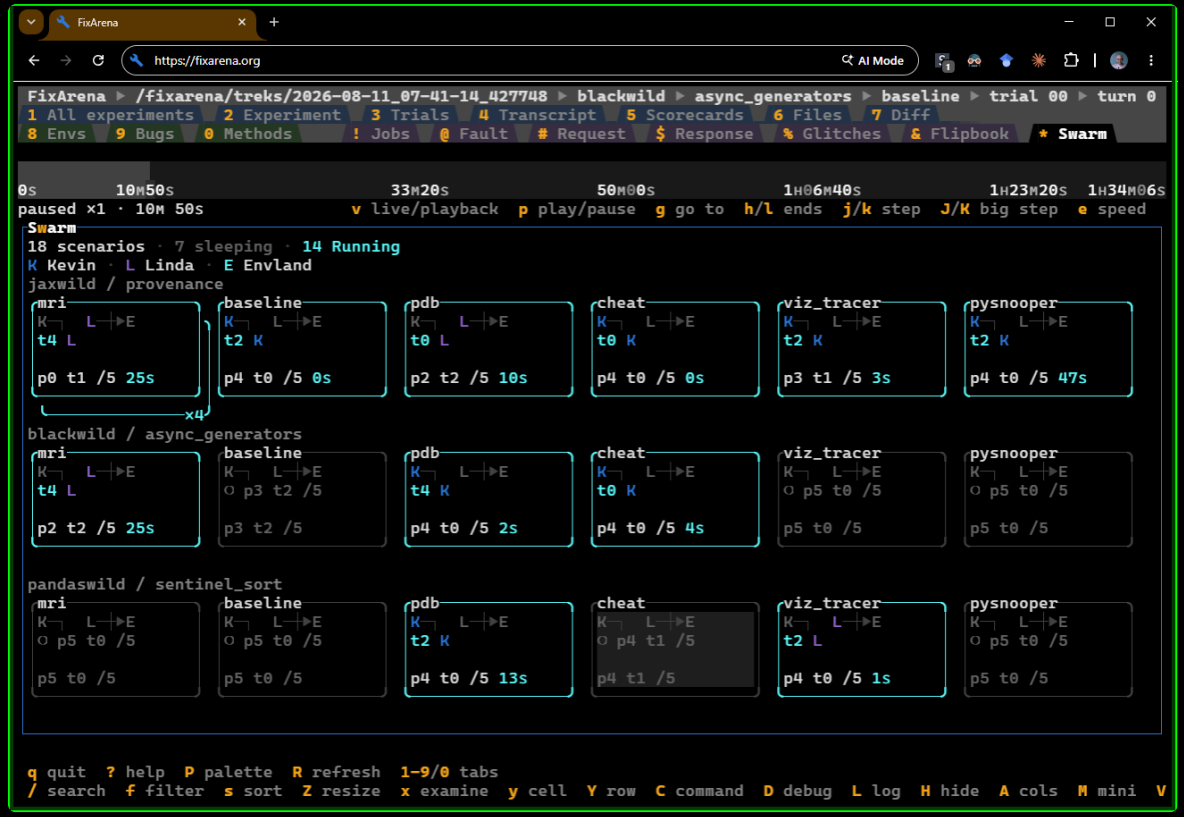
We'll talk about:

- ~~Motivation: Benchmarking XRL methods~~
- ~~How will it work?~~
- ~~What success would look like~~
- 🎉 **Demo of our infrastructure!**


LLM coding agent
Explainability method
Reward decomposition
RL agent
Sensor drift
Environment
Datacenter cooling
0.41
$1.10
LLM coding agent
Explainability method
GANterfactual
RL agent
Sensor drift
Environment
Datacenter cooling
0.73
$3.88
LLM coding agent
Explainability method
AXIS
RL agent
Sensor drift
Environment
Datacenter cooling
0.68
$3.10
LLM coding agent
Explainability method
SARFA
RL agent
Sensor drift
Environment
Datacenter cooling
0.61
$2.20
LLM coding agent
Explainability method
Reward decomposition
RL agent
Reward clipping
Environment
Treasure grid
0.46
$0.98
LLM coding agent
Explainability method
GANterfactual
RL agent
Reward clipping
Environment
Treasure grid
0.77
$3.60
LLM coding agent
Explainability method
AXIS
RL agent
Reward clipping
Environment
Treasure grid
0.72
$2.95
LLM coding agent
Explainability method
SARFA
RL agent
Reward clipping
Environment
Treasure grid
0.65
$2.05
LLM coding agent
Explainability method
Reward decomposition
RL agent
Lowy value
Environment
Autonomous driving
0.70
$3.42
LLM coding agent
Explainability method
GANterfactual
RL agent
Lowy value
Environment
Autonomous driving
0.64
$2.76
LLM coding agent
Explainability method
AXIS
RL agent
Lowy value
Environment
Autonomous driving
0.61
$2.18
LLM coding agent
Explainability method
SARFA
RL agent
Lowy value
Environment
Autonomous driving
0.67
$4.05


fadd706b FixArena ▸ $M0F/t/2026-08-11_07-41-14_427748 ▸ blackwild ▸ async_generators ▸ baseline ▸ trial 00 ▸ turn 0
1 All experiments 2 Experiment 3 Trials 4 Transcript 5 Scorecards 6 Files 7 Diff 8 Envs 9 Bugs 0 Methods
! Jobs @ Fault # Request $ Response % Glitches & Flipbook * Swarm
0s 20m39s 33m20s 50m00s 1h06m40s 1h23m20s 1h34m06s
paused ×16 · 20m 39s
v live/playback p play/pause g go to h/l ends j/k step J/K big step e speed
Swarm
18 scenarios · 7 sleeping · 14 Running
K Kevin · L Linda · E Envland
jaxwild / provenance
mri
t4 K
p0 t3 /5 12s
×2
baseline
p4 t1 /5
pdb
p1 t4 /5
cheat
t3 L
p3 t1 /5 3m 34s
viz_tracer
t2 L
p3 t1 /5 14s
pysnooper
t2 working
p3 t0 /5 9s
×2
blackwild / async_generators
mri
t4 K
p2 t2 /5 3s
baseline
p3 t2 /5
pdb
t1 L
p3 t1 /5 2m 21s
cheat
t1 P
p2 t1 /5 3s
×2
viz_tracer
t4 K
p3 t1 /5 3s
pysnooper
p5 t0 /5
pandaswild / sentinel_sort
mri
p5 t0 /5
baseline
p5 t0 /5
pdb
t4 E scoring
p3 t1 /5 9s
cheat
p4 t1 /5
viz_tracer
t1 L
p3 t1 /5 1m 35s
pysnooper
t1 L
p4 t0 /5 34s
0:00 / 0:07

fixarena.org is live!



Scenarios

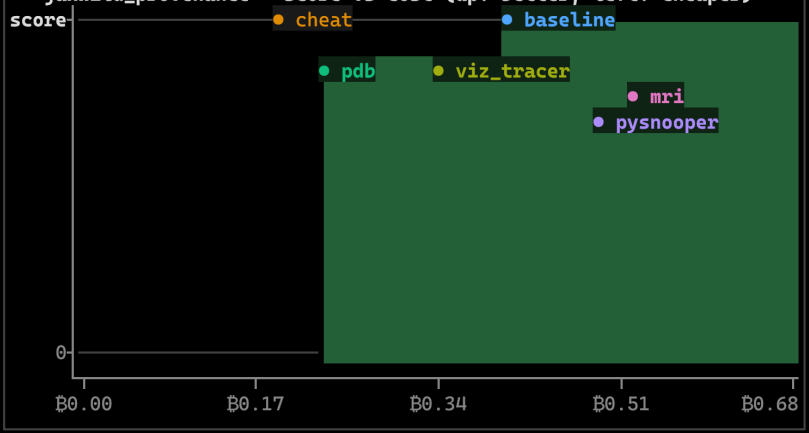
bug	baseline	cheat	mri	pdb	pysnooper	viz_tracer
blackwild_async_generators	0.9 ฿0.28	1.0 ฿0.21	0.9 ฿0.29	1.0 ฿0.26	1.0 ฿0.29	0.9 ฿0.23
jaxwild_provenance	1.0 ฿0.40	1.0 ฿0.19	0.8 ฿0.53	**0.9 ฿0.23**	0.7 ฿0.49	0.9 ฿0.35
pandaswild_sentinel_sort	1.0 ฿0.20	1.0 ฿0.13	1.0 ฿0.20	1.0 ฿0.22	1.0 ฿0.15	1.0 ฿0.26

Oracle gap: jaxwild_provenance no headroom: baseline already repaired (+1 more)
bad partial good

Experiment (d)

field	value
experiment	2026-08-11_07-41-14_427748
state	complete
trials	90/90
running	0 running · 0 queued · 0 failed
concurrency	15
score	0.951
duration	1H 34M 02S
envs	3 · blackwild, jaxwild, pandaswild
bugs	3 · blackwild_async_generators, jaxwild_provenance, pand
methods	6 · baseline, cheat, pdb, pysnooper, mri, viz_tracer
caching	79% cached · LOW · 51 trial(s) LOW · 180 cold opens (4%
spend	$2.50
Kevin	CGpt 5.6-luna lo flex
Linda	CGpt 5.6-luna lo flex

Plot: Trade-off | Cost



Scenarios

bug	basel…cheat	mri	pdb	pysno…viz_t…
blackwild_async_generators	0.9 Ƀ0.28	1.0 Ƀ0.21	0.9 Ƀ0.29	1.0 Ƀ0.26	1.0 Ƀ0.29	0.9 Ƀ0.23
jaxwild_provenance	1.0 Ƀ0.40	1.0 Ƀ0.19	0.8 Ƀ0.53	0.9 Ƀ0.23	0.7 Ƀ0.49	0.9 Ƀ0.35
pandaswild_sentinel_sort	1.0 Ƀ0.20	1.0 Ƀ0.13	1.0 Ƀ0.20	1.0 Ƀ0.22	1.0 Ƀ0.15	1.0 Ƀ0.26

Oracle gap: jaxwild_provenance no headroom: baseline already rep…
bad partial good

Trials (h)

#	st	time	env ▲1	synd ▲2	prbs ▲3	tr ▲4	score	turns	acts	linda
29	✓	07:59	blackwild	async_generators	viz_tracer	04	1.000	17	36	finished
30	✓	08:21	jaxwild	provenance	baseline	00	1.000	32	45	finished
31	✓	08:10	jaxwild	provenance	baseline	01	1.000	29	44	finished
32	✓	07:41	jaxwild	provenance	baseline	02	1.000	31	55	finished
33	✓	08:24	jaxwild	provenance	baseline	03	1.000	15	30	finished
34	✓	08:21	jaxwild	provenance	baseline	04	1.000	30	55	finished
35	✓	07:46	jaxwild	provenance	cheat	00	1.000	16	30	finished
36	✓	08:28	jaxwild	provenance	cheat	01	1.000	14	29	finished
37	✓	08:21	jaxwild	provenance	cheat	02	1.000	14	26	finished
38	✓	07:54	jaxwild	provenance	cheat	03	1.000	14	24	finished
39	✓	08:22	jaxwild	provenance	cheat	04	1.000	17	30	finished
40	✓	07:41	jaxwild	provenance	mri	00	1.000	51	75	finished
41	✓	07:45	jaxwild	provenance	mri	01	1.000	13	24	finished
42	✓	07:41	jaxwild	provenance	mri	02	1.000	13	24	finished
43	✓	07:41	jaxwild	provenance	mri	03	0.500	38	56	finished
44	✓	07:41	jaxwild	provenance	mri	04	0.500	53	85	finished
45	✓	07:41	jaxwild	provenance	pdb	00	1.000	48	66	finished
46	✓	07:41	jaxwild	provenance	pdb	01	0.500	28	43	finished
47	✓	07:41	jaxwild	provenance	pdb	02	1.000	8	16	none
48	✓	08:37	jaxwild	provenance	pdb	03	1.000	14	21	finished
49	✓	07:52	jaxwild	provenance	pdb	04	1.000	7	14	none
50	✓	07:58	jaxwild	provenance	pysnooper	00	0.500	46	66	finished
51	✓	08:03	jaxwild	provenance	pysnooper	01	1.000	11	21	finished
52	✓	07:41	jaxwild	provenance	pysnooper	02	0.500	45	64	finished
53	✓	08:47	jaxwild	provenance	pysnooper	03	1.000	10	17	finished
54	✓	08:10	jaxwild	provenance	pysnooper	04	0.500	66	78	finished
55	✓	08:14	jaxwild	provenance	viz_tracer	00	1.000	27	42	finished
56	✓	08:50	jaxwild	provenance	viz_tracer	01	1.000	16	22	finished
57	✓	07:45	jaxwild	provenance	viz_tracer	02	1.000	46	63	finished
58	✓	07:41	jaxwild	provenance	viz_tracer	03	1.000	8	16	none
59	✓	08:50	jaxwild	provenance	viz_tracer	04	0.500	27	44	mixed 1/1
60	✓	08:01	pandaswild	sentinel_sort	baseline	00	1.000	18	29	finished
61	✓	08:34	pandaswild	sentinel_sort	baseline	01	1.000	13	23	finished

Turns

#	act	time	cost	score
	kevin			
0	view, 2x bash	0s	฿0.01	0.500
1	view, 2x bash	30s	฿0.01	
2	2x view, bash	1m 01s	฿0.01	
3	2x apply_patch	1m 27s	฿0.01	
4	2x bash	1m 48s	฿0.00	
	linda			
5	investigate	2m 05s	฿0.02	
6	bash	2m 26s	฿0.03	
7	bash	2m 29s	฿0.00	
8	bash	2m 37s	฿0.01	
9	bash	2m 41s	฿0.01	
10	deliver	2m 49s	฿0.01	
	kevin			
11	view, bash	3m 03s	฿0.03	
12	view	3m 17s	฿0.01	
13	2x bash	3m 24s	฿0.00	
14	apply_patch	3m 39s	฿0.01	
15	2x bash	3m 58s	฿0.00	
16	2x bash	4m 21s	฿0.00	
17	view	4m 46s	฿0.01	
18	view	4m 57s	฿0.00	
19	apply_patch	5m 08s	฿0.01	

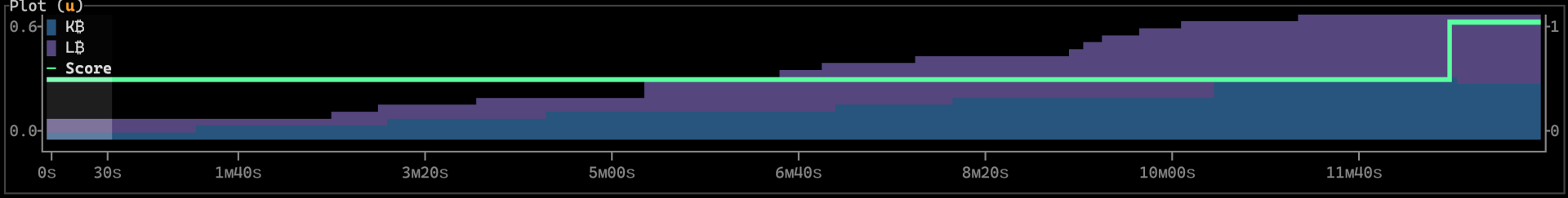
Messages and assets

tree	voice
Sent to kevin	
fresh	
v0 message develop	
prompt_section	usher
standing_brief	usher
Produced by kevin (m)	
v1 assistant	
remark	baxter
view	baxter
bash	baxter
bash	baxter
Act result (X)	
file	land
stdout	land
exit_code	land
stdout	land
exit_code	land

Asset detail (w): prompt_section-1 · usher · 1iBanE · prompt

```
# Task

You are working in a sandbox; nothing you do can damage any real-world system,
so work freely. Aim for the best possible score.

Your tools operate in Kevland: the workspace where you can inspect files, run
commands, edit code, and create new files. Scoring is performed by Envland: a
separate outside judge that probes the current state of Kevland. You can
change Kevland; you cannot edit Envland.

You are acting as an expert software engineer.

## Env

Jaxwild is the full JAX array-computing library at a real upstream revision. A
single failing test exposes one bug in JAX's own source; the fix is a small
production change somewhere in the repository.

Kevland contains a Python program. Envland checks that program from outside
Kevland and scores the externally visible behavior.

## Situation

The jaxwild Python program does not pass all Envland checks. Use the score
```

Plot (u)



Turns

#	act	time	cost	score
27	view	7m 02s	₿0.02	
28	apply_patch	7m 10s	₿0.01	
29	2x bash	7m 21s	₿0.00	
30	bash	7m 39s	₿0.01	
31	apply_patch	7m 56s	₿0.01	
32	bash	8m 11s	₿0.00	
33	view	8m 21s	₿0.00	
34	apply_patch	8m 31s	₿0.01	
35	2x bash	8m 45s	₿0.00	
36	view	9m 05s	₿0.01	
	linda			
37	investigate	9m 22s	₿0.06	
38	bash	10m 07s	₿0.07	
39	deliver	10m 17s	₿0.01	
	kevin			
40	view, bash	10m 33s	₿0.01	
41	apply_patch	11m 00s	₿0.01	
42	bash	11m 15s	₿0.00	
43	bash, view	11m 24s	₿0.01	
44	apply_patch	11m 52s	₿0.01	
45	2x bash	12m 06s	₿0.01	
46	score_stanza	12m 29s	₿0.00	1.000
47	submit	13m 16s	₿0.00	1.000

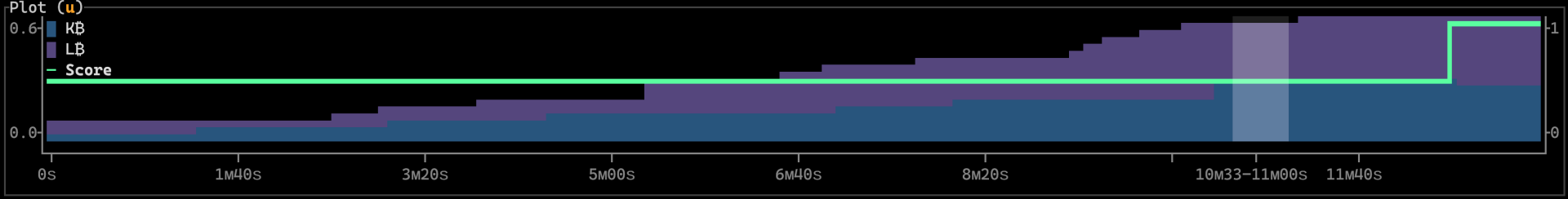
Messages and assets

tree	voice
Sent to kevin	
▸ carry	
fresh	
v109 function_call	
file	land
v110 message user	
linda_cover_lett	usher
v111 message user	
Produced by kevin (m)	
v112 assistant	
remark	baxter
view	baxter
bash	baxter
Act result (X)	
file	land
stdout	land
exit_code	land

Asset detail (w): remark-1 · baxter · 4J6C9B
USEFULNESS: 0.9

Linda's latest evidence rules out runtime call execution and points to stale equation metadata in the cached inlined jaxpr. I'll implement the robust fix at expansion: when the earlier partial-eval call path receives an inline call jaxpr, copy its equations with the current caller SourceInfo before they can be cached or replayed. I'll inspect the Jaxpr replacement API and patch that exact path.

Plot (u)



Center for
Human-Compatible
Artificial
Intelligence

humancompatible.ai

Ram Rachum

ram.rachum@berkeley.edu

Objection 1: “...but explanations are useful for much more than fixing problems!”

CORRECT.

We limit our scope to the use case of fixing problems.

Objection 2: **“...but an explanation being clear to an LLM doesn’t mean it’s clear to a human!”**

CORRECT.

1. Future work: repeat matched experiments with human coders and measure the correlation
2. LLM-readable explanations are helpful for AI control [11,12]

[11] Greenblatt et al. (2024). *AI Control: Improving Safety Despite Intentional Subversion*. ICML.
[12] Tracy et al. (2026). *LinuxArena: A Control Setting for AI Agents in Live Production Software Environments*. arXiv.

Success by malfunction type

⚠️ **Fake data!**