

# AI Progress in Retrospective: Can you Really Vibe?

MNNOG7 Keynote

Hervey Allen  
NSRC Assistant Director



Mongolian  
Network Operators  
Group

# Sequence

About me

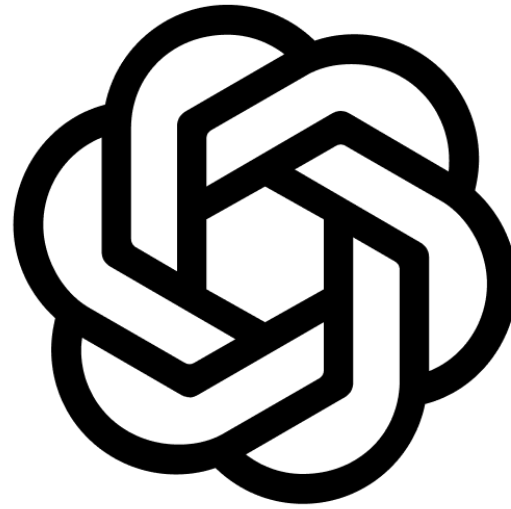
About NSRC

Neural Networks to LLMs

OpenAI and ChatGPT's History

Vibe Coding

A Real-World Vibe Coding Example



Hervey Allen



# The NSRC

**NSRC**  
Network Startup Resource Center

Formed as a non-profit organization by Randy Bush and John Klensin in 1990.

Located at the University of Oregon starting in 1996

1992

1990

1996

First National Science Foundation (NSF) grant formalized by the NSRC in 1992 at MIT.

**O**  
UNIVERSITY OF  
OREGON

**MIT** Massachusetts  
Institute of  
Technology





Eugene







*The NSRC cultivates collaboration among a community of peers to build and improve a global Internet that benefits all parties. We facilitate the growth of sustainable Internet infrastructure via technical training and engineering assistance to enrich the network of networks.*

*Our goal is to connect people.*

# NSRC Staff and Contractors

A world map illustrating the global distribution of NSRC staff and contractors. The map uses colored pins to mark locations across various continents. The colors of the pins are: yellow, dark green, orange, red, purple, and blue. The locations are as follows:

- North America:** Multiple yellow pins in the western and central US, and one yellow pin in Canada. One dark green pin is located in the northeast US.
- Europe:** One orange pin in the UK, one red pin in France, one yellow pin in Norway, and one dark green pin in Germany.
- Africa:** One orange pin in West Africa, one purple pin in Nigeria, and one orange pin in East Africa.
- Asia:** One yellow pin in China and one dark green pin in Southeast Asia.
- Oceania:** One blue pin in Australia and one dark green pin in New Zealand.
- South America:** One green pin in Colombia and one green pin in Venezuela.

- # NSRC Staff and Contractors
- 
- A world map illustrating the global distribution of NSRC staff and contractors. The map uses colored pins to mark locations across various continents. The colors of the pins correspond to the categories defined in the legend: Yellow (United States, Canada, Mexico, Central America, Caribbean), Dark Green (South America), Blue (Africa), Orange (Europe, Middle East, Asia), and Dark Blue (Australia, Oceania). The map includes labels for major countries and oceans.
- | Region            | Staff/Contractor Locations (Pin Color)   |
|-------------------|--|
| North America     | United States (Yellow), Canada (Yellow), Mexico (Yellow)   |
| South America     | Venezuela (Dark Green), Colombia (Dark Green)  |
| Africa            | Algeria (Blue), Mali (Blue), Nigeria (Blue), Chad (Blue), Sudan (Blue), Ethiopia (Blue), Kenya (Blue), Tanzania (Blue)   |
| Europe            | United Kingdom (Orange), Norway (Yellow), Sweden (Yellow), Poland (Orange), Germany (Orange), France (Orange), Spain (Orange), Italy (Orange), Turkey (Orange) |
| Asia              | China (Orange), South Korea (Orange), Japan (Orange), Thailand (Orange), Indonesia (Orange), Papua New Guinea (Orange)   |
| Australia/Oceania | Australia (Dark Blue), New Zealand (Dark Blue)   |

# Neural Networks to LLMs

What is a Neural Network?

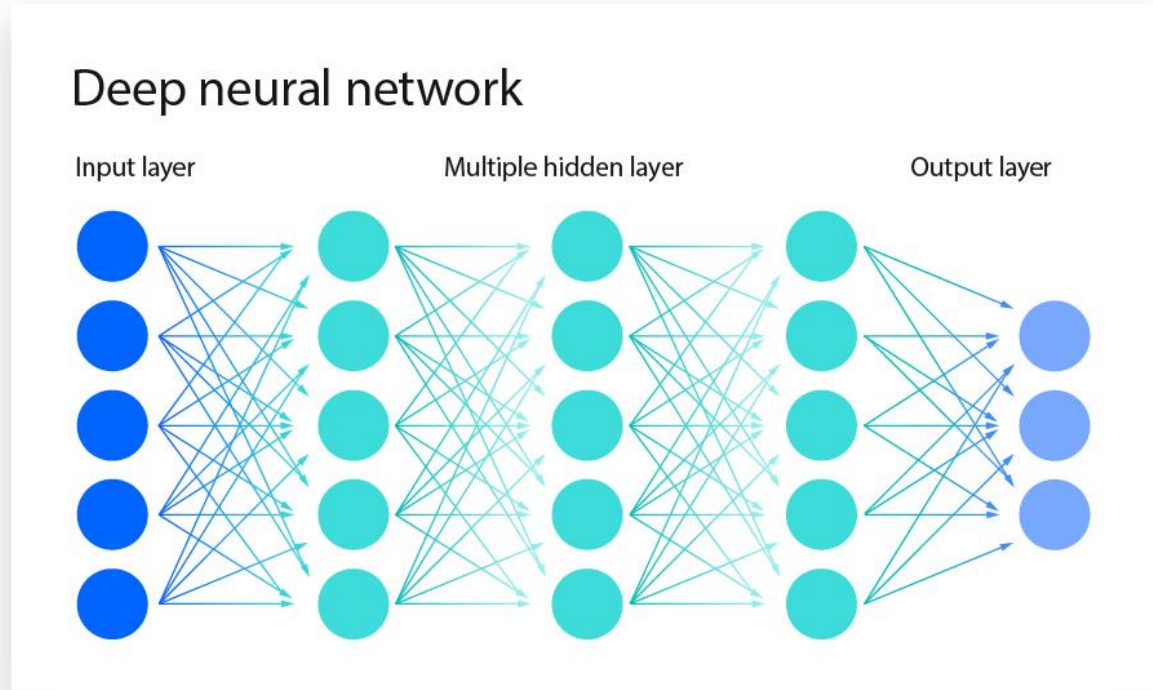
What was the Google paper called, “Attention Is All You Need,” about?

What is a LLM? (Large Language Model)

What has OpenAI done to move LLMs forward so quickly?



# A Neural Network

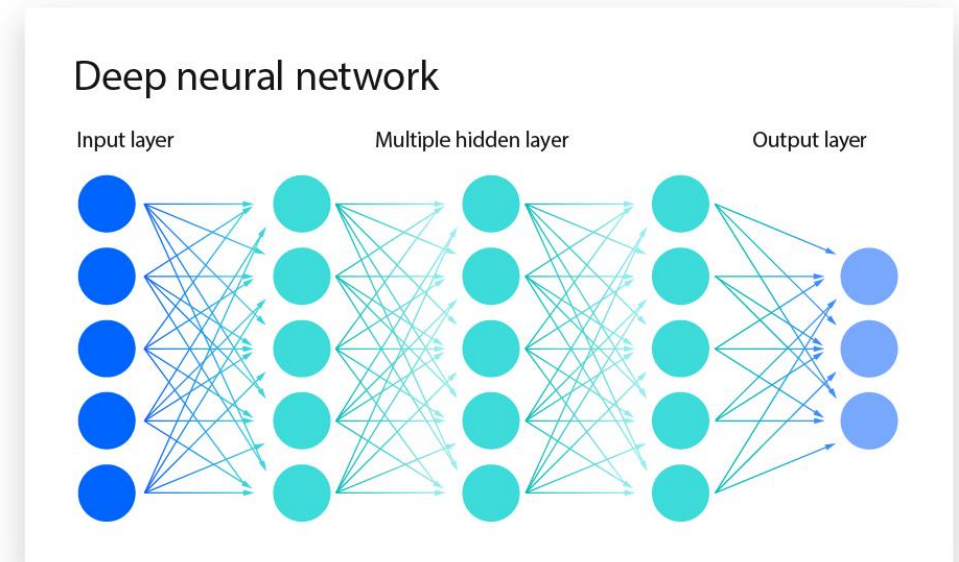


*Thank you, IBM*

- Computational Neural Network History starts in the 1940's and 1950's.
- Lots of details online:
  - [https://en.wikipedia.org/wiki/History\\_of\\_artificial\\_neural\\_networks](https://en.wikipedia.org/wiki/History_of_artificial_neural_networks)
  - IBM's excellent example:  
<https://www.ibm.com/think/topics/neural-networks>

# Details on Computational Neural Networks

*A computational neural network is a system of interconnected nodes, or artificial neurons, that processes information in layers, mimicking the human brain. The network learns to recognize patterns in data and make predictions by adjusting the "weights" and "biases" of the connections between neurons during a training process called backpropagation.\**



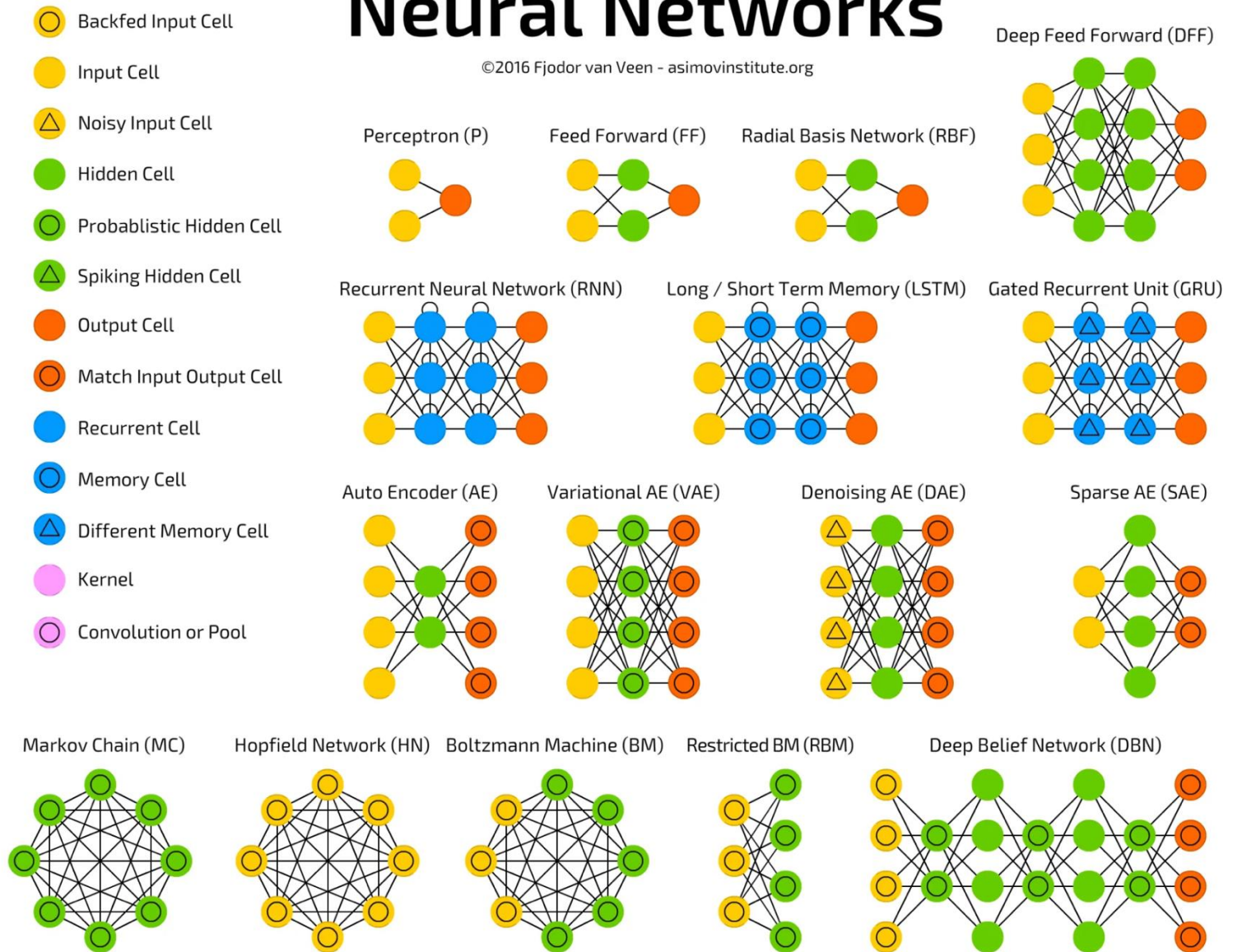
\*<https://www.google.com/search?q=sample+diagram+and+explanation+of+a+computational+neural+network>

<https://medium.com/data-science/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>

# Detailed Neural Network Map 😊

## A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org



# 1980's DECtalk DTC01 Training

**Details of this device:**

<https://en.wikipedia.org/wiki/DECtalk>

**Virtual DTC01 to use:**

<https://archive.org/details/dectalk>

**Sample text that was optimized using Neural Network training:**

*My dear fellow, be a Musketeer or an abbé. Be one or the other, but not both," replied Porthos. "You know what Athos told you the other day; you eat at everybody's mess. Ah, don't be angry, I beg of you, that would be useless; you know what is agreed upon between you, Athos and me. You go to Madame d'Aguillon's, and you pay your court to her; you go to Madame de Bois-Tracy's, the cousin of Madame de Chevreuse, and you pass for being far advanced in the good graces of that lady....*





# Machine Learning, Neural Networks

“Artificial Intelligence”

In 2017 a team of 8  
Google researchers  
wrote the research  
paper titled:

***“Attention Is All You Need”***

<https://arxiv.org/html/1706.03762v7>

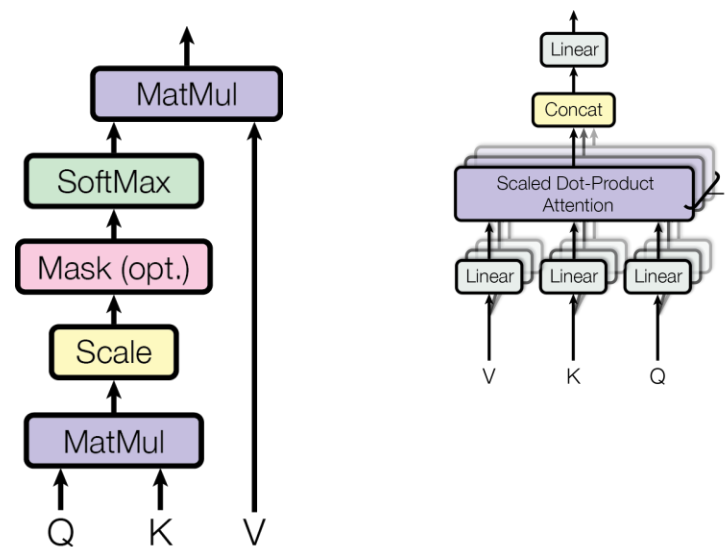
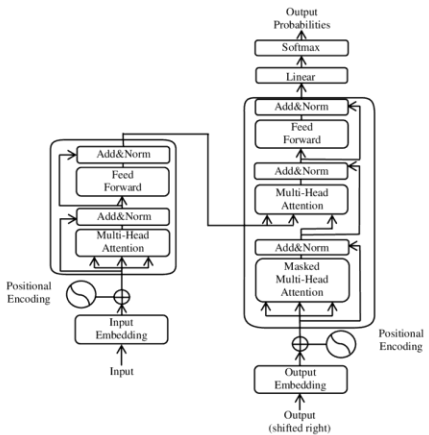


Figure:(left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

# “Attention Is All You Need”

## : Key Changes



The paper introduced a new deep learning architecture known as the **transformer**, based on the **attention mechanism** proposed in 2014 by Bahdanau et al. It is considered a foundational paper in modern artificial intelligence, and a main contributor to the AI boom, as the **transformer** approach has become the main architecture of a wide variety of AI, such as large language models.\*

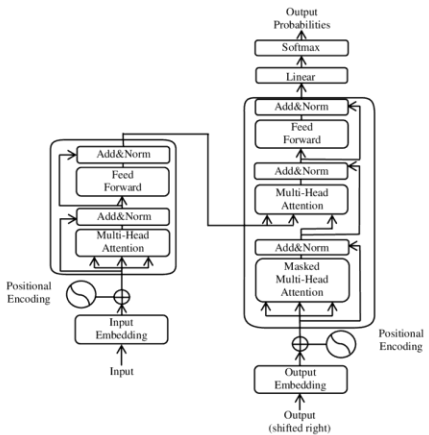
<https://poloclub.github.io/transformer-explainer/>

\*[https://en.wikipedia.org/wiki/Attention\\_Is\\_All\\_You\\_Need](https://en.wikipedia.org/wiki/Attention_Is_All_You_Need)

# *“Attention Is All You Need”*

:

## TRANSFORMER



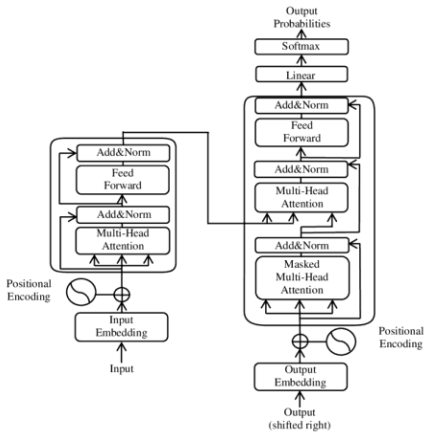
## TRANSFORMER

The **Transformer** model, from the paper "Attention Is All You Need," is a neural network that processes a sequence of data, such as a sentence, by focusing on the most important parts of the input. Instead of processing the sequence word-by-word like older models, it looks at the entire sequence at once, a change that made it faster and more effective.

***“Attention Is  
All You Need”***

:

## ATTENTION MECHANISM



## ATTENTION MECHANISM

A way for a neural network to focus on the most relevant parts of a sequence of data, like a sentence. It was introduced in the **Transformer** model, which uses attention exclusively, without the need for older, less efficient components like recurrent or convolutional neural networks.\*

\*<https://www.google.com/search?q=plain+english+text+to+describe+the+attention+mechanism+model+in+the+paper+%22attention+is+all+you+need>



# What happened next?\*



Early OpenAI LLMs used the **transformer** architecture to enable their progress



Unlike earlier sequential models, the **transformer** neural network architecture used a self-**attention** mechanism. This mechanism could process an entire input at once, which improved performance. Foundation for almost all modern LLMs.



OpenAI was an early adopter of transformers at scale. OpenAI was one of the first organizations to realize the potential of scaling up transformer-based models. The first model in their GPT (Generative Pre-trained Transformer) series, released in 2018, was built using the decoder-only part of the transformer architecture.



OpenAI's early GPT models defined the frontier of LLM performance for the public.

\*Thank you Google

<https://www.google.com/search?q=did+openai+use+google+paper+to+create+first+llm>

# ChatGPT Versions – Pre Public Release

## **GPT-1 (2018)**

- Unsupervised multitask learning, proving the potential of large language models.
- Established the foundational **transformer** architecture but was significantly smaller than its successors.

## **GPT-2 (2019)**

- An advancement in scale, utilizing a significantly larger dataset and more parameters than GPT-1.
- Demonstrated more robust language understanding and generation, capable of more reliable and coherent outputs due to increased size and data.

# ChatGPT Versions – Pre Public Release

## **GPT-3 (2020)**

- A dramatic leap in model scale, with billions of parameters.
- Showcased unprecedented performance in a wide array of natural language processing (NLP) tasks, including text generation, translation, and summarization, by training on a massive dataset.

## **InstructGPT**

- Fine-tuning to improve the models' ability to follow instructions and be more helpful, honest, and harmless.
- Utilizes Reinforcement Learning from Human Feedback (RLHF) on top of a GPT-3 base.

# ChatGPT First Public Release

## March 2022

### **GPT-3.5**

Details <https://en.wikipedia.org/wiki/GPT-3#GPT-3.5>

- Major breakthrough was the interface
- Improvements with models

Training with chat data, so better at chatting (better conversationalist)



# ChatGPT 4 and 5

## **GPT-4 (2023)**

- Another level of scaling, more compute, fix some bugs... big leap in ability and used chat

## **GPT-4o (2024)**

- First models that could take pre-training, post-training and use reasoning at run time...  
Good at logical reasoning, math

## **GPT-4.1 (2025)**

- Now we have scaling laws bigger, chat training, now we add tools -execute code, search web, better overall model

## **GPT-4.5 (2025)**

- Focused on increasing world model accuracy and intuition through unsupervised learning.

## **GPT 5 (2025)**

- People confused about tools in O-3... make the model user select what is needed, that most users are not sure how to do
- By default, GPT 5 figures out level of reassigning needed, but you can force it in to "thinking mode" where full reasoning is used. I.E., maybe deeper answer, but slower

# ChatGTP 5 for “Free”

## How to Access GPT-5 for Free

### **ChatGPT Free Tier:**

- Simply sign in to your free account on the ChatGPT website. You'll have access to GPT-5, but with usage limits before being switched to a different model.

### **Microsoft Copilot:**

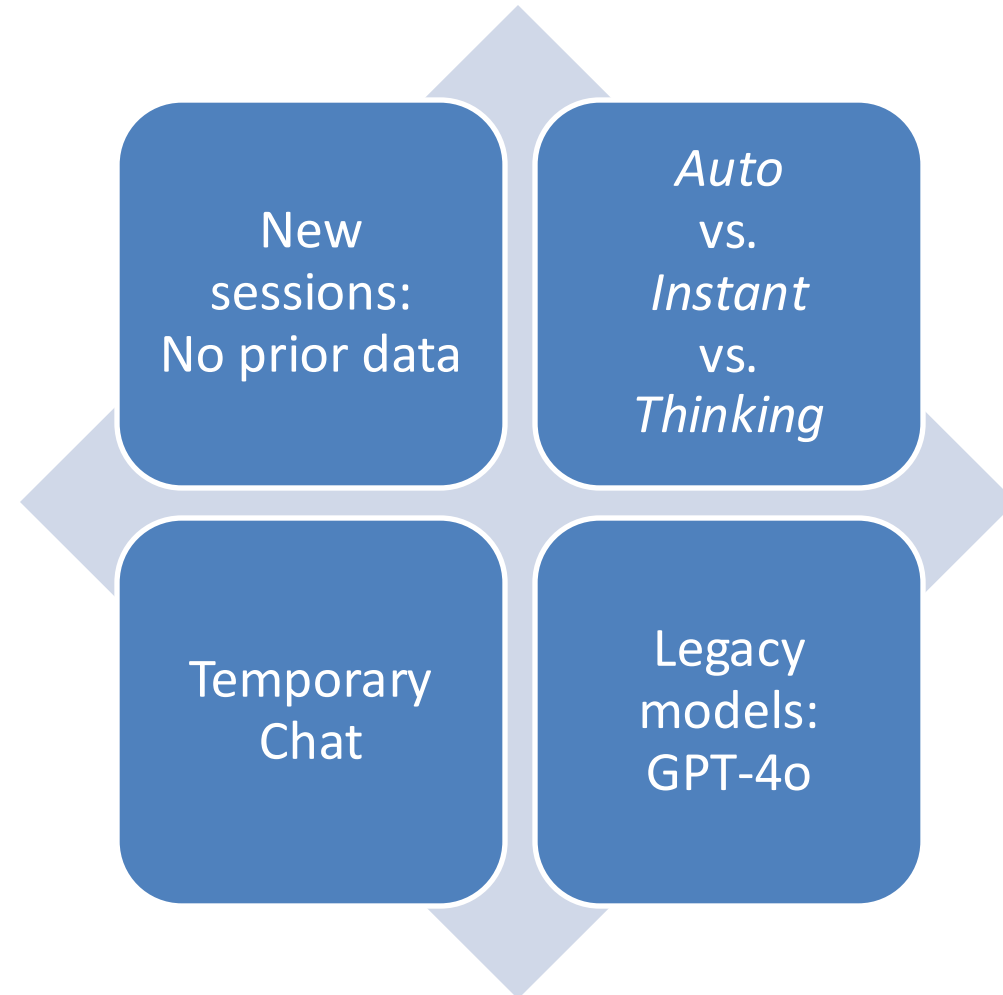
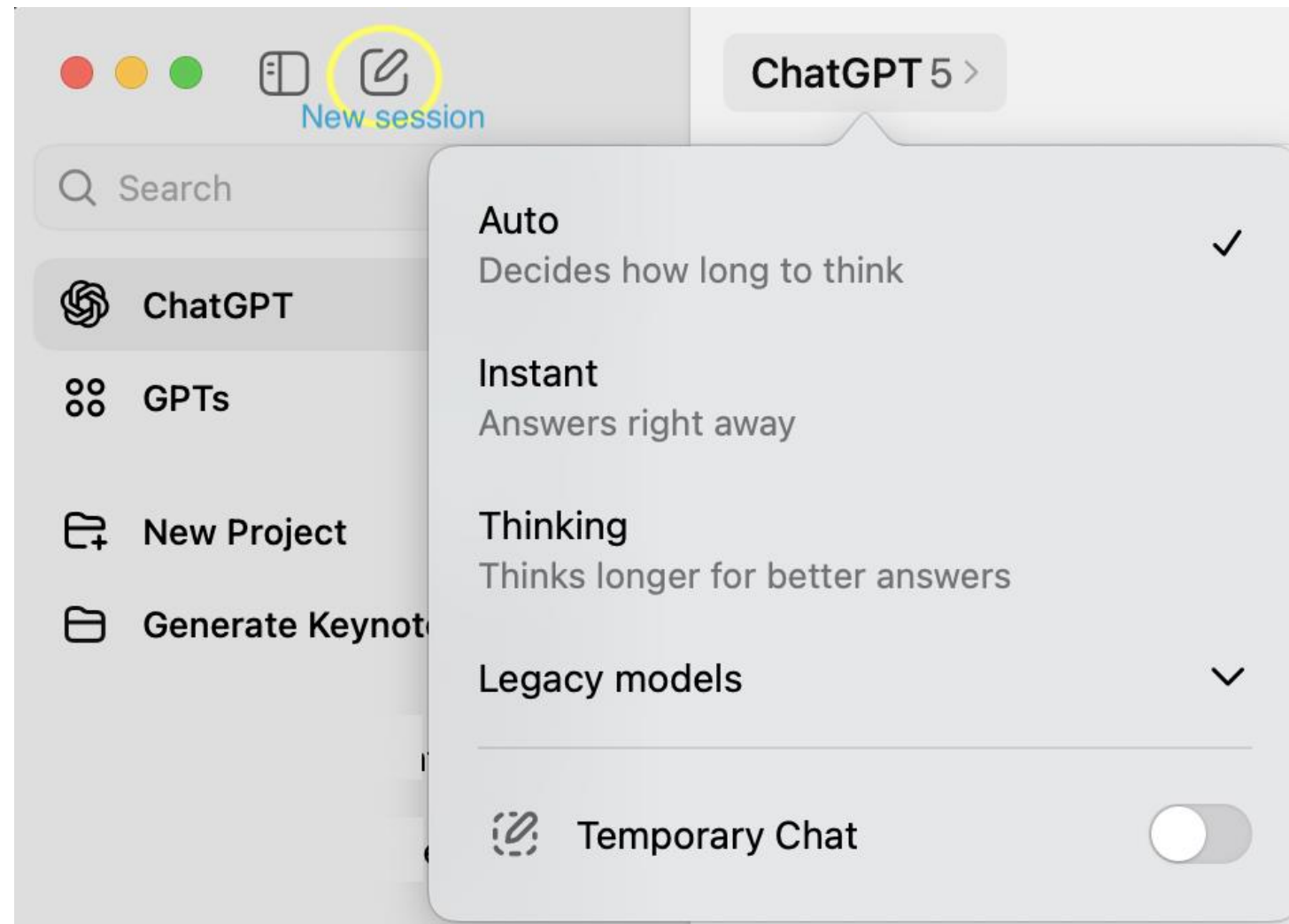
- Users of Windows 11 can also use GPT-5 for free through Microsoft Copilot by selecting the "[Smart](#)" option in the taskbar, or a dedicated key in newer computers, or with Windows key + C shortcut, or in Settings.

# ChatGTP 5 for “Free”

## Understanding the Free Tier Limitations

- **Message Limits:** You get a limited number of GPT-5 "Thinking messages" before your limit resets. The exact number can vary.
- **Fallback Model:** When you hit your GPT-5 limit, your chat will automatically switch to a less capable "mini" version of the model for the rest of the window.
- **Peak Hours:** During peak usage times, free tier users may hit their limits faster or experience slower responses, as OpenAI prioritizes paid subscribers.

# A few ChatGPT 5 Tips and Tricks



# A few ChatGPT 5 Tips and Tricks

## New Session

- Can lead to clearer, context-free responses, while continuing a chat saves time and builds on your previous work.

## Modes

- **Auto** mode (default) adjusts between fast and deep reasoning,
- **Instant** mode prioritizes quick, straightforward answers
- **Thinking** mode performs more thorough, step-by-step analysis for complex tasks, which takes longer but yields better results for challenging queries (requires much more compute time).

## Temporary Chat

- **Private**, "incognito" mode where conversations are not saved to your history, not used to train OpenAI's models, and the AI does not remember past interactions or memories.



# Legacy models

Legacy models have specific strengths. Some creative or specialized tasks exist where newer models may not perform as well or feel different. Users with workflows built around older models can use them during a transition period with GPT-5, or even long-term for certain accounts, to maintain stability.

## Reasons for Legacy Model Utility

### Specialized Performance:

- Some legacy models, such as those in the "o3" or "GPT-4o" family, are preferred for specific types of output, like generating creative stories, personal writing, or maintaining consistent character voices.

### Workflow Consistency:

- Users who have built their workflows, projects, or even emotional support around a specific model's consistent performance can continue using it.

# Legacy models

## **Familiarity:**

- Older models provide stability and familiarity for users, especially when moving to new versions like GPT-5.

## **Enterprise and Edu Users:**

- These users can temporarily enable legacy model access in their workspaces to adapt to the newest models while updating custom GPTs or finishing ongoing projects.

## **Pro Users:**

- Plus and above users can retain access to some legacy models, such as GPT-4o, for a limited time to maintain their workflows.

## **Transition Period**

- The availability of legacy models is part of a temporary transition period, with OpenAI moving users toward the newest defaults, like GPT-5.

## **Future Availability**

- While legacy access is currently available for some users, this feature is likely to be for a limited time as the transition to new models completes.

# Providers & Strategic Advantages (High-Level)

- **OpenAI**
  - Broad capability focus; strong tooling ecosystem
- **Anthropic (Claude.AI)**
  - Emphasis on safety/alignment; strong coding assistants
- **Google**
  - Deep science research; integration with Workspace & Search
- **Microsoft**
  - Enterprise distribution; integrates OpenAI across M365/Azure
- **xAI**
  - Leverages social data signals; fast iteration in consumer UX
- **Sovereign AI trend**
  - Local data residency, cultural norms, in-region DCs

# Let's “Vibe”

## We've already been doing this 😊

- Look at the presentation...
- Finding data used in the presentation...
- I tried to use multiple AI-based Presentation Creator tools:
  - Gamma
  - ChatGPT
  - Plusai.com
  - Beautiful.ai
  - PowerPoint design options per slide...\*

\*This one worked

# Let's Vibe Code

## Many tools...

- Many tools have an LLM in the core and a framework built around it.
- **Some tools include:**
  - Claude.AI\* (today's tool, by Anthropic)
  - Bolt (<https://bolt.new/>)
  - ChatGPT 5\* (OpenAI)
  - Cursor\* (Cursor.com)
  - Trae (<https://www.trae.ai/>)
  - Lovable (<https://lovable.dev/>)

\*Fairly language agnostic for python, rust, javascript

<https://www.digitalocean.com/resources/articles/vibe-coding-tools>



+

•

○

# Vibe Coding: Core components and functionalities

*The concept of "vibe coding," where a developer uses a conversational style to generate code with an LLM, is powered by these frameworks. Tools like Lovable and Bolt allow users to describe their ideas in natural language, and the underlying LLM-centric framework handles the complex translation into a functional application*



# Our tool today...

- Claude by Anthropic, former OpenAI employees in San Francisco, CA, USA
- Web based (claude.ai)
- For starting out a project it performs better when it already has a structure presented to it, but Claude.AI is better at building this itself than many other tools.
- <https://claude.ai/public/artifacts/89d8449f-a40d-4d4f-83a7-d5a5661a601a>

# First “vibe” session... : What I did

H

I'm going to give a keynote presentation in Mongolia about AI...

Can you assist by creating a php script that will run on Ubuntu 24.04 to read the JSON file entry that is at:

<https://services.swpc.noaa.gov/json/goes/primary/xray-flares-latest.json>

And, will print out a to a file `/var/www/html/current_flare.html`

that has the title "GOES-18 Satellite Current Solar Flare Level"

and, then writes a new just below the title by a few lines that says:

"The latest GOES-18 satellite Xray Flare Class level is" <level found from JSON here>

Where "<level found from JSON here>" is the value found in the JSON file in the field "current\_class":

And, can you put after the value the UTC time value?

And, can you keep all the prior values in a bullet list below the statement near the top ordered by most current dates at the top and oldest date at the bottom.

We will assume that we will create a cron job on the Ubuntu 24.04 server to run this script every 1 minute, but if the first JSON field "time\_tag" has not changed, then you can skip updating the web page as the JSON file has not updated.

Can, you, also provide the sample cron file that we can install in `/etc/cron.d` and you can assume that the cron job will run the script, so that it has permissions to write the file `/var/www/html/current_flare.html` each minute.

First “vibe”  
session...  
:  
What  
Claude.AI did  
1 of 2

I'll create a PHP script that monitors solar flare data and generates an HTML page, along with the cron configuration.

Solar Flare Monitor Script  
Code



Perfect! I've created both the PHP script and cron configuration for your solar flare monitoring system.

### Installation Instructions:

#### 1. Save the PHP script:

```
bash

sudo nano /usr/local/bin/solar_flare_monitor.php
```

Copy the first artifact content, then make it executable:

```
bash

sudo chmod +x /usr/local/bin/solar_flare_monitor.php
```

#### 2. Install the cron job:

```
bash

sudo nano /etc/cron.d/solar-flare-monitor
```

Copy the second artifact content.

First “vibe”  
session...  
:  
What  
Claude.AI did  
2 of 2

### 3. Create the log file and set permissions:

```
bash

sudo touch /var/log/solar_flare_monitor.log
sudo chown www-data:www-data /var/log/solar_flare_monitor.log
sudo chown www-data:www-data /var/www/html
```

### 4. Test the script manually:

```
bash

sudo -u www-data /usr/bin/php /usr/local/bin/solar_flare_monitor.php
```

### Key Features:

- **Checks for updates:** Only updates the HTML if the `time_tag` has changed
- **Maintains history:** Keeps the last 50 readings in reverse chronological order
- **State persistence:** Uses `/tmp/solar_flare_state.json` to track previous readings
- **Styled HTML:** Clean, professional output with responsive design
- **Error logging:** All errors are logged to `/var/log/solar_flare_monitor.log`
- **Runs as www-data:** Ensures proper permissions for writing to `/var/www/html`

The script will run every minute via cron, check the NOAA data, and only update the HTML page when new data arrives. Perfect for your keynote presentation in Mongolia!



# Now, I built a server (VM)

- Did this after I got started when I realized what I was going to need.
- Started with VM at Linode (Singapore data center).
  - Ubuntu 24.04, installed php and apache2 plus log files
  - Set up DNS for domain (herveyallen.com)
  - Set up ssl for domain and sub-domains (mnnog7.herveyallen.com)
  - Pushed Claude.AI PHP file (solar\_flare\_monitor.php) to VM

- Asked Claude.AI to share a sample cron file, it gave me\*:

```
# Run every minute
* * * * * www-data
/usr/bin/php
/usr/local/bin/solar_flare_monitor.php
>>
/var/log/solar_flare_monitor.log 2>&1
```

- I knew this was incorrect

```
# Run every minute
* * * * * www-data /usr/bin/php /usr/local/bin/solar_flare_monitor.php >> /var/log/solar_flare_monitor.log 2>&1
```

# Second “Vibe” Initial changes

- I considered making the changes I needed manually, but that’s not the idea!
- I kept the original PHP file to review changes once I requested them from Claude.AI
- What did I ask for? My exact request...

H OK - I have one change I want to make... The generated PHP script is going to reside in the home directory of the user hervey. The file will now be at:

`/home/hervey/mnnog/`

It can still be named `solar_flare_monitor.php`. Can you make any needed updates to make sure this will work to both the PHP script and to the cron.d file `solar-flare-monitor`?

# What did I get?

## : *Changes*

H

OK - I have one change I want to make... The generated PHP script is going to reside in the home directory of the user hervey. The file will now be at:

`/home/hervey/mnnog/`

It can still be named `solar_flare_monitor.php`. Can you make any needed updates to make sure this will work to both the PHP script and to the cron.d file `solar-flare-monitor`?

Edit

### Solar Flare Monitor Script

Code • Version 2



Perfect! I've updated both files. Here are the changes:

### Changes Made:

- 1. PHP Script:** Changed the state file location from `/tmp/` to `/home/hervey/mnnog/solar_flare_state.json` so it's stored alongside your script
- 2. Cron file:** Updated the path to `/home/hervey/mnnog/solar_flare_monitor.php`

# What did I get?

## :

# *How to install*

## Installation Steps:

### 1. Create the directory and save the PHP script:

```
bash
mkdir -p /home/hervey/mnnog
nano /home/hervey/mnnog/solar_flare_monitor.php
```

### 2. Set permissions so www-data can read the script and write the state file:

```
bash
sudo chmod 755 /home/hervey/mnnog
sudo chmod 644 /home/hervey/mnnog/solar_flare_monitor.php
```

### 3. Install the cron job:

```
bash
sudo nano /etc/cron.d/solar-flare-monitor
```

Paste the updated cron content

### 4. Ensure www-data can write to the necessary locations:

```
bash
sudo chown www-data:www-data /var/www/html
sudo touch /var/log/solar_flare_monitor.log
sudo chown www-data:www-data /var/log/solar_flare_monitor.log
```

### 5. Test it:

```
bash
sudo -u www-data /usr/bin/php /home/hervey/mnnog/solar_flare_monitor.php
```

The state file will now be stored at `/home/hervey/mnnog/solar_flare_state.json` alongside your script!

I see  
problems...

**Following the instructions I get the error:**

*“Could not open input file: /home/hervey/mnnog/solar\_flare\_monitor.php”*

**Anyone have any ideas what kind of issues we might have?**

- Yes, No, Maybe...
- How to solve this without expertise?

Use “google” for help? Use our Vibe tool for help?  
Know where to look and resolve from experience?

I asked Claude (again):

H Hi Claude - Thanks for this. So, I followed the instructions, then as root I ran the command:

```
sudo -u www-data /usr/bin/php /home/hervey/mnnog/solar_flare_monitor.php
```

and the error I get is:

```
Could not open input file: /home/hervey/mnnog/solar_flare_monitor.php
```

Any ideas what I can do to resolve this?



# Claude's Fix : Third "Vibe"

The issue is that the `www-data` user doesn't have permission to access the `/home/hervey/mnnog/` directory. Home directories typically have restrictive permissions (usually 700 or 750) that don't allow other users to enter them.

Here are a few solutions:

## Option 1: Grant `www-data` access to the directory (Recommended)

```
bash

# Allow others to traverse into hervey's home and the mnnog directory
chmod 755 /home/hervey
chmod 755 /home/hervey/mnnog
chmod 644 /home/hervey/mnnog/solar_flare_monitor.php
```

Then test again:

```
bash

sudo -u www-data /usr/bin/php /home/hervey/mnnog/solar_flare_monitor.php
```

## Option 2: Move the script to a shared location

If you prefer not to open up permissions on your home directory, move the script:

```
bash

sudo mkdir -p /opt/mnnog
sudo mv /home/hervey/mnnog/solar_flare_monitor.php /opt/mnnog/
sudo chmod 755 /opt/mnnog
sudo chmod 644 /opt/mnnog/solar_flare_monitor.php
```

Then I'll need to update the paths in both the PHP script (for the state file) and the cron file.

## Option 3: Run as the `hervey` user instead

Change the cron file to run as `hervey` instead of `www-data`, but you'll need to ensure `hervey` has write permissions to `/var/www/html/`.

**Which option would you prefer?** Option 1 is usually the quickest solution if you're comfortable with those permissions.

Anything  
wrong with  
this?

## Yes...

Options 1 and 3 are not great

- Option 1 exposes user data
- Option 3 adds exposure to web files

Option 2 is more commonly used

Option 4 (not provided)

- Place the script in  
`/var/www/<script_dir>`
- We have not discussed where the  
cron file should live 😊

# Fixing using Claude's Option 2

## Option 2: Move the script to a shared location

If you prefer not to open up permissions on your home directory, move the script:

bash

```
sudo mkdir -p /opt/mnnog  
sudo mv /home/hervey/mnnog/solar_flare_monitor.php /opt/mnnog/  
sudo chmod 755 /opt/mnnog  
sudo chmod 644 /opt/mnnog/solar_flare_monitor.php
```



Then I'll need to update the paths in both the PHP script (for the state file) and the cron file.

Obtain the  
cron.d file  
again

H

Can you provide me with the updated cron.d entry again?

Here's the complete cron.d file content:

bash



```
# Solar Flare Monitor Cron Job
# File: /etc/cron.d/solar-flare-monitor
# Description: Runs every minute to check GOES-18 satellite solar flare o

# Run every minute
* * * * * www-data /usr/bin/php /opt/mnnog/solar_flare_monitor.php >> /va
```

To install it:

bash

```
sudo nano /etc/cron.d/solar-flare-monitor
```

Paste the content above, save and exit. The cron daemon will pick it up automatically within a minute!

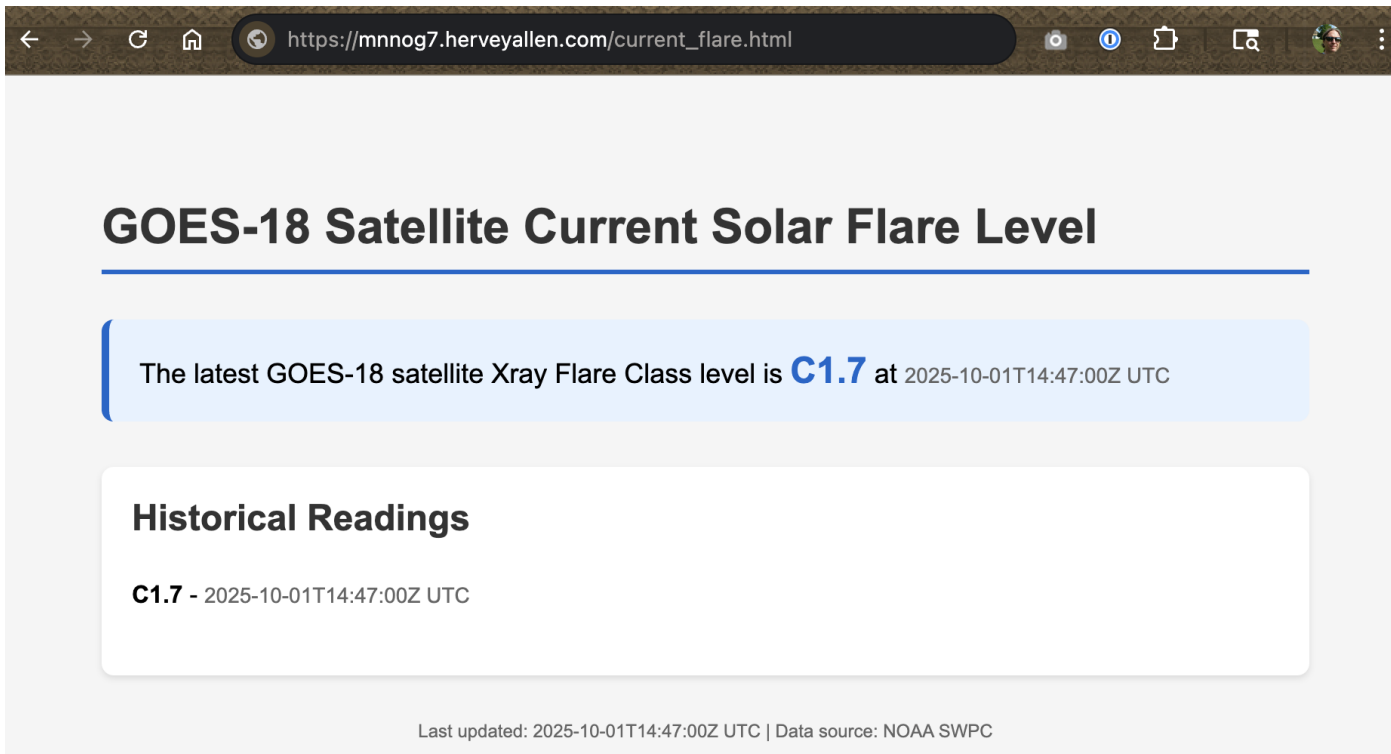
Execute by hand...

...

It works! 😊

# Manual execution

```
[root@mnnog7:/opt/mnnog# sudo -u www-data /usr/bin/php /opt/mnnog/solar_flare_monitor.php  
[Successfully updated solar flare data. Current class: C1.7
```



The screenshot shows a web browser window with the URL [https://mnnog7.herveyallen.com/current\\_flare.html](https://mnnog7.herveyallen.com/current_flare.html). The page title is "GOES-18 Satellite Current Solar Flare Level". The main content area shows a blue box with the text: "The latest GOES-18 satellite Xray Flare Class level is **C1.7** at 2025-10-01T14:47:00Z UTC". Below this is a section titled "Historical Readings" which lists "C1.7 - 2025-10-01T14:47:00Z UTC". At the bottom of the page, it says "Last updated: 2025-10-01T14:47:00Z UTC | Data source: NOAA SWPC". A blue arrow points from the terminal output above to the "C1.7" value on the webpage.

Flare Class	Time (UTC)
C1.7	2025-10-01T14:47:00Z





After a few minutes...

# ...it kind of works...

---

The web page displays only current result,  
not a bullet list of past results.

# Wash, rinse, repeat (“Vibing”)

## Fourth and final “Vibe”

### I asked Claude.AI:

H Ok. We are getting closer. The web page is only showing the most current value found. This is perfect for the main entry at the top of the page, but for the Historical Readings section the page should update after each minute and the bullet list should grow. In reverse order, most current number at the top, next most current measurement, etc...

I saw that you had planned on showing this up to 50 entries at the most (that would be 50 minutes of data). Can you make the list be 100 entries in the bullet list instead.

Thank you.

### Claude.AI said...

Solar Flare Monitor Script  
Code • Version 4



Perfect! I've updated the script to keep 100 entries instead of 50.

The historical readings list should already be growing with each update. The script works like this:

1. Each time new data arrives (with a different `time_tag`), it adds the new reading to the beginning of the history array
2. It keeps the last 100 entries
3. The HTML displays them in order from most recent to oldest

To see it working:

- Wait a minute or two for the cron job to run again (when new data arrives from NOAA)
- Refresh the page at `/var/www/html/current_flare.html`
- You should see the bullet list growing with each new reading

To verify the state file is saving history:

```
bash
cat /opt/mnnog/solar_flare_state.json
```

You should see a JSON file with a `history` array that contains multiple entries.

If the list isn't growing, it might be because the NOAA data hasn't updated yet (the `time_tag` is still the same), so the script is correctly skipping updates. The solar flare data typically updates every few minutes depending on satellite reporting intervals.

# Wash, rinse, repeat (“Vibing”)

## I realized... and, Claude.AI said...

**H** Oh, I see the problem, the file `/opt/mnnog/solar_flare_state.json` is not being written. Any ideas what might be wrong?

The issue is that `www-data` doesn't have permission to write to `/opt/mnnog/`. Let's fix that:

```
bash
# Give www-data ownership of the directory so it can create/write the state file
sudo chown www-data:www-data /opt/mnnog

# Or if you prefer to keep root ownership, just make it writable
sudo chmod 775 /opt/mnnog
```

Then test again:

```
bash
sudo -u www-data /usr/bin/php /opt/mnnog/solar_flare_monitor.php
```

## Furthermore...

Now check if the state file was created:

```
bash
ls -la /opt/mnnog/
cat /opt/mnnog/solar_flare_state.json
```

You should see the `solar_flare_state.json` file now, and it should contain your history data.

**Alternative solution** if you want to keep `/opt/mnnog` more restricted:

```
bash
# Create the state file with proper ownership
sudo touch /opt/mnnog/solar_flare_state.json
sudo chown www-data:www-data /opt/mnnog/solar_flare_state.json
sudo chmod 644 /opt/mnnog/solar_flare_state.json
```

The first option (making the directory writable) is easier since it allows `www-data` to create the file automatically.



# The alternate solution listed

This is what you want to do:

1. Create the file you need first
2. Now that it exists, give it a tighter restriction to www-data
3. This is a better security practice

Phew!

Will this get done?






Continue  
vibing?

Do we need to do more...?

**Yes!**

1. Review the PHP code in

`/opt/mnnog/solar_flare_monitor.php`

- Check for efficiency
  - Check for security
  - Does it need to be updated for failure situations?
  - Make the results nicer
- 




Continue  
vibing?

Who does this?

- Claude.AI?
- Yourself?
- Both?

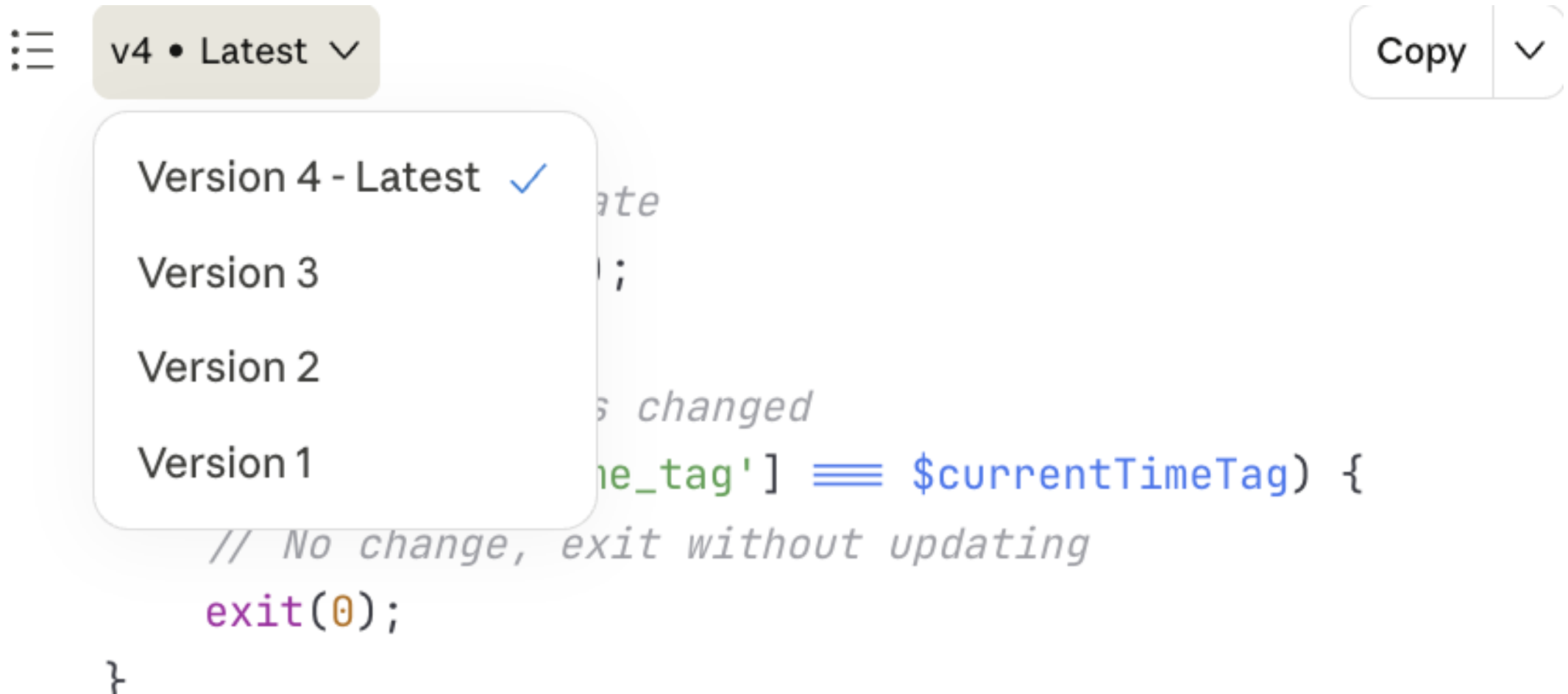
How do you know if it's decent code  
(particularly for security)?

- You either trust what has been created,  
or
  - You use someone with expertise to  
review and verify
- 



# How I left this... (after fourth “Vibe”)

4 revisions of the PHP file were created... I installed the 4<sup>th</sup> one:



# How I left this...

After a while the web page  
is looking good...

## GOES-18 Satellite Xray Flare Class Page location

[https://mnnog7.herveyallen.com/current\\_flare.html](https://mnnog7.herveyallen.com/current_flare.html)

## The source code location

<https://claude.ai/public/artifacts/43227dc3-a023-4ae6-b62f-a2fd0ea70032>

## US NOAA GOES JSON file that is read

<https://services.swpc.noaa.gov/json/goes/primary/xray-flares-latest.json>

## The US NOAA GOES X-Ray Flux Page

<https://www.swpc.noaa.gov/products/goes-x-ray-flux>

## GOES-18 Satellite Current Solar Flare Level

The latest GOES-18 satellite Xray Flare Class level is **C1.8** at 2025-10-01T15:36:00Z UTC

### Historical Readings

**C1.8** - 2025-10-01T15:36:00Z UTC

**C1.8** - 2025-10-01T15:35:00Z UTC

**C1.8** - 2025-10-01T15:34:00Z UTC

**C1.7** - 2025-10-01T15:33:00Z UTC

**C1.7** - 2025-10-01T15:32:00Z UTC

**C1.7** - 2025-10-01T15:30:00Z UTC

**C1.7** - 2025-10-01T15:28:00Z UTC

**C1.7** - 2025-10-01T15:27:00Z UTC

**C1.7** - 2025-10-01T15:26:00Z UTC

**C1.7** - 2025-10-01T15:25:00Z UTC

**C1.6** - 2025-10-01T15:24:00Z UTC

# What does it all mean?

---

The grand question posed by many...

A typical response...

<https://online.uc.edu/blog/artificial-intelligence-ai-benefits/>

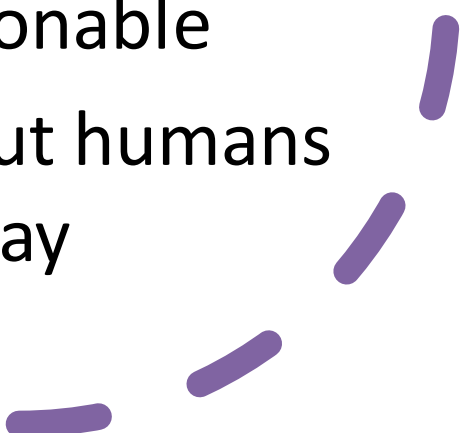
## BENEFITS OF ARTIFICIAL INTELLIGENCE

- 
- The illustration features a large, stylized circuit board with red and gold traces. A robot with a tan and white body stands on the left, and a person in a red blazer and black pants stands on the right. A white rectangular box with a black border is centered on the board, containing a list of nine benefits of artificial intelligence.
1. Enhanced Healthcare
  2. Boosted Economic Growth
  3. Climate Change Mitigation
  4. Advanced Transportation
  5. Customer Service Excellence
  6. Scientific Discovery
  7. Enhanced Financial Services
  8. Improved Agriculture
  9. Enhanced Cybersecurity



What does it  
all mean?

### **This author's take...**

- LLM-based systems get part of many questions right, but not all of it
  - The classic issue, how do you hire an expert without being an expert yourself?
  - How do you let this tool create something if you are not expert enough to know if the creation is reasonable
  - For now, let AI do the basic, but humans step in at some point of the way
- 




What did we  
See?

## **This author's take, cont...**

- Claude.AI helped speed up core coding
- Claude.AI made some (imho) less than reasonable suggestions to get code running
- We have not yet reviewed the code for security, reliability or efficiency

*When can we stop reviewing  
what's being created?*



Баярлалаа / Thanks



Mongolian  
Network Operators  
Group



# References

## History

- History of artificial neural networks  
[https://en.wikipedia.org/wiki/History\\_of\\_artificial\\_neural\\_networks](https://en.wikipedia.org/wiki/History_of_artificial_neural_networks)
- Attention is All You Need (Google Research Paper)  
[https://en.wikipedia.org/wiki/Attention\\_Is\\_All\\_You\\_Need](https://en.wikipedia.org/wiki/Attention_Is_All_You_Need)

## OpenAI and ChatGPT – Vibe Coding

- “10 Best Vibe Coding Tools: LLM-Powered Code Generators to Try”  
<https://www.digitalocean.com/resources/articles/vibe-coding-tools>
- “What is vibing with AI?”  
<https://www.google.com/search?q=what+is+vibing+with+ai>
- “What is Vibe Coding”  
<https://www.ibm.com/think/topics/vibe-coding>
- “What is Vibe Coding”  
<https://theconversation.com/what-is-vibe-coding-a-computer-scientist-explains-what-it-means-to-have-ai-write-computer-code-and-what-risks-that-can-entail-257172>