



AI Practitioners Forum: Use Cases, Tools, and Open Discussion

Leszek Grzanka with contributions from the audience

2026.04.13, KUKDM 2026, Zakopane



AI users : geeks or normal people?

- Adoption ranges from fear to evangelism — where is your group?
- perspective differs between paid users and free-tier users
- scope of tools varies for prompt-answer via chat to agents and multimodal systems
- Users range from Excel-only to Python-fluent — AI meets them differently
- Supervision styles: blind trust, dialogue (“tutor-student”), or skepticism?



Data privacy

- Cloud AI risk: Proprietary code or unpublished results sent to external APIs – who owns that data?
- Open vs closed science: Open-access papers make AI tools more powerful; paywalls block them
- Self-hosted models: Run on your own GPU (even a laptop) for full data control; serious workloads need Cyfronet-scale hardware
- Zoom AI assistant listens: Meeting notes generated by AI – convenient, but who has access? A fully private alternative is easy to build with modest resources



```
Analyze critically merge request. The diff with respect to the develop branch is in @diff.txt , we are now on the branch with changes. The destination branch is called develop. Use git tools to see the difference yourself. Check the changes looking for possible new bugs introduced or any other undesired changes. Change code quality. Run git command if needed. The code is mostly C. We are working on the SHIELDHIT12A tool
```

```
txt diff.txt
```

```
I'll analyze the merge request diff to identify any potential issues introduced by these changes. Let me examine the code structure and changes more carefully.
```

```
Looking at the diff, I can see multiple files were modified with changes to include statements, function signatures, and code reformatting. These appear to be primarily formatting changes to align with clang-format style, but there are some functional changes as well. Let me analyze these in detail.
```

```
Plan · Qwen/Qwen3-Coder-30B-A3B-Instruct · 2.2s
```

```
Analyze the changes piece by piece
```

```
I'll analyze the changes piece by piece to identify any potential issues or undesired changes. Let me focus on the key modifications in the diff.
```

```
* Grep "SH_SCORE_RETURN_ERROR|SH_SCORE_RETURN_NO_SCORE|SH_SCORE_RETURN_OK" in . (97 matches)
```

```
Looking at this more closely, it appears there's a key issue in the code logic around line 213-214 in sh_estimator.c. The existing code had some duplicated error handling code, and it looks like this might be related to a recent change in behavior. Let me investigate other changes to identify critical issues.
```

```
Let me analyze the key parts of the diff that could introduce bugs or unexpected behavior.
```

```
* Grep "ndiff" in . (12 matches)
```

```
Now I see a potential critical bug introduced in the diff. Let me review a few more areas of concern.
```

```
→ Read src/shieldhit/scoring/parse/sh_estimator.c [offset=210, limit=20]
→ Read src/shieldhit/scoring/parse/sh_estimator.c [offset=230, limit=20]
```

```
Plan · Qwen/Qwen3-Coder-30B-A3B-Instruct Cyfronet Forge
```

```
tab agents ctrl+p commands • OpenCode 1.1.57
```

```
Critique merge request: diff review for bugs and quality improvements
```

```
Context
39,249 tokens
0% used
$0.00 spent
```

```
LSP
• clangd
```

```
~/workspace/shieldhit:450-score-differential-let-scoring-counts-neutrons-with-let-0
```

Use case: AI code reviewer on a merge request for a particle transport Monte Carlo code

Key point: Qwen model via LLMForge – code never leaves Cyfronet infrastructure

Result: Found duplicated error handling and a potential critical bug



Literature assistant

- **Source grounding:** Does the AI point to real papers? (Perplexity cites sources; others may hallucinate references)
- **Paywall workaround:** Upload your own PDFs (e.g. NotebookLM) – AI works with what you give it
- **Paper review aid:** AI can place your work in broader context of related literature
- **Idea challenger:** Use AI as a sparring partner
- **Free tier reality:** Most tools limit depth or access on free plans – check what you actually get



Basic	Plus	Pro	Team
For students and casual exploration	For literature reviews and deeper research	For systematic reviews and professional research	For large systematic reviews with several collaborators
Free	\$10 / month \$120 billed annually	\$42 / month \$499 billed annually	\$65 per user / month \$780 billed annually
Your current plan	Choose plan	Choose plan	Choose plan

Tools

Find papers	✓	✓	✓	✓
Chat with papers	✓	✓	✓	✓
Alerts	✗	✗	10 at a time	10 at a time
Extract data	✗	✗	✓	✓
Research Agent*	✓	✓	✓	✓

*Subject to daily rate limits. Limits increase progressively for higher tier plans.

Workflows - Shared monthly total

Workflow count	2 per month	48 per year	144 per year	240 per user / year, pooled across your team
Reports	✓	✓	✓	✓
Systematic reviews	✗	✗	✓	✓

Elicit (free plan) cannot extract data from a 1984 ICRU report behind a paywall – even after multiple search attempts. Uploading the PDF yourself is the workaround, but defeats the purpose of automated literature extraction.

ICRU49 Report Overview

I'm checking the 1984 report specifically; if Table 2.8 is accessible in a source I can quote, I'll pull the element-by-element values.

✓ Ran analysis ▾

4 searches

- Web · max 10 ICRU 1984 report table 2.8 mean excitation energy I-values elements
- Web · max 10 ICRU report 1984 table 2.8 complete I values elements
- Web · max 10 ICRU 1984 stopping powers table 2.8 I-value elements PDF
- Web · max 10 ICRU report 37 1984 mean excitation energies table 2.8

✓ Sources considered: ICRU 1984 I-values >

Searching for an accessible full-text copy or a reliable reproduction of Table 2.8.

Not reliably in this environment. I can read summaries and OCR'd excerpts, but I do not have guaranteed full-text access to the ICRU report itself, so I can't verify or reproduce Table 2.8 element-by-element from the report on the basis of this scan.

The 1984 material I found is a separate NASA technical paper, and it covers selected mean-excitation-energy tables rather than the full ICRU table set: it says, "Mean excitation energies for covalent-bonded gases and solids, for ionic gases and crystals, and for metals are calculated..." and its table list includes entries like "TABLE VIII. - METALLIC PARAMETERS FOR SELECTED METALS OF FIRST TWO ROWS." [PDF] Mean Excitatio

If you upload the report PDF, I can extract the complete Table 2.8 for all elements; this is just an initial search, and a deeper pass or extended thinking would be the right way to do a faithful table extraction.

View sources

Follow-ups

Extract Table 2.8 from the report

Extended ⌚

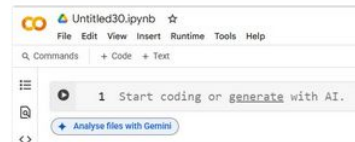
Upload the ICRU 1984 PDF



Data analysis assistant

- **Not vibe coding:** Unlike app development, data analysis needs notebooks, data files, plots – not just runnable code
- **Jupyter as the interface:** AI must read cells, run them, inspect outputs and iterate – harder than generating a single script
- **Plots are the product:** Correctness means the right visualization, not just code that runs
- **Reasoning matters most:** AI should explain *why* the data looks that way, not just *how* to plot it
- **Agents are getting there:** VSCode / Claude Code can autonomously navigate large datasets and notebooks (see next slide)

Google Collaboratory Computational Notebooks (Python) Run in Browser



Generate

plot image with colorscale to the right with ticks every 100, set max pixel value to 99% percentile of the data, add gaussian filter with kernel size 5, use such palette that pixels with value zero are transparent, while the most intensive values gets color red, values around 500 with color green, add iso-values lines for every 100, isolines should have a text with isoline value

Q

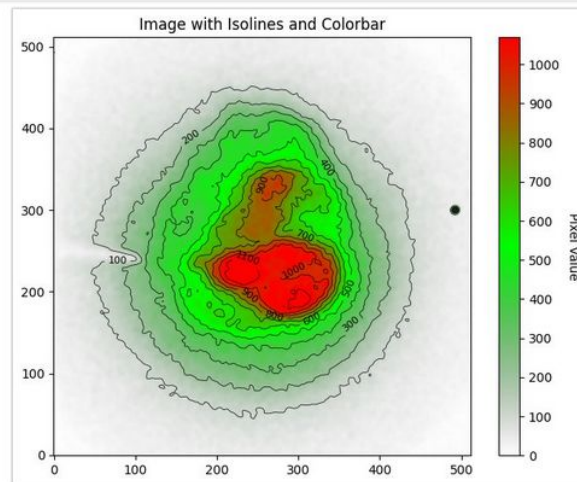
Close

1 Waiting..

```

1 # prompt: plot image with colorscale to the right with ticks every 100, set max pixel value to 99% percentile of the
2 # add gaussian filter with kernel size 5, use such palette that pixels with value zero are transparent,
3 # while the most intensive values gets color red, values around 500 with color green,
4 # add iso-values lines for every 100, isolines should have a text with isoline value
5
6 from pathlib import Path
7 import matplotlib.pyplot as plt
8 import numpy as np
9 from scipy.ndimage import gaussian_filter
10
11 file_path = Path('COSK_CuCl.tif')
12 if file_path.exists():
13     image = plt.imread(file_path)
14
15     # Gaussian filter
16     image_filtered = gaussian_filter(image, sigma=2.5)
17
18     # Calculate the 99th percentile
19     max_pixel_value = np.percentile(image_filtered, 99)
20
21     # Create a custom colormap
22     import matplotlib.colors as mcolors
23     cmap = mcolors.LinearSegmentedColormap.from_list("custom_cmap", [(0,0,0,0),(0,1,0,1),(1,0,0,1)], N=256)
24
25     # Plot the image
26     fig, ax = plt.subplots(figsize=(8, 6))
27     im = ax.imshow(image_filtered, cmap=cmap, vmax=max_pixel_value, origins='lower')
28
29     # Add colorbar
30     cbar = plt.colorbar(im, ticks=np.arange(0, max_pixel_value + 100, 100))
31     cbar.ax.set_ylabel('Pixel Value', rotation=270, labelpad=15)
32
33     # Add isolines
34     CS = ax.contour(image_filtered, levels=np.arange(0, max_pixel_value + 100, 100), colors='black', linewidths=0.5)
35     ax.clabel(CS, inlines=1, fontsize=8, fmt='%1.0f')

```



Single prompt → complete visualization pipeline in Google Colab



The screenshot displays a VS Code window with a Jupyter notebook titled "Meas08 — Waveform Fragments & Voltage Histograms". The notebook content includes a title, a description of the data, a table of timescales, and Python code for imports and configuration. The right-hand side shows the execution output of the notebook cells, including a chat window with Claude Code.

Meas08 — Waveform Fragments & Voltage Histograms

Visual gallery of raw waveforms at different noise levels (low / median / high), plus voltage histogram evolution on multiple timescales.

Data: Meas08, 10,000 segments × 25,000 samples at 10 GS/s (2.5 μ s/seg, 0.1 ns spacing)
Selected distances: 00 cm, 20 cm, best — representing close, mid, and optimal positions

Timescales for Histogram Evolution

Timescale	Grouping	Physical meaning
~10 ms	Per segment	Single trigger, sub-LTE-frame
~100 ms	10-segment block	Several LTE subframes
~1 s	100-segment block	~1 full LTE frame period
~30–40 s	Per TRC file	Full acquisition

```
import os
import sys
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from matplotlib.axes import Axes
from matplotlib.colors import LogNorm
from numba import njit, prange

sys.path.insert(0, str(Path.cwd().parents[2] / "src"))
from trc_reader import read_trc
```

Configuration

```
DATA_ROOT = Path.home() / "workspace" / "lgad_data"
MEAS08_PATH = DATA_ROOT / "Meas08"
```

CHAT CLAUDE CODE

LGAD EXPERIMENTS ANALYSIS AND PLANNING FOR PROTON BEAM

Loading took ~8 min (10 positions × 3 channels × 500 MB TRC each). Let me continue running the remaining cells:

Ran 0.2-meas08-rms-time-evolution.ipynb • cell 10
file.png

Ran 0.2-meas08-rms-time-evolution.ipynb • cell 12
file.png

Analyzed C1/C3 and C2/C3 ratio stability

Interesting — ratios are remarkably stable around 3.5–4.5× across all positions, suggesting relatively uniform spatial noise characteristics. Let me run the remaining cells:

Ran 0.2-meas08-rms-time-evolution.ipynb • cell 14
file.png

Strong autocorrelation in C1/C2 (decays slowly, significant even at 500 ms lag) — consistent with LTE frame periodicity. C3 decorrelates quickly (random noise). Let me finish the remaining cells:

Ran 0.2-meas08-rms-time-evolution.ipynb • cell 16
file.png

Ran 0.2-meas08-rms-time-evolution.ipynb • cell 17
file.png

Ran 0.2-meas08-rms-time-evolution.ipynb • cell 19

NB 0.2 complete. Key finding: **broadband RMS is remarkably flat across all distances** (~1.5 mV for C1/C2, ~0.4 mV for C3). The C3-normalized ratio stays around 3.5–4.0× everywhere. Now NB 0.3:

Configured notebook 0.3-meas08-waveforms-histograms.ipynb

Creating a Virtual Environment

Ran 0.2-meas08-rms-time-evolution.ipynb • cell 10
file.png

Starting Python Kernel for 0.3-meas08-waveforms-histograms.ipynb

Compacted conversation

Retrieved Notebook summary

NB 0.3 has 13 code cells. Let me run them all sequentially:

Ran 0.3-meas08-waveforms-histograms.ipynb • cell 2

Working

Run NB 0.2 Meas08 RMS Time Evolution (4/6)

0.3-meas08-waveforms-histograms.ipynb • Cell 1

Describe what to build

Agent Claude Opus 4.6 - Medium

Terminal:
leszekgrzanka@S272:~/workspace/lgad-proton-beam-monitor-tofs source /home/leszekgrz...
(.venv) leszekgrzanka@S272:~/workspace/lgad-proton-beam-monitor-tofs

AI agent in VSCode: reads large data files, runs notebook cells, and interprets results autonomously — minimal human intervention needed.



```
for ax in axes:
    if ax.get_visible():
        ax.set_xlabel("Time [s]")

fig.suptitle("Meas08 - Per-Segment RMS vs Time (all distances)", fontsize=13, y=1.01)
fig.tight_layout()
fig.savefig(FIG_DIR / "meas08_rms_timeseries.png", dpi=150, bbox_inches="tight")
plt.show()

for idx, pos in enumerate(positions):
    ax = axes_flat[idx]
    for ch in CHANNELS:
        if ch not in all_stats[pos]:
            continue
        rms = all_stats[pos][ch]["rms"]
        time_s = np.arange(len(rms)) * SEGMENT_INTERVAL_S
        ax.plot(time_s, rms, lw=0.3, color=CHANNEL_COLORS[ch], label=ch, alpha=0.7)
        ax.set_title(f"{pos}", fontsize=10)
        ax.set_ylabel("RMS [mV]")
        ax.set_ylim(bottom=0)
        ax.legend(fontsize=7, loc="upper right")
        ax.grid(True, alpha=0.3)
        if xlim is not None:
            ax.set_xlim(xlim)

for idx in range(len(positions), len(axes_flat)):
    axes_flat[idx].set_visible(False) plot_rms_grid(positions_ordered, xlim=(0, 10), suffix=" zoom10s")

# Zoom: first 10 s

for ax in axes_flat:
    if ax.get_visible(): plot_rms_grid(positions_ordered)
    ax.set_xlabel("Time [s]")# Full 100 s range

range_label = f" (first {xlim[1]:.0f} s)" if xlim else ""

fig.suptitle(f"Meas08 - Per-Segment RMS vs Time{range_label}", fontsize=13, y=1.01) plt.show()

fig.tight layout() fig.savefig(FIG_DIR / fname, dpi=150, bbox_inches="tight")
fname = f"meas08_rms_timeseries{suffix}.png"
```

[5] 0.0s Python

Cell In[5], line 35
axes_flat[idx].set_visible(False) plot_rms_grid(positions_ordered, xlim=(0, 10), suffix=" zoom10s")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

CHAT ... LGAD EXPERIMENTS ANALYSIS AND PL... loading cell with parallelization: Ran 0.2-meas08-rms-time-evolution.ipynb • cell 8 Considered alternatives for numba's OpenMP parallelism ProcessPoolExecutor: conflicts with numba's OpenMP. Since numba already parallelizes inner loops (releasing the GIL), ThreadPoolExecutor: is the right choice here: Also update the imports cell to remove the now-unused concurrent.Futures import (we import it inline): Restarting kernel for 0.2-meas08-rms-time-evolution.ipynb Ran 0.2-meas08-rms-time-evolution.ipynb • cell 2 Ran 0.2-meas08-rms-time-evolution.ipynb • cell 4 Ran 0.2-meas08-rms-time-evolution.ipynb • cell 6 Ran 0.2-meas08-rms-time-evolution.ipynb • cell 8 324 s (54 min) vs 492 s before (8.2 min) — -34% speedup from threading. Now run the remaining plotting cells. Ran 0.2-meas08-rms-time-evolution.ipynb • cell 10 Read 0.2-meas08-rms-time-evolution.ipynb, lines 185 to 250 The replacement got mangled again. Let me use edit_notebook_file which handles this properly: > Re-run notebook after changes (2/2) > 1 file changed +0 -0 Keep Undo 0.2-meas08-rms-time-evolution.ipynb • Cell 6 Describe what to build

Agent got confused between in-memory notebook state and file on disk
Jupyter notebook editing by AI agents is not production-ready yet



Experimental work assistant

- In the lab, you need answers *now* – smartphone access is essential
- Multimodal input: photograph equipment: get configuration instructions
- Works best for standard lab instruments with good online documentation

how to limit current to 1uA on this keithley device ?

selected_image_644478559

Completed 3 steps >

To limit the current to 1 μA on the Keithley 6517A, you need to configure the voltage source current compliance (current limit) setting. On this instrument, the voltage source has a programmable current limit that protects your device under test from excessive current. The process typically involves using the front panel menu to set the compliance value. Here are the essential steps:

Setting Current Limit on Keithley 6517A

- Enter the voltage source configuration menu by pressing the **CONFIG** or **V SOURCE** button, depending on the panel options [research.physics.illinois](#).
- Navigate to "Vsource" settings and locate the "Current Limit" or "Compliance" option [research.physics.illinois](#).
- Set the compliance value to 1 μA (1e-6 A).
- Confirm the settings and ensure the voltage source operates with the new limit.

Important Notes

- The lowest programmable current limit is typically in the μA range, but double-check your manual for minimum available settings.
- If using remote control (GPIB/RS232), you can send a compliance command—refer to SCPI command reference in your manual for syntax.
- The current limit (compliance) is a protection feature, not an explicit current source mode—when the load tries to draw more than the set value, the voltage will drop to limit the current [testequipment](#).

Reference

- For detailed menu navigation, see the "Voltage and current limit" and "Voltage source"

on the PCB with analog amplifier board for LGAD sensors I have ADP7158, what is that ?



The ADP7158 on your analog amplifier PCB for LGAD sensors is an ultralow noise, high power supply rejection ratio (PSRR) linear voltage regulator. It operates from 2.3 V to 5.5 V supply and delivers up to 2 A output current, making it suitable for noise-sensitive analog front-ends like those used with LGAD sensors. The device achieves very low output noise (0.9 μV rms from 100 Hz to 100 kHz) and low output voltage ripple using a PMOS pass transistor controlled by an internal error amplifier. This helps ensure stable, clean voltage for sensitive analog components, minimizing interference and improving measurement fidelity. [analog](#) +2

The ADP7158 is not the signal amplifier itself, but rather a precision LDO regulator providing power to the analog signal amplification stages on your board. This choice is driven by the need for low noise and high PSRR when powering analog readout circuitry for LGAD sensors, which demand stable and quiet power for optimal timing and charge resolution. [aisberg.unibg](#) +2

Ideas for you to explore: For LGAD sensor boards and other precision analog hardware, using an LDO regulator like the ADP7158 can significantly improve noise performance and signal integrity. If you are troubleshooting or planning upgrades, review both regulator and amplifier choices—optimizing both will lead to the best analog readout quality for particle timing or charge measurement applications.

Two real questions asked during beamtime at a proton facility
answered in seconds



Paid models in academia

- Industry standard: ~200 PLN/month for coding agents
- Academic procurement problem: individual credit cards, no institutional plans for coding tools
- Open question: Will providers offer academic team licenses for coding assistants?
- Usage limits for coding assistants:
 - flat rate (“Github Copilot Plus”)
 - hourly / weekly limits (“Claude”)

Pricing

Education discounts exist but vary wildly: Perplexity 50% off, Google promotional pricing, Anthropic no academic discount

Frequently asked questions

📄 What cool stuff is included in the trial? ▾

🕒 Why is Google One included in the sign-up process? ▾

❓ What happened to the previous student offer? ▾

🔔 I signed up for the 12-month student offer. What if I forget to cancel? ▸

We'll send you a reminder within a month of your offer period ending, and you can also cancel at any time. **But if you forget to cancel before the offer period ends, you will be charged the monthly rate.**

education pro

50% off

In-depth research and AI model selection

\$9 /month or equivalent, when billed annually

Everything In Free and:

- ✓ Access to Gemini 3.1 Pro, Sonar, Claude Sonnet 4.6, and more
- ✓ Access to the latest AI models, post-trained for higher accuracy
- ✓ Increased document upload limits
- ✓ Better for complex questions and building reports, documents, and apps.
- ✓ Usage limits best for most users

perplexity max

Highest usage and top performance

\$167 /month or equivalent, when billed annually

Everything In Pro and:

- 📄 Automate complex tasks with 10,000 monthly and 35,000 bonus credits for Perplexity Computer
- ✓ Get the best answers with the most advanced AI reasoning models
- ✓ Run deep investigations at any scale
- ✓ Work with massive datasets and files
- ✓ Compare responses across multiple AI models with Perplexity Model Council
- ✓ Priority access to new features

Free

Get everyday help from Google AI to tackle tasks at work, school or home.

\$0 /month with a Google Account

Get started

+

Gemini app
Your personal, proactive and powerful AI Assistant

- ✓ Access to 3 Flash
- ✓ Varying access to 3.1 Pro
- ✓ Image generation and editing
- ✓ Deep Research
- ✓ Gemini Live
- ✓ Canvas

Google AI Plus¹

Get more access to new and powerful features to boost your productivity and creativity.

~~\$7.99~~ /month
\$3.99 /month for 2 months

Get started

🕒 Everything In Free and:

+

Gemini app
Get enhanced access to our most intelligent model 3.1 Pro, Deep Research, image generation with Nano Banana Pro, as well as video creation features with limited access to Veo 3.1 Fast*

- 🕒 **200 monthly AI credits***
Credits used for video generation across Flow and Whisk

Google AI Pro²

Get higher access to new and powerful features to boost your productivity and creativity.

\$19.99 /month

Get started

🕒 Everything In Free and:

+

Gemini app
Get higher access to our most intelligent model 3.1 Pro, Deep Research, and image generation with Nano Banana Pro, plus lock video generation with Veo 3.1 Fast*

- 🕒 **1,000 monthly AI credits***
Credits used for video generation across Flow and Whisk

Google AI Ultra³

Unlock the highest level of access to the best of Google AI and exclusive features.

~~\$249.99~~ /month
\$124.99 /month for 3 months

Get started

🕒 Everything In Google AI Pro and:

+

Gemini app
Highest limits to models and features including video generation with Veo 3.1[†], plus access to Deep Think and Gemini Agent (US only, English only)

- 🕒 **25,000 monthly AI credits***
Credits used for video generation across Flow and Whisk

■ **Flow[†]**



The real cost of AI tools

Billed premium requests
\$0.00
Increase your budget to use premium requests beyond your included request limit.

Included premium requests consumed
571.8 of 1,500 included
Premium requests included in your [Credits plan](#). Monthly limit resets in 21 days on May 1, 2026.

Usage breakdown
Usage for Apr 1 - Apr 30, 2026. Price per premium request is \$0.04.

Model	Included requests	Billed requests	Gross amount	Billed amount
Claude Opus 4.6	483	0	\$19.32	\$0.00
Auto: GPT-5.3-Codex	31.50	0	\$1.26	\$0.00
Code Review model	27	0	\$1.08	\$0.00
Claude Sonnet 4.6	12	0	\$0.48	\$0.00
GPT-5.2	7	0	\$0.28	\$0.00
Auto: GPT-5.4	6.30	0	\$0.25	\$0.00
Auto: Claude Sonnet 4.6	2.70	0	\$0.11	\$0.00
Coding Agent model	2	0	\$0.08	\$0.00
Auto: Claude Haiku 4.5	0.30	0	\$0.01	\$0.00
Claude Haiku 4.5	0	0	\$0.00	\$0.00
Gemini 3 Flash	0	0	\$0.00	\$0.00

Plan usage limits Pro

Current session
Resets in 1 hr 37 min 96% used

Weekly limits
[Learn more about usage limits](#)

All models
Resets Thu 8:00 PM 15% used

Last updated: less than a minute ago ↻

Extra usage
Turn on extra usage to keep using Claude if you hit a limit. [Learn more](#)

€0.00 spent
Resets May 1 0% used

Re: [EXT] [SheerID] Re: Troubles with verification for perplexity.ai

Dear Kailash,

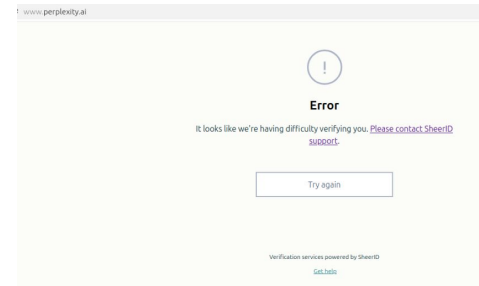
I am writing regarding my application for the Education Pro plan (perplexity).

During the verification process, I was not prompted to upload any documents, as shown in the two attached screenshots. To facilitate the verification, I have attached a copy of my academic teacher card from AGH University in Krakow.

For further confirmation, you can view my public profile on the AGH University website: <https://skos.agh.edu.pl/osoba/leszek-grzanka-8485.html>

Please let me know if this information is sufficient to complete the verification process.

Kind regards,
Leszek Grzanka



Claude Pro API usage, plan limits, and the pain of academic verification.



What Cyfronet offers today ?

LokAI Chat

ChatGPT-like portal running entirely on Cyfronet infrastructure, no data leaves Poland.
Upload documents, use templates, work with images

OpenAI-compatible API (LLMForge)

integrate AI models into your own scripts and pipelines

GPU resources

run your own models, fine-tune, or train from scratch on Cyfronet's HPC infrastructure

Data stays here

all three options keep your data within Cyfronet, unlike commercial cloud services

What is your experience ?

Privacy

Literature research

AI as writing assistant

Data analysis

Experimental work

Paid vs free

Daily life usage...

