

Generative AI models running in your own infrastructure

Escuela de Ingeniería de Fuenlabrada

Fuenlabrada, Spain, October 14th, 2025

Jesus M. Gonzalez-Barahona

<https://jgbarah.github.io/presentations/>

What's in a generative model

- Architecture
 - Weights, just weights
 - Software to make inferences

But also:

- Data to train, benchmark
- Software to train, benchmark
- Weights of intermediate models
- Documentation, explaining everything

A wide spectrum

- Behind-app model
- Directly accessible model
- Available weights model
- Open weight model
- Open source model
- Reproducible (libre) model

Behind-app model

Directly accessible model

Available weights model

Open weight model

Open source model

Reproducible (libre) model

Classes of "openness"

THE MODEL OPENNESS FRAMEWORK: PROMOTING
COMPLETENESS AND OPENNESS FOR REPRODUCIBILITY,
TRANSPARENCY, AND USABILITY IN ARTIFICIAL INTELLIGENCE

Matt White*
Linux Foundation
San Francisco, CA, USA

Ibrahim Haddad
Linux Foundation
San Francisco, CA, USA

Cailean Osborne
University of Oxford
Oxford, UK

Xiao-Yang Yanglet Liu
Columbia University
New York, USA

Ahmed Abdelmonsef
Generative AI Commons
Cairo, Egypt

Sachin Mathew Varghese
Generative AI Commons
Jacksonville, FL, USA

Arnaud Le Hors
IBM & Generative AI Commons
San Jose, CA, USA

<https://arxiv.org/abs/2403.13784>

Classes of "openness"

- Open model: architecture, model parameters, weights & metadata, tech report, model card, data card
- Open tooling: open model plus training, inference, and evaluation code, libraries, evaluation data
- Open science: open tooling plus paper, datasets, log files, intermediate models parameters, weights & metadata

For each of them...

- Freedom of use
- Freedom of study
- Freedom of modification
- Freedom of sharing

The "four freedoms" from "What is Free Software?"

Some specific aspects

- Access: can you run inferences the way you want?
- Control on the model: can you modify the way the model works? (eg, finetune it)
- Control on your data: can you control the prompt, the results?
- Autonomy: how much you depend on the model provider?
- Trust: can you ensure the model works as intended? (eg, backdoors, etc.)

Why this matters?

- Model access: Use cases, innovation, integration
- Model control: Use cases, innovation, integration
- Data control: Privacy, independence, reliance
- Autonomy: Market competition, independence, reliance
- Trust: Security, transparency.

Behind-app model

- An application uses one or more models to provide some service
- It can be a local or cloud application
- The application may be generalist or very specific
- The model may change over time

 Understanding Free Software Licenses

Sources

+ Add

🔍 Discover

Select all sources ☒

 gbiblioteca,+1-M-barahona.pdf ☒

 https://jgbarah.github.io/presenta... ☒

Chat



Understanding Free Software Licenses

2 sources

Los recursos proporcionados por Jesús M. González-Barahona ofrecen una **visión completa del software libre**, explicando sus fundamentos y evolución. Se detalla que el software libre se define por **cuatro libertades clave**: usarlo sin restricciones, estudiar y modificar su funcionamiento, redistribuirlo y distribuir versiones modificadas, siendo esencial el acceso al código fuente. El autor también **aborda la historia del software libre**, desde sus orígenes hasta su madurez actual, y explora las **implicaciones para los usuarios**, incluyendo ventajas como la competencia en el mercado y la independencia del productor, así como **desafíos**. Finalmente, se hace un **examen profundo de las licencias de software libre**, diferenciando entre

Start typing...

2 sources



¿Cómo las licencias de software libre definen las libertades del usuario?

¿Qué >

Studio

 Audio Overview

⋮

 Video Overview

⋮

 Mind Map

 Reports

▼

 Understanding Free Softwar...
2 sources · 127d ago

⋮

 A Concise History of Free Software
Timeline · 2 sources · 127d ago

⋮

 Add note

NotebookLM can be inaccurate; please double check its responses.

- Google NotebookLM

11



DeepL



Start free trial

Menu



Translate text

35 languages



Translate files

.pdf, .docx, .pptx



DeepL Write

AI-powered edits

Spanish (detected) ▾



English (American) ▾

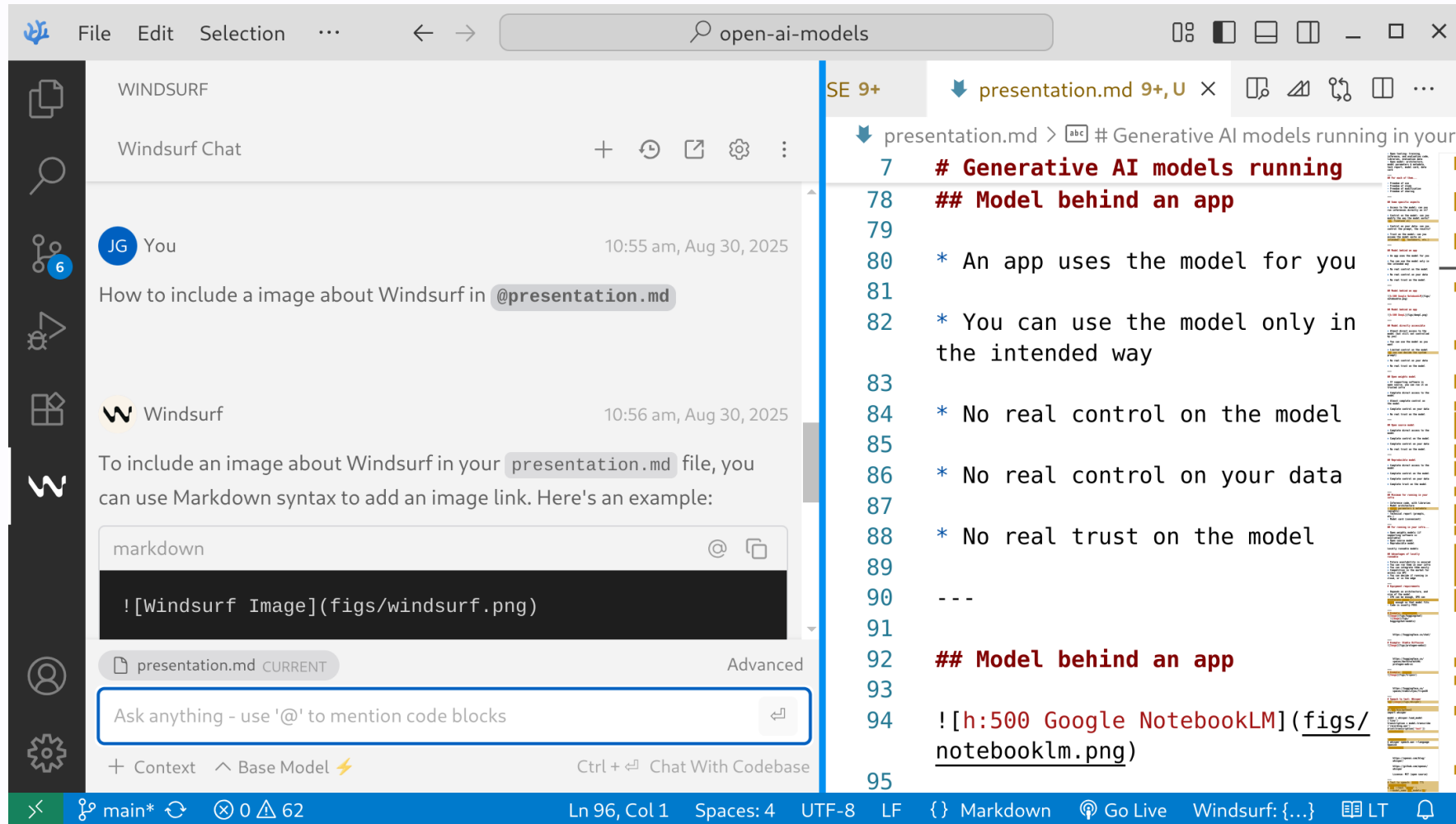
Options ▾

Un LLM al que se accede vía
una aplicación



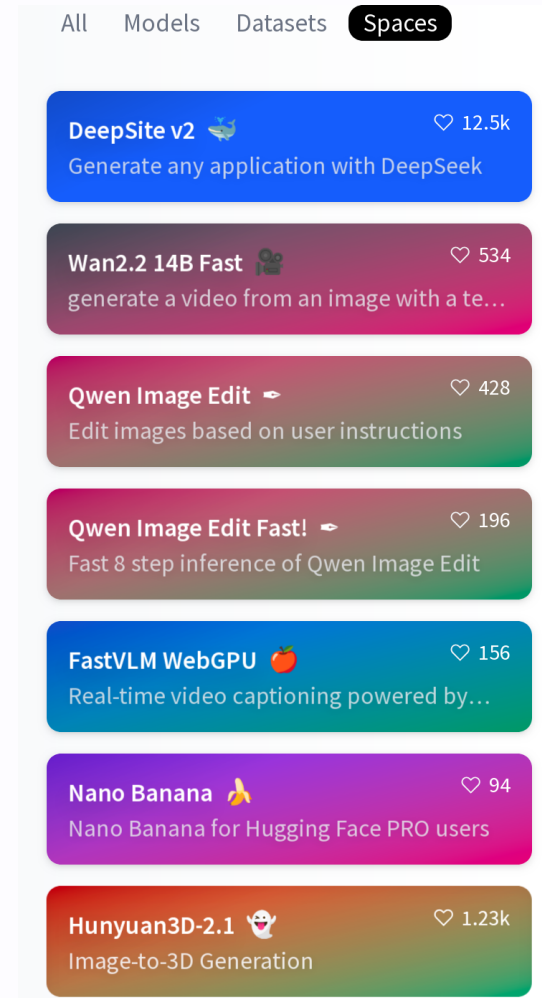
An LLM accessed via an
application

- DeepL



- Windsurf in Visual Studio code

- HuggingFace Spaces

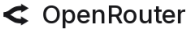


Behind-app model

- Access: use the model only in the intended way
- Model control: none
- Data control: none
- Autonomy: none
- Trust: none

Directly accessible model

- Access usually via HTTP API
- The API defines to which extent the model can be controlled
- Libraries and SDKs may be available
- Designed for building apps, depending on the API
- The model may change over time

 OpenRouter



The Unified Interface For LLMs

Better [prices](#), better [uptime](#), no subscription.



Featured Models [View Trending](#)

Gemini 2.5 Pro
by [google](#)

171.5B	2.4s	+8.82%
Tokens/wk	Latency	Weekly growth

GPT-5
by [openai](#)

69.1B	5.8s	+21.67%
Tokens/wk	Latency	Weekly growth

Claude Sonnet 4
by [anthropic](#)

624.0B	1.8s	+4.04%
Tokens/wk	Latency	Weekly growth

- OpenRouter

 **groq**cloud

 **OpenAI GPT-OSS (20B & 120B) models now available for instant inference.** These models have built-in browser search and code execution capabilities.[Learn about GPT-OSS >](#)

REASONING

-  GPT OSS 120B
-  GPT OSS 20B
-  DeepSeek R1 Distill Llama
-  Qwen 3 32B

TEXT TO SPEECH

-  PlayAI TTS

VISION

-  Llama 4 Scout
-  Llama 4 Maverick

FUNCTION CALLING / TOOL USE

-  GPT OSS 120B
-  GPT OSS 20B
-  Llama 4 Scout
-  Qwen 3 32B
-  Kimi K2

SPEECH TO TEXT

-  Whisper Large v3
-  Whisper Large v3 Turbo

TEXT TO TEXT

-  GPT OSS 120B
-  GPT OSS 20B
-  Kimi K2
-  Llama 4 Scout
-  Llama 3.3 70B

MULTILINGUAL

-  GPT OSS 120B
-  GPT OSS 20B
-  Kimi K2
-  Llama 4 Scout
-  Llama 3.3 70B
-  Whisper Large v3

SAFETY / CONTENT MODERATION

-  Llama Guard

- Groq

Directly accessible model

- Access: use as you want, but API restricts parameters
- Model control: limited, depending on the API
- Data control: none
- Autonomy: none
- Trust: none

Available weights model

- Weights are available
- Usually, software for inferences is available / FOSS
- Can be run on trusted infrastructure
- Finetuning, etc. is usually possible
- Redistribution, modification, use may be conditioned or forbidden

In some cases, referred as "open weight models"

Available weights model

- Access: use as you want if conditions are met
- Model control: deep control, if conditions are met
- Data control: complete
- Autonomy: depends on the conditions
- Trust: none

Available weights models examples

- Kimi K2
 - Modified MIT License
- LLaMa3 models (2025-04)
 - Meta Community License
 - Meta's LLaMa license is still not Open Source

Available weights models examples

- [Gema3](#)
 - Gemma Terms of Use
- [Mistral Large](#) (2024-11)
 - Mistral Research License

Available weights models examples

- **Tülu3**
 - Llama Community License
 - Finetuned from Llama3.1
 - All details and data of the finetune available
- **Cohere Command A** (2025-08)
 - Creative Commons Attribution-NonCommercial 4.0 International and Cohere Terms of Use

Open weight model

- Allows use, redistribution, derived works
- No conditions for use
- Derived works: finetuning, integration...
- Does not require information about the model, its training, etc. (no freedom of study)

Open Weight Definition

Open-Weight AI Models: What They Are, and Why
OpenAI's Next Move Matters

Open weight model

- Access: use as you want
- Model control: deep control
- Data control: complete
- Autonomy: only study is restricted
- Trust: none

Open weight model example

- **Granite Code** (2024-11)
 - Apache 2.0 License
 - Very few information about the model, training, etc.

Open source model

- Allows use, redistribution, derived works
- No conditions for use
- Derived works: finetuning, integration...
- Open source software for training, inferencing
- Detailed description of training, doesn't require availability of the training dataset

Open Source AI Definition

What are Open Weights?

Open source model

- Access: use as you want
- Model control: deep control
- Data control: complete
- Autonomy: detailed study is restricted
- Trust: partial

Open source model examples

- [DeepSeek](#) (2025-08)
 - MIT License
- [GPT-OSS](#) (2025-08)
 - Apache 2.0 License
- [Qwen3](#) (2025-04)
 - Apache 2.0 License

Reproducible (libre) model

- Allows use, redistribution, derived works
- No conditions for use
- Derived works: finetuning, integration...
- All information about the model
- Requires availability of the training dataset

Reproducible (libre) model

- Access: use as you want
- Model control: deep control
- Data control: complete
- Autonomy: complete
- Trust: complete

Reproducible (libre) model examples

- Olmo2, technical report (2025-05)
 - Apache 2.0 License
- MAP-Neo (2025-04)
 - Apache 2.0 License

Reproducible (libre) model examples

- LLM360 K2, technical report (2024-07)
 - Other LLM360 models
- Apertus (2025-09)
 - Apache 2.0 License

Behind-app model

Directly accessible model

Available weights model

Open weight model

Open source model

Reproducible (libre) model

	Access	Model Control	Data Control	Autonomy	Trust
Behind-app	App-defined	None	None	None	None
Directly accessible	API restrictions	API restrictions	None	None	None
Available weights	With conditions	With conditions	Complete	With conditions	None
Open weight	Use as you want	Deep control	Complete	Study restricted	None
Open source	Use as you want	Deep control	Complete	Detailed study restricted	Partial
Reproducible	Use as you want	Deep control	Complete	Complete	Complete

Reproducibility in AI research

Reproducible AI: Why it Matters & How to Improve it

Guidelines for Empirical Studies in Software
Engineering involving Large Language Models

Ethical model

- Conditions on use: "ethical use"
- Conditions on training: "ethical datasets"

Depends on what is considered as "ethical"

Most Ethical LLMs

[Home](#)[Ranking](#)[Methodology](#)[Pricing](#)[EN](#)

Rank	Model	Company	Ethical Score	Data Legality (0-5 points)	Transparency (0-5 points)	Bias Audit (0-5 points)	Opt-out	CO ₂
1	Phi-3	Microsoft	20/25	4/5	5/5	4/5	2/5	5/5
2	Snowflake Arctic (15B)	Snowflake	17/25	5/5	5/5	4/5	0/5	3/5
3	Claude 3.7 Sonnet	Anthropic	16/25	3/5	4/5	4/5	3/5	2/5
4	Gemma 2	Google	15/25	3/5	5/5	2/5	2/5	3/5
5	LLaMA 3	Meta	14/25	2/5	3/5	3/5	1/5	5/5
6	GPT-4o	OpenAI	13/25	2/5	3/5	3/5	4/5	1/5

LLM Responsible AI Rankings

Open issues

- Complex relationship of data, recipes, architecture, weights, software...
 - Missing exact definitions for several of the categories
 - It is not easy to find out in which category is a model
- Adapted definitions for finetunes and other evolutions of models?
- How to ensure that declarations are true?

Self-hostable models

Minimum for running in your infra

- Inference code, with libraries
- Model parameters & metadata (weights)
- Technical report (prompts, etc.)
- Model card (convenient)

At least, "available weights" if inference code is available

Self-hostable models

Self-hostable models

- If supporting software is available
 - Available weights models
 - Open weights models
- Open source models
- Reproducible models
- Ethical models (usually)

Behind-app model

Directly accessible model

Available weights model

Open weight model

Open source model

Reproducible (libre) model

The screenshot shows the Hugging Face website interface. At the top, there's a navigation bar with the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Docs, and Pricing. Below the navigation bar, the left sidebar contains tabs for Main, Tasks, Libraries, Languages, Licenses, and Other. The 'Tasks' section lists various tasks like Text Generation, Image-Text-to-Text, Image-to-Image, Text-to-Image, Text-to-Video, and Text-to-Speech. The 'Parameters' section shows a slider for model size, ranging from < 1B to > 500B. The 'Libraries' section lists various libraries like PyTorch, TensorFlow, JAX, Transformers, Diffusers, Safetensors, ONNX, GGUF, Transformers.js, MLX, and Keras. The main content area displays a list of models, including microsoft/VibeVoice-1.5B, xai-org/grok-2, openbmb/MiniCPM-V-4_5, Qwen/Qwen-Image-Edit, deepseek-ai/DeepSeek-V3.1, and Wan-AI/Wan2.2-S2V-14B. Each model entry shows its name, task, size, update time, and download/like counts.

- HuggingFace Models

Advantages of self-hostable

- Future availability is ensured
- You can run them in your infra
- You can integrate them easily
- Competition in the market for access via API
- You decide if running in cloud, or in the edge

Disadvantages of self-hostable

- Available models are usually not as good as the state of the art
 - (maybe 6-12 months delay?)
- Improvement and support are not always happening

Advantages of self-hosting

- Full control of everything
- For many kinds of tasks, it is good enough
- Easier for reproducibility

Disadvantages of self-hosting

- Investment in infrastructure
- Limitations of the hardware you can get
- Maintenance is your responsibility
- Being up-to-date maybe a pain
 - Decissions on best models
 - Decisions on best software
 - Decisions on best hardware

Technical skills required!

Equipment requirements

- Depends on architecture, and size of the model
- CPU can be enough, GPU can accelerate a lot
 - Cloud options for GPUs
- RAM enough so that model fits
- Code is usually FOSS

You can also deploy in a cloud-based host

Economic aspects

- Hardware requirements
 - Payback period
- Size and characteristics of the workload
- Sources of complexity:
 - Multiple users, multiple models, etc
- Maintenance costs
- Energy consumption

You have to do the math

Locally runnable via API

- Many different providers, similar APIs
 - Most of them use OpenAI API
 - Many of them provide several models
 - Examples: OpenRouter, Groq
- Frameworks provide backends for several providers
 - Examples: LiteLLM, LangChain
- Easy testing of several models
- Depending on workload, can be cheaper

Quantization

- Used to reduce hw requirements
- From float32 to fp16, to int8, or less
- Usually, post-training
- Several formats:
 - GPTQ, usually using `safetensors` files
 - GGUF, popular in the llama.cpp ecosystem

[HuggingFace Guide on Quantization](#)

[GGUF: Structure and Usage](#)

Finetuning

- Adapting a model to a specific task with a (usually smaller) specialized dataset
- Starting point: weights of given model.
- Adjust weights to fit the specialized dataset
- Adapter: new layers of weights added to a model to finetune it

What is fine-tuning?

HuggingFace: Quantizations and finetunes

📁 Model tree for Qwen/Qwen3-8B

Base model [Qwen/Qwen3-8B-Base](#)

📁 Finetuned ([154](#)) [this model](#)

└─ Adapters [123 models](#)

└─ Finetunes [230 models](#)

└─ Merges [19 models](#)

└─ Quantizations [144 models](#)

Inference engines

- llama.cpp
 - provides a web-based UI and HTTP API
- vLLM
 - provides HTTP API

Frameworks for LLMs

- **LangChain**: Chain together large LLM operations into sophisticated workflows, usually to build agent tools
- **LiteLLM**: Agile toolset designed for efficiency and simplicity.

Both can use local models, or models via HTTP API

Chat / assistant frontends

- [Ollama](#), based on llama.cpp
- [Oobabooga WebUI](#)
- [Open WebUI](#): "almost" FOSS
- [LibreChat](#): [online](#), [source code](#)
- [Jan](#): Local assistant

Most of them also provide an HTTP API

Ollama: how to run

```
curl -fsSL https://ollama.com/install.sh | sh  
ollama serve
```

```
ollama run gemma3:1b
```

```
curl http://localhost:11434/api/generate -d '{  
  "model": "gemma3:1b",  
  "prompt": "Why is the sky blue?"  
}'
```

Using Ollama to host an LLM on CPU-only equipment
to enable a local chatbot and LLM API

Open WebUI: how to run

```
uv venv --python 3.11  
uv pip install open-webui  
uv run open-webui serve
```

Now, open <http://localhost:8080>

gemma3:1b ▾ +



Hi, WebUI, how are you

OI gemma3:1b

Hello there! As a large language model, I don't "feel" in the same way humans do, but I'm functioning perfectly and ready to help you.

I'm doing well, thank you for asking! 😊

How about you? How are *you* doing today? Is there anything you'd like to chat about or any task you'd like me to assist with?

Send a Message



Code Interpreter

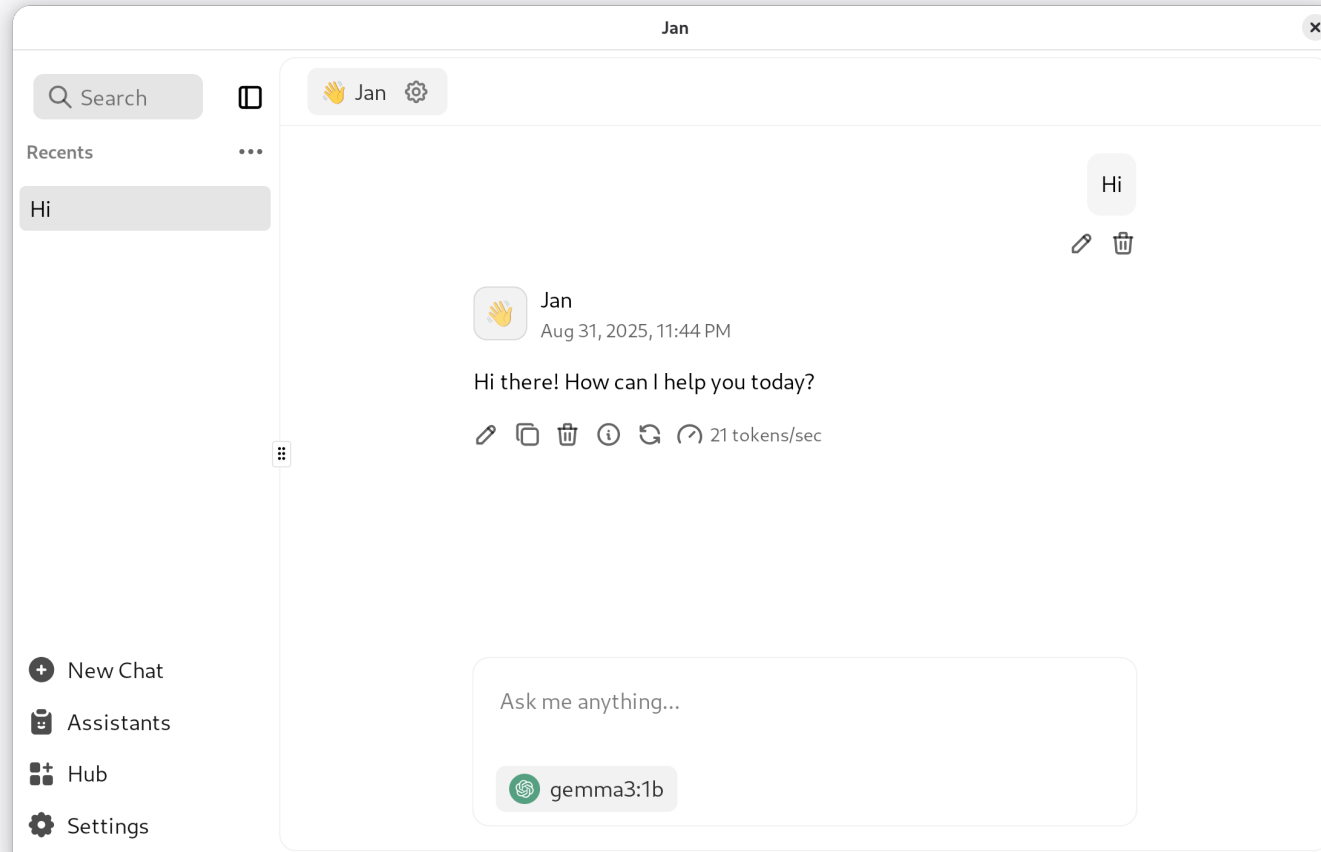


Jan: how to run

- Fetch the Debian package (or the one for your OS)

```
sudo pkg -i Jan_0.6.9_amd64.deb  
Jan
```

- Settings > Model Providers > OpenAI
- API Key "ollama"
- Base URL: <http://localhost:11434/v1>
- Models: add a new one ("+"), "gemma3:1b"
- Select the model in the Chat



Other self-hostable generative models

Producing images

- [Qwen-Image](#), Apache 2.0 (2025-08)
- [HiDream-I1](#), MIT License (2025-07)
- [FLUX.1Kontext\[dev\]](#), [models](#), Flux Non-Commercial License (2025-08)
- [Stable Diffusion](#), [models](#), StabilityAI Community License (2025-01)

[A Guide to Open-Source Image Generation Models](#)

[Text-to-image Arena](#)

Producing video

- [Wan 2.2, repo](#), Apache 2.0 (2025-08)
- [Hunyuan Video, repo](#), Tencent Hunyuan Community License
- [LTX Video, repo](#), Apache 2.0
- [Stable Video Diffusion](#), proprietary license, gratis for some uses

Text to video and image (apps & finetunes)

- [ComfyUI](#), [repo](#): front-end and UI for several self-hostable text-to-image and text-to-video models
- [Wan2GP](#): front-end and UI for several self-hostable text-to-video models
- [CivitAI](#): models and finetunes

Speech to text

Whisper, MIT License

```
uv venv
uv pip install openai-whisper
uv run whisper speech.wav --language Spanish
```

```
#!/usr/bin/python3
import whisper

model = whisper.load_model('tiny')
transcription = model.transcribe('recording.wav')
print(transcription['text'])
```

```
$ whisper speech.wav --language Spanish
```

Text to speech

- [CoquiTTS](#), [model](#), Coqui Public Model License (2023-11)

```
$ tts --text "Texto" \  
  --model_name tts_models/es/mai/tacotron2-DDC \  
  --out_path speech.wav
```

- [Kokoro](#), Apache 2.0 (2025-01)
- [Higgs Audio V2](#), Boson Higgs Audio 2 Community License (2025-07)

Text to speech (2)

- [Chatterbox](#), MIT License (2025-04)
- [MeloTTS](#) & [OpenVoice v2](#) MIT License (2024-02, 2024-04)
- [FishSpeech](#), CC Attribution-NonCommercial-ShareAlike (2025-08) (2024-11)

Exploring the World of Open-Source Text-to-Speech
Models

Other random models

- Understanding and reasoning about time series: [ChatTS](#) Apache 2.0, includes training dataset (2025-08)
- 360 immersive and explorable 3D worlds: [HunyuanWorld](#)
- Text to 3D: [LlamaMesh](#), Llama Community License (2024-11)

Other applications

- [SurfSense](#): comprehensive assistant
 - [Source code](#)
- [DeerFlow](#): comprehensive assistant
 - [Try it at VolcEngine](#)
 - [Source code](#)
- [Hyprnote](#): note taking tool for meetings
- [TransformerLab](#): Train, Tune, Chat with LLMs
 - [Source code](#)
- [MobiRAG](#): chat with PDFs in your mobile

Open training datasets

- Chatbot Arena Leaderboard (How it works)
 - Datasets
- LAION datasets
- Common Corpus: The Largest Collection of Ethical Data for LLM Pre-Training
- The Common Pile v0.1: An 8TB Dataset of Public Domain and Openly Licensed Text
- CommonCrawl Dataset
- FineWeb, 18.5T tokens cleaned from CommonCrawl

Benchmarks

- Chatbot Arena Leaderboard
- LiveBench
- LLM Leaderboard
- SWE-bench
- SWE-bench-live

Bonus track:

Are LLMs deterministic?

Why this matters?

- Reproducible research: we want experiments that can be repeated by others, with the same results
- Reproducible results: in some cases, it is important to be sure that a given input produces a given output. Always.
- Reliable debugging: for debugging a problem, exact reproduction is often needed
- Deterministic software: in some cases, we need software that is deterministic. Always.

Are LLMs deterministic?

- Once weights are settled... the network itself doesn't change
- Determinism depends on:
 - Inference engine
 - System software
 - Hardware

Inference engine

It is "regular", deterministic software...
except when it tries to be random

Inference engine: controlling randomness (API parameters)

- `seed` can be fixed (initializes the pseudo-random number generator)
- `temperature` : 0 means "greedy sampling" (most probable next token)
- `top_k` (shortlist selector pool): 1 means the pool for selectable words is 1 (the most likely)
- `top_p` (nucleus sampling): chains of tokens to be considered. Difficult to control, interferes with

Inference engine: controlling randomness

- Other parameters (eg, frequency or presence penalty) should be equal
- Beware: the software may use random number generators in some other places

Inference engine: the balance

- Randomness in inference is there for a reason: it can be useful for creativity, for getting better outputs
- Two strategies:
 - Keep randomness, but control it (controlling all seeds for randomness)
 - Remove randomness, by controlling temperature , top_k , top_p

Both can be combined

Supporting software

- Mixtures of experts: prompt tokens routed differently depending on composition of batches from different users
- Framework (PyTorch, TensorFlow): non-deterministic convolution algorithms (can be configured to be deterministic)
- Differences in compilers, GPU drivers...

Hardware

- Floating point rounding: different rounding approaches in different hardware (happens even in int-quantified models)
- Non-deterministic hardware

References

- [Achieving Consistency and Reproducibility in Large Language Models \(LLMs\)](#)
- [The Art of Sampling: Controlling Randomness in LLMs](#)
- [Controlling Randomness in LLMs: Temperature and Seed](#)
- [Solving Reproducibility Challenges in Deep Learning and LLMs: Our Journey](#)
- [Defeating Nondeterminism in LLM Inference](#)

Beware!

- This is about reproducibility settings keeping setup:
 - Keeping setup, and the same input, produce exactly the same output.
 - Changing the hardware, GPU driver version, inference engine version, etc, may change results.
- This is not about predictable results:
 - Slightly different inputs may lead to very different results, even with reproducible settings.

Summary

- Define a seed, this may be enough
- `temperature = 0`
- `top_k = 1`
- All other parameters should be equal
- Same inference software
- No mix of different inferences
- Same system software (GPU drivers, etc)
- Same hardware (CPU, GPU, etc)