CSCE 585: Machine Learning Systems

Lecture 2: Machine Learning Systems in Production

Pooyan Jamshidi

UNIVERSITY OF
South Carolina

ML in research vs. production

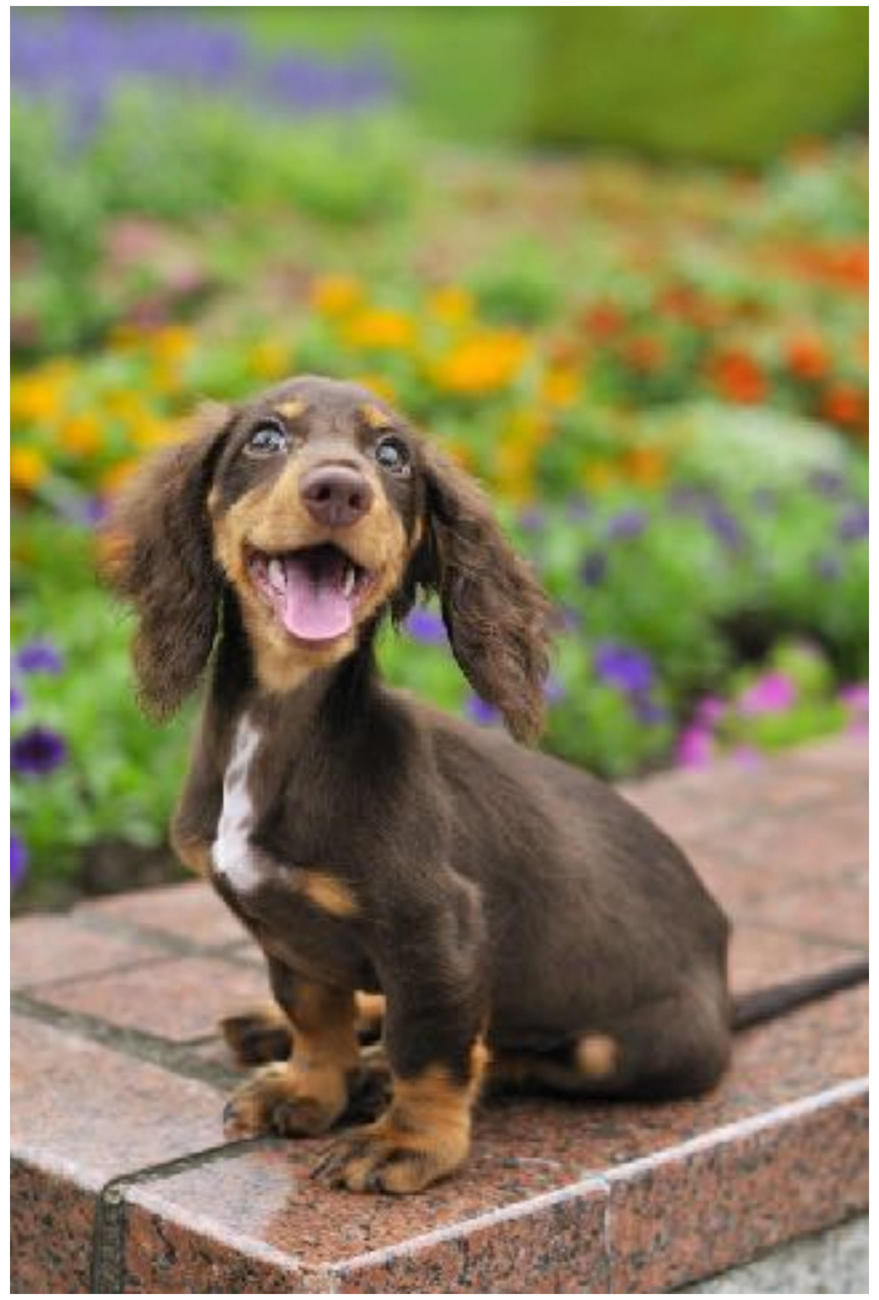
ML in research vs. in production

	Research	**Production**
Objectives	Model performance*	Different stakeholders have different objectives

“*” It’s actively being worked. See Utility is in the Eye of the User: A Critique of NLP Leaderboards (Ethayarajh and Jurafsky, EMNLP 2020)

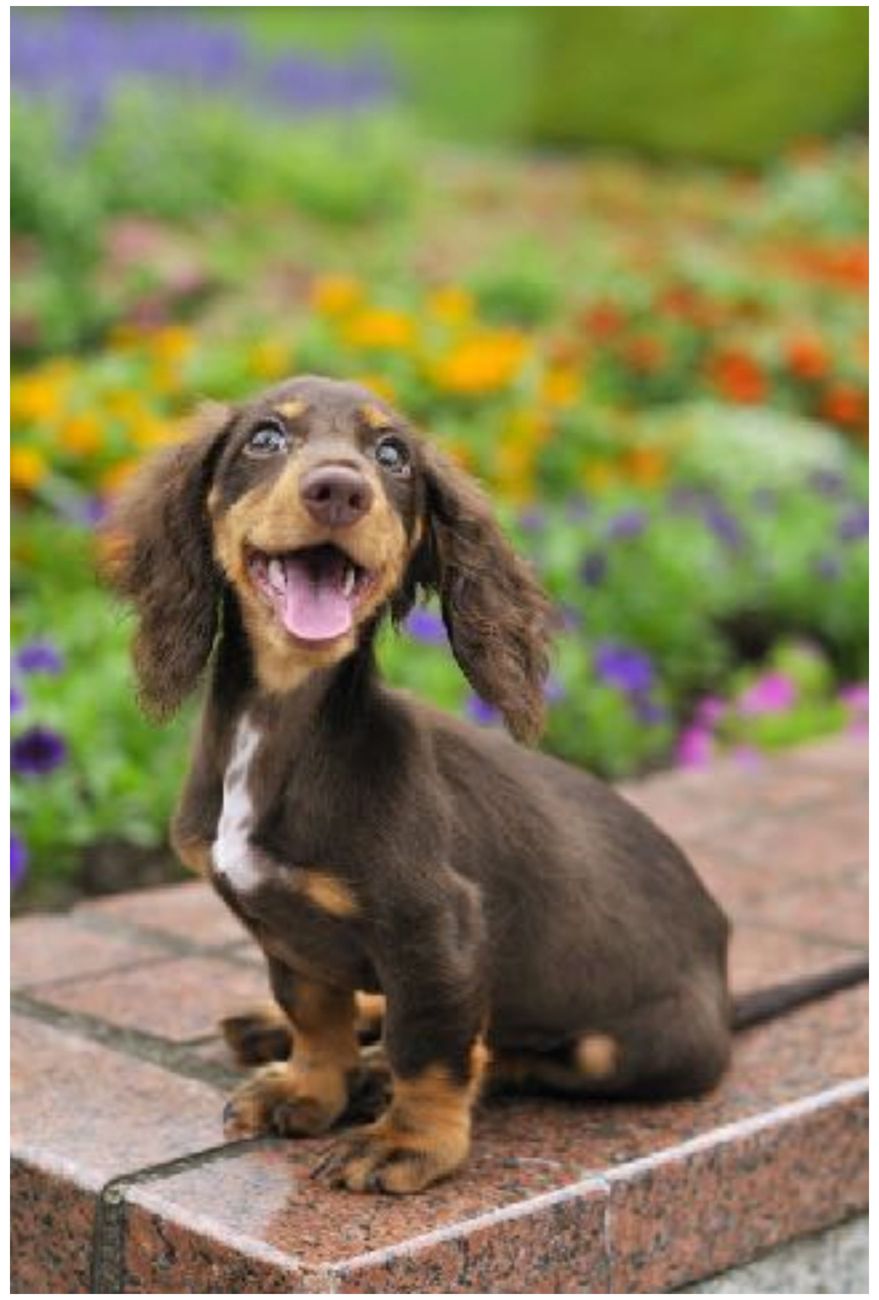
Stakeholder objectives

ML team
highest accuracy



Stakeholder objectives

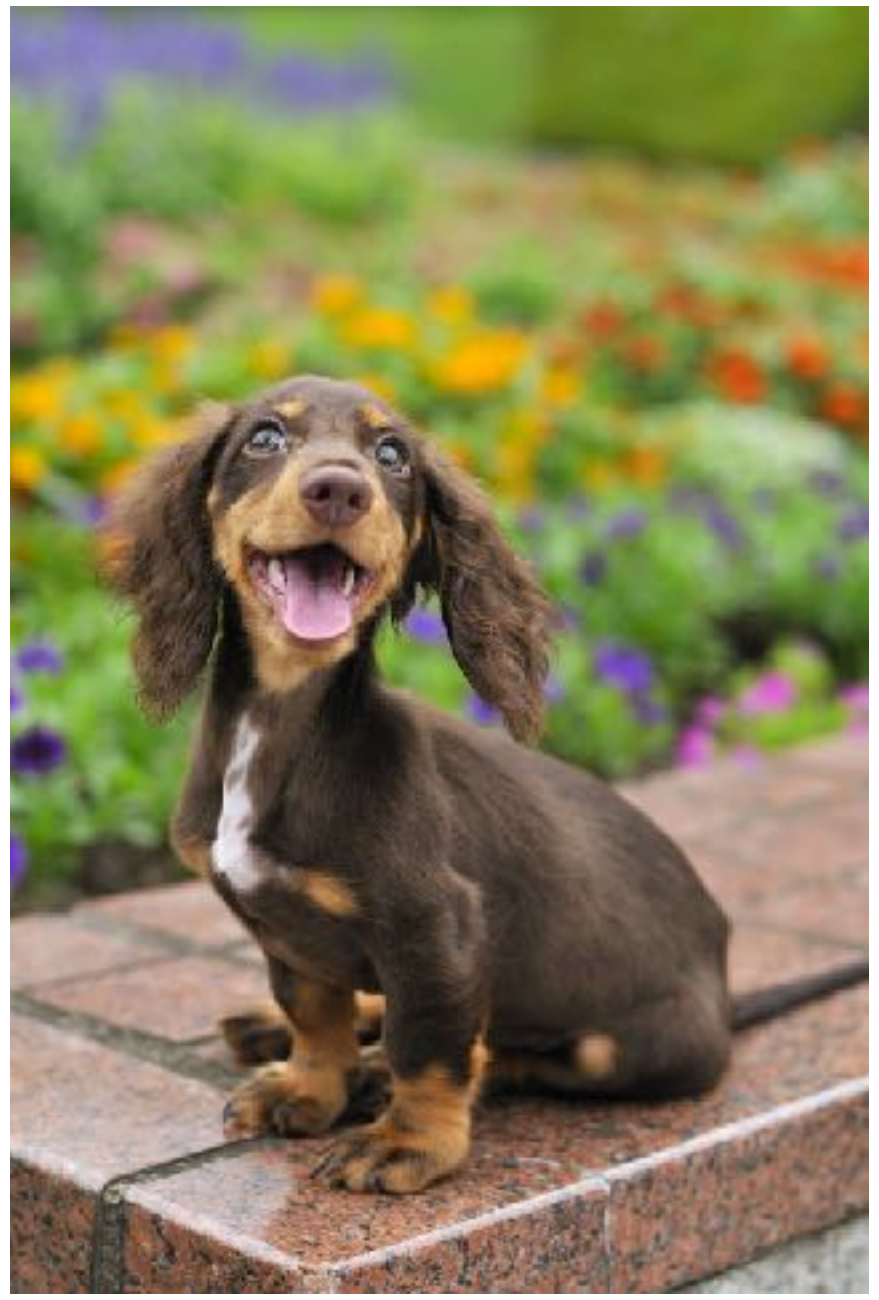
ML team
highest accuracy



Sales
sells more ads

Stakeholder objectives

ML team
highest accuracy

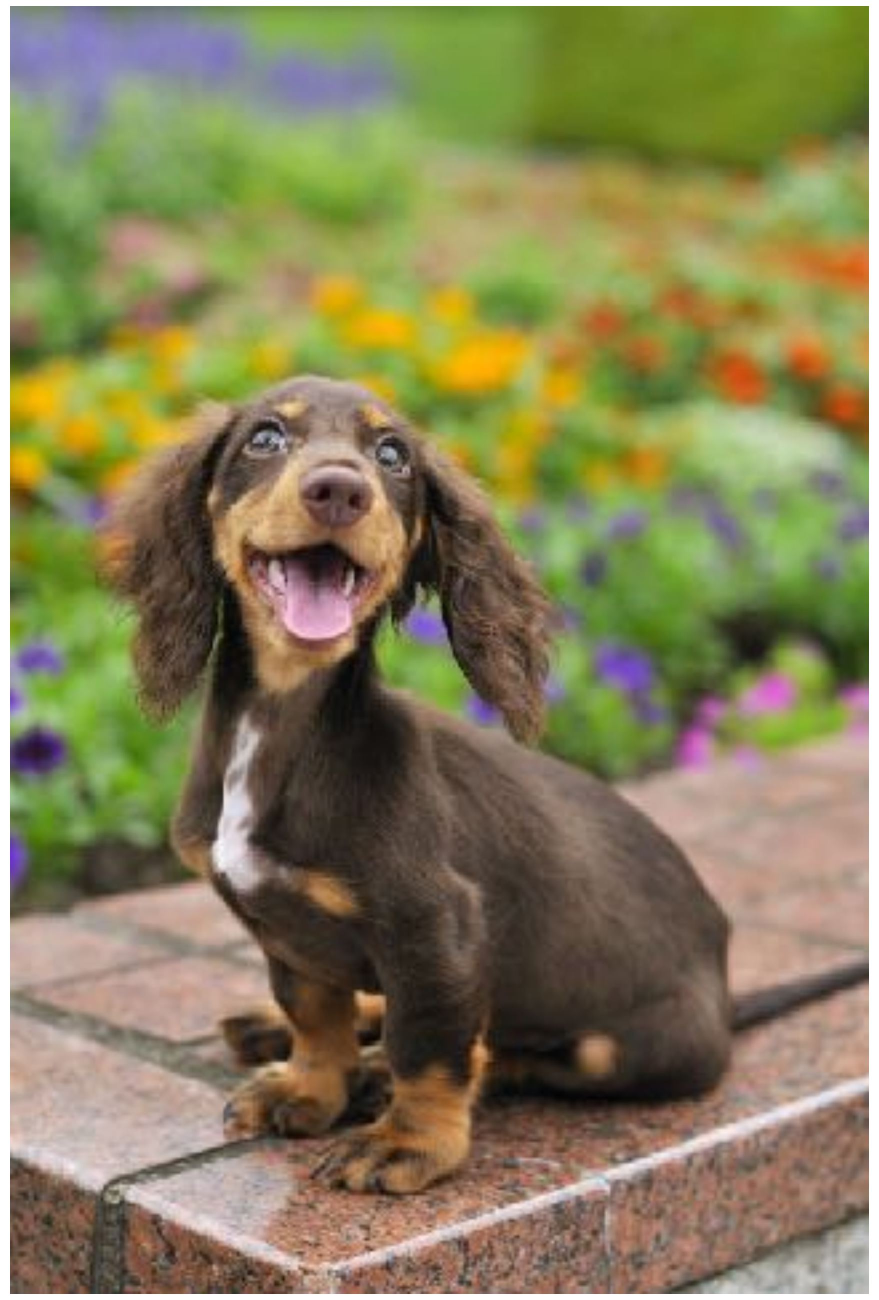


Sales
sells more ads

Product
fastest inference

Stakeholder objectives

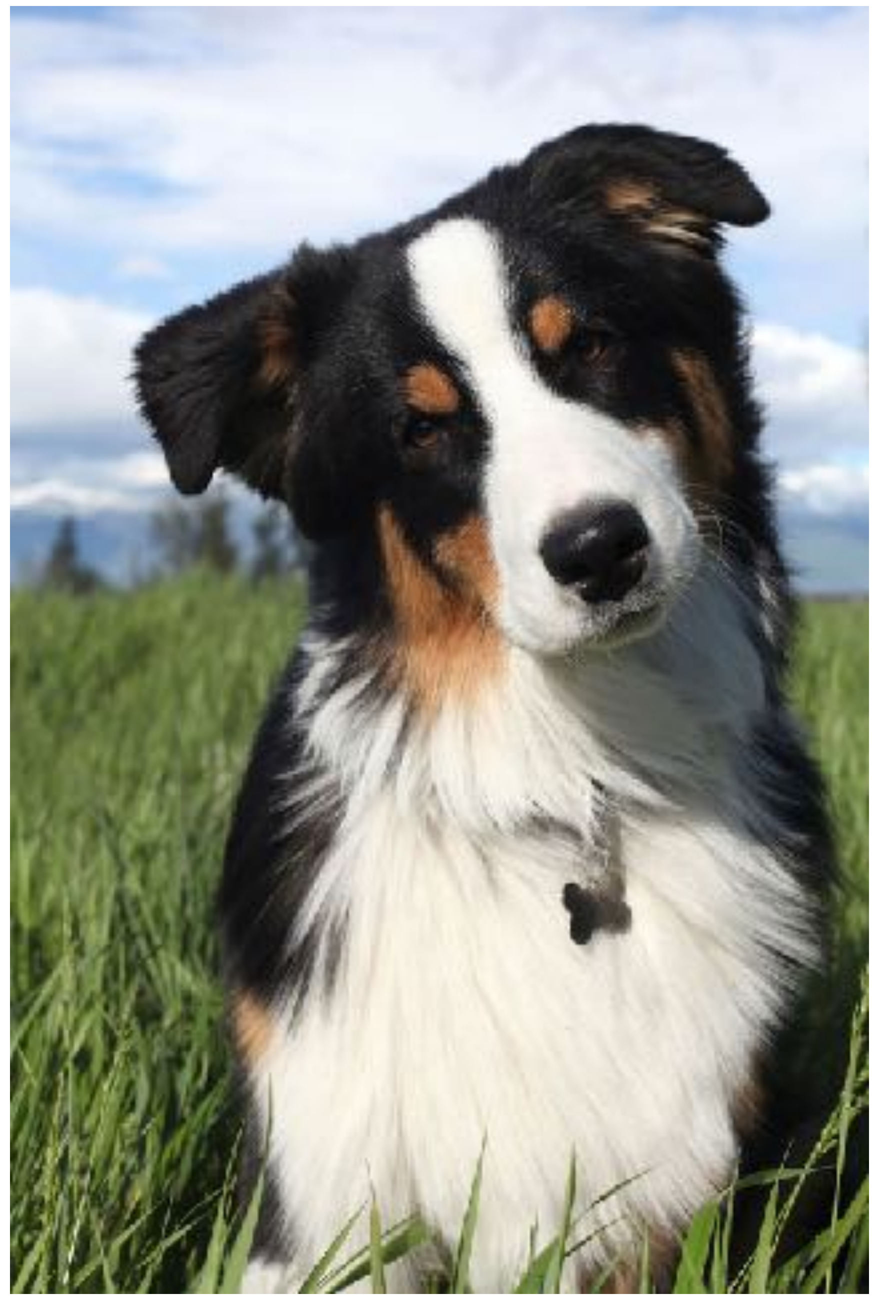
ML team
highest accuracy



Sales
sells more ads

Product
fastest inference

Manager
maximizes profit
= laying off ML teams



Computational priority

	Research	**Production**
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency

generating predictions

Latency matters

Latency 100 -> 400 ms reduces searches 0.2% - 0.6% (2009)

30% increase in latency costs 0.5% conversion rate (2019)

- **Latency: time to move a leaf**
- **Throughput: how many leaves in 1 sec**

- Real-time: low latency = high throughput
- Batched: high latency, high throughput

ML in research vs. in production

	Research	**Production**
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting

Data

Research	Production
● Clean ● Static ● Mostly historical data	● Messy ● Constantly shifting ● Historical + streaming data ● Biased, and you don’t know how biased ● Privacy + regulatory concerns

THE COGNITIVE CODER

By Armand Ruiz, Contributor, InfoWorld | SEP 26, 2017 7:22 AM PDT

The 80/20 data science dilemma

Most data scientists spend only 20 percent of their time on actual data analysis and 80 percent of their time finding, cleaning, and reorganizing huge amounts of data, which is an inefficient data strategy

ML in research vs. in production

	Research	Production
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting
Fairness	Good to have (sadly)	Important

Fairness

The Berkeley study found that both face-to-face and online lenders rejected a total of 1.3 million creditworthy black and Latino applicants between 2008 and 2015. Researchers said they believe the applicants "would have been accepted had the applicant not been in these minority groups." That's because when they used the income and credit scores of the rejected applications but deleted the race identifiers, the mortgage application was accepted.

ML in research vs. in production

	Research	**Production**
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting
Fairness	Good to have (sadly)	Important
Interpretability*	Good to have	Important

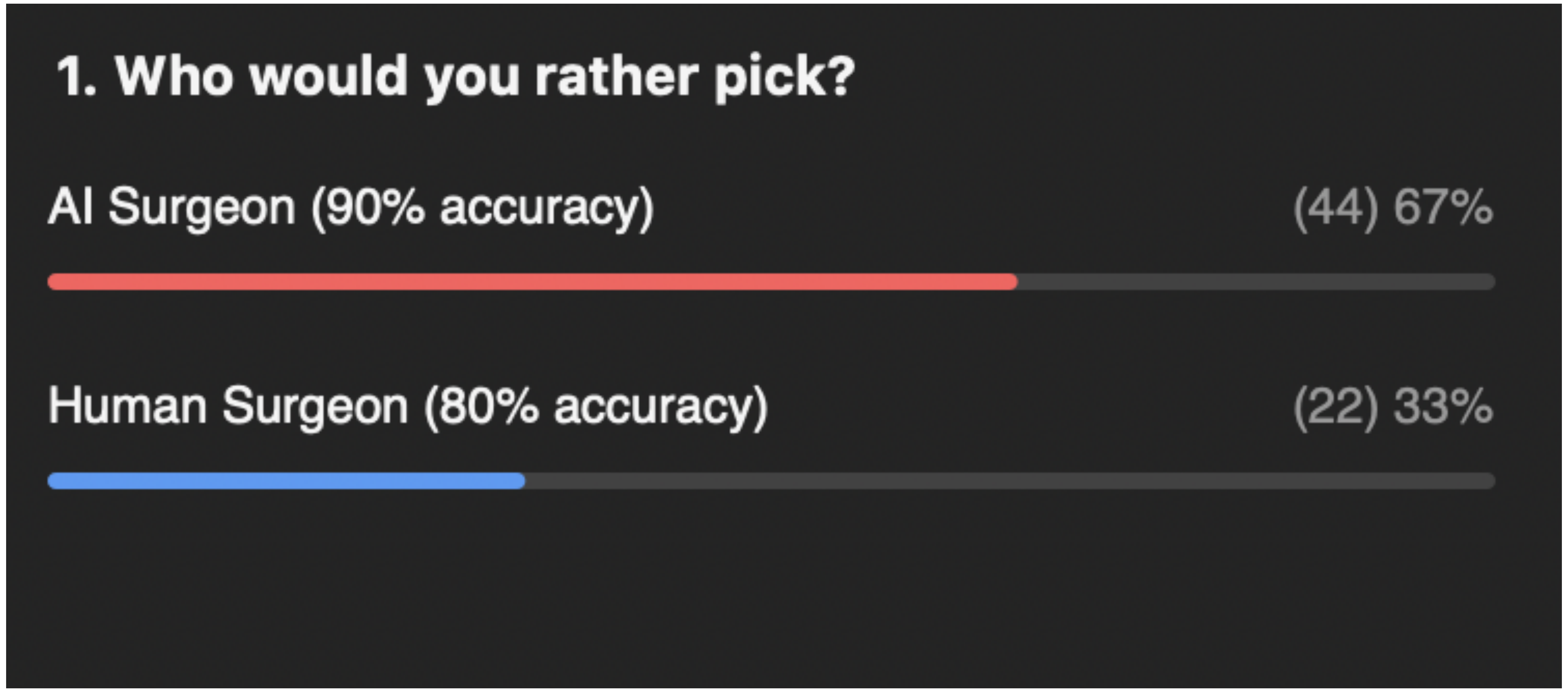
Interpretability

Geoffrey Hinton
@geoffreyhinton

Suppose you have cancer and you have to choose between a black box AI surgeon that cannot explain how it works but has a 90% cure rate and a human surgeon with an 80% cure rate. Do you want the AI surgeon to be illegal?

12:37 PM · Feb 20, 2020 · Twitter Web App

1.1K Retweets **5.2K** Likes



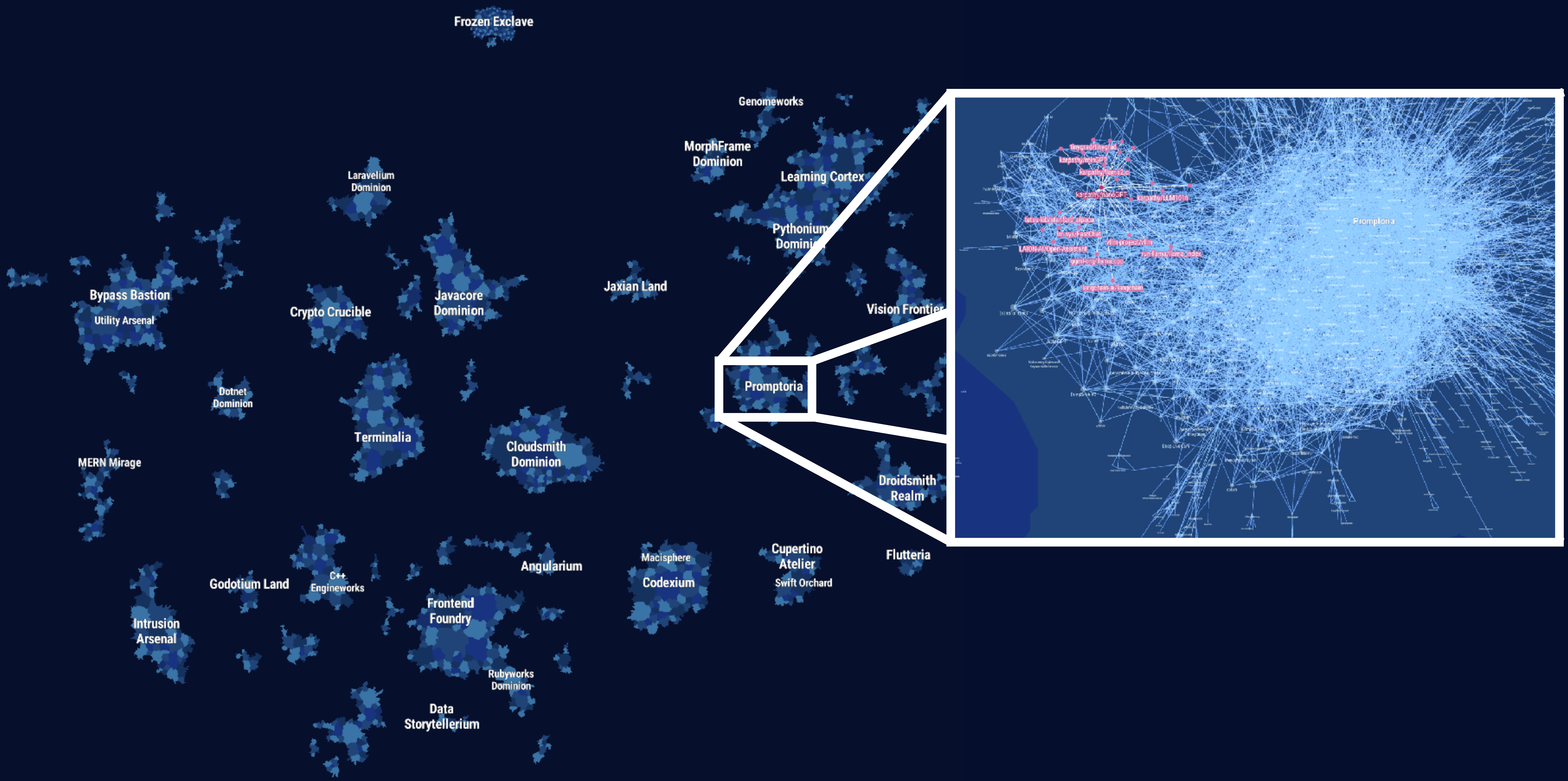
Result from the Zoom poll

ML in research vs. in production

	Research	**Production**
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting
Fairness	Good to have (sadly)	Important
Interpretability	Good to have	Important

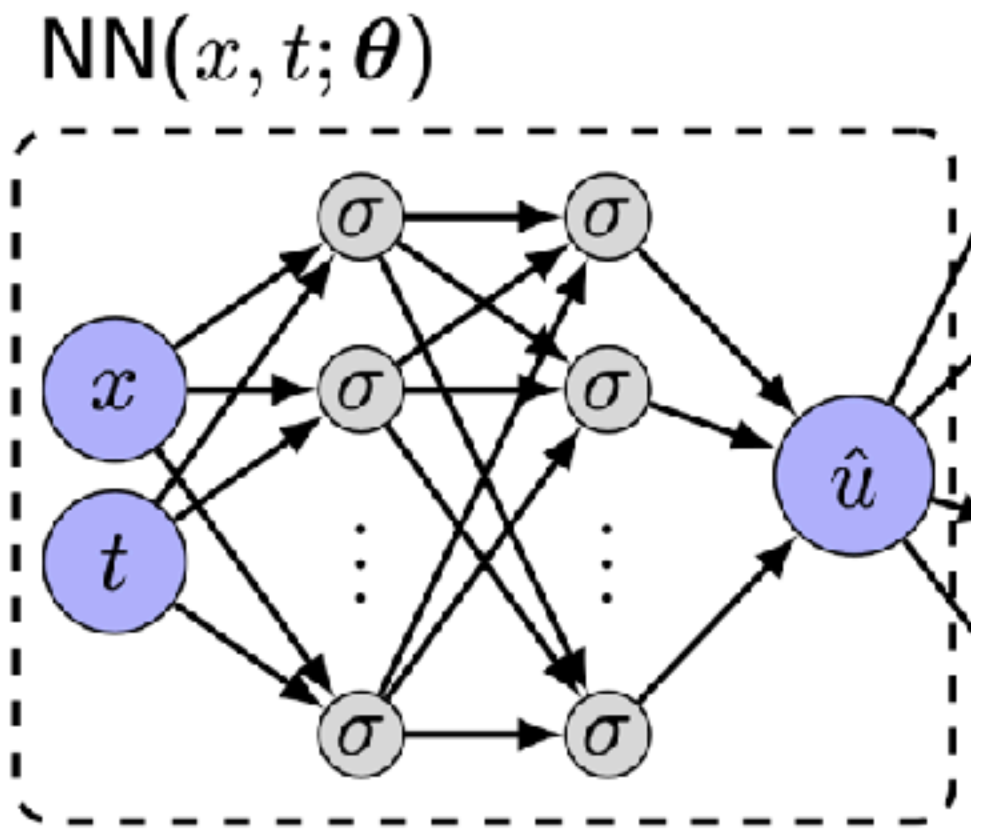
Software 1.0 ->
Software 2.0 ->
Software 3.0!

"Map of GitHub"



Software 1.0 vs Software 2.0

- Written in code (C++, ...)
- Requires domain expertise
  1. Decompose the problem
  2. Design algorithms
  3. Compose into a system

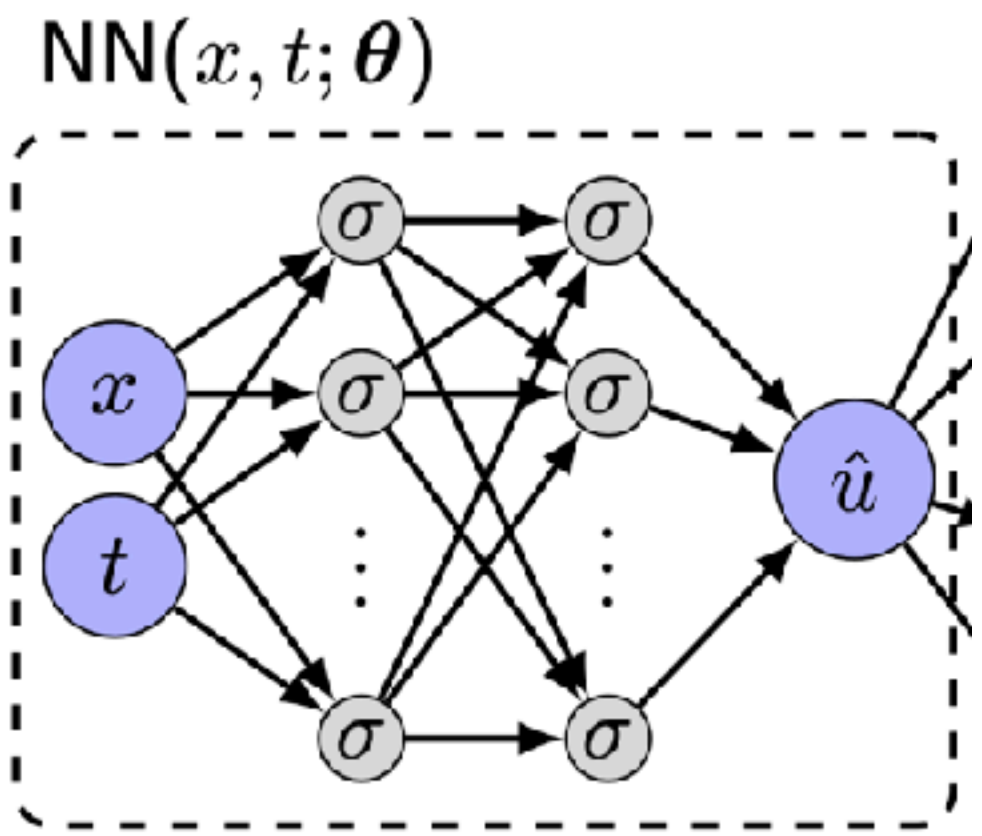


- Written in terms of a neural network model with
  - A model architecture
  - Weights that are determined using optimization

https://medium.com/@karpathy/software-2-0-a64152b37c35

Software 1.0 vs Software 2.0

- **Input**: Algorithms in code
- **Compiled to**: Machine instructions



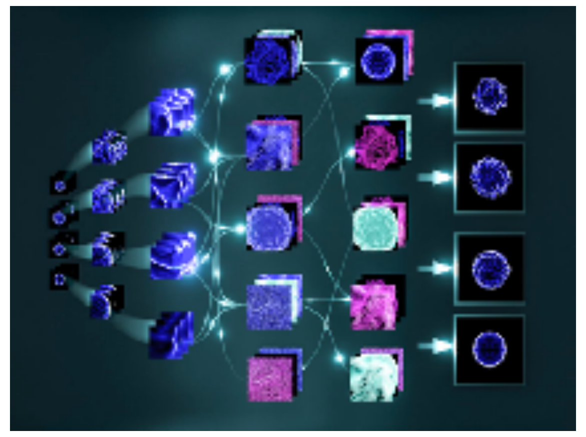
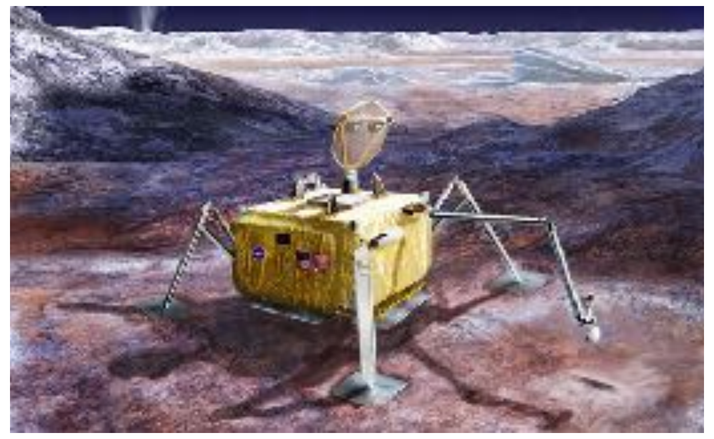
Input: Training data

Compiled to: Learned parameters

https://medium.com/@karpathy/software-2-0-a64152b37c35

Software 1.0 vs Software 2.0




AI for Social Good

- **Easier to build and deploy**
  - Build products faster
  - Predictable runtimes and memory use: easier qualification

- A **wide range of applications** from self-driving cars, to game, healthcare, robotics, space, and social good.
- **1000x Productivity**: Google shrinks language translation code from 500k LoC to 500

https://jack-clark.net/2017/10/09/import-ai-63-google-shrinks-language-translation-code-from-500000-to-500-lines-with-ai-only-25-of-surveyed-people-believe-automationbetter-jobs/

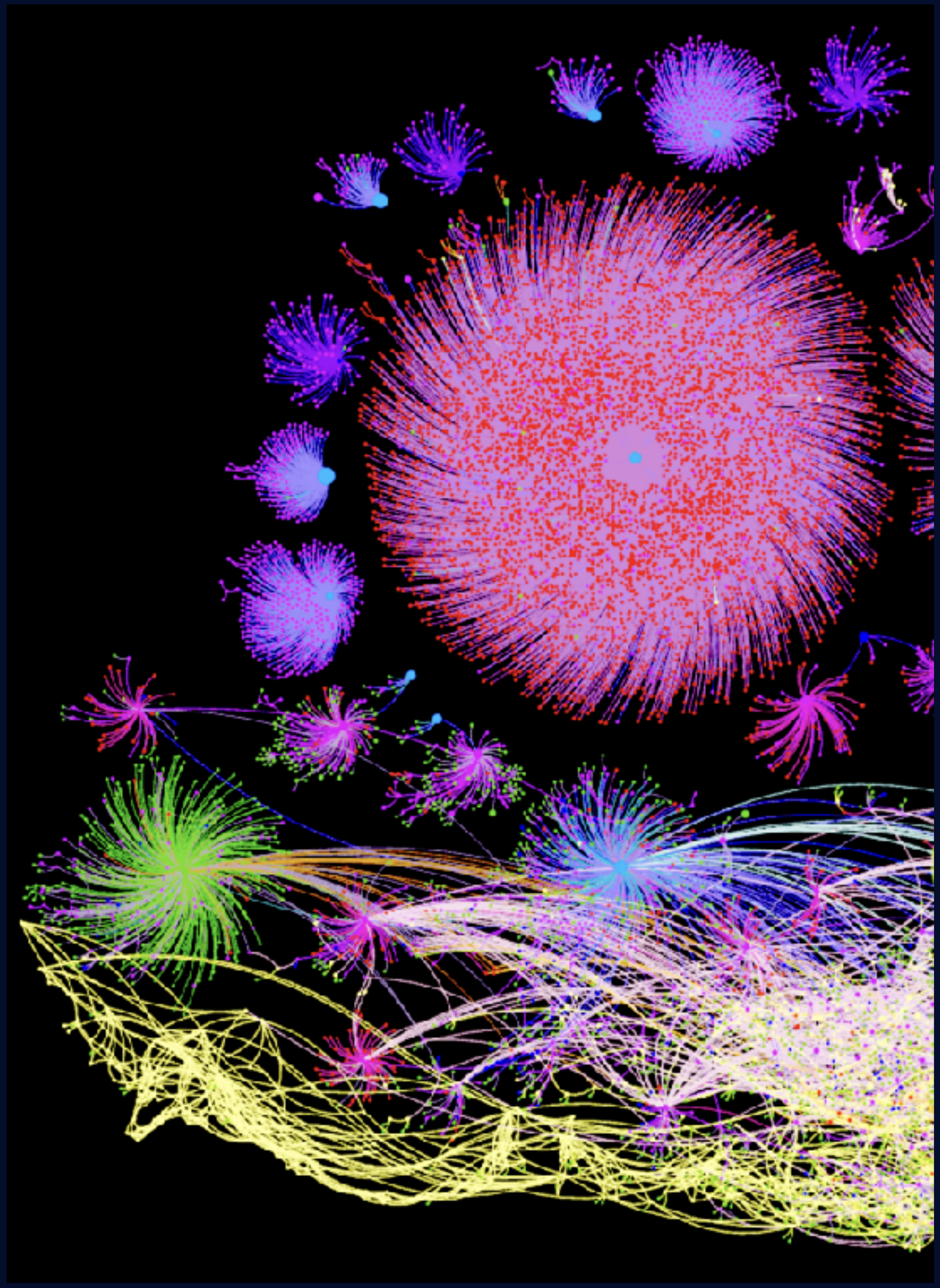
https://ai.google/social-good/

"Map of GitHub" (Software 1.0)

computer code

HuggingFace Model Atlas (Software 2.0)

neural network weights



Software is changing again

Software 1.0

computer code

programs

computer

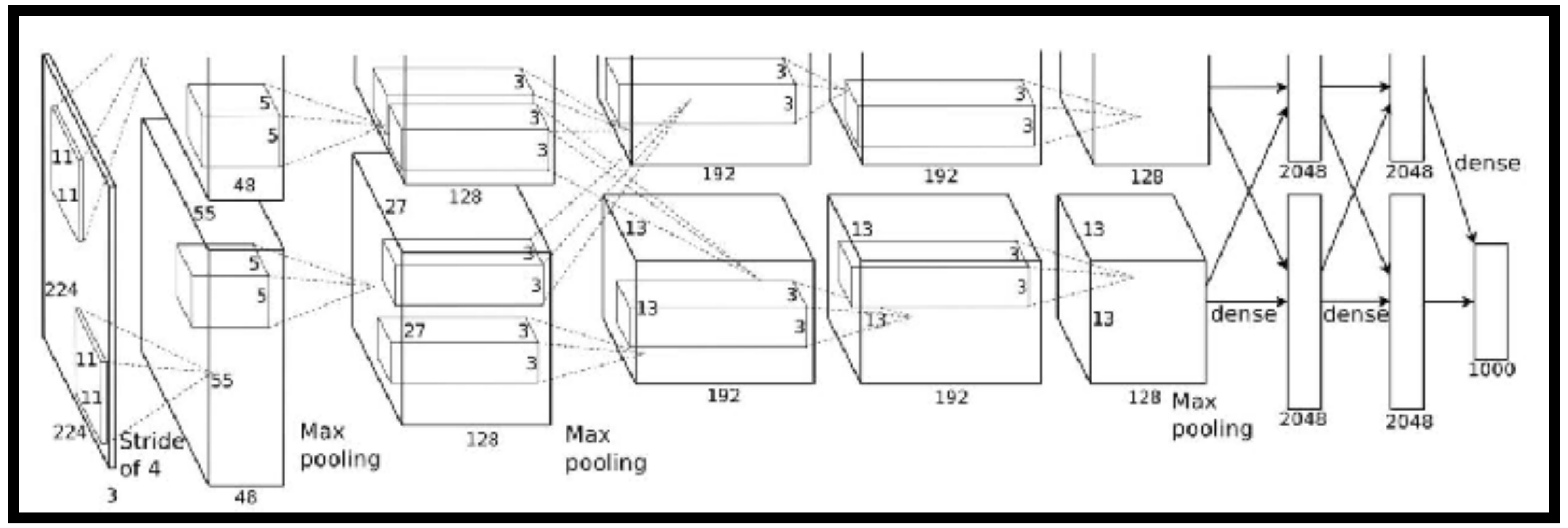
became programmable in **~1940s**

Software 2.0

weights

programs

neural net



fixed function neural net

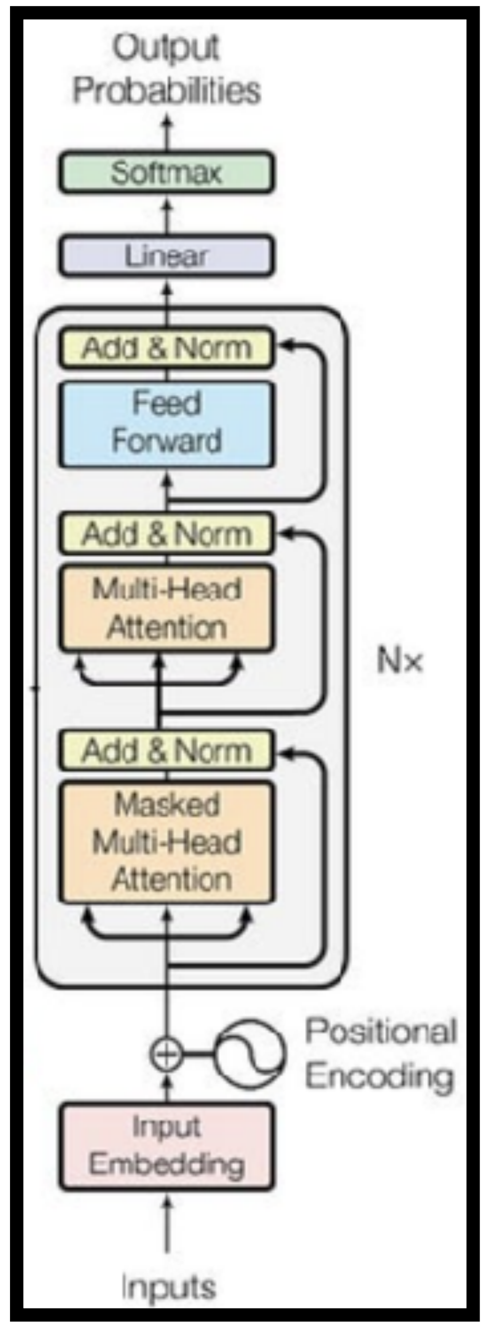
e.g. AlexNet: for image recognition (**~2012**)

Software 3.0

Prompts

programs

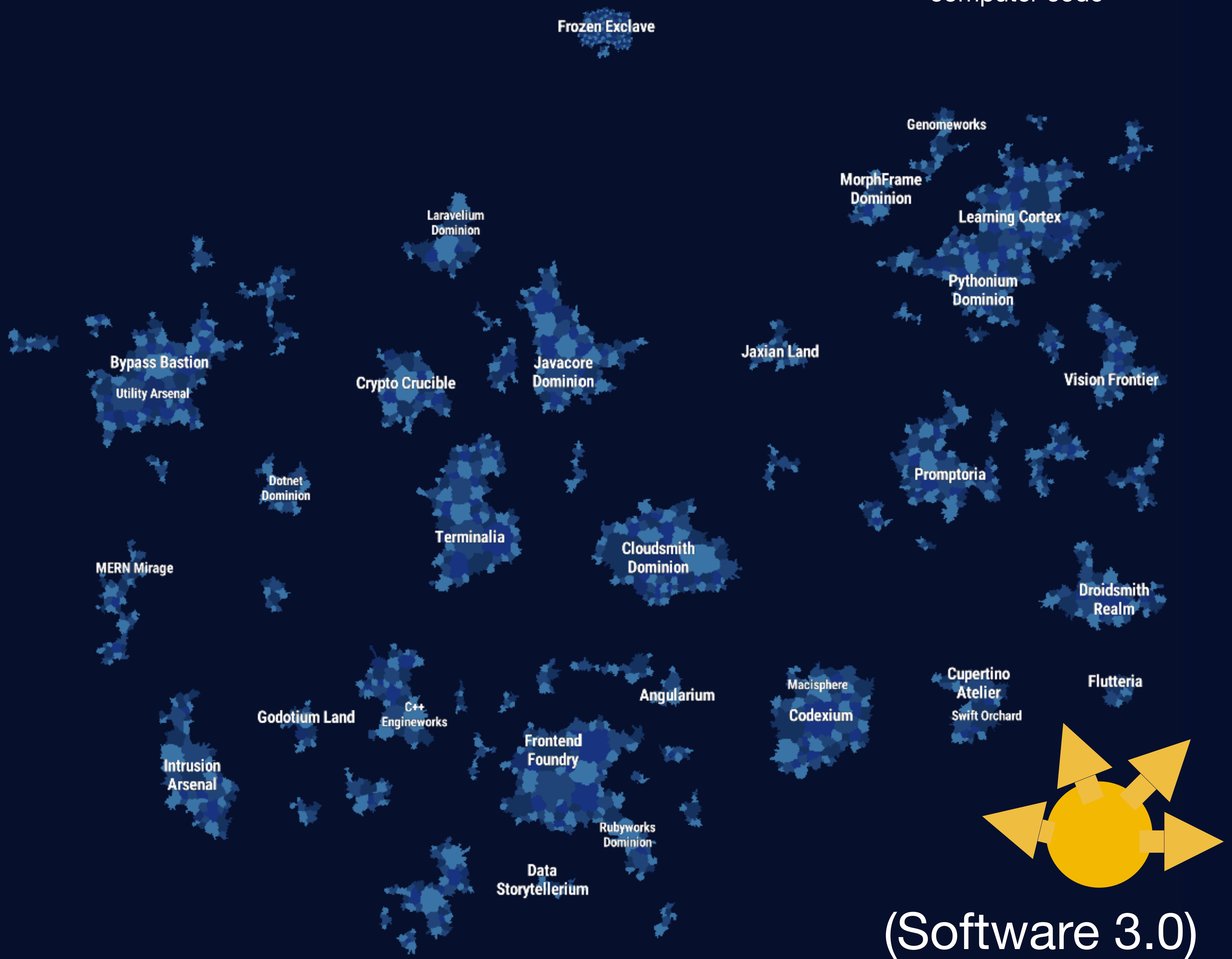
LLM



LLM = Programable Neural Netowrk (**~2019**)

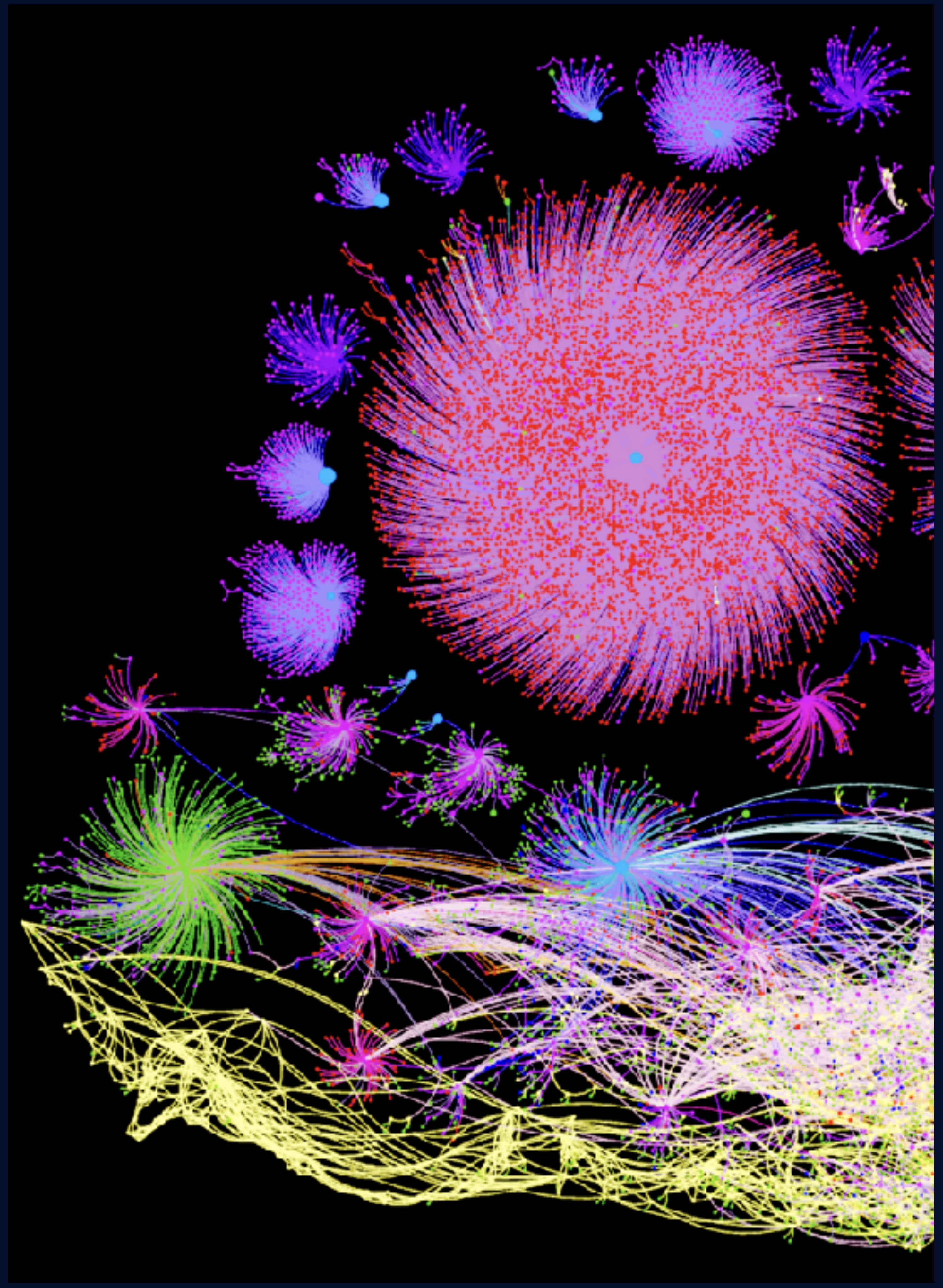
"Map of GitHub" (Software 1.0)

computer code



HuggingFace Model Atlas (Software 2.0)

neural network weights



(Software 3.0)

LLM prompts, in English

Example: Sentiment Classification

Software 1.0

python Copy

```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

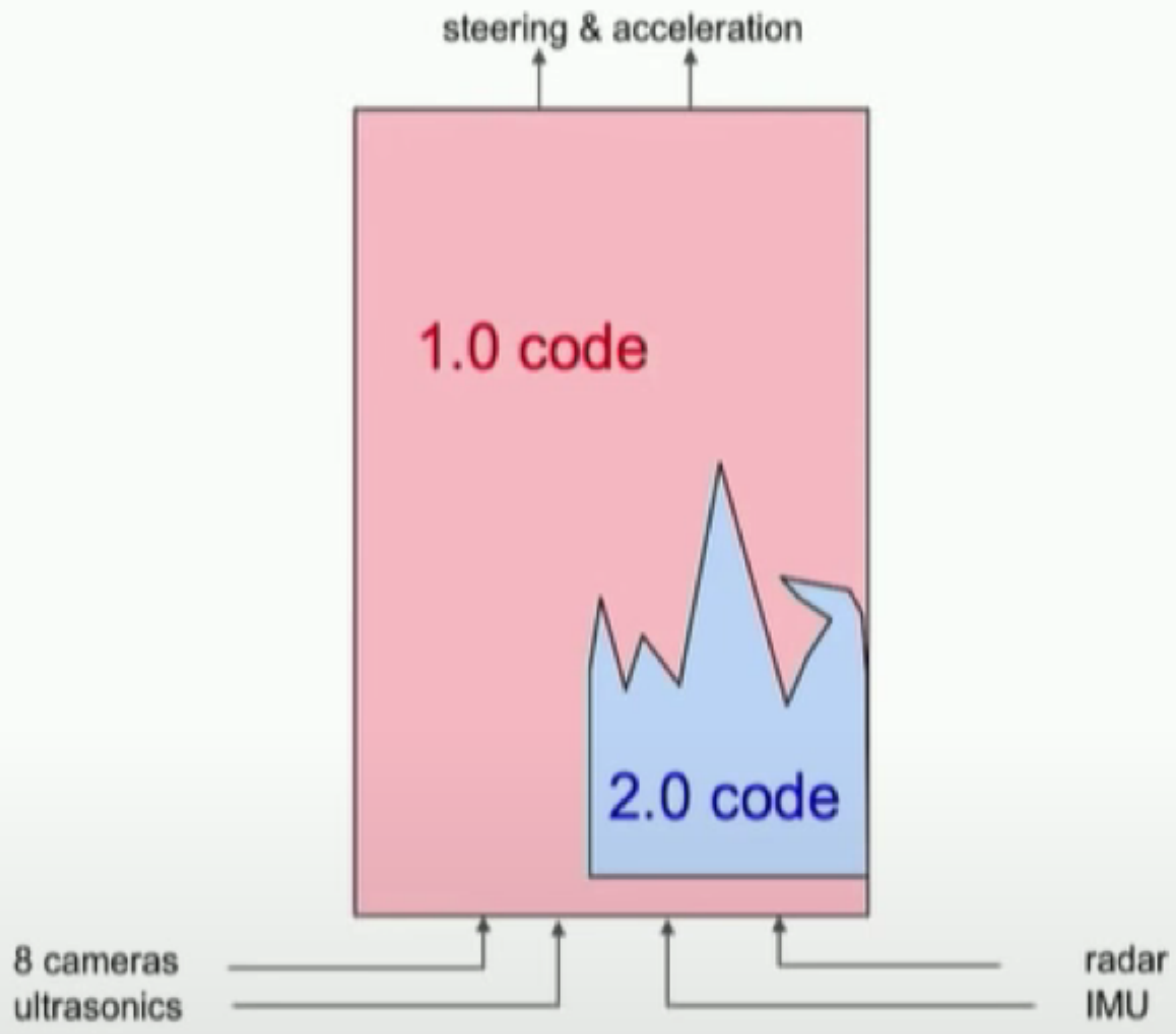
Example 3

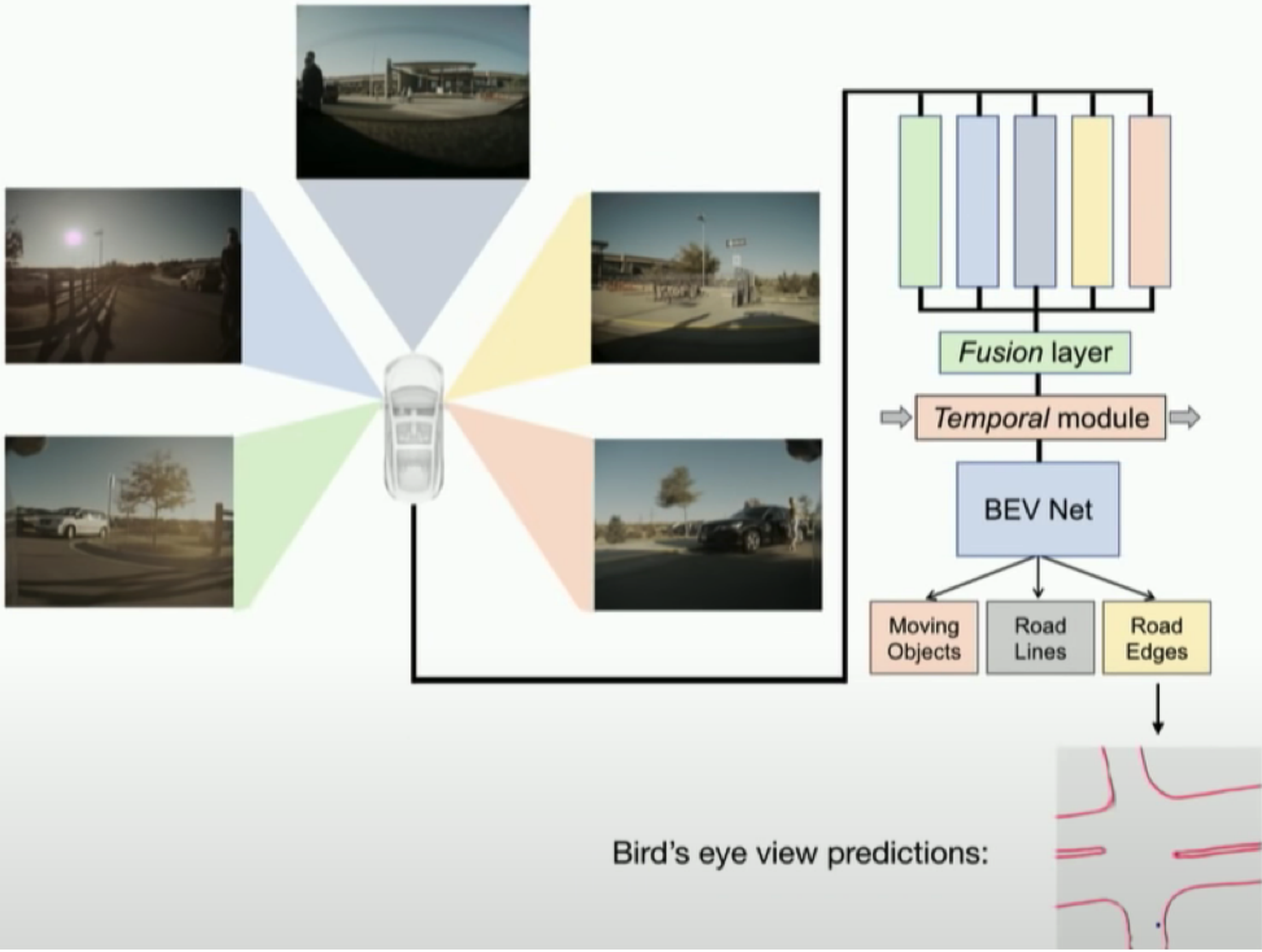
<REVIEW>An energetic soundtrack and solid visuals almost save it, but the story drags and the jokes fall flat.</REVIEW>

NEGATIVE

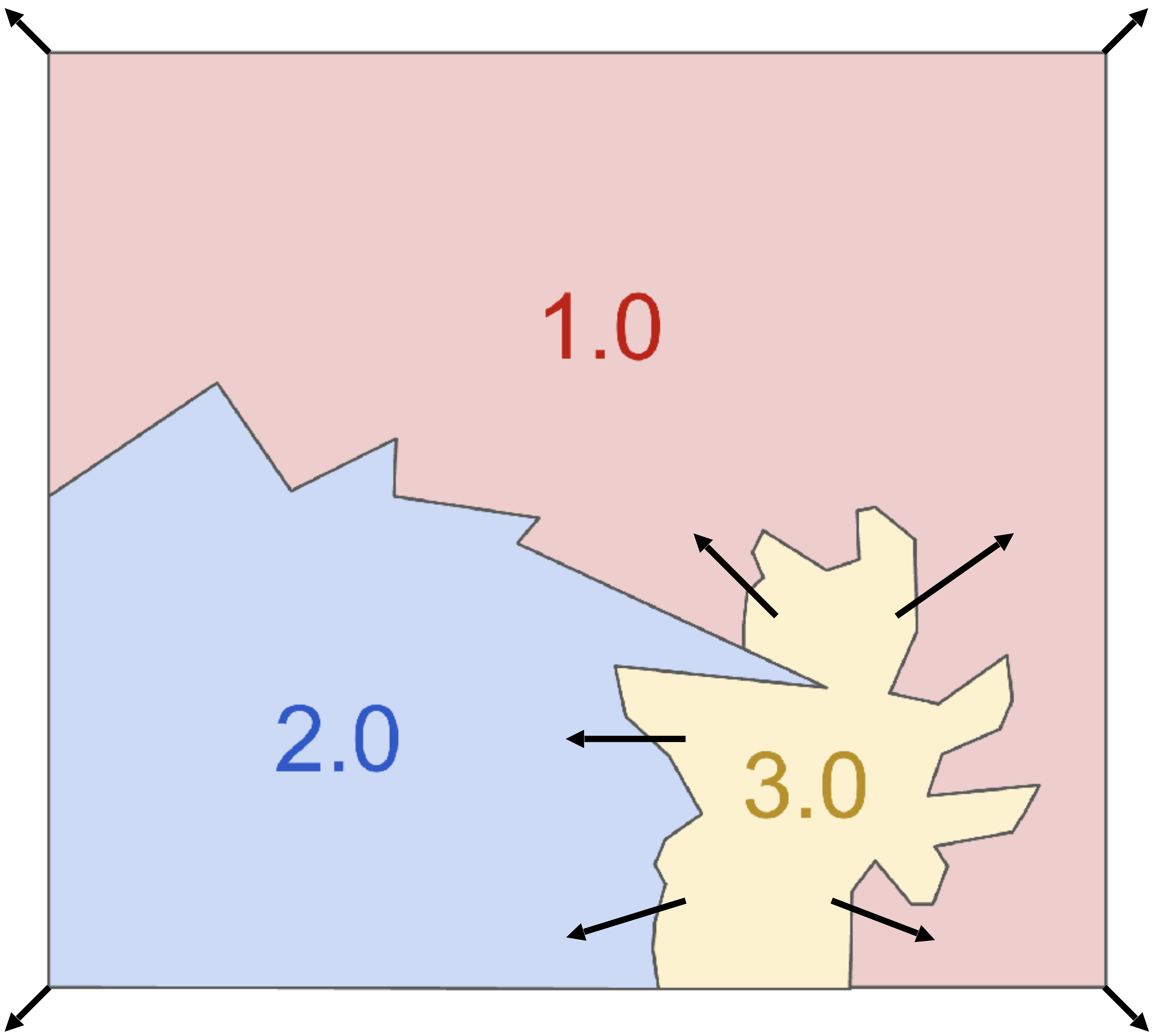
Now classify the next review.

Software is eating the world
Software 2.0 eating Software 1.0





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

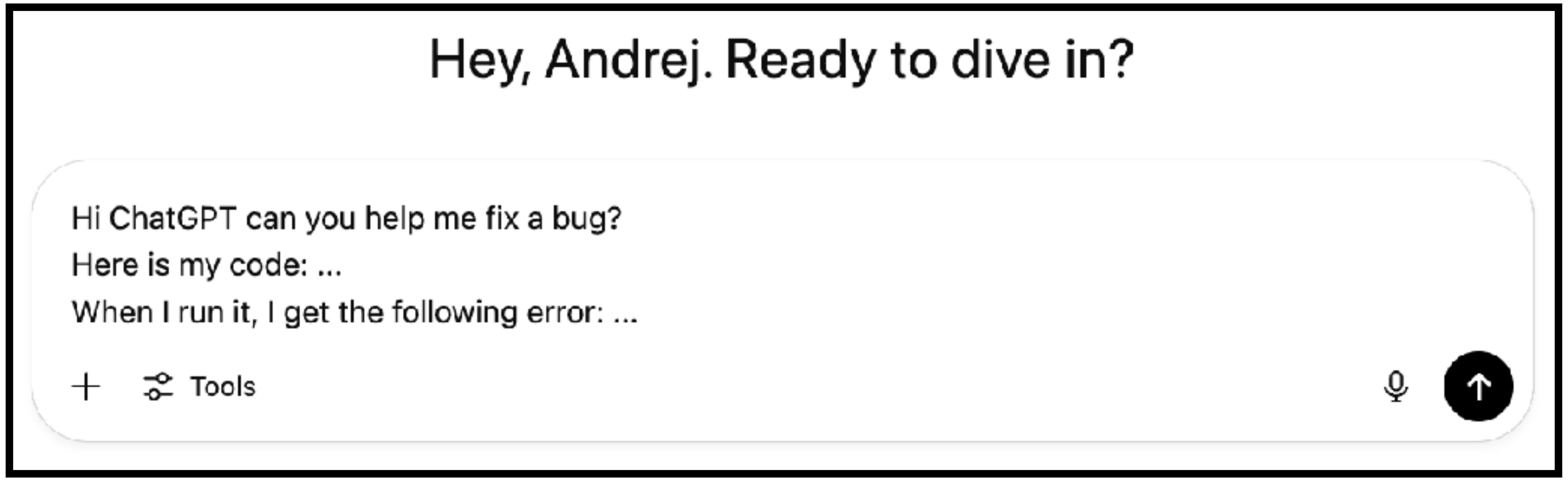


Opportunities

Partial autonomy apps ⚙️

"Copilot" / "Cursor for X"

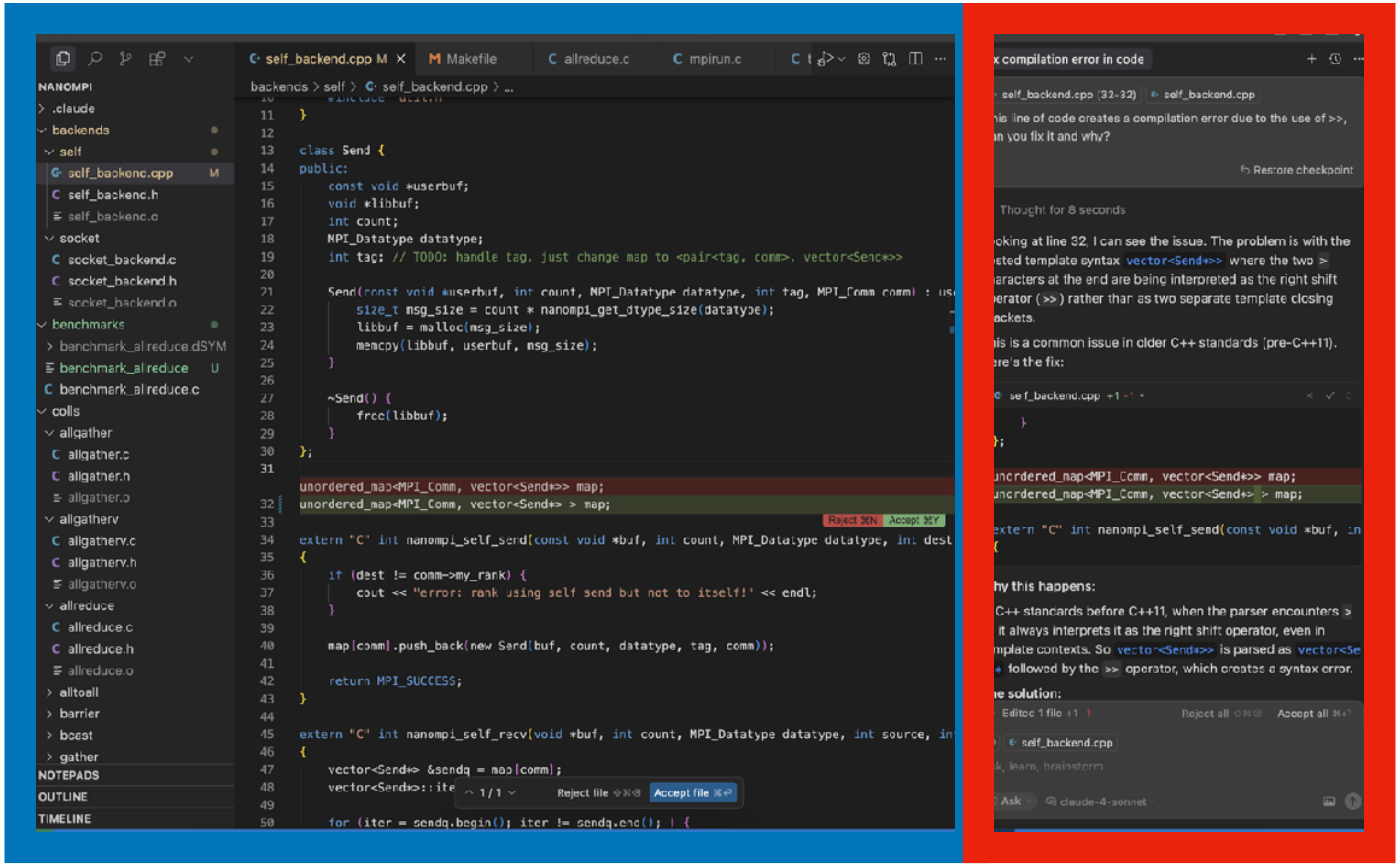
Example: you could go to an LLM to chat about code...



Example: Anatomy of Cursor

Traditional interface

LLM integration



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 — Meta's AI shortfall prompts Zuckerberg's $15B investment i...
reuters — Meta to pay nearly $15 billion for Scale ...
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.

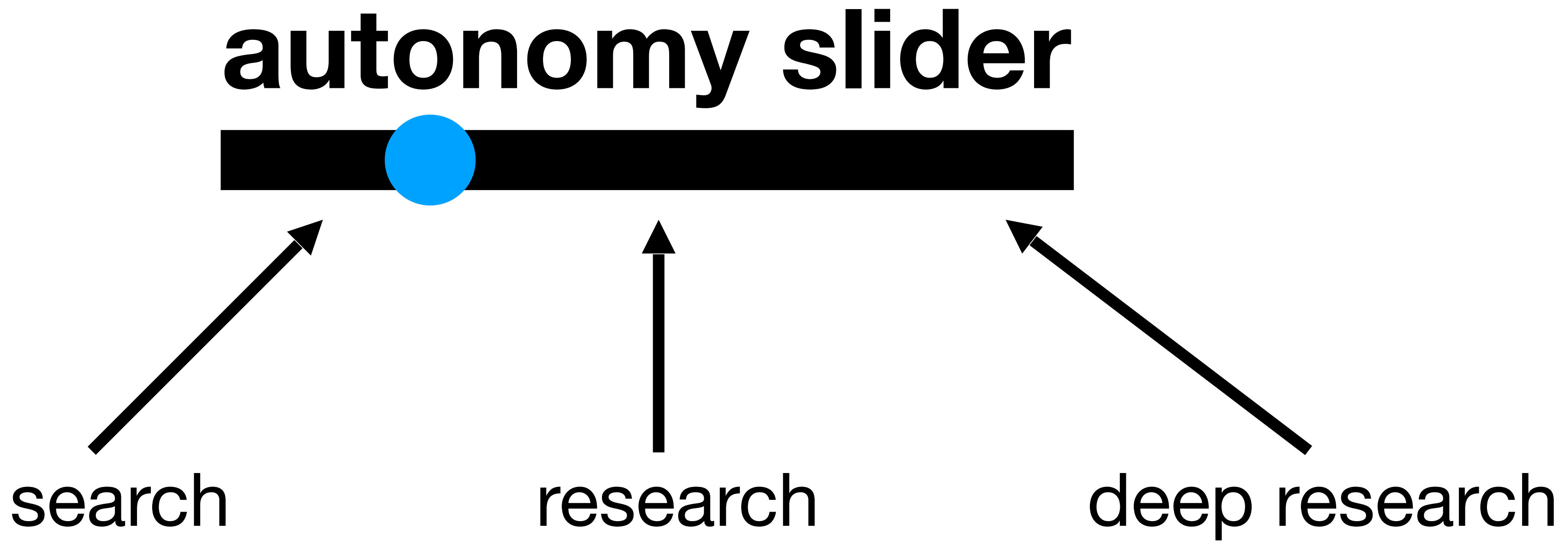
What Is Scale AI and Why Does Meta Want It?

- **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.

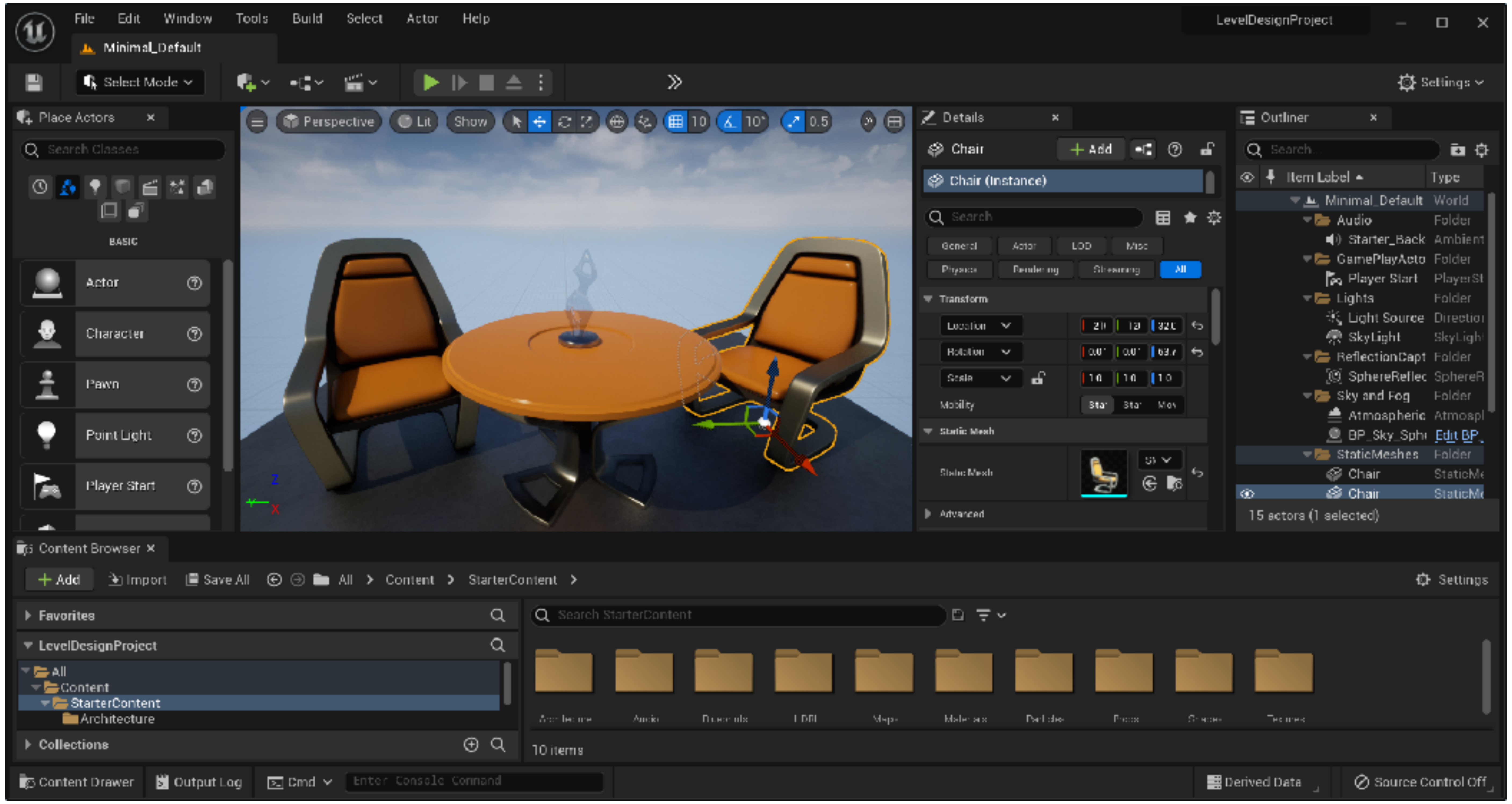
1. Package information into a context window
2. Orchestrate multiple LLM models
3. Application-specific GUI for Input/Output UIUX



(+suggested followup questions)

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

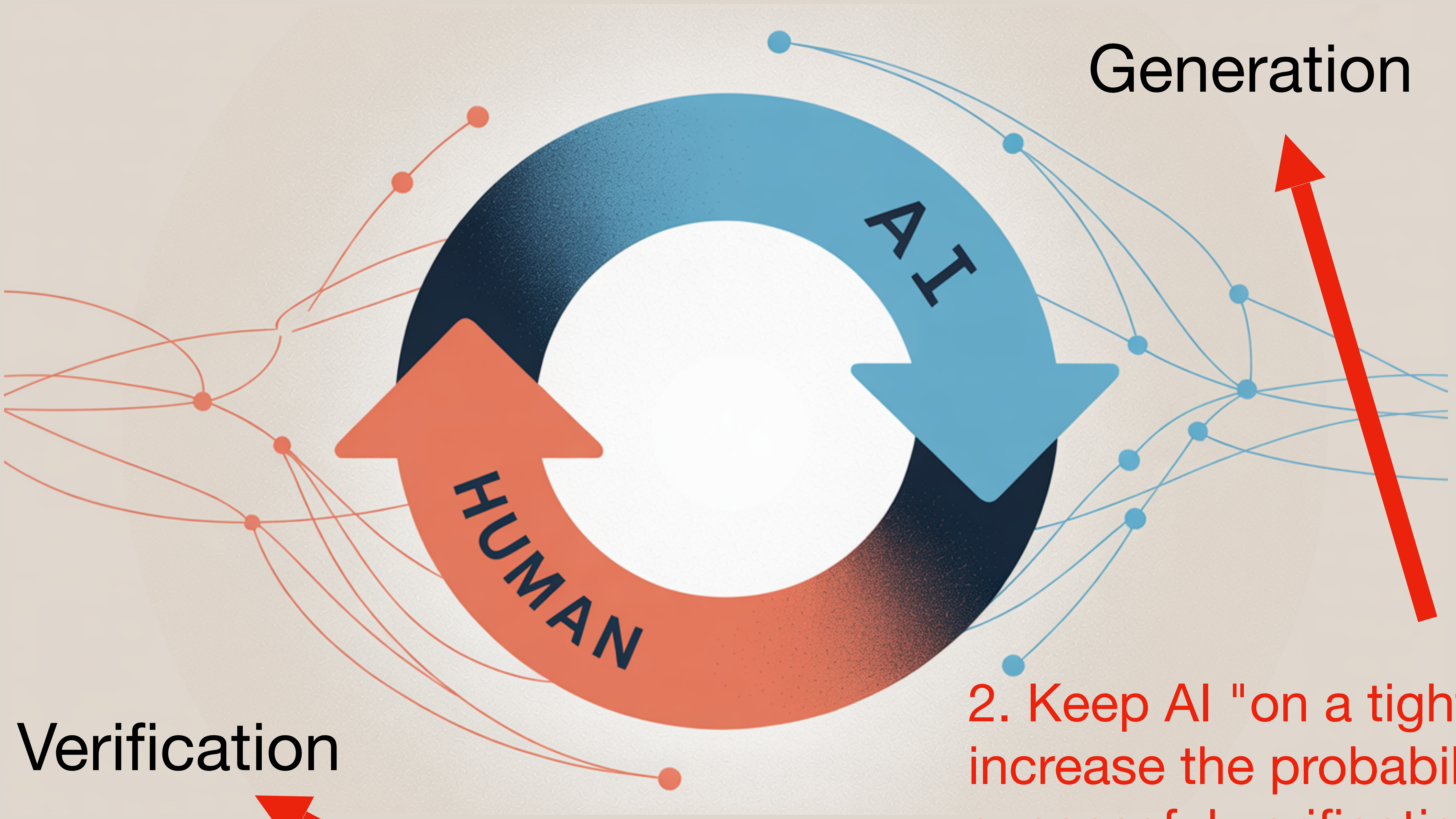
Adobe photoshop



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



Generation

Verification

1. Make this EASY, FAST to win.

2. Keep AI "on a tight leash" to increase the probability of successful verification

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"

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.

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).
```

Atharva Raykar
Read more by Atharva here

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

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://answerdotai.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.githubusercontent.com/path/starlette-sml.md): A subset of the
Starlette documentation useful for FastHTML development.
```

Docs for people

Copy page

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.

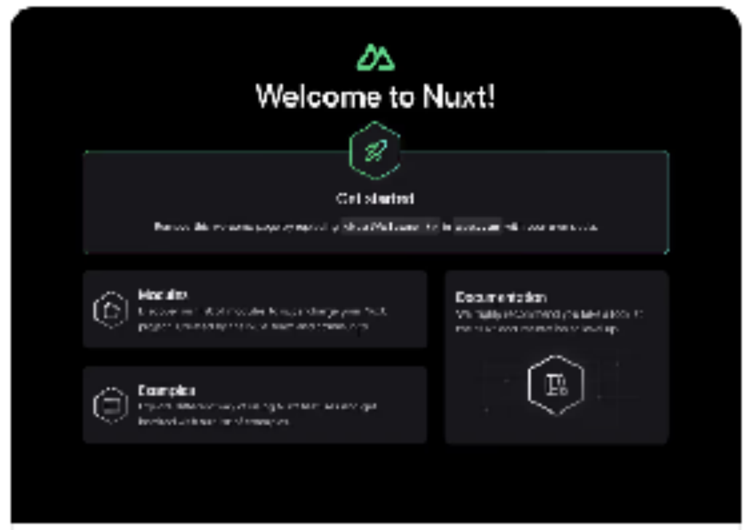
Get started in minutes

Deploy a Template

View All Templates

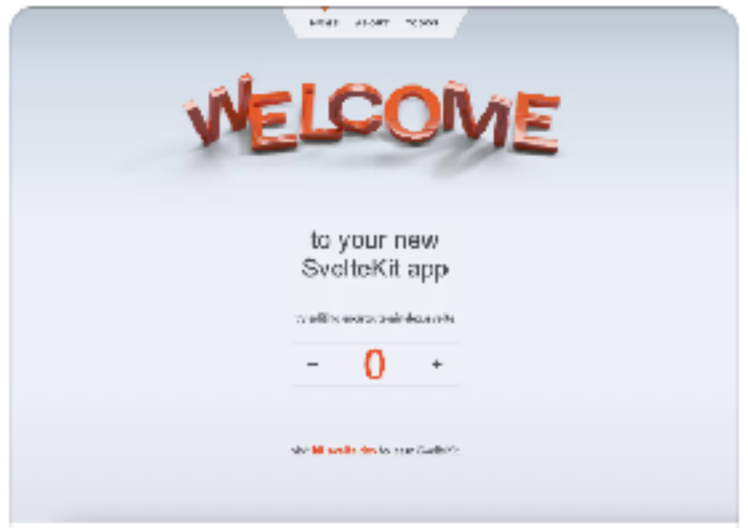
Next.js Boilerplate

Get started with Next.js and React in seconds.



Nuxt.js 3 Boilerplate

A Nuxt.js 3 app, bootstrapped with create-nuxt-app.



SvelteKit Boilerplate

A SvelteKit app including nested routes, layouts, and page endpoints.

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.com/docs/llms.txt

```
# Vercel Documentation

- [Getting Started](https://vercel.com/docs/getting-sta
  - [Projects and Deployments](https://vercel.com/docs/
  - [Use a Template](https://vercel.com/docs/getting-st
  - [Import Existing Project](https://vercel.com/docs/g
  - [Add a Domain](https://vercel.com/docs/getting-star
  - [Buy a Domain](https://vercel.com/docs/getting-star
  - [Transfer an Existing Domain](https://vercel.com/do
  - [Collaborate](https://vercel.com/docs/getting-start
  - [Next Steps](https://vercel.com/docs/getting-starte
- [Supported Frameworks](https://vercel.com/docs/framew
  - [Next.js](https://vercel.com/docs/frameworks/nextjs
  - [SvelteKit](https://vercel.com/docs/frameworks/svel
  - [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/r
```

Home / Get started

Build on Stripe with LLMs

Copy page

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.

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

```
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.

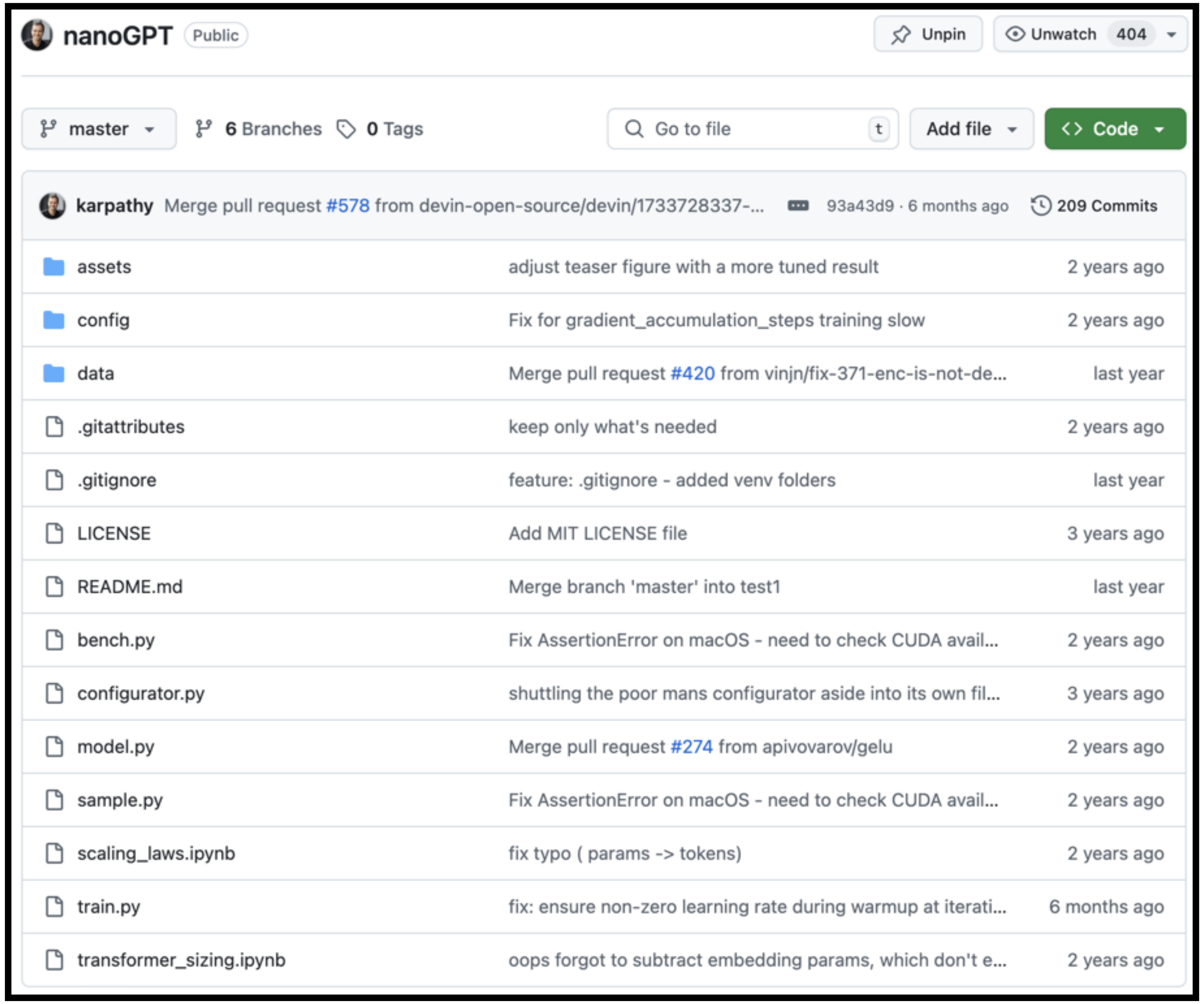
Alternatively, add the following to your `~/.cursor/mcp.json` file.

```
{
  "mcpServers": {
    "stripe": {
      "command": "npx",
      "args": ["-y", "@stripe/mcp", "--tools=all"],
      "env": {
        "STRIPE_SECRET_KEY": "sk_test_BQokikJOvBiI2HlWgH4olfQ2"
      }
    }
  }
}
```

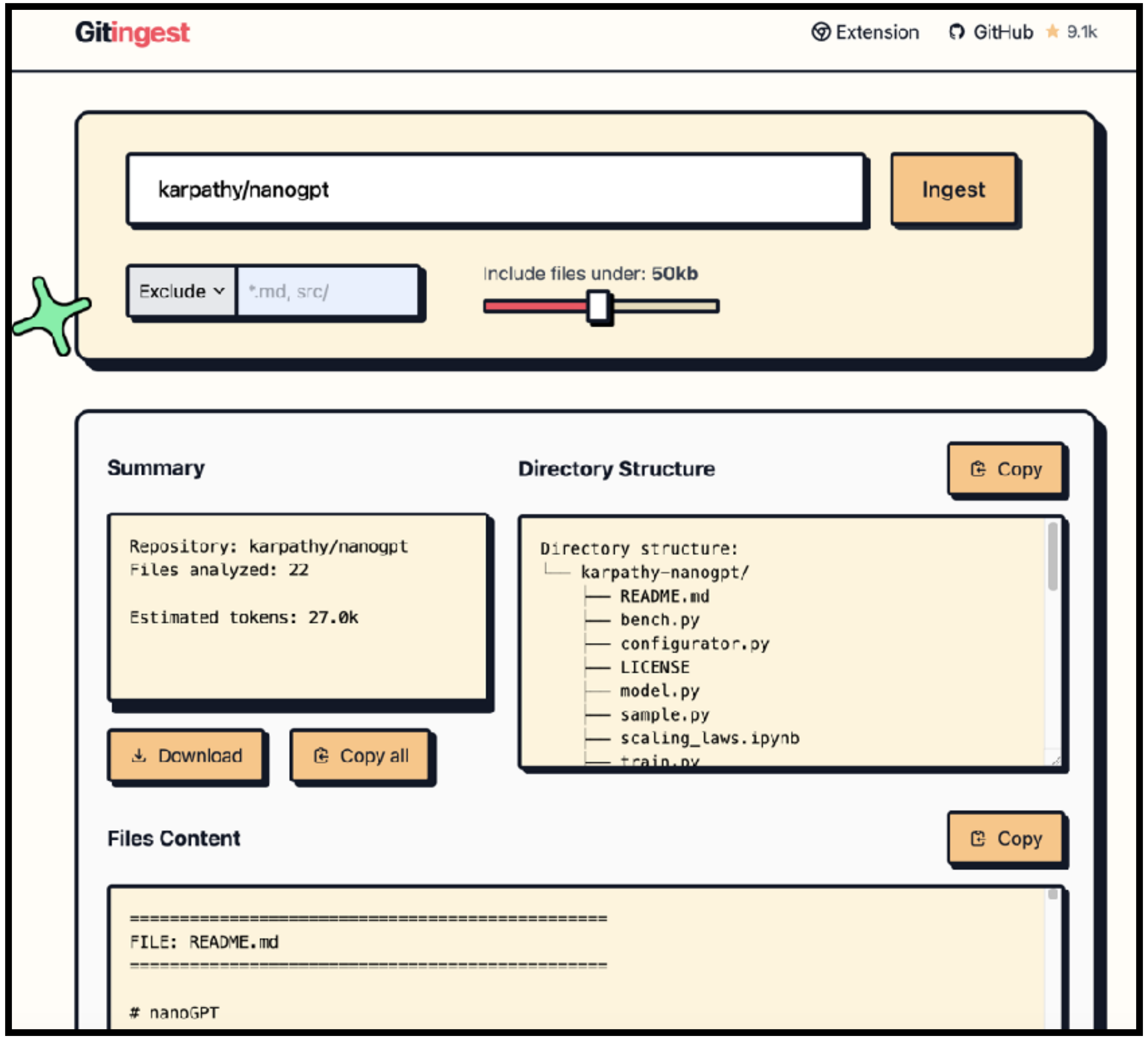
The code editor agent automatically detects all the available tools and calls the relevant tool when you post a related question in the chat. See the Cursor documentation for more details.

Context builders, e.g.: Gitingest

https://github.com/karpathy/nanogpt

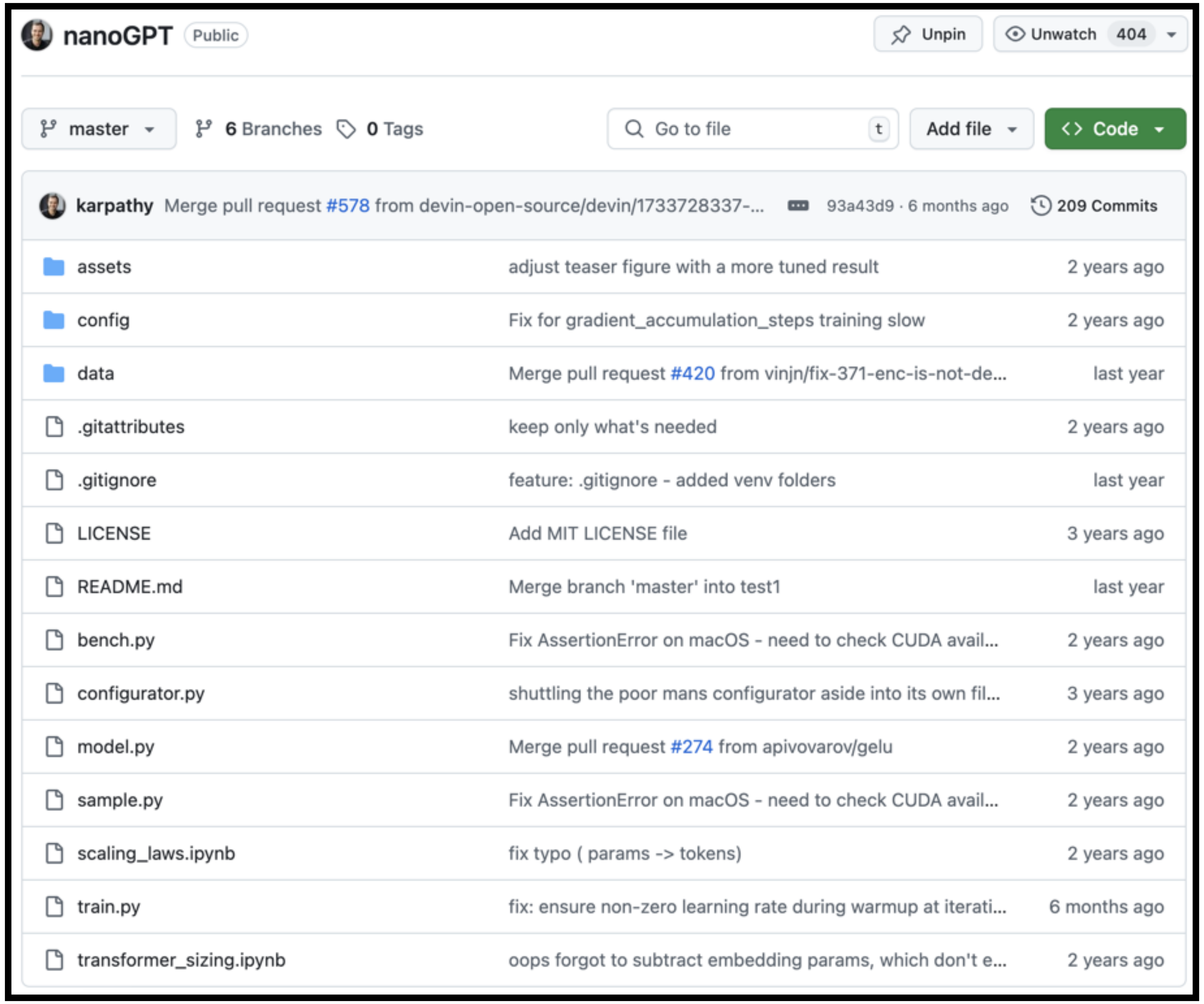


https://gitingest.com/karpathy/nanogpt

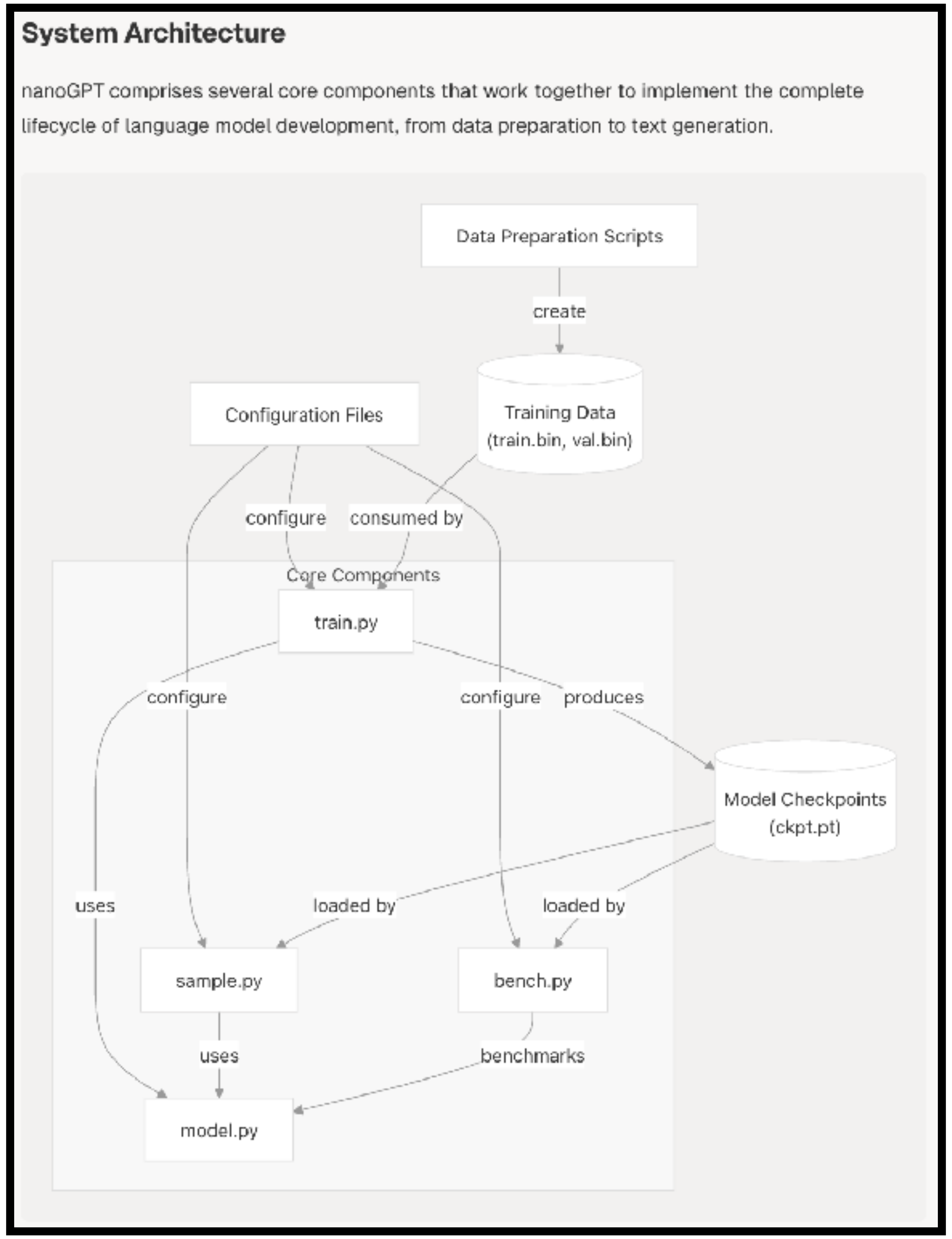


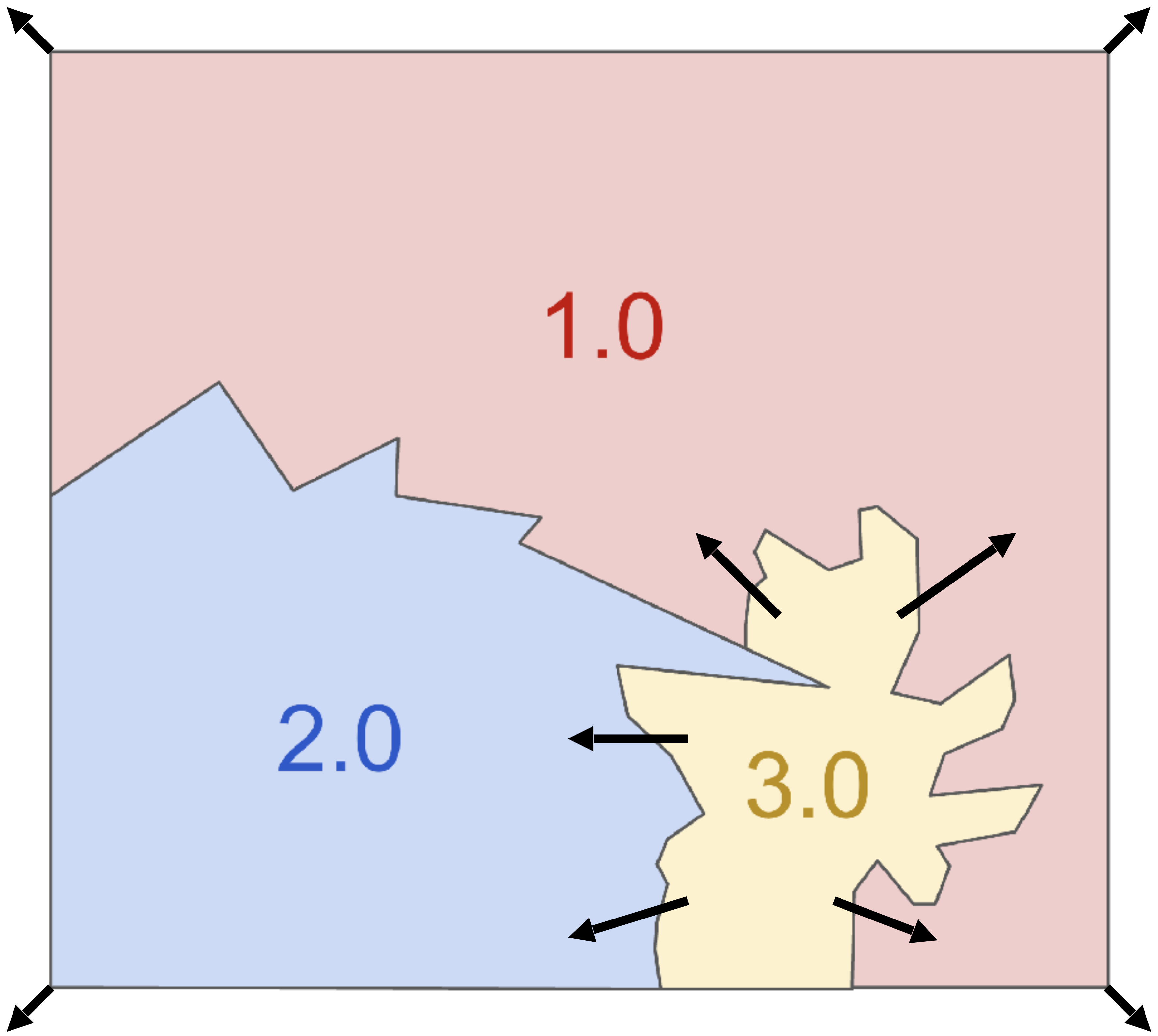
Context builders, e.g.: Devin DeepWiki

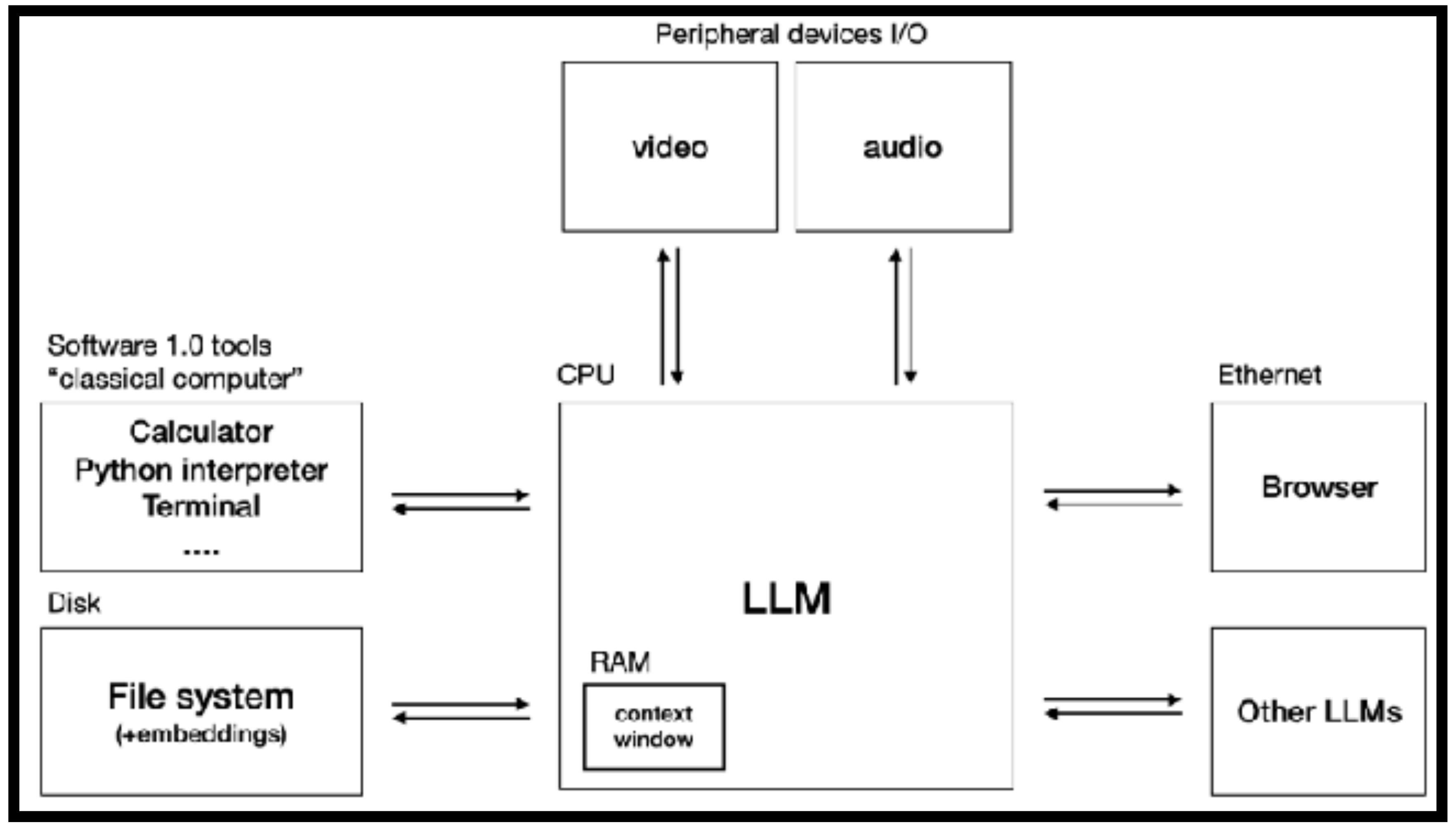
https://github.com/karpathy/nanogpt



https://deepwiki.com/karpathy/nanoGPT/1-overview

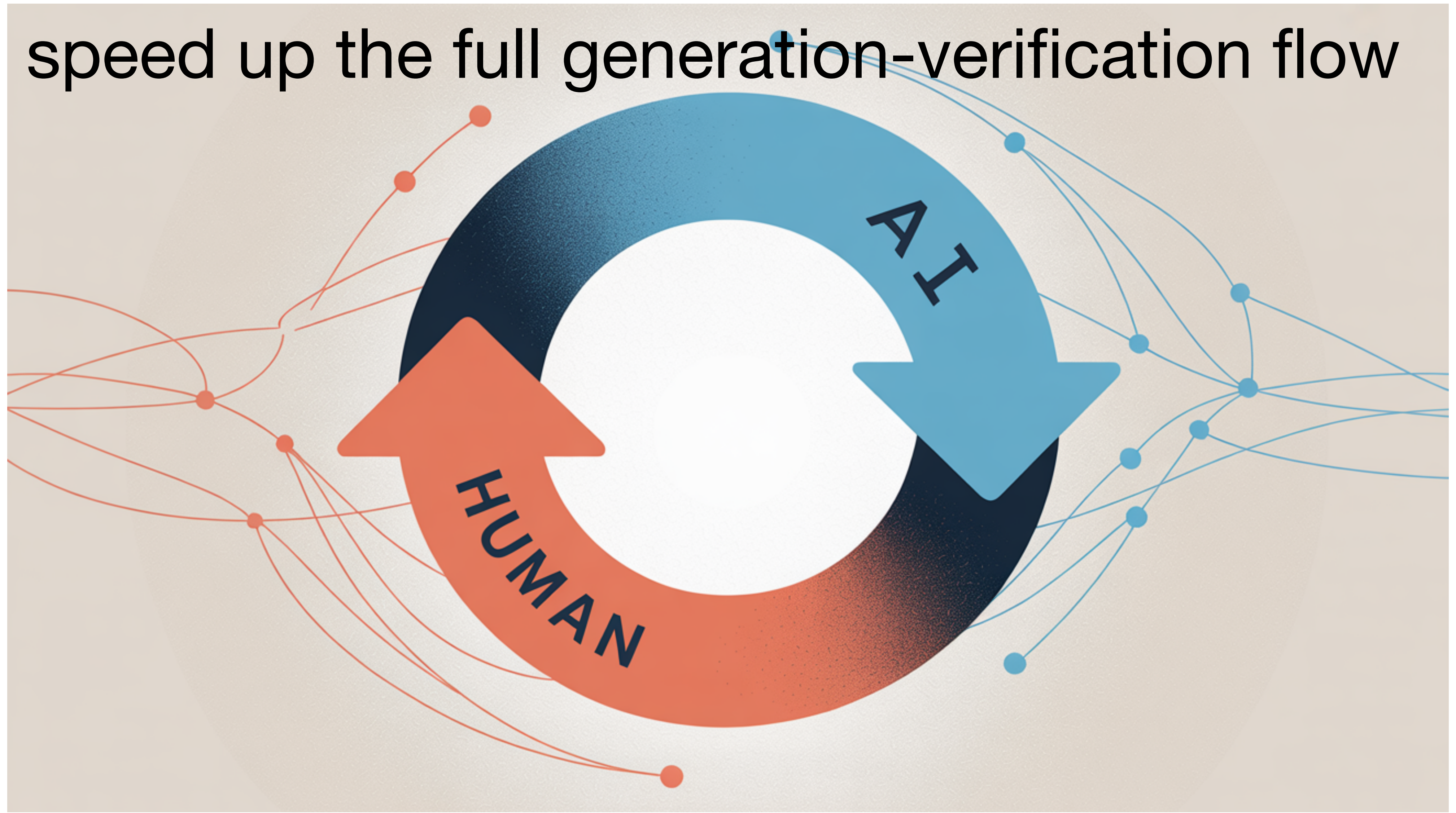






Partial autonomy LLM apps:

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



Build for agents 🤖

ML production myths

Myth #1: Deploying is hard

Myth #1: Deploying is hard

Deploying is easy. Deploying reliably is hard

Myth #2: You only deploy one or two ML models at a time

Myth #2: You only deploy one or two ML models at a time

modern orgs deploy hundreds of micro-models + multiple LLM instances.

Booking.com: 150+ models, Uber: thousands

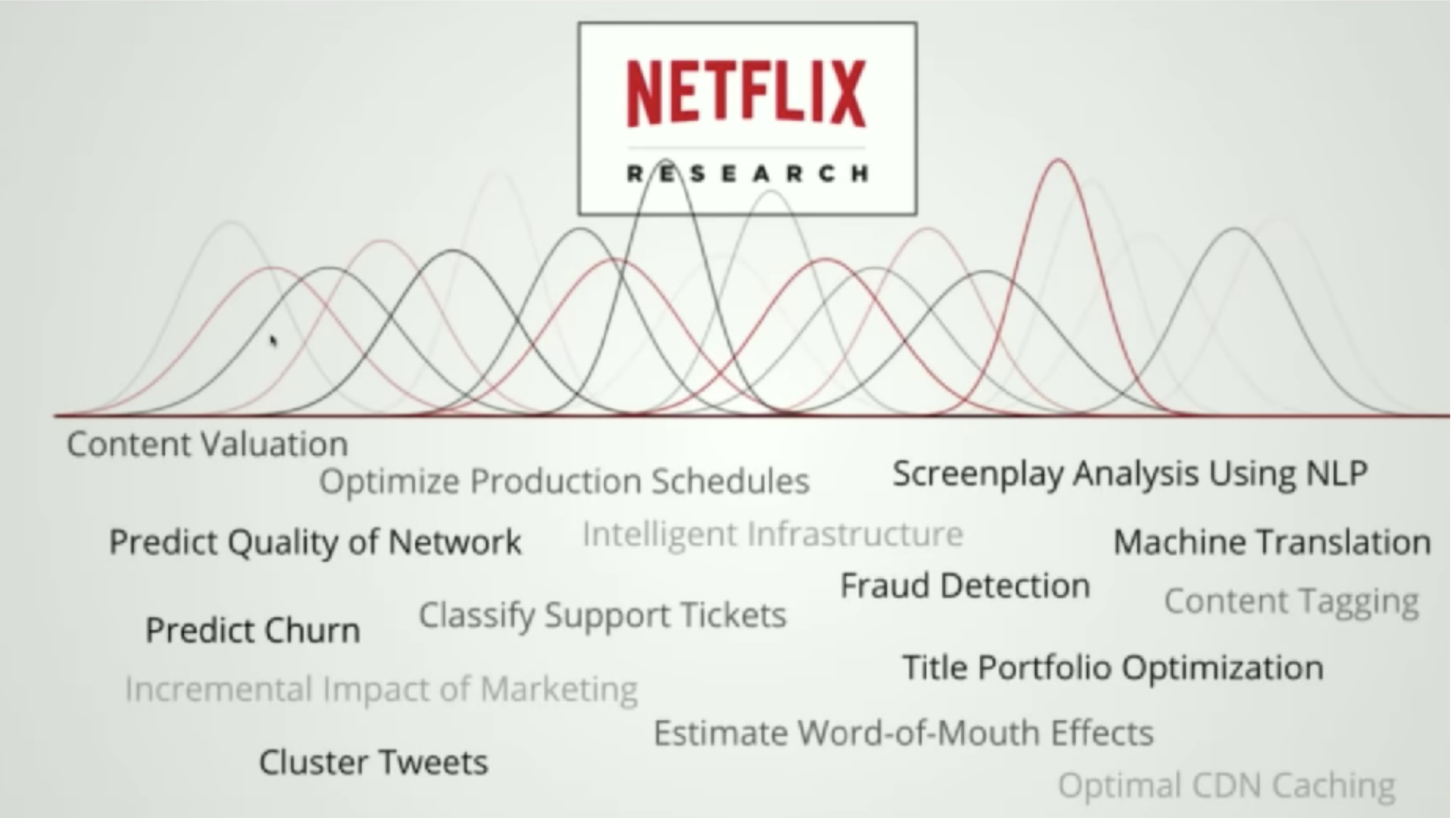


Image from Ville Tuulos (Netflix, Outerbounds)

Myth #3: You won’t need to update your models as much

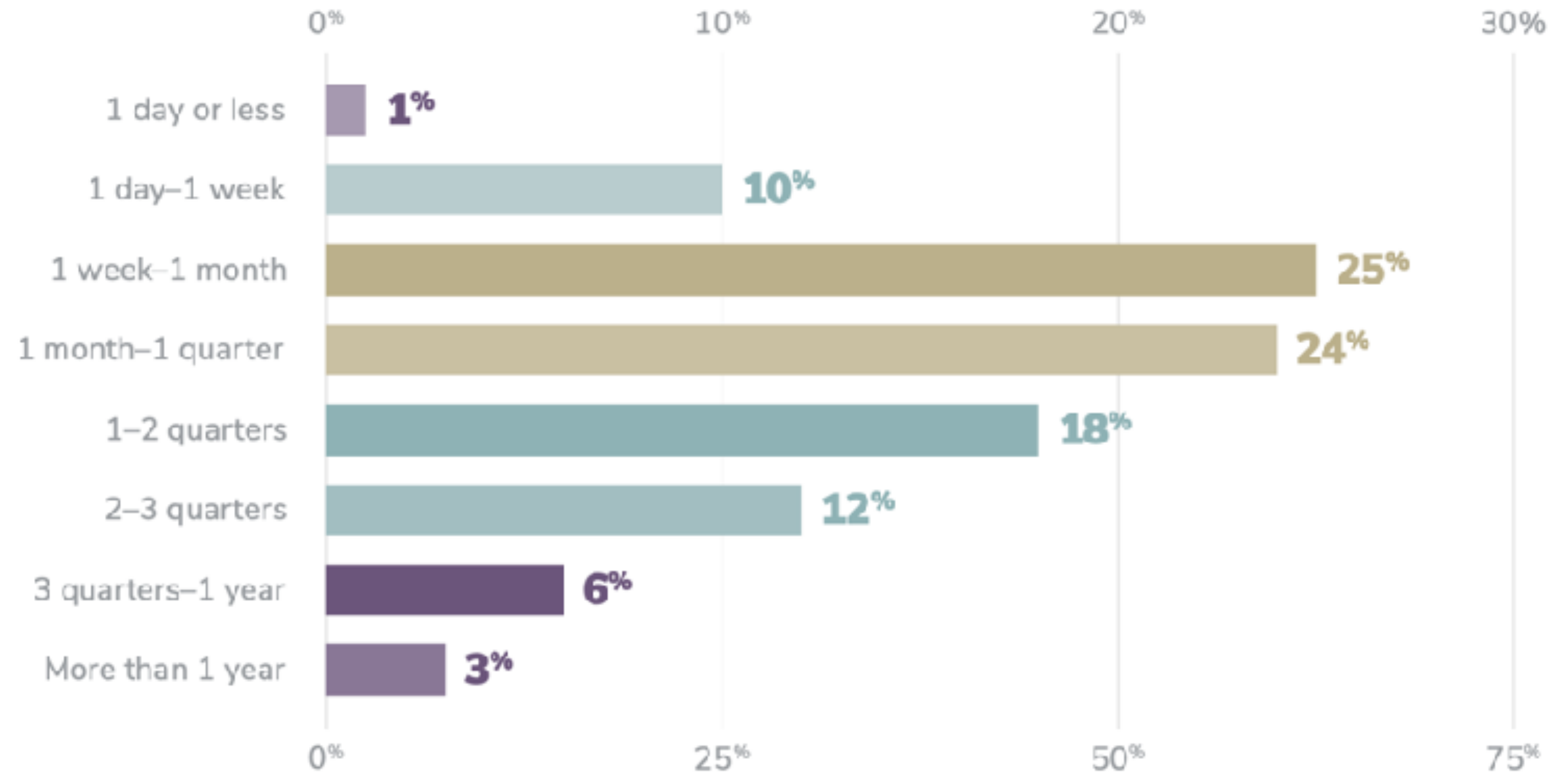
DevOps: Pace of software delivery is accelerating

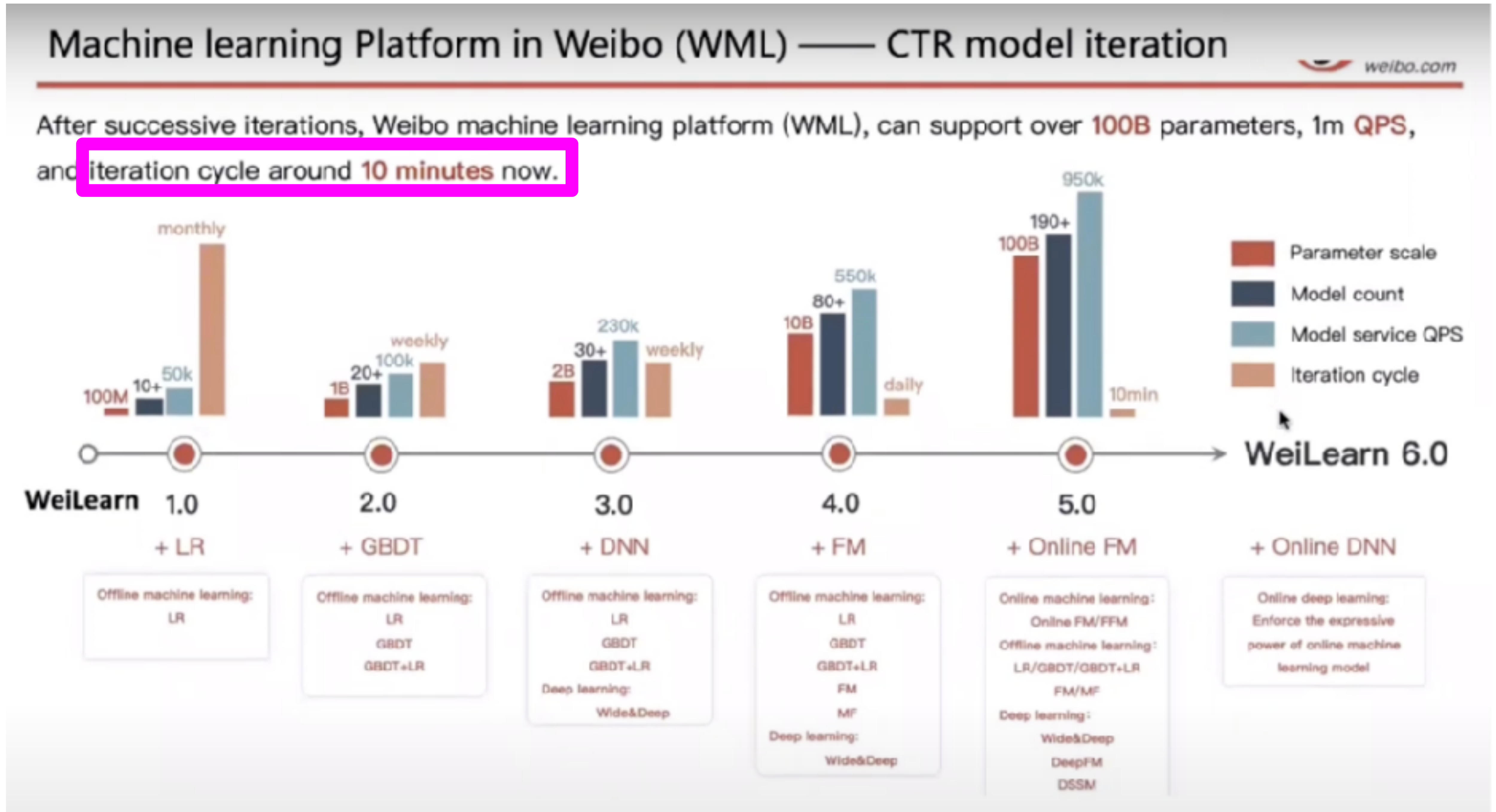
- Elite performers deploy **973x** more frequently with **6570x** faster lead time to deploy (Google DevOps Report, 2021)
- DevOps standard (2015)
    - Etsy deployed 50 times/day
    - Netflix 1000s times/day
    - AWS every 11.7 seconds

DevOps to MLOps: Slow vs. Fast

We'll learn how to do minute-iteration cycle!

Only 11% of organizations can put a model into production within a week, and 64% take a month or longer





Left image from Algorithmia | Right image: Machine learning with Flink in Weibo (Qian Yu, QCon 2019)

Accelerating ML Delivery

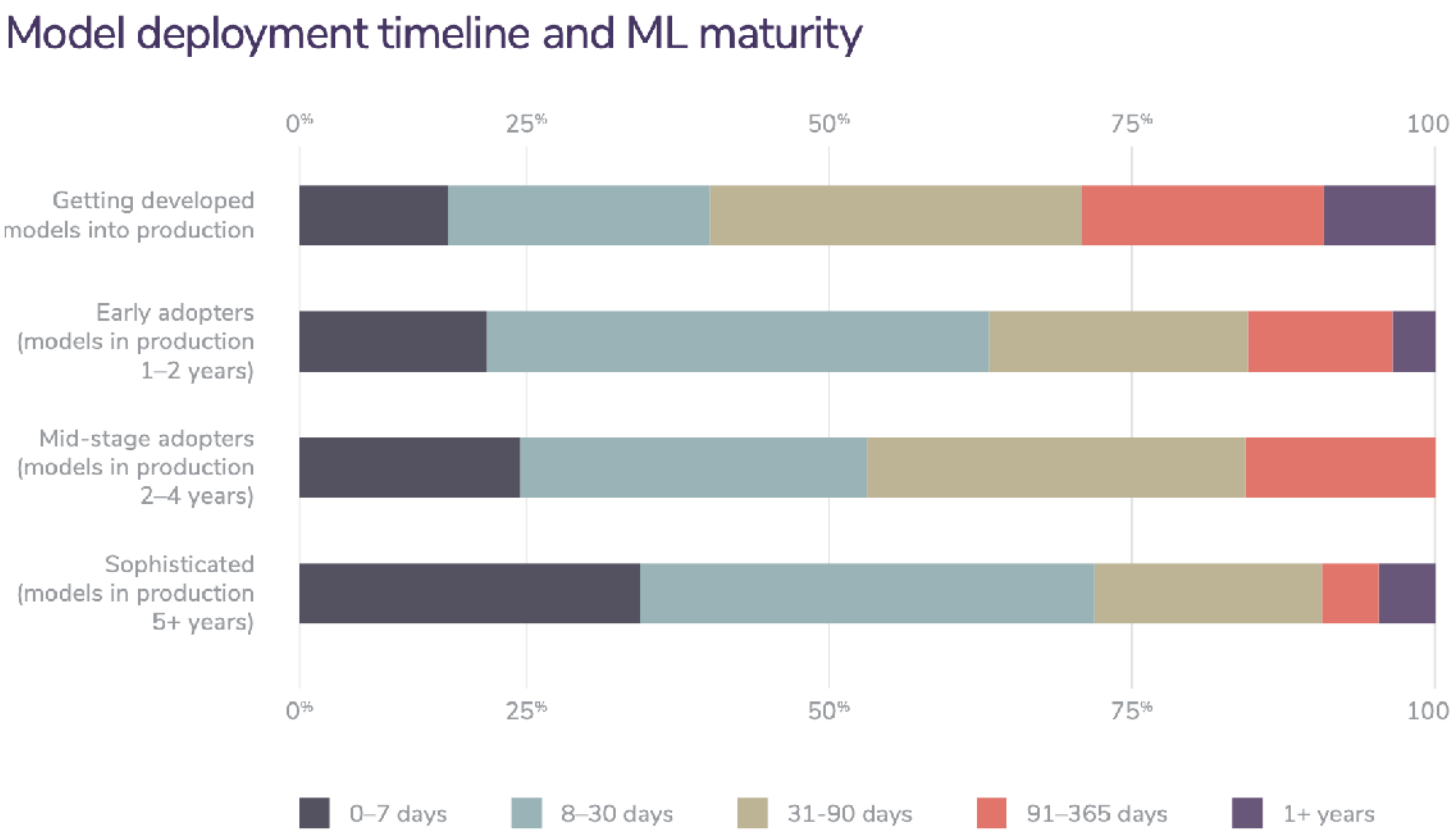
ML + DevOps = 🚀

Myth #4: ML can magically transform your business overnight

Myth #4: ML can magically transform your business overnight

Magically: possible
Overnight: no

Efficiency improves with maturity



2020 state of enterprise machine learning (Algorithmia, 2020)

ML engineering is more engineering than ML

MLEs might spend most of their time:

- wrangling data
- understanding data
- setting up infrastructure
- deploying models

instead of training ML models

Myth #5: Most ML engineers don’t need to worry about scale

Myth #5: Most ML engineers don’t need to worry about scale