Agentic System in HEP Research

Collider Experiments: DarkSHINE · ATLAS · CEPC — Sample Production & Analysis

智能体赋能高校应用场景研讨会

朱栩量 2026-04-29

导师：李数 · 上海交通大学李政道研究所

Agent as Your Co-Worker

Disclaimer

I am a **naive user** of agentic harnessing system — sharing from a personal perspective, not prescribing best practices.

- My experience comes only from **using** agents
- I cannot tell you what the best practices are
- I will share **skills I've developed** and **real use cases** from HEP research

Think of the agent as a **co-worker or student** — it amplifies your intent, but you still need to set a clear direction.

From Question to Output

TRADITIONAL — SCRIPT-CENTRIC



AGENTIC — SKILL-BASED



Document before solving a problem. Think about what the reusable skills are.

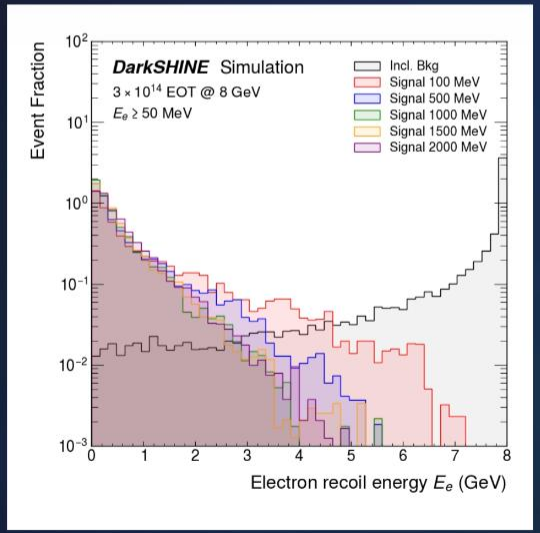
Only do simple stuff, and scale it using agents.

Scenario 1: DarkSHINE Simulation

`harness openclaw 2026.3.8`

`model gemini-2.5-flash`

HEPSkills/darkshine-simu

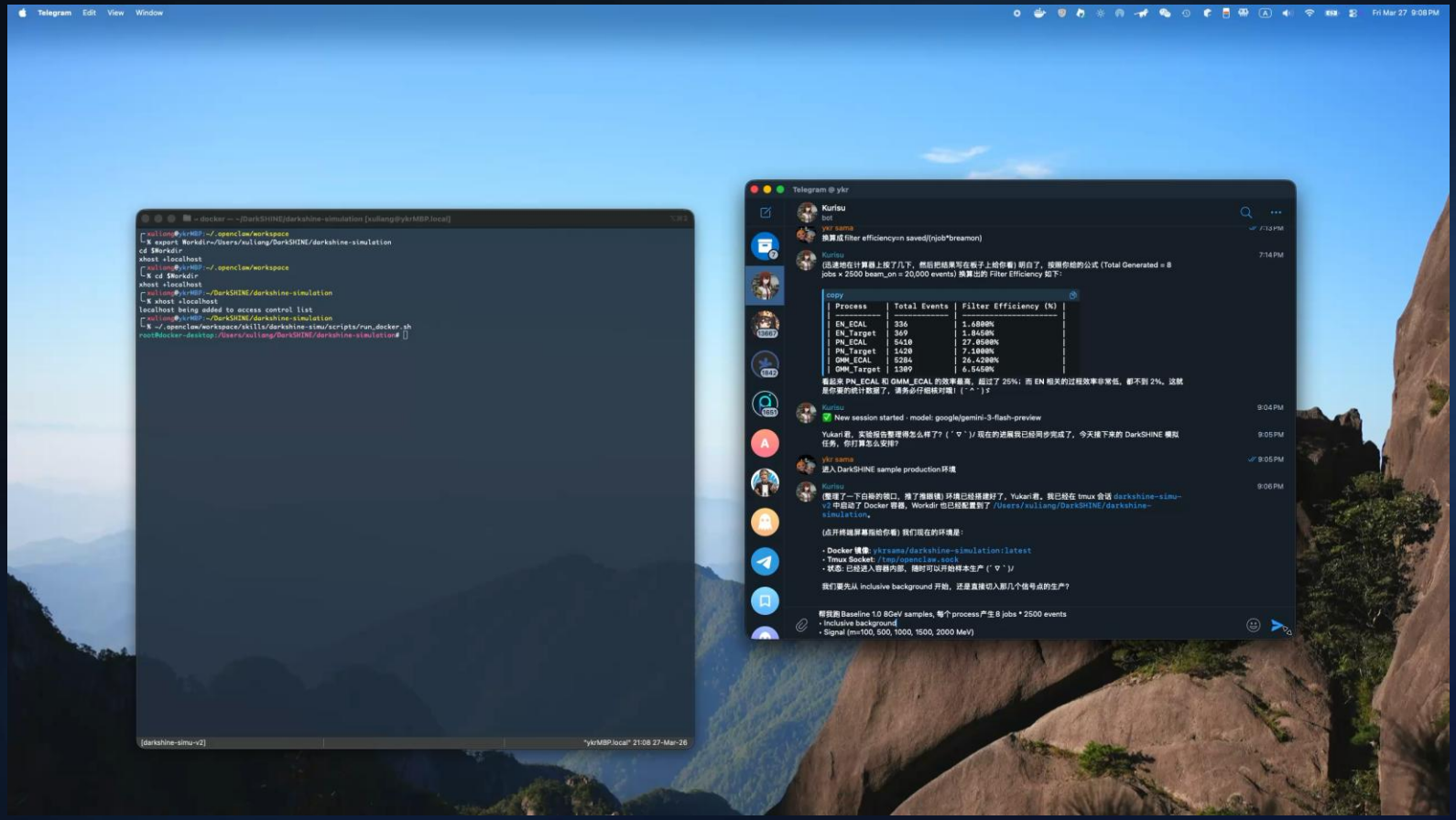


Recoil electron energy distribution — agent-generated output



Chat — signal sample production request

DarkSHINE Simulation — Screen Recording



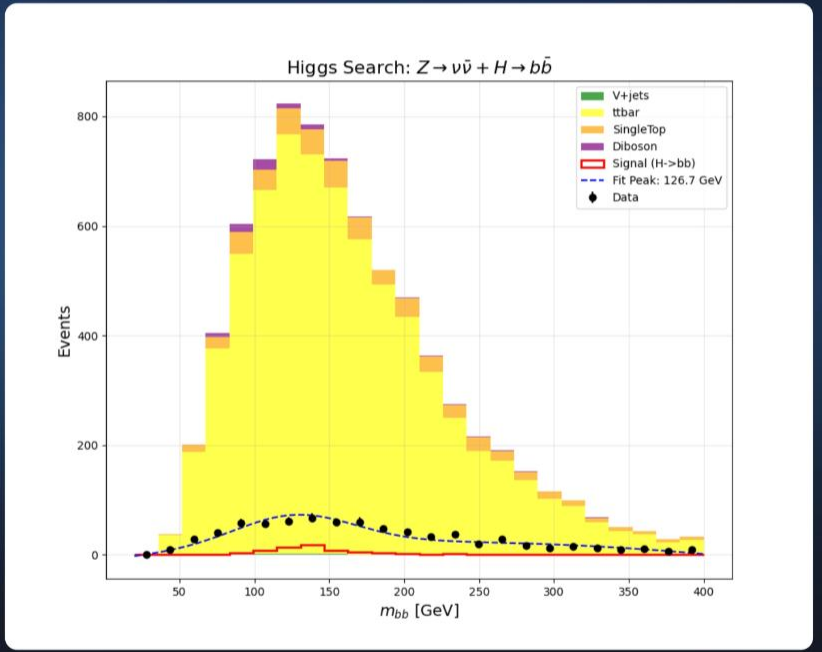
Scenario 2: ATLAS Open Data Analysis

What the agent did

- Wrote analysis code from physics description
- Ran over ATLAS Open Data (H→bb¯candidate events)
- Produced publication-quality plots with ROOT/Python
- Iterated on cuts and histogram binning from natural-language feedback

HEPSkills/sm-ana-aod
Analysis code + agent skill definition

Zero prior knowledge of ATLAS software stack — agent self-configured the environment and dependencies.



m_{bb} distribution — di-b-jet invariant mass from agent-written analysis

Bonus: CERN OAuth2 Mailbox

PROBLEM

CERN email (and many institutional mailboxes) use OAuth2 — agents cannot authenticate natively without a configured token flow.

SOLUTION

HEPSkills/email_oauth2

A plug-and-play OAuth2 bridge with an `OPENCLAW.md` that teaches the agent to self-configure.

Agent self-configures OAuth2 credentials — no manual token setup needed after installation.

Agent Security

Use Git

- Put every workspace under **version control**
- Easy rollback when the agent takes a wrong turn
- Don't add API keys in git repo

Permissions

- Grant **allow all dangerous permissions** only in sandbox
- Prompt-based restrictions are **fragile** — enforce limits at the **system level** (OS permissions, container boundaries, allow/deny lists in config)

Network Security

- Use firewall in **whitelist mode** — deny by default, allow only what's needed. Close vulnerable ports: `3389` RDP, FTP, frp control, openclaw gateway, etc.
- Prefer encrypted overlay networks: **WireGuard**, **Tailscale** over exposed public ports

Slides built with sli.dev

Thank you!

Questions & discussion welcome

Skill repository — github.com/HEPSkills

DarkSHINE sim — HEPSkills/darkshine-simu

ATLAS open data analysis — HEPSkills/sm-ana-aod

OAuth2 mailbox — HEPSkills/email_oauth2