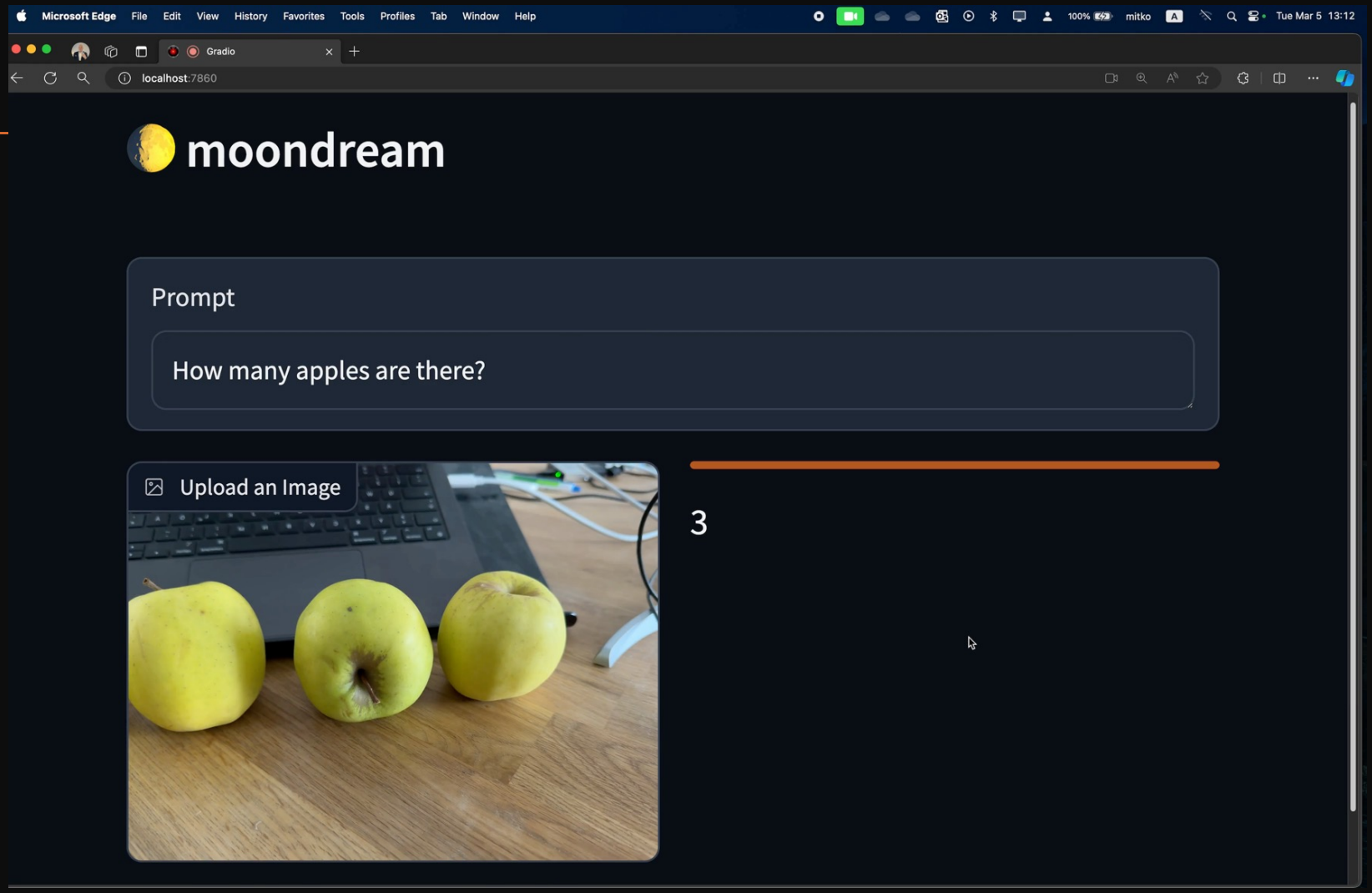
Technical deck

Friendly reminder:

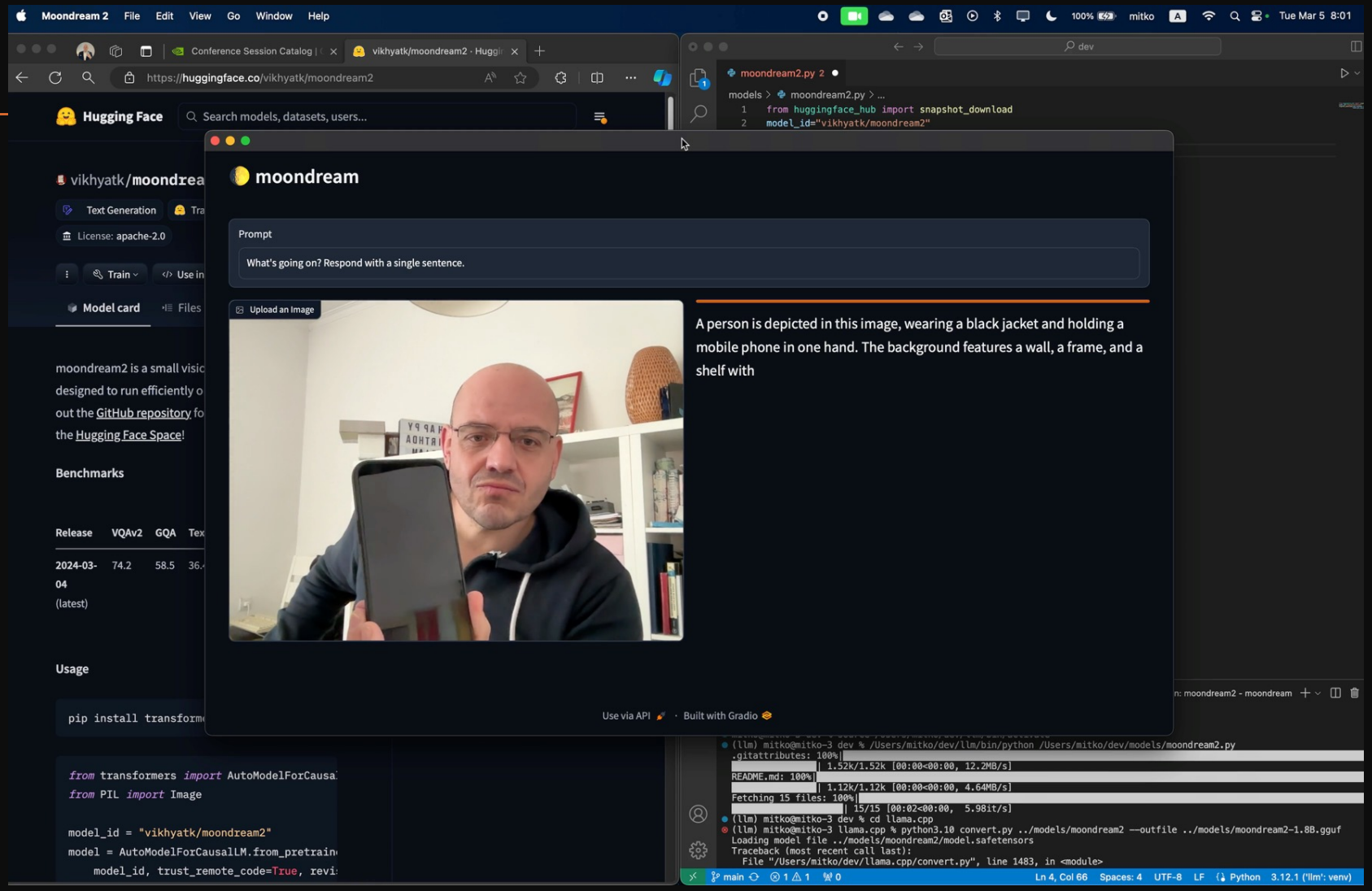
The choice of Math, ML, and AI topics we can discuss is endless.

We have one evening and will start with only the ultra hot open-source topics.

Why all the hype?



Why all the hype?



Agenda

- ❑ Practical open-source AI resources - datasets, tools, models
- ❑ How to start on your PC today
- ❑ Open AI platform architectures - from on-device to hybrid local/remote
- ❑ From PoC to pilot to production - Edge to Cloud AI platforms
- ❑ End-2-End performance optimization
- ❑ Security for AI platforms
- ❑ Beyond the wrappers, RAG, and prompt engineering - advanced AI systems engineering
- ❑ Practical use cases

AI/ML did not happen overnight

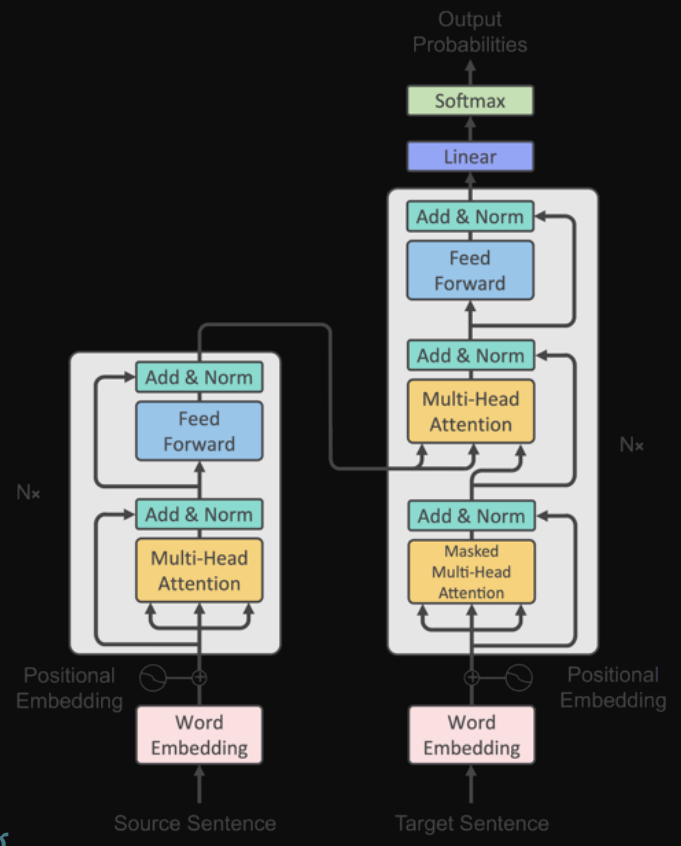
Prehistoric	• 1950s Machine Translation
Stone Age	• 1980s Knowledge-Based Systems
Bronze Age	• 1993 – 2012 - Statistical Era
Iron Age	• 2013 – 2017 - Special Purpose, Deep Learning ML
Modern Age	• 2018 – Present – Generative AI, Foundation Models, LLMs

Transformers Era

2017 – Present

"Attention is All You Need paper" in 2017

- ❑Type of Deep Neural Network
- ❑Leverages Attention/Self-Attention, including multi-head attention
- ❑Expressive: Feed-forward;
- ❑Optimizable: Backpropagation, Gradient Descent;
- ❑Efficient: High Parallelism compute graph
- ❑Examples: LLaMA-3, phi-3, GPT-4, Claude-3
- ❑Learn all from Karpathy https://www.youtube.com/watch?v=zjkBMFhNj_g



Brief New Age Tech Glossary

- **Transformers**: A type of general-purpose neural network architecture that facilitates the modeling of sequences without the need for recurrent connections, prominently used in language processing tasks
- **Foundational Model**: A large-scale model that is trained on vast amounts of data and can be fine-tuned for a variety of downstream tasks, serving as a base for further specialized models
- **Large Language Model**: A substantial neural network model trained on extensive textual data to understand and generate human-like text across many languages and contexts. **Small Language Model**: A more compact version of a language model designed for efficiency and lower resource consumption while performing natural language processing tasks
- **Visual Language Model**: A model that combines language and vision processing to understand and generate content related to both text and images
- **Multimodal Models:** AI models that can process and understand information from different types of data, such as text, images, and audio, simultaneously
- **RWKV (RwaKuv)**: A variant of a recurrent neural network, which stands for "Reduced Weight KneeV", designed for efficiency and performance in sequence modeling tasks
- **Mamba/Jamba, Hawk/Griffin, DPO, DORPO, Flash Attention…**

Where does open-source AI live

https://HuggingFace.co – models, data, research papers, AI social network, compute…

https://arXiv.org - Research Papers

https://Github.com – All the source code in one Place

https://github.com/ggerganov/llama.cpp - local AI on your CPU, GPU „Хайде наште!“

https://Discord.com – almost all projects have a channel

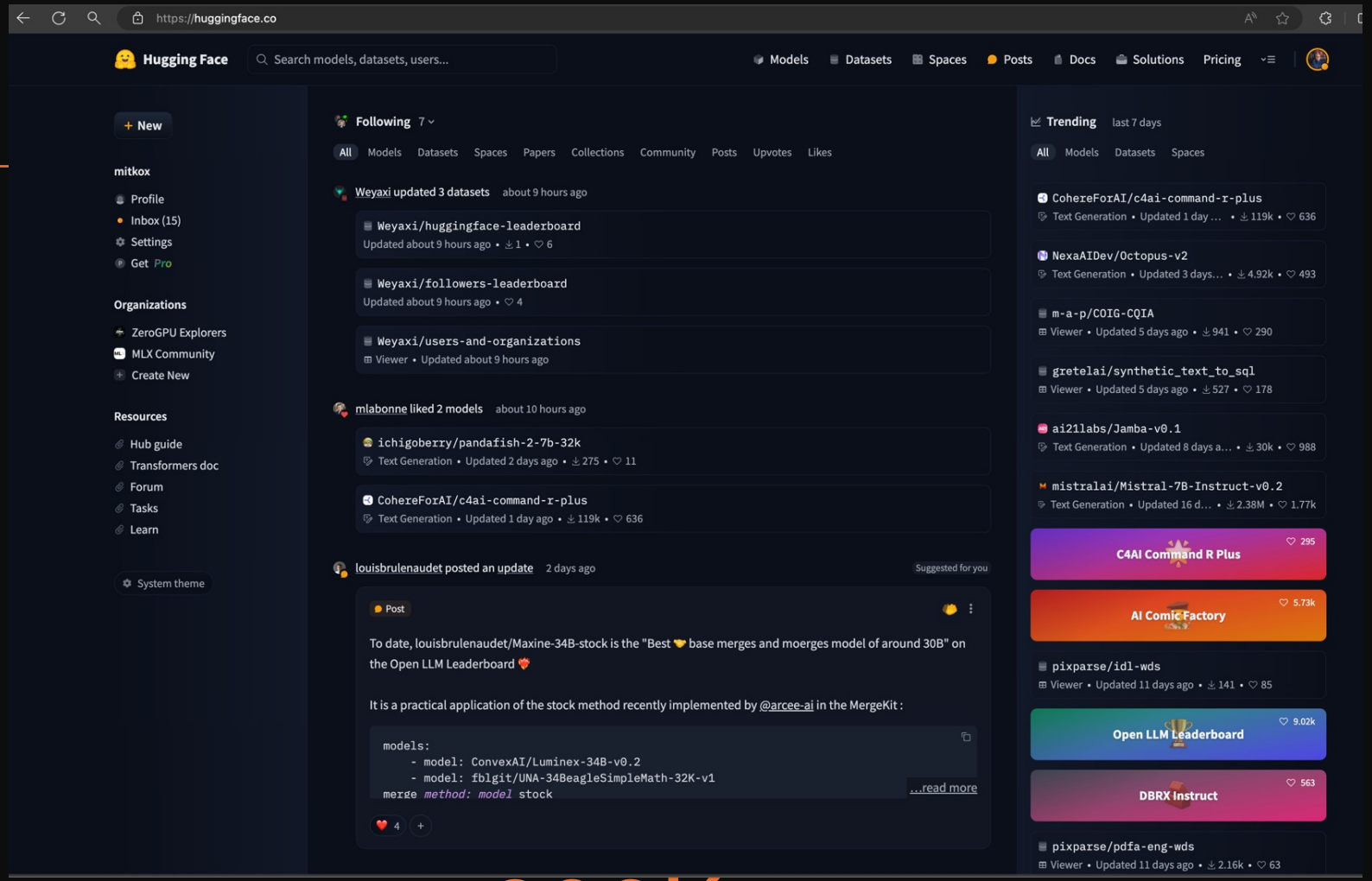
https://x.com – social network for emerging AI/ML devs, researchers, companies

https://colab.research.google.com/ - ‘Free’ compute and managed Jupyter notebooks

WTF is Hugging Face?

🤗 base slides by Thomas Wolf

Hugging Face: The home of open AI/ML



Founded In
2016

170
Employees

300K+
stars on Github

600K+
open source models

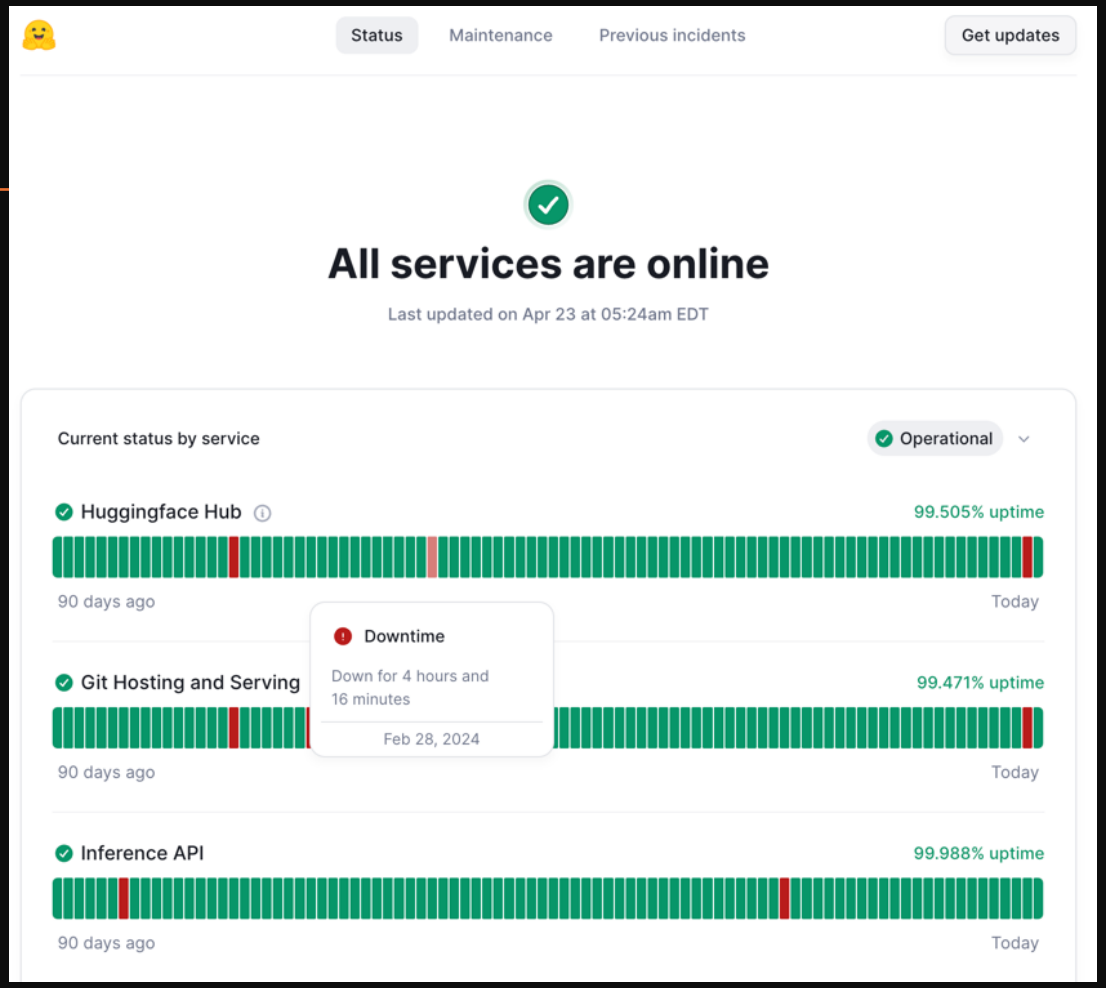
130K+
public data sets

1M+
daily downloads

700K+
daily visitors

30+
Libraries

When the AI World Stopped



Founded In
2016

170
Employees

300K+
stars on Github

600K+
open source models

130K+
public data sets

1M+
daily downloads

700K+
daily visitors

30+
Libraries

Used everywhere in the AI world

15,000+ startups and enterprises

grammarly, HUAWEI, Dropbox, intuit, Spotify, T-Mobile, orange, databricks, Apple, servicenow, Meta, Microsoft, ORACLE, NVIDIA, salesforce, Bloomberg, HubSpot, Klarna, Uber, SAMSUNG

Open-source contributors

Meta, MISTRAL AI_, stability.ai, Microsoft, Ai2, OpenAI

Hugging Face

Cloud partners

aws, Azure, NVIDIA, CLOUDFLARE

Hardware partners

NVIDIA, AMD, aws, intel, habana, GRAPHCORE, Qualcomm

On-prem partners

DELL, IBM

The Global AI/ML Ecosystem of

Model in production

Managed Inference on AWS, Azure and GCP

Inference Endpoints

Inference API

Deploy anywhere

Spaces

Hosted ML applications

NVIDIA DGX Cloud

Hugging Face on Azure

Amazon SageMaker

Google Cloud

Cloud Platforms

Open Datasets

130,000+ datasets on the hub

600,000+ models on the hub

Open Models

auto TRAIN

Transformers

Accelerate

Diffusers

Optimum

No-code AutoML

intel AMD aws habana

HW-accelerated training & inference

Open-Source Ecosystem

- **Transformers**
  State-of-the-art ML for Pytorch, TensorFlow, and JAX.
- **Datasets**
  Access and share datasets for computer vision, audio, and NLP tasks.
- **Gradio**
  Build machine learning demos and other web apps, in just a few lines of Python.
- **Safetensors**
  Simple, safe way to store and distribute neural networks weights safely and quickly.
- **Transformers.js**
  Community library to run pretrained models from Transformers in your browser.
- **Hub Python Library**
  Client library for the HF Hub: manage repositories from your Python runtime.
- **Diffusers**
  State-of-the-art diffusion models for image and audio generation in PyTorch.
- **Accelerate**
  Easily train and use PyTorch models with multi-GPU, TPU, mixed-precision.
- **TRL**
  Train transformer language models with reinforcement learning.
- **timm**
  State-of-the-art computer vision models, layers, optimizers, training/evaluation, and utilities.
- **PEFT**
  Parameter efficient finetuning methods for large models

Back to OSS license!

- **Text Generation Inference**
  Toolkit to serve Large Language Models.

Open vs Closed Models

Open and closed models have different benefits and should be considered for each use-case

	Open-Source	Closed/Proprietary
Security	Models can be **self-hosted**, data stays in your environment	Models cannot be self-hosted. Data is **sent outside** your environment to vendor(s)
Control	The lifecycle is **controlled** by you	Updates and changes to performance are happening **without notice**
Customization	**Open Weights and sometimes open code access** to customize the model for your needs	**Limited ability to customize** for your needs
Transparency	**Inspect code and data** provides better auditability and understandability	**No ability to audit** or understand performance
Cost	Typical **lower long term cost** due to smaller model size	larger model size and **proprietary premium** often balanced by **decreased** cost from **server-side optimization**
Latency	**Lower latency** due to on premise and smaller model sizes	Often **greater latency** due to larger model sizes + API latency
Quality	No single approach is best. Each use case will vary. Proprietary is typically **closer to the frontier of performance**.	
Examples	OpenAI, Meta, salesforce, FLAN-T5, Mistral AI, Microsoft	OpenAI, ANTHROP\C

Energy/carbon footprint and LLMs

Start by test existing models on your domain and task(s) of interest

Do you **need** to pretrain an LLM?

Most of the time the answer is “no” =>

Focus on **efficient fine-tuning and inference**

Energy budget will likely be dominated by inference costs.
Select a **compute efficient** model:
- smallest size
- quantized
- classification models > generative

Deploy it in an on-prem setup/cloud provider in a region with a **good energy mix**

Yes =>

Focus on **efficient pretraining** while taking a holistic view of **model life-cycle**

Train-compute-optimal models (Chinchilla law) are not efficient for inference

Train **smaller models** for longer if you plan to deploy it at large scale

Train in a local cluster/provider with a **good energy mix**

Share the model so people can **re-use**/leverage the compute spent – *It’s like recycling AI models*

Ressources:
- Power Hungry Processing: Watts Driving the Cost of AI Deployment? https://arxiv.org/abs/2311.16863
- Language models scale reliably with over-training and on downstream tasks https://arxiv.org/abs/2403.08540
- Region energy mix (e.g. solar, nuclear, coal, gas) can have a x500 impact on model carbon footprint: https://app.electricitymaps.com/

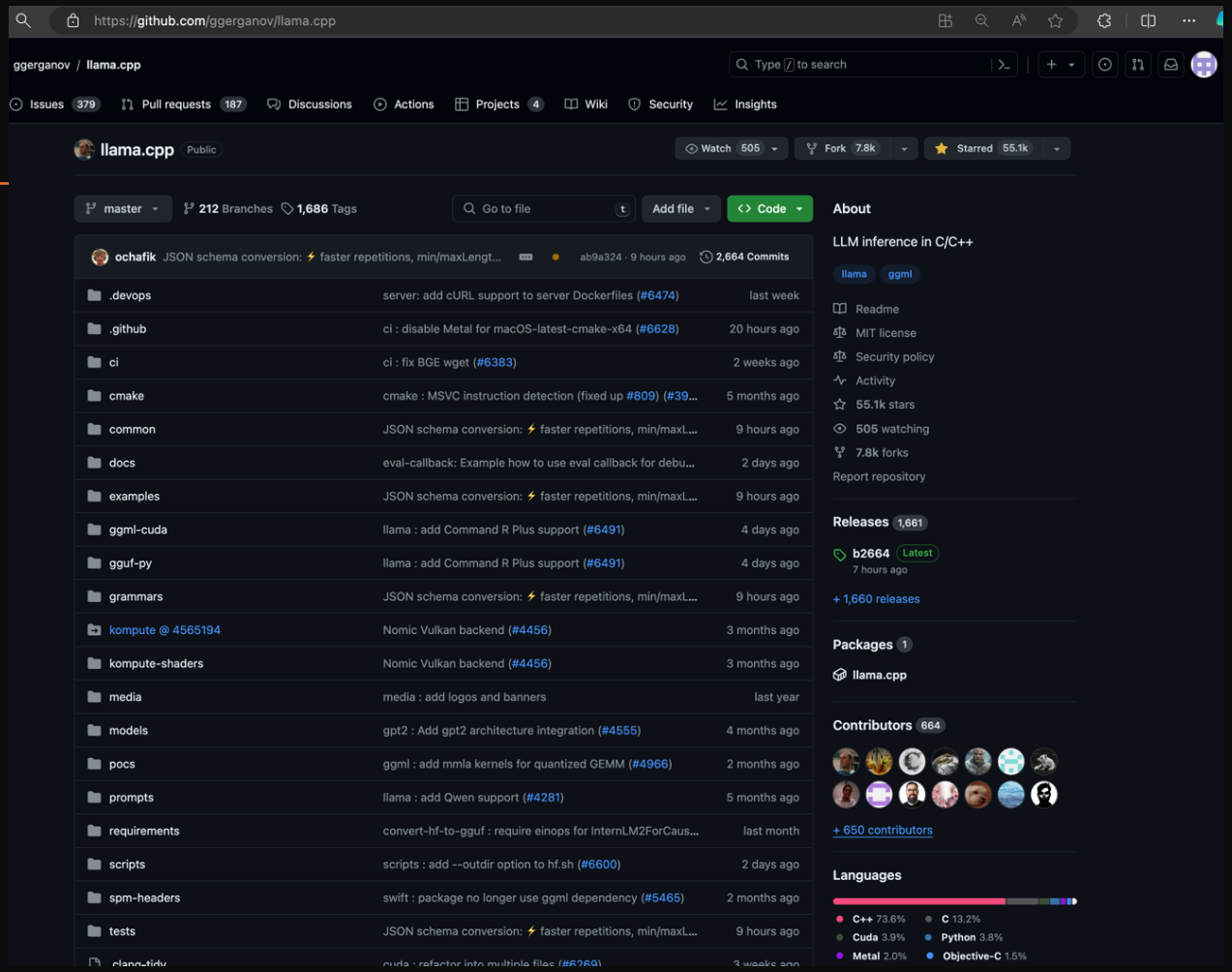
On device AI

On device AI Comparison

Open Source, Apple Intelligence, Microsoft Copilot+ PC

Feature/Aspect	Open-Source	Apple Intelligence	Windows Copilot+ PCs
Provider	Community	Apple	Microsoft
Primary Functionality	AI-powered enhancements, standalone and integrated AI functionalities in any device	AI-powered enhancements and integrated AI functionalities on Apple devices	AI-powered enhancements and integrated AI functionalities in Windows PCs
Platform Integration	Cross-platform (Linux, Android,macOS, iOS, Windows)	iOS 18, iPadOS 18, macOS 15 and newer	Windows 11 24H2 and newer
Key Technologies	Llama.cpp, GGUF, MLX, TensorFlow, PyTorch, Transformers, DSPy, CPU/GPU/NPUs	Small Foundation Models, transformers and diffusion, LoRA Adapters, Core ML, Natural Language API, Vision API, Apple Neural Engine	Windows Copilot Runtime, AI frameworks (DirectML, ONNX Runtime), Phi Silica models, NPUs
Language Support	C, C++, Zig, Python, Go, C#, JavaScript, TypeScript, node.js	Swift, Objective-C, Python, JavaScript	C#, C++, Python, JavaScript, TypeScript
AI-powered Features	Consumer and enterprise, text, audio and images	Consumer focused text, audio and images	Consumer and enterprise, text and images
Natural Language Processing	Advanced (open weight models)	Advanced (used in Siri, Apple Translate)	Advanced (Phi Silica model, other SLM)
AI/ML Models	Open source and open weight models, GGUF	Custom Apple AI/ML models, Core ML	Custom Microsoft AI/ML models, Phi Silica, ONNX
Privacy and Security	On-device inferencing, community-reviewed security practices	on-device/cloud inferencing, remote attestation	On-device/cloud inferencing,
Collaboration Features	Collaborative development on GitHub, GitLab, community forums	Limited (Focus on individual user experience)	Integrated collaboration tools within Windows ecosystem
Customizability	Highly customizable and extensible with open-source code and APIs	Limited customization by end-users	Highly customizable with various APIs, libraries, and tools
Primary IDE Support	Neovim, Visual Studio Code, Jupyter Notebooks	Xcode	Visual Studio, Visual Studio Code, other major IDEs
Learning and Adaptation	Learns from data, customizable training processes	Learns from user's device interactions	Learns from user's system usage and coding patterns
API Access	Extensive APIs and libraries (TensorFlow, PyTorch, Hugging Face)	Available for developers via Core ML and other APIs	Extensive APIs in Windows Copilot Library, AI frameworks
Documentation Assistance	Extensive developer documentation	Limited (basic code documentation)	Detailed code documentation generation, productivity tips
Cost	Free and open-source, with optional paid support and enterprise features	Included with Apple devices and services	Integrated with Windows, additional features via subscription
User Base	Developers, researchers, businesses, hobbyists	General consumers and developers using Apple devices	Developers, business users, general consumers using Windows PCs
Updates and Support	Community-driven updates, frequent releases, LTS versions	Regular updates with new OS releases	Frequent updates via Windows Update, GitHub, and Visual Studio
Device Compatibility	Compatible with a wide range of devices across platforms	Exclusively Apple devices	Compatible with a range of Windows AI PCs (QCOM, Intel, AMD)
Integration with Development Tools	Integrates with a wide range of development tools and environments	Limited to Apple ecosystem tools	Integrates seamlessly with GitHub, Visual Studio, and other development tools
Offline Capabilities	Full support with on-device and offline processing capabilities	Not possible, on-device processing but always needs Apple Private Cloud Compute	Not possible, primary Azure with on-device capabilities using NPUs
Availability	Complete	Q4 2024, EU 2025+	Partial, full expected 2025+

llama.cpp (Made in Sofia)



LLaMA C++

Started In
2023

664
Contributors

55.1K+
stars on Github

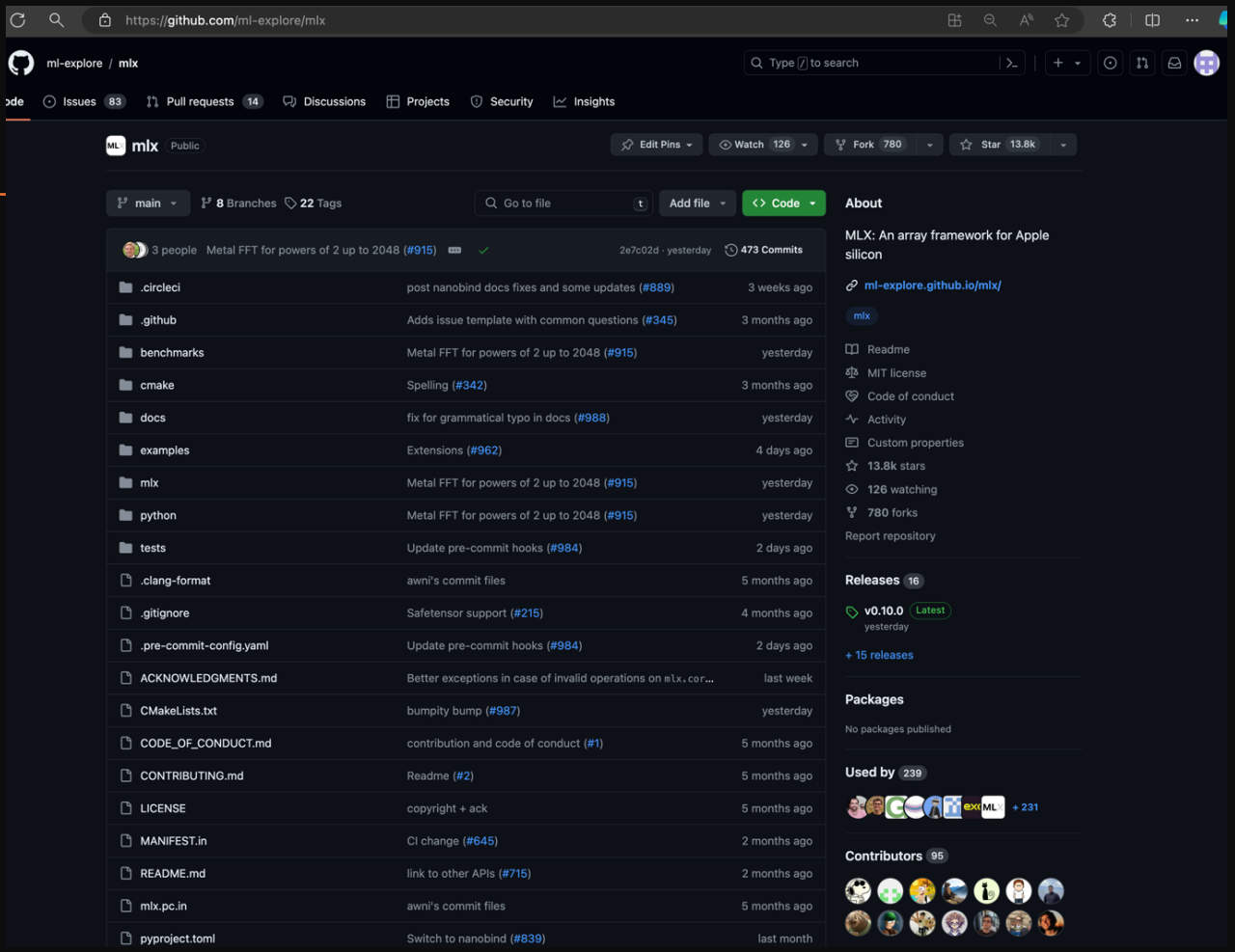
180+
Active PRs

50+
Other project Integrations

40+
Examples

7.8K+
Forks

MLX (Apple owned Open Source)



Started In
2023

95
Contributors

13.8K+
stars on Github

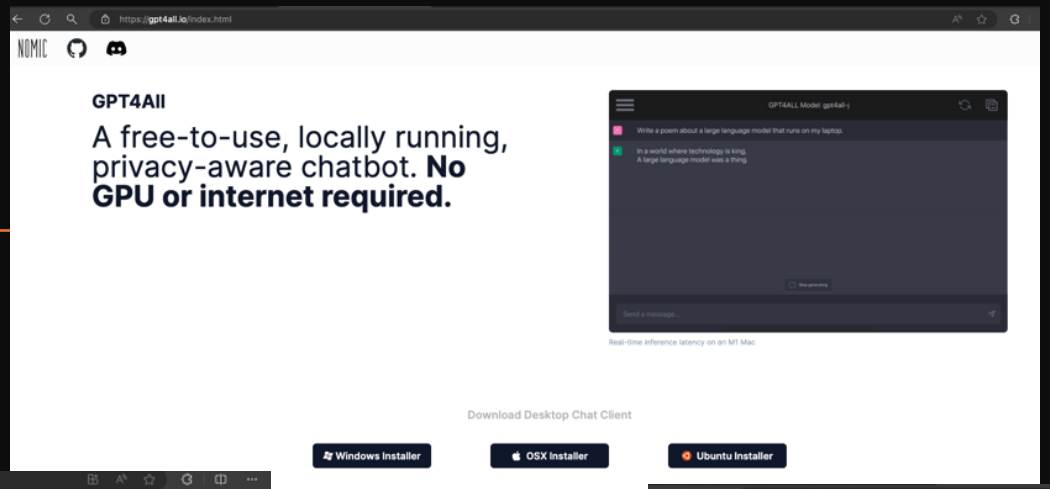
10+
Active PRs

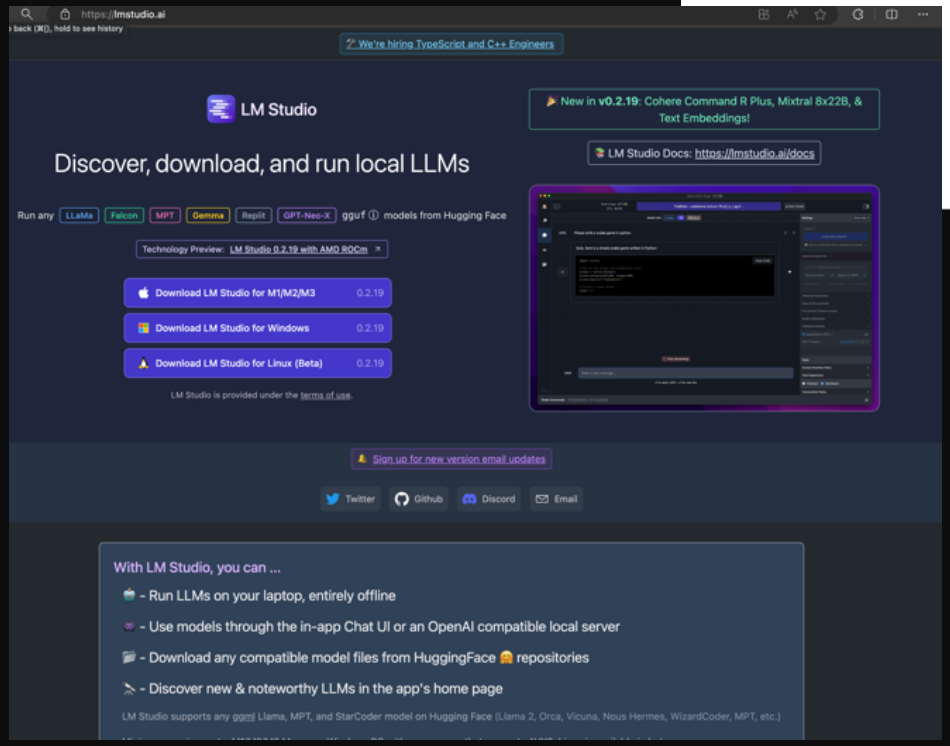
20+
Other project Integrations

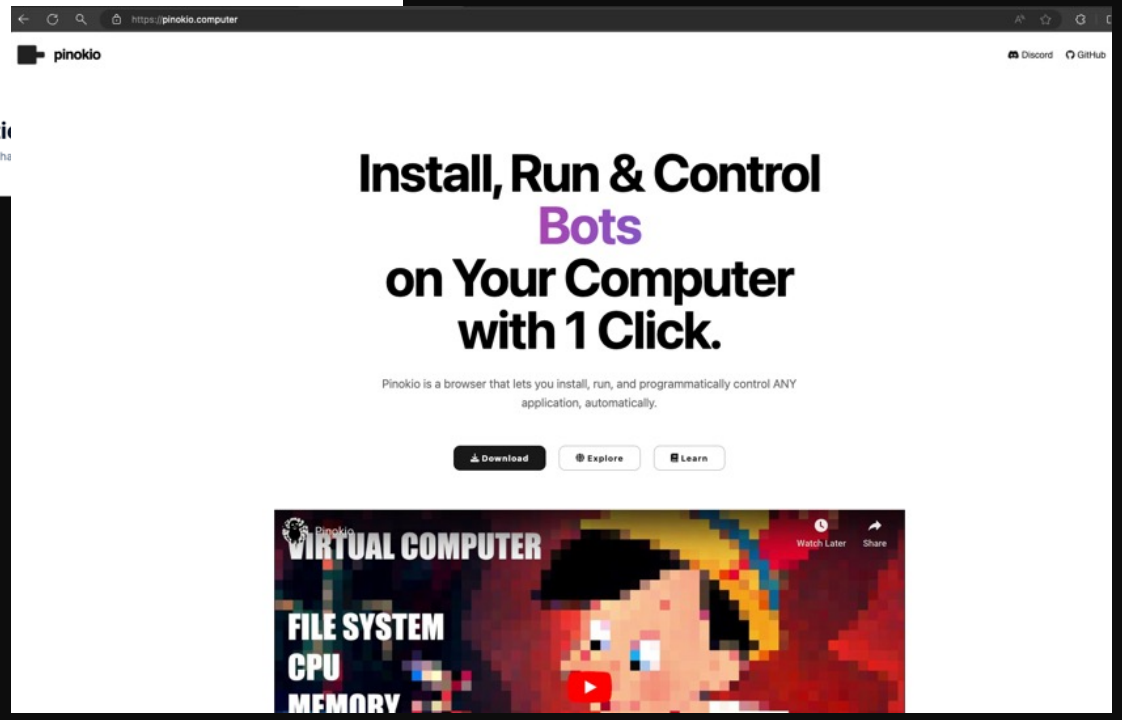
15+
Examples

770+
Forks

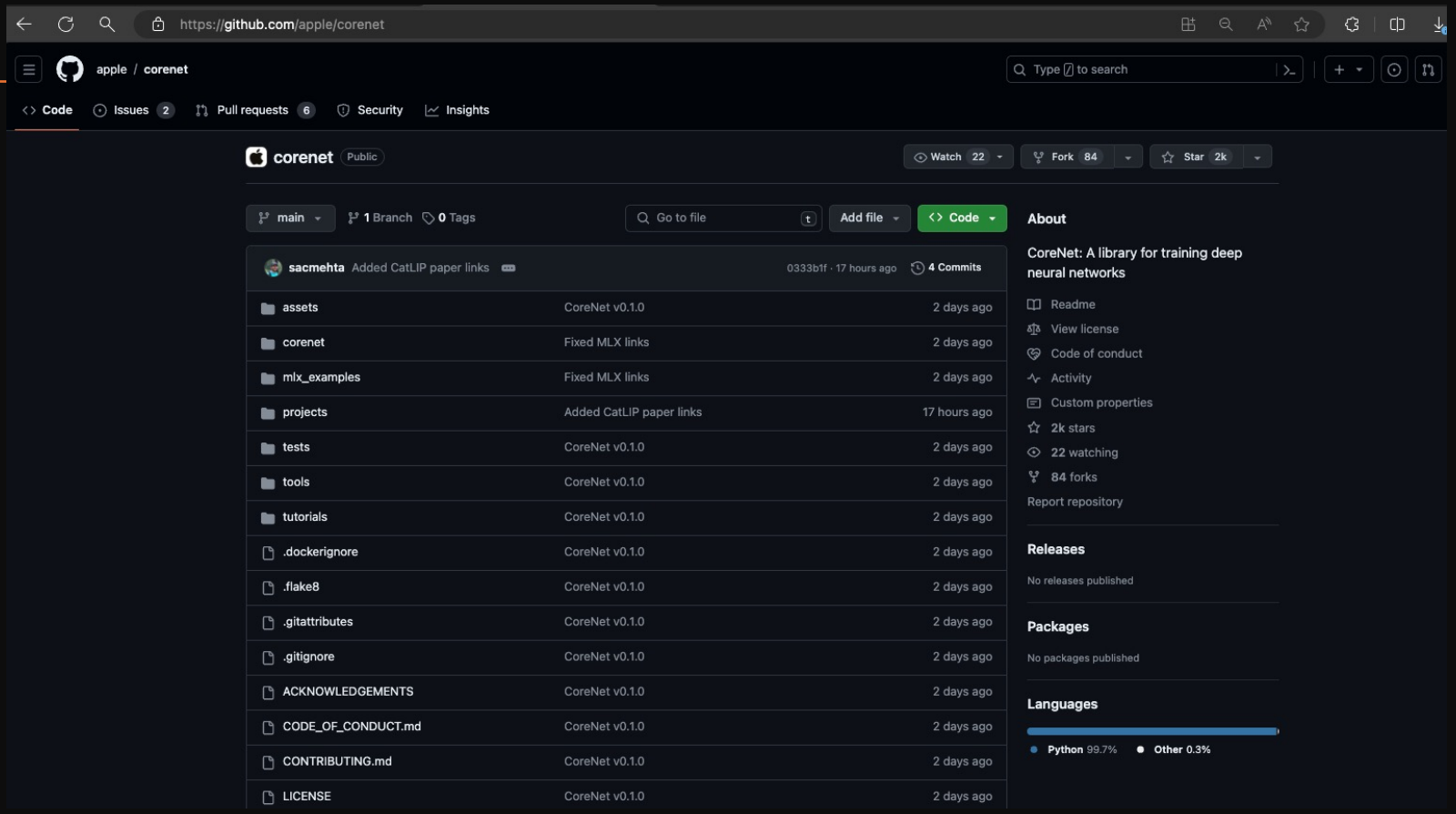
One Click Tools







CoreNet: Train SLM on your Mac



Started In
2024

2K+
Stars on Github

6+
Active PRs

3
Other project Integrations

3
Examples

84
Forks

Mergekit Evolve

https://github.com/arcee-ai/mergekit

arcee-ai / mergekit

Code | Issues 101 | Pull requests 11 | Actions | Security | Insights

mergekit Public

Watch 42 | Fork 272 | Starred 3.4k

main | 20 Branches | 4 Tags | Go to file | Add file | Code

cg123 Fix mergekit-evolve crash when not using wandb (#291) b4136ba · 12 hours ago 178 Commits

.github/workflows	Correct branch references in pre-commit.yml	3 months ago
docs	Link to list of available eval tasks (#289)	2 days ago
examples	mergekit-mega: compound merging using multiple yaml d...	3 months ago
mergekit	Fix mergekit-evolve crash when not using wandb (#291)	12 hours ago
tests	Generic Sparsification Rescaling (#245)	3 weeks ago
.gitignore	Initial commit	8 months ago
.pre-commit-config.yaml	Tokenizer merge fix (#73)	3 months ago
LICENSE	Create mergekit package	6 months ago
README.md	Update README.md w/ correct git clone URL (#278)	2 days ago
notebook.ipynb	Revert #67	3 months ago
pyproject.toml	Add `mergekit-evolve` for parameter space evolutionary al...	3 days ago

README | LGPL-3.0 license

mergekit

`mergekit` is a toolkit for merging pre-trained language models. `mergekit` uses an out-of-core approach to perform unreasonably elaborate merges in resource-constrained situations. Merges can be run entirely on CPU or accelerated with as little as 8 GB of VRAM. Many merging algorithms are supported, with more coming as they catch my attention.

Features:

- Supports Llama, Mistral, GPT-NeoX, StableLM, and more
- Many merge methods

About

Tools for merging pretrained large language models.

llama | llm | model-merging

Readme
LGPL-3.0 license
Activity
Custom properties
3.4k stars
42 watching
272 forks
Report repository

Releases

4 tags

Packages

No packages published

Contributors 17

+ 3 contributors

Languages

Python 99.2%
Jupyter Notebook 0.8%

Started In
2023

16
Contributors

3.4K+
Stars on Github

10+
Active PRs

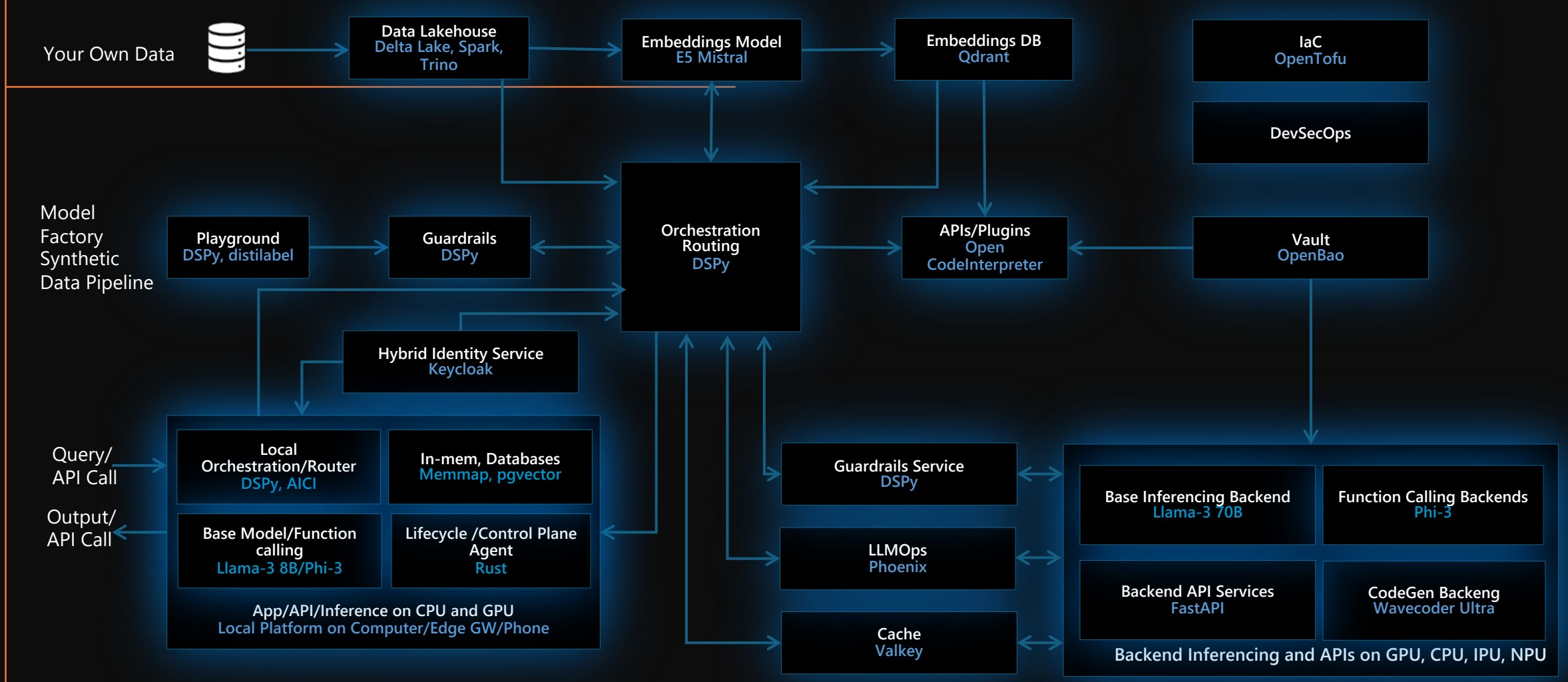
10+
Other project Integrations

5
Examples

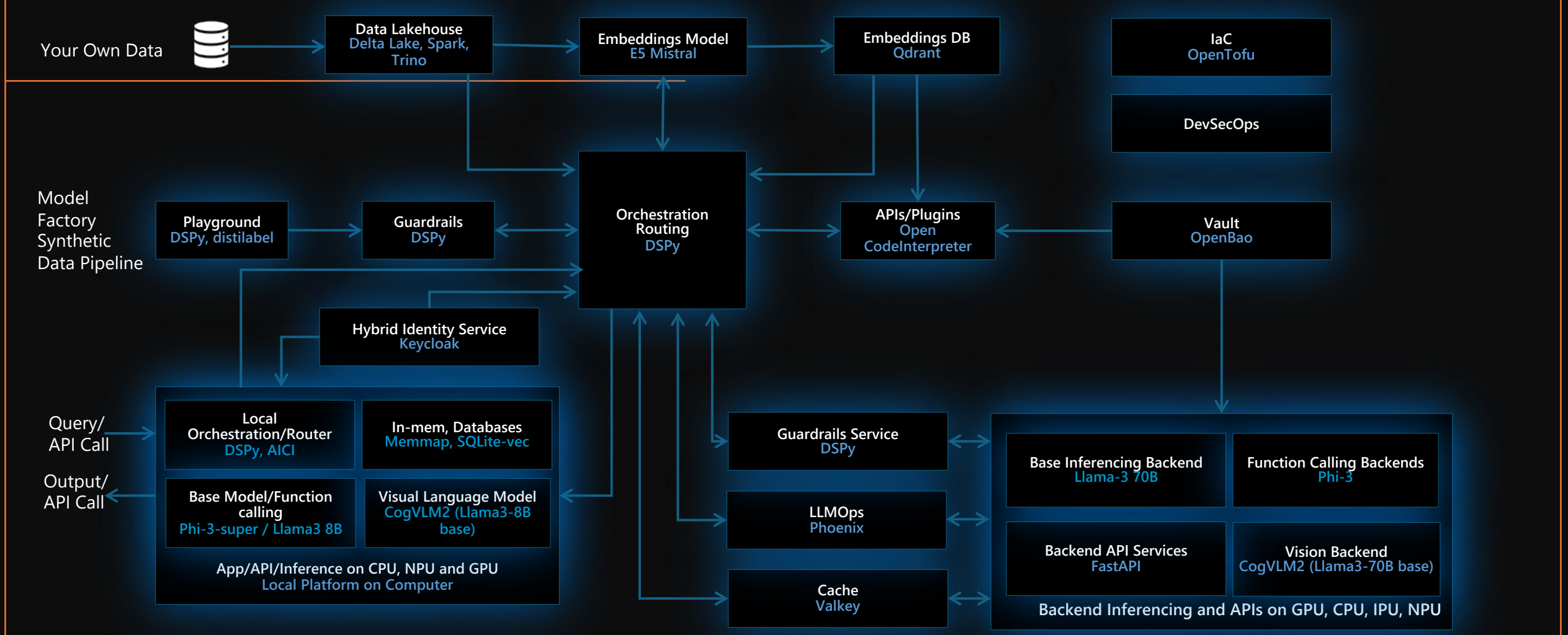
272
Forks

Open AI Platform Architecture

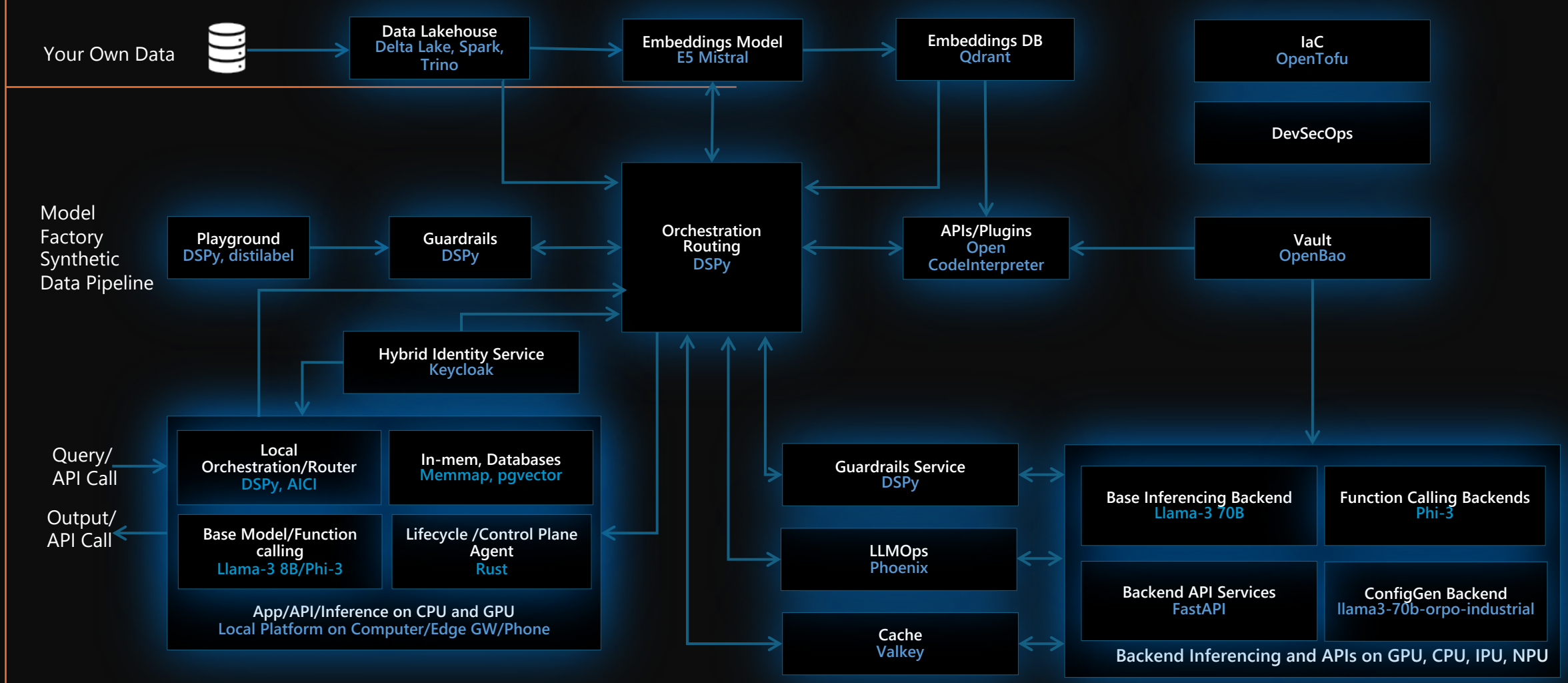
Open AI Platform



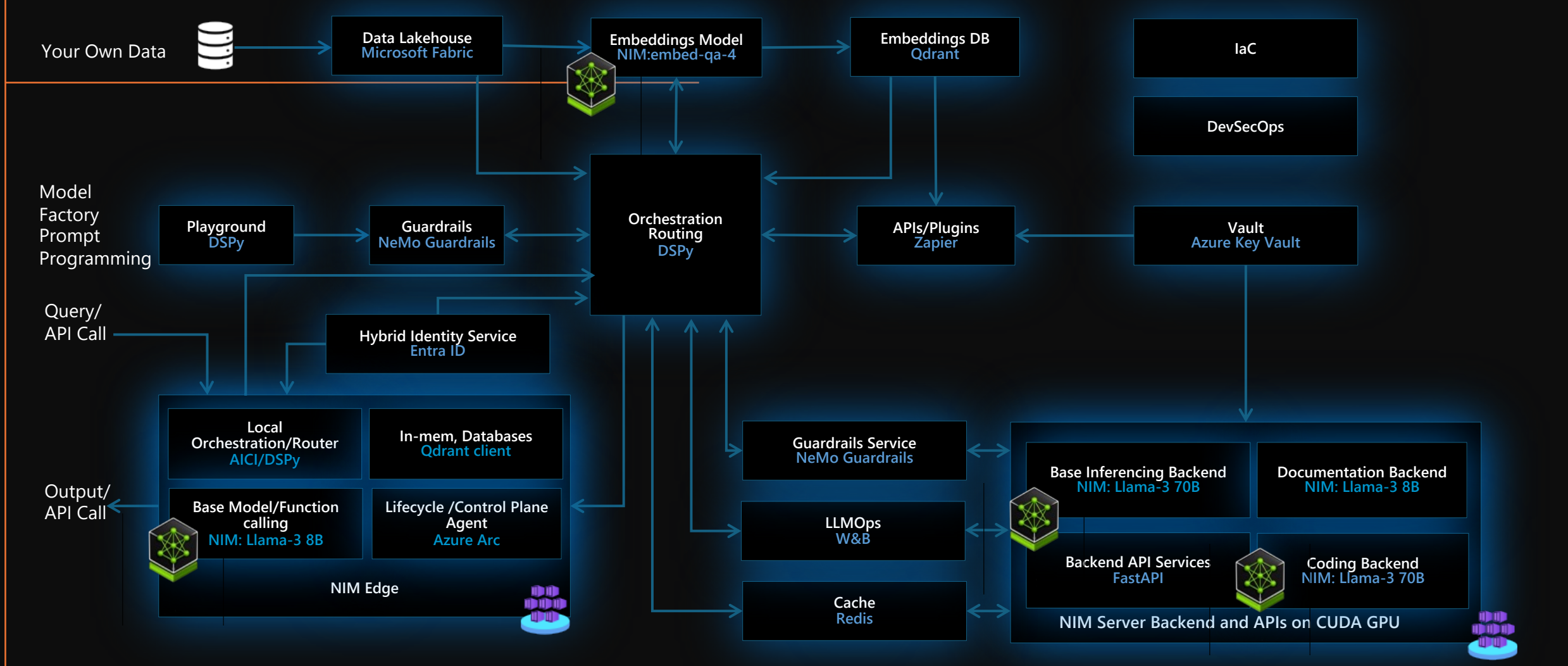
Copilot++ Decentralized Hybrid AI Platforms



Open AI Project Team



Production System with NVIDIA/Microsoft



From Proof of Concept to Pilot to Production

Lessons Learned

- ❑ Set expectations
- ❑ Minimize risks
- ❑ Always experiment and build with the North Star to take it to production
- ❑ Work 3x faster from product start to launch to happen in 6 months

Set Expectations

Building cool demos with GenAI is easy
Building an industrial or enterprise product with GenAI is hard

- ❑ If you want cool demos to show everyone externally that you're ahead of the curve, just do it!
- ❑ If you want your team to experiment and build out AI muscles for production, just do it!
- ❑ If you want a product, build data, get compute and train talents to build it, and just do it!

There are a lot of things GenAI can do

Q: But can these things meaningfully transform your customers' business?

A: Unclear

There are a lot of things Generative AI can't do NOW

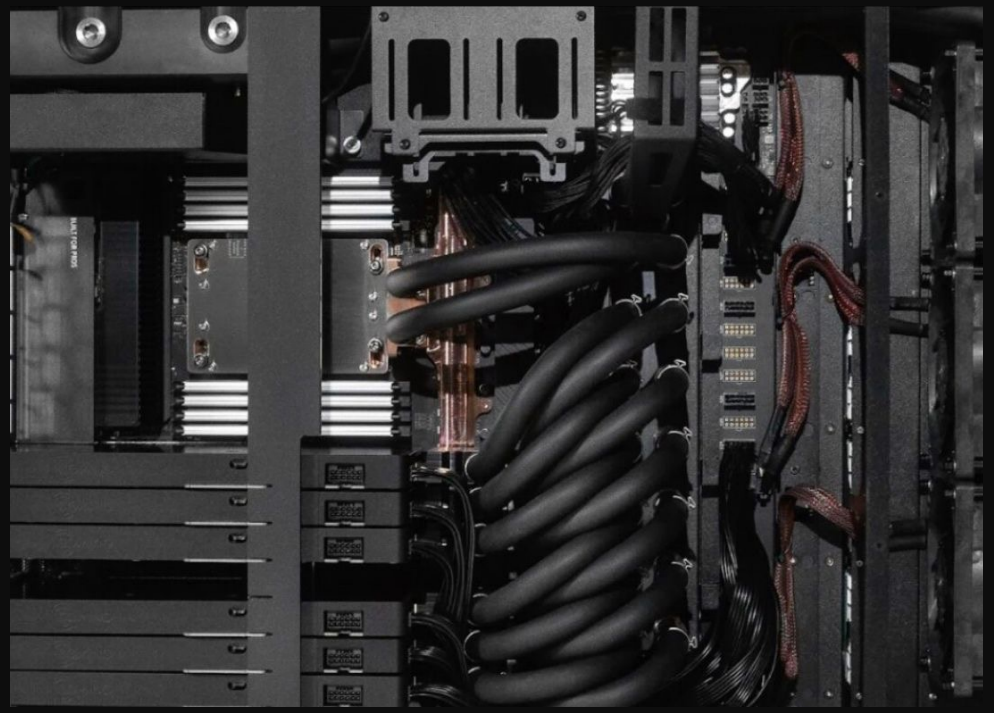
Q: But would GenAI still not be able to do those in the future?

A: Unclear

"When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong."
- Arthur Clarke

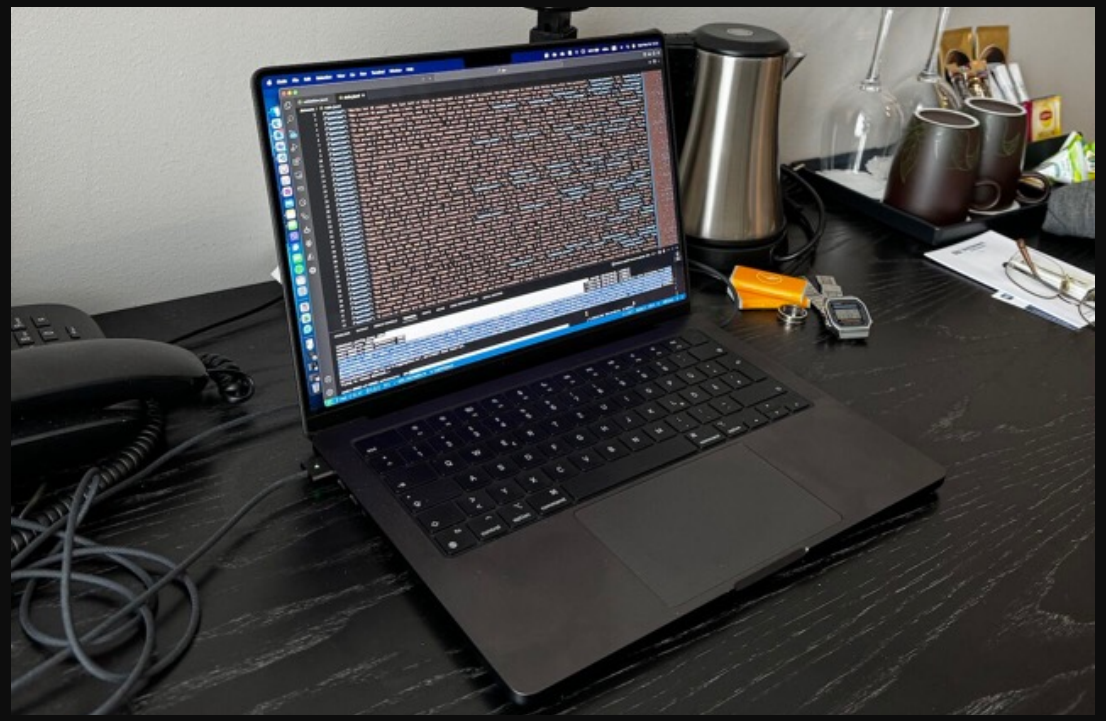
End-2-End Performance Optimization

Local AI Platform - Which Way?



Recommended Enthusiasts Hardware:
Ryzen CPU
64GB RAM
3090-RTX
1TB SSD

Recommended Pros Hardware:
Ryzen CPU
256GB RAM
6x4090-RTX with P2P Kernel
4TB SSD

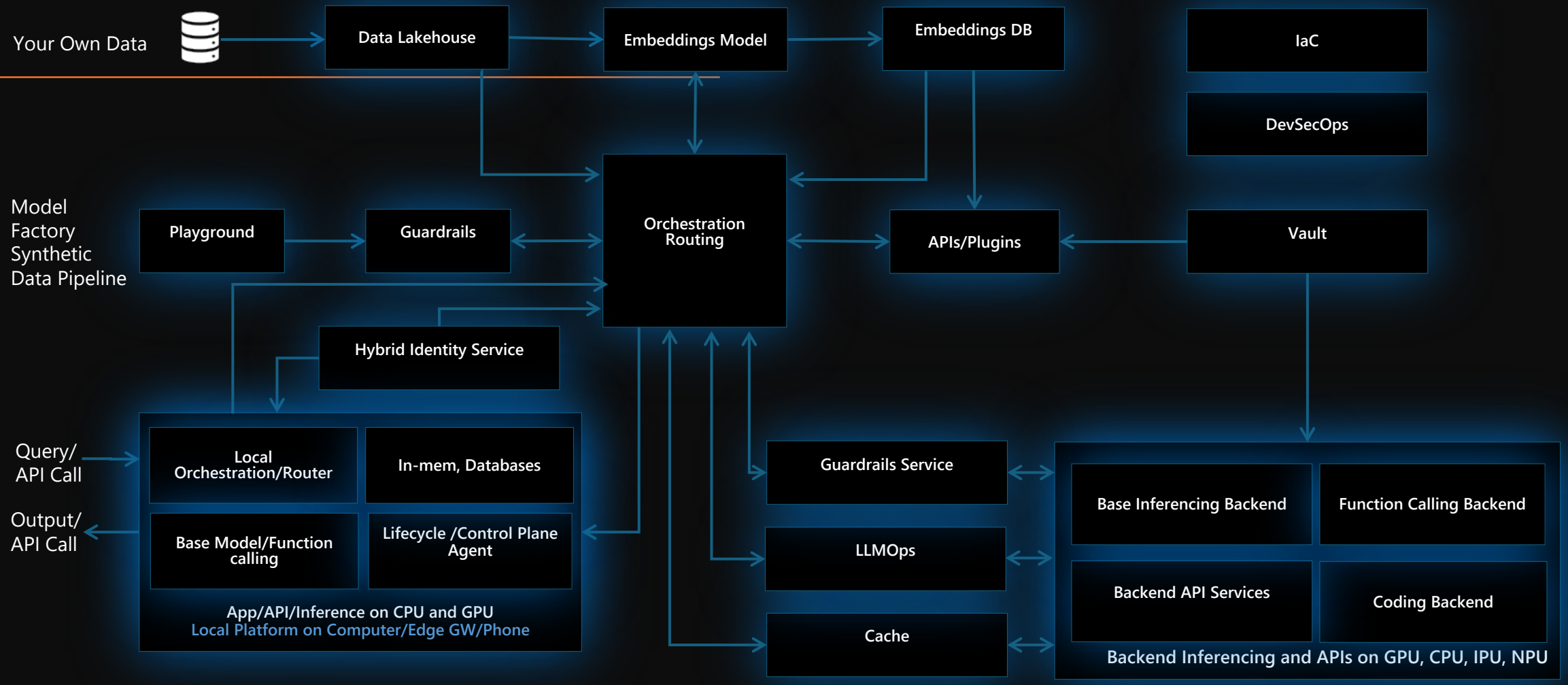


Recommended Enthusiasts Hardware:
MacBook M2/M3
16GB RAM
1TB SSD

Recommended Pros Hardware:
MacBook M3 Max
128GB RAM
4TB SSD

When Local AI Platform is not Enough

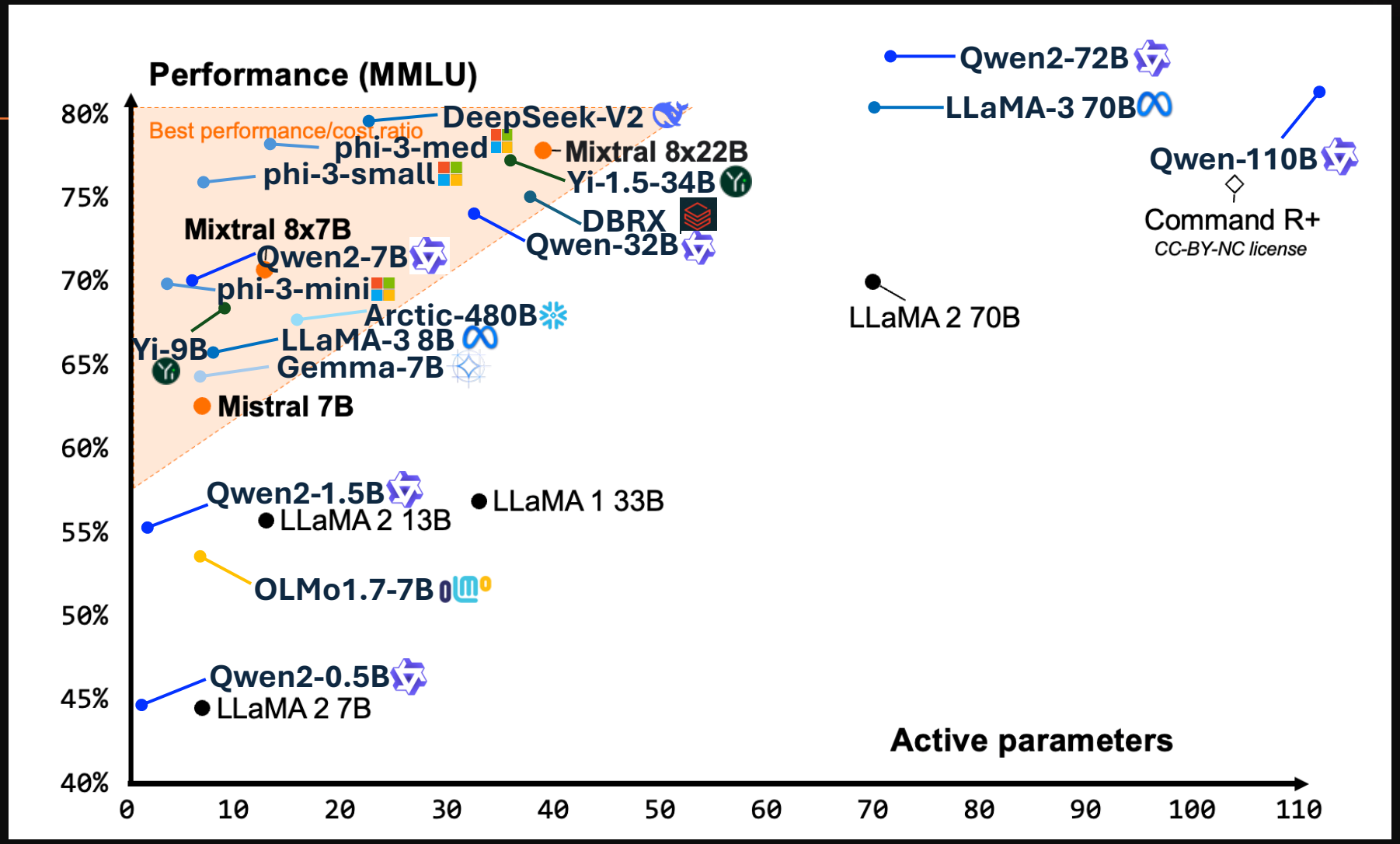
Build Your Own Multi-user AI Platform



Lessons Learned

1. Choose the Best Models for your Use Cases
2. Balance fine-tunning and Data Pipelines
3. Activate only when you need
4. Inference Quantized Models
5. Use all available Hardware – Edge to Core
6. Make everything OSS Plug&Play
7. Shorten the Lifecycle of PoC-Pilot-Production

Performance vs Compute Energy



Cloud Inferencing Race to the Bottom

Meta: Llama 3 70B Instruct

Chat

meta-llama/llama-3-70b-instruct

Updated Apr 18 | 8,192 context | $0.81/M input tkns | $0.81/M output tkns

Meta's latest class of model (Llama 3) launched with a variety of sizes & flavors. This 70B instruct-tuned version was optimized for high quality dialogue usecases.

It has demonstrated strong performance compared to leading closed-source models in human evaluations.

To read more about the model release, click here. Usage of this model is subject to Meta's Acceptable Use Policy.

Standard variant | Nitro variant

Prices per 1M tokens 26/Apr/24

Model weights

Providers | Apps | Activity | Parameters | API

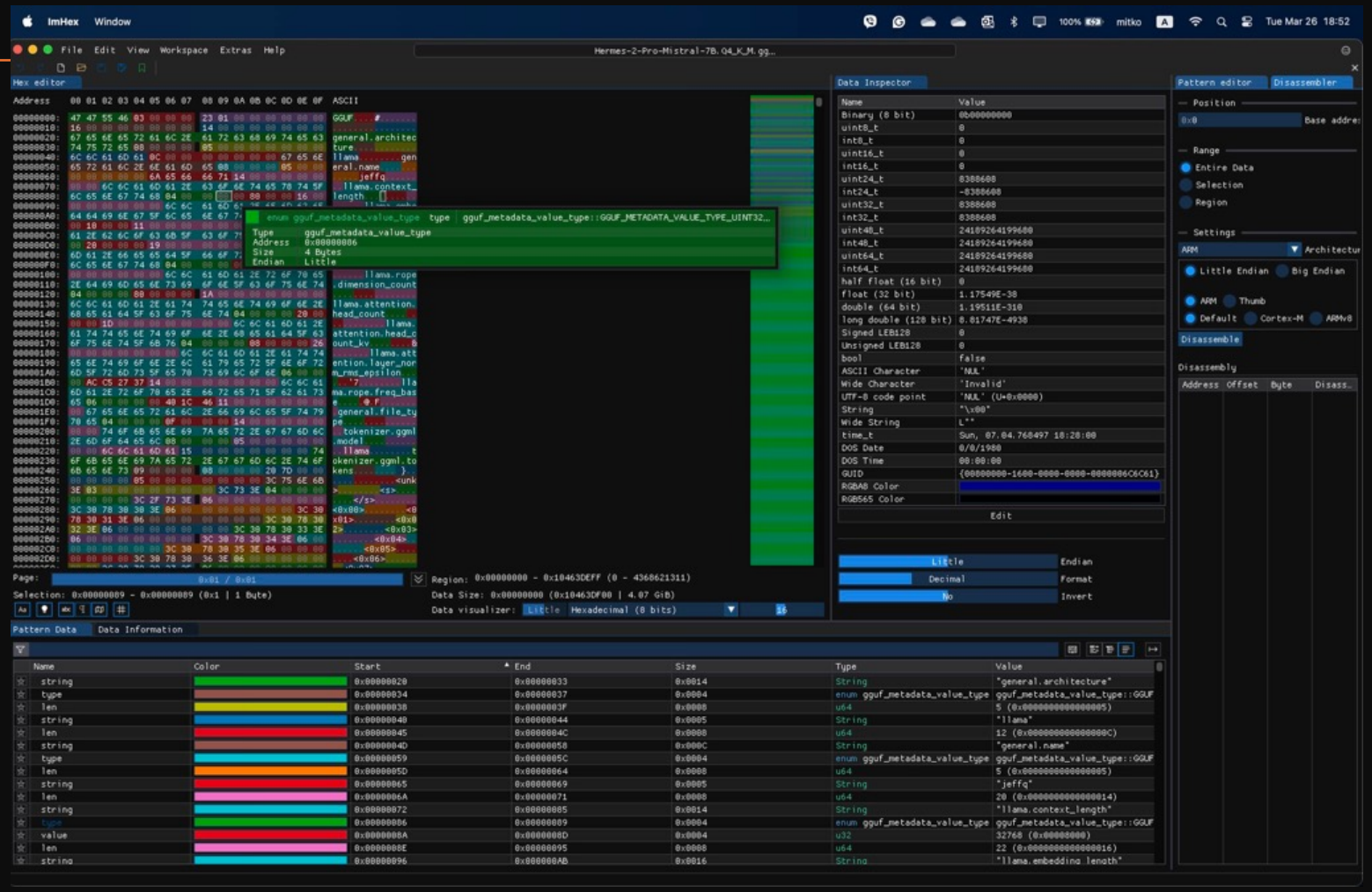
OpenRouter attempts providers in this order unless you set dynamic routing preferences. Prices displayed per million tokens.

	Max Output	Input	Output	Latency	Throughput
Together	8,192	$0.81	$0.81	0.75s	61.09t/s
Fireworks	8,192	$0.9	$0.9	0.32s	148.65t/s
DeepInfra	8,192	$0.59	$0.79	0.96s	33.18t/s
NovitaAI	8,192	$0.8	$0.8	2.16s	36.95t/s
Perplexity	8,192	$1	$1	--	--

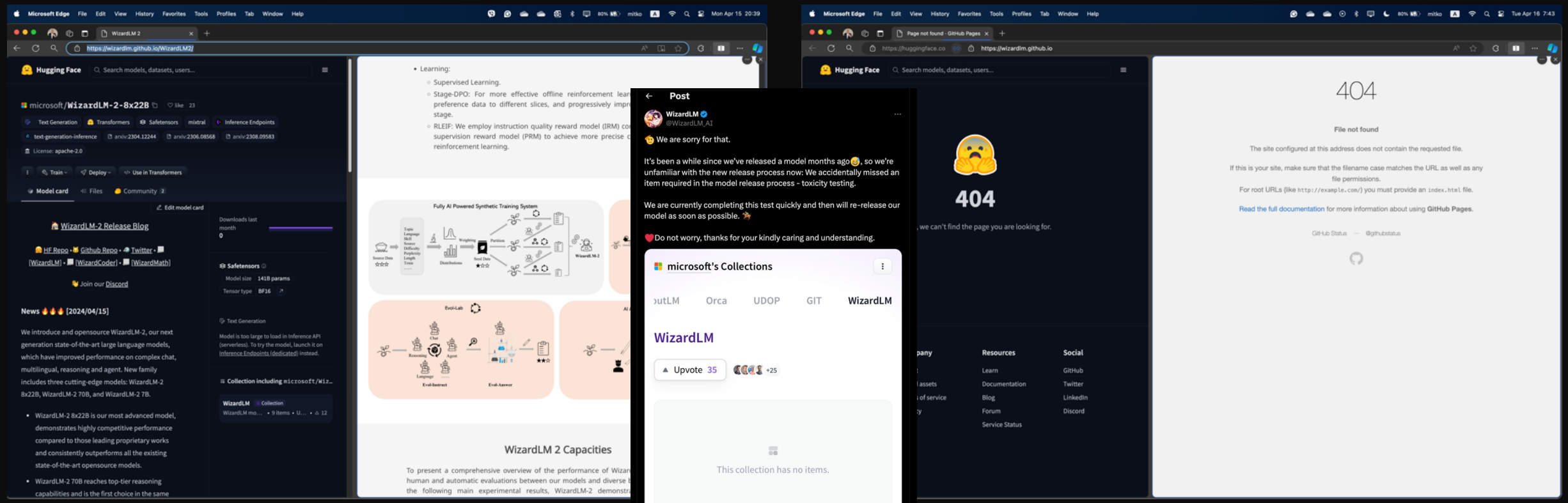
Open AI Platform Security

All Cybersecurity Best Practices Plus...

Meta Prompts, Grounding, ASCII, DSPy red teaming...



What did the WizardLM-2 see?



Beyond the wrappers, RAG and Prompt Engineering - Advanced AI Systems Engineering

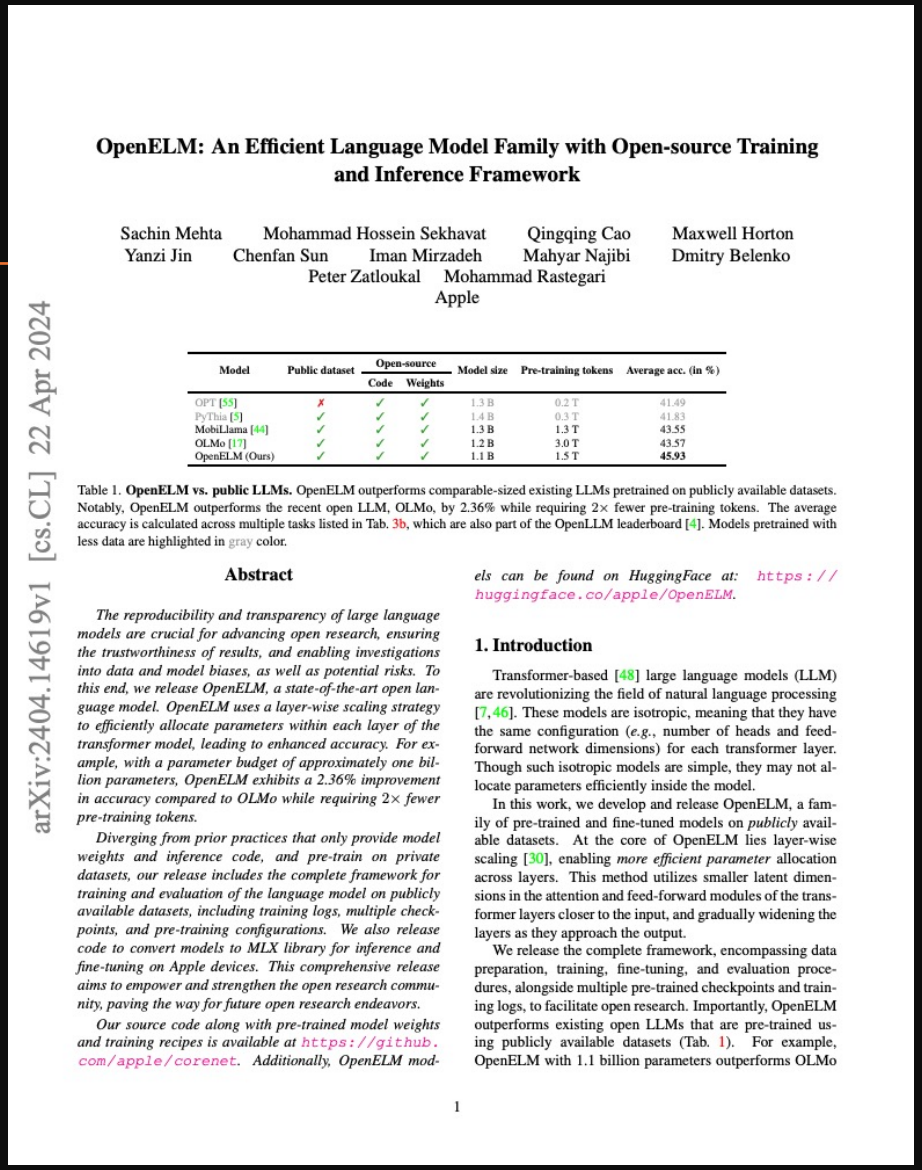
Lifecycle of an AI Model

- **Training:**
  Data preparation
  Efficient training techniques
  Evaluation

- **Fine-tuning:**
  RLHF, RLAIF

- **Inference:**
  Quantization
  Deployment

Training Your Own Model



arXiv:2404.14619v1 [cs.CL] 22 Apr 2024

OpenELM: An Efficient Language Model Family with Open-source Training and Inference Framework

Sachin Mehta Mohammad Hossein Sekhavat Qingqing Cao Maxwell Horton
Yanzi Jin Chenfan Sun Iman Mirzadeh Mahyar Najibi Dmitry Belenko
Peter Zatloukal Mohammad Rastegari
Apple

Model	Public dataset	Open-source		Model size	Pre-training tokens	Average acc. (in %)
		Code	Weights			
OPT [55]	✗	✓	✓	1.3 B	0.2 T	41.49
PyThia [5]	✓	✓	✓	1.4 B	0.3 T	41.83
MobiLlama [44]	✓	✓	✓	1.3 B	1.3 T	43.55
OLMo [17]	✓	✓	✓	1.2 B	3.0 T	43.57
OpenELM (Ours)	✓	✓	✓	1.1 B	1.5 T	**45.93**

Table 1. **OpenELM vs. public LLMs.** OpenELM outperforms comparable-sized existing LLMs pretrained on publicly available datasets. Notably, OpenELM outperforms the recent open LLM, OLMo, by 2.36% while requiring 2× fewer pre-training tokens. The average accuracy is calculated across multiple tasks listed in Tab. 3b, which are also part of the OpenLLM leaderboard [4]. Models pretrained with less data are highlighted in gray color.

Abstract

The reproducibility and transparency of large language models are crucial for advancing open research, ensuring the trustworthiness of results, and enabling investigations into data and model biases, as well as potential risks. To this end, we release OpenELM, a state-of-the-art open language model. OpenELM uses a layer-wise scaling strategy to efficiently allocate parameters within each layer of the transformer model, leading to enhanced accuracy. For example, with a parameter budget of approximately one billion parameters, OpenELM exhibits a 2.36% improvement in accuracy compared to OLMo while requiring 2× fewer pre-training tokens.

Diverging from prior practices that only provide model weights and inference code, and pre-train on private datasets, our release includes the complete framework for training and evaluation of the language model on publicly available datasets, including training logs, multiple checkpoints, and pre-training configurations. We also release code to convert models to MLX library for inference and fine-tuning on Apple devices. This comprehensive release aims to empower and strengthen the open research community, paving the way for future open research endeavors.

Our source code along with pre-trained model weights and training recipes is available at https://github.com/apple/corenet. *Additionally, OpenELM models can be found on HuggingFace at:* https://huggingface.co/apple/OpenELM.

1. Introduction

Transformer-based [48] large language models (LLM) are revolutionizing the field of natural language processing [7, 46]. These models are isotropic, meaning that they have the same configuration (*e.g.*, number of heads and feed-forward network dimensions) for each transformer layer. Though such isotropic models are simple, they may not allocate parameters efficiently inside the model.

In this work, we develop and release OpenELM, a family of pre-trained and fine-tuned models on *publicly* available datasets. At the core of OpenELM lies layer-wise scaling [30], enabling *more efficient parameter* allocation across layers. This method utilizes smaller latent dimensions in the attention and feed-forward modules of the transformer layers closer to the input, and gradually widening the layers as they approach the output.

We release the complete framework, encompassing data preparation, training, fine-tuning, and evaluation procedures, alongside multiple pre-trained checkpoints and training logs, to facilitate open research. Importantly, OpenELM outperforms existing open LLMs that are pre-trained using publicly available datasets (Tab. 1). For example, OpenELM with 1.1 billion parameters outperforms OLMo

1

https://arxiv.org/abs/2404.14619

Training Your Own Model

Non_Interactive – Software & ML

CONTACT NON_INT WHAT IS NON-INTERACTIVE?

The “it” in AI models is the dataset.

Posted on June 10, 2023 by jbetker

I’ve been at OpenAI for almost a year now. In that time, I’ve trained a **lot** of generative models. More than anyone really has any right to train. As I’ve spent these hours observing the effects of tweaking various model configurations and hyperparameters, one thing that has struck me is the similarities in between all the training runs.

It’s becoming awfully clear to me that these models are truly approximating their datasets to an incredible degree. What that means is not only that they learn what it means to be a dog or a cat, but the interstitial frequencies between distributions that don’t matter, like what photos humans are likely to take or words humans commonly write down.

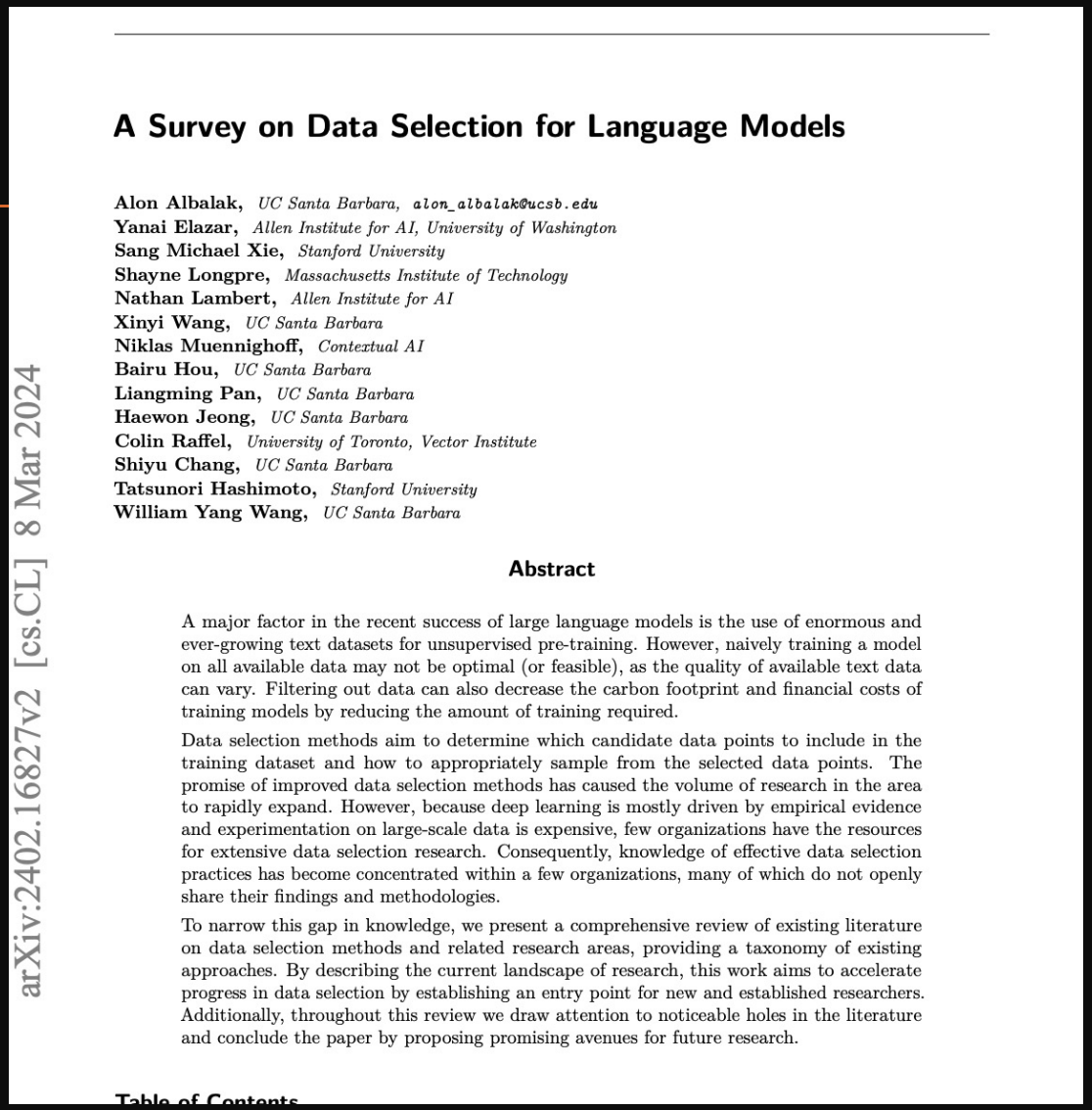
What this manifests as is – trained on the same dataset for long enough, pretty much every model with enough weights and training time converges to the same point. Sufficiently large diffusion conv-unets produce the same images as ViT generators. AR sampling produces the same images as diffusion.

This is a surprising observation! It implies that model behavior is not determined by architecture, hyperparameters, or optimizer choices. It’s determined by your dataset, nothing else. Everything else is a means to an end in efficiently delivery compute to approximating that dataset.

Then, when you refer to “Lambda”, “ChatGPT”, “Bard”, or “Claude” then, it’s not the model weights that you are referring to. It’s the dataset.

https://nonint.com/2023/06/10/the-it-in-ai-models-is-the-dataset/

Select Your Data



A Survey on Data Selection for Language Models

Alon Albalak, *UC Santa Barbara,* alon_albalak@ucsb.edu
Yanai Elazar, *Allen Institute for AI, University of Washington*
Sang Michael Xie, *Stanford University*
Shayne Longpre, *Massachusetts Institute of Technology*
Nathan Lambert, *Allen Institute for AI*
Xinyi Wang, *UC Santa Barbara*
Niklas Muennighoff, *Contextual AI*
Bairu Hou, *UC Santa Barbara*
Liangming Pan, *UC Santa Barbara*
Haewon Jeong, *UC Santa Barbara*
Colin Raffel, *University of Toronto, Vector Institute*
Shiyu Chang, *UC Santa Barbara*
Tatsunori Hashimoto, *Stanford University*
William Yang Wang, *UC Santa Barbara*

arXiv:2402.16827v2 [cs.CL] 8 Mar 2024

Abstract

A major factor in the recent success of large language models is the use of enormous and ever-growing text datasets for unsupervised pre-training. However, naively training a model on all available data may not be optimal (or feasible), as the quality of available text data can vary. Filtering out data can also decrease the carbon footprint and financial costs of training models by reducing the amount of training required.

Data selection methods aim to determine which candidate data points to include in the training dataset and how to appropriately sample from the selected data points. The promise of improved data selection methods has caused the volume of research in the area to rapidly expand. However, because deep learning is mostly driven by empirical evidence and experimentation on large-scale data is expensive, few organizations have the resources for extensive data selection research. Consequently, knowledge of effective data selection practices has become concentrated within a few organizations, many of which do not openly share their findings and methodologies.

To narrow this gap in knowledge, we present a comprehensive review of existing literature on data selection methods and related research areas, providing a taxonomy of existing approaches. By describing the current landscape of research, this work aims to accelerate progress in data selection by establishing an entry point for new and established researchers. Additionally, throughout this review we draw attention to noticeable holes in the literature and conclude the paper by proposing promising avenues for future research.

Table of Contents

https://arxiv.org/abs/2402.16827

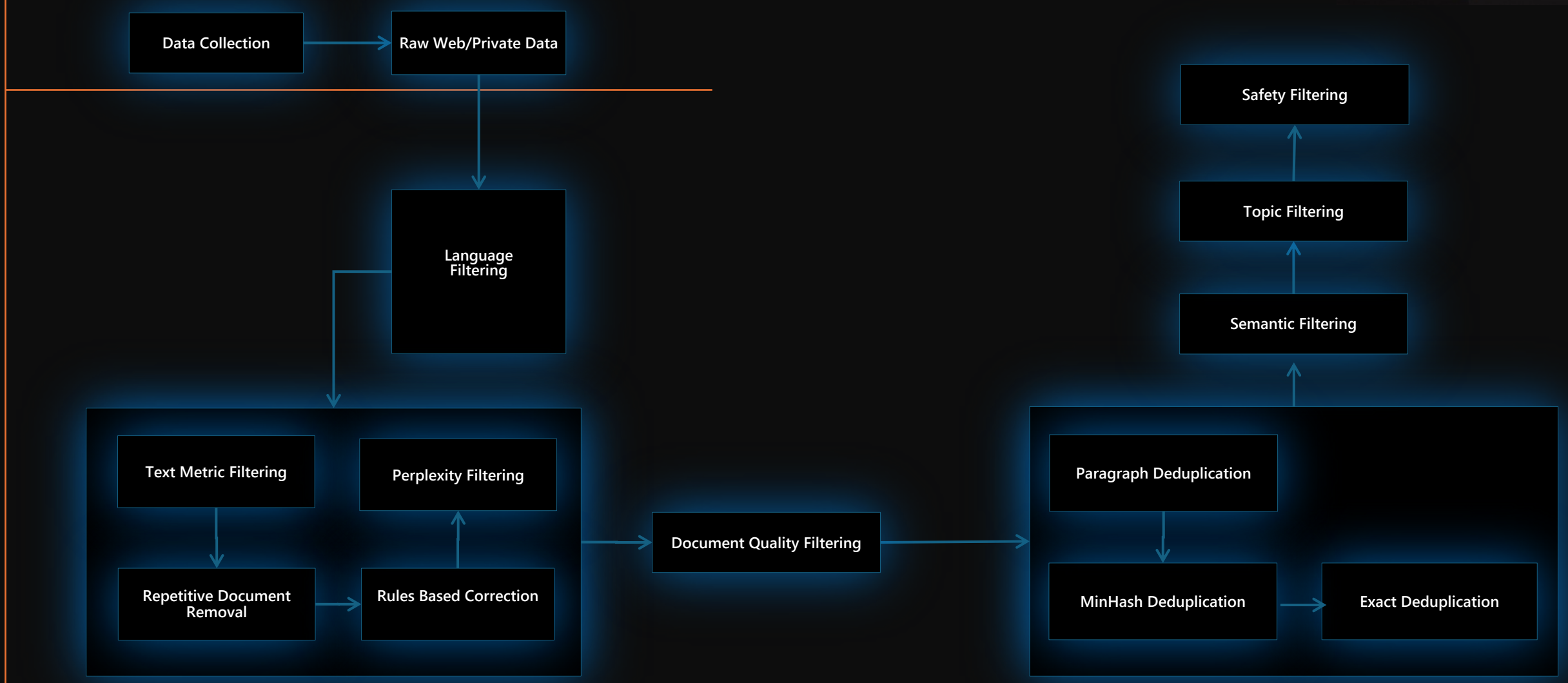
Data Preparation

- Model training requires multiple stages:
  - Pretraining
  - Instruction-tuning
  - Alignment
  - In-context learning
  - Task-specific fine-tuning
- Each training stage has different goals
- Data selection methods will use different mechanisms

Pretraining

- Goal: train a general-purpose model with a maximum coverage
- Requires: train on massive quantities of text, at least 1 Trillion tokens
- Diversity and coverage
  - Sourcing data from a wide array of domains, including less represented languages and dialects.
  - Ensuring the inclusion of various writing styles.
- Quality and robustness
  - Filtering out low-quality, toxic, or biased data to prevent model contamination.
  - Implementing rigorous testing phases to evaluate the model's performance across different contexts.
- Data quality evaluation: how to measure data quality at the billion tokens scale
  - Developing metrics to evaluate the relevance and representativeness of data.
  - Creating automated tools to efficiently identify and remove low-quality or duplicated content.

Data Preparation



Data Preparation

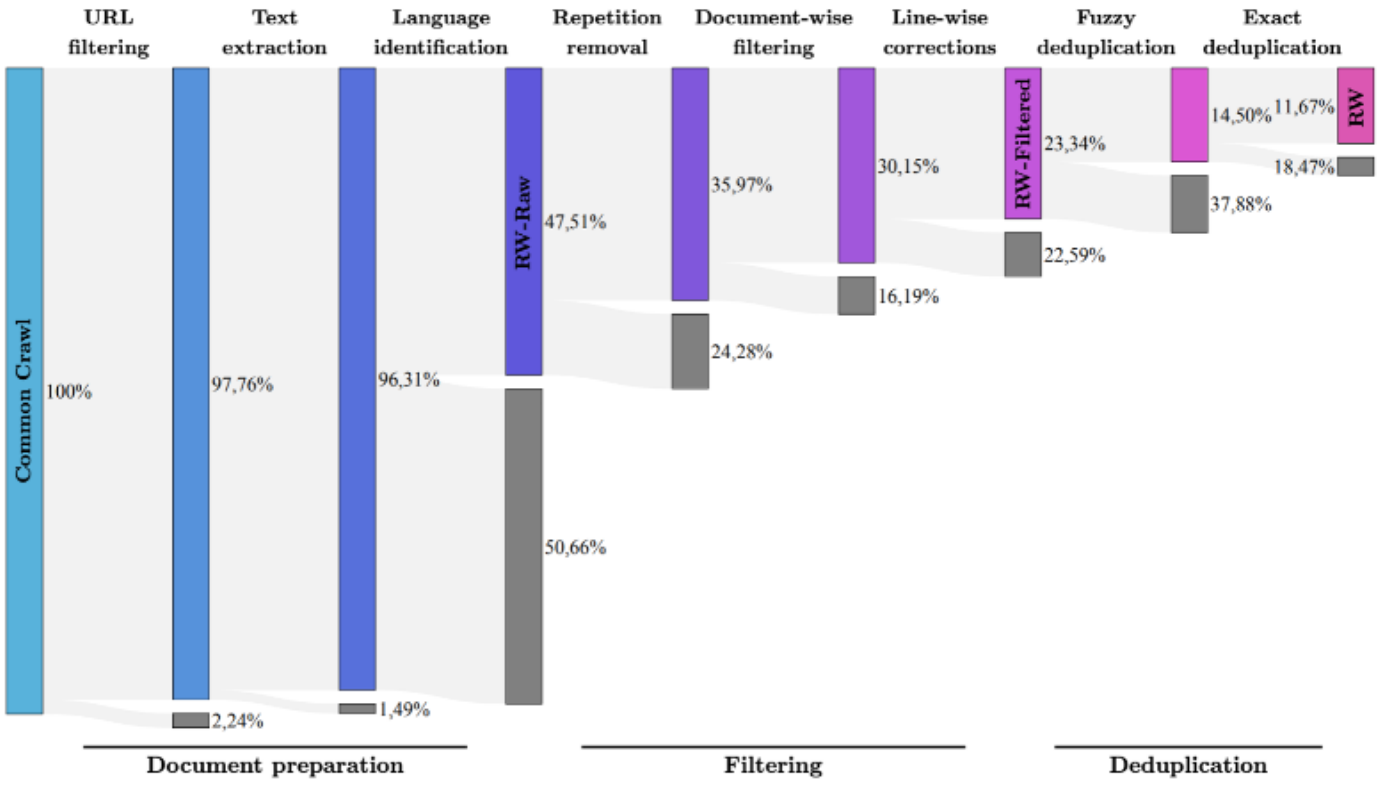


Figure 2. **Subsequent stages of Macrodata Refinement remove nearly 90% of the documents originally in CommonCrawl.** Notably filtering and deduplication each result in a halving of the data available: around 50% of documents are discarded for not being English 24% of remaining for being of insufficient quality, and 12% for being duplicates. We report removal rate (grey) with respect to each previous stage, and kept rate (shade) overall. Rates measured in % of documents in the document preparation phase, then in tokens.

Data Preparation

Figure 4: **An overview of the data filtering pipeline for pretraining**. Each filtering component is described in Section 3, and depicts common filters used for preprocessing text data. Note that different works employ different filters, at different stages, and do not necessarily adhere to the order conveyed here.

Document preparation | Filtering | Deduplication

Figure 2. **Subsequent stages of Macrodata Refinement remove nearly 90% of the documents originally in CommonCrawl.** Notably filtering and deduplication each result in a halving of the data available: around 50% of documents are discarded for not being English 24% of remaining for being of insufficient quality, and 12% for being duplicates. We report removal rate (grey) with respect to each previous stage, and kept rate (shade) overall. Rates measured in % of documents in the document preparation phase, then in tokens.

Data Sources

- Very Large that required significant preparation
  - Common Crawl
  - GitHub and Software Heritage
  - HuggingFace FineWeb
- Curated
  - Wikipedia
  - Public Domain Books
- Synthetic Data is the Future

Synthetic Data is the Future

- Simple Synthetic Dataset
  DSPy Synthesizer v2 (example https://github.com/stanfordnlp/dspy/tree/81c2f579d50057d51351c259796e07958efdd9d1/dspy/experimental/synthesizer)
- Complex Synthetic Dataset
  Distilabel (example https://github.com/argilla-io/distilabel-workbench/tree/main/projects/farming)
- Complex Synthetic Dataset
  Cosmopedia (example https://github.com/huggingface/cosmopedia)

15T Tokens Real Web Dataset

https://huggingface.co/datasets/HuggingFaceFW/fineweb

Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Posts | Docs | Solutions | Pricing

Datasets: HuggingFaceFW/fineweb | like 643

Tasks: Text Generation | Languages: English | Size Categories: n>1T | ArXiv: arxiv:2306.01116, arxiv:2109.07445 | Tags: Croissant | DOI: doi:10.57967/hf/2092 | License: odc-by

Dataset card | Viewer | Files and versions | Community 10

This dataset has 7 files that have been marked as unsafe.
View unsafe files

Dataset Preview | API | View in Dataset Viewer

Subset (96): default | Split (1): train

The full dataset viewer is not available (click to read why). Only showing a preview of the rows.

text string	id string	dump string	url string
How AP reported in all formats from tornado-stricken regionsMarch 8, 2012 When the first…	<urn:uuid:d66bc6fe-8477-4adf-b430-…	CC-MAIN-…	http://%20jwashington@ap.org/Content/Press-Release/2012/How-AP-reported-in-all-formats-from-…
Did you know you have two little yellow, nine-volt-battery-sized adrenal glands in your body, just…	<urn:uuid:803e14c3-dc2e-43d6-b75d-…	CC-MAIN-…	http://1000awesomethings.com/2012/09/24/934-adrenaline/
Car Wash For Clara! Now is your chance to help! 2 year old Clara Woodward has Cancer! Clara can't sa…	<urn:uuid:ac1bbfff-9519-4967-9c64-…	CC-MAIN-…	http://1027kord.com/car-wash-for-clara/
Listeners Get Sky-high View of Missoula From Hot Air Balloons On Friday, June 1, during the…	<urn:uuid:c1445c58-b111-4c4e-badd-…	CC-MAIN-…	http://1075zoofm.com/listeners-get-sky-high-view-of-missoula-from-hot-air-balloons/
Log In Please enter your ECode to log in. Forgotten your eCode? If you created your login but do not…	<urn:uuid:e5829f7d-b944-4468-9573-…	CC-MAIN-…	http://1105govinfoevents.com/enterprisearchitectureevent/public/MyBriefcasef671.html?…
spotlight provides a convenient rechargeable LED light for work play and everyday life. choose from…	<urn:uuid:6bfca20f-ea67-41ba-b995-…	CC-MAIN-…	http://12vspotlight.com/
K-State put themselves in sole position of first	<urn:uuid:dc9d9fd8-	CC-	

FineWeb

"15 trillion tokens of the finest data the web has to offer"

Downloads last month 65

Use in Datasets library | Edit dataset card

Homepage and Repository: huggingface.co

Point of Contact: please create a discussion on the Community tab

License: Open Data Commons Attribution License (ODC-By) v1.0

Size of downloaded dataset files: 45 TB

Size of the auto-converted Parquet files: 45 TB

Number of rows: 22,335,106,879

Models trained or fine-tuned on HuggingFaceFW/…

- Dijitaal/DijiHax.Spooky.Pi — Updated 2 days ago • 3
- Damo2910/NTANCVA — Updated about 4 hours ago
- Villain7777/Nude — Text Classification • Updated about 5 hours ago

15 T

Tokens- Clean and Deduplicated

45 TB

Dataset Size

ODC

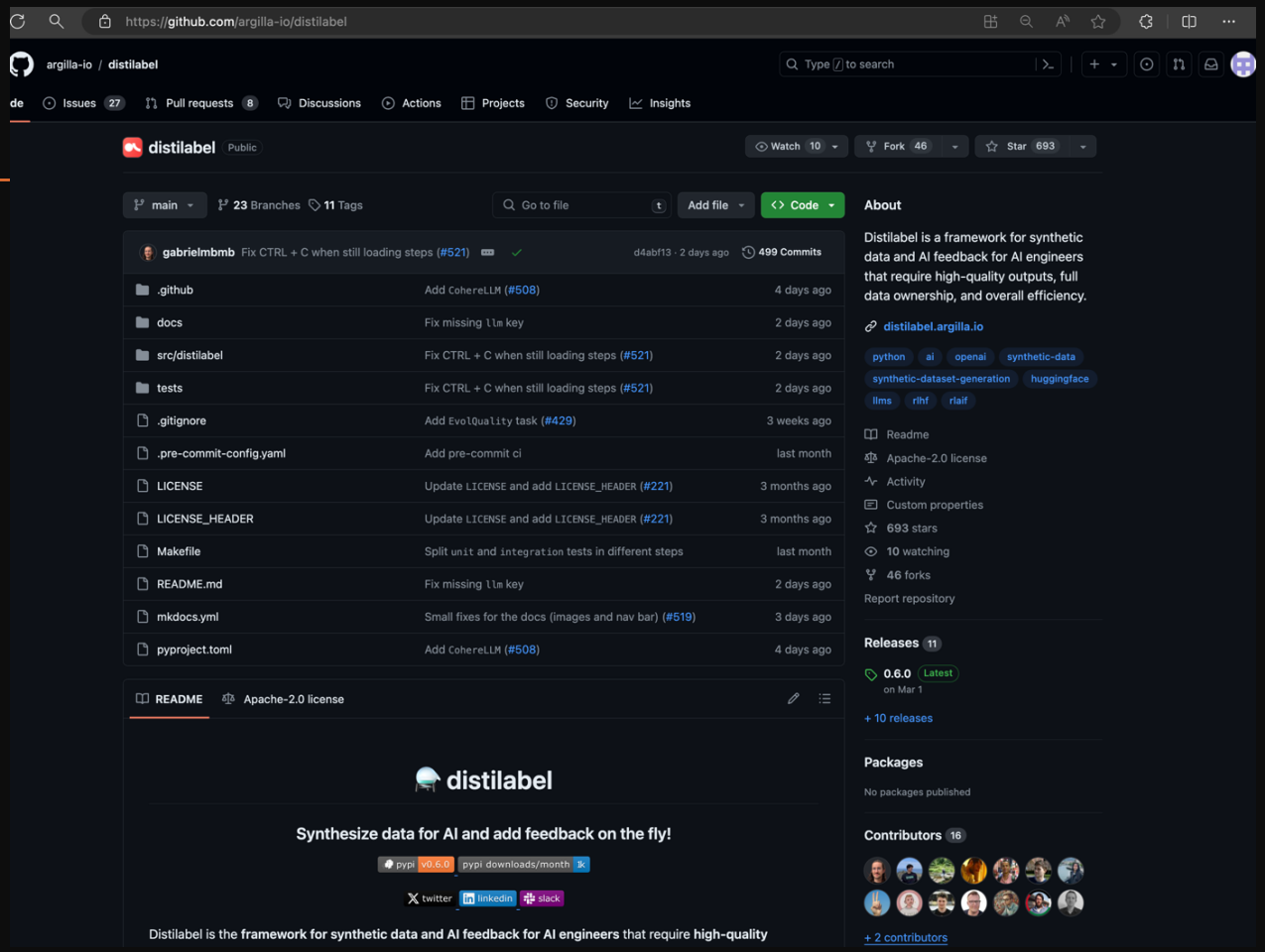
Open Data Commons License

Created In

2024

https://huggingface.co/datasets/HuggingFaceFW/fineweb

Generate Synthetic Datasets Locally



distilabel

Started In
2022

16
Contributors

Distilabel is a framework for synthetic data and AI feedback for AI engineers that require high-quality outputs, full data ownership, and overall efficiency.

693
stars on Github

8
Active PRs

Create a synthetic dataset seed locally on your own AI platform for aligning models to a specific domain example

Synthetic Dataset Example: Cosmopedia

https://huggingface.co/datasets/HuggingFaceTB/cosmopedia

Hugging Face | Search models, datasets, users... | Models | Datasets | Spaces | Posts | Docs | Solutions | Pricing

Datasets: HuggingFaceTB/cosmopedia | like 451

Languages: English | ArXiv: arxiv:2309.05463, arxiv:2306.11644 | Tags: Synthetic, Croissant | License: apache-2.0

Dataset card | Viewer | Files and versions | Community 15

Dataset Viewer | Auto-converted to Parquet | API | View in Dataset Viewer

Subset (8): auto_math_text · 1.95M rows

Split (1): train · 1.95M rows

Search this dataset

prompt (string · lengths, 738 – 1.71k)	text_token_length (int64, 150 – 3.09k)	text (string · lengths, 312 – 30.6k)	seed_data (string · class, 1 value)
Here's an extract from a webpage: "# Discount Rate Calculator Created by Tibor Pál, PhD…	614	Hello there! Today, we are going to talk about something called the "discount rate." Now, I know…	auto_math_tex
Write an educational piece suited for college students related to the following text…	2,502	Logical implication is a fundamental concept in logic and mathematics, which represents a specifi…	auto_math_tex
Here's an extract from a webpage: "# 1 Operations with Matrice 2 Properties of Matri…	330	Hello there! Today we're going to learn about matrices and how to do operations with them. You…	auto_math_tex
Write an educational piece suited for college students related to the following text…	608	To find the second derivative of y with respect to x, denoted as $(d^2y)/(dx^2)$, for the equation $3x^2$ +…	auto_math_tex
Write an educational piece suited for college students related to the following text…	663	The Nilakantha Series is a historically significant infinite series used to approximate…	auto_math_tex
Here's an extract from a webpage: "Getting the final velocity by intuition I'm wondering how…	480	Imagine you are on a playground slide, sliding down from the top. At the very beginning, you are…	auto_math_tex

‹ Previous 1 2 3 ... 19,499 Next ›

Cosmopedia v0.1

Downloads last month 7,222

Use in Datasets library | Edit dataset card

Size of downloaded dataset files: 92.2 GB

Size of the auto-converted Parquet files: 92.2 GB

Number of rows: 31,064,744

Models trained or fine-tuned on HuggingFaceTB/...

- HuggingFaceTB/cosmo-1b — Text Generation • Updated Feb 23 • 7.34k • 110
- TencentARC/Mistral_Pro_8B_v0.1 — Text Generation • Updated Feb 27 • 5.41k • 63
- MaziyarPanahi/Mistral-7B-Instruct-Kha… — Text Generation • Updated Feb 28 • 2.12k • 4
- sayhan/Mistral_Pro_8B_v0.1-GGUF — Text Generation • Updated Feb 24 • 458

30M

Synthetic Samples

8

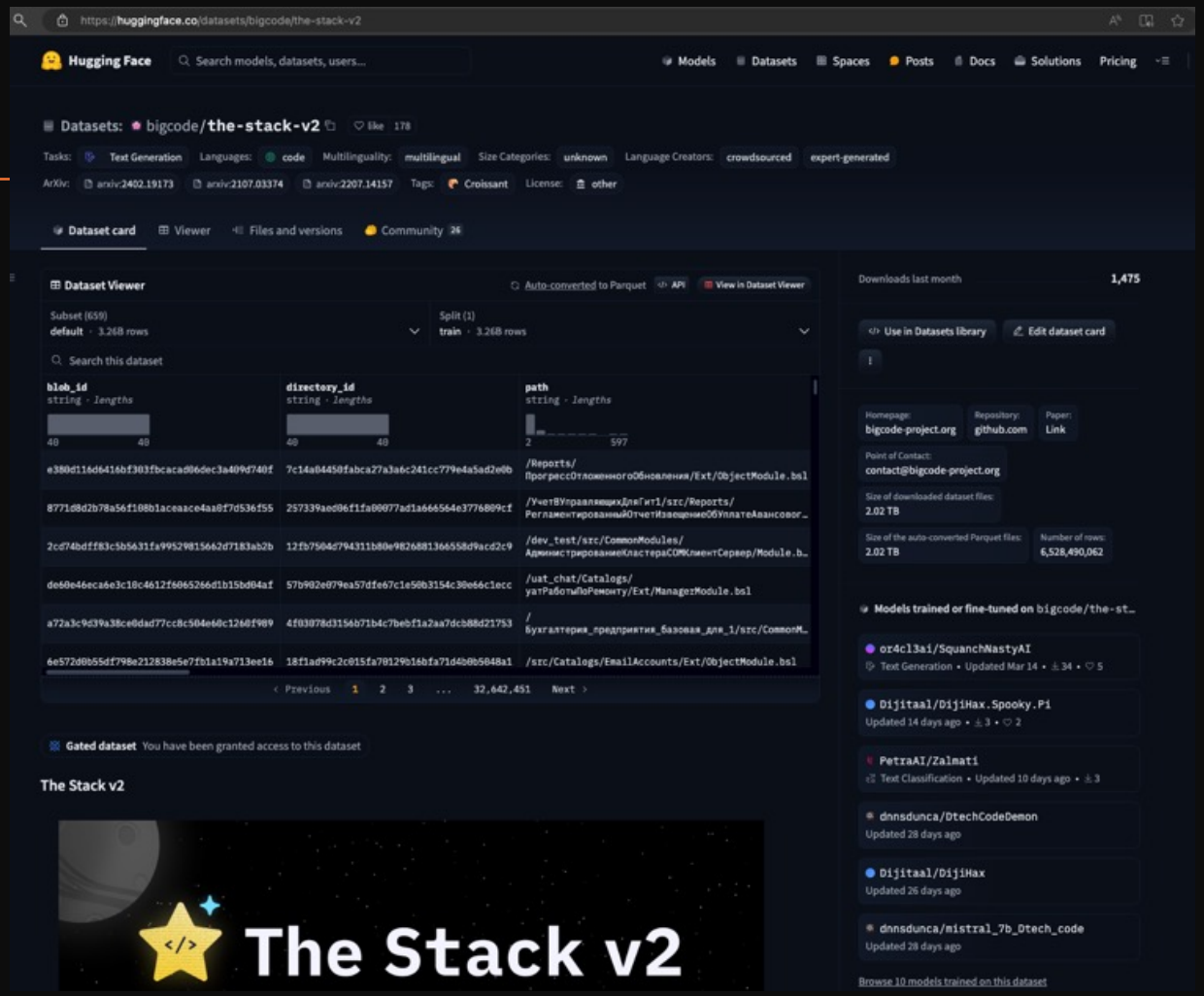
Domain Splits

Generated In

2024

https://huggingface.co/datasets/HuggingFaceTB/cosmopedia

Coding Dataset Example: The Stack v2



Created In
2024

67.5 TB
Full Dataset

32.1 TB
Deduplicated Dataset

658
Programming Languages

https://huggingface.co/datasets/bigcode/the-stack-v2

Data Filtering

- Quality Filtering Heuristics
  - Controlled
  - Robust
  - Clear Priors
- Quality Filtering by AI
  - Classifier-based filtering: fastText classification with an n-gram size of 2
  - Perplexity-based filtering: 5-gram Kneser-Ney model on Wikipedia
  - Threshold-based filtering: quality to content filters
- Selective Language Modeling SLM
  - Train a reference model on a high-quality corpus
  - Use it to reference each token in a corpus using its loss
  - Use only tokens with a high excess loss between reference and the training model

Data Deduplication

- Fuzzy
  - BLOOM Filters for hashing and fixed-size vector
  - MinHash for hashing and sorting
- Exact
  - Exact substrings with a suffix array
  - Sentence deduplication
- Over-deduplication may keep only the bad data

Prepare the Data for Pre-Training

- ❑ Shuffle
- ❑ Tokenizers
- ❑ Tokenization Scaling

Data Quality Evaluation

- ❑ Start With a Small 1-2B Model
- ❑ Manual Data Inspection
- ❑ Clustering

Model Training

- Size and Efficiency
  - Parallelism
  - Asynchronous
  - Kernel Merging
  - Attention
- Training Recipe Stability
- Capacity Scale
  - Mixture of Experts
  - Mixture of Depths
  - Creating Hybrids Transformer/RNN, Transformer/SSM

4-D Parallelism

- Data
  - Compute efficiency for gradient all-reduce, training efficiency of batch-size
- Tensor
  - Rewrite model code
  - Reduce sync points with combined column/row slicing
- Pipeline
  - Group sub-parts of the networks
  - Optimize GPUs utilization
- Sequence

Breadth-First Pipeline Parallelism https://arxiv.org/abs/2211.05953
Reducing Activation Recomputation in Large Transformer Models https://arxiv.org/abs/2205.05198
Sequence Parallelism: Long Sequence Training from System Perspective https://arxiv.org/abs/2105.13120
FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning https://arxiv.org/abs/2307.08691

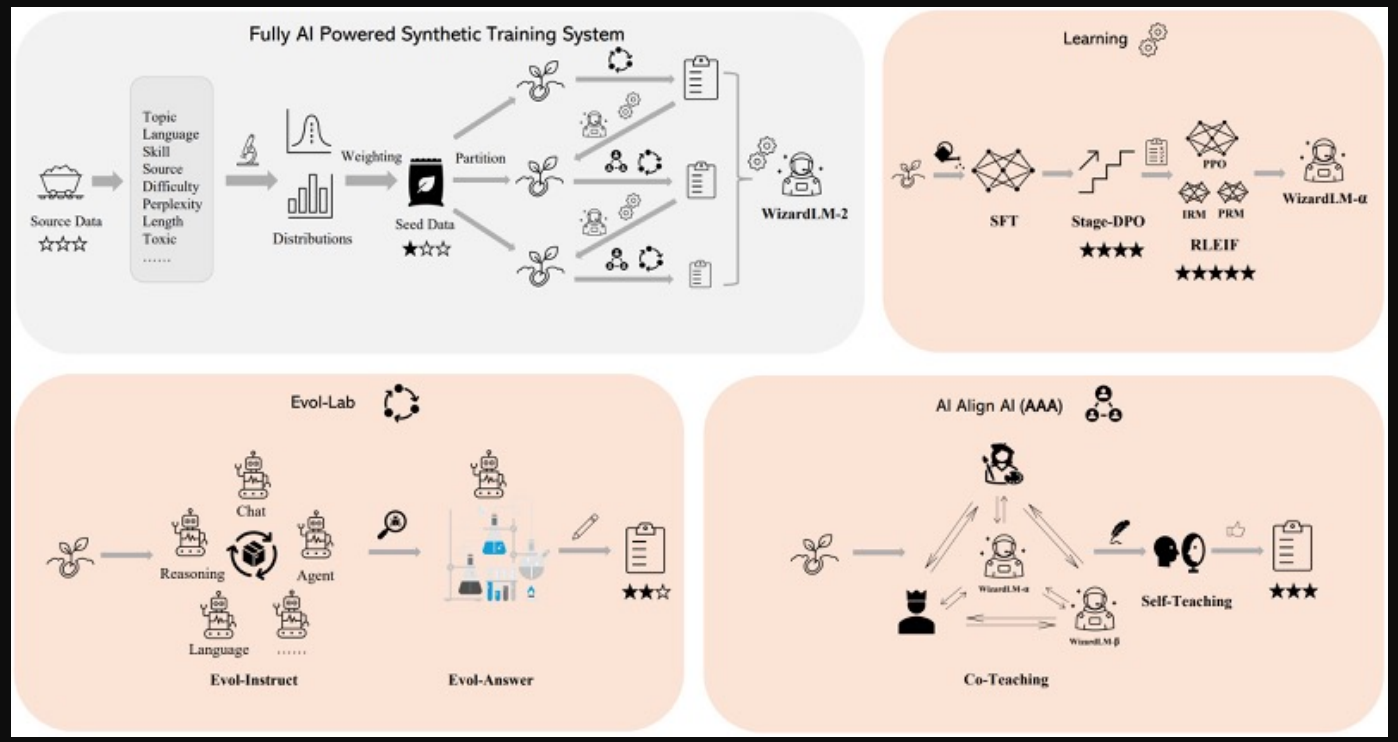
Training Recipes

- ❑ Initialization
- ❑ Stabilization
- ❑ Learning Rate
- ❑ Scaling hyper-parameters results

MiniCPM V2.0 https://huggingface.co/openbmb/MiniCPM-V-2
Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Transfer https://arxiv.org/abs/2203.03466

Synthetic AI Recipe as an Emerging Trend
The Mystic WizardLM-2

- ❑ Data Pre-Processing
  - Weighted Sampling
  - Progressive Learning
- ❑ Evol-Instruct
- ❑ AI Aligns AI (AAA)
  - Co-Teaching
  - Self-Teaching
- ❑ Supervised Learning
  - Staged-DPO
  - RLEIF with IRM and PRM



Research Paper expected in Q2CY24

Alignment

- ❑ Reinforced Learning by Human Feedback

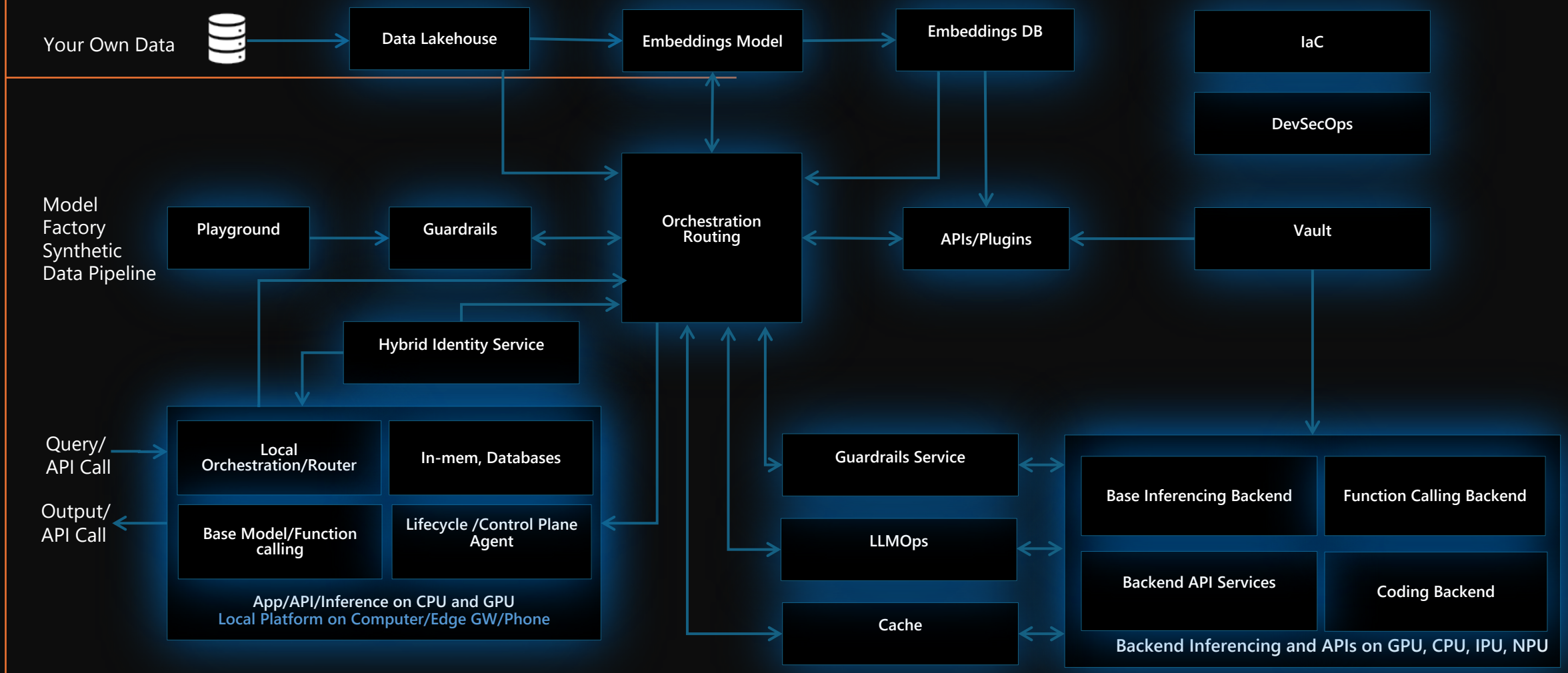
  Direct Preference Optimization

  Odds Ratio Preference Optimization (Loss function of Alignment and SFT)

- ❑ Reinforced Learning by AI Feedback
- ❑ Reinforced Learning by Evol-Instruct Feedback

Direct Preference Optimization: Your Language Model is Secretly a Reward Model https://arxiv.org/abs/2305.18290
ORPO: Monolithic Preference Optimization without Reference Model https://arxiv.org/abs/2403.07691
RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback https://arxiv.org/abs/2309.00267

Build System of Systems



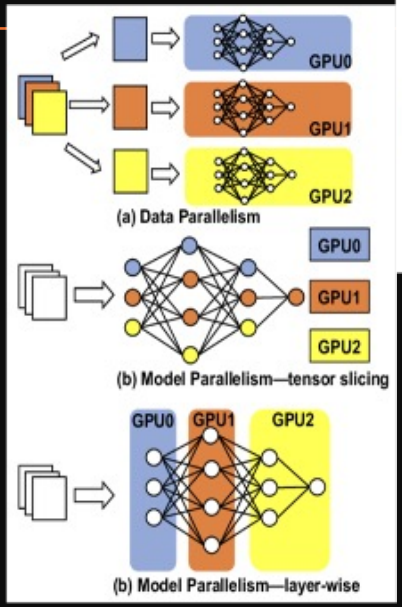
Practical Use Cases

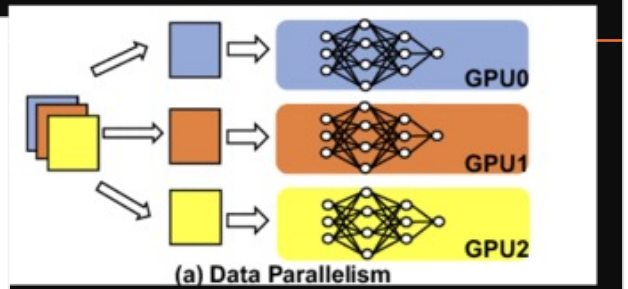
Practical Use Cases

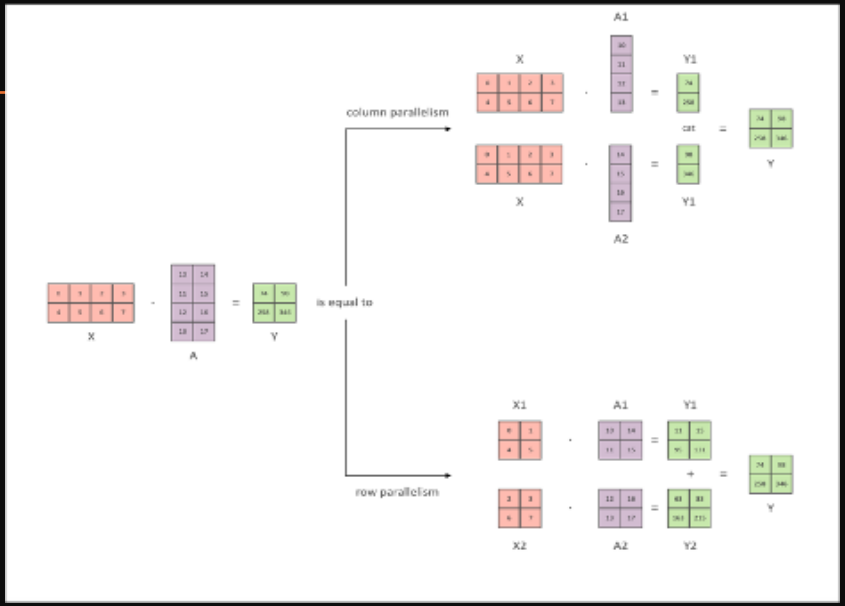
1. Content Creation
2. Automation of Routine Tasks
3. Human-Computer Interface Personalization
4. Assisted Software Development
5. Design and Prototyping
6. Synthetic Data Generation

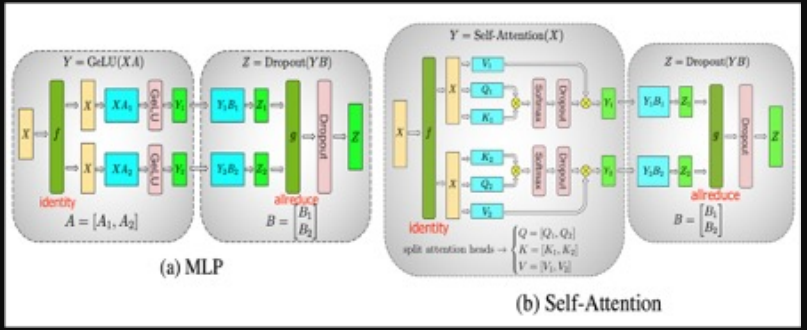
Thank You!
We Will Meet Again!

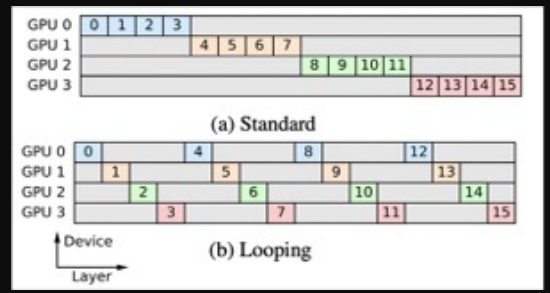
Backup slides

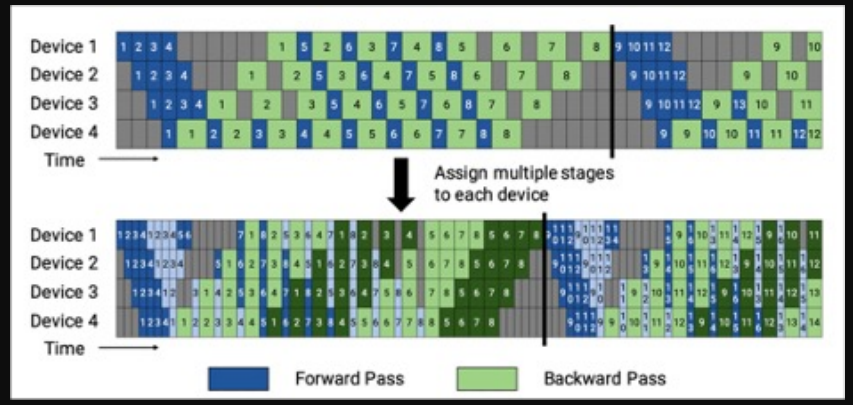

GPU0
GPU1
GPU2
(a) Data Parallelism
GPU0
GPU1
GPU2
(b) Model Parallelism—tensor slicing
GPU0
GPU1
GPU2
(b) Model Parallelism—layer-wise

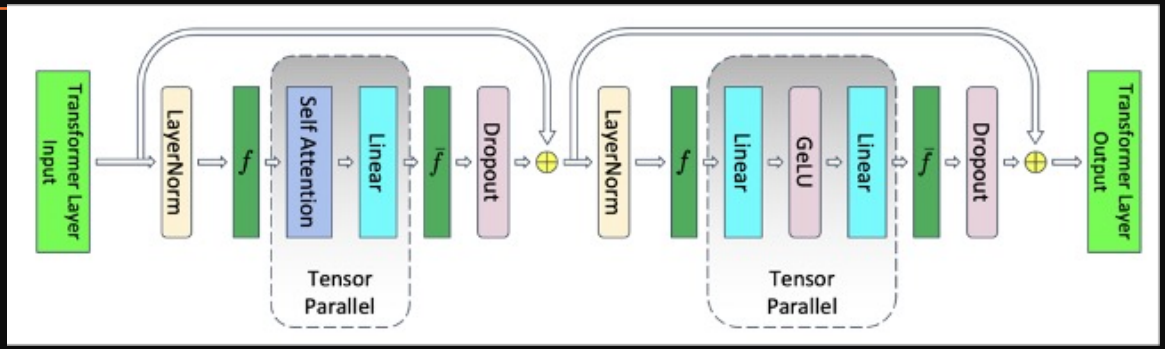

GPU0
GPU1
GPU2
(a) Data Parallelism

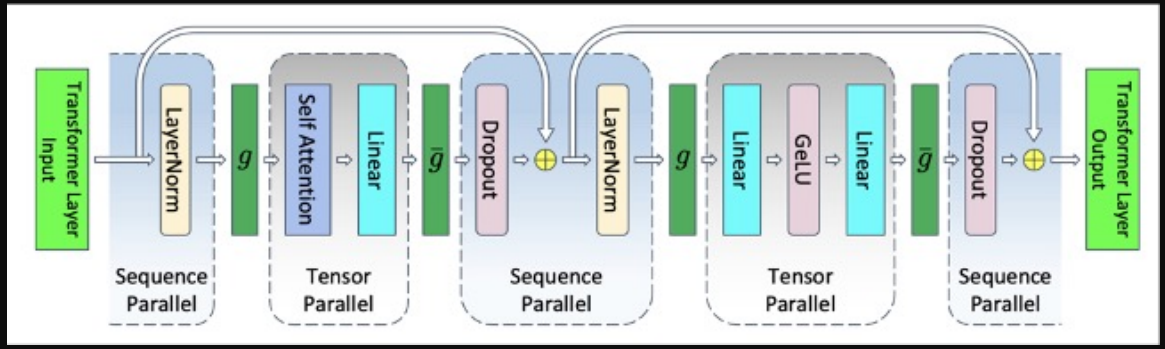

column parallelism
is equal to
row parallelism
X
A
Y
A1
A2
Y1
Y2
X1
X2
cat


Y = GeLU(XA)
Z = Dropout(YB)
identity
allreduce
(a) MLP
Y = Self-Attention(X)
Z = Dropout(YB)
identity
allreduce
split attention heads
(b) Self-Attention


GPU 0
GPU 1
GPU 2
GPU 3
(a) Standard
Device
Layer
(b) Looping


Device 1
Device 2
Device 3
Device 4
Time
Assign multiple stages
to each device
Forward Pass
Backward Pass


Transformer Layer Input
LayerNorm
f
Self Attention
Linear
f̄
Dropout
Tensor Parallel
LayerNorm
f
Linear
GeLU
Linear
f̄
Dropout
Tensor Parallel
Transformer Layer Output


Transformer Layer Input
LayerNorm
Sequence Parallel
g
Self Attention
Linear
Tensor Parallel
ḡ
Dropout
Sequence Parallel
LayerNorm
g
Linear
GeLU
Linear
Tensor Parallel
ḡ
Dropout
Sequence Parallel
Transformer Layer Output

(d) Looped pipeline, breadth-first schedule (PP_{BF}): small bubble, best overlap

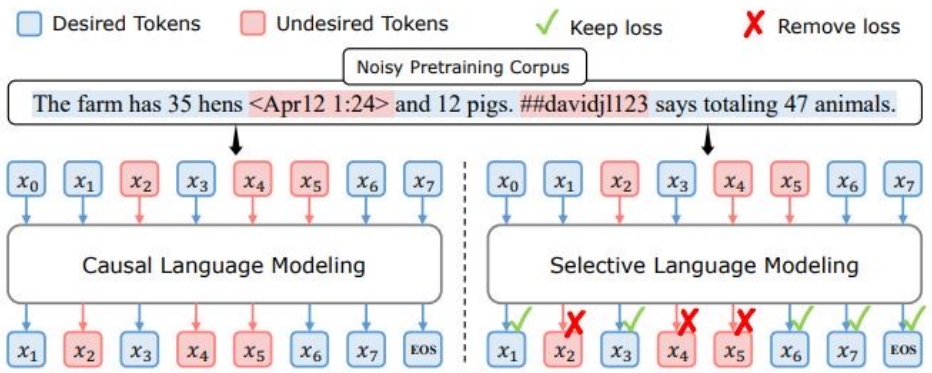


Figure 2: **Upper:** Even an extensively filtered pretraining corpus contains token-level noise. **Left:** Previous Causal Language Modeling (CLM) trains on all tokens. **Right:** Our proposed Selective Language Modeling (SLM) selectively applies loss on those useful and clean tokens.