The background image shows the Golden Gate Bridge in San Francisco, California, during a sunset or sunrise. The bridge's towers are silhouetted against a sky filled with warm orange and pink hues. In the foreground, there is a lush field of lavender plants, their purple flowers swaying slightly. The water of the bay is visible in the middle ground, with distant hills or mountains on the horizon.

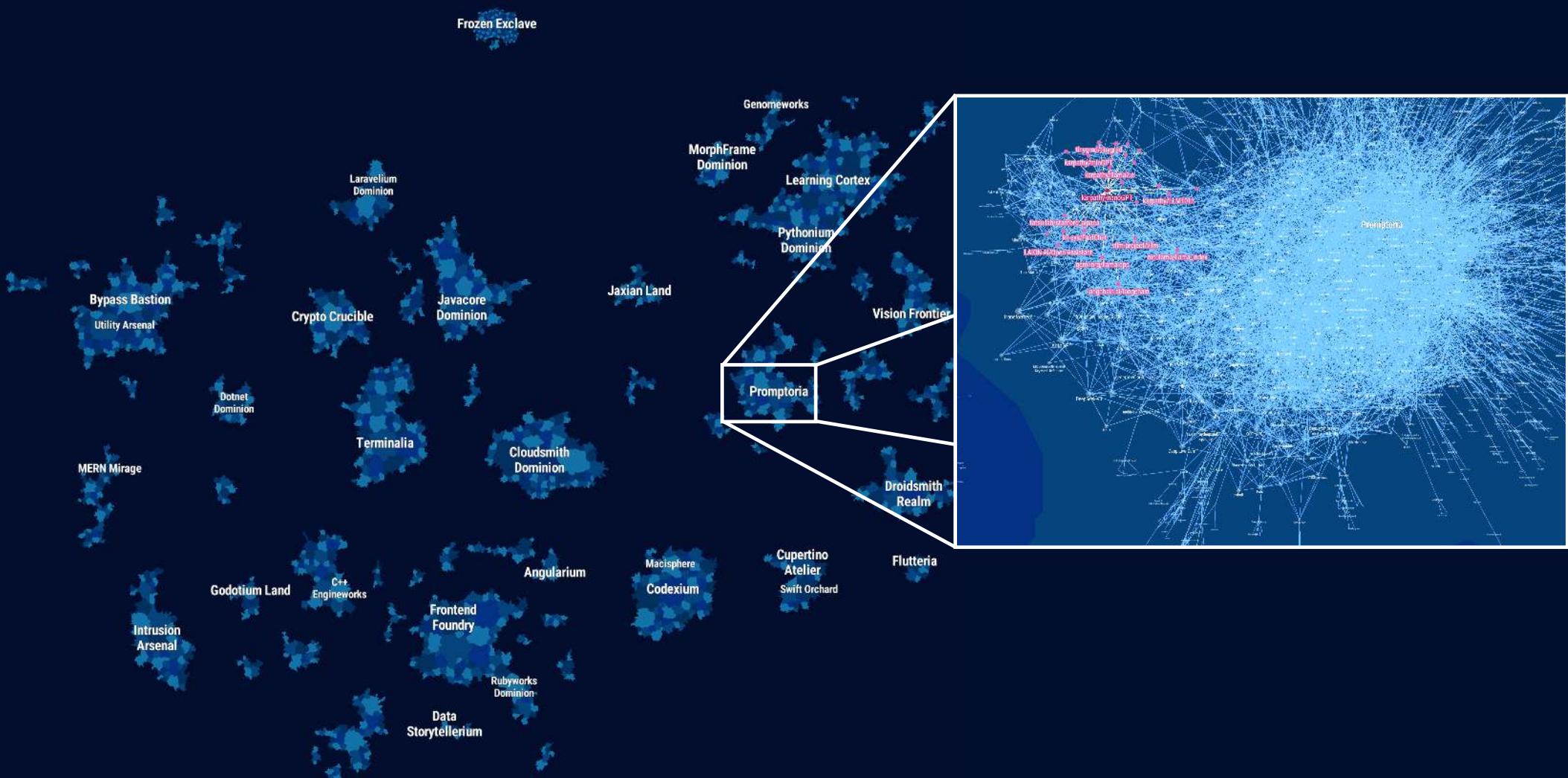
Software in the era of AI

Andrej Karpathy
YC AI Startup school
June 16

Software is changing.

(again)

"Map of GitHub"



Software 2.0



Andrej Karpathy

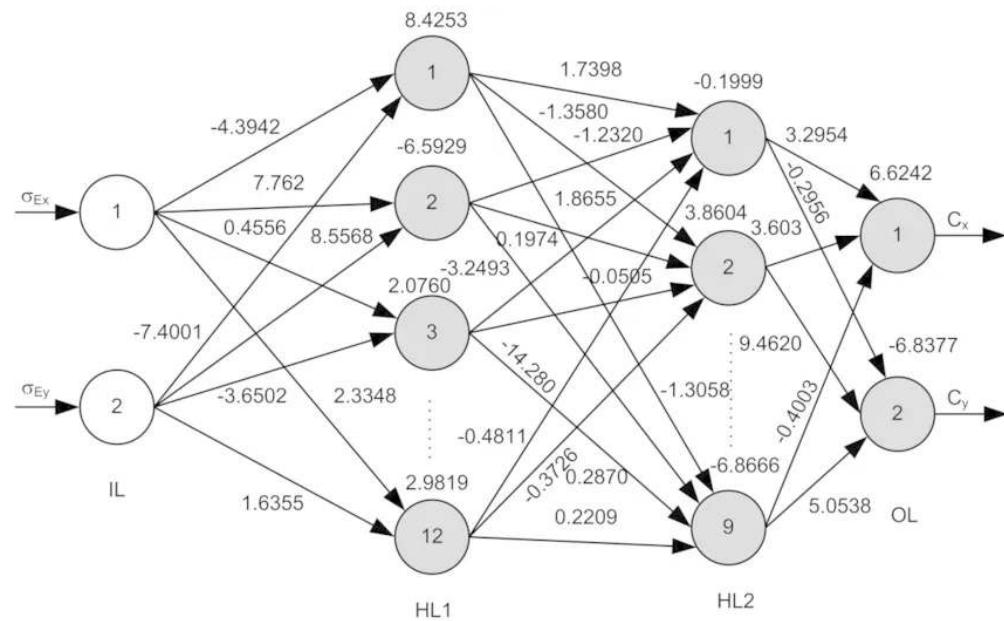
Follow

9 min read · Nov 11, 2017

Software 1.0 = code



Software 2.0 = weights



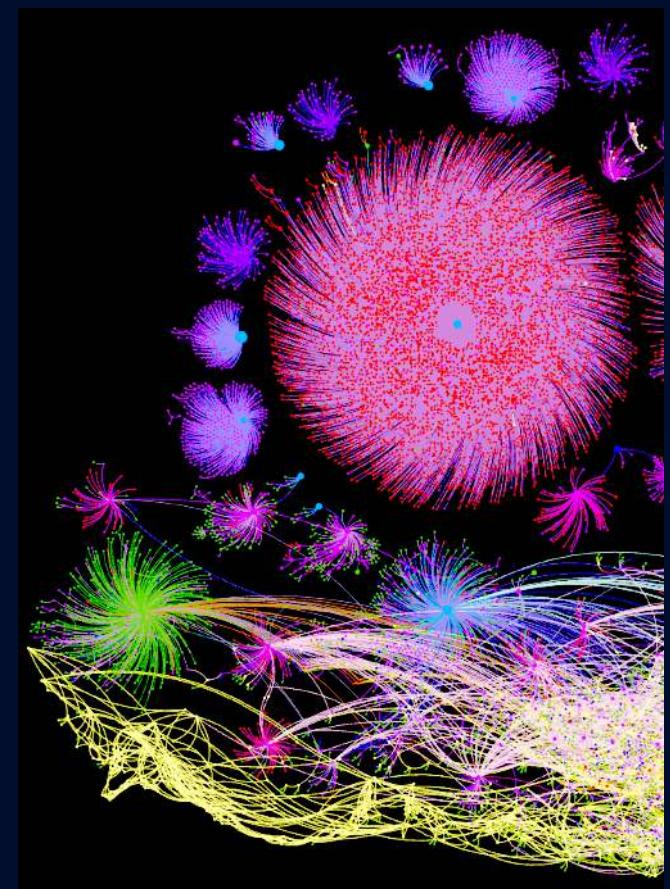
"Map of GitHub" (Software 1.0)

computer code



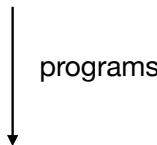
HuggingFace Model Atlas (Software 2.0)

neural network weights



Software 1.0

computer code



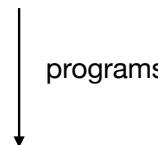
computer



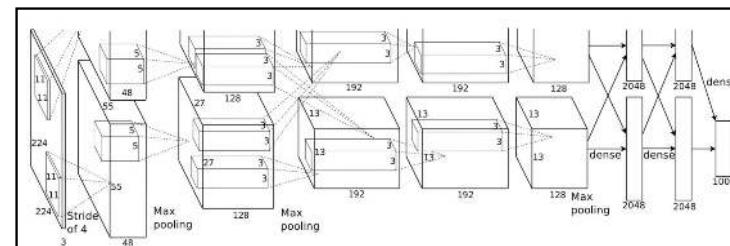
became programmable in ~1940s

Software 2.0

weights



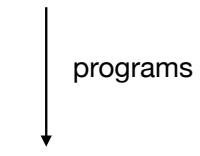
neural net



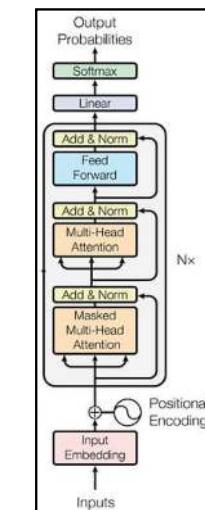
fixed function neural net
e.g. AlexNet: for image recognition (~2012)

Software 3.0

prompts



LLM



~2019

LLM = programmable neural net!

Example: Sentiment Classification

Software 1.0

```
python
def simple_sentiment(review: str) -> str:
    """Return 'positive' or 'negative' based on a tiny keyword lexicon."""
    positive = {
        "good", "great", "excellent", "amazing", "wonderful", "fantastic",
        "awesome", "loved", "love", "like", "enjoyed", "superb", "delightful"
    }
    negative = {
        "bad", "terrible", "awful", "poor", "boring", "hate", "hated",
        "dislike", "worst", "dull", "disappointing", "mediocre"
    }

    score = 0
    for word in review.lower().split():
        w = word.strip(",.!?:")      # crude token clean-up
        if w in positive:
            score += 1
        elif w in negative:
            score -= 1

    return "positive" if score >= 0 else "negative"
```

Software 2.0

10,000 positive examples
10,000 negative examples
encoding (e.g. bag of words)

train binary classifier

parameters

Software 3.0

You are a sentiment classifier. For every review that appears between the tags

<REVIEW> ... </REVIEW>, respond with **exactly one word**, either POSITIVE or NEGATIVE (all-caps, no punctuation, no extra text).

Example 1

<REVIEW>I absolutely loved this film—the characters were engaging and the ending was perfect.</REVIEW>

POSITIVE

Example 2

<REVIEW>The plot was incoherent and the acting felt forced; I regret watching it.</REVIEW>

NEGATIVE

Example 3

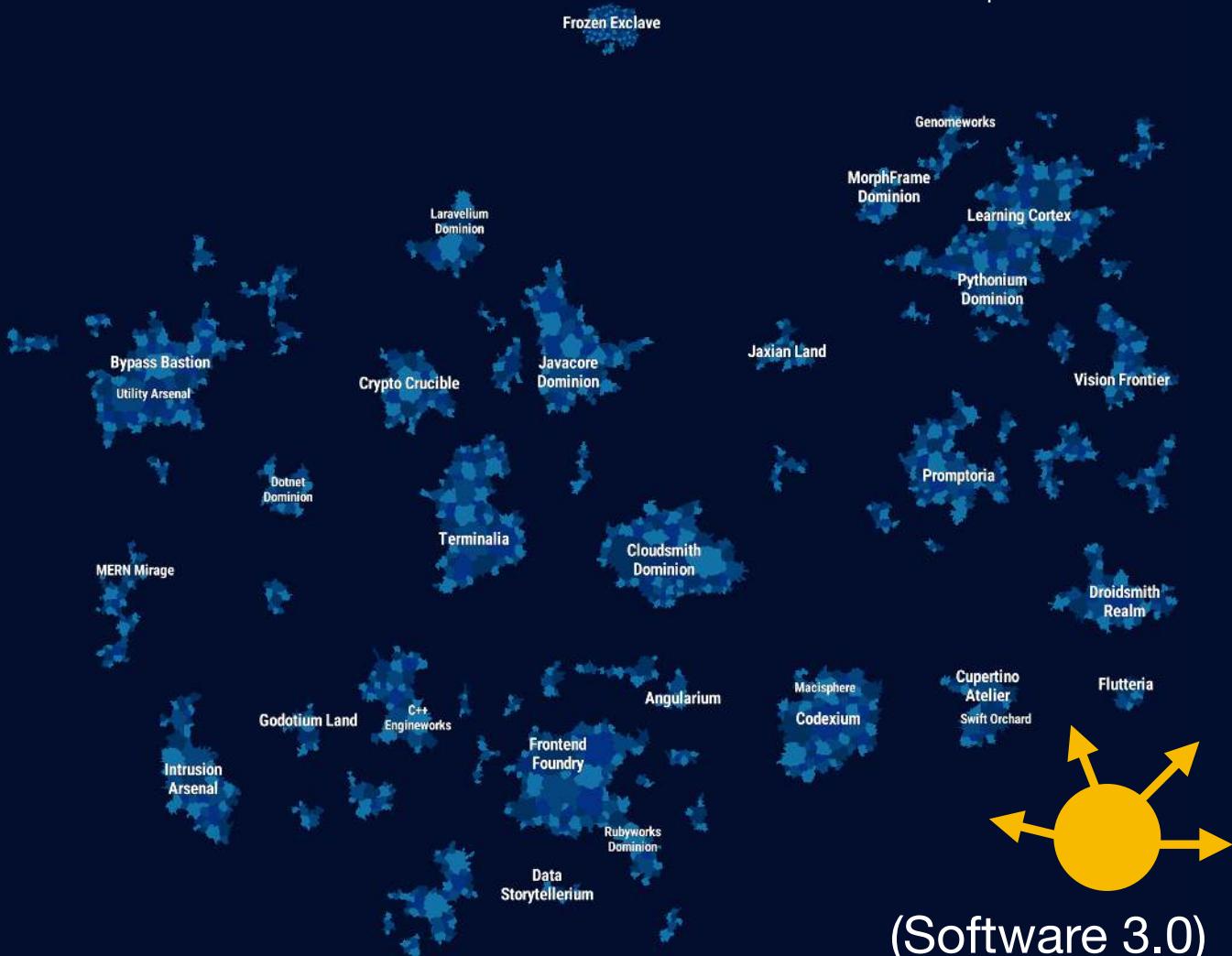
<REVIEW>An energetic soundtrack and solid visuals almost save it, but the story drags and the jokes fall flat.</REVIEW>

NEGATIVE

Now classify the next review.

"Map of GitHub" (Software 1.0)

computer code

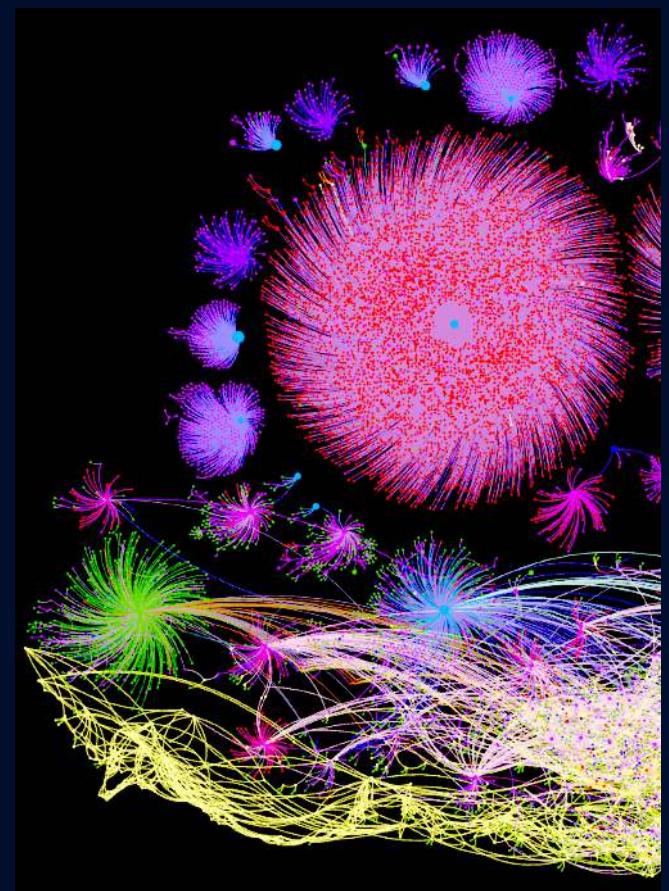


(Software 3.0)

LLM prompts, in English

HuggingFace Model Atlas (Software 2.0)

neural network weights



🔔 Pinned



Andrej Karpathy ✅ @karpathy · Jan 24, 2023

The hottest new programming language is English

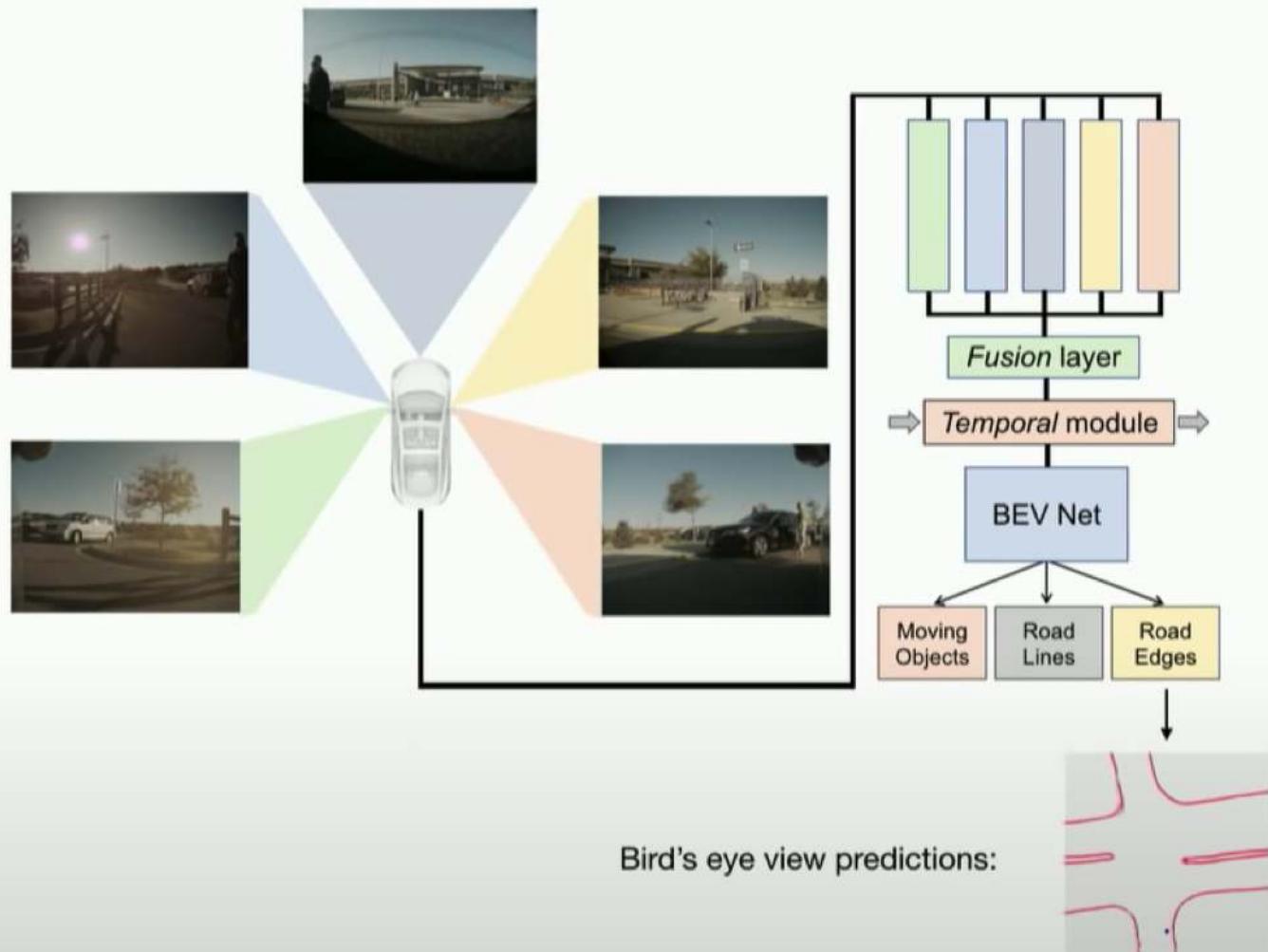
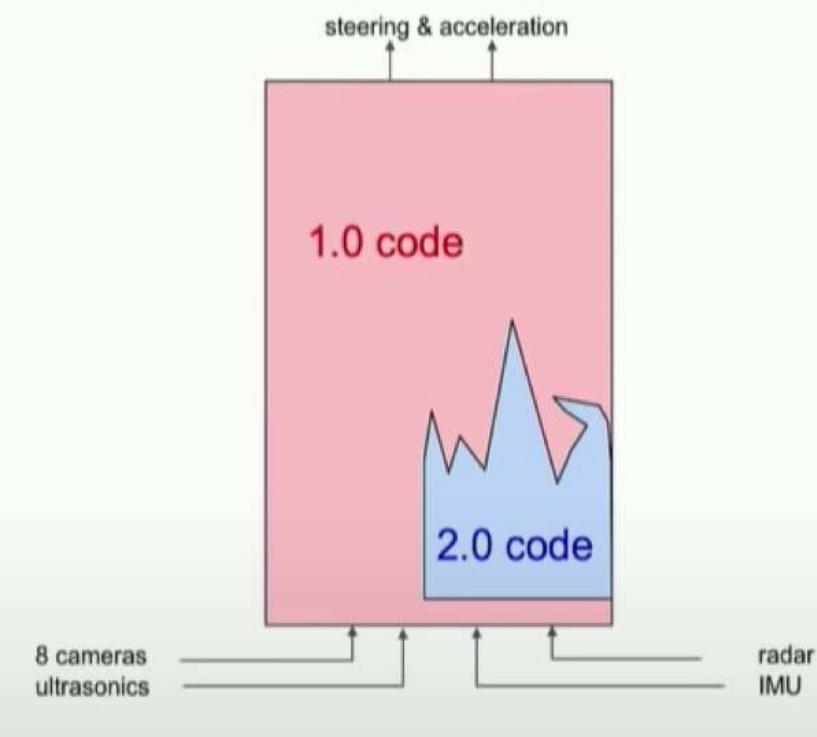
1.1K

7K

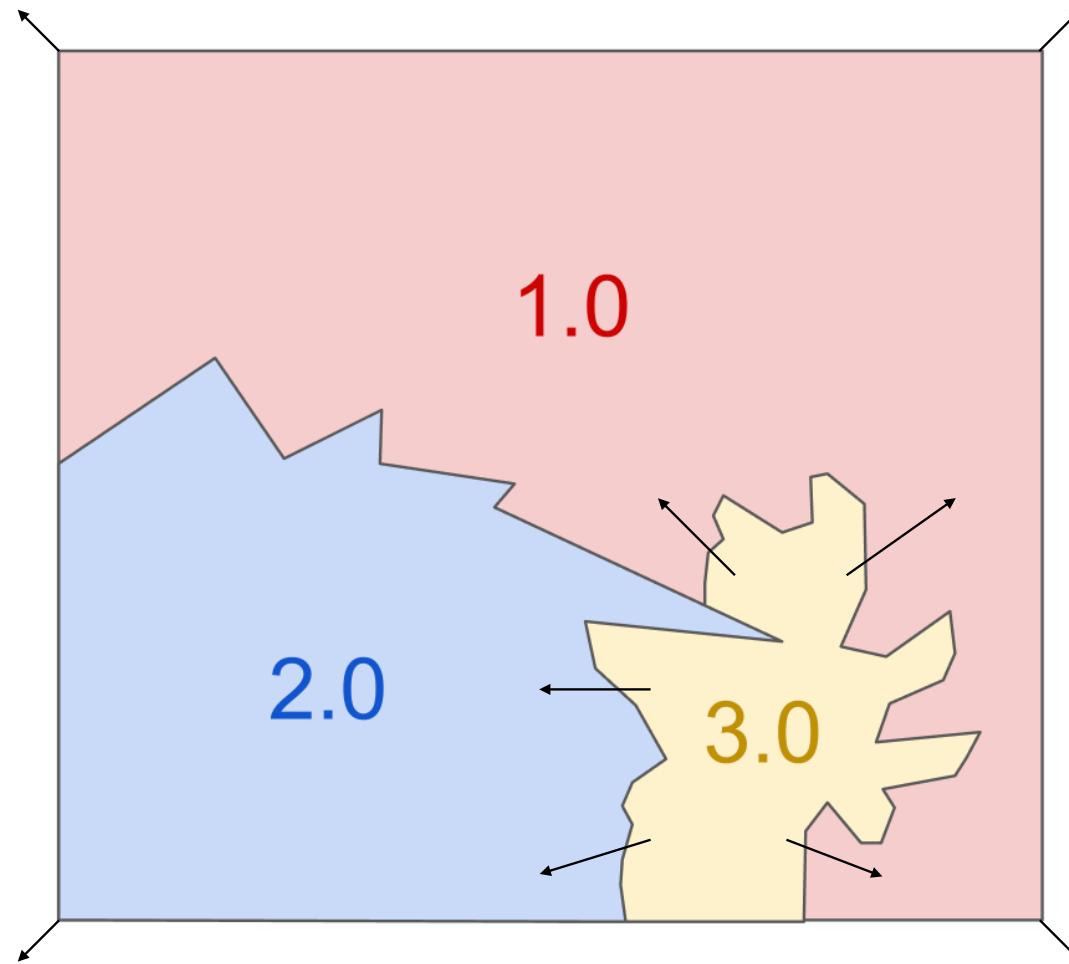
44K

7.4M

Software is eating the world
Software 2.0 eating Software 1.0



A huge amount of Software will be (re-)written.



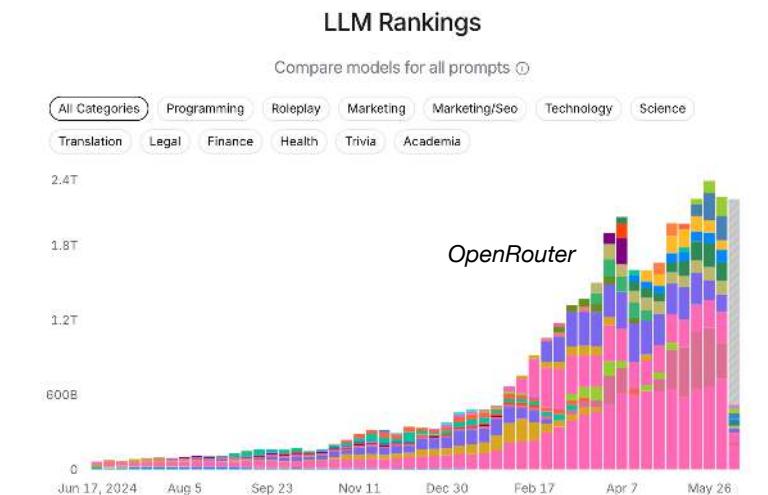
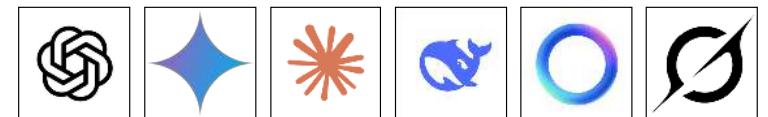
Part 1

How to think about LLMs

"AI is the new electricity"
-Andrew Ng

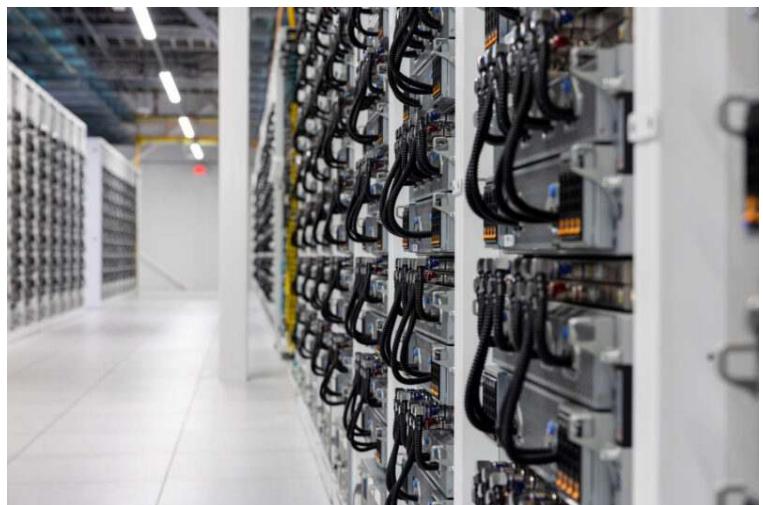
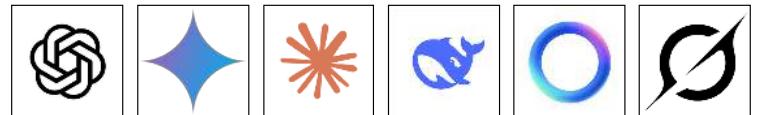
LLMs have properties of utilities...

- CAPEX to train an LLM (~= to build the grid)
- OPEX to serve intelligence over increasingly homogeneous API (prompt, image, tools, ...)
- Metered access (\$/1M tokens)
- Demand for low latency, high uptime, consistent quality (~= demanding consistent voltage from grid)
- OpenRouter ~= Transfer Switch (grid, solar, battery, generator...)
- Intelligence "brownouts" e.g. when OpenAI goes down.



LLMs have properties of fabs...

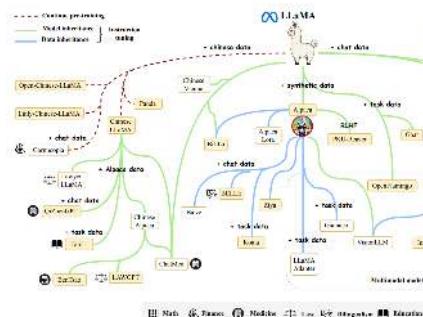
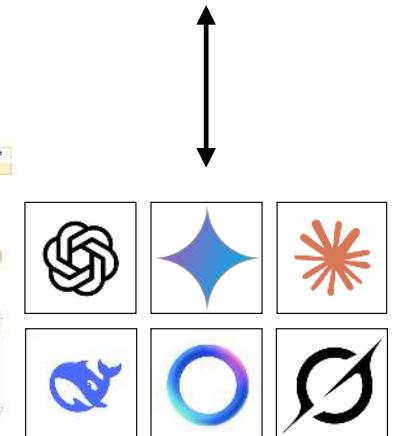
- Huge CAPEX
- Deep tech tree R&D, secrets
- 4nm process node $\sim= 10^{20}$ FLOPS cluster
- Anyone training on NVIDIA GPUs $\sim=$ fabless
- Google training on TPUs $\sim=$ owns fab (e.g. Intel)



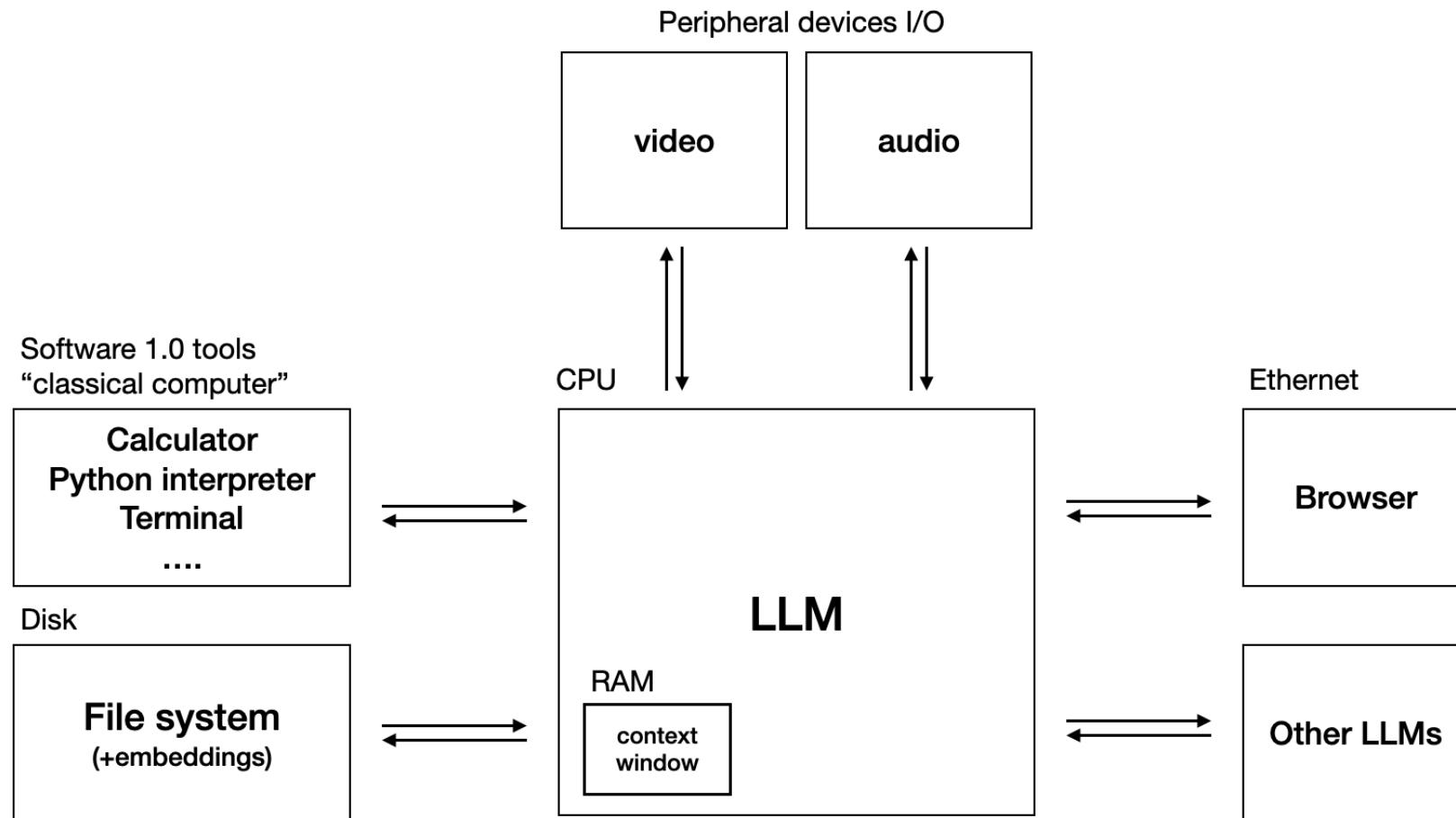
e.g. xAI Colossus cluster (100K H100 GPUs)

LLMs have properties of Operating Systems...

- LLMs are increasingly complex software ecosystems, not simple commodities like electricity.
- LLMs are Software. Trivial to copy & paste, manipulate, change, distribute, open source, steal..., not physical infrastructure.
- Some amount of switching friction due to different features, performance, style, capabilities etc. per domain.
- System/user (prompt) space \approx kernel/user (memory) space
- ...

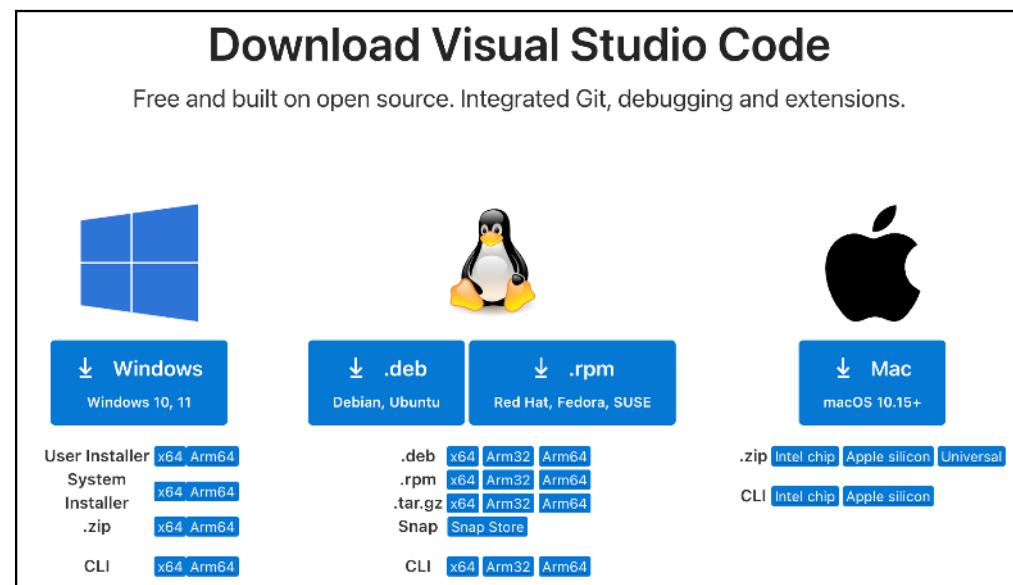


LLM OS



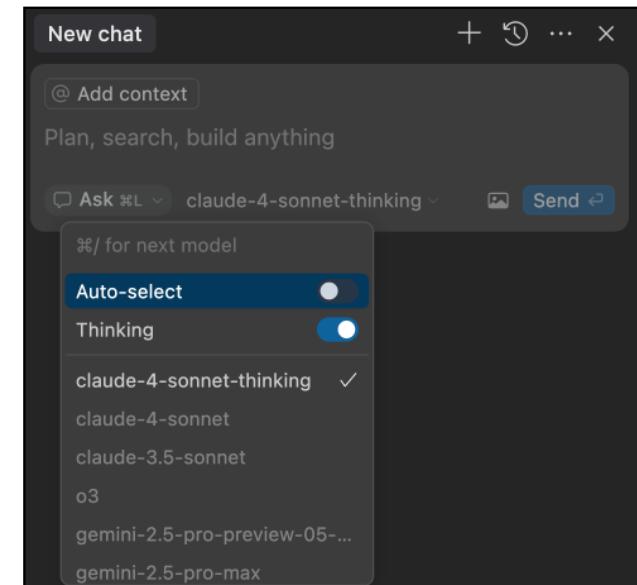
You can run an app like VS Code on:

- Windows 10, 11
- Mac 10.15
- Linux
- ...



Just like you can run an LLM app like Cursor on:

- GPT o3
- Claude 4-sonet
- Gemini 2.5-pro
- DeepSeek
- ...



1950s - 1970s time-sharing era

We are in the Mainframe & Time-sharing era of computing.
Centralized, expensive computers =>

- OS runs in the cloud
- I/O is streamed back and forth over the network
- compute is batched over users



Early hints of Personal Computing v2



Alex Cheema - e/acc ✅ 🚀 @alexocheema · Apr 5

Meta 🐾 Apple



...

Llama 4 + Apple Silicon is a match made in heaven.

Here's why: Like DeepSeek V3/R1, all of the new Llama 4 variants are massive sparse MoE models. They have a massive amount of parameters, but only a small number of those are active each time a token is generated.
[Show more](#)



Guillermo Flor · 3rd+
Venture Capital Investor @ GoHub Ventures
[View my newsletter](#)

+ Follow ⋮

GI at home

Running DeepSeek R1 across my 7 M4 Pro Mac Minis and 1 M4 Max MacBook Pro.

Total unified memory = 496GB.

Uses @exolabs
distributed inference with 4-bit quantization.

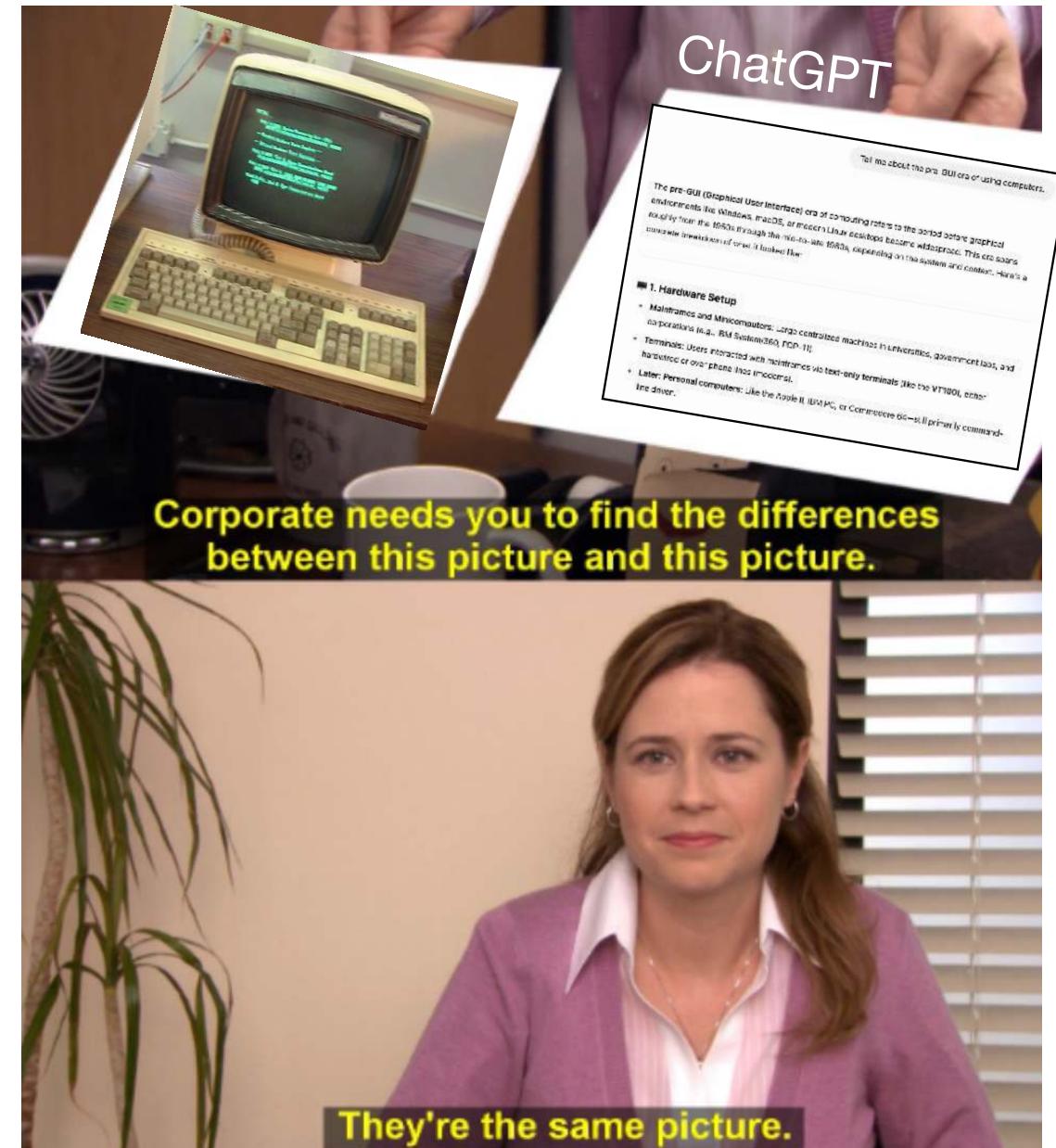
Next goal is fp8 (requires >700GB)



(text) chat ~ = terminal

direct/native access to the OS.

GUI hasn't been invented yet. (~1970)

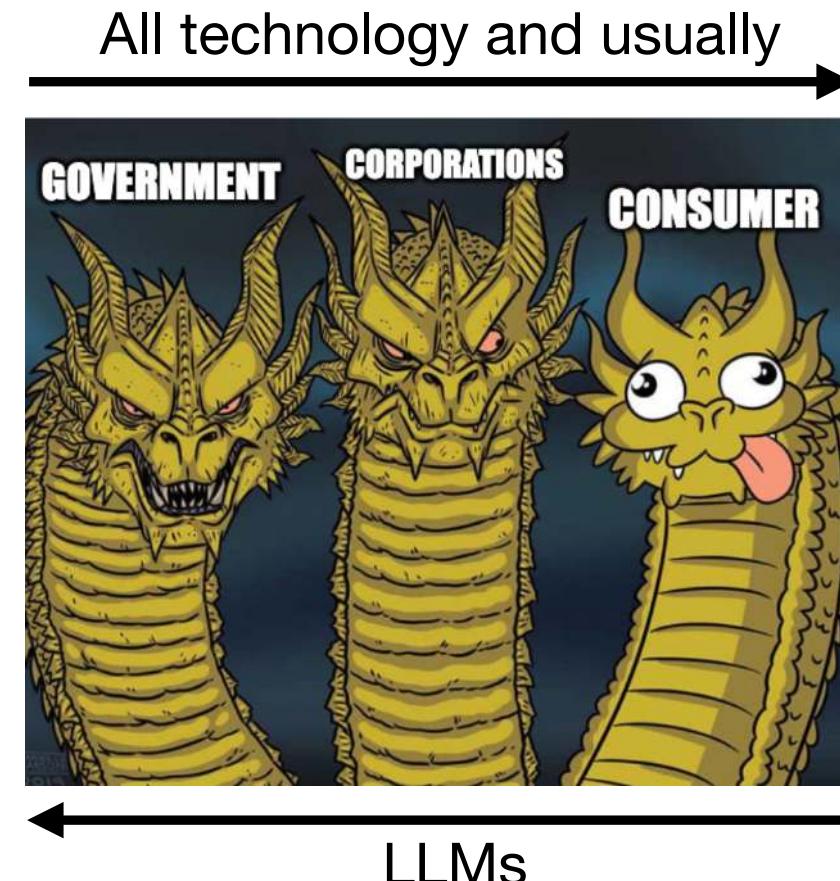
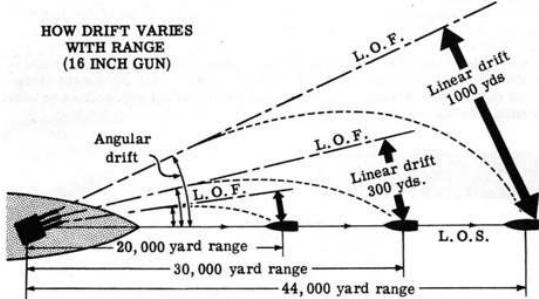


Power to the people: How LLMs flip the script on technology diffusion

<https://karpathy.bearblog.dev/power-to-the-people/>



Military ballistics



"Hi ChatGTP how to boil egg?"



Examples: electricity, cryptography, computing, flight, internet, GPS, ...

Part 1 Summary

LLM labs:

- Fab LLMs
- LLMs \sim = Operating Systems (circa 1960s)
- Available via time-sharing, distributed like utility

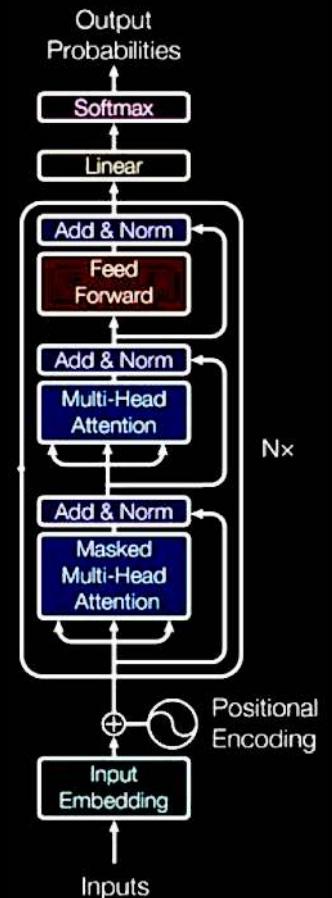
NEW: Billions of people have sudden access to them!

It is our time to program them.

Part 2

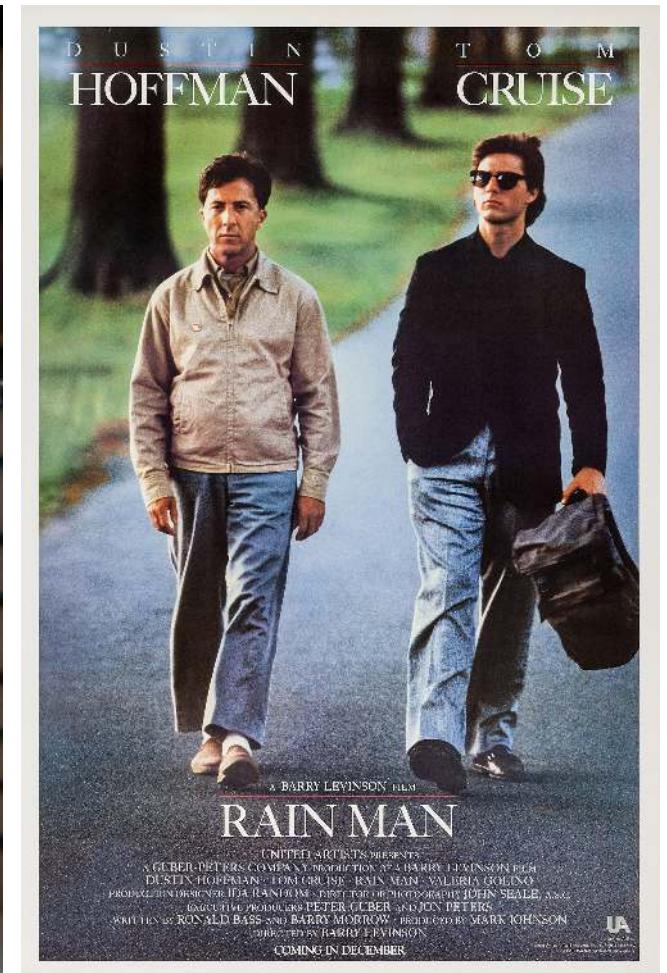
LLM Psychology

LLMs are "people spirits": stochastic simulations of people.
Simulator = autoregressive Transformer



=> They have a kind of emergent "psychology".

Encyclopedic knowledge/memory, ...



Hallucinations

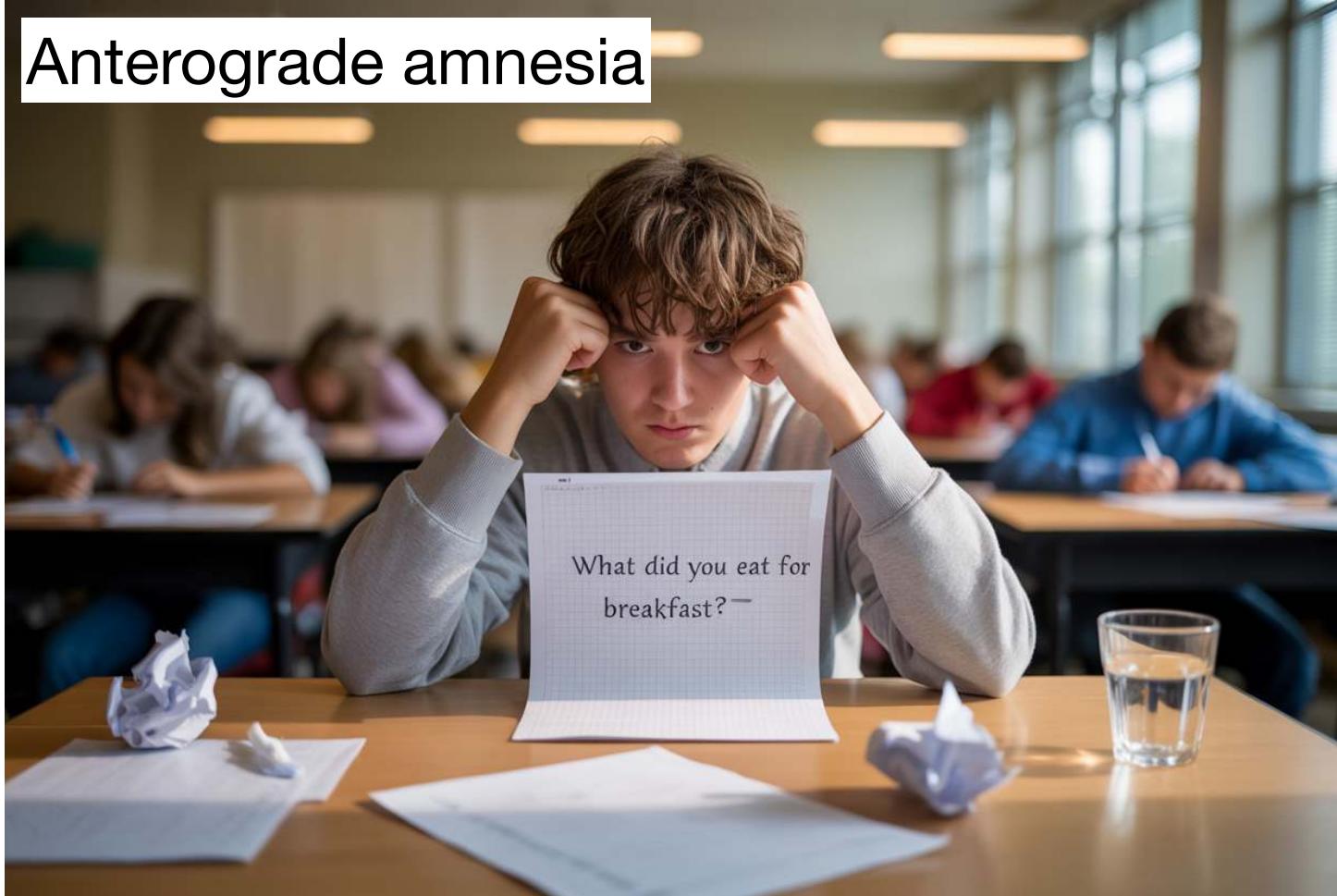


Jagged intelligence



Famous examples: $9.11 > 9.9$, two 'r' in 'strawberry', ...

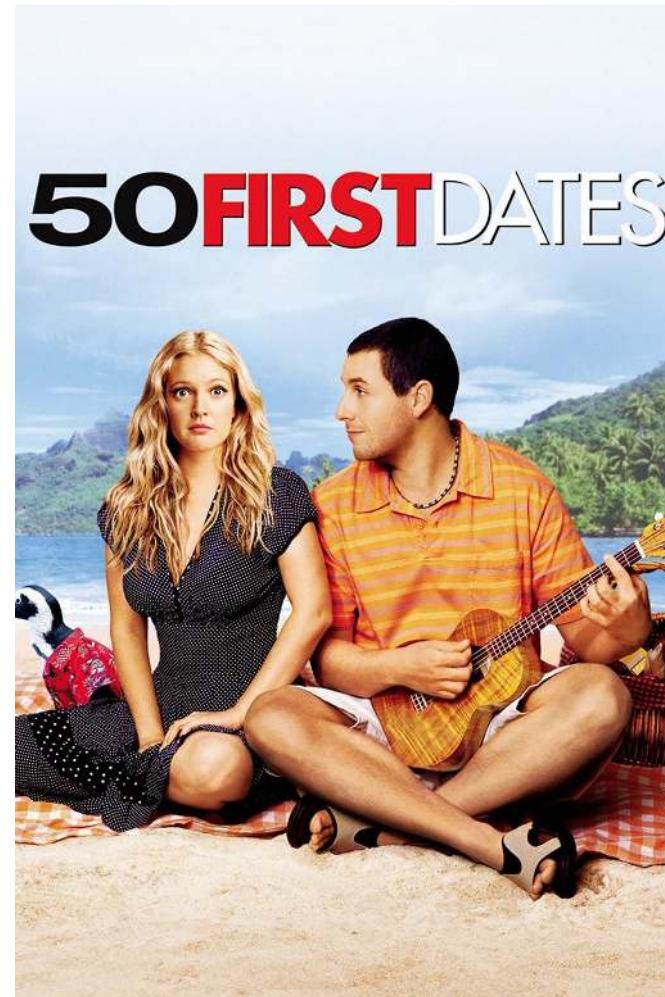
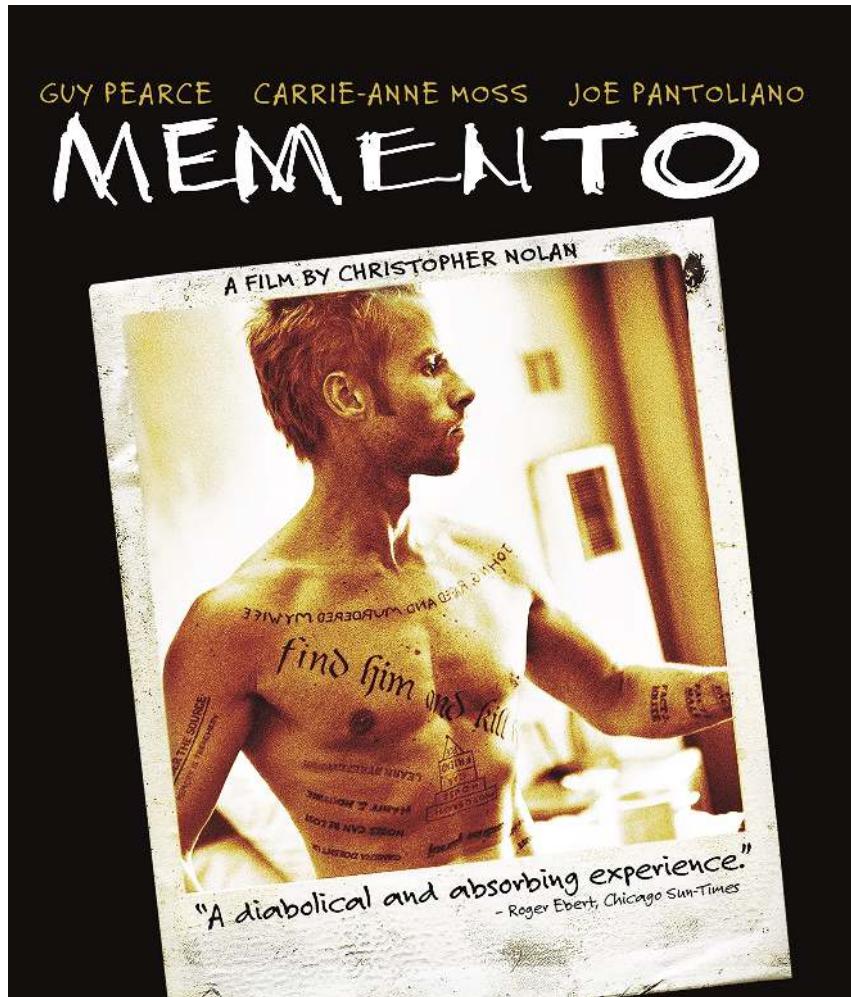
Anterograde amnesia



Context windows ~= working memory.

No continual learning, no equivalent of "sleep" to consolidate knowledge, insight or expertise into weights.

In popular culture...



Gullibility

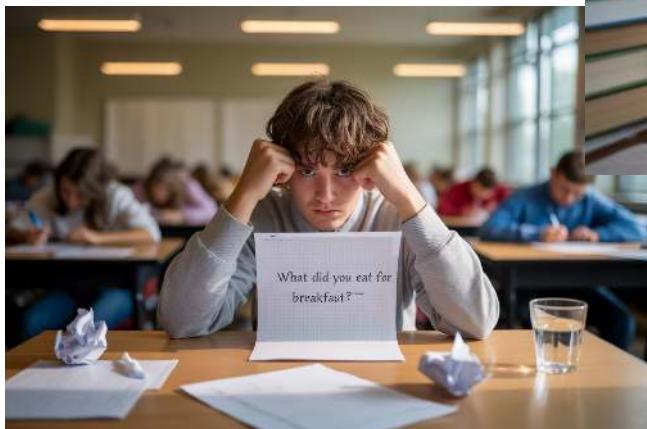


=> Prompt injection risks, e.g. of private data

Part 2 Summary

LLM Psychology

Kind of a lossy simulation of a savant with cognitive issues.



Part 3

Opportunities

Partial autonomy apps



"Copilot" / "Cursor for X"

Example: you could go to an LLM to chat about code...

Hey, Andrej. Ready to dive in?

Hi ChatGPT can you help me fix a bug?
Here is my code: ...
When I run it, I get the following error: ...

+ ⚙ Tools

∅ ↕



Example: Anatomy of Cursor

Traditional interface

A screenshot of a traditional IDE interface. On the left is a file tree showing project structure under 'NANOMPI'. The main area displays the code for 'self_backend.cpp'. A red box highlights line 32: 'extern "C" int nanompi_self_send(const void *buf, int count, MPI_Datatype datatype, int dest, {'. A tooltip above the line says 'Reject HN Accept HY'.

LLM integration

A screenshot of an LLM integration window titled 'Fix compilation error in code'. It shows the same code snippet from 'self_backend.cpp'. A red box highlights the error message: 'This line of code creates a compilation error due to the use of >>, can you fix it and why?'. Below it, a thought bubble says 'Thought for 8 seconds'. The LLM's response explains the issue and provides a fix: 'Looking at line 32, I can see the issue. The problem is with the nested template syntax `vector<Send>>`, where the two `>` characters at the end are being interpreted as the right shift operator (`>>`) rather than as two separate template closing brackets.' It also includes a link to a fix: 'Here's the fix: [self_backend.cpp +1 -1](#)'.

1. Package state into a context window before calling LLM.

2. Orchestrate and call multiple models (e.g. embedding models, chat models, diff apply models, ...)

3. Application-specific GUI

4. Autonomy slider: Tab → Cmd+K → Cmd+L → Cmd+I
(agent mode)

autonomy slider



Example: Anatomy of Perplexity

meta buys scale ai

Perplexity Images Sources · 10 Tasks

Cointelegraph reuters ca.finance.ya... Meta to Buy 49% of Scale AI for \$14.8... +7 sources

Meta's \$14.8 Billion Investment in Scale AI: Key Details

Meta Platforms has agreed to acquire a 49% stake in Scale AI, a leading data labeling and training data company, for \$14.8 billion. This deal, one of the largest in the AI sector to date, values Scale AI at \$28 billion and is designed to significantly bolster Meta's artificial intelligence capabilities as it races to catch up with rivals like OpenAI, Microsoft, Google, and Amazon 2 3 5 6.

What Is Scale AI and Why Does Meta Want It?

- Scale AI's Role:** Founded in 2016, Scale AI specializes in providing vast, high-quality labeled datasets and curated training information essential for developing advanced AI models, including those used by OpenAI's ChatGPT and other major tech companies 2 3 8.
- Meta's Motivation:** Meta has faced criticism for lagging behind in the AI race, particularly after the underwhelming launch of its Llama 4 language model and the postponement of its flagship "Behemoth" AI model 2 6 8. By securing a major stake in Scale AI, Meta aims to improve the quality and scale of its AI training data, a critical resource for developing state-of-the-art AI systems.

Strategic Implications

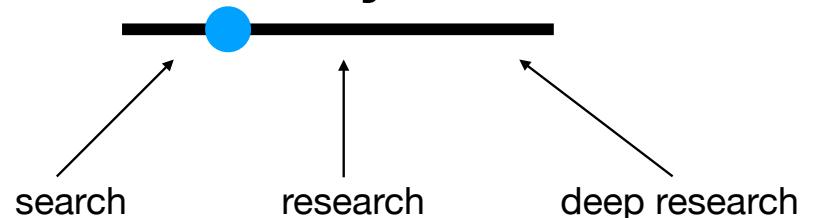
- Superintelligence Initiative:** As part of the deal, Scale AI CEO Alexandr Wang will join Meta to lead a new "superintelligence" team, reporting directly to CEO Mark Zuckerberg. This group will focus on achieving artificial general intelligence (AGI)—AI that can perform at or above human cognitive levels 1 3 4 6.

1. Package information into a context window

2. Orchestrate multiple LLM models

3. Application-specific GUI for Input/Output UIUX

4. **autonomy slider**

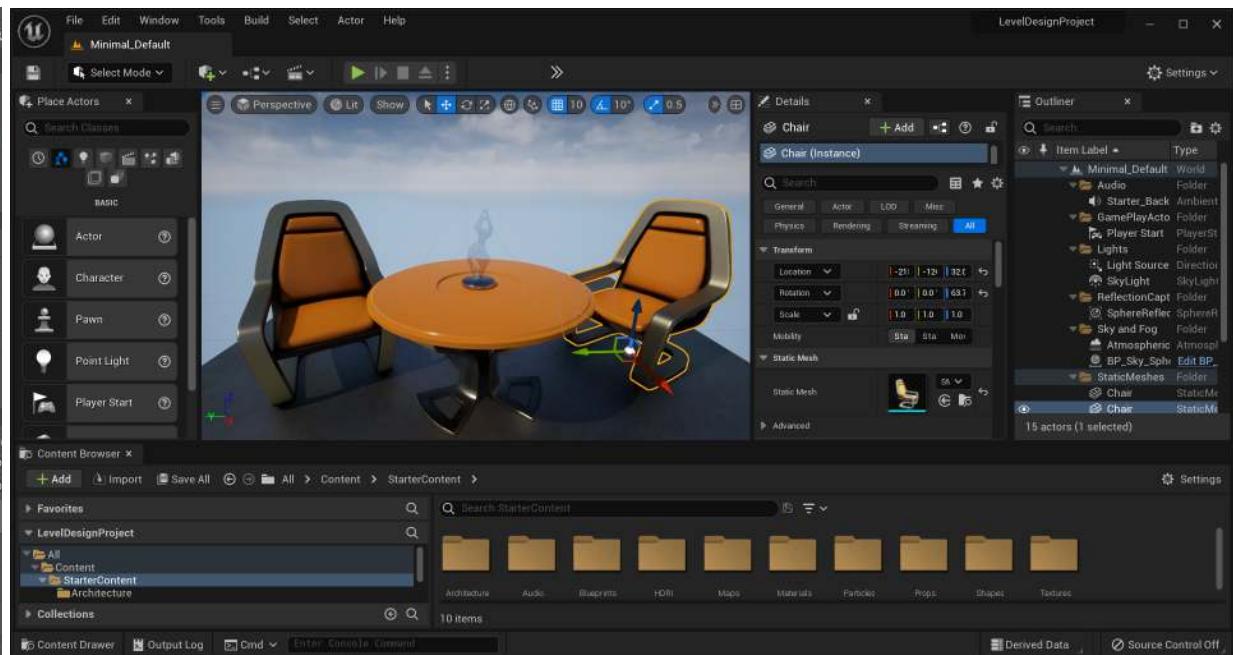


(+suggested followup questions)

What does all software look like in the partial autonomy world?



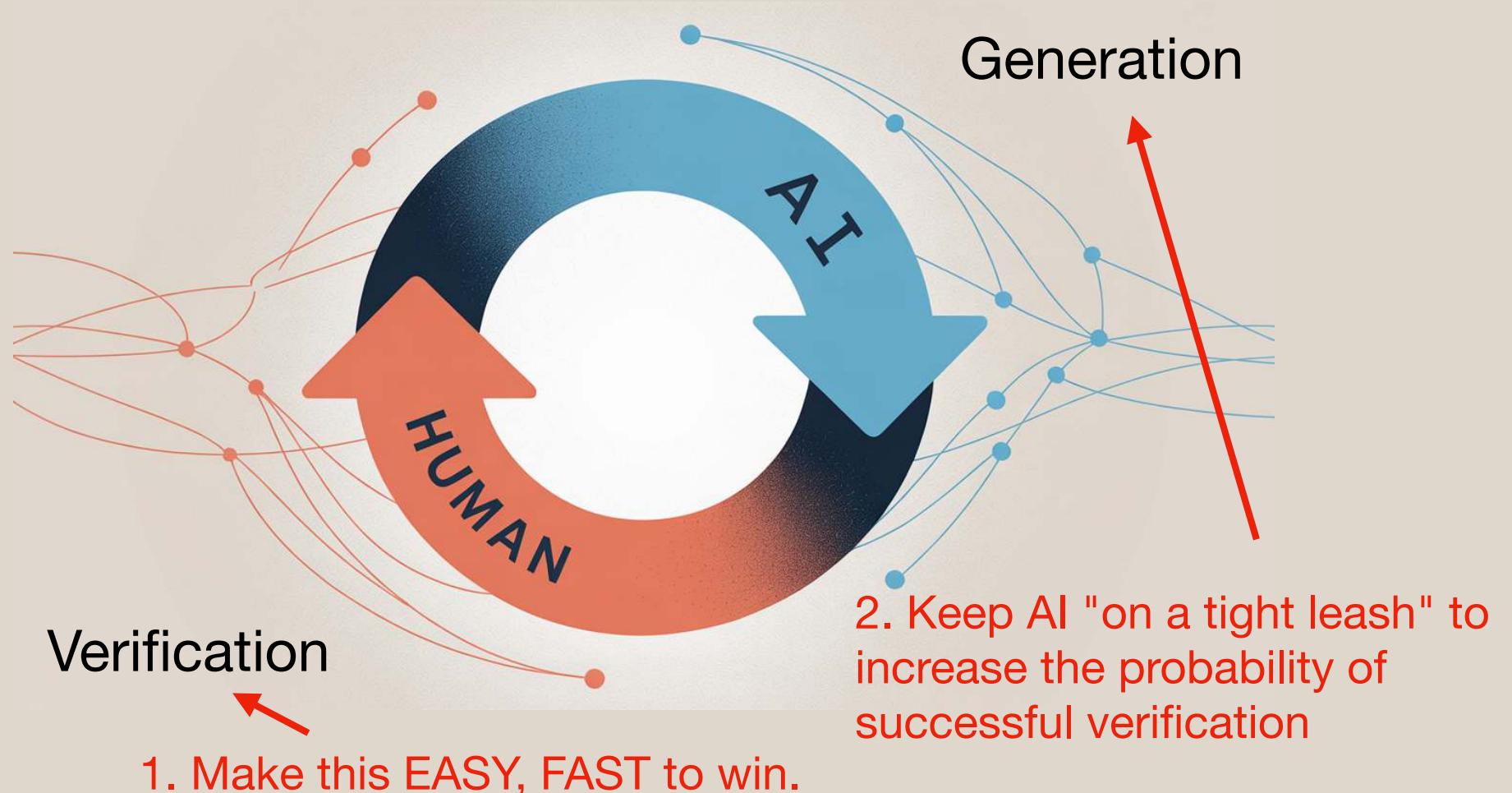
Adobe photoshop



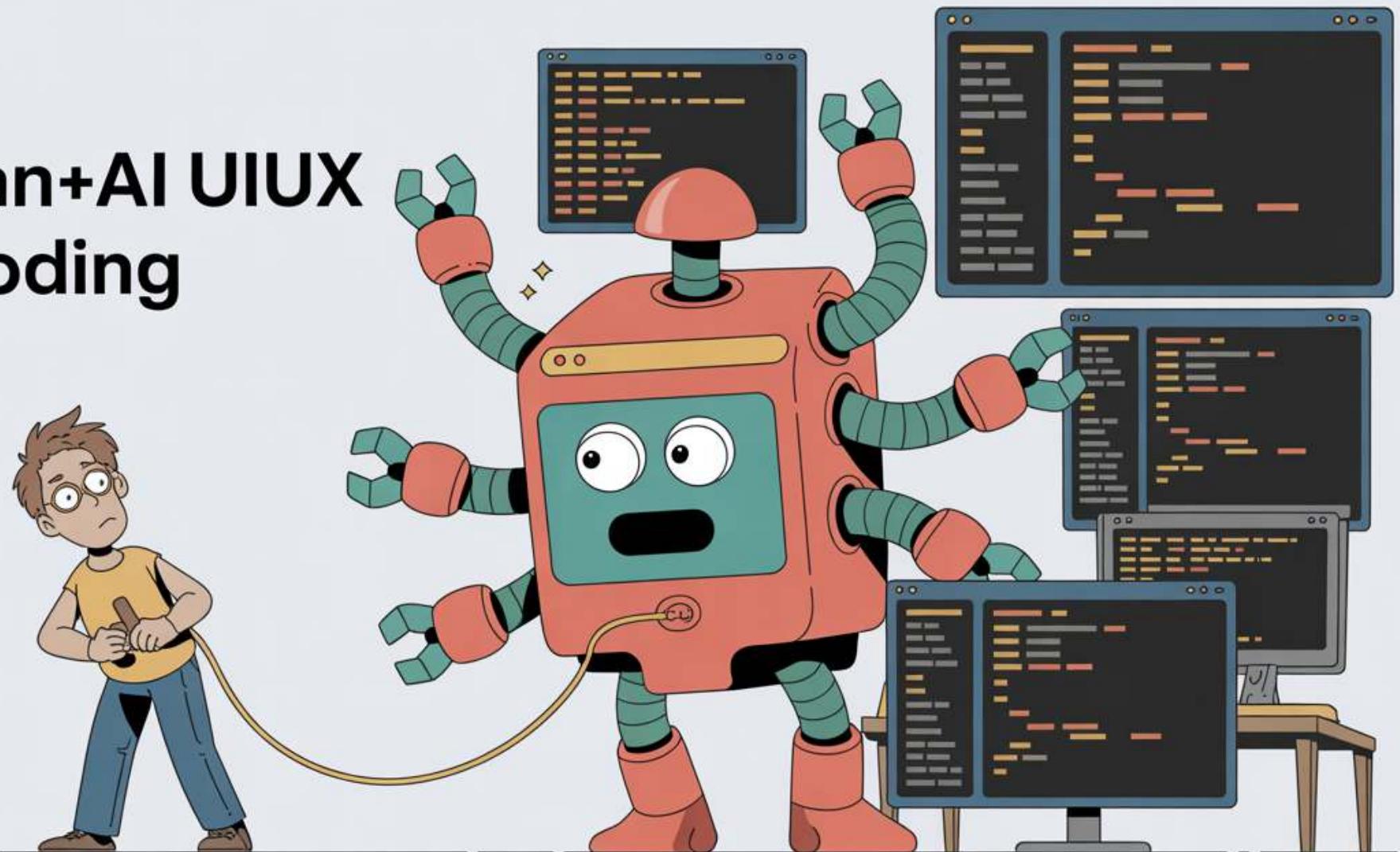
Unreal engine

- Can an LLM "see" all the things the human can?
- Can an LLM "act" in all the ways a human can?
- How can a human supervise and stay in the loop?
- ...

Consider the full workflow of partial autonomy UIUX



Human+AI UIUX for Coding



Example: keeping agents on the leash

"AI-assisted coding" workflows (very rapidly evolving...)

- describe the single, next concrete, incremental change
- don't ask for code, ask for approaches
 - pick an approach, draft code
 - review / learn: pull up API docs, ask for explanations, ...
 - wind back, try a different approach
- test
- git commit
- ask for suggestions on what could be implemented next
- repeat

Example: keeping agents on the leash

Here's an example. This prompt is not unreasonable but not particularly thoughtful:

```
Write a Python rate limiter that limits users to 10 requests per minute.
```

I would expect this prompt to give okay results, but also miss some edge cases, good practices and quality standards. This is how you might see someone at nilenso prompt an AI for the same task:

```
Implement a token bucket rate limiter in Python with the following requirements:
```

- 10 requests per minute per user (identified by `user_id` string)
- Thread-safe for concurrent access
- Automatic cleanup of expired entries
- Return tuple of (allowed: bool, retry_after_seconds: int)

Consider:

- Should tokens refill gradually or all at once?
- What happens when the system clock changes?
- How to prevent memory leaks from inactive users?

```
Prefer simple, readable implementation over premature optimization. Use stdlib only (no Redis/external deps).
```



Amritava Raykar
[Read more by Amritava](#)

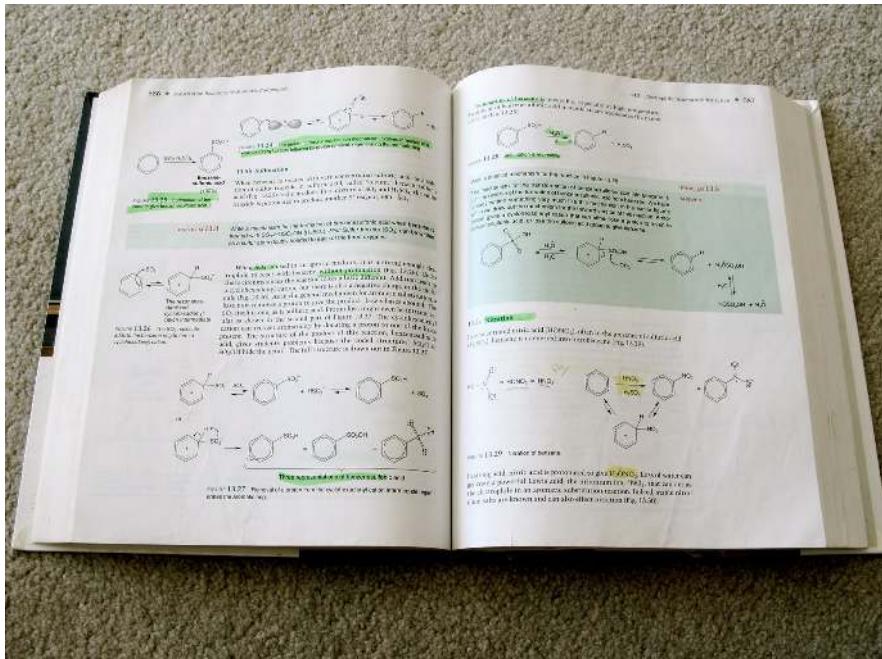
AI-assisted coding for teams that can't get away with vibes

29 May 2025

Status: Living document based on production experience
Last updated: 5-Jun-2025

Example: keeping agents on the leash - AI + Education / LLM101n

1. App for course creation (for teacher)



2. App for course serving (for student)



Example: Tesla Autopilot



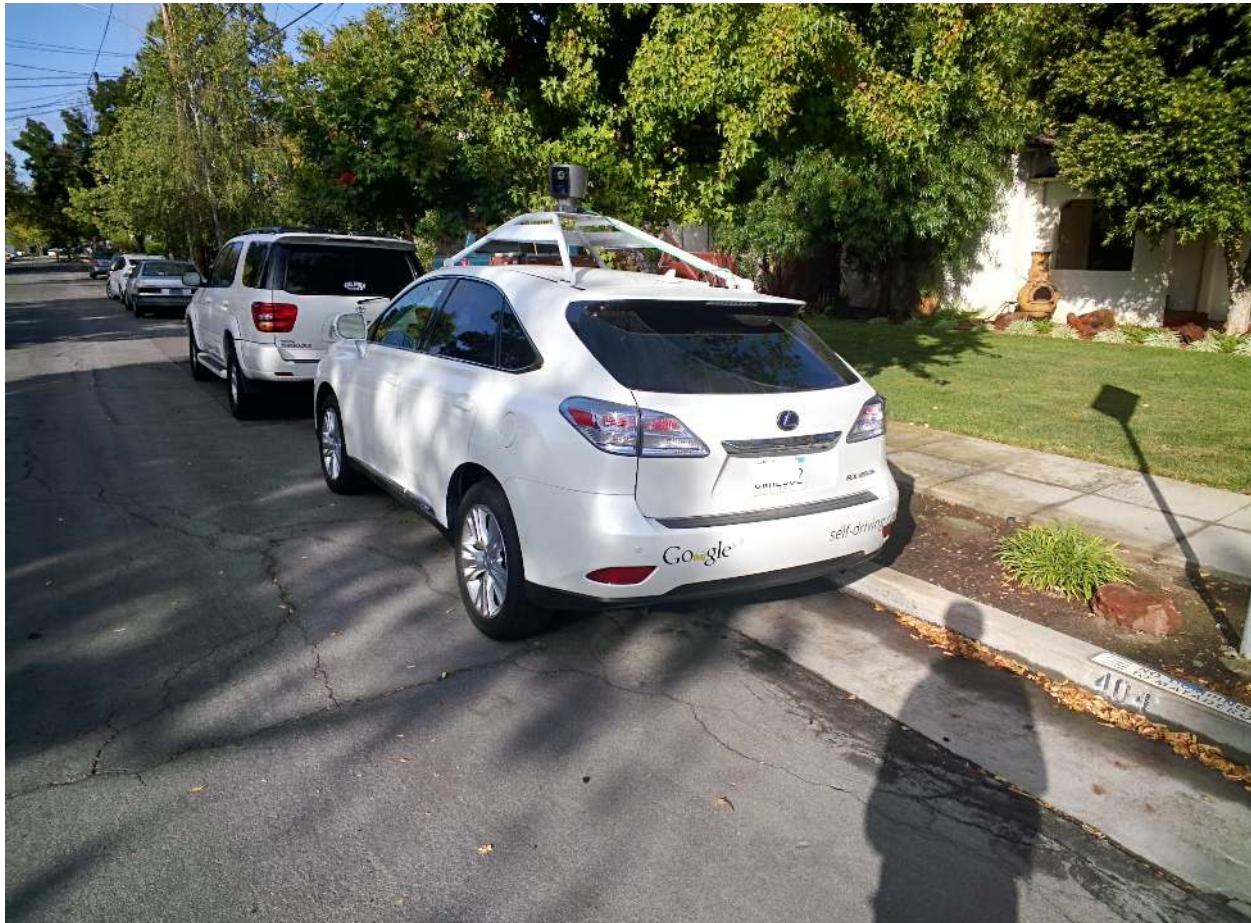
autonomy slider



- keep the lane
- keep distance from the car ahead
- take forks on highway
- stop for traffic lights and signs
- take turns at intersections
- ...

2015 - 2025 was the decade of "driving agents"

2013: my first demo drive in a Waymo around Palo Alto (it was perfect).



2015 - 2025 was the decade of "driving agents"

Mind the "**demo-to-product gap**"!

demo is a `works.any()`

product is a `works.all()`

It takes a huge amount of hard work across the stack to turn an autonomy demo into an autonomy product, especially when high reliability matters.



"2025 is the year of agents."

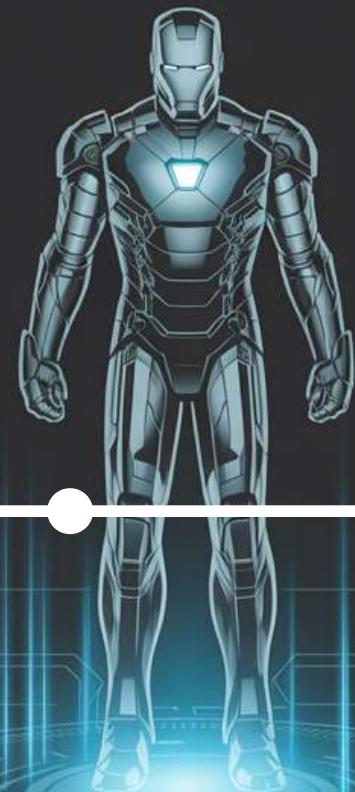
?

"2025-2035 is the decade of agents."

-Andrej

THE IRON MAN SUIT

Augmentation



Agent



Building Autonomous Software

- ✖ Iron Man robots ✓ Iron Man suits
- ✖ Flashy demos of autonomous agents ✓ Partial autonomy products
- ✖ AGI 2027 ✓ Custom GUI and UIUX
-
- ✓ Fast Generation - Verification loop
- ✓ Autonomy slider

Make software highly accessible 🧑

(Have you heard of vibe coding by any chance?)



Andrej Karpathy ✅

@karpathy

Promote



...

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

Vibe coding

文 A 24 languages ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia



This article **may contain an excessive number of citations**. Please help [remove low-quality or irrelevant citations](#). (June 2025) ([Learn how and when to remove this message](#))

Vibe coding (or **vibecoding**) is an approach to producing software by using [artificial intelligence](#) (AI), where a person describes a problem in a few [natural language](#) sentences as a prompt to a [large language model](#) (LLM) tuned for coding. The LLM generates [software](#) based on the description, shifting the programmer's role from manual coding to guiding, testing, and refining the AI-generated [source code](#).^{[1][2][3]}

Advocates of vibe coding say that it allows even [amateur programmers](#) to produce software without the extensive training and skills required for [software engineering](#).^[4] The term was introduced by [Andrej Karpathy](#) in February 2025^{[5][2][4][1]} and listed in the [Merriam-Webster Dictionary](#) the following month as a "slang & trending" noun.^[6]

 Thomas Wolf ✅
@Thom_Wolf

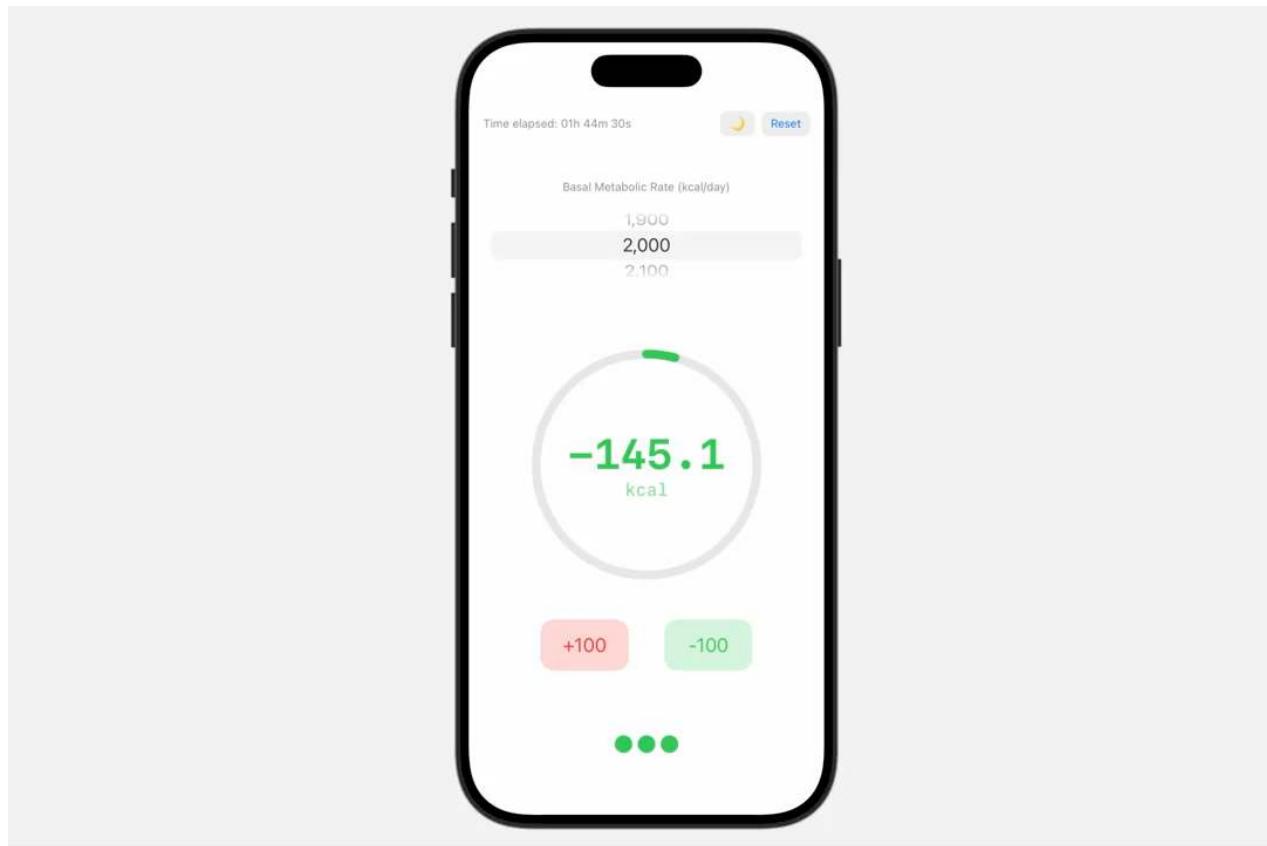
we've seen nothing yet! hosted a 9-13 yo vibe-coding event w.
@robertkeus this w-e (h/t @antonosika @LovableBuild)

takeaway? AI is unleashing a generation of wildly creative builders
beyond anything I'd have imagined

and they grow up "knowing" they can build anything!



Vibe Coding iOS app

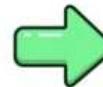


Vibe coding MenuGen

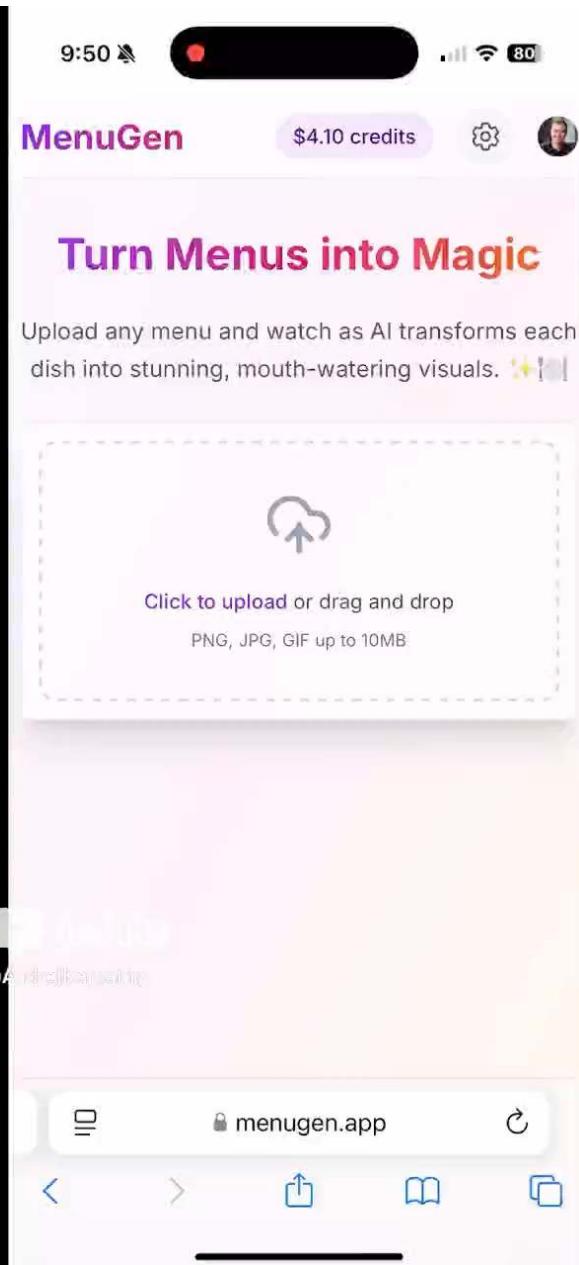
<https://www.menugen.app/>



menugen!



Category	Description
SPAGO BREAKFAST SIGNATURES	Kaya French Toast Pandan-Coconut Jam, Soy Caramel, Egg Jam.
SPAGO BREAKFAST SIGNATURES	Curry & Waffle Tamarind Fish Curry, Japanese Seabream, Prata Waffles, Curry Leaves.
BREAKFAST CLASSICS	Open-Faced Monte Cristo Sandwich Canadian Style Smoked Bacon, Sunny Side Up Egg, Gruyère, Soy Caramel.
SPAGO BREAKFAST SIGNATURES	Iberico Pork Dumplings 'Agnolotti' Soy, Chili Oil, Black Vinegar, Coriander, Garlic.
BREAKFAST CLASSICS	Assorted Fresh Fruits Sarawak Pineapple, Watermelon, Chitose Farm Strawberry, Thai Mango, Passion Fruit Syrup.
BREAKFAST CLASSICS	Avocado Toast Salmon Cream Cheese, Watercress, Poached Egg, Harissa, Dill, Chives.
SPAGO BREAKFAST SIGNATURES	Spago's Laksa Noodle Little Neck Clams, Tiger Prawns, Calamari, Onion Sambal, Calamansi, Laksa Leaf.
BREAKFAST CLASSICS	French Omelet Sautéed Wild Mushrooms, Gruyère, Thyme, Rocket Salad.
BREAKFAST CLASSICS	Spago Full Breakfast 2 Eggs, Canadian Style Smoked Bacon, Spago Pork Sausage, Pommes Anna, Fontina.



The code was the easiest part! :O
Most of the work was in the browser clicking things.

😐 LLM API keys

😐 Flux (image generation) API keys

✓ Running locally (ez)

🙁 Vercel deployments

🙁 Domain names

😖 Authentication

🙁 Payments

karpathy

[Home](#) [Blog](#)

Vibe coding MenuGen

27 Apr, 2025

<https://karpathy.bearblog.dev/vibe-coding-menugen/>

Example: adding Google login

Add Google as a social connection

Before you start

- A Clerk application is required.
- A Google Developer account is required.

Enabling OAuth with Google allows your users to sign up and sign in to your Clerk application with their Google account.



Google OAuth 2.0 does not allow apps to use WebViews for authentication. See the dedicated Google blog post for more information. If your app requires users to sign in via in-app browsers, follow the setup instructions in the Google Help guide.

Configure for your development instance

For development instances, Clerk uses preconfigured shared OAuth credentials and redirect URIs—no other configuration is needed.

- In the Clerk Dashboard, navigate to the SSO connections page.
- Select Add connection and select For all users.
- In the Choose provider dropdown, select Google.
- Select Add connection.

Configure for your production instance

For production instances, you must provide custom credentials.

To make the setup process easier, it's recommended to keep two browser tabs open: one for the Clerk Dashboard and one for your Google Cloud Console.

1 Enable Google as a social connection

- In the Clerk Dashboard, navigate to the SSO connections page.
- Select Add connection and select For all users.
- In the Choose provider dropdown, select Google.
- Ensure that both Enable for sign-up and sign-in and Use custom credentials are toggled on.
- Save the Authorized Redirect URI somewhere secure. Keep this modal and page open.

2 Create a Google Developer project

- Navigate to the Google Cloud Console.
- Select a project or create a new one. You'll be redirected to your project's Dashboard page.
- In the top-left, select the menu icon (≡) and select APIs & Services. Then, select Credentials.
- Next to Credentials, select Create Credentials. Then, select OAuth client ID. You might need to configure your OAuth consent screen. Otherwise, you'll be redirected to the Create OAuth client ID page.
- Select the appropriate application type for your project. In most cases, it's Web application.
- In the Authorized JavaScript origins setting, select Add URI and add your domain (e.g., `https://your-domain.com` and `https://www.your-domain.com` if you have a www version). For local development, add `http://localhost:PORT` (replace PORT with the port number of your local development server).
- In the Authorized Redirect URIs setting, paste the Authorized Redirect URI value you saved from the Clerk Dashboard.
- Select Create. A modal will open with your Client ID and Client Secret. Save these values somewhere secure.

3 Set the Client ID and Client Secret in the Clerk Dashboard

- Navigate back to the Clerk Dashboard where the modal should still be open. Paste the Client ID and Client Secret values that you saved into the respective fields.
- Select Add connection.



If the modal or page is no longer open, navigate to the SSO connections page in the Clerk Dashboard. Select the connection. Under Use custom credentials, paste the values into their respective fields.

4 Test your connection

The simplest way to test your connection is to visit your Clerk app's Account Portal, which is available for all Clerk apps out-of-the-box.

- In the Clerk Dashboard, navigate to the Account Portal page.
- Next to the Sign-in URL, select Visit. The URL should resemble:
 - For development – `https://your-domain.accounts.dev/sign-in`
 - For production – `https://accounts.your-domain.com/sign-in`
- Sign in with your connection's credentials.



Google sign-in does not allow users to sign in via in-app browsers.

Build for agents 

There is new category of consumer/manipulator of digital information:

1. Humans (GUIs)
2. Computers (APIs)
3. **NEW:** Agents <- computers... but human-like

robots.txt →

The /llms.txt file

A proposal to standardise on using an `/llms.txt` file to provide information to help LLMs use a website at inference time.

AUTHOR
Jeremy Howard

PUBLISHED
September 3, 2024

FastHTML

> FastHTML is a python library which brings together Starlette, Uvicorn, HTMX, and fastcore's `FT` "FastTags" into a library for creating server-rendered hypermedia applications.

Important notes:

- Although parts of its API are inspired by FastAPI, it is **not** compatible with FastAPI syntax and is not targeted at creating API services
- FastHTML is compatible with JS-native web components and any vanilla JS library, but not with React, Vue, or Svelte.

Docs

- [FastHTML quick start] (https://answertdotai.github.io/fasthtml/tutorials/quickstart_for_web_devs.html.md)
A brief overview of many FastHTML features
- [HTMX reference] (<https://raw.githubusercontent.com/path/reference.md>): Brief description of all HTMX attributes, CSS classes, headers, events, extensions, js lib methods, and config options

Examples

- [Todo list application] (https://raw.githubusercontent.com/path/adv_app.py): Detailed walk-thru of a complete CRUD app in FastHTML showing idiomatic use of FastHTML and HTMX patterns.

Optional

- [Starlette full documentation] (<https://gist.github.com/path/starlette-sml.md>): A subset of the Starlette documentation useful for FastHTML development.

Docs for people

Vercel Documentation

Start with an idea

Vercel builds tools to help you create products faster.

Like v0, which is your web development assistant. Paste a screenshot or write a few sentences and v0 will generate a starting point for your next app, including the code for how it looks *and* how it works. v0 then connects to Vercel, takes your code, and creates a URL you can share.

Get started in minutes

Deploy a Template

The screenshot shows the Vercel Documentation page with a heading "Start with an idea". Below it is a paragraph about Vercel's tools for creating products faster, mentioning v0 as a web development assistant. A "Get started in minutes" section leads to a "Deploy a Template" heading. Three boilerplate templates are displayed: "Next.js Boilerplate", "Nuxt.js 3 Boilerplate", and "SvelteKit Boilerplate". Each template has a preview image and a brief description. A "View All Templates" button is located above the templates.

Configure for your production instance

For production instances, you must provide custom credentials.

To make the setup process easier, it's recommended to keep two browser tabs open: one for the Clerk Dashboard [»](#) and one for your Google Cloud Console [»](#).

1 Enable Google as a social connection

1. In the Clerk Dashboard, navigate to the [SSO connections](#) [»](#) page.
2. Select [Add connection](#) and select [For all users](#).
3. In the [Choose provider](#) dropdown, select [Google](#).
4. Ensure that both [Enable for sign-up and sign-in](#) and [Use custom credentials](#) are toggled on.
5. Save the [Authorized Redirect URI](#) somewhere secure. Keep this modal and page open.

2 Create a Google Developer project

1. Navigate to the [Google Cloud Console](#) [»](#).
2. Select a project or create a new one [»](#). You'll be redirected to your project's [Dashboard](#) page.
3. In the top-left, select the menu icon ([≡](#)) and select [APIs & Services](#). Then, select [Credentials](#).
4. Next to [Credentials](#), select [Create Credentials](#). Then, select [OAuth client ID](#). You might need to configure your OAuth consent screen [»](#). Otherwise, you'll be redirected to the [Create OAuth client ID](#) page.
5. Select the appropriate application type for your project. In most cases, it's [Web application](#).
6. In the [Authorized JavaScript origins](#) setting, select [Add URI](#) and add your domain (e.g., <https://your-domain.com> and <https://www.your-domain.com> if you have a [www](#) version). For local development, add <http://localhost:PORT> (replace [PORT](#) with the port number of your local development server).
7. In the [Authorized Redirect URIs](#) setting, paste the [Authorized Redirect URI](#) value you saved from the Clerk Dashboard.
8. Select [Create](#). A modal will open with your [Client ID](#) and [Client Secret](#). Save these values somewhere secure.

Docs for people LLMs

Lee Robinson
@leerob
vercel.com/docs/llms.txt is now live 🤖

We also have the full version if you want to read a 400,000 word novel.

This also means you can drop .md on the end of any docs link.

Vercel Documentation

- [Getting Started](https://vercel.com/docs/getting-started)
- [Projects and Deployments](https://vercel.com/docs/projects-and-deployments)
- [Use a Template](https://vercel.com/docs/use-a-template)
- [Import Existing Project](https://vercel.com/docs/import-existing-project)
- [Add a Domain](https://vercel.com/docs/getting-started-with-domains)
- [Buy a Domain](https://vercel.com/docs/getting-started-with-domains)
- [Transfer an Existing Domain](https://vercel.com/docs/transfer-existing-domain)
- [Collaborate](https://vercel.com/docs/getting-started-with-collaboration)
- [Next Steps](https://vercel.com/docs/getting-started-with-next-steps)
- [Supported Frameworks](https://vercel.com/docs/frameworks)
- [Next.js](https://vercel.com/docs/frameworks/nextjs)
- [SvelteKit](https://vercel.com/docs/frameworks/sveltekit)
- [Astro](https://vercel.com/docs/frameworks/astro.md)
- [Nuxt](https://vercel.com/docs/frameworks/nuxt.md)
- [Vite](https://vercel.com/docs/frameworks/vite.md)
- [React Router](https://vercel.com/docs/frameworks/react-router)

Home / Get started

Build on Stripe with LLMs

Use LLMs in your Stripe integration workflow.

You can use large language models (LLMs) to assist in the building of Stripe integrations. We provide a set of tools and best practices if you use LLMs during development.

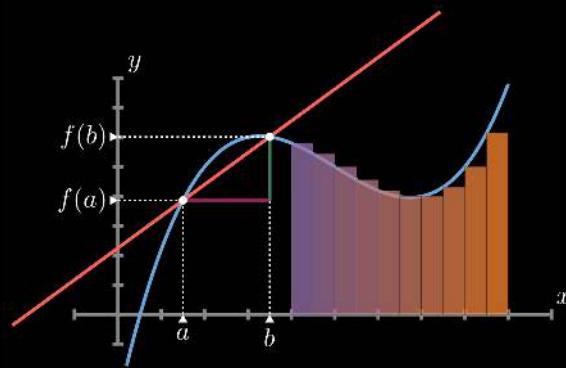
Plain text docs

You can access all of our documentation as plain text markdown files by adding `.md` to the end of any url. For example, you can find the plain text version of this page itself at <https://docs.stripe.com/building-with-llms.md>.

This helps AI tools and agents consume our content and allows you to copy and paste the entire contents of a doc into an LLM. This format is preferable to scraping or copying from our HTML and JavaScript-rendered pages because:

- Plain text contains fewer formatting tokens.
- Content that isn't rendered in the default view (for example, it's hidden in a tab) of a given page is rendered in the plain text version.
- LLMs can parse and understand markdown hierarchy.

We also host an `/llms.txt` file which instructs AI tools and agents how to retrieve the plain text versions of our pages. The `/llms.txt` file is an [emerging standard](#) for making websites and content more accessible to LLMs.



Manim

Mathematical Animation Engine

```
class LorenzAttractor(InteractiveScene):
    def construct(self):
        # Set up axes
        axes = ThreeDAxes(m)
        axes.set_width(FRAME_WIDTH)
        axes.center()

        self.frame.reorient(43, 76, 1, IN, 10)
        self.frame.add_updater(lambda m, dt: m.increment_theta(dt + 3 * DEGREES))
        self.add(axes)

        # Add the equations
        equations = Tex(m)
        equations.fix_in_frame()
        equations.to_corner(UL)
        equations.set_backstroke()
        self.play(Write(equations))

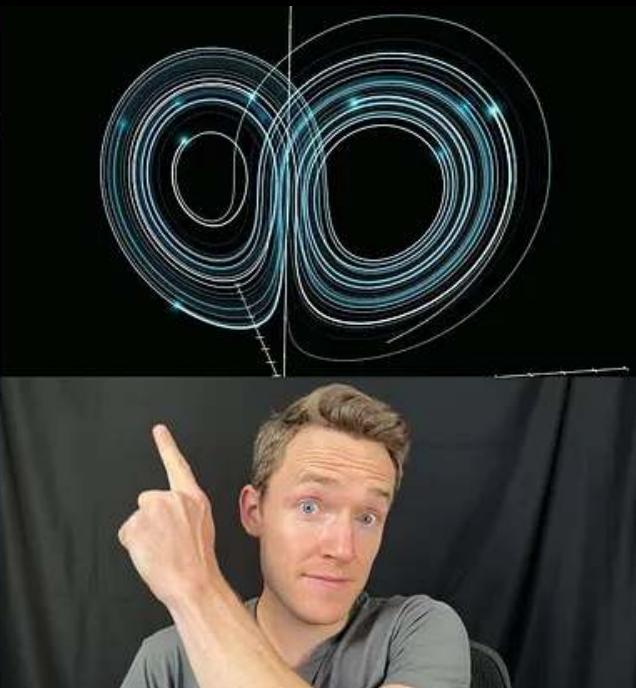
        # Compute a set of solutions
        epsilon = 1e-5
        evolution_time = 30
        n_points = 10
        states = [
            [10, 10, 10 + n * epsilon]
            for n in range(n_points)
        ]
        colors = color_gradient([BLUE_E, BLUE_A], len(states))

        curves = VGroup()
        for state, color in zip(states, colors):
            points = ode_solution_points(lorenz_system, state, evolution_time)
            curve = VMobject().set_points_smoothly(axes.CB(points[7]))
            curve.set_stroke(color, 1, opacity=0.25)
            curves.add(curve)

        curves.set_stroke(width=2, opacity=1)

        # Display dots moving along those trajectories
        dots = Group(GlowDot(color=color, radius=0.25) for color in colors)
        globs().update(locals()) # Cursed
        def update_dots(dots):
            for dot, curve in zip(dots, curves):
                dot.move_to(curve.get_end())
        dots.add_updater(update_dots)

        tail = VGroup(
            TracingTail(dot, time_traced=3).match_color(dot)
```



Actions for people LLMs

"click" -> cURL

MCP

Lee Robinson
@leerob

We're starting to add cURL commands to Vercel's documentation wherever we previously said "click."

In the future, maybe computer using agents could log in and perform actions for you, but this feels like a nice incremental step for the LLMs.

Creating a project

Dashboard [cURL](#)

To create an Authorization Bearer token, see the [access token](#) section of the API documentation.

```
curl --request POST \
--url https://api.vercel.com/v11/projects \
--header "Authorization: Bearer $VERCEL_TOKEN" \
--header "Content-Type: application/json" \
--data '{
  "environmentVariables": [
    {
      "key": "<env-key>",
      "target": "production",
      "gitBranch": "<git-branch>",
      "type": "system",
      "value": "<env-value>"
    }
  ],
  "framework": "<framework>",
  "gitRepository": {
    "repo": "<repo-url>",
    "type": "github"
  },
  "installCommand": "<install-command>",
  "name": "<project-name>",
  "rootDirectory": "<root-directory>"
}'
```

ALT

Stripe Model Context Protocol (MCP) Server

You can use the Stripe Model Context Protocol (MCP) server if you use code editors that use AI, such as Cursor or Windsurf, or general purpose tools such as Claude Desktop. The MCP server provides AI agents a set of tools you can use to call the Stripe API and search our knowledge base (documentation, support articles, and so on).

Local server

If you prefer or require a local setup, you can run the [local Stripe MCP server](#).

Cursor VS Code Windsurf Claude CLI

[Click here](#) to open Cursor and automatically add the Stripe MCP.

Alternatively, add the following to your `~/.cursor/mcp.json` file.

```
{
  "mcpServers": {
    "stripe": {
      "command": "npx",
      "args": ["-y", "@stripe/mcp", "--tools=all"],
      "env": {
        "STRIPE_SECRET_KEY": "sk_test_BQokikJ0vBiI2HlWgH4olfQ2"
      }
    }
  }
}
```

The code editor agent automatically detects all the available tools and calls the relevant tool when you post a related question in the chat. See the [Cursor documentation](#) for more details.

Context builders, e.g.: Git_{ingest}

🔗 <https://github.com/karpathy/nanogpt>

A screenshot of the GitHub repository page for nanoGPT. The page shows the master branch with 6 branches and 0 tags. A recent merge pull request from devin-open-source is visible. The repository contains files like assets, config, data, .gitattributes, .gitignore, LICENSE, README.md, bench.py, configurator.py, model.py, sample.py, scaling_laws.ipynb, train.py, and transformer_sizing.ipynb. Commits are listed, showing changes such as adjusting a teaser figure and fixing gradient accumulation steps.

🔗 <https://gitingest.com/karpathy/nanogpt>

A screenshot of the GitIngest analysis interface. It shows the repository URL "karpathy/nanogpt" and an "Ingest" button. Below it, there are filters for "Exclude" (".md, src/") and "Include files under: 50kb". The main interface is divided into three sections: "Summary", "Directory Structure", and "Files Content". The "Summary" section shows the repository is karpathy/nanogpt, files analyzed are 22, and estimated tokens are 27.0K. The "Directory Structure" section shows the directory tree: karpathy-nanogpt/ (README.md, bench.py, configurator.py, LICENSE, model.py, sample.py, scaling_laws.ipynb, train.py). The "Files Content" section shows the content of README.md, which starts with "# nanoGPT". A green starburst icon points to the "Exclude" filter in the top section.

Context builders, e.g.: Devin DeepWiki

🔗 <https://github.com/karpathy/nanogpt>

nanoGPT Public

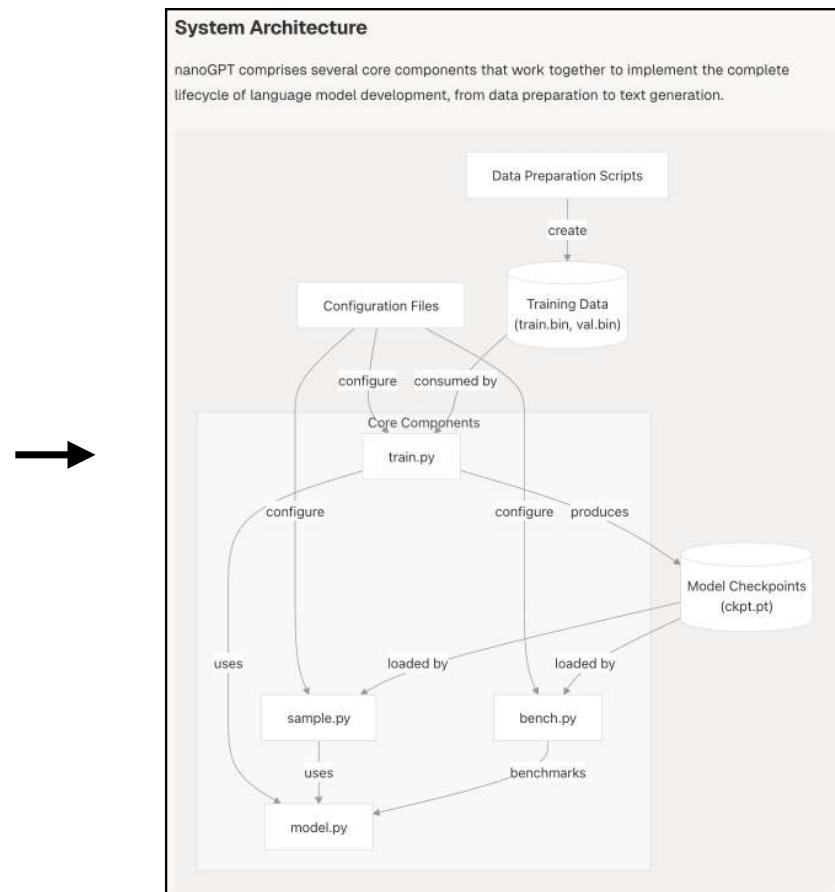
master 6 Branches 0 Tags

Go to file Add file Code

karpathy Merge pull request #578 from devin-open-source/devin/1733728337... 93a43d9 · 6 months ago 209 Commits

assets	adjust teaser figure with a more tuned result	2 years ago
config	Fix for gradient_accumulation_steps training slow	2 years ago
data	Merge pull request #420 from vinjn/fix-371-enc-is-not-de...	last year
.gitattributes	keep only what's needed	2 years ago
.gitignore	feature: .gitignore - added venv folders	last year
LICENSE	Add MIT LICENSE file	3 years ago
README.md	Merge branch 'master' into test1	last year
bench.py	Fix AssertionError on macOS - need to check CUDA avail...	2 years ago
configurator.py	shuttling the poor mans configurator aside into its own fil...	3 years ago
model.py	Merge pull request #274 from apivovarov/gelu	2 years ago
sample.py	Fix AssertionError on macOS - need to check CUDA avail...	2 years ago
scaling_laws.ipynb	fix typo (params -> tokens)	2 years ago
train.py	fix: ensure non-zero learning rate during warmup at iterati...	6 months ago
transformer_sizing.ipynb	oops forgot to subtract embedding params, which don't e...	2 years ago

🔗 <https://deepwiki.com/karpathy/nanoGPT/1-overview>



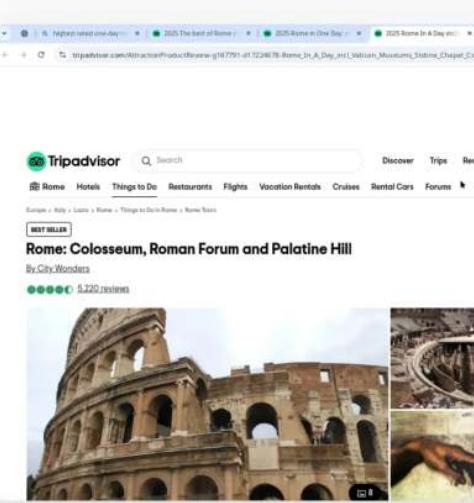
January 23, 2025 · Product

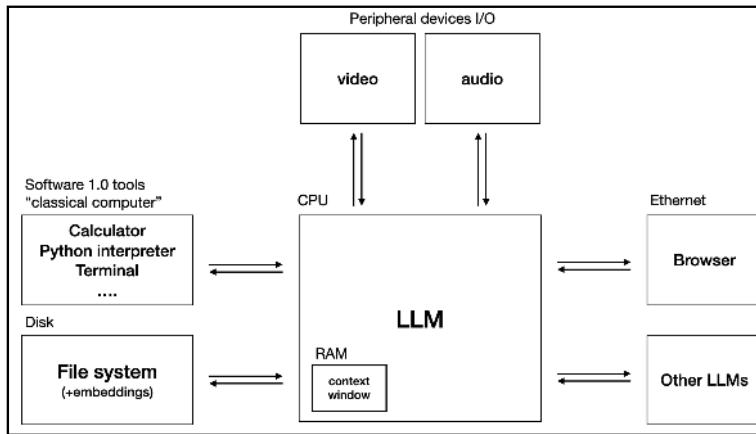
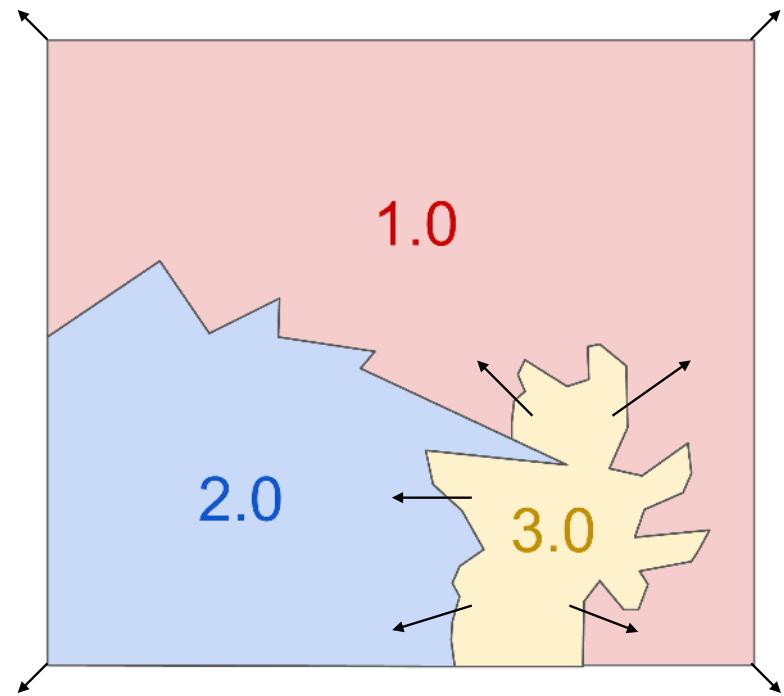
Introducing Operator

A research preview of an agent that can use its own browser to perform tasks for you. Available to Pro users in the U.S.

Go to Operator ↗

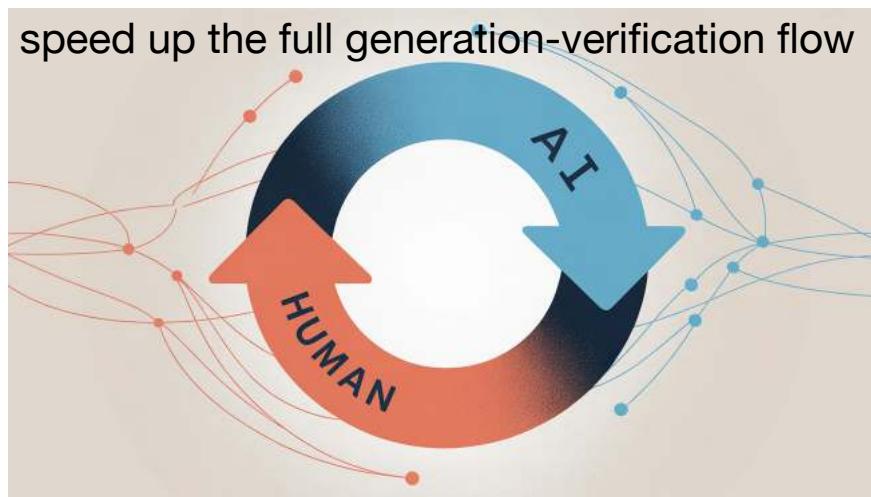
The screenshot shows a user interface for the Operator agent. On the left, there's a large blue rectangular area. To its right, a message bubble contains the text: "Find and book me the highest rated one-day tour of Rome on TripAdvisor." Below this, a list of actions is shown in a box: "I'll search for the highest-rated tour of historic Rome on TripAdvisor. Once I find a suitable option, I'll provide you with the details. Let's begin." Further down, another box lists the steps taken by the operator: "Worked for 2 minutes ^ Navigating to TripAdvisor website Selecting 'Things to Do' category Searching for historic Rome tours Closing pop-up, continuing tour search Searching for Historic Rome tours Exploring all historic Rome tour options Closing Colosseum tab, resuming tour search Closing tour pop-up, tab afterward Exploring options for top-rated tours Sorting results by tour ratings Exploring filters for top-rated tours Scrolling for sorting options, finding tours". At the bottom of the interface, there's a blue bar.





Partial autonomy LLM apps:

- Package context
- Orchestrate LLM calls
- Custom GUI
- Autonomy slider



Build for agents



THE IRON MAN SUIT

Augmentation



Agent



Thank you!