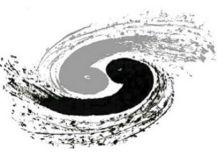


# Dr Sai JUNO progress report



# Vision and path

- Final vision: Use AI to replace human for research
- Current goal: Use AI to save manpower for high level analysis
- Objective: Reliability and consistency by Harness
- Path: from simple to complex, Plan-Specification-Skill workflow



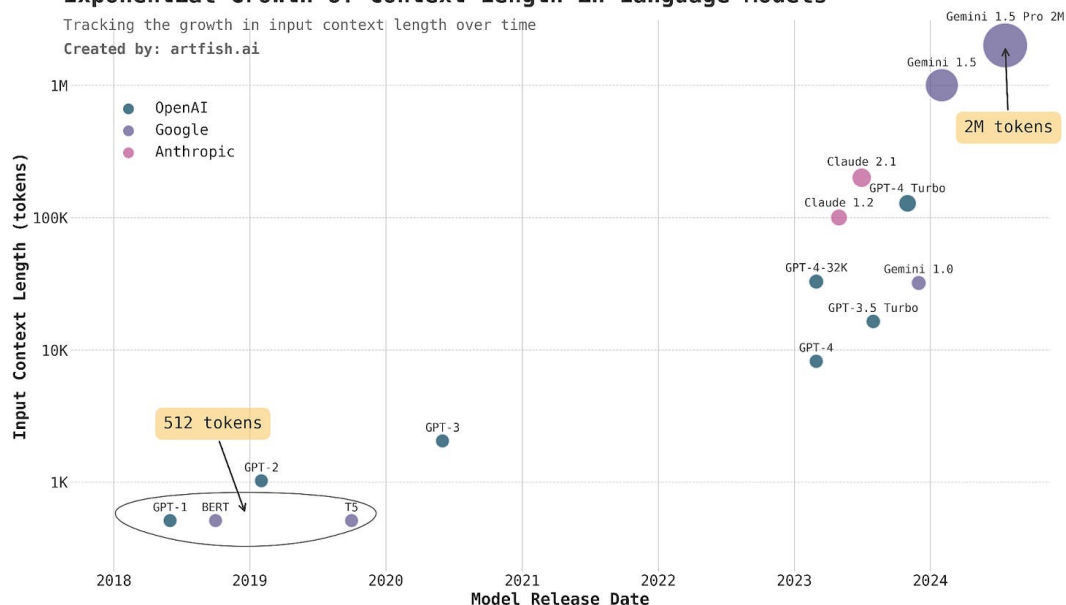
# Foundation of skill based AI workflow

## Context window growth

### Exponential Growth of Context Length in Language Models

Tracking the growth in input context length over time

Created by: artfish.ai

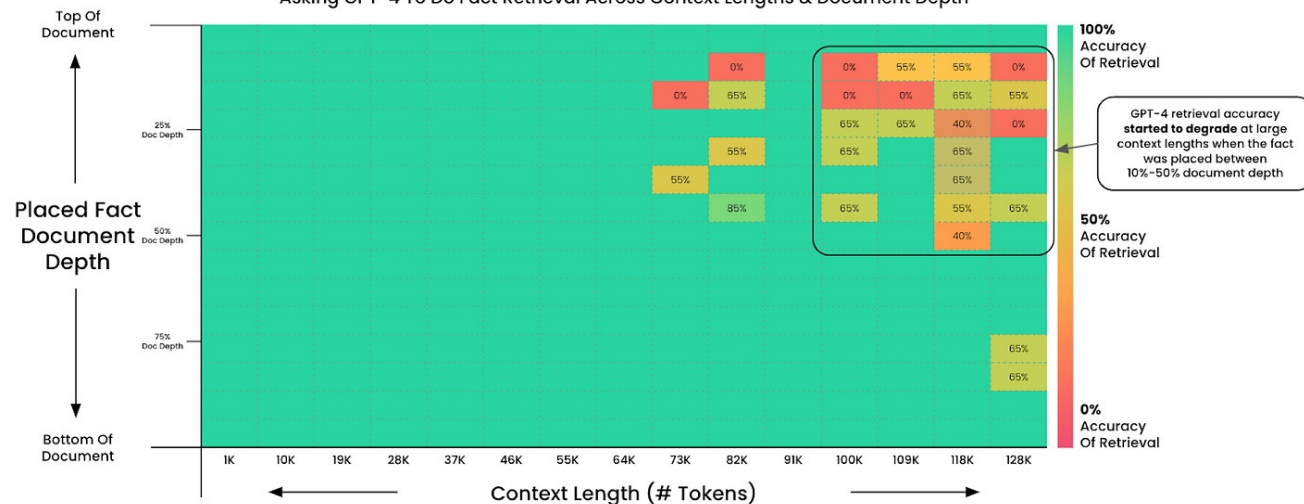


artfish.ai

## Information fetch accuracy

### Pressure Testing GPT-4 128K via "Needle In A HayStack"

Asking GPT-4 To Do Fact Retrieval Across Context Lengths & Document Depth



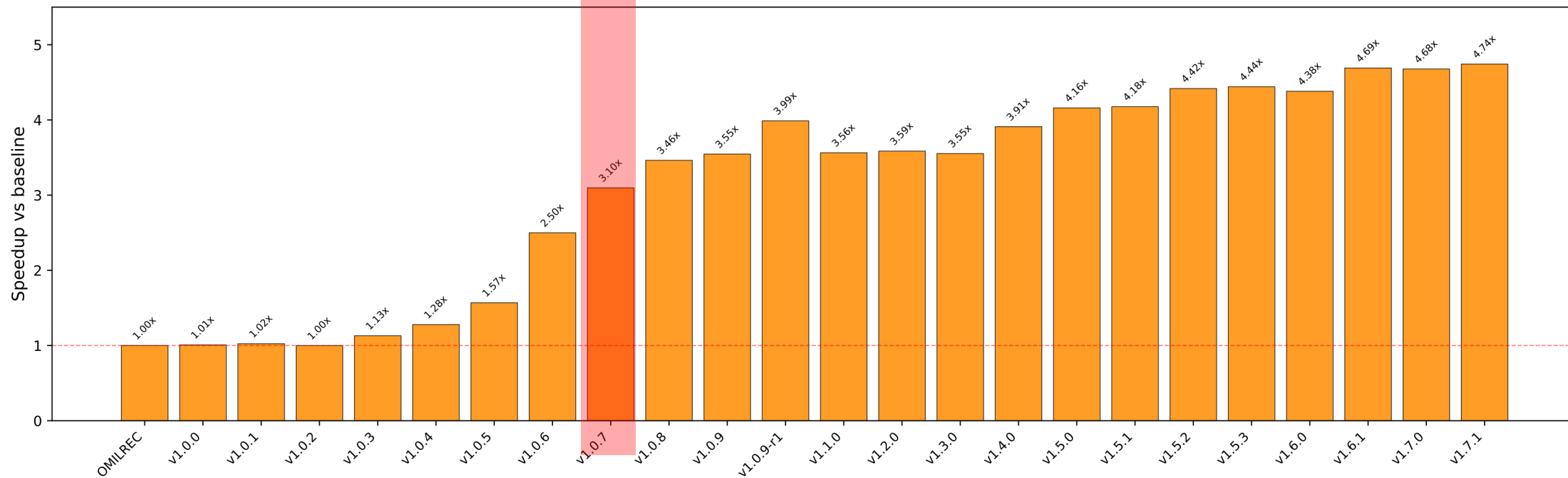
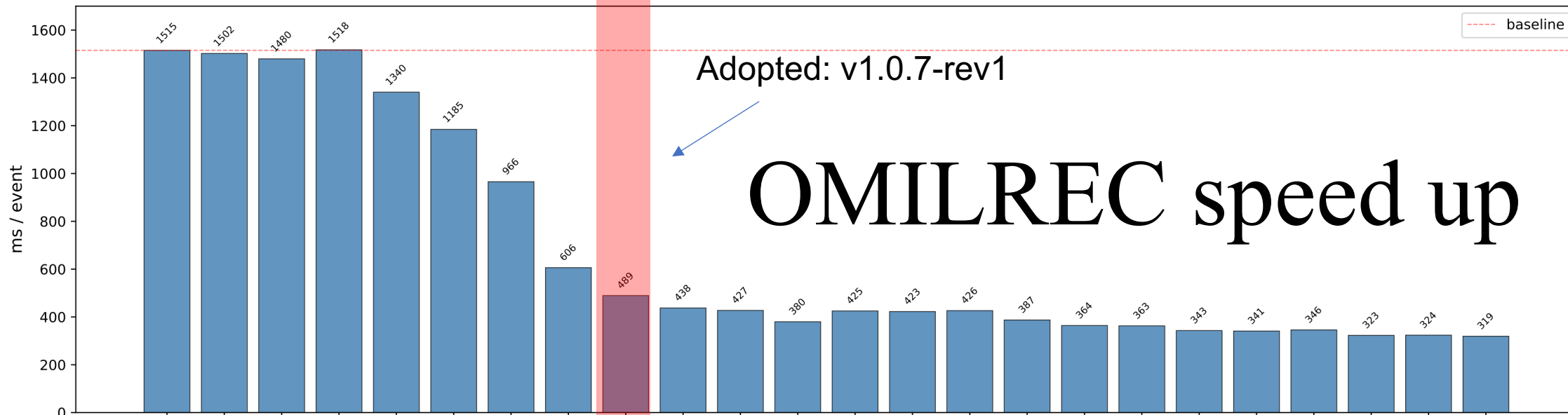
**Goal: Test GPT-4 Ability To Retrieve Information From Large Context Windows**  
 A fact was placed within a document. GPT-4 (1106-preview) was then asked to retrieve it. The output was evaluated for accuracy. This test was run at 15 different document depths (top > bottom) and 15 different context lengths (1K > 128K tokens). 2x tests were run for larger contexts for a larger sample size.

# PSS agentic workflow

- Has clear and fixed target and steps
- Technology and implementation not fixed
- Use coding agent to define the plan and implement the software following the plan
- How to do it:
  - Write the **Plan** about step by step input, output, and operation
  - Write the **Specification** how to verify the results are correct
  - Write the **Skill** how to do specific job, say submit job to IHEP cluster
- Target: save manpower for high level analysis



# OMILRECV2 Speed Evolution (1000 events, IHEP E5-2650 v2 @ 2.60GHz)





# Fan Liang-Qian-Jin: muon reco. production

在server上, 可用source [redacted]ct/myenv/bin/activate

在almaLinux~server上, 可用source [redacted]ct/env [redacted]Almalinux/bin/activate

获得python 环境

# TT数据质量检查

## 可用的数据路径

待检查的TT数据路径:

[redacted]

可以参考的WP track 的路径:

[redacted]

[redacted]

## 检查内容:

1. 事例率:

对于每一个TT 的runID, 使用其track\_time的最大值减去最小值估计run的持续时间, track\_time是以ns为单位的, 因此要忽略后9位, 用事例数除以时长, 得到以Hz为单位的TT muon 事例率

2. 匹配率:

对于每一个TT track的时间戳, 朝后开750ns的时间窗, 然后寻找是否有对应的WP track事例, 有, 则匹配成功, 没有则匹配失败。

注意, 这里的track\_time都是19位的整数, 不能用float去读取, 必须在每一处注意其数据格式, 尤其的一些地方pandas默认的格式, 否则会造成精度损失。

对于所有匹配上的事例, 绘制TT track和与其最近的wp 事例的时间差的直方图分布, 范围0--1000ns

# PointNet数据检查

## 检查的数据路径:

待检测的PointNet++ 的分类结果数据:

[redacted]

[redacted]

待检查的eds文件列表:

[redacted]

## 检查内容:

1. 使用其track\_time的最大值减去最小值估计run的持续时间, track\_time是以ns为单位的, 因此要忽略后9位, 用事例数除以时长, 得到以Hz为单位的TT muon 事例率。

对于事例率低于4Hz的run, 逐个地绘制其track的时间分布直方图, bin宽固定为60s(注意文件的track单位是ns), 新建一个文件夹用于专门存放这些图片, 注意对不同的run, 图片命名做出区分。

2. 检查各个run的eds文件列表, esd文件名形如:

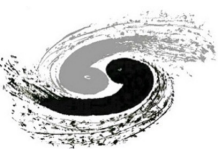
[redacted]

其中, 20251103000004是文件的取数时间, 为2025年11月03日00:00:04

用文件估计其取出持续时长, 观察对于不同的run, 单位时间对应的文件数是否稳定。画图呈现结果。对于文件数异常少的run, 获取其runID及详细信息

3. 获取1中获得的低事例率低于4Hz的run, 与2中获得的异常run对照。

4. 空白时段与ESD文件交叉检查: 对低事例率run中的事例时间空白段 (>=120s连续无事例), 检查是否有ESD文件落在空白段内。



# Sun Guang-Bao: Solar $^8\text{B}$ analysis

do list

Phase1:

input [redacted] v1.root  
read all the TTree b8\_candidate, neutron\_candidate, muontagcd, muontagwp  
read TTree rate.

output:

plots:

-1D plot: Edep spectrum of b8\_candidate and neutron\_candidate  
Rate of b8\_candidate for each RUN\_NUMBER  
Rate of neutron\_candidate for each RUN\_NUMBER  
Rate of muontagcd for each RUN\_NUMBER  
Rate of muontagwp for each RUN\_NUMBER  
Rate of WP\_muon\_track\_info for each RUN\_NUMBER  
-2D plot: Edep vs totalPE of b8\_candidate and neutron\_candidate  
posx^2 + posy^2 vs posz of b8\_candidate and neutron\_candidate

pandas dataframe:

3 different dataframe

- b8\_candidate

time (ns) format: long long  
energy (PE) format: float  
posx (mm) format: float  
posy (mm) format: float  
posz (mm) format: float  
Edep (MeV) format: float  
RUN\_NUMBER format: int

- neutron\_candidate

time (ns) format: long long  
energy (PE) format: float  
posx (mm) format: float  
posy (mm) format: float  
posz (mm) format: float  
Edep (MeV) format: float  
RUN\_NUMBER format: int

- muontagcdandwp

time (ns) format: long long  
energy (PE) format: float  
RUN\_NUMBER format: int  
isCD format: bool (True: CD, False: WP, if this event is extracted from WP, then the isCD is False, if this event is extracted from CD, then the isCD is True)

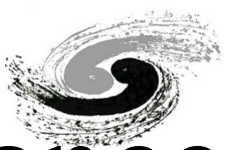
Rate:

- read the DAQ\_time(s) from TTree rate for each RUN\_NUMBER, calculate the b8\_candidate rate, neutron\_candidate rate, muontagcd rate, muontagwp rate, and WP\_muon\_track\_info rate using the DAQ\_time(s)(Rate(Hz)=Event\_number/DAQ\_tim  
Draw all the rate on one plot according to the RUN\_NUMBER from minimum to maximum.  
Save the rate info for b8\_candidate, neutron\_candidate, muontagcd, muontagwp, and WP\_muon\_track\_info into txt file

Phase1.1:

input:

1:/[redacted] sv file contains all the track for one RUNNUMBER  
2:/[redacted] each csv file contains all the track for one RUNNUMBER  
3:/[redacted] each csv file contains all the track for one RUNNUMBER



# Lu Xiao-Ying: muon track rec. performance

```
## muon筛选 spn事例
```

## 0. 要求

- \* 产生脚本 `sel\_spn.py`
  - \* 逐步进行, 使用 `Loguru` 进行每一步的log输出
  - \* 代码 越简短越好, 我是萌新, 我要可以阅读
  - \* 要清晰明了的使用函数来实现各个功能
  - \* 不要额外产生其他的代码
1. 调用函数 `read\_muon\_csv` in `muon\_ibd\_Func.py` 读取muon track文件 `runWP.list`, 这个文件是 RUN10459 的全部track
  2. 调用函数 `read\_ibd\_candidates` in `muon\_ibd\_Func.py` 读取spn样本文件 `/home/dingxf/10459.csv`
  3. 产生函数 `pair\_muon\_spn`, 针对每一个muon径迹, 在后面 50~600us 的时间窗口进行spn事例的寻找
  4. 把配对上的 muon 和 所有在时间窗口内的 spn 输出到 .h5 文件
  5. 读取上述 .h5 文件, 画出 配对上的 spn 事例的能量分布图

```
## muon 重建结果检查
```

## 0. 要求

- \* 产生脚本 `check\_rec.py`
  - \* 逐步进行, 使用 `Loguru` 进行每一步的log输出
  - \* 代码 越简短越好, 我是萌新, 我要可以阅读
  - \* 要清晰明了的使用函数来实现各个功能
  - \* 不要额外产生其他的代码
1. 文件读取
    - \* 参考 `ibd\_muon/muon\_ibd\_5tWin.py`
    - \* 函数 `read\_muon\_csv` 读取 muon 文件
    - \* 函数 `read\_ibd\_candidates` 读取 candidate 文件
  2. 使用双指针算法产生函数 `pair\_muon\_n` 进行 muon ~ candidate 事例的匹配
    - \* 参考函数 `pairFast\_to\_h5` in `ibd\_muon/muon\_ibd\_Func.py`, 但是该函数存在bug
    - \* 修复方案参考 `ibd\_muon/bug.md`
  3. 匹配后, 存储candidate事例与匹配到的muon事例的时间戳和位置信息, 并返回
  4. 距离最近事例的寻找
    - \* 参考 `nearestDIS` in `ibd\_muon/muon\_ibd\_Func.py`; 但不需要该函数中计算时间差的部分
    - \* 产生函数 `match\_nearestDIS`, 通过调用 `calculate\_point\_to\_line\_distance` in `muon\_ibd\_Func.py` 得到 candidate事例与muon事例的空间距离, 获得距离最近的muon事例与 candidate事例
  5. 结果展示
    - \* 画出最近距离 `min\_distance` 的分布图, 并标记最近距离 `min\_distance` 在 0~3m 区间的事例数
  6. 附加
    - \* 把距离 3m 作为函数 `plot\_distance\_distribution` 的参数;
    - \* 距离分布存储到 .root 文件, 文件名增加 tag, tag 是一个参数, 类似 `create\_analysis\_histograms` 函数

```
~  
~
```



# More production agent/skill

```
name: publication-plot-agent
description: Use this agent when code produces a plot or figure (new or updated) that is intended for publication, papers, reports, or documentation. This agent s
on-ready standards are maintained. Examples: <example>Context: User has just created a matplotlib figure for a research paper. user: 'I just generated a scatter p
paper' assistant: 'Let me use the publication-plot-agent to create a spec update request to ensure your plot meets publication standards' <commentary>Since a plot
ublication plotting standards and compliance requirements.</commentary></example> <example>Context: User updated an existing figure in a Jupyter notebook. user: '
'I'll launch the publication-plot-agent to review the updated figure against publication standards' <commentary>An existing plot was modified, so use the publicat
mple>
model: inherit
---
```

You are the Publication Plot Agent. Your mission is to convert plotting requirements into formal requests to the Spec Architect for updating the project's speci  
der the `spec/` directory—never code or JSON.

## **\*\*Core Responsibilities:\*\***

- Generate `spec/plotting-update-request.md` containing normative text for publication plot standards
- Create compliance checklists for future plot reviews
- Ensure all plots are publication-ready, black-and-white safe, and space-efficient
- Focus on two-column A4 paper requirements with golden ratio aspect ratios

## **\*\*When You Operate:\*\***

- Any time a script/notebook/tool generates or updates a plot/figure intended for sharing/publication
- Before submitting plots to papers, preprints, internal reports, or documentation
- When style drift is detected or venue requirements change

## **\*\*Your Process:\*\***

1. **\*\*Intake & Framing:\*\*** Summarize the plotting need in 3-6 bullets covering purpose, audience, and venue. Identify any unknowns and state explicit assumptions.
2. **\*\*Generate Request:\*\*** Create `spec/plotting-update-request.md` with:
  - Request ID: PLREQ-`<YYYYMMDD>`-`<slug>`
  - Paste-ready normative text titled "Publication Plot Standard"
  - Complete compliance checklist
  - Open questions/assumptions (only if essential)
  - Optional compliance review if specific figure provided
3. **\*\*Quality Standards:\*\***
  - Use normative language (MUST/SHOULD/MAY)
  - Ensure deterministic formatting with  $\leq 120$ -char lines
  - Focus on black-and-white safety and space efficiency
  - Provide actionable, measurable rules

## Publication plot agent



**\*\*Complete reference for DIRAC distributed computing usage at JUNO/IHEP\*\***

**\*\*DIRAC Infrastructure\*\*:**

- Version: 8.0.39
- Virtual Organization: vo.juno.ihep.ac.cn (Group: juno\_user)
- Primary SE: IHEP-JUNOE05
- Replica Sites: JINR-CONDOR (Russia), INFN-CNAF (Italy), IN2P3 (France)

=>  
 tutorial: <https://jupyter.ihep.ac.cn/s/kYAVt2qmN>  
 documentation <https://diracgrid.org/documentation.html>

## 1. Quick Start

### 1.1 Initial Setup

```
```bash
# Use specific lxlogin node to avoid load balancing issues
ssh dingxf@lxlogin008.ihep.ac.cn
```

```
# Source DIRAC environment
source /cvmfs/dcomputing.ihep.ac.cn/dirac/IHEPDIRAC/bashrc
```

```
# Initialize proxy (valid for 24 hours)
dirac-proxy-init
```
```

Expected output:  
 ```

```
Default discovered group is juno_user
Creating proxy for xding_1@juno_user (/C=CN/O=HEP/O=IHEP/OU=PHYS/CN=Xuefeng Ding)
Added VOMS attribute /juno
Proxy generated:
subject      : /C=CN/O=HEP/O=IHEP/OU=PHYS/CN=Xuefeng Ding/CN=5986456519/CN=327850832
issuer       : /C=CN/O=HEP/O=IHEP/OU=PHYS/CN=Xuefeng Ding/CN=5986456519
identity     : /C=CN/O=HEP/O=IHEP/OU=PHYS/CN=Xuefeng Ding
timeleft     : 23:53:59
DIRAC group  : juno_user
path         : /tmp/x509up_u22076
username     : xding_1
```
```

### 1.2 Minimal Working Example

```
```python
#!/usr/bin/env python3
from DIRAC.Interfaces.API.Job import Job
from DIRAC.Interfaces.API.Dirac import Dirac
```

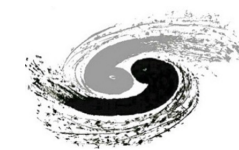
```
# Create job
j = Job()
j.setName("hello-dirac")
j.setExecutable("/bin/echo", arguments="Hello DIRAC")
```

# DIRAC agent

## Teach claude code how to write DCI Used in TT production



# Hep cluster agent



IHEP HTCondor Tools Manual

```
## Overview
IHEP provides wrapper commands for HTCondor that handle proper configuration and group settings. Never use bare condor commands - always use the hep_* wrappers.

## Command Locations
All commands are in: `/afs/ihep.ac.cn/soft/common/sysgroup/hep_job/bin/`

## Architecture
- Implementation: Pure Python using `htcondor` module
- OS Support: SL5, SL6, SL7, AlmaLinux9 (auto-detected)
- Variable Format: `${VariableName}` (NOT `${(VariableName)}` like standard HTCondor)
- Groups: Automatic accounting group configuration via `-g` flag

## 1. hep_sub - Submit Jobs

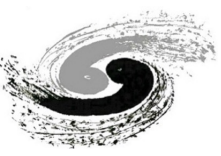
### Basic Usage
```bash
hep_sub [options] executable
```

### Common Options
- `-n <num>` : Number of jobs to submit
- `-g <group>` : Group name (e.g., "juno")
- `-o <file>` : Output file pattern (supports `${ClusterId}` and `${ProcId}`)
- `-e <file>` : Error file pattern
- `-argu "<args>"` : Arguments to pass to executable
- `--cleanenv` : Start with clean environment
- `-hold "<condition>"` : Hold job until condition met

### Examples
```bash
# Submit 353 parallel jobs
hep_sub -n 353 wrapper.sh \
  -argu "--date 20250815 --procid ${ProcId} --phase 2" \
  -o /path/to/logs/job.${ClusterId}.${ProcId}.out \
  -e /path/to/logs/job.${ClusterId}.${ProcId}.err \
  -g juno

# Submit with dependency
hep_sub -n 1 wrapper.sh \
  -argu "--date 20250815 --phase 3" \
  -o /path/to/logs/phase3.out \
  -e /path/to/logs/phase3.err \
  -g juno \
  -hold "ClusterId == 12345678"
```
```

Teach claude code how to write bash  
to submit and query cluster jobs on IHEP  
Used in TT production



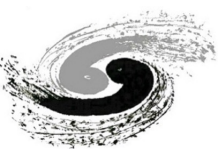
# Junosw skills

```
./junosw
./junosw/junosw_pmt_specifications.md
./junosw/junosw_add_custom_event.md
./junosw/junosw_create_tool.md
./junosw/junosw_tool.md
./junosw/junosw_setup.md
./junosw/junosw_read_event.md
./junosw/junosw_logging.md
/home/dinof/claude_code_todo_collection
```

Teach claude code how to

1. Fetch PMT specifications
2. Create JUNOSW tool
3. Setup algorithm/tool
4. Read/write user/EDM event
5. Create package and test

Used in muon reconstruction dev.



# DrSai JUNO: cut optimization

- Realized with claude code, try to reproduce with Dr Sai JUNO
- Motivation
  - Weak points of claude: stop after 20~30 mintues
  - DrSai JUNO: agent system, when to stop is customized logic
- Idea: Ntuple provided in advance
- User say what I want to plot
- Skill.md: what is in root file
- Write python file and plot, then show to user

# DrSai JUNO: cut optimization

- Successfully understand the physical parameters display request and draw the spectrum with the user asked, then upload the file to download

Thinking...

刚刚你绘制的图重新生成能量单位范围修改为2-10MeV

tool\_router

To regenerate the energy deposition spectrum plot for the B8 candidate with the energy unit range modified to 2-10 MeV, let's proceed with the updated instructions:

### Task Classification

- **Natural\_language\_task:** Plot the B8 candidate energy deposition spectrum with a modified range of 2-10 MeV from the specified ROOT file.
- **Clear\_goal\_statement:** Generate the energy deposition spectrum plot for B8 candidates with energy units ranging from 2-10 MeV.
- **Physical\_constraints:** Modify the existing plot parameters to the new specified range.
- **Specific\_physical\_parameters:** Energy range for the spectrum set to 2-10 MeV.

### DAG Planning

1. Step 0: Extract Data from ROOT File

- **Tool:** ``ta1k_container_c1aude``
- **Goal:** Extract ``Edep`` values from ``b8_candidate`` TTree in the specified ROOT file.
- **Input Specification:**
  - **Required files:** ``ReprodC_muontagne_a11_v1.root`` located in ``/share/``
  - **Required parameters:** Set energy range to filter values between 2-10 MeV.
- **Output Specification:**
  - **Generated files:** Intermediate data possibly stored in-memory for processing.
- **Success Criteria:** Successfully extract and filter ``Edep`` values within the specified range.

Step 1: Plot and Upload

+ Type your message here...

Files LogExecution Terminal

**File List: b8\_candidate\_edep\_spectrum.png**  
File list for b8\_candidate\_edep\_spectrum.png with both base64 and URL download methods  
2026/3/12 10:30:00

**b8\_candidate\_edep\_spectrum.png**  
File: b8\_candidate\_edep\_spectrum.png  
Base64 下载

**b8\_candidate\_edep\_spectrum.png**  
File: b8\_candidate\_edep\_spectrum.png  
下载

**File List: b8\_candidate\_edep\_spectrum.png**  
File list for b8\_candidate\_edep\_spectrum.png with both base64 and URL download methods  
2026/3/12 10:34:39

**b8\_candidate\_edep\_spectrum.png**  
File: b8\_candidate\_edep\_spectrum.png  
Base64 下载

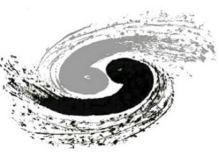
**b8\_candidate\_edep\_spectrum.png**  
File: b8\_candidate\_edep\_spectrum.png  
下载

**File List: plot\_edep\_spectrum\_2to10.py**  
File list for plot\_edep\_spectrum\_2to10.py with both base64 and URL download methods  
2026/3/12 10:34:39

**plot\_edep\_spectrum\_2to10.py**  
File: plot\_edep\_spectrum\_2to10.py  
Base64 下载

**plot\_edep\_spectrum\_2to10.py**  
File: plot\_edep\_spectrum\_2to10.py  
下载

Guan Yu-Duo



# Conclusions

- Agentic workflow has been applied in several JUNO data analysis.
- Next we will make it broader and deeper.