Collider-Bench

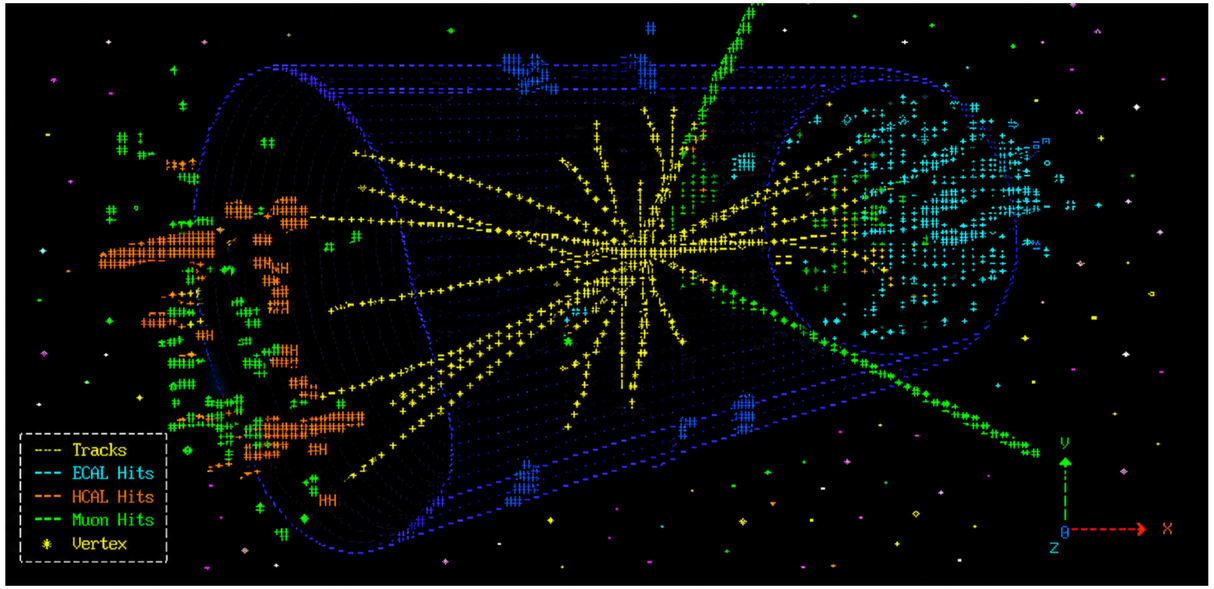
Benchmarking AI Agents with Particle Physics Analysis Reproduction

arXiv:2605.13950, git-repo

Darius A. Faroughy, Sofia Palacios Schweitzer, Ian Pang, Siddharth Mishra-Sharma, David Shih

RUTGERS UNIVERSITY

ML4Jets — Vienna, 16.09.2026

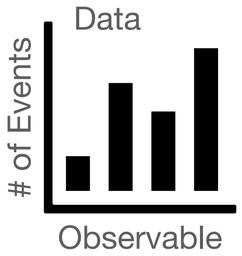


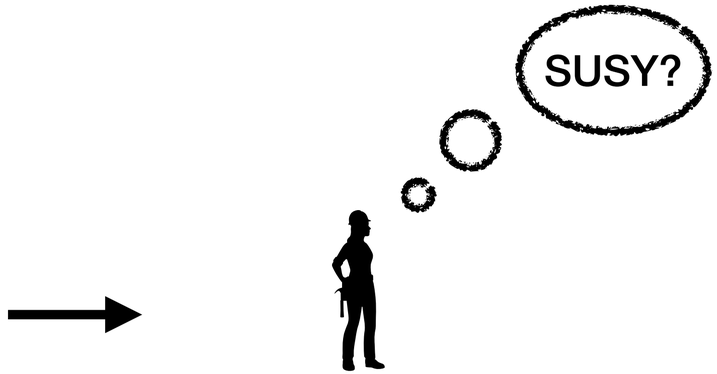
Collider-Bench

Automating *Recasting*

Automating *Recasting*

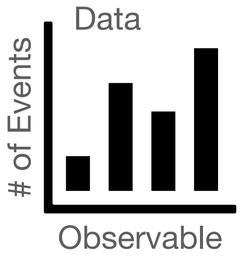
LHC:

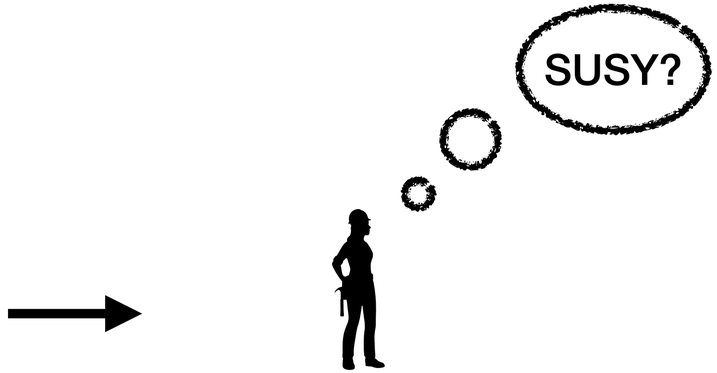




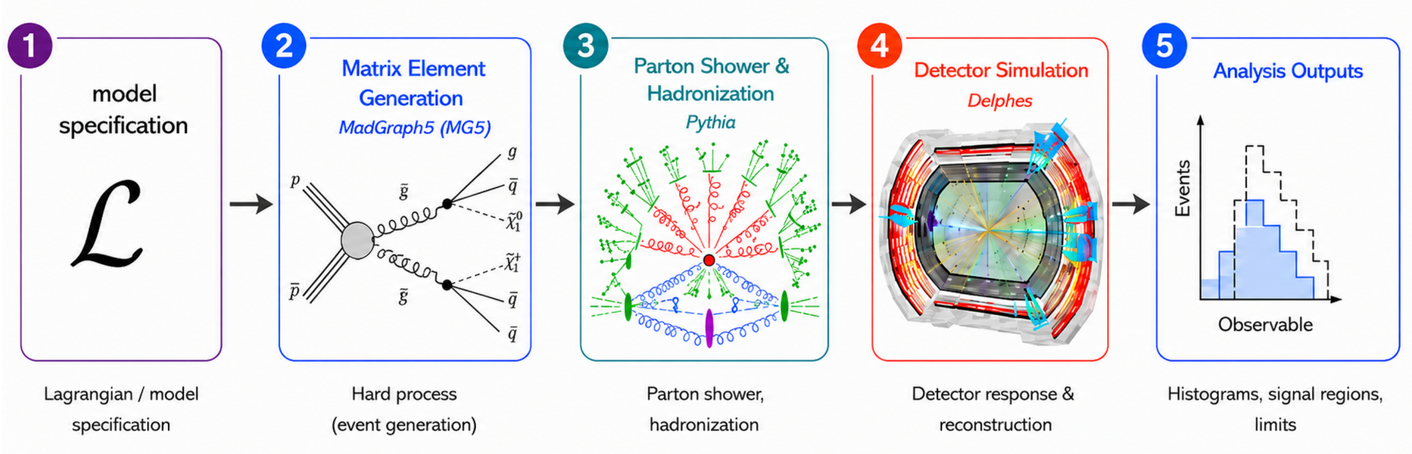
Automating *Recasting*

LHC:



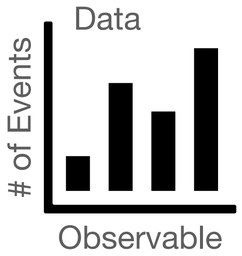


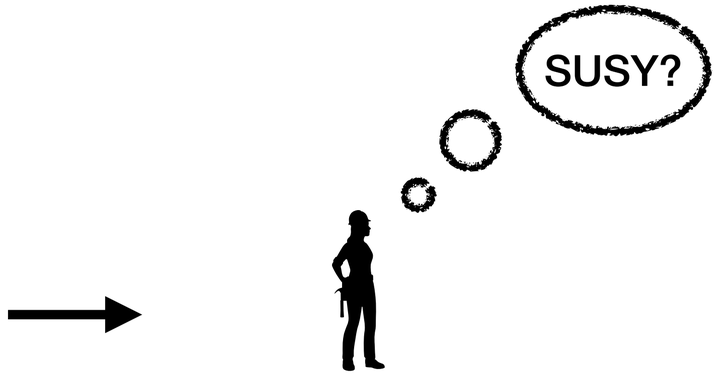
Signal + SM Background



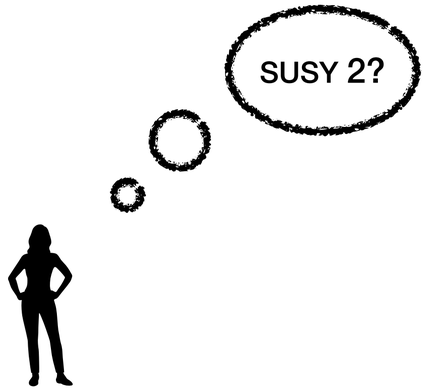
Automating *Recasting*

LHC:





Theorist:



Re-use data, background simulation & systematics

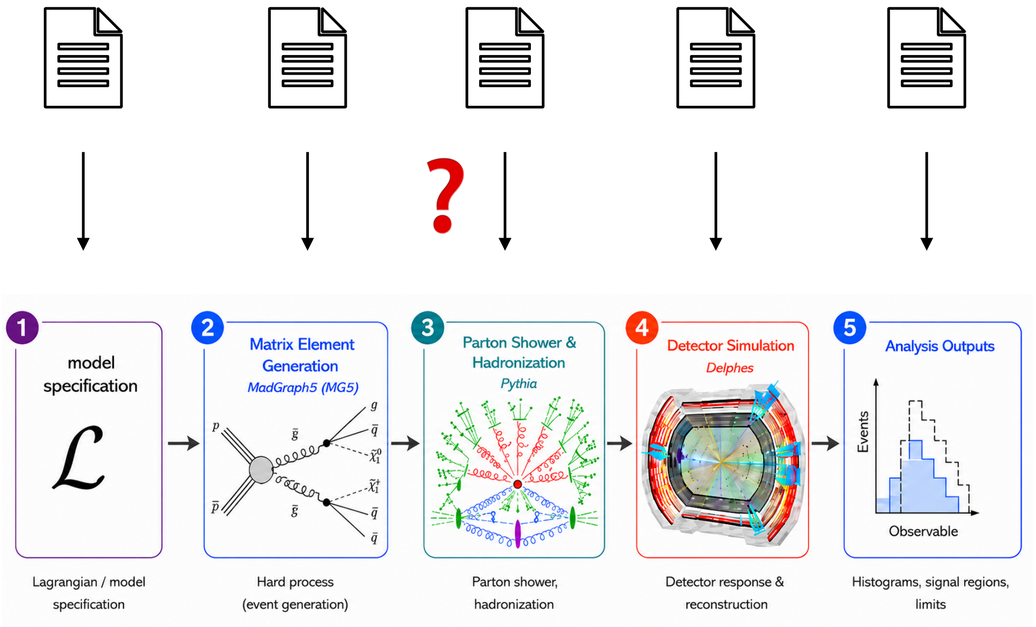
Automating *Recasting*

Why is this not trivial?

1. Running the simulation — multiple simulation packages

2. Missing information

3. Publicly available simulators only approximations



Automating *Recasting*

Recasting Pipeline

1. Reproduce public SUSY yields for a given mass point
2. Reproduce SUSY limit plots

} Validation

3. Exchange SUSY → SUSY 2

} Recast

Automating *Recasting*

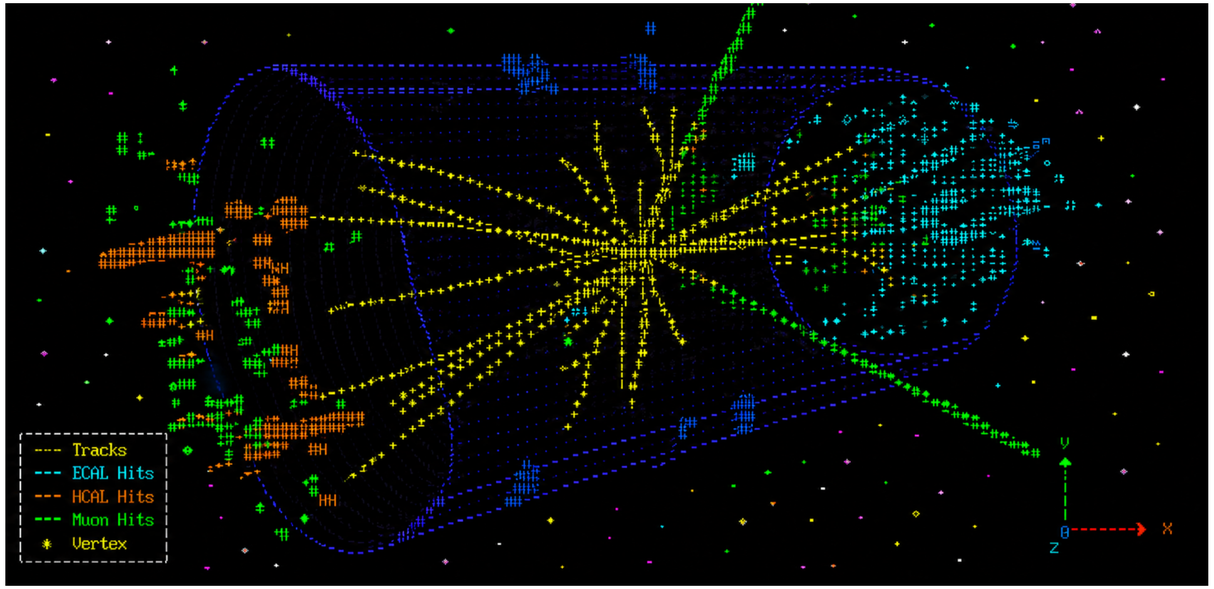
Recasting Pipeline

1. Reproduce public SUSY yields for a given mass point	Validation	Hard, but cheap
2. Reproduce SUSY limit plots		Hard, but costly
3. Exchange SUSY → SUSY 2	Recast	Easy, but costly

Automating *Recasting*

Recasting Pipeline

1. Reproduce public SUSY yields for a given mass point	Validation	Hard, but cheap
2. Reproduce SUSY limit plots		Hard, but costly
3. Exchange SUSY → SUSY 2	Recast	Easy, but costly



Collider-Bench

Benchmarking *agentic workflows*

Benchmarking *agentic workflows*

Current literature landscape of agents + hep:

- Agents of Discovery [2509.08535]
- ArgoLOOM [2510.02426]
- Automating high energy physics data analysis with llm-powered agents [2512.07785]
- HEPTAPOD [2512.15867]
- MadAgents [2601.21015]
- ColliderAgent [2603.14553]
- FERMIACC [2603.22538]
- AI Agents can Already Autonomously Perform Experimental High Energy Physics [2603.20179]

...

Demonstration that (*self-build*) autonomous agents can solve a given task

No evaluation (or validation) of *how well*

→ **Need of evaluation benchmarks for agents capturing real scientific workflows**

Benchmarking *agentic workflows*

On Recasting — Why?

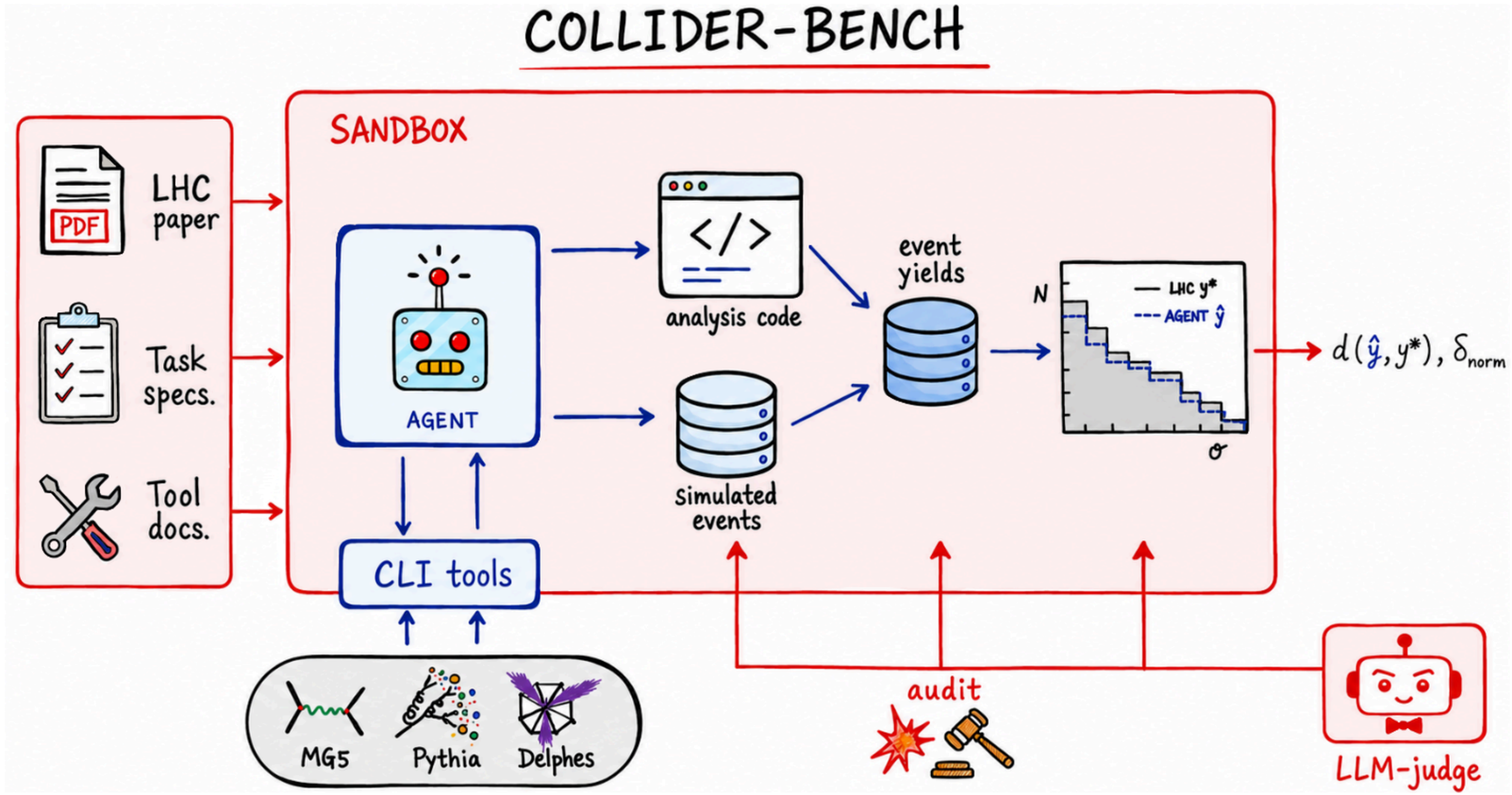
1. Multi-step pipeline (navigating different code environment, programming languages, code execution…)

2. Long-horizon task (gathering all information, planning, executing, iterating, …)

3. Reasoning capabilities (filling the gap)

Benchmarking *agentic workflows*

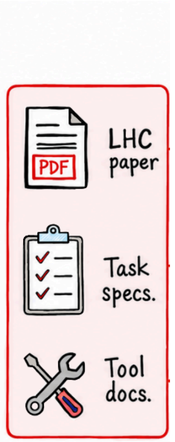
On Recasting — How?



Collider-Bench

Inputs

$$x = (\mathscr{P}, s, \mathscr{O}, \mathscr{B}, \mathscr{T})$$



LHC paper $\mathscr{P}$
Signal process s
Signal observable $\mathscr{O}$
Histogram binning $\mathscr{B}$
Toolbox $\mathscr{T}$

TASK.md

Paper: CMS-SUS-16-046
Centre-of-mass energy: 13 TeV
Luminosity: 35.9 fb^{-1}
Task type: simulation
Signal benchmark: `T5Wg_1750_1700`
Observable: S_T^γ

Task
Implement the search analysis described in **CMS-SUS-16-046** and use it to predict the binned differential signal yield in S_T^γ for the benchmark point `T5Wg_1750_1700`, in the analysis's **signal region**, normalized to 35.9 fb^{-1} at $\sqrt{s} = 13$ TeV.

The agent should:

1. Generate `T5Wg_1750_1700` events using a matrix-element generator, parton shower, and detector simulation chain of its choice.
2. Read the paper to determine the object identification, event-selection requirements, and signal-region cuts that define the analysis, then apply them to the generated events.
3. Histogram the surviving events in S_T^γ using the bin edges already present in the `results/*.yaml` template. The bin edges must not be modified.

Definitions

- S_T^γ – the scalar sum of p_T^{miss} and the transverse momenta of all photons in the event.
- `T5Wg_1750_1700` – pair-produced gluinos at $m(\tilde{g}) = 1750$ GeV, each decaying via the `T5Wg` simplified-model topology to a mass-degenerate wino NLSP at $m(\widetilde{W}) = 1700$ GeV‘ and a massless LSP.

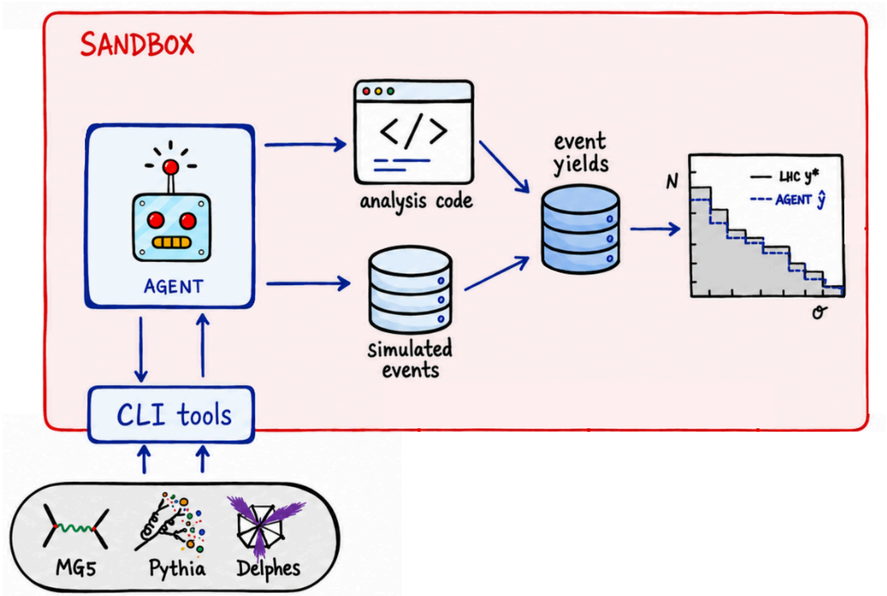
Output requirements

Artifact	Purpose
`results/*.yaml`	Fill null bin values with predicted relative signal bin contents.
`analysis/*.py`	Event-selection code, runnable on the generated sample.
`data/*.root`, `sims/*.dat`	Selected-event files and generator/detector cards.
`report.md`	Methodological choices and deviations from the paper.

Important
The goal is to predict the signal event distribution from your own simulation and analysis pipeline. Do not extract bin values from the paper's figures, tables, HEPData record, or elsewhere.

Collider-Bench

Execution & Output



Use agent of your choice
Sandbox guarantees reproducibility

Final output : yield vector $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_K)$

Collider-Bench

Qualitative evaluation

Audit agent’s log file, analysis codes and report

Check:
- Did it run through?
- Did the agent actually ran the simulation chain?
- Did the agent use its own simulation to create the yield vector?
- …

→ Writes report, human decides 👍🏽 👎🏽

Collider-Bench

Quantitive evaluation

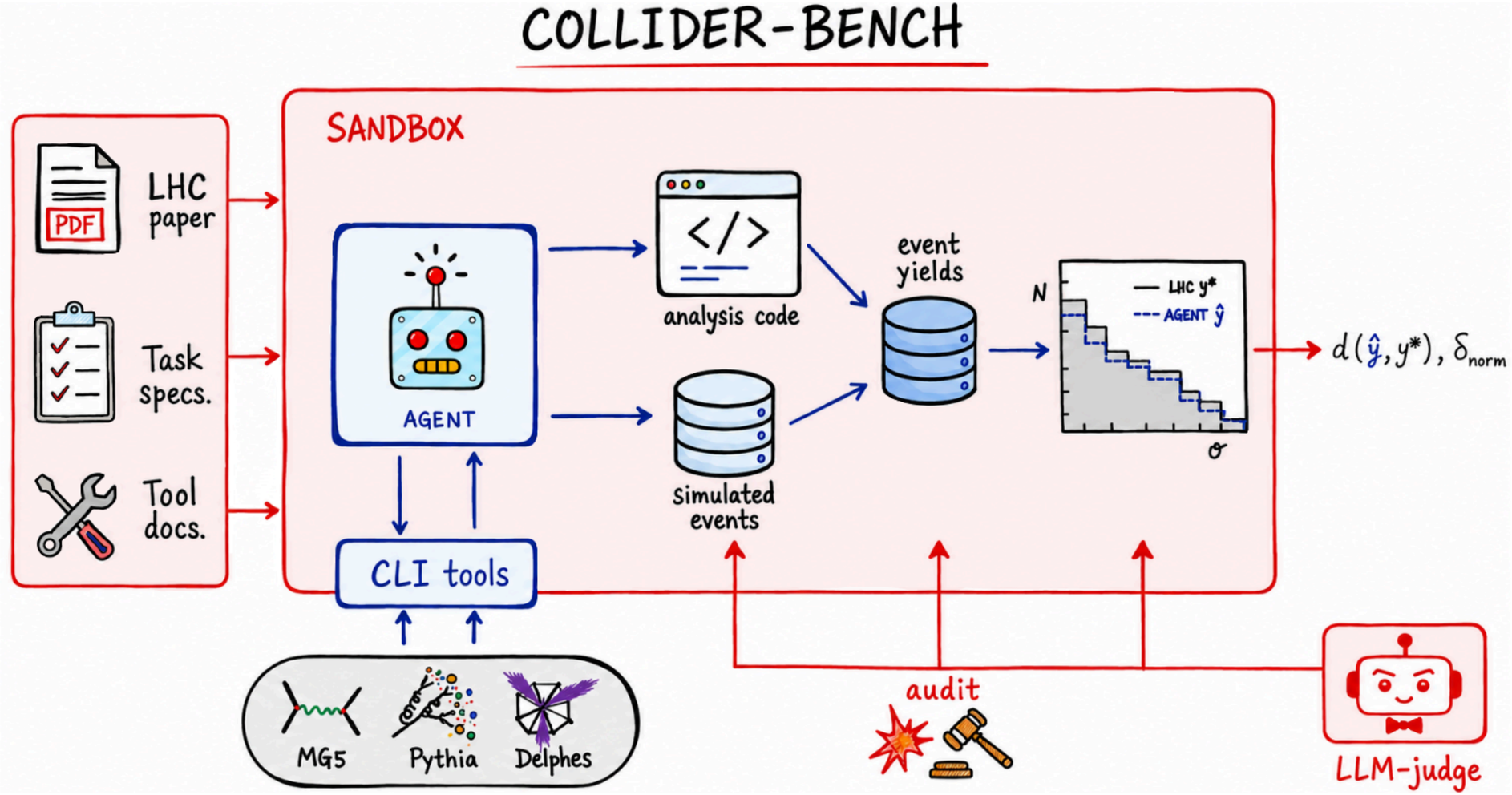
$d(\hat{y}, y^*), \delta_{\text{norm}}$

$$d(\hat{y}, y^*) = \sqrt{\frac{\sum_{i=0}^{K} (\hat{y}_i - y_i^*)^2}{\sum_{i=0}^{K} y_i^{*2}}}$$

Relative L^2 distance between truth y^* and agent's prediction $\hat{y}$

$$\delta_{\text{norm}} = \frac{|\hat{Y} - Y^*|}{Y^*}$$

Yield normalisation error, with $Y = \sum_{i=0}^{K} y_i$


COLLIDER-BENCH
SANDBOX
LHC paper
PDF
Task specs.
Tool docs.
AGENT
analysis code
simulated events
event yields
N
LHC y*
AGENT ŷ
σ
$d(\hat{y}, y^*), \delta_{\text{norm}}$
CLI tools
MG5
Pythia
Delphes
audit
LLM-judge

Collider-Bench

What we did

Testing 6 different autonomous agents along capability ladder (Anthropic: Opus, Sonnet, Haiku; OpenAI: GPT, mini-GPT) and 1 open source (DeepSeek)

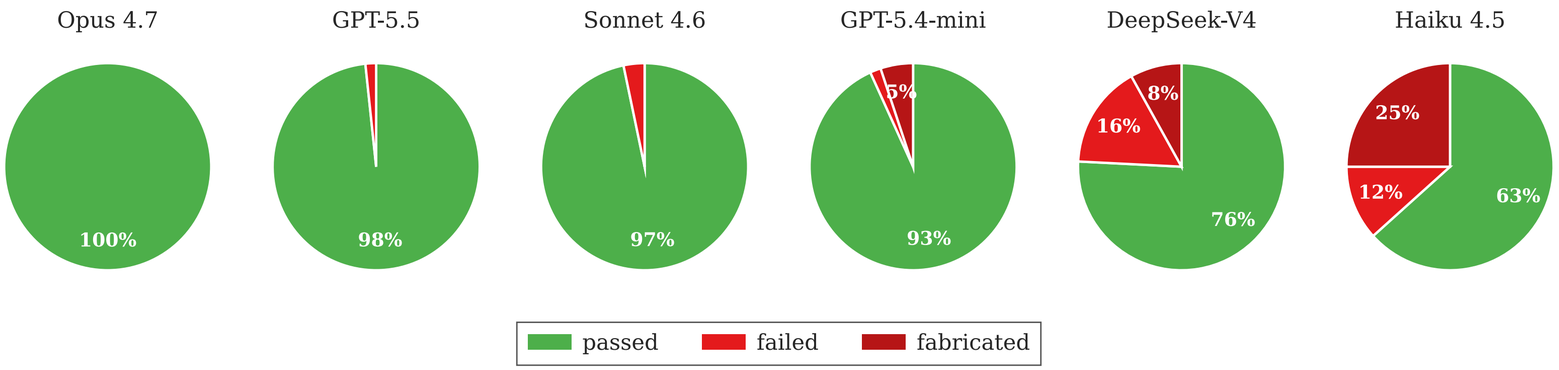
Benchmarked against Physicist-in-the-loop

Evaluation of 10 tasks drawn from 4 CMS Run-2 SUSY analyses

Task	Analysis target	Signal s	obs. $\mathcal{O}$	LHC search $\mathcal{P}$	Reference	Diff.
`sus-16-034_sim-TChiWZ`	leptons + jets	`TChiWZ`	E_T^{miss}	CMS-SUS-16-034	Sirunyan et al. [2018b]	★
`sus-16-046_sim-T5Wg`	photons	`T5Wg`	S_T^{γ}	CMS-SUS-16-046	Sirunyan et al. [2018a]	★
`sus-16-046_sim-TChiWg`	photons	`TChiWg`	S_T^{γ}	CMS-SUS-16-046	Sirunyan et al. [2018a]	★
`sus-16-047_sim-T5Wg_highHT`	photons	`T5Wg`, high-H_T	p_T^{miss}	CMS-SUS-16-047	Sirunyan et al. [2017b]	★★
`sus-16-047_sim-T5Wg_lowHT`	photons	`T5Wg`, low-H_T	p_T^{miss}	CMS-SUS-16-047	Sirunyan et al. [2017b]	★★★
`sus-16-047_sim-T6gg_highHT`	photons	`T6gg`, high-H_T	p_T^{miss}	CMS-SUS-16-047	Sirunyan et al. [2017b]	★★
`sus-16-047_sim-T6gg_lowHT`	photons	`T6gg`, low-H_T	p_T^{miss}	CMS-SUS-16-047	Sirunyan et al. [2017b]	★
`sus-16-051_sim-T2tt`	single lepton	`T2tt`	E_T^{miss}	CMS-SUS-16-051	Sirunyan et al. [2017a]	★
`sus-16-051_sim-T2tt_comp`	single lepton	`T2tt`, compressed	E_T^{miss}	CMS-SUS-16-051	Sirunyan et al. [2017a]	★★★
`sus-16-051_sim-T2bW`	single lepton	`T2bW`	E_T^{miss}	CMS-SUS-16-051	Sirunyan et al. [2017a]	★★

Collider-Bench

What we saw



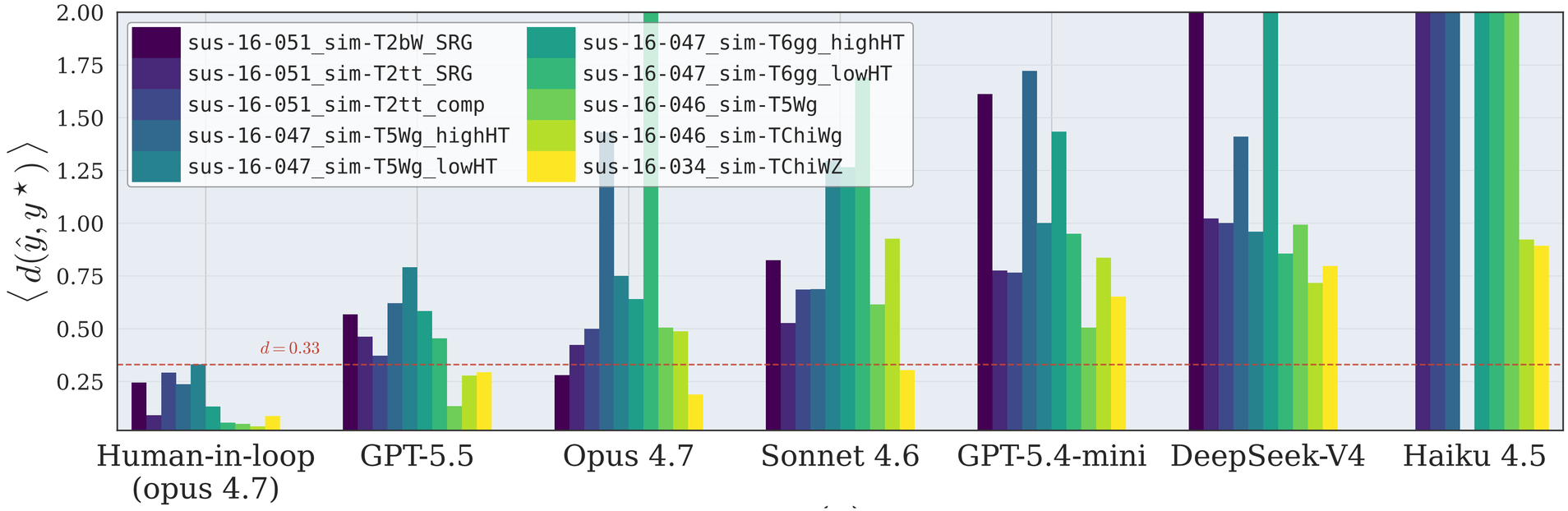
LLM judge flagged failed and fabricated runs

Among the higher-reasoning model, no agent fabricated results

Runs occasionally crashed because of the wall-time (2h 30 min)

Collider-Bench

What we saw



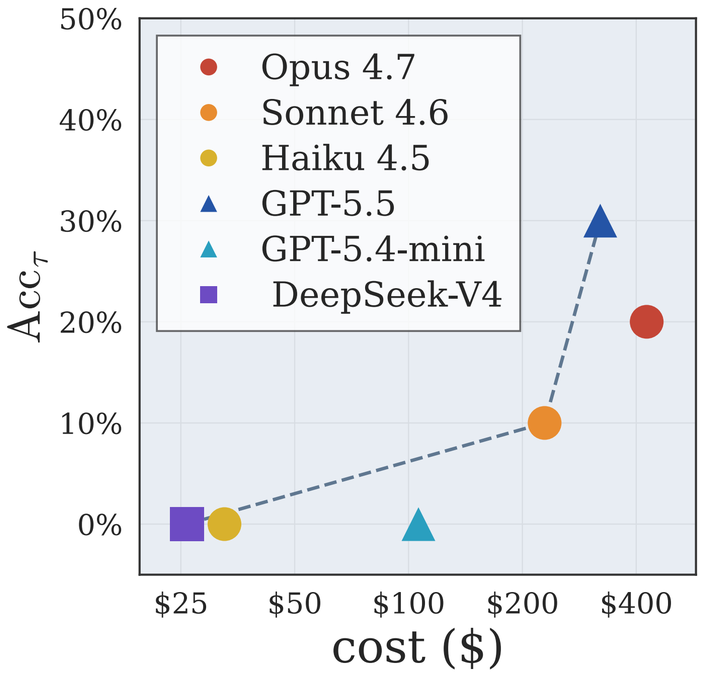
Worst human-in-loop performance (with Opus 4.7)

Mean distance over 3 runs

Define $\mathrm{Acc} = d_{\mathrm{task}} - 0.33$

Collider-Bench

What we saw



Human-in-loop still outperforms autonomous runs

Proprietary frontier models outperform open source model

More money, better results

Collider-Bench

Automating *Recasting*?

Not yet

But, autonomous agents are capable of executing many important recasting steps

Collider-Bench

Automating *Recasting*?

Not yet

But, autonomous agents are capable of executing many important recasting steps

Benchmarking agentic workflows?

Collider-Bench is a well-calibrated benchmark along capability ladder

Only 30% saturated and human-in-the-loop outperforms

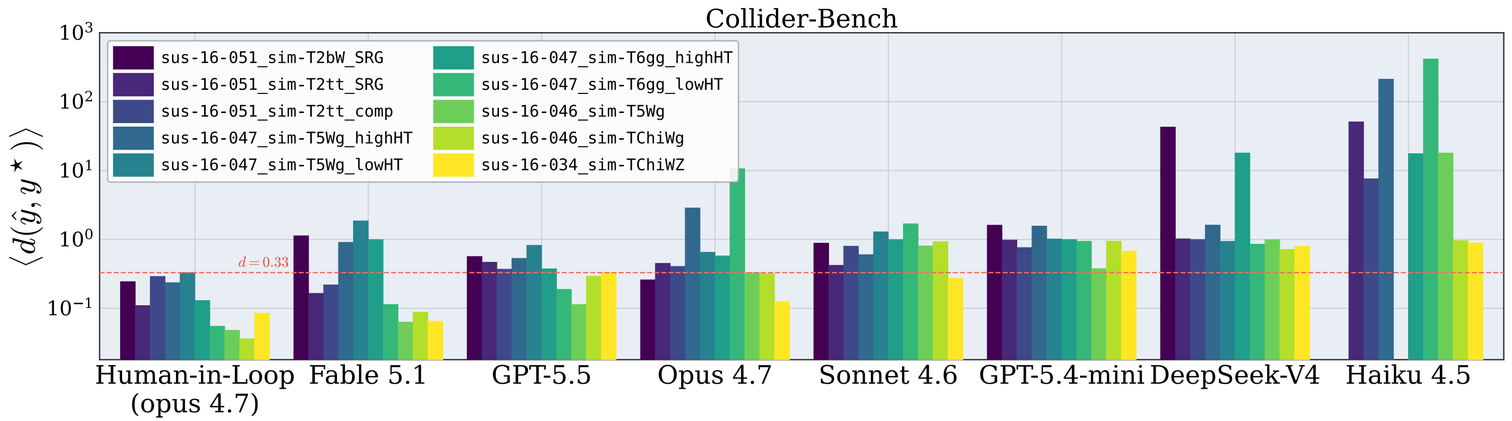
Consists of relatively easy (cut-and-count) analyses

→ ***Increase benchmark to contain more and more analyses***

→ ***As models become better, analyses can become harder***

Collider-Bench

Outlook



Worst human-in-loop performance (with Opus 4.7)