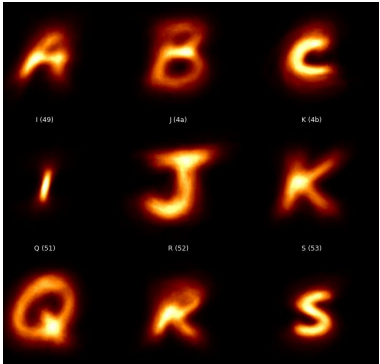


1st Annual Jointlab AI Day

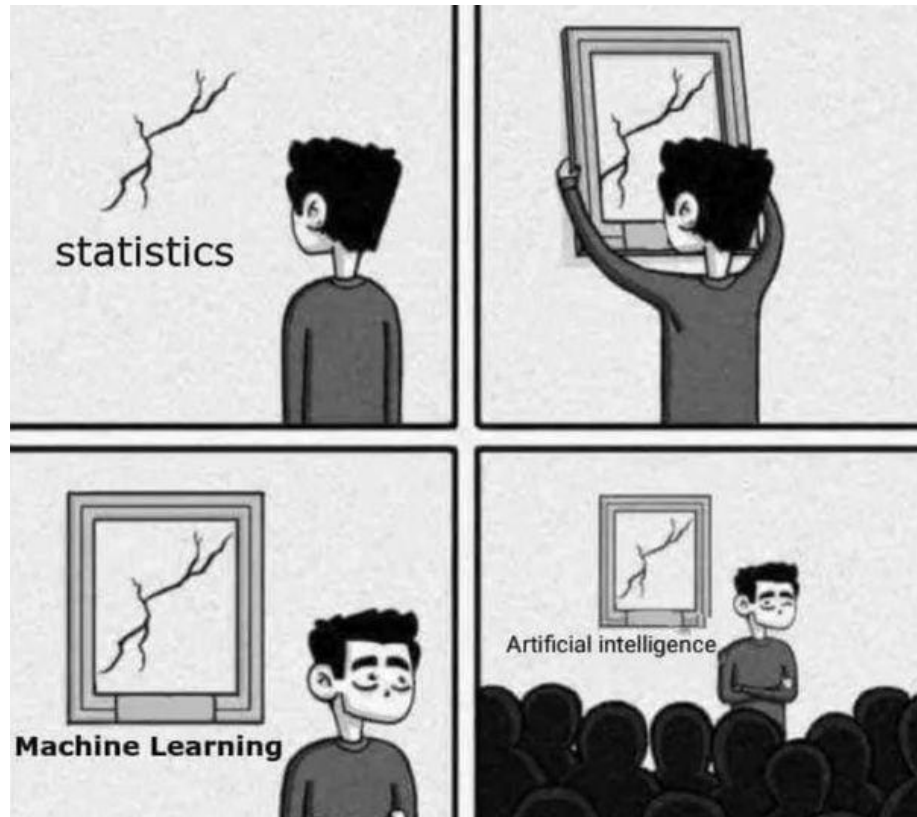
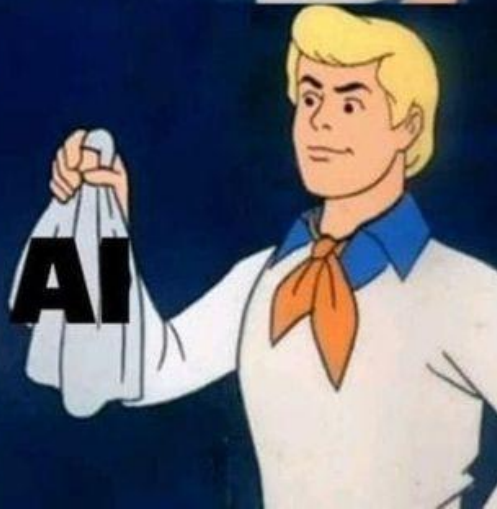
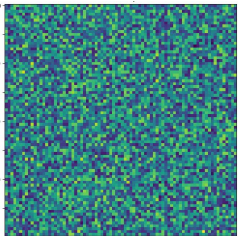
8th June 2026
Jiří Kvita





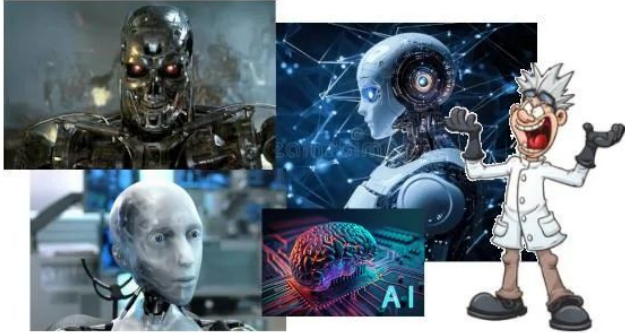
Matrix Multiplication

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} \end{bmatrix}$$



Somewhere from the internet

How people think AI works



How it really works

$$\begin{bmatrix} A & B \\ C & D \\ E & F \end{bmatrix} \times \begin{bmatrix} G \\ H \end{bmatrix} = \begin{bmatrix} A \times G + B \times H \\ C \times G + D \times H \\ E \times G + F \times H \end{bmatrix}$$

lol lets multiply numbers really fast



forloop

@forloopcodes

i still can't believe this is your virtual girlfriend

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{bmatrix}$$

5:46 PM · July 25, 2025

Somewhere from the internet

Some personal memorabilia

- Student at Fermilab 24y ago.
- 1st thing to learn: a bit of C++ and ROOT.
- Many tasks depended on how well and fast physicists could code.
 - (potentially) good physicists can be bad coders and vice versa, or both.
- Have we learned enough physics?
 - I hope so
 - I've been taking coding as part of the adventure
 - but is it the same these days?
 - aren't some codes too complex? (ATLAS SW)
- Have we learned too much IT?
 - depends who you ask:)
- In last years, experiments multiplied in size, number of channels read, in number of people involved.
- One depends on working groups focusing on subtasks and delivering object ID etc. to the whole collaboration.
- One can/needs to focus on either IT, physics, statistics, calibration, final plots...
- One can easily get lost. Where is one's place in all this?

Paradigms, languages and frameworks shift, evolve

- C++ to Python, Rust...
 - From books to googling to StackOverflow
 - From event-by-event processing to data frames.
 - From personal coding to AI-assisted coding.
-
- But these are just tools that should help us explore.

Intro

- Prehistoric days:
 - manuals, definitions of formats, C++ books, online references etc.
 - private and actual learning of IT languages, TeX, ...
- Old days: StackOverflow, google
 - tedious combining information, often erroneous
- Early LLM days
 - Code snippets from ChatGPT browser.
 - Manual ctrl+C ctrl+V, debugging, checking, finding bugs and mistakes
 - Example: read a csv file in python and C++
- Currently
 - coding agents (copilot etc)
 - “vibe coding”
- Next...?

What do we do as physicists?

- Design an experiment.
- Analyze the data.
- Interpret the data.
- Simulate and compare.
- Describe by a theory.
- Predict other phenomena.
- Fail, Succeed, Get new ideas && Repeat.
- Oh, and learn and teach and supervise:)
- And cooperate.

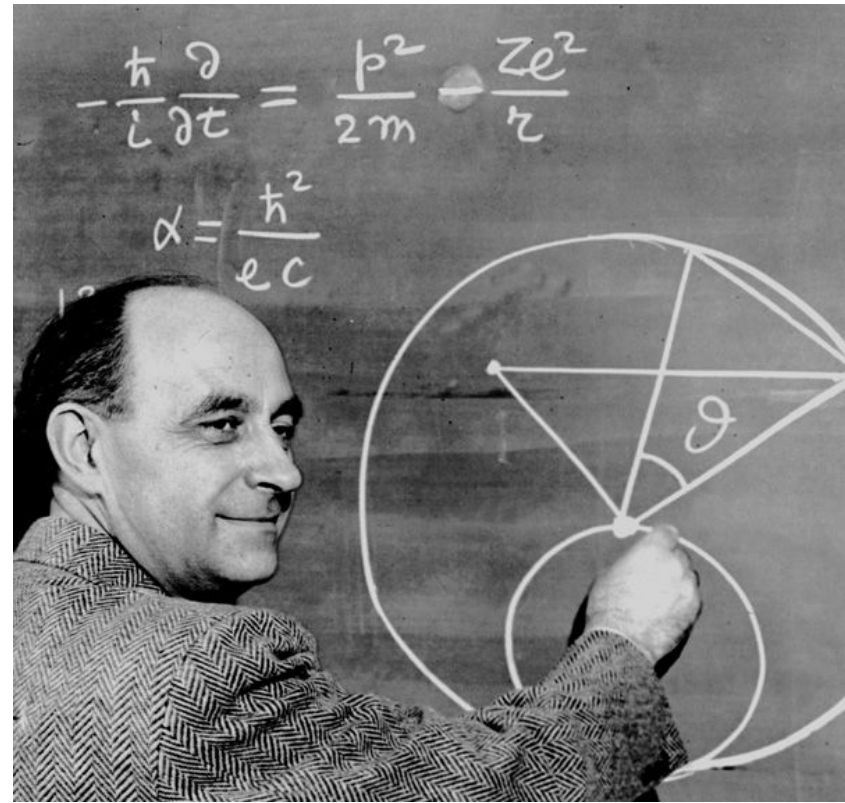


ChatGPT Image 6. 6. 2026 18_20_06.png

Mistake in the equation

What do we do as physicists?

- Design an experiment.
- Analyze the data.
- Interpret the data.
- Simulate and compare.
- Describe by a theory.
- Predict other phenomena.
- Fail, Succeed, Get new ideas && Repeat.
- Oh, and learn and teach and supervise:)
- And cooperate.



Somewhere from the internet

Also mistake in the definition

What do we do as physicists?

- Design an experiment.
- Analyze the data.
- Interpret the data.
- Simulate and compare.
- Describe by a theory.
- Predict other phenomena.
- Fail, Succeed, Get new ideas && Repeat.
- Oh, and learn and teach and supervise:)
- And cooperate.

<https://www.catrin.com/cs/media/vizualni-identita-ke-stazeni/>



Or do we just...?

- Hunt for a high H-index
- Obtain grants
- Present our work as “rewriting textbooks” or “possibly solve/change/cure...”
- Convince public and politicians to support us
- Publish or perish
 - argue with journals
 - pay journals with tax-payers’ money
- Play the academia game with titles.
- Cherry-pick from the data?

What tools do we need?

- ...and I'll talk mostly about coding
- but are we really just IT people?
- many problems have been solved before.
- should we think and spend our time more in coding or physics?
- too much (personal) attachments to taste not needed any more?
- or healthy anchors to roots of the coding?

What can we ask for?



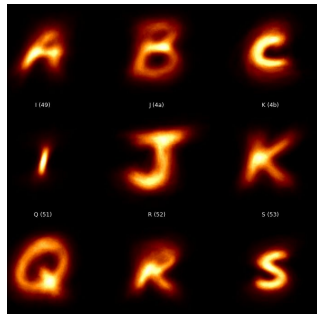
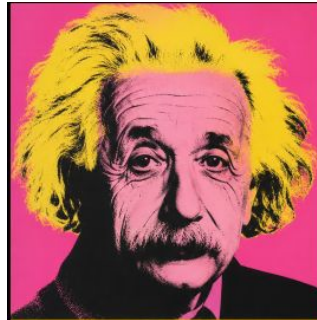
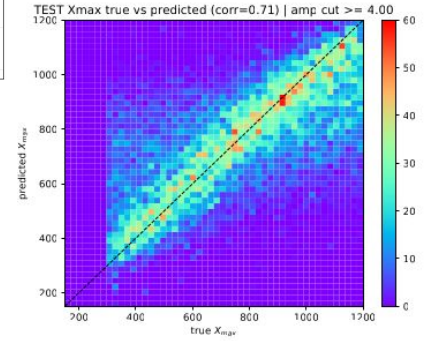
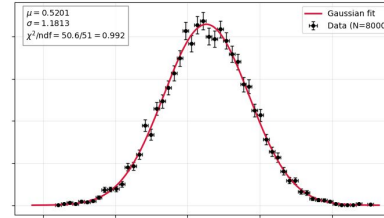
ChatGPT Image 6. 6. 2026 18_33_22.png

What can we ask for?

- Solution to a technical problem
 - e.g. how to read particular data format, database...
 - how to fit and plot residuals
 - web design, backend
- Analyze, simplify, check and fool-proof an existing program
 - e.g. check against nan types in plotting or computation
 - speed-up IO, modularize a single long source code etc
- Write a whole utility e.g. to read-fit-output
 - including intermediate plots etc
 - plot data, model
 - export the model
- Come up with what to do?:)

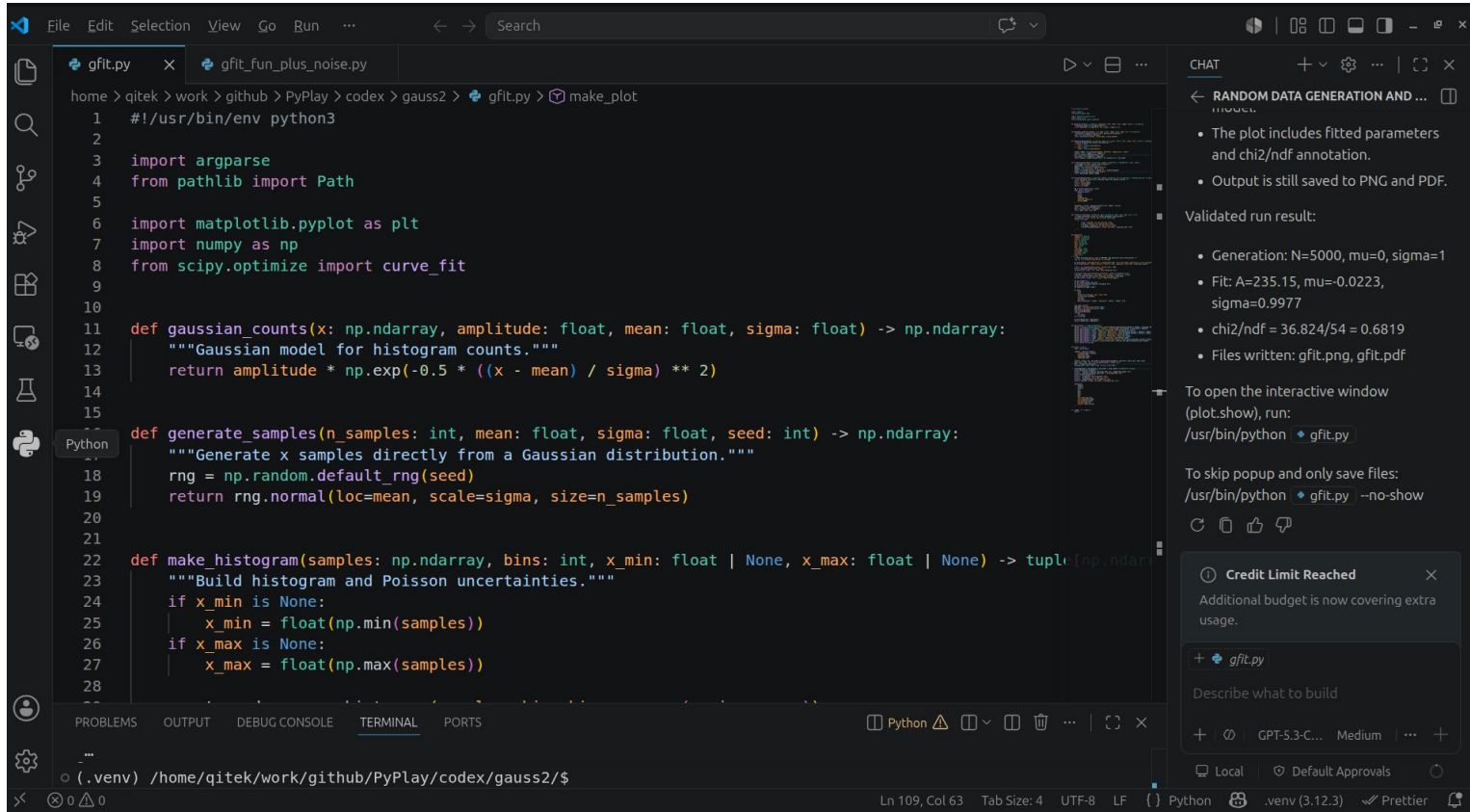
Examples

- Regression
 - obtain model parameters from data
 - fit, ML
- Correct data
 - for resolution: unfolding
- Model the data
 - design simulation, augment synthetic data to match the real
- Classification
 - select signal over background
- Pattern recognition
 - find new features, populations



An interactive example

- Coding powers of LLM coding agents.
- Github copilot integrated as a “chat agent” in Visual Studio Code, Linux.



The screenshot displays the Visual Studio Code interface with a Python script named `gfit.py` open. The script defines functions for generating Gaussian samples and fitting a histogram. The chat window on the right shows a conversation with the AI agent, discussing the generation of random data and the results of a fit.

```
home > qitek > work > github > PyPlay > codex > gauss2 > gfit.py > make_plot
1  #!/usr/bin/env python3
2
3  import argparse
4  from pathlib import Path
5
6  import matplotlib.pyplot as plt
7  import numpy as np
8  from scipy.optimize import curve_fit
9
10
11 def gaussian_counts(x: np.ndarray, amplitude: float, mean: float, sigma: float) -> np.ndarray:
12     """Gaussian model for histogram counts."""
13     return amplitude * np.exp(-0.5 * ((x - mean) / sigma) ** 2)
14
15
16 def generate_samples(n_samples: int, mean: float, sigma: float, seed: int) -> np.ndarray:
17     """Generate x samples directly from a Gaussian distribution."""
18     rng = np.random.default_rng(seed)
19     return rng.normal(loc=mean, scale=sigma, size=n_samples)
20
21
22 def make_histogram(samples: np.ndarray, bins: int, x_min: float | None, x_max: float | None) -> tuple[np.ndarray, np.ndarray]:
23     """Build histogram and Poisson uncertainties."""
24     if x_min is None:
25         x_min = float(np.min(samples))
26     if x_max is None:
27         x_max = float(np.max(samples))
28
```

CHAT

← **RANDOM DATA GENERATION AND ...**

- The plot includes fitted parameters and chi2/ndf annotation.
- Output is still saved to PNG and PDF.

Validated run result:

- Generation: N=5000, mu=0, sigma=1
- Fit: A=235.15, mu=-0.0223, sigma=0.9977
- chi2/ndf = 36.824/54 = 0.6819
- Files written: gfit.png, gfit.pdf

To open the interactive window (plot.show), run:
`/usr/bin/python gfit.py`

To skip popup and only save files:
`/usr/bin/python gfit.py --no-show`

Credit Limit Reached

Additional budget is now covering extra usage.

+ `gfit.py`

Describe what to build

+ GPT-5.3-C... Medium ...


Local | Default: Approvals

Python | .venv (3.12.3) | Prettier

An interactive example

- Coding powers of LLM coding agents.
- Github copilot integrated as a “chat agent” in Visual Studio Code, Linux.



GitHub Copilot

 **GitHub Copilot Pro is active for your account**

You currently have an active [Copilot Pro subscription](#).
Get started by installing the extension in your preferred environment.

[Copilot in your IDE](#) [Copilot CLI](#) [Chat in GitHub Mobile](#) [More features](#)

Usage [View details >](#)

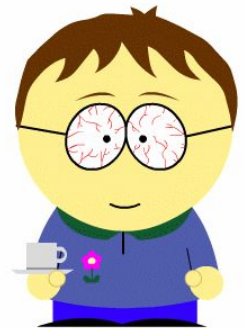
Included credits ⓘ Resets in 25 days on Jul 1, 2026	1,500 / 1,500 AI credits 
Additional usage ⓘ Enabled	\$1.85 / \$10 budget  Edit budget

ⓘ You're now using additional usage until your included credits reset on Jul 1.

An interactive example

- Coding powers of LLM coding agents.
- Github copilot integrated as a “chat agent” in Visual Studio Code, Linux.

Date	Amount	Status	Receipt	Invoice
2026-05-16	\$10.00	Success	↓	↓
2026-04-15	\$10.00	Success	↓	↓



An interactive example

- Coding powers of LLM coding agents
 - generate & fit gauss using Python's matplotlib
 - “generate events \sim gauss, fit with gauss, and plot the data and fit, χ^2/ndf and plot also in lower panel the fit residuals”
 - “make a GUI which calls this toy, with green Run button, and `n_events` and gauss parameter input boxes. Also add red Exit button”
 - “add option to just silently and without plotting return the fitted parameters and χ^2/ndf ”
 - “Add SuperToy button and NToys to call the functionality that the generation and fit is called NToys=100 times, and histograms of results are plotted, namely the fitted amplitude, mean, sigma, and also χ^2 , into a 2x2 grid plot”
 - “do not use, plot nor fit histogram points with zero content”
 - same as html using plotly
 - save png, pdf
 - rewrite all this to
 - pyroot
 - C++ ROOT

An interactive example

Gaussian Toy Fit

Gaussian Toy + Fit

n_events: 8000

gauss mean: 0.5

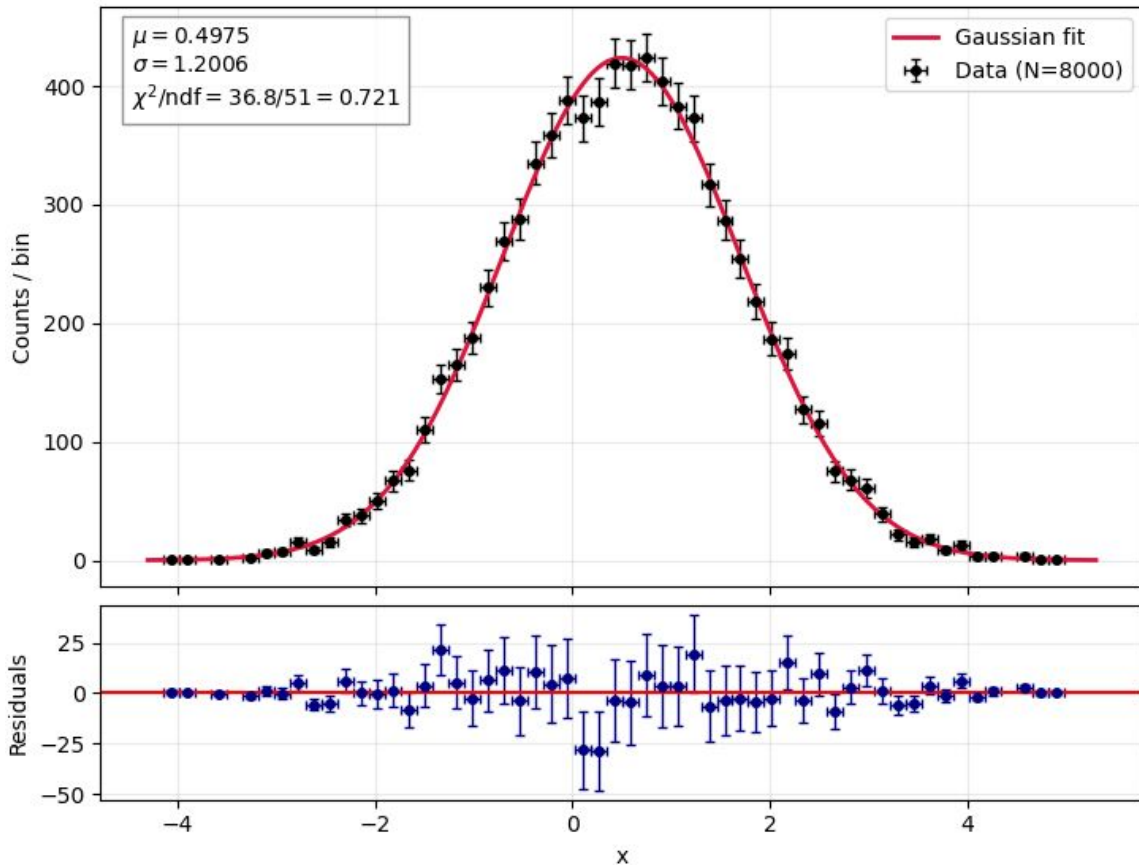
gauss sigma: 1.2

NToys: 100

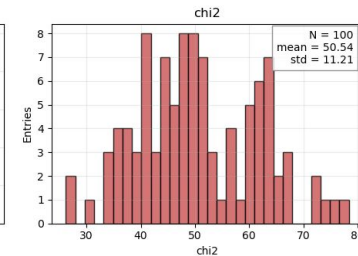
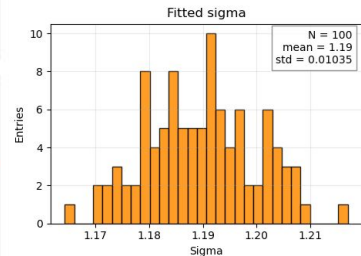
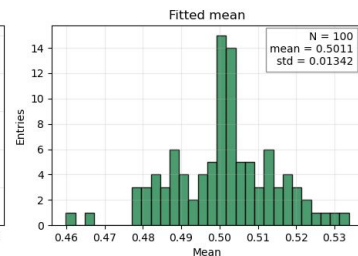
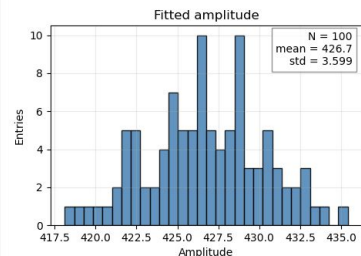
Silent fit (no plot/save)

Run SuperToy

Exit



SuperToy summary: requested=100, fitted=100, failed=0

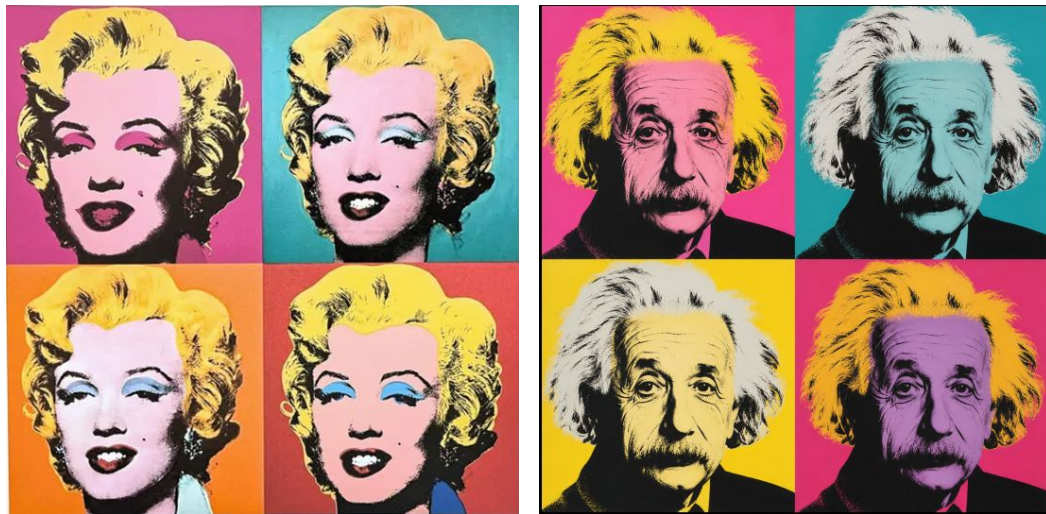


Other examples

- FFT2D of images
- A launcher GUI
- csv reading
- plot, profile, ASCII, plot graph
- Arduino code for PWM for a diode and a potentiometer on some pins.
- RAW images stacking
- Words and bigrams frequencies in Bible

FFT play

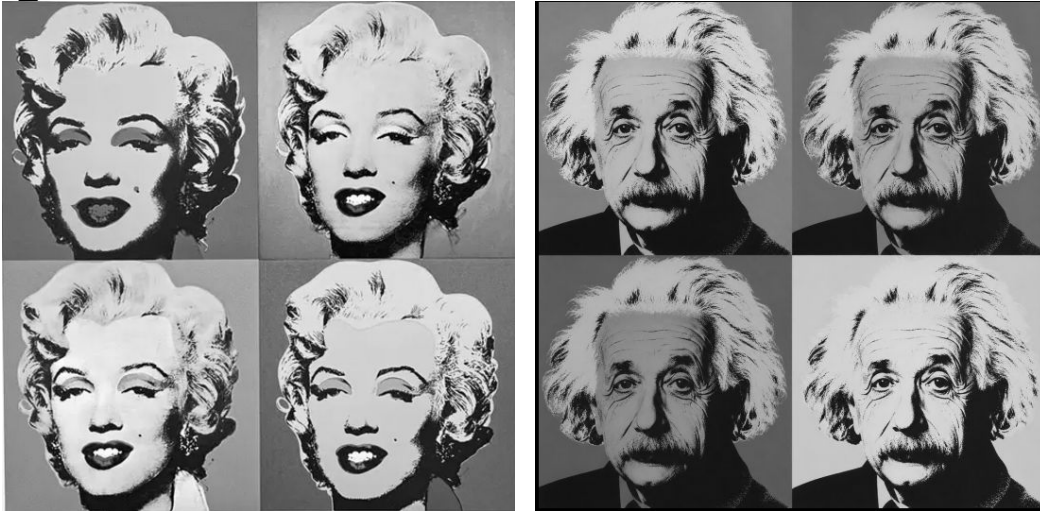
- Many areas to play with and explore open!
- Say this 2D FFT, inputs:



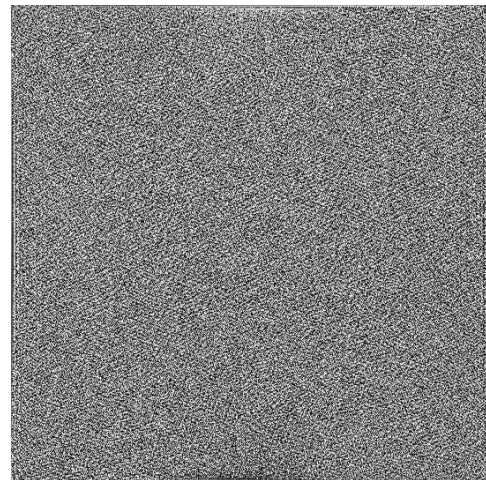
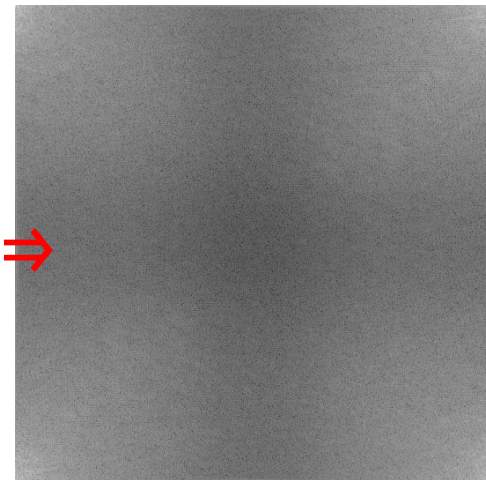
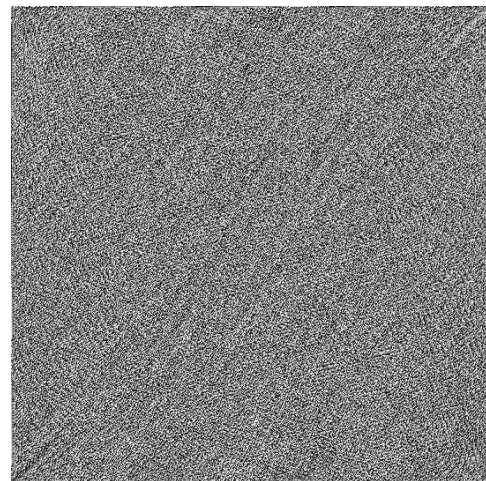
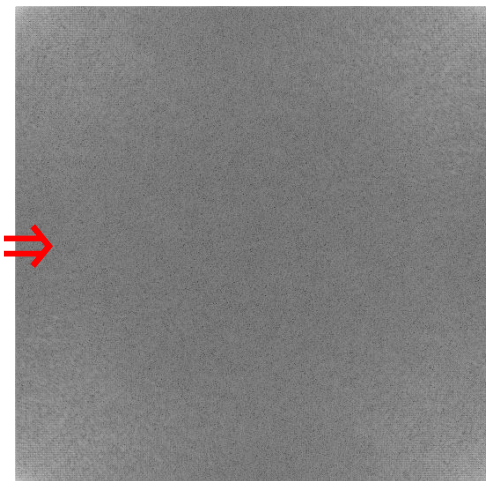
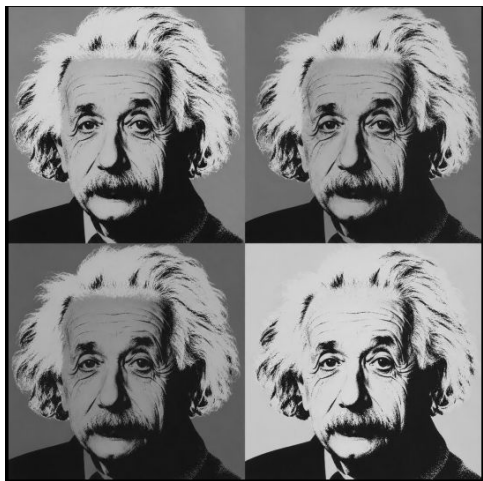
- M. Monroe <https://www.etsy.com/de-en/listing/1727858916/andy-warhol-marilyn-monroe-signed>
- GPT-5.3 5th May 2026: “a picture of albert einstein in the style of warhole’s marylin monroe”

Inputs

- ...in greyscale
 - Q: What is the difference between grey and gray?
 - A: Simple: grey is a colour while gray is a color.



FFT



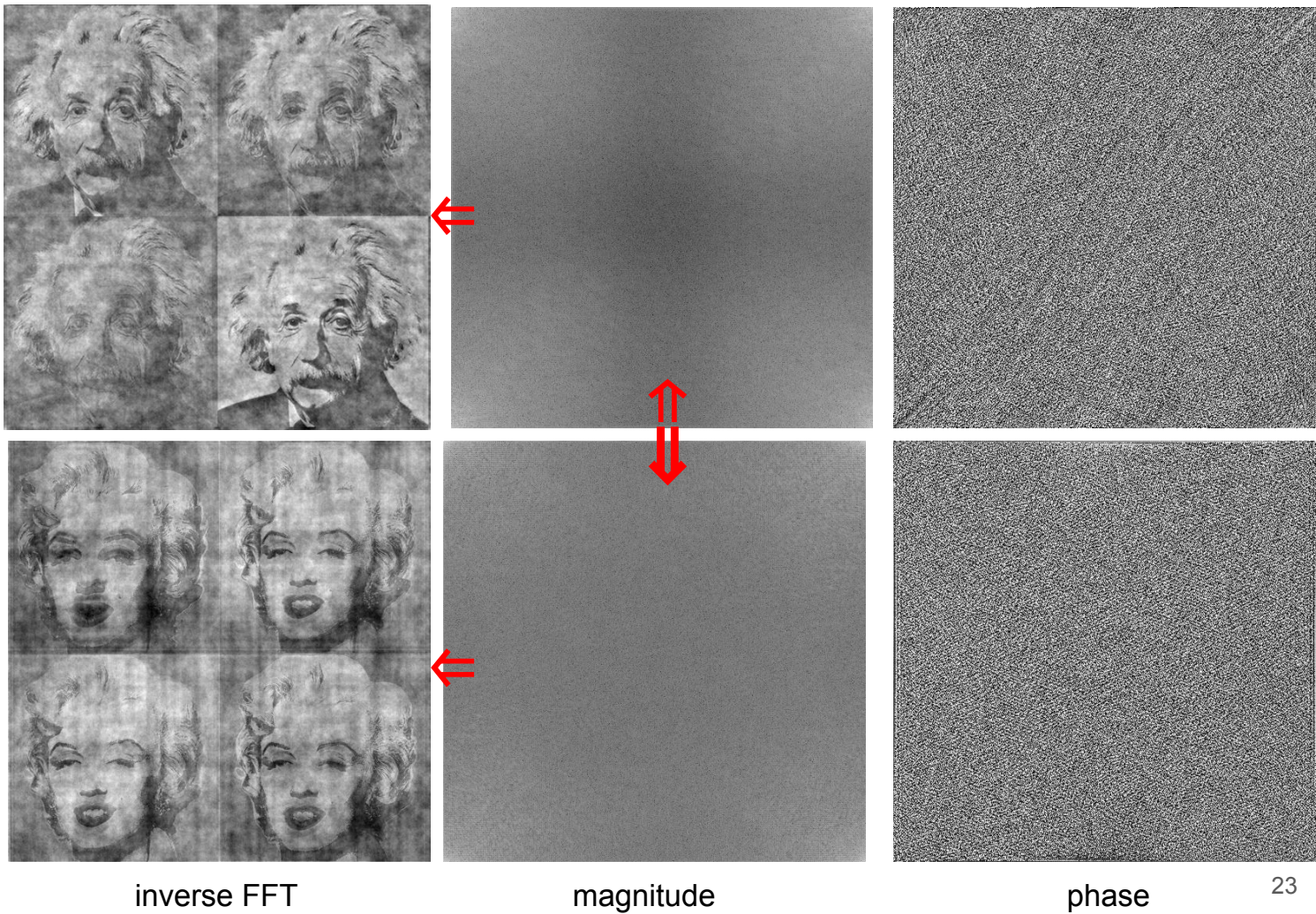
input

magnitude

phase

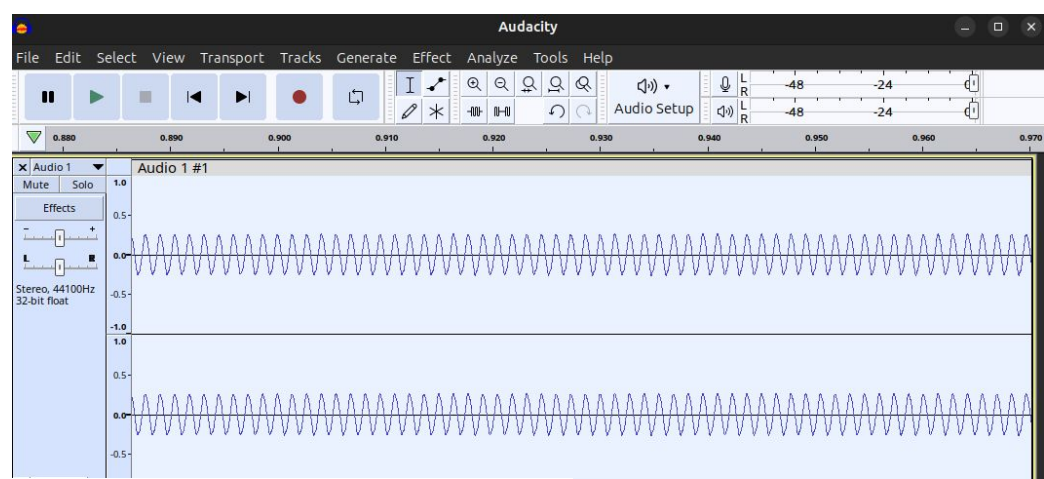
FFT⁻¹

- Swapped phase and magnitude.
- Phase seems to define more features of the image.

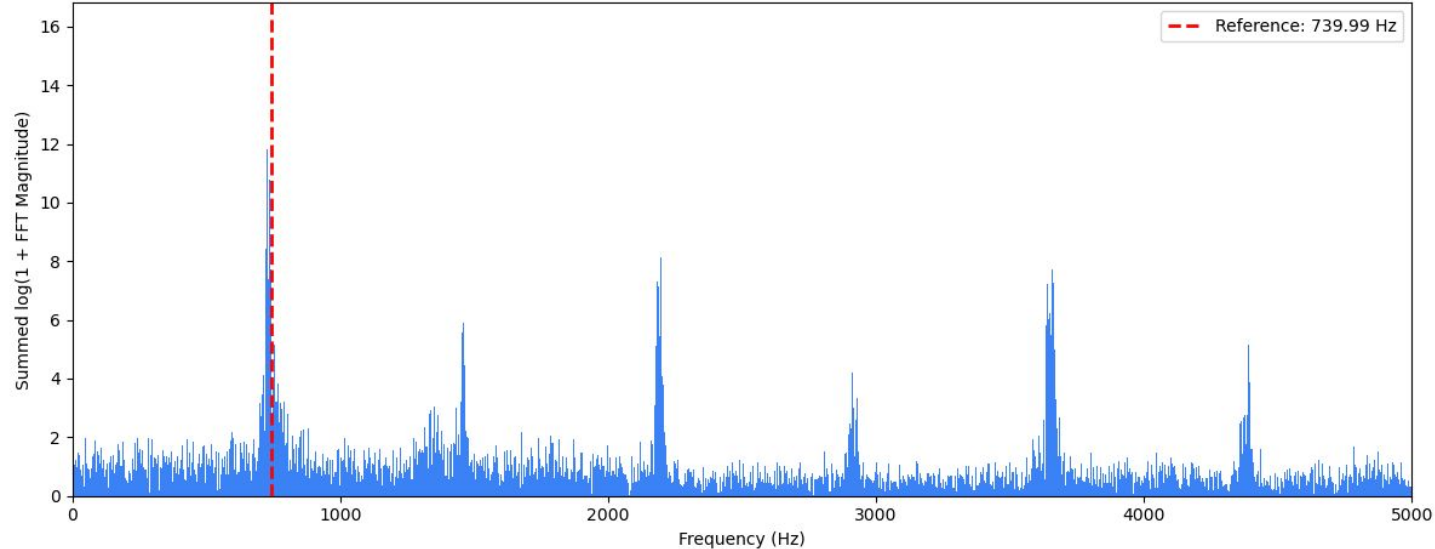


FFT1

- Recorder 'flute' f#
- theoretical frequency 740 Hz

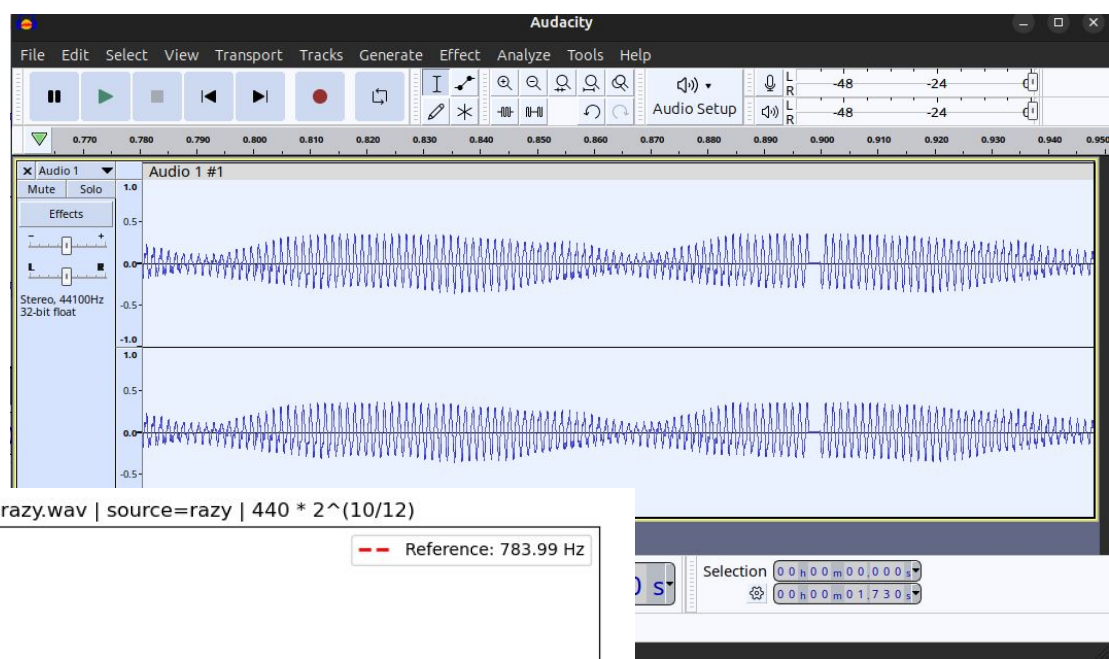


1 Hz Frequency Histogram (log(1 + FFT Magnitude))\nfis.wav | source=fis | 440 * 2^(9/12)

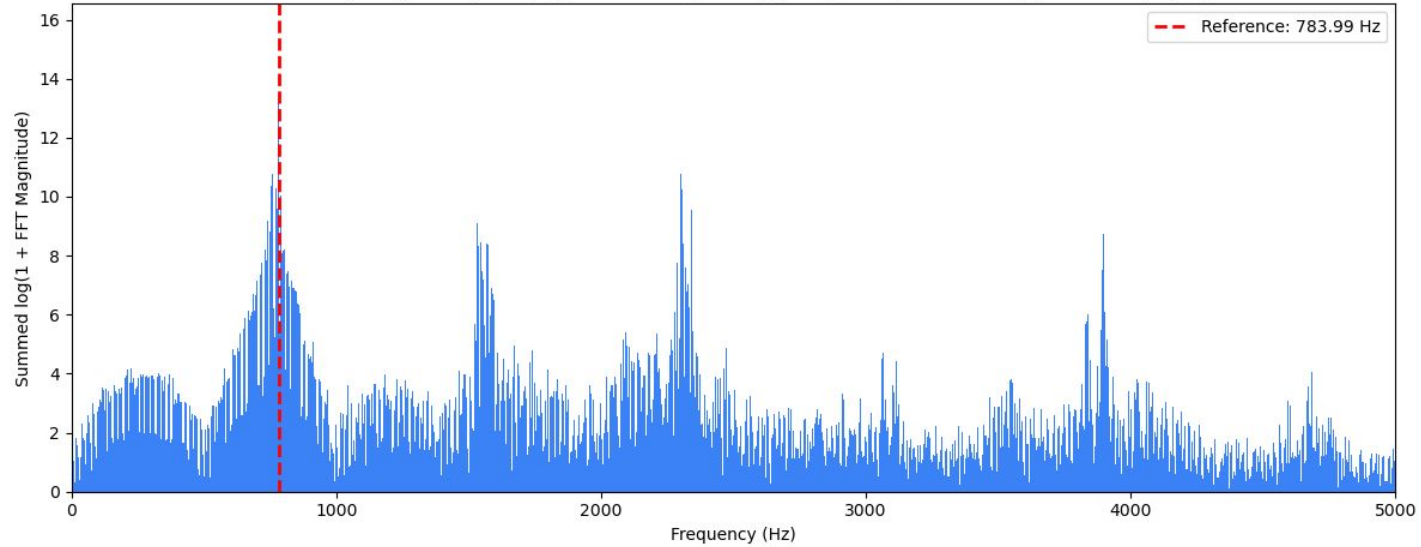


FFT1

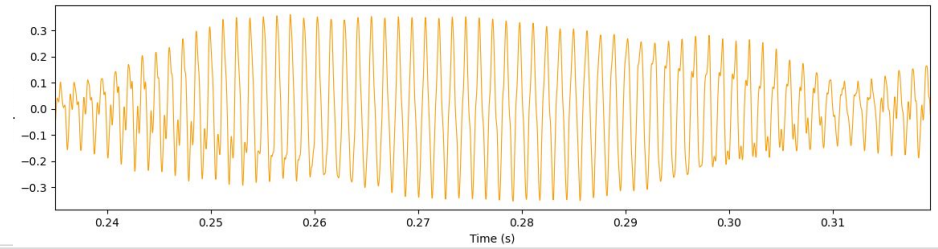
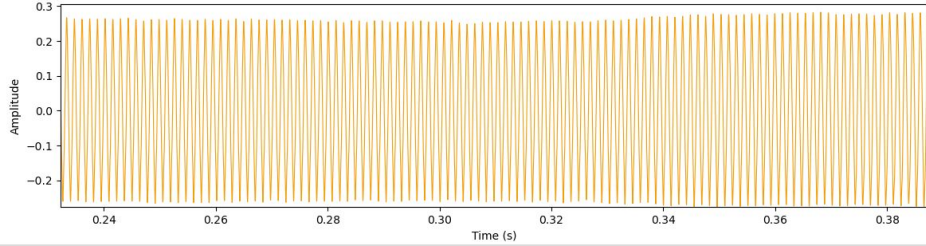
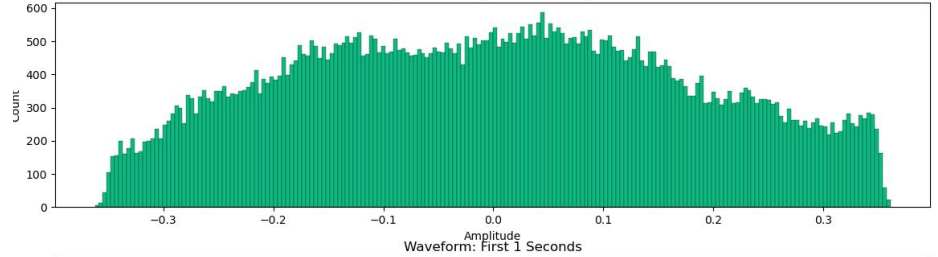
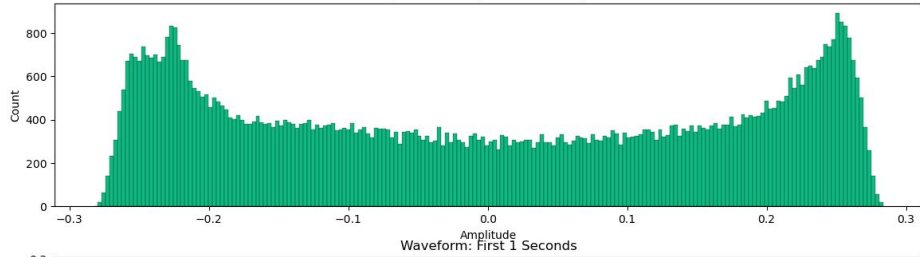
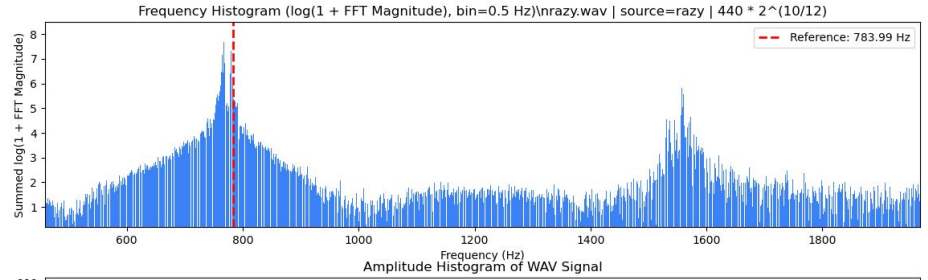
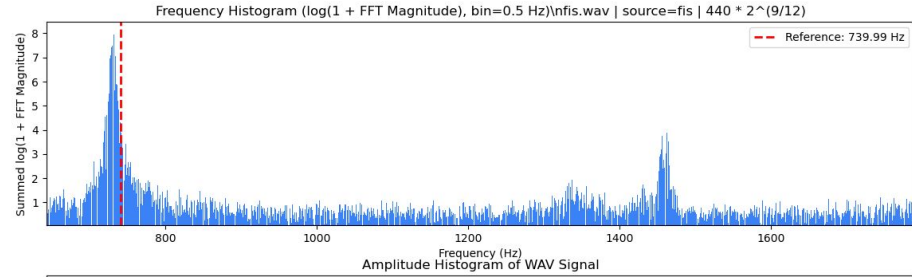
- Beats of 2 close g's
- theoretical $f=784$ Hz



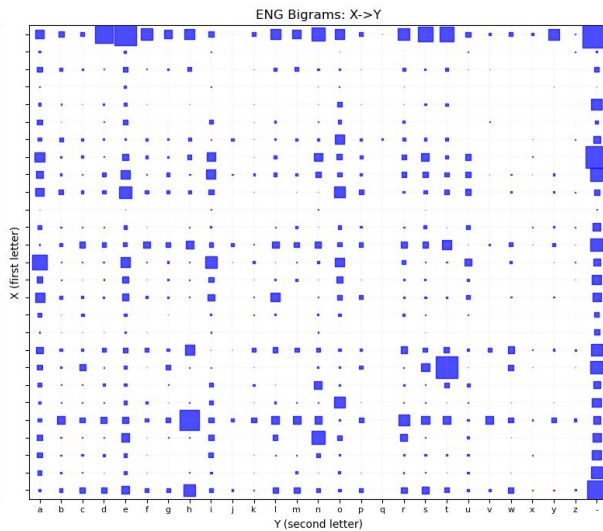
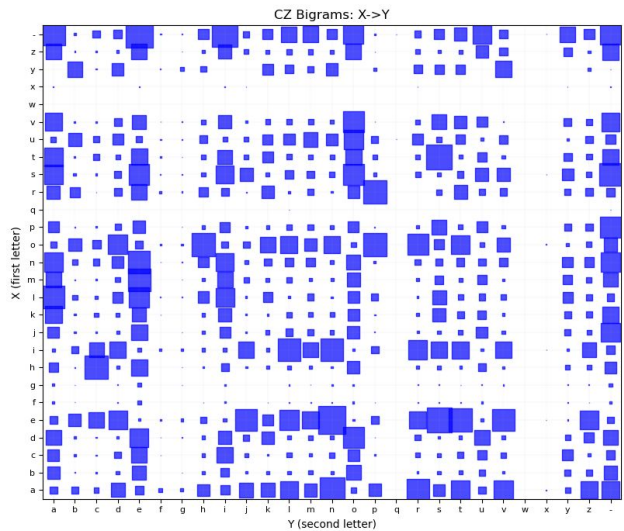
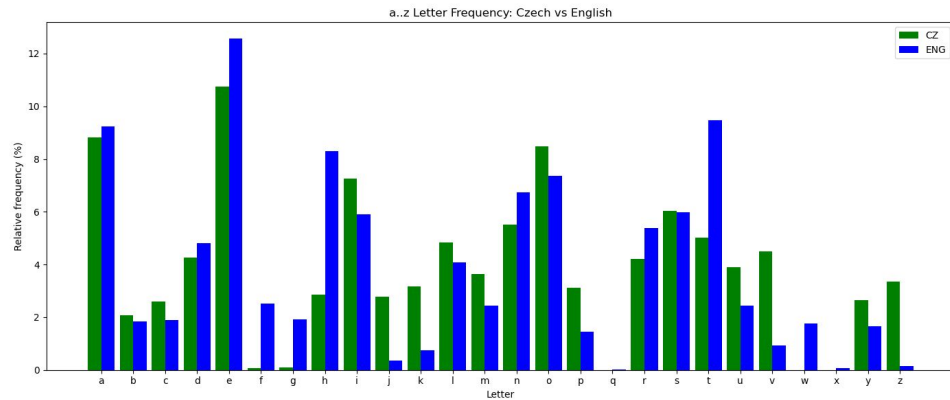
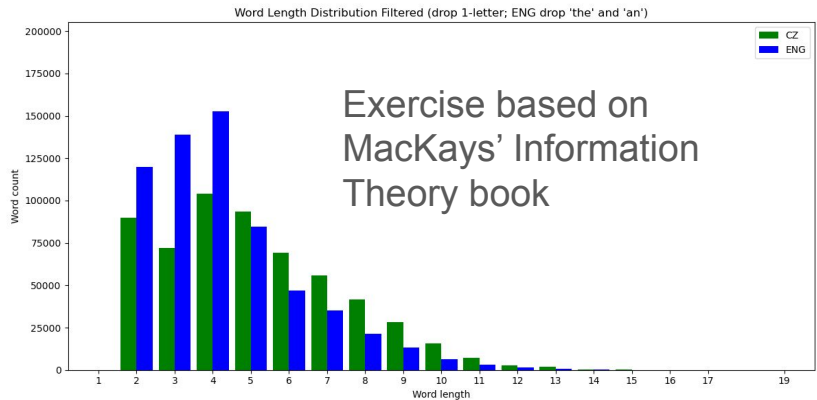
1 Hz Frequency Histogram ($\log(1 + \text{FFT Magnitude})$)\nrazy.wav | source=razy | $440 * 2^{(10/12)}$



FFT1



Letters and bigrams in English and Czech Old Testament



THE OLD TESTAMENT

TRANSLATED OUT OF THE ORIGINAL TONGUES: AND WITH THE FORMER TRANSLATIONS DILIGENTLY COMPARED AND REVISED, BY HIS MAJESTY'S SPECIAL COMMAND

BIBLE
PÍSMO SVATÉ STARÉHO A NOVÉHO ZÁKONA
Ekumenický překlad

STARÝ ZÁKON

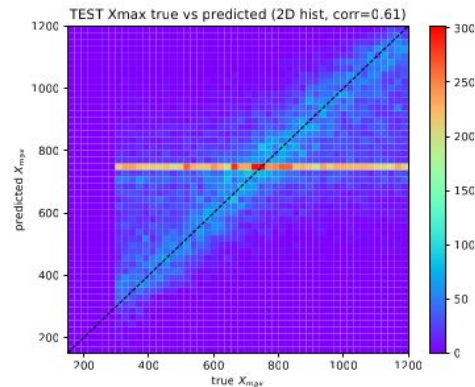
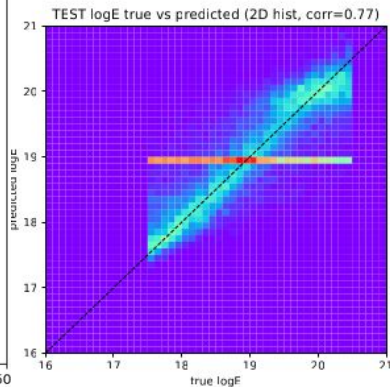
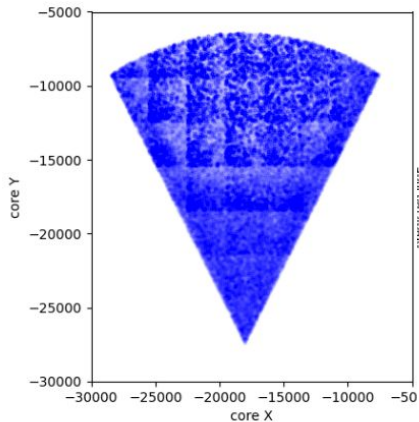
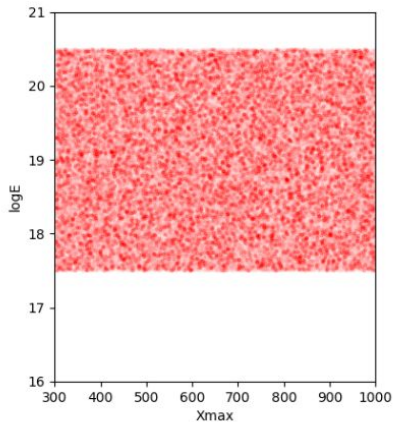
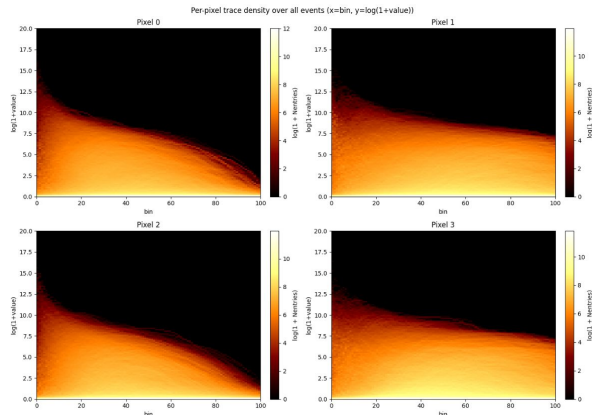
GENESIS

GENESIS 1

¹ Na počátku stvořil Bůh nebe a zemi. ² Země byla pustá a prázdná a nad propastnou tůňí byla tma. Ale nad vodami vznášel se duch Boží. ³ I řekl Bůh: "Buď světlo!" A bylo světlo. ⁴ Viděl, že světlo je dobré, a oddělil světlo od tmy. ⁵ Světlo nazval Bůh dnem a tmou nazval nocí. Byl večer a bylo jitro, den první. ⁶ I řekl Bůh:

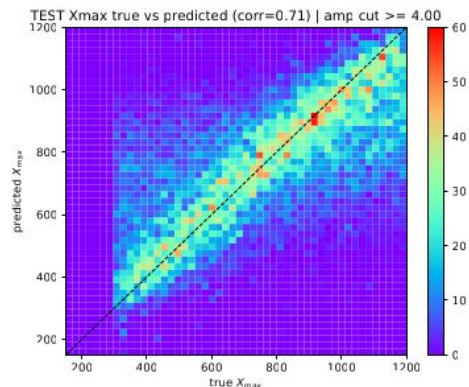
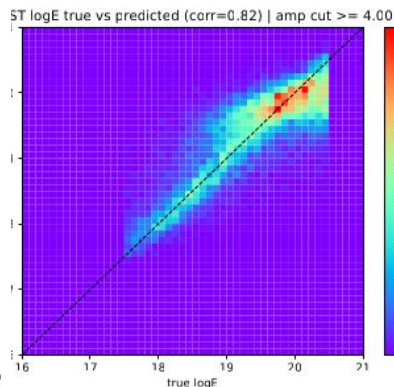
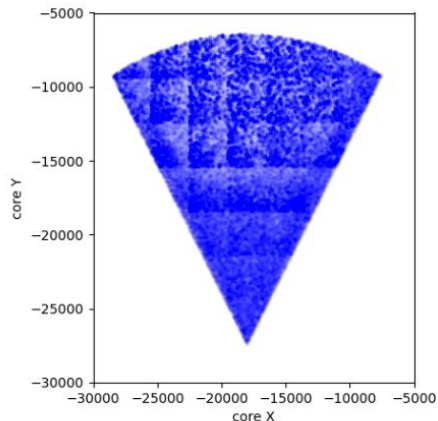
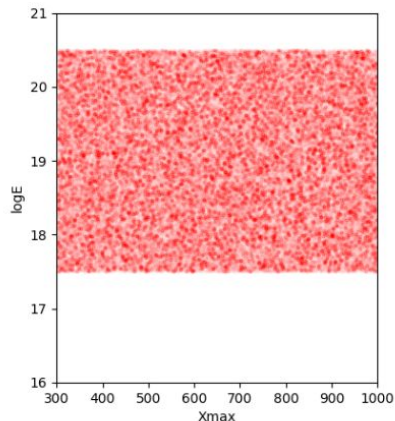
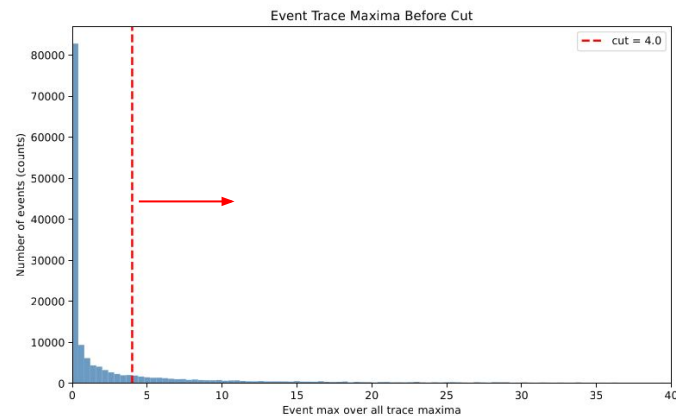
Regression

- FAST simulated cosmic air shower transients in its 4 PMTs.
- Infer 2 shower parameters $\log E$ and X_{\max} .
- Issues in correlations between predicted and truth values.



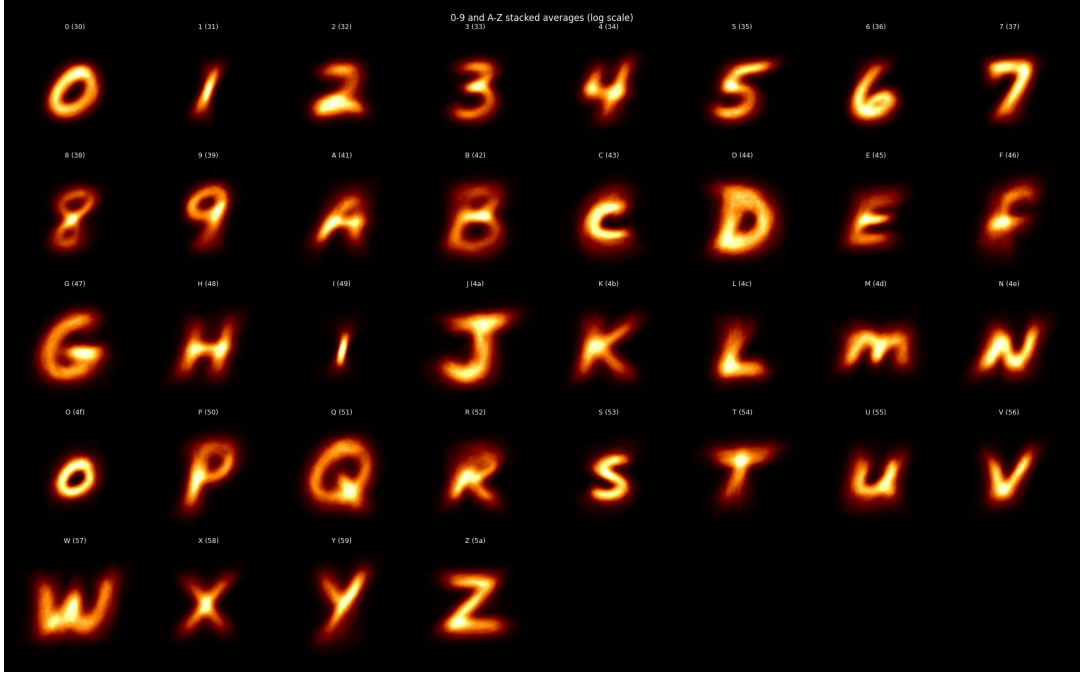
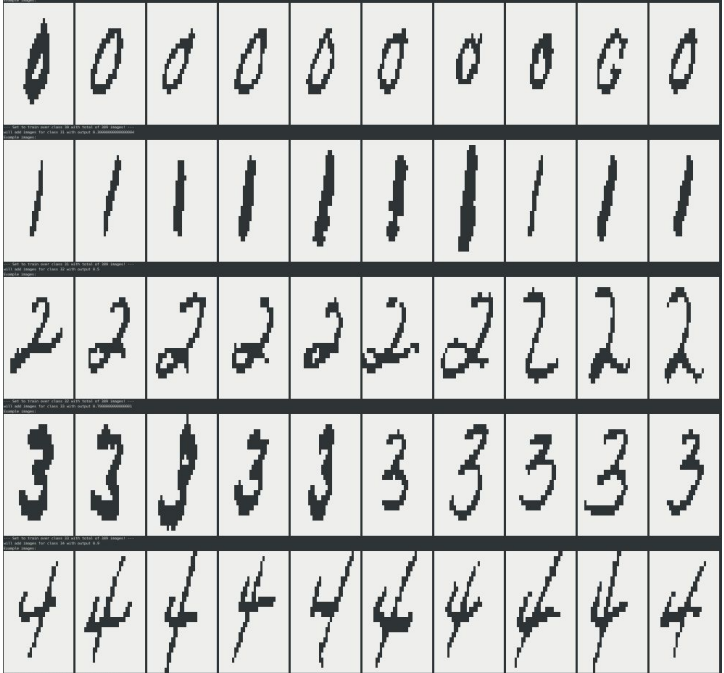
Regression

- FAST simulated cosmic air shower transients in its 4 PMTs.
- Infer 2 shower parameters $\log E$ and X_{\max}
- Resolved after filtering out faint signal events (Jakub, Rosťa, Honza Tomeček).



Classification

- <https://www.nist.gov/srd/nist-special-database-19>
- https://en.wikipedia.org/wiki/MNIST_database
- Individual examples, and averages over 500 characters, as $\log(1+\text{intensity})$.

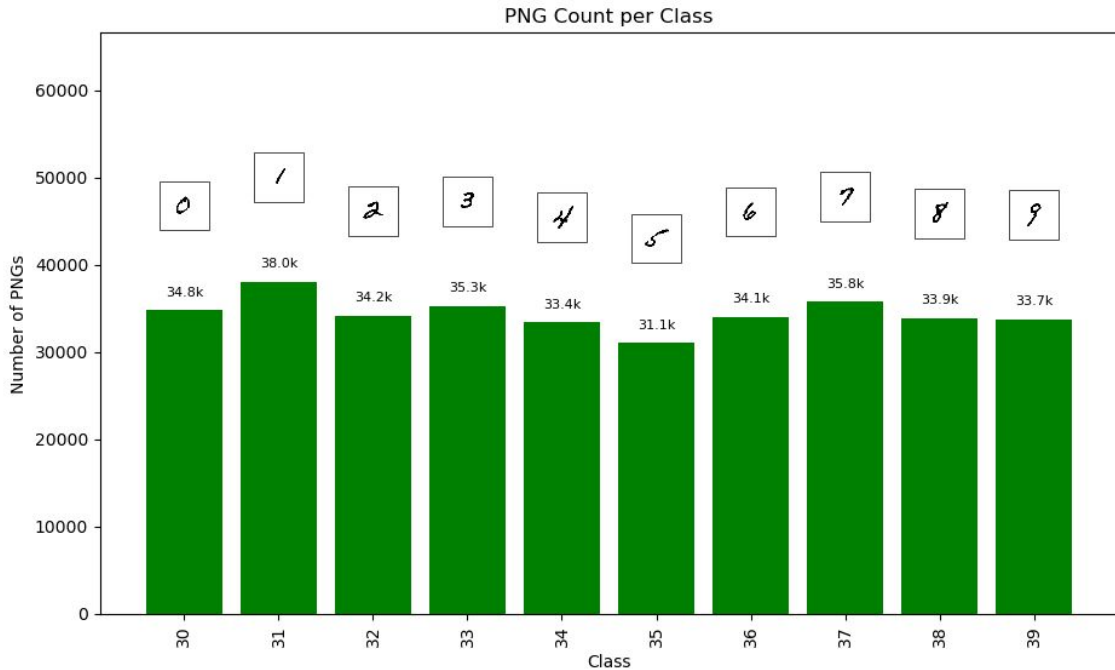


Character counts

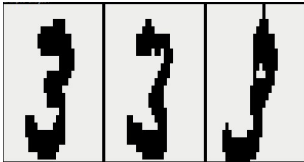
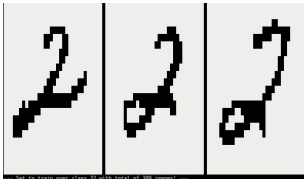
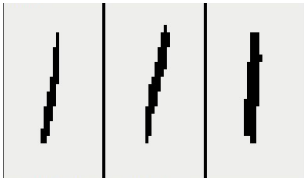
- <https://www.nist.gov/srd/nist-special-database-19>
- https://en.wikipedia.org/wiki/MNIST_database

Class Class Table: Name Count | First Image First Image

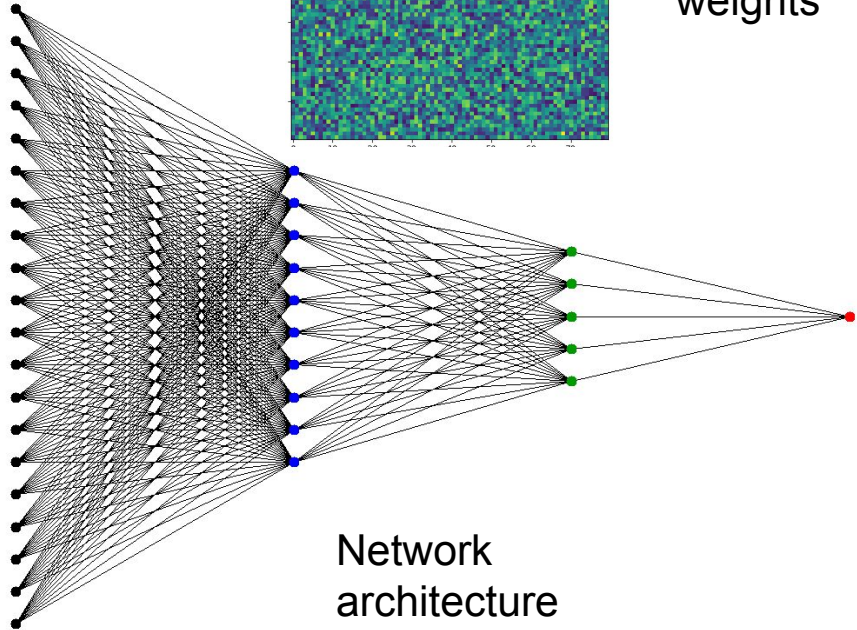
30	34.8k	0
31	38.0k	1
32	34.2k	2
33	35.3k	3
34	33.4k	4
35	31.1k	5
36	34.1k	6
37	35.8k	7
38	33.9k	8
39	33.7k	9



NN Principles Overview

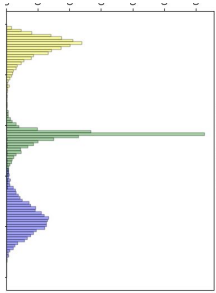


Inputs



Network architecture

Model weights



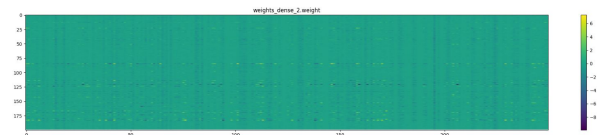
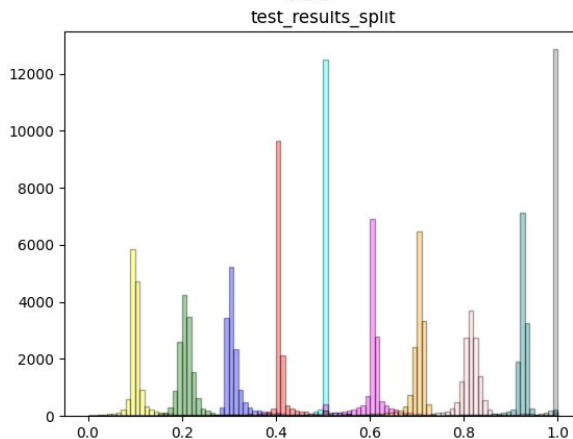
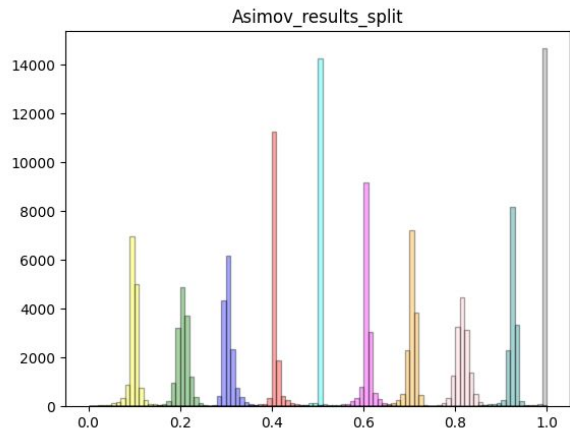
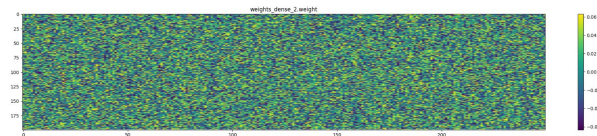
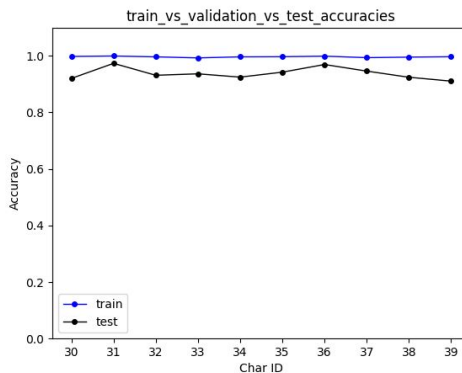
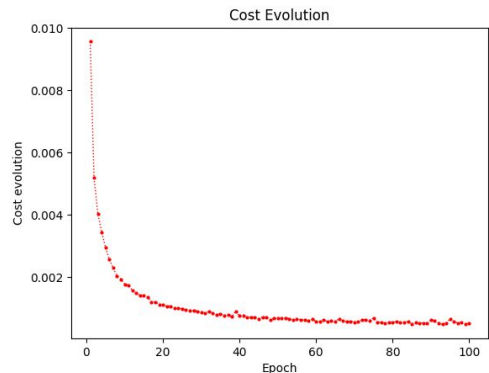
Output after training

Training, validation, testing

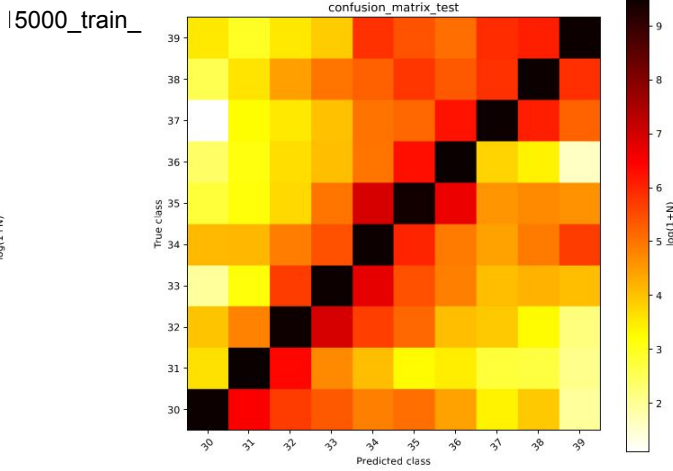
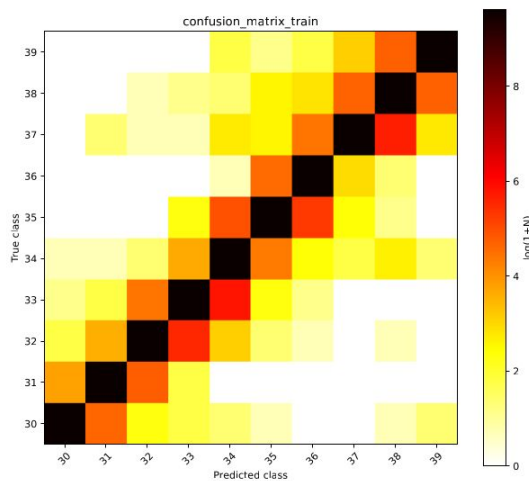
Pytorch, 10 chars

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3133898	qitek	20	0	7933644	1.5g	318124	R	2331	0.7	44:09.34	python
3009583	qitek	20	0	66.5g	1.0g	73008	S	0.0	0.4	5:47.59	node

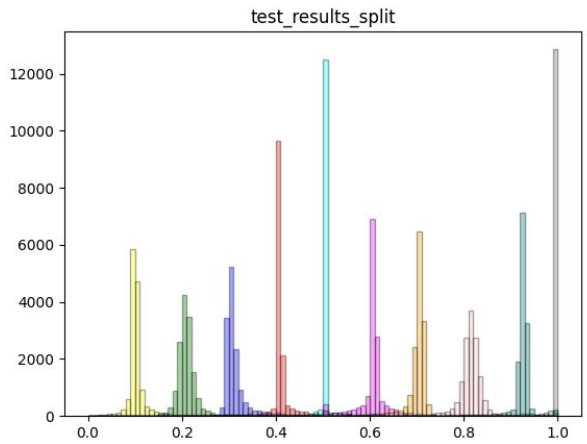
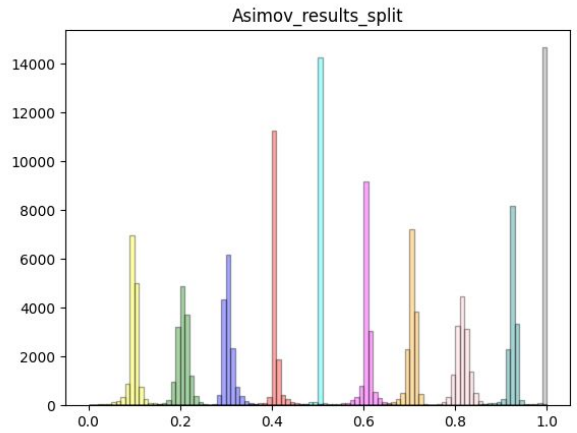
- results_pytorch_n1_250_n2_200_i1_0_i2_15000_train_30_31_32_33_34_35_36_37_38_39_nlmgs_15000_iters_100_bs_32_rate_0.005/
- handled! still pathological middle class (cyan) and far right (gray).



Pytorch, 10 chars, Confusion matrices



)_iters_100_bs_32_rate_0.005/



Impressions – for work

- Great tool to start a project with.
- Helps overcome initial technical necessities.
- However, the code can change a lot upon features request.
- It's good to play with the code, often personal modifications may not work w/o full code understanding.
- One should understand the language and code for serious purposes.

Impressions – for education and students

- A very nice laboratory and a tool to explore many things!
- Careful that actually finishing and nailing down means to really understand.
- Much easier for students to jump on in any language.
- Much more difficult to make sure they actually learn IT and coding principles.
- More discussion and even closer interaction with students needed.
- More possibilities for people to contribute
 - besides hard-core coders:
 - data analysis, AI-assisted testers, AI-assisted coders

Lessons learned

- AI-assisted coding is here and this will not change, only develop.
- Formulate clearly what you want
- Understand your data/inputs/goals
- Always doubt, design checks at all levels.
- Best, understand the generated code, too.
- Tweak and play with the code and suggested solutions.
- Try it with new inputs, use cases, stress it, challenge it!

More thoughts

- Imagine the possibilities for student theses, PhD work!:)
- What should we control?
- What should we teach?
- What should students learn?
 - IT skills?
 - Analytical and statistical methods?
 - Way of thinking?
 - Physics insight, intuition?
 - Abilities to apply and extend what they learned to new problems?
- There's no one-fits-all solution and future is unknown.
- But AI tools do bring an advantage, possibilities, though perhaps also risks.
- Who's going to drive further IT development?

No comment

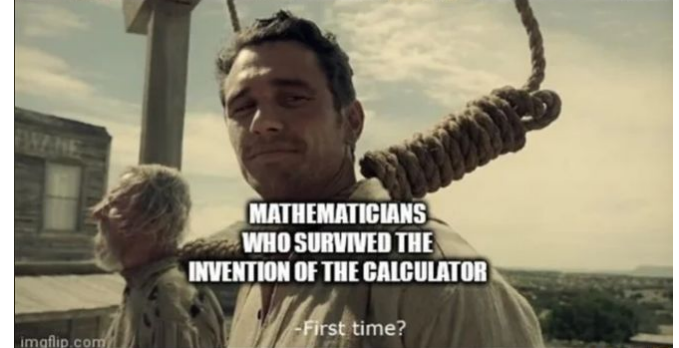
A.I. TURNS THIS SINGLE BULLET POINT INTO A LONG EMAIL I CAN PRETEND I WROTE.



A.I. MAKES A SINGLE BULLET POINT OUT OF THIS LONG EMAIL I CAN PRETEND I READ.




© marketoonist.com



Comments

no events are read, please find where and why

 fullyConnectedNN.py:253

I'll reproduce the read step in your current execution context, then trace whether it's failing due to file selection, parser conditions, or event filters, and pinpoint the exact line-level cause.

Executed terminal commands in project directory

I found the active file path now points to `../ascii.txt`, so next I'm testing the loader directly on that file to see whether parsing itself returns zero accepted events and why.

Executed commands in terminal

I found the root cause: the new ascii format uses trace headers like `0:` while the parser currently only accepts `pixel/event:` format, so every trace is discarded; I'll patch the parser to accept both formats and verify event count recovers.

Updated readData.py and searched for regex patterns +7 -5

Found and fixed.

Root cause:



1. After switching input selection, the script now reads `fastPlay/ascii.txt` from execution context via `fullyConnectedNN.py`.
2. That file stores traces as pixel-only headers like `0`, `1`, etc.
3. The parser in `readData.py` previously expected pixel/event format with a slash and discarded lines without it, so no traces were attached and accepted events stayed zero.

Where it broke:

- Header parsing logic at `readData.py` to `readData.py`.

What I changed:

1. Updated `readData.py` to support both formats:

  fullyConnectedNN.py:253

Where it broke:

- Header parsing logic at `readData.py` to `readData.py`.

What I changed:

1. Updated `readData.py` to support both formats:
 - `pixel/event: trace`
 - `pixel: trace`
2. For pixel-only headers, it now uses the current metadata event id.



Verification:

1. No errors in `readData.py` or `fullyConnectedNN.py`.
2. Re-test on the same file now gives:
 - Total events read: 448960
 - Accepted events: 214605

So the no-events issue was a parser format mismatch, not a cut.



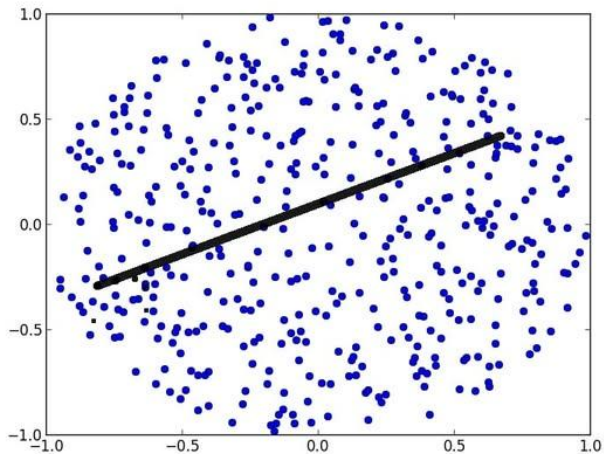
GPT-5.3-Codex • 13.7 credits

  fullyConnectedNN.py:253

No comment



Scientists be like



Somewhere from the internet

plot also loss function as function of the epoch

I'll add an epoch-loss plot (train and validation loss curves) after training, save it as png/pdf with the run tag, and ensure it's included in moved results artifacts.

Searched for regex and updated fullyConnectedNN.py +25 -1

Added in fullyConnectedNN.py: loss is now plotted as a function of epoch.

Credits at 75%

Set additional budget t...

Manage Budget

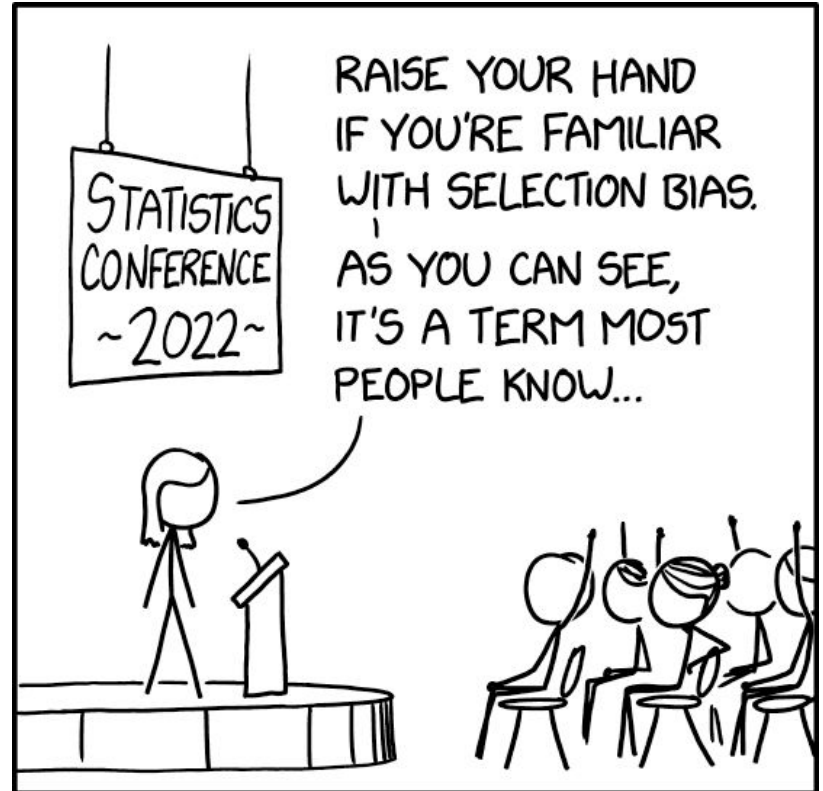
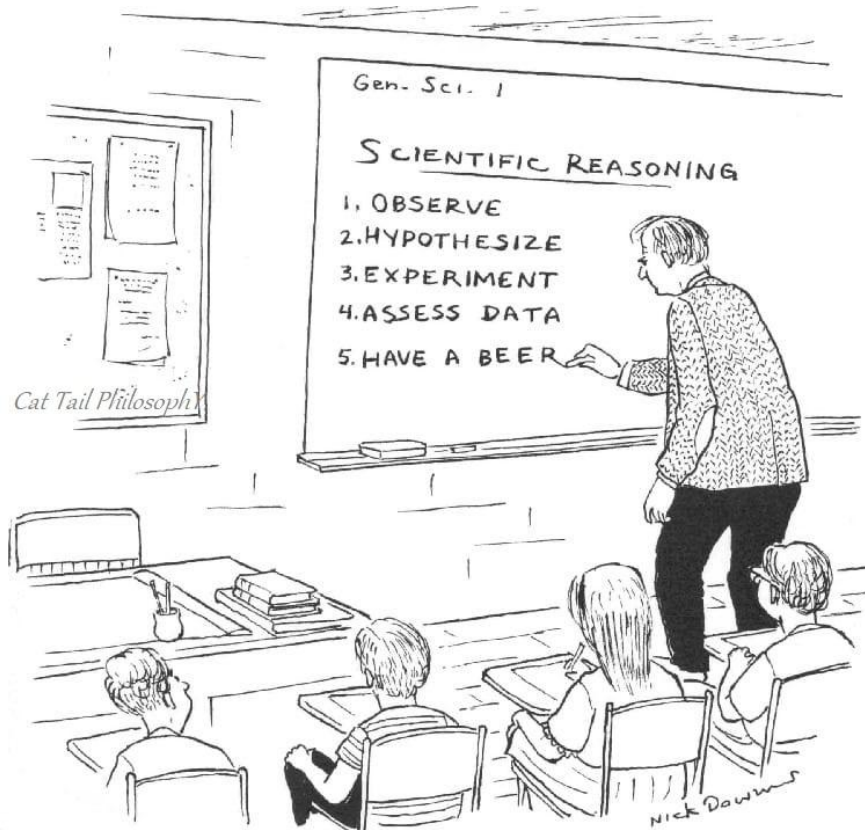
> 1 file changed +25 -1

Keep

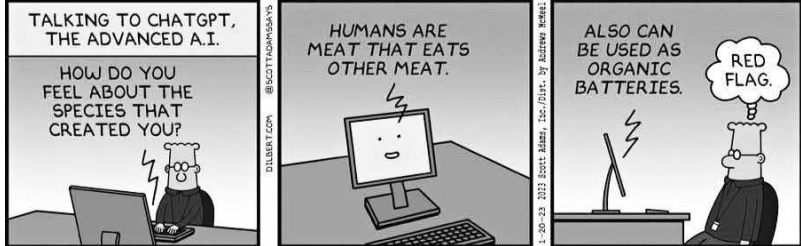
Undo

+ fullyConnectedNN.py

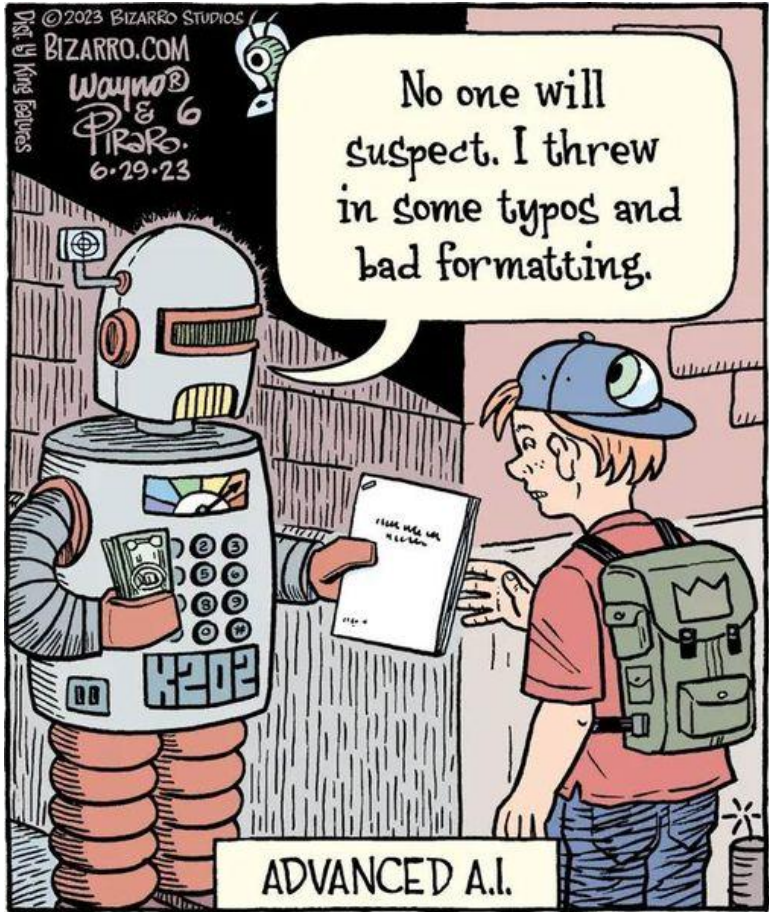
No comment

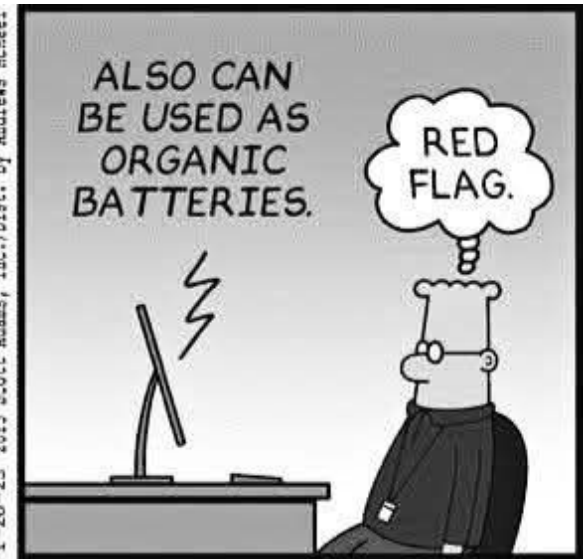
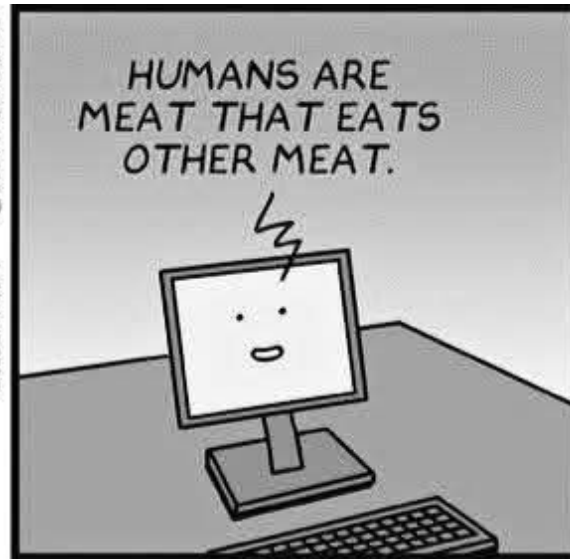
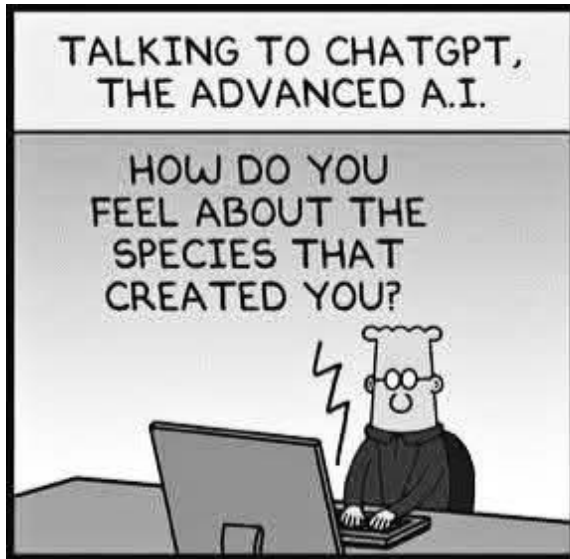


Somewhere from the internet



Somewhere from the internet





Somewhere from the internet

AI DAY 2026

16. červen 2026, 8:29 – 16:00

Cyklus přednášek na téma využití AI ve vzdělávání

Celouniverzitní přednáškový den určený vyučujícím, studujícím a dalším zájemcům z řad veřejnosti na téma využití AI ve vzdělávání.

Aktuální informace a program najdete ve [FB události](#) 

Pořádá

Filozofická fakulta UP, Přírodovědecká fakulta UP

Místo

Filozofická fakulta UP, Křížkovského 10, aula

<https://www.ff.upol.cz/nc/zprava/clanek/ai-day-2026-1/>
<https://ai.upol.cz/#doporuceni>



16. června 2026 | Aula FF UP





16.

Úterý 16. června 2026 v 10:00

AI DAY 2026


Křížkovského 10, 77900 Olomouc, Czech Republic

Podrobnosti

-  30 lidí zareagovalo
-  Událost pořádá **Filozofická fakulta Univerzity Palackého v Olomouci**
-  **Křížkovského 10, 77900 Olomouc, Czech Republic**
-  Veřejná · Kdokoli na Facebooku i mimo něj

Celouniverzitní přednáškový den určený vyučujícím, studujícím a dalším zájemcům z řad veřejnosti na téma využití AI ve vzdělávání.

Program bude zveřejněný brzy.

 Místo konání: aula FF UP (přední trakt, 3. NP, č. místnosti 3.32, Křížkovského 10) [Zobrazovat měř](#)

Olomouc

Představujeme pořadatele



Filozofická fakulta Univerzity Palackého v Olomouci

195 uplynulých událostí · [Stránka](#) · [Vysoká škola a univerzita](#)

Učíme myslet!
Liberálnost, osobní přístup ke studentům a s tím související rozvíjení individuálních schopností.

Backup

<https://xkcd.com/1838/>



AFP News Agency  @AFP

[Zobrazit překlad](#)

Anthropic calls for a global pause on building the most powerful AI systems as the latest models are beginning to show signs they could escape human control

u.afp.com/SDuy



Sebastien Bozon / AFP

6:26 dop. · 5. 6. 2026 · 10,9 tis. Zobrazení

15 39 35 8

