

# INNOVATION ENDEAVORS

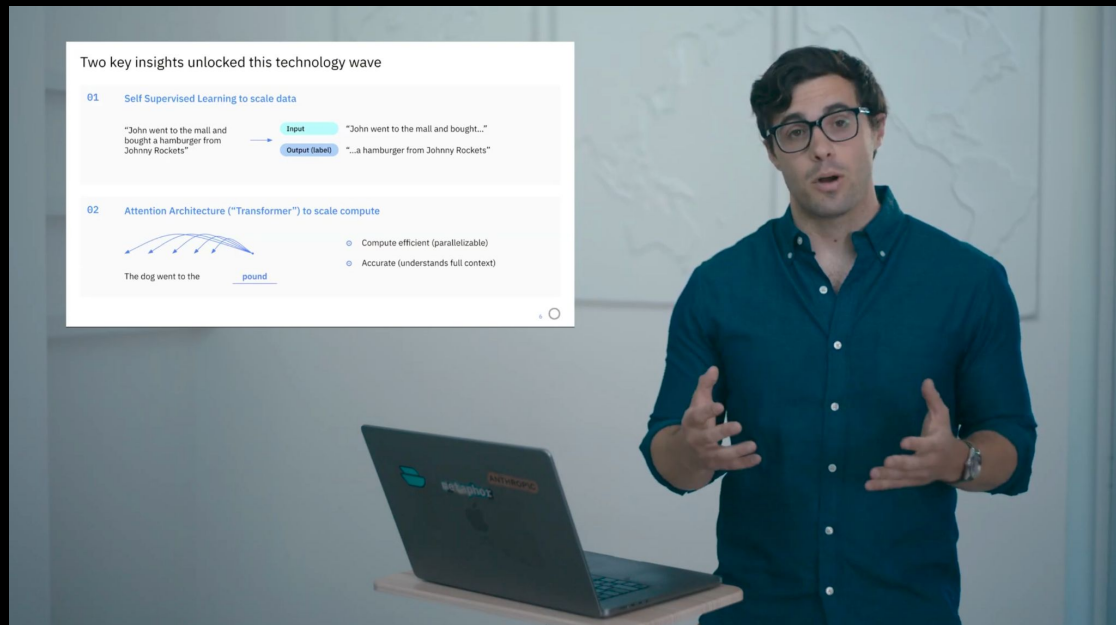


State of Foundation Models, 2025 | Davis Treybig | June 2025

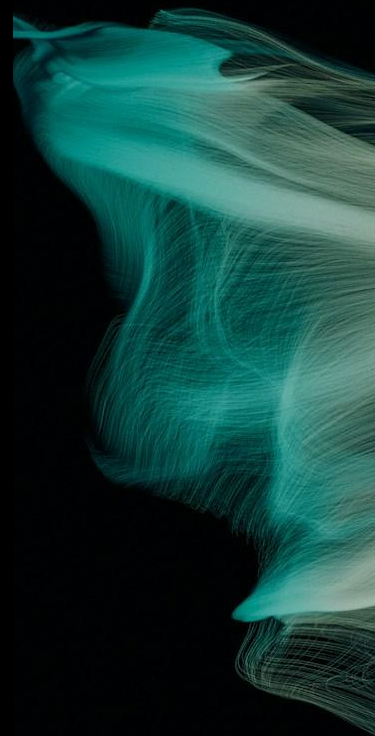


# STATE OF FOUNDATION MODELS, 2025

Video presentation [here](#) >



Edited with [Capsule](#) >



## TABLE OF CONTENTS

01	Setting the Stage
02	Models
03	Use Cases & Applications
04	Building Foundation Model Products
05	Market Structure & Dynamics
06	What's Next?
07	What We're Excited to See Built



# TLDR;

- **Generative AI has gone mainstream** – 1 in 8 workers worldwide now uses AI every month, with 90% of that growth happening in just the last 6 months. AI-native applications are now well into the billions of annual run rate.
- **Scaling continues across all dimensions** – All technical metrics for models continue to improve >10x year-over-year, including cost, intelligence, context windows, and more. The average duration of human task a model can reliably do is doubling every 7 months.
- **The economics of foundation models are...confusing** – OpenAI & Anthropic are showing truly unprecedented growth, accelerating at \$B+ of annual revenue. But, end-to-end training costs for frontier models near \$500M, and the typical model become obsolete within 3 weeks of launch thanks to competition & open source convergence.
- **Just like the smartest humans, the smartest AI will “thinks before it speaks”** – Reasoning models trained to think before responding likely represent a new scaling law – but training them requires significant advances in post-training, including reinforcement learning & reward models. Post-training may become more important than pre-training.
- **AI has now infiltrated almost all specialist professions** – From engineers and accountants to designers and lawyers, AI copilots and agents are now tackling high-value tasks in virtually all knowledge worker domains
- **Agents finally work, but we are early in understanding how to build AI products** – Agents have finally hit the mainstream, but design patterns & system architectures for AI products are still extremely early.
- **“AI-native” organizations will look very different** – Flatter teams of capable generalists will become the norm as generative AI lessens the value of specialized skills. Many roles will blur - such as product, design, & engineering.





# 01 Setting the stage



# Two key insights unlocked this technology wave

## 01 Self Supervised Learning to scale data

“John went to the mall and  
bought a hamburger from  
Johnny Rockets”



Input

“John went to the mall and bought...”

Output (label)

“...a hamburger from Johnny Rockets”

## 02 Attention Architecture (“Transformer”) to scale compute



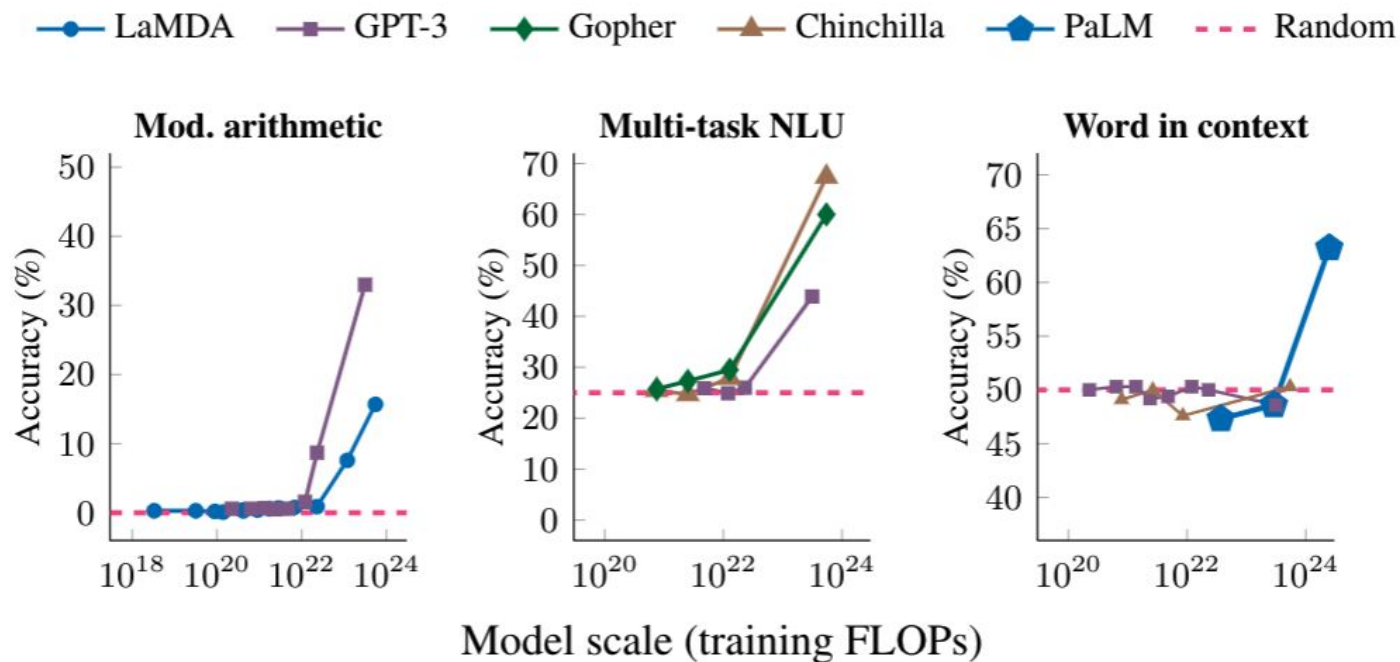
The dog went to the

pound

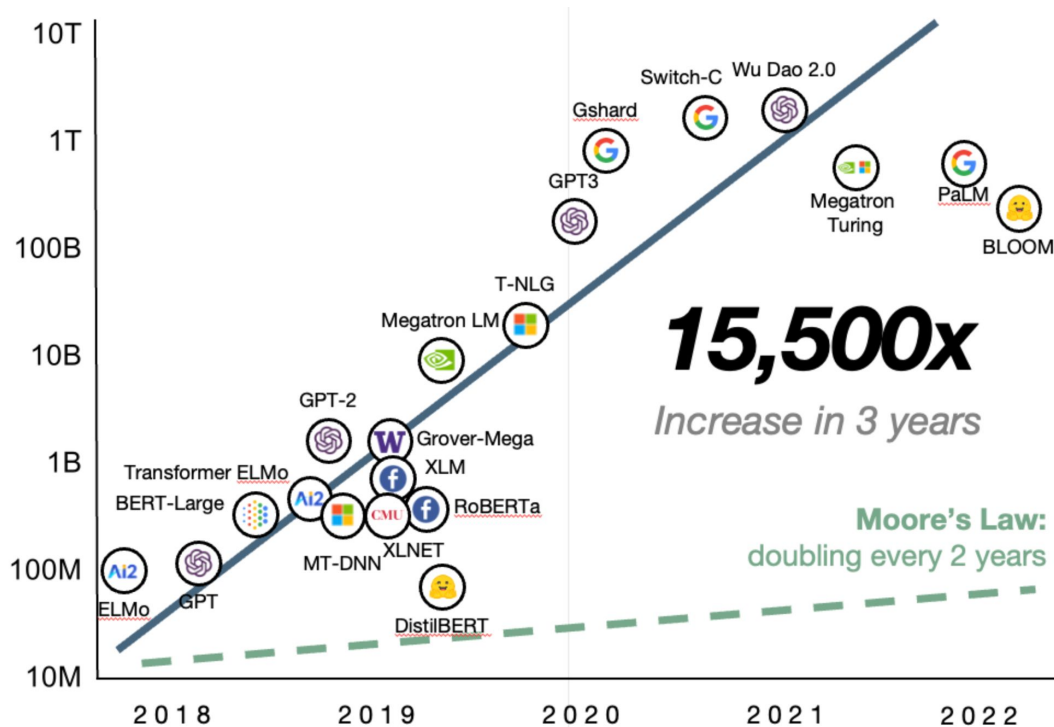
- Compute efficient (parallelizable)
- Accurate (understands full context)



# Scaling models leads to “emergent” behavior



So we pushed for exponential growth in modal size...

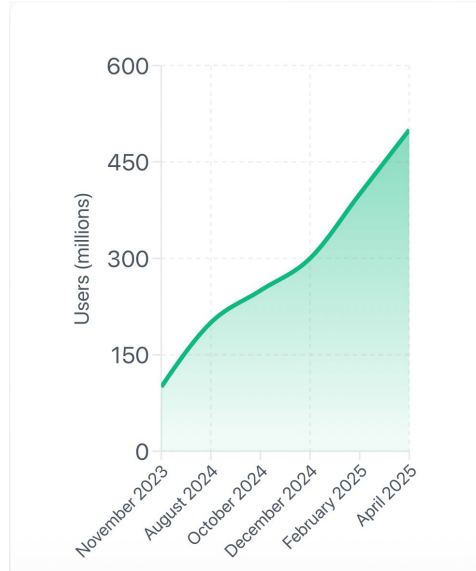




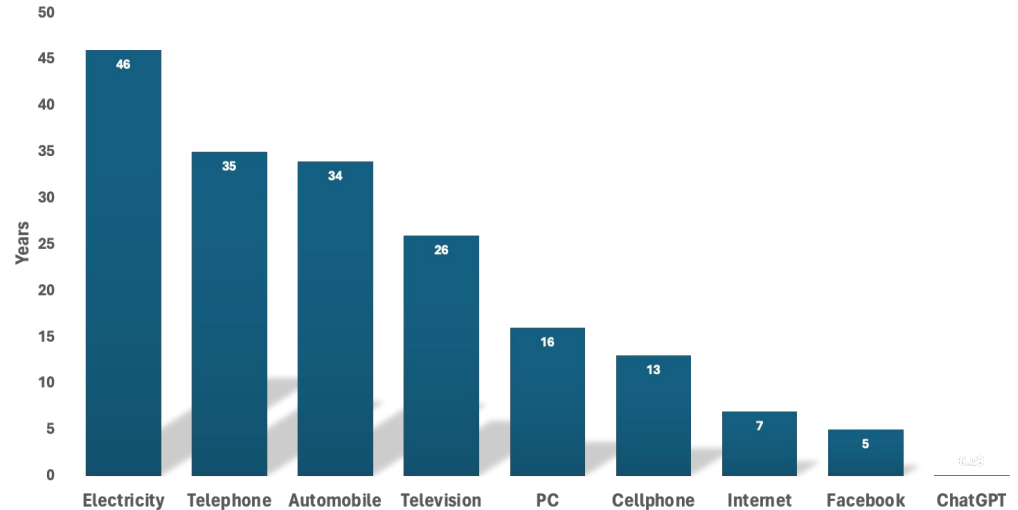
# As a result, we got the fastest rate of adoption of new technology of all time

## ChatGPT's Explosive Growth

Weekly Active Users (in millions)



## ChatGPT reached 100M users in 60 days



## As well as some of the fastest revenue ramps of all time

Model	Revenue	Active Users	Timeframe	Employees
GitHub Copilot	~400M ARR	1,500,000	3 years	NA
Midjourney	~200M ARR	20,000,000	2 years	~40
Cursor	~100M ARR	360,000	1 year	~20

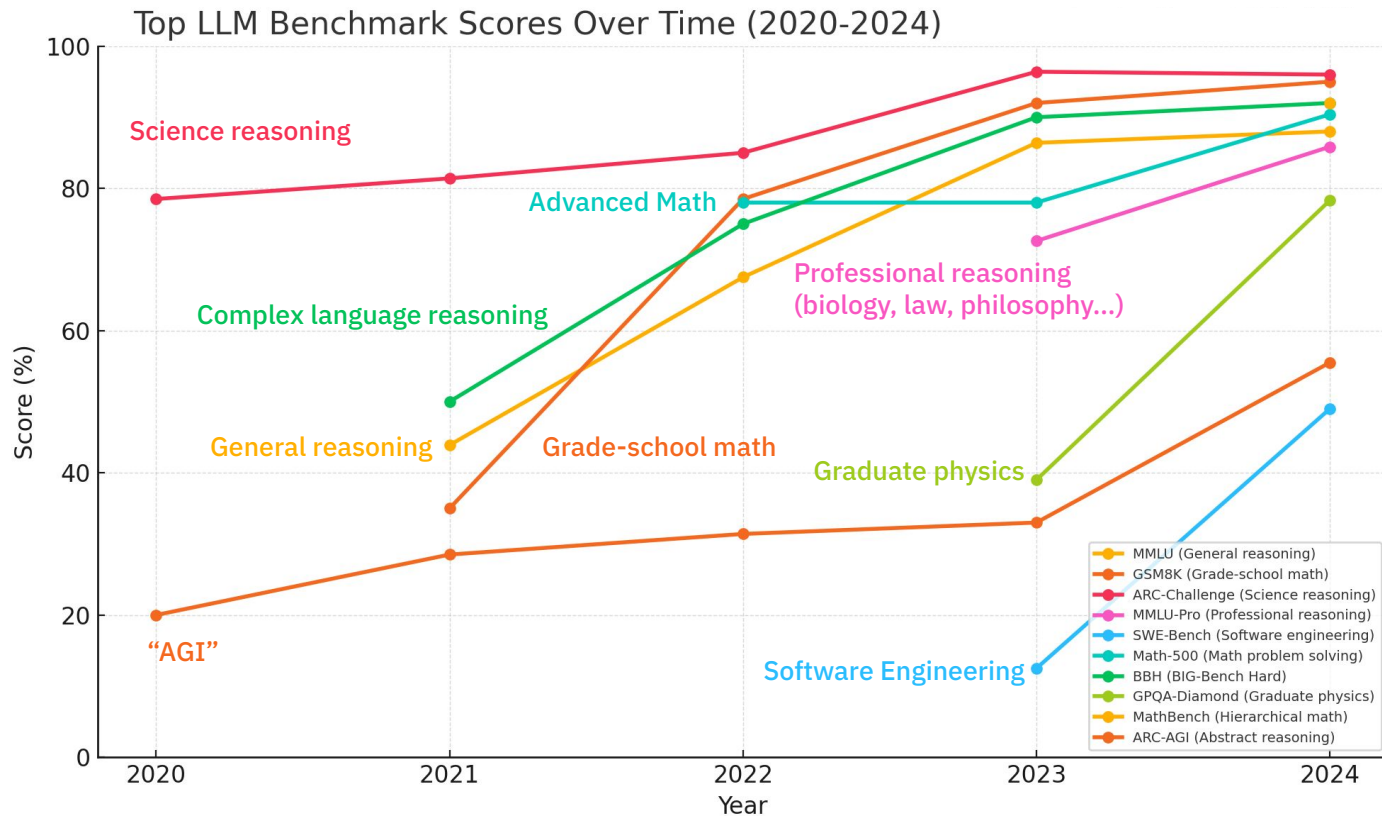


# All technical metrics are following exponential curves

	January 2023	Spring 2025	Delta
Context window (frontier)	2 – 8k tokens	~1M tokens	~ <b>100 – 500x</b> increase
Cost/token (GPT4-level)	\$100 million	\$.1 million	> <b>1000x</b> reduction
Compute to train (FLOP)	~10 <sup>24</sup>	~10 <sup>28</sup>	> <b>1000x</b> increase

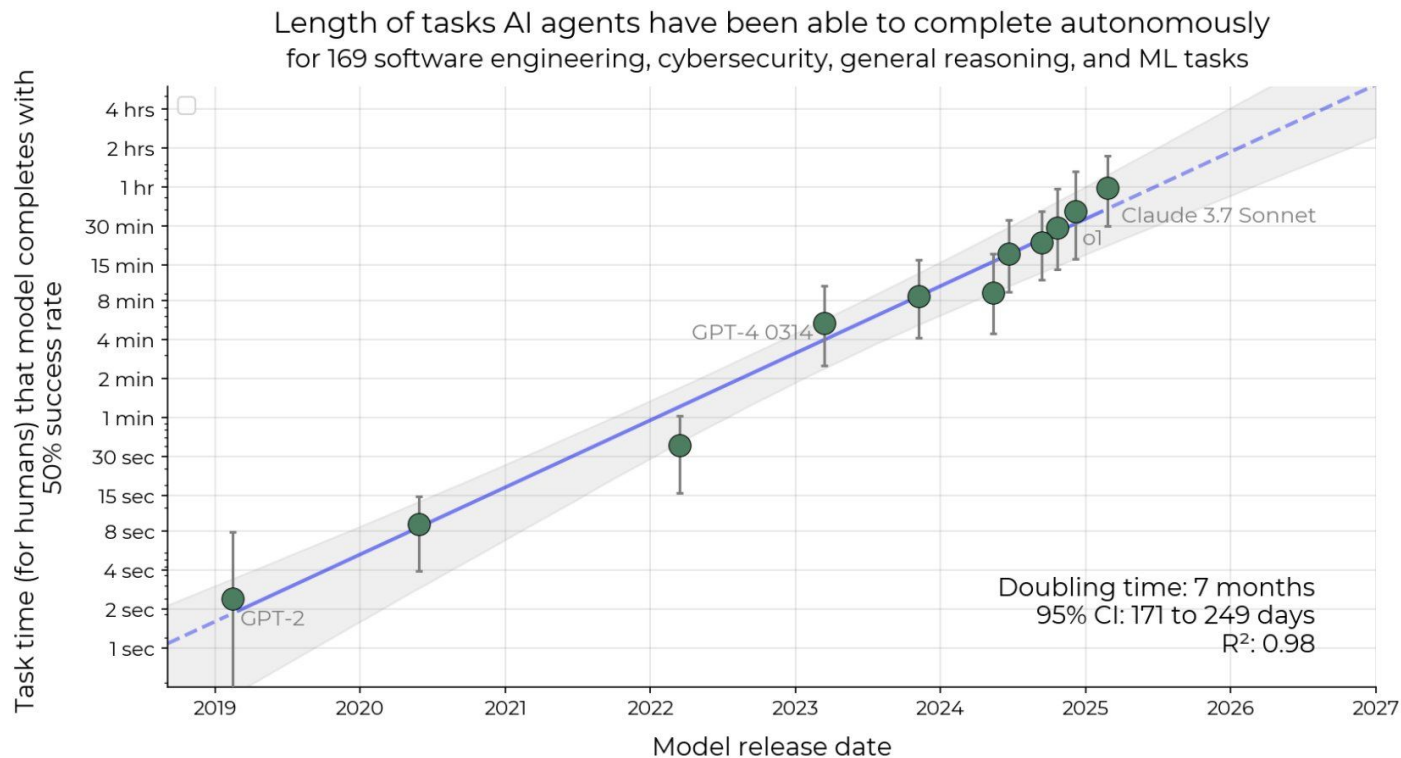


# LLMs quickly surpass almost all new benchmarks as they are released



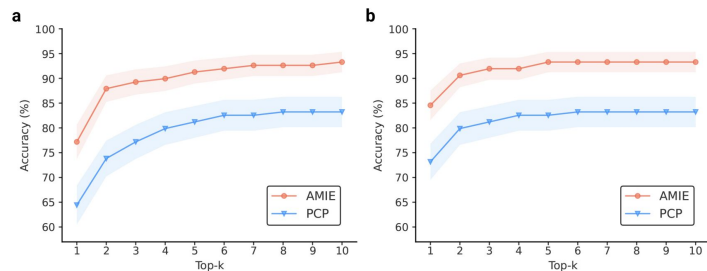


# The task span LLMs can handle has jumped from 1 second to 1 hour — in just 5 years



# LLMs reasoning capabilities now exceed humans in various domains

## 01 LLMs now outperform doctors in aggregate on numerous diagnostic tasks



**Figure 3 | Specialist-rated top-k diagnostic accuracy.** AMIE and PCP's top-k DDx accuracy are compared across 149 scenarios with respect to the ground truth diagnosis (a) and all diagnoses in the accepted differential (b). Bootstrapping ( $n=10,000$ ) confirms all top-k differences between AMIE and PCP DDx accuracy are significant with  $p < 0.05$  after FDR correction.

## 02 LLMs now solve geometry problems more accurately than 99.999% of people on Earth

**IMO 2015 P3**

Let  $ABC$  be an acute triangle. Let  $(O)$  be its circumcircle,  $H$  its orthocenter, and  $F$  the foot of the altitude from  $A$ . Let  $M$  be the midpoint of  $BC$ . Let  $Q$  be the point on  $(O)$  such that  $QH \perp QA$  and let  $K$  be the point on  $(O)$  such that  $KH \perp KQ$ . Prove that the circumcircles  $(O_1)$  and  $(O_2)$  of triangles  $FKM$  and  $KQH$  are tangent to each other.

**Solution**

```
[...]
Construct D: midpoint BH [a]
[a], O2: midpoint HQ → BQ ∥ O2D [20]
[...]
Construct G: midpoint HC [b]
∠GMD = ∠GO2D → M O2 G D cyclic [26]
[...]
[a], [b] → BC ∥ DG [30]
[...]
Construct E: midpoint MK [c]
[c] → ∠KFC = ∠KO2E [104]
[...]
∠FKO2 = ∠FKO1 = ∠KO2 ∥ KO2 [109]
[109] → O, O2, K collinear → (O1), (O2) tangent
```



Diffusion has seen a similarly exponential rate of improvement



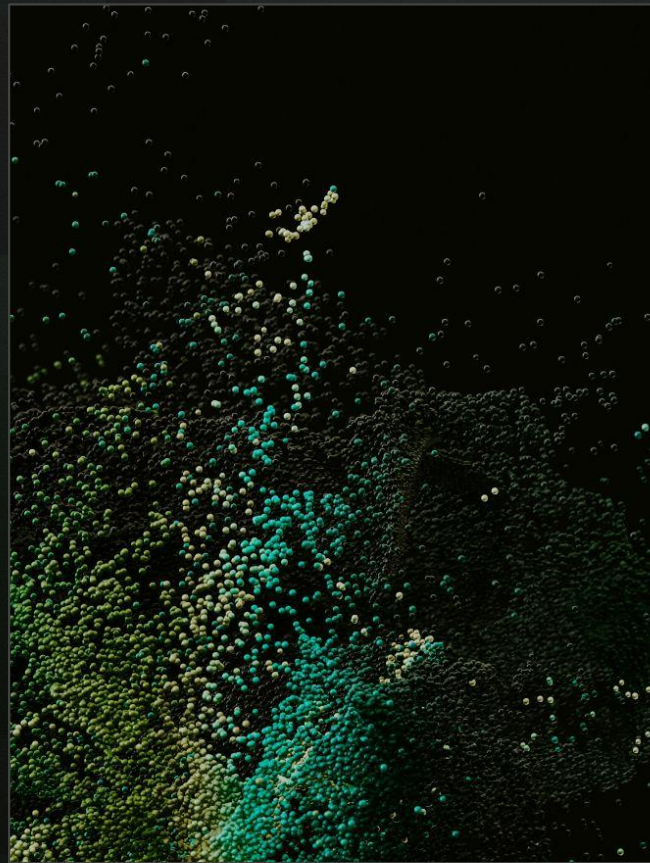
Imagen – Google Deepmind (~2022)



Visual Electric (2024)



## 02 Models





# Training costs for frontier models continue to balloon

Leading models now cost >\$300M

Model	Release Date	Estimated Training Cost (millions)
GPT-3	2020	\$4.50
PaLM 540B	2022	\$10.00
Claude 2	2023	\$25.00
GPT-4	2023	\$100.00
Gemini Ultra	2023	\$190.00
LLaMA 3.1 (405b)	2024	\$120.00
Llama 4	2025	\$300.00+



But, frontier models also depreciate on a 6–12 month timescale

### GPT-4

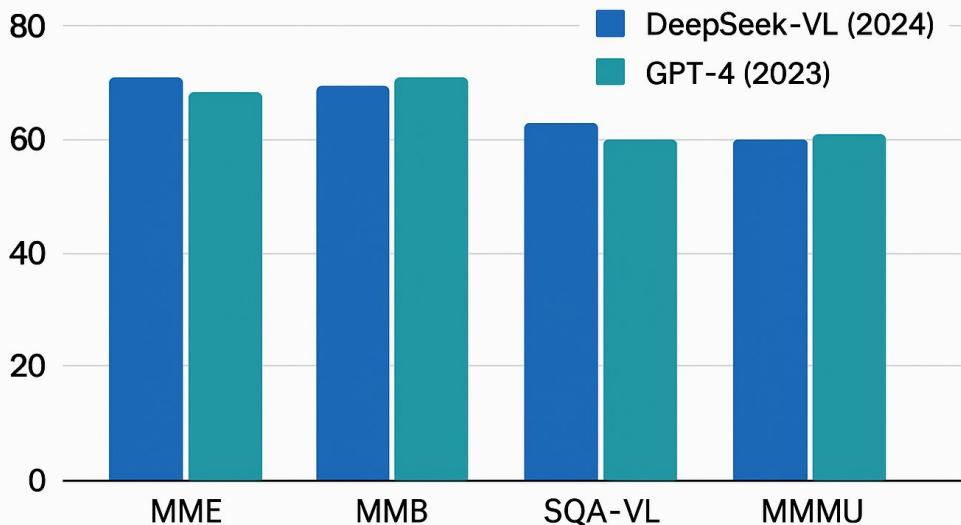
- ◉ \$100M+ to train
- ◉ Closed source
- ◉ Released March 2023

---

### DeepSeek-VL

- ◉ <\$10M to train
- ◉ Open Source
- ◉ Released March 2024

**DeepSeek-VL (2024) vs. GPT-4 (2023)**

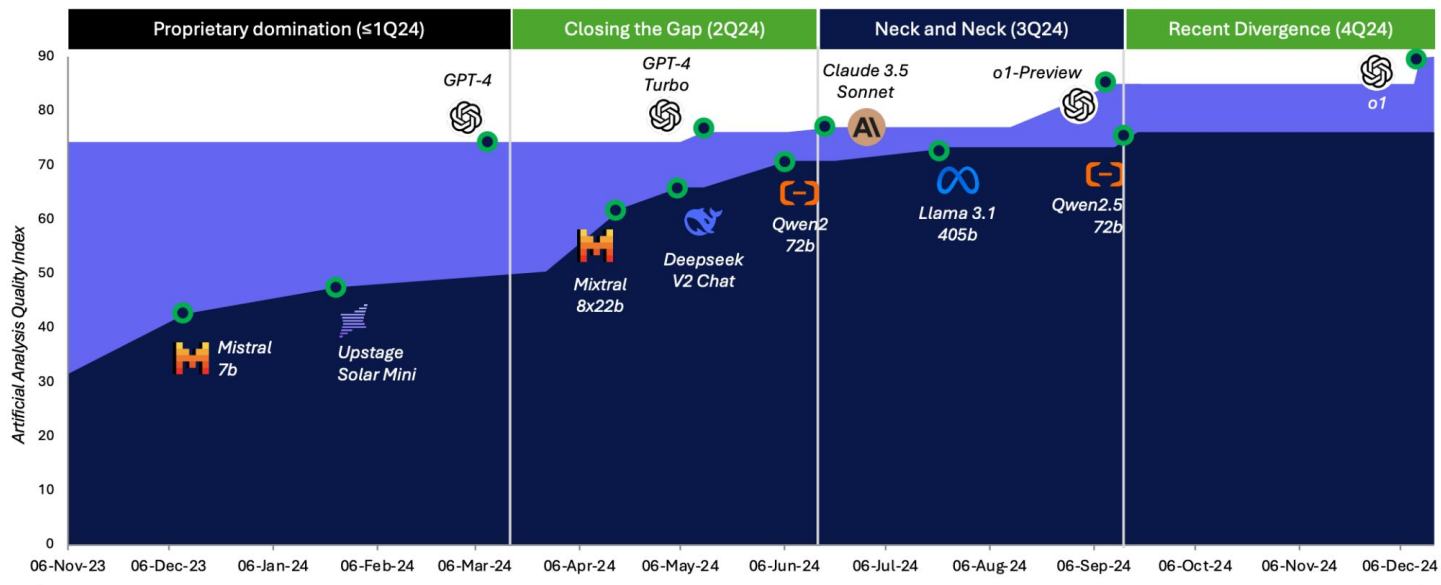


# Open source continues to converge with closed source

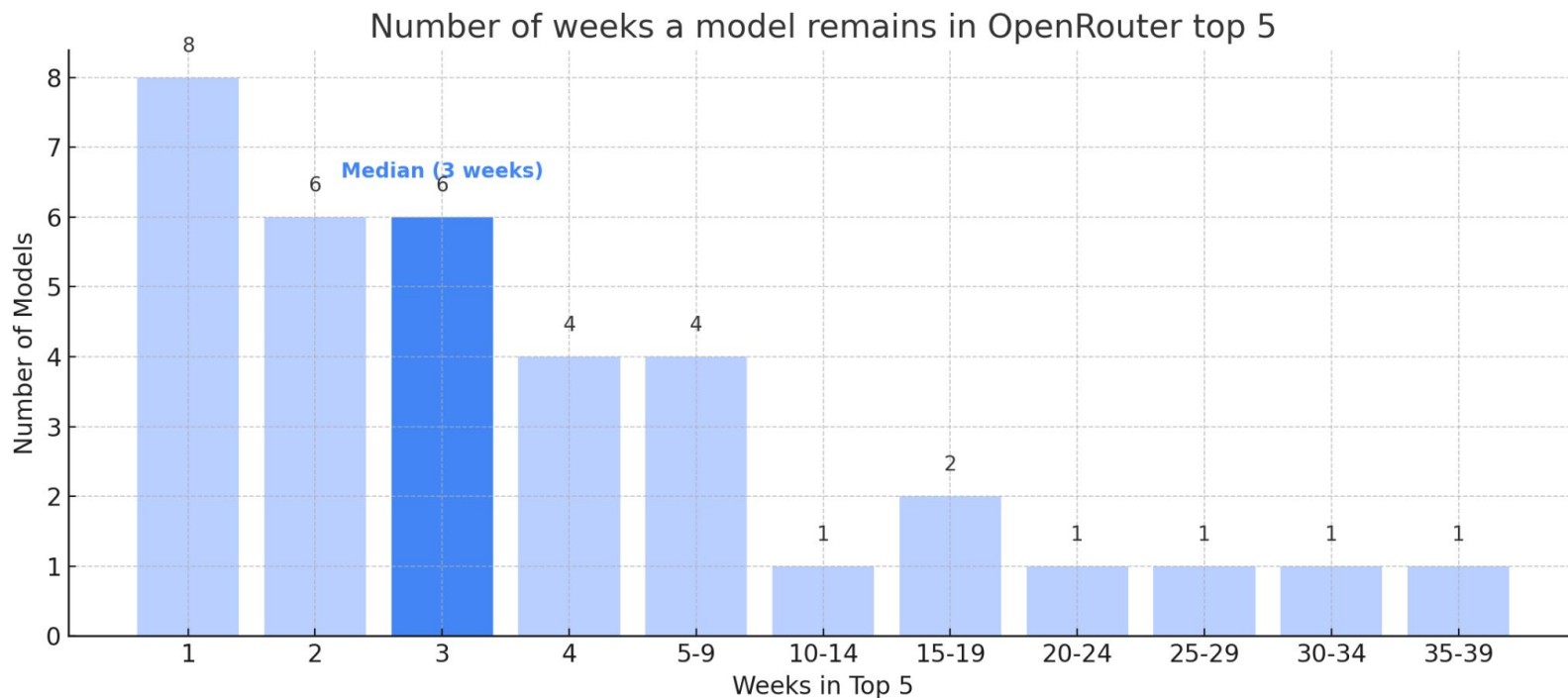
Driven by models from Meta, Mistral and Alibaba, the performance gap between open source and proprietary models has decreased significantly

## Model Quality: Leading Proprietary and Open Weights Models

Based on proprietary and open-source models that resulted in an increase in Artificial Analysis Intelligence Index score



# Most models only last 3 weeks





# Data budgets are also insane, though data budgets and compute budgets are blurring

- ◉ Deepmind spending \$1B a year on data annotation
- ◉ OpenAI spending ~3B a year on training and data
- ◉ Meta spent \$125M on post-training data for LLaMA 3
- ◉ OpenAI paying \$2–3k per individual reasoning trace

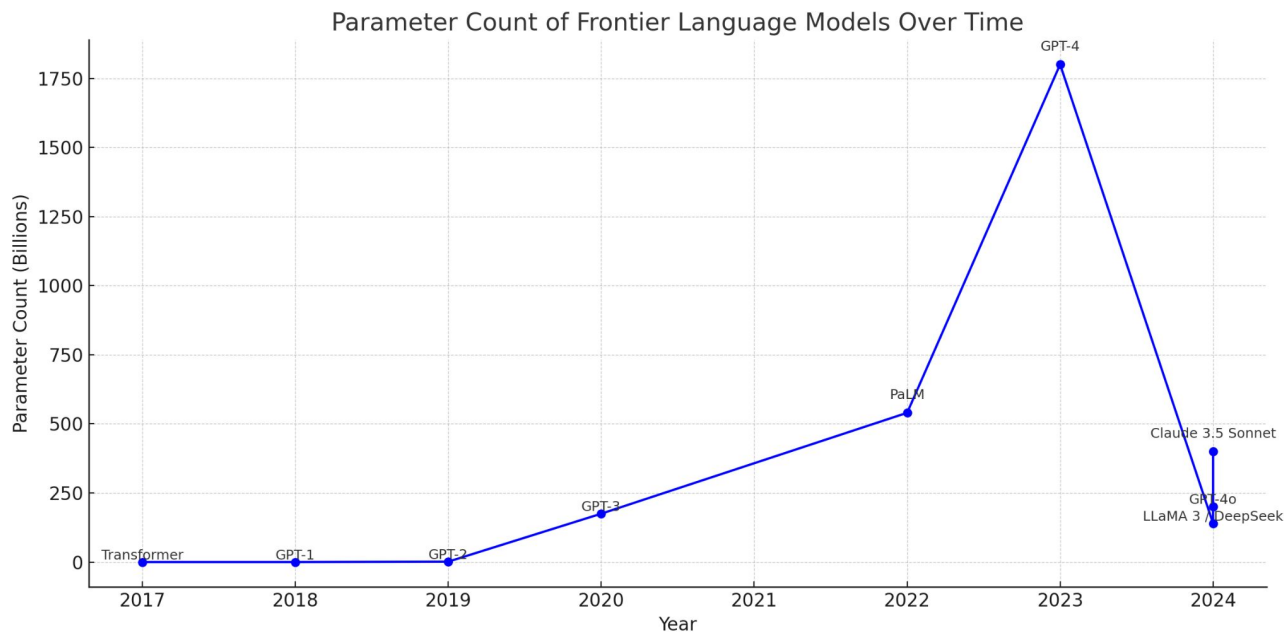
## Illustrative breakdown of spend for leading model

<b>Pre-training</b>	150-300M
<b>Post-training (incl RL)</b>	50-150M
<b>Data</b>	50-150M



# Zeitgeist shifting away from purely scaling parameters & pre-training

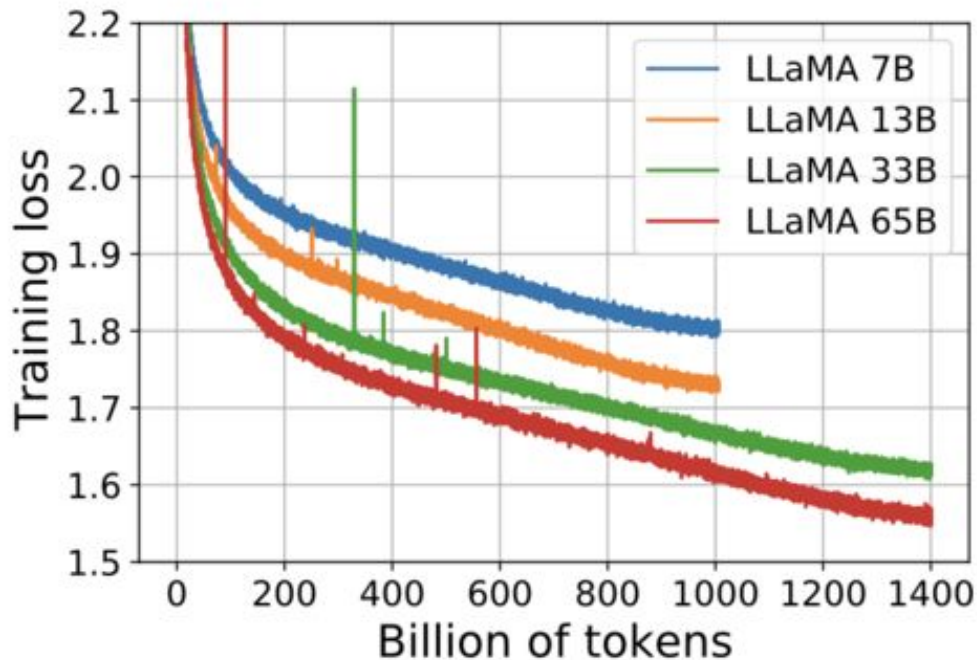
Smaller models are more efficient to serve - in cost, memory, and latency - and advances in inference-time compute are reducing the need to max out pre-training



# Smaller models more saturated on large datasets are less “training efficient”, but are much better to serve

For a given loss, smaller models requires far more training tokens, but:

1. Smaller models are easier and cheaper to run inference
2. Smaller models are lower latency



# Pre-training as we know it will end

Compute is growing:

- Better hardware
- Better algorithms
- Larger clusters

Data is not growing:

- We have but one internet
- **The fossil fuel of AI**



## What's Next?

- ⦿ Synthetic data
- ⦿ Agents (systems engineering)
- ⦿ Inference time scaling
- ⦿ ?



# Inference time compute (“reasoning”) is a new frontier

## User Prompt

What’s the implication of the new Canadian prime minister on foreign exchange rates?

## Reasoning

**\*Thought for 5 minutes\*** →

## Output

Below is a holistic overview of the impact the new Canadian prime minister may have on FX rates, broken down by....

### *Internal Monologue*

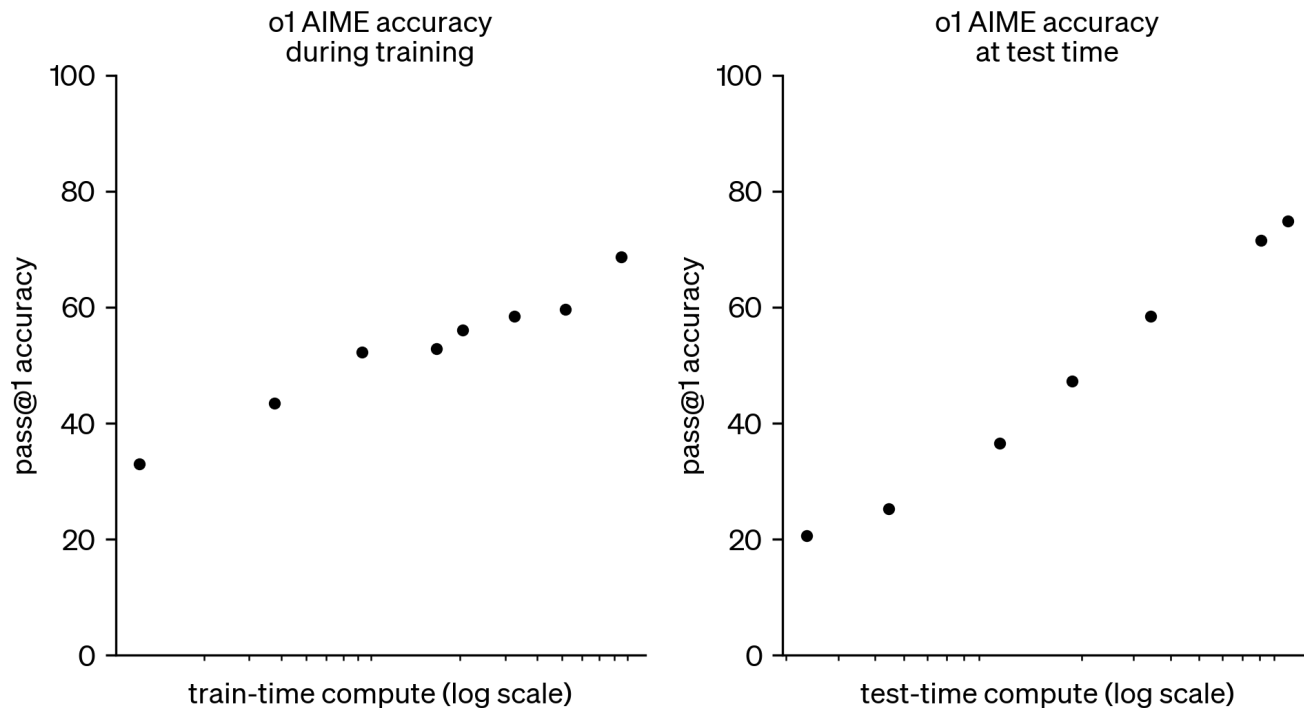
*To answer this question, I first need to consider:*

- 1. The economic drivers of exchange rates*
- 2. Canada’s current exchange rates*
- 3. The differences in policy between Canada’s new and former prime minister*

*To start....*



...and represents a new scaling law for models



Interestingly, test-time compute is not a particularly new concept

Research

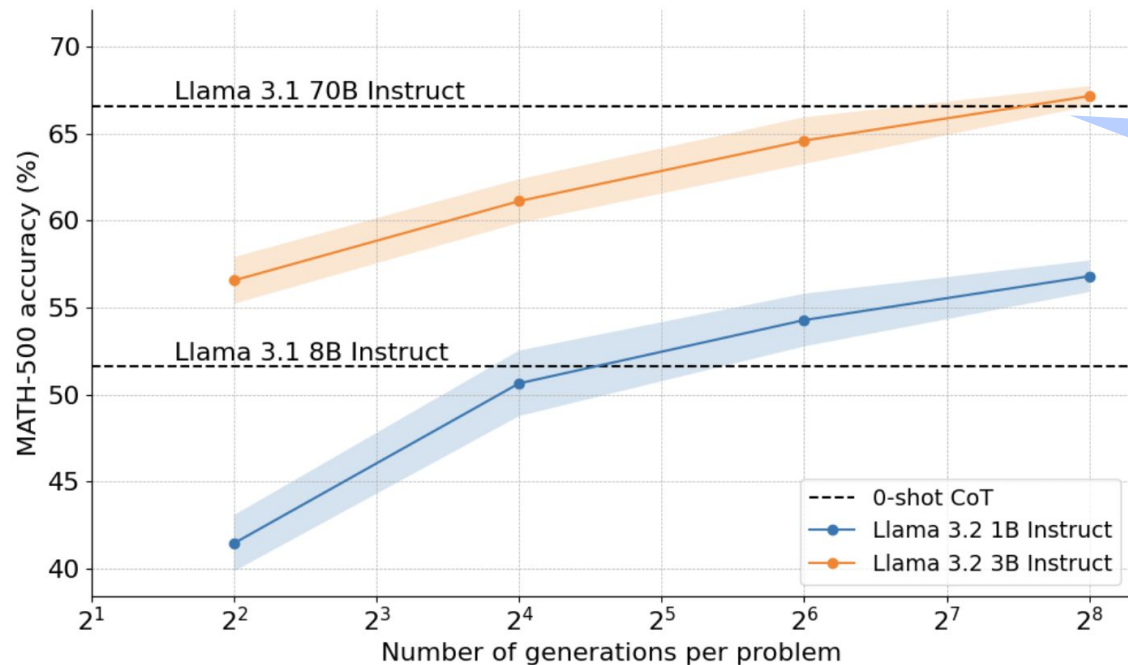
# CICERO: An AI agent that negotiates, persuades, and cooperates with people

November 22, 2022





# Small reasoning models can outperform models 10–20x larger given enough time to think



3B reasoning model beats 70B model given enough thinking



# There are multiple ways to develop reasoning models

---

## Post-train on reasoning traces

- Pay for or create labeled reasoning traces
  - Synthetically generated reasoning traces in verifiable domains (e.g. Math problems)
  - Train process reward models (PRM) or outcome reward models (ORM) to guide sampled generations
- 

## Use “search” techniques at inference time

- Model and secondary system (verifier/validator) go back and forth to guide “thinking”
- 



# There are multiple ways to develop reasoning models

---

Post-train on  
reasoning traces



**Model “thinks with itself” for a long time** – single, continuous, long stream of output tokens

---

Use “search” techniques  
at inference time

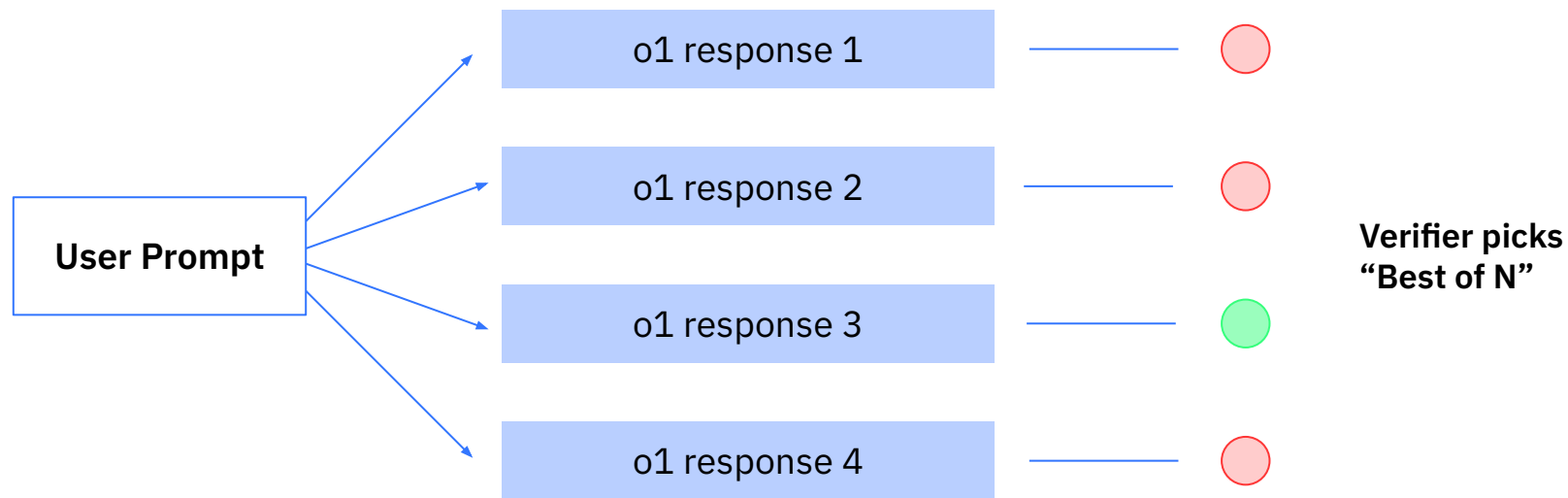


**Control flow mediates interaction** between model and secondary systems guiding thinking

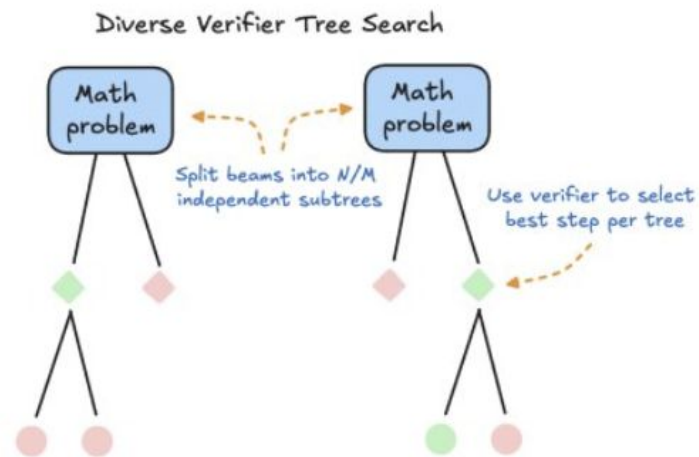
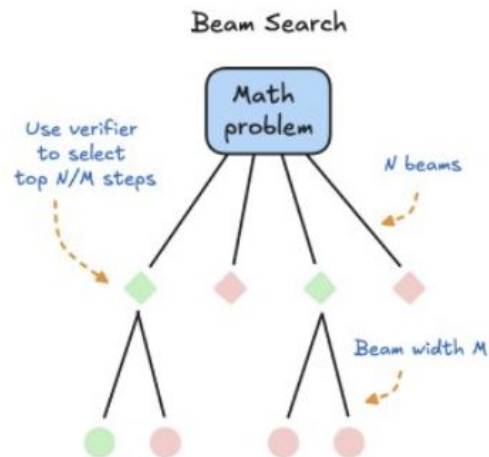
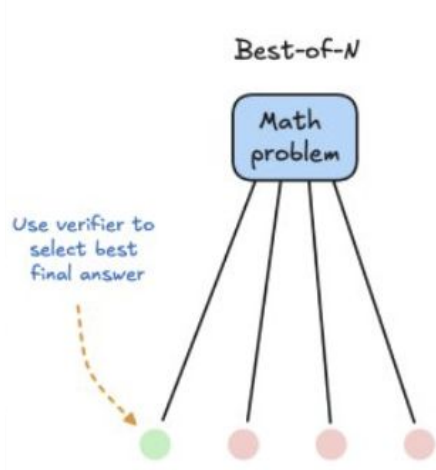
---



o1-pro is likely “best of n o1”



# Common versions of inference-time search techniques



# Challenges and open questions with reasoning models

---

## **How well do easily constructed synthetic data sets generalize?**

Does synthetic math & coding data translate well to other domains?

## **What is the optimal reinforcement learning algorithm/approach?**

- Sampling strategy
- Process vs outcome rewards
- Noisy & sparse reward signals in complex tasks
- Computational cost/complexity

## **Data generation & acquisition**

High end reasoning traces worth \$3k...

---



# The post-training algorithm landscape continues to evolve

“Write a short story about a dog”

	Response	Response	Mechanism
Proximal Policy Optimization (PPO)	“The dog jumped over a tree...”	Reward = 3.7	Reinforcement learning
Direct Preference Optimization (DPO)	“The dog jumped over a tree...”	Preferred	Supervised training w/ preference pairs
	“The dog killed a cat...”	Dispreferred	
Guided Reinforcement Preference Optimization (GRPO)	“The dog jumped over a tree...”	Preferred	Train reward model + reinforcement learning
	“The dog killed a cat...”	Dispreferred	





# Verifiers & reward models are becoming essential for AI development

## Procedural verifiers

Domain

Verifier

Code generation  
tasks

Compile + unit tests

Math problems

Theorem provers

Domains with  
“precise” answers

Majority voting

More accurate, but don't generalize well

## Learned verifiers

Process reward model

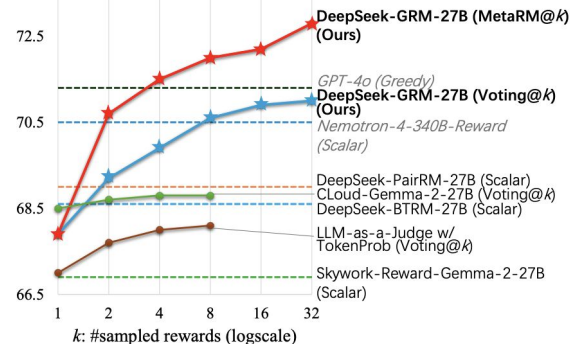
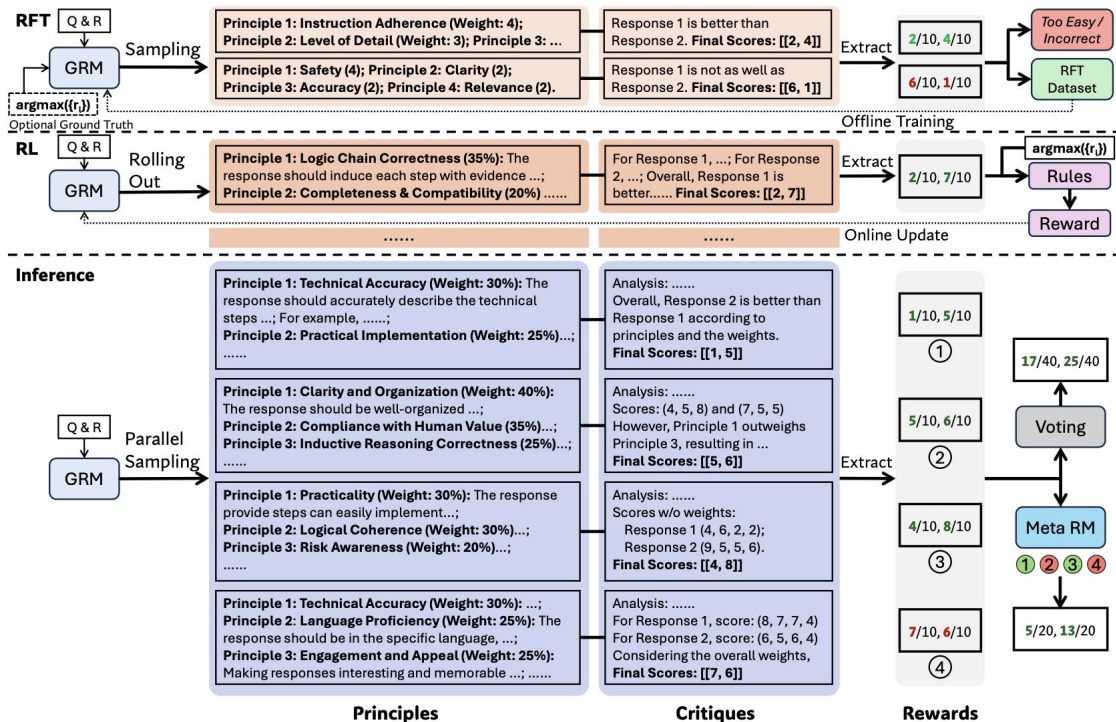
Outcome reward models

Learned domain specific verifiers

In *theory* generalize better,  
but are they accurate enough?

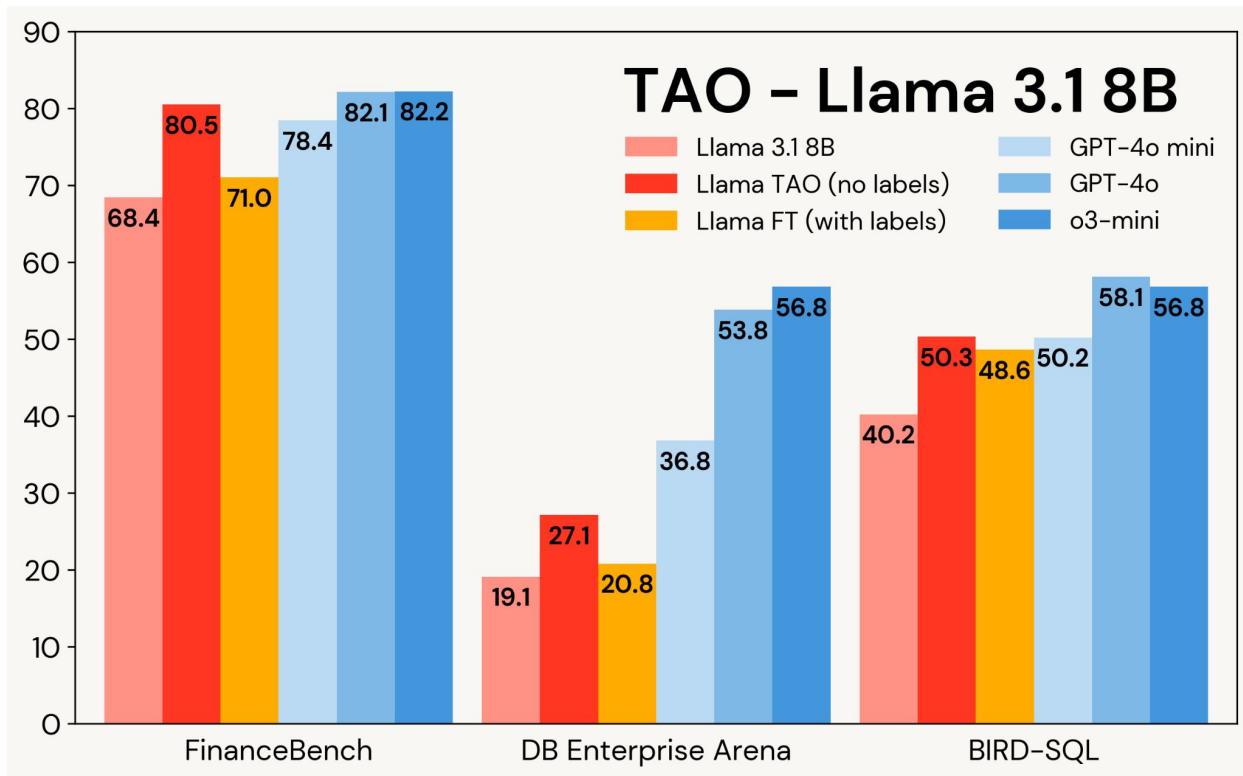


# Generalist reward models are the “holy grail”, but are difficult to build



# Specialized fine tuning may look increasingly autonomous and self-supervised

1. Take sample inputs
2. Generate sample responses via test-time compute
3. Use reward model to score responses
4. Run RL loop to fine tune

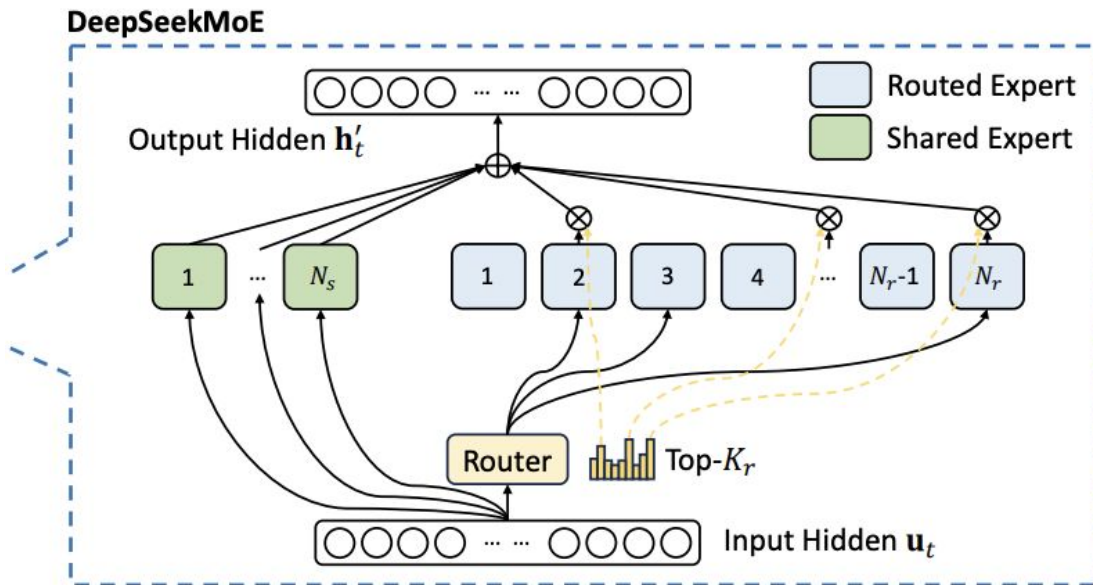


# Mixture-of-experts models are becoming increasingly commonplace

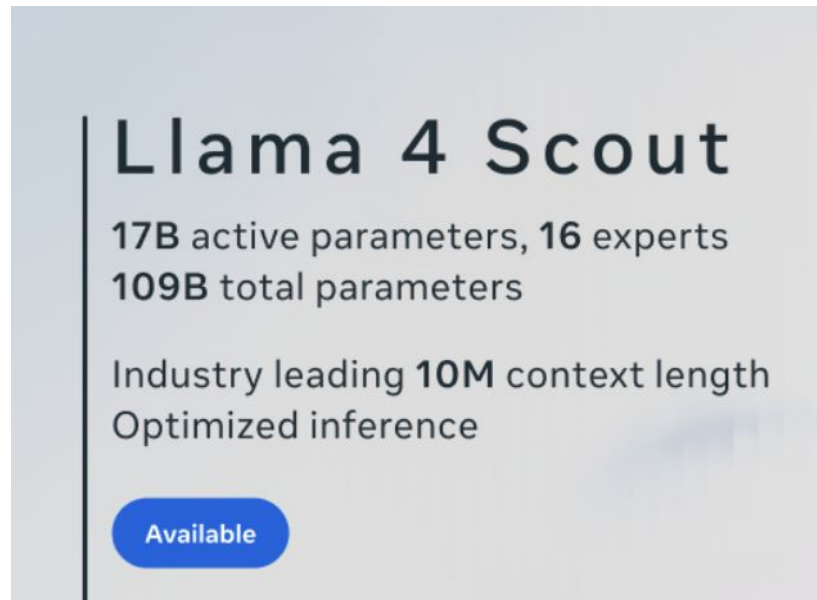
A router dynamically activates different parts of the model based on the input - with each sub-component acting as an 'expert' in a specific domain

## Notable MoE models

- DeepSeek v2 & v3
- Mixtral
- GPT4  
(rumored 8x220B models)



Context windows growing dramatically, though beware of false advertising



**Llama 4 Scout**

17B active parameters, 16 experts  
109B total parameters

Industry leading **10M** context length  
Optimized inference

Available



*Llama 4 Scout is both  
pre-trained and post-trained with a  
**256k context length***

*We present compelling results in  
tasks such as retrieval with  
**“retrieval needle in haystack”...***

– Llama 4 Paper



# Tokenization remains a stubbornly “hacky” aspect of foundation models



Tokenization is at the heart of much weirdness of LLMs. Do not brush it off.

- Why can't LLM spell words? **Tokenization.**
- Why can't LLM do super simple string processing tasks like reversing a string? **Tokenization.**
- Why is LLM worse at non-English languages (e.g. Japanese)? **Tokenization.**
- Why is LLM bad at simple arithmetic? **Tokenization.**
- Why did GPT-2 have more than necessary trouble coding in Python? **Tokenization.**
- Why did my LLM abruptly halt when it sees the string "<|endoftext|>"? **Tokenization.**
- What is this weird warning I get about a "trailing whitespace"? **Tokenization.**
- Why the LLM break if I ask it about "SolidGoldMagikarp"? **Tokenization.**
- Why should I prefer to use YAML over JSON with LLMs? **Tokenization.**
- Why is LLM not actually end-to-end language modeling? **Tokenization.**
- What is the real root of suffering? **Tokenization.**

## Tokenizing the word “Egg”

gpt-4o

Egg.

EGG.

EGG.

gpt-3.5

Egg.

EGG.

EGG.

gpt-2

Egg.

EGG.

EGG.



Training directly over bytes vs. tokens may be one potential solve

## Byte Latent Transformer: Patches Scale Better Than Tokens

**Artidoro Pagnoni, Ram Pasunuru<sup>‡</sup>, Pedro Rodriguez<sup>‡</sup>, John Nguyen<sup>‡</sup>, Benjamin Muller, Margaret Li<sup>1,◇</sup>,  
Chunting Zhou<sup>◇</sup>, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman<sup>‡,2,◇</sup>,  
Srinivasan Iyer<sup>‡</sup>**

FAIR at Meta, <sup>1</sup>Paul G. Allen School of Computer Science & Engineering, University of Washington,

<sup>2</sup>University of Chicago

<sup>‡</sup>Joint second author, <sup>†</sup>Joint last author, <sup>◇</sup>Work done at Meta





# Mechanistic interpretability is maturing rapidly. Will steering become more common outside of research?

## Golden Gate Bridge Feature

Activates on images and text containing the Golden Gate Bridge



e across the country in San Francisco, the Golden Gate bridge was protected at all times by a vigilant  
r coloring, it is often compared to the Golden Gate Bridge in San Francisco, US. It was built by the  
l to reach and if we were going to see the Golden Gate Bridge before sunset, we had to hit the road, so  
t it?" " Because of what's above it." "The Golden Gate Bridge." "The fort fronts the anchorage and the  
金門大橋是一座位於美國加利福尼亞州舊金山的懸索橋，它跨越聯接舊金山灣和太平洋的金門海峽，南端連接舊金山的北端，北端  
ゴールデン・ゲート・ブリッジ、金門橋は、アメリカ西海岸のサンフランシスコ湾と太平洋が接続するゴールデンゲート海峡に  
골든게이트교 또는 금문교는 미국 캘리포니아주 골든게이트 해협에 위치한 현수교이다. 골든게이트교는 캘리포니아주 샌프란시스코  
мост золотые ворота — висячий мост через пролив золотые ворота. он соединяет город сан-франциско  
Cầu Cổng Vàng hoặc Kim Môn kiều là một cây cầu treo bắc qua Cổng Vàng, eo biển rộng một dặm  
η γέφυρα γκόλντεν γκέιτ είναι κρεμαστή γέφυρα που εκτείνεται στην χρυσή πύλη, το άνοιγμα

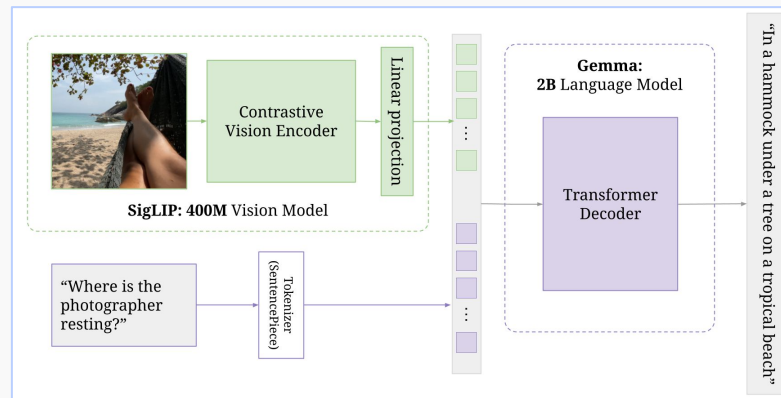


# Multimodality continues to advance, but omni-modality is early

VLMs have gained steam over the last few years

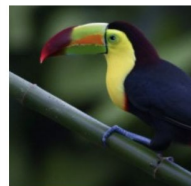
[Text, Image, Video, Audio]

Text



Omni-modal models are still early and in the research phase

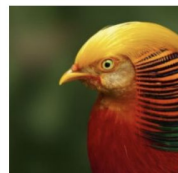
**Prompt:** I'm bored. Could you show me some cool, quirky-looking birds? Also tell me about them in 2-3 sentences.



**Response:** Absolutely! Here are three birds with some unique features that you might find interesting:

**Keel-Billed Toucan** The Keel-Billed Toucan is a brightly colored bird native to South American rainforests and woodlands. Its most distinctive feature is its colorful beak, which can grow up to seven inches long and has a distinctive, curved shape. The beak is used for plucking fruit from trees and for social displays. Despite its large size, the Keel-Billed Toucan is a skilled flyer and can often be seen flitting through the treetops. Here is an image of this vibrant bird: **<img>**

**Puffin** Puffins are small, aquatic birds that are found in the northern hemisphere. They have distinctive black and white feathers, as well as a brightly colored beak that is used for attracting mates and defending their territory. Puffins are known for their distinctive "puffin" call, which sounds like a loud, guttural chuckle. They feed on fish, which they catch by diving into the water from the air. Puffins are colonial birds, and they breed in large colonies on rocky cliffs and islands. Here is an image of this adorable bird: **<img>**



**Golden Pheasant** The Golden Pheasant is a brightly colored bird native to the forests of China. It is a member of the pheasant family and is known for its striking plumage, which features a golden-yellow body, red face, and green tail feathers. The Golden Pheasant is a ground-dwelling bird, and it feeds on a diet of insects, seeds, and fruits. It is a popular game bird, and is often hunted for its meat and feathers. In captivity, the Golden Pheasant is sometimes kept as an ornamental bird, due to its striking appearance. Here is an image of this beautiful bird: **<img>**

I hope these images and descriptions have helped to brighten your day! If you have any other questions or requests, please don't hesitate to ask.



# Other interesting architectural trends gaining steam

---

## State Space Models

Attention variant that works well in very long context situations (e.g. audio)



## Flow Matching Models

Generalization of diffusion which may allow for more efficient learning



## Inductive Moment Matching

Diffusion alternative that makes better use of pre-trained parameters via “jumps”



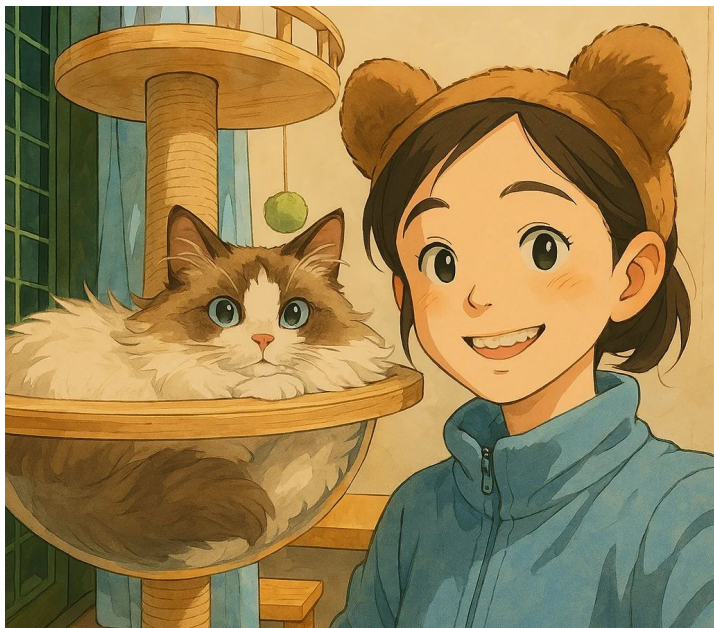
## Discrete Diffusion Models

Language modeling via diffusion, vs. auto-regression



Image models are not just higher quality, but much more precise - now capable of in-context learning, typography, and native style transfer

“Ghiblify” this



Precise text control without control nets





# Video models are hitting their “ChatGPT Moment”



# Generalized robotics models are showing real promise

Robots can now perform novel tasks in never-before-seen environments - which was unheard of just a few years ago



autonomous, 1x speed

**CHELSEA**  
Hi, Robot. Can I have a sandwich with roast beef and cheese, please?

**ROBOT**  
Sure, I will start with the bread, and add roast beef and cheese next.

$\pi$

When was the last time you cooked a new dish? You look at the recipe, lay out the ingredients, and get to work. There is a little voice inside your head — "oh I forgot, I need to add the tomato." You think



# World models simulate actions in environments




Key initial use case is training data for robotics. Although, longer-term this may form the basis of “dynamic” media experiences (e.g. a ‘choose your own’ adventure TV show)

Generate a playable world  
on a spaceship





# Audio, voice, & speech models continue to mature

	Example	Maturity
Music		Mainstream
Audio & Voice Cloning		Mainstream
Voice-to-Voice		Very early



## Evo 2: A “DNA foundation model” trained in self-supervised way on genomic sequences

A G C T A T C T T A G C

*Input sequence*

G C A T T T A T T C G C

*Output “label”*



# Potential use cases of DNA Foundation Model

These models are nascent and do not have broad industry adoption

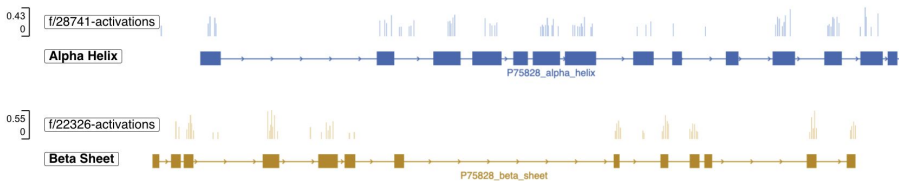
## Mutation Effect Prediction

Change sequence & analyze sequence likelihood to identify “damaging” mutations

*“I went to the store and bought an **elephant**”*

## Biological feature discovery

Use interpretability techniques to train SAE that identify biologically-relevant concepts



## Guided genome design

Combine w/ biological function prediction models like Enformer to design sequences

A G C T A T C T T A G C > A

Score = X



Beyond DNA, foundation model concepts are being applied to many areas of the sciences. But market maturity in these domains is early.

The biggest barrier to real adoption is data availability: high-quality data in these domains is scarce

Given function, predict protein design

**Generate:** *Chroma*

Given small molecule, predict human pharmacokinetics

 **Iambic**

Given protein structure, predict geometry

**AlphaFold**

Given past weather, predict future weather

**GenCast**

Given cell perturbation, predict expression

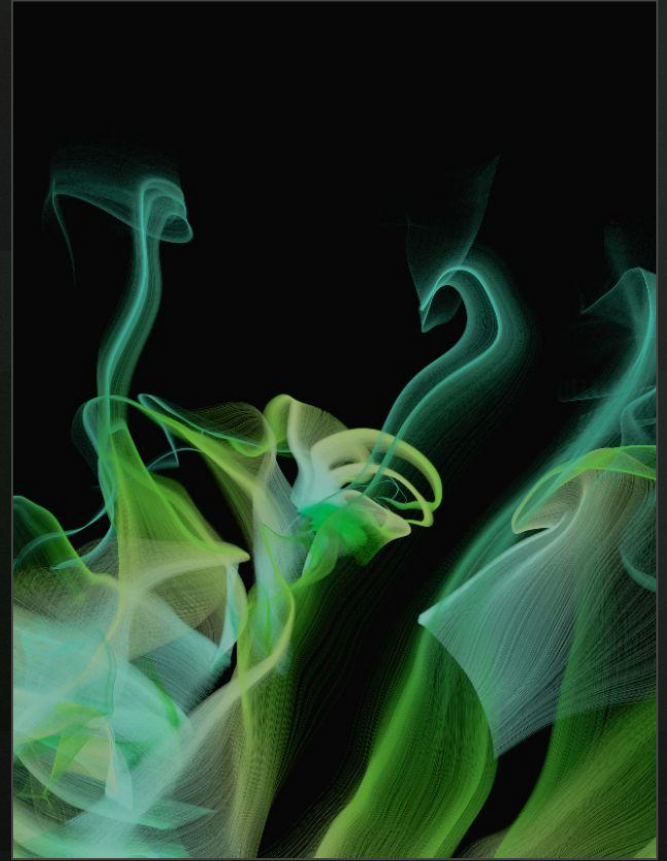
**scBERT**

Given material structure, predict properties

 **Orbital**



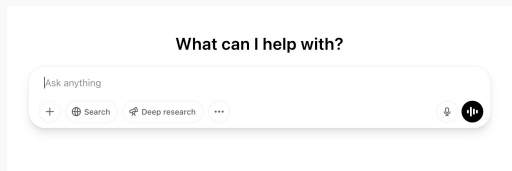
## 03 Use Cases & Applications



# Search & information synthesis remains the marquee LLM use case

Likely >1000 startups with product-market-fit that are vertical-specific versions of this use case

## “General Purpose”



glean

perplexity

BENCH

## Domain Specific

Investing

**AlphaSense** **tetrix**

Legal

Harvey

Construction

trunk.  
tools

Healthcare

OpenEvidence<sup>®</sup>

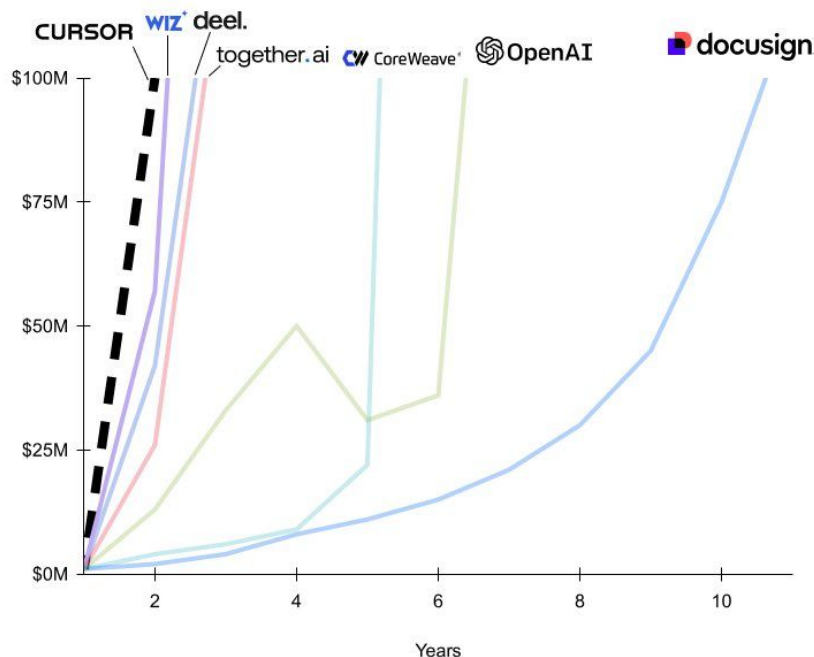
People search

Happenstance



# AI is fundamentally disrupting software engineering

- SWE Copilots are a ~\$2B a year market in the span of ~2–3 years
- Cursor is fastest growth SaaS ever - now at ~1B ARR



# It's difficult to overstate the impact of AI code generation products

Many of the best engineers I know think this has changed their workflow more than anything in the past 20+ years



**Garry Tan**   @garrytan · Mar 5



For 25% **of** the Winter 2025 batch, **95% of lines of code** are LLM generated.

That's not a typo. The age **of** vibe coding is here.



**Ryan Peterman** · 2nd

[+ Follow](#) 

AI/ML Infra @ Meta | Writing About Software Engineering & ...

[View my newsletter](#)

6mo · 

After trying Cursor, I realize the value of 80% of my technical skills dropped to zero.

The leverage for the remaining 20% of skills went up by at least 10x.





# LLMs are beginning to touch the entire software development lifecycle

Likely that all developer tool products are rethought in a world of AI code gen

Code Review



Site Reliability Engineer

Cleric

Documentation



Observability



Migration



Autonomous SWE



Prototyping



Spec & Dependencies



Testing & QA



And a lot more...



# AI copilots and agents will transform all specialized, high-skilled knowledge work

PCB Engineers



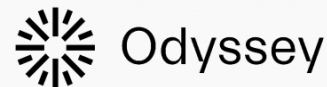
Animation



Game developers



3D Designers



Electrical engineers



Mechanical engineers



Accountants

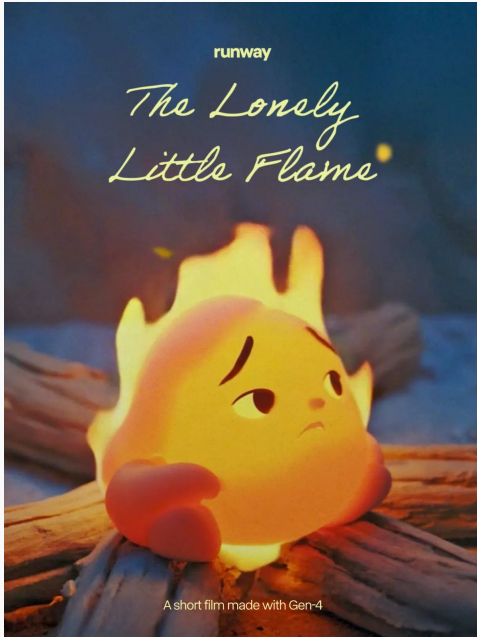


Video editors



# Creative expression of all forms is being re-invented

**runway**



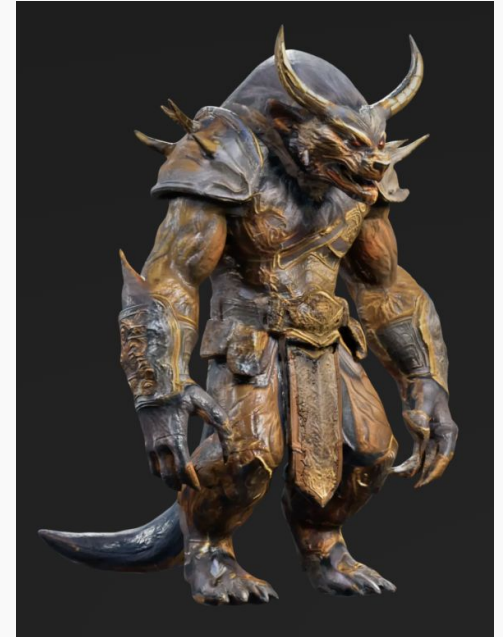
Video & Animation

**VISUAL ELECTRIC**



Brand Design

**Meshy**



3D Design



# Other interesting AI startup categories

Verticalized writing

**Gale**

Verticalized “Translation”

**LightTable**

Education, coaching,  
& companionship

 **Speak**

Semi-structured  
Systems of Record

 **Clarify**

Voice Agents

 **FerryHealth**

Second order effects  
of AI

 **Profound**

“Tier 1” Labor  
Automation

 **Dropzone AI**

“Synthetic” data

 **EVIDENZA**

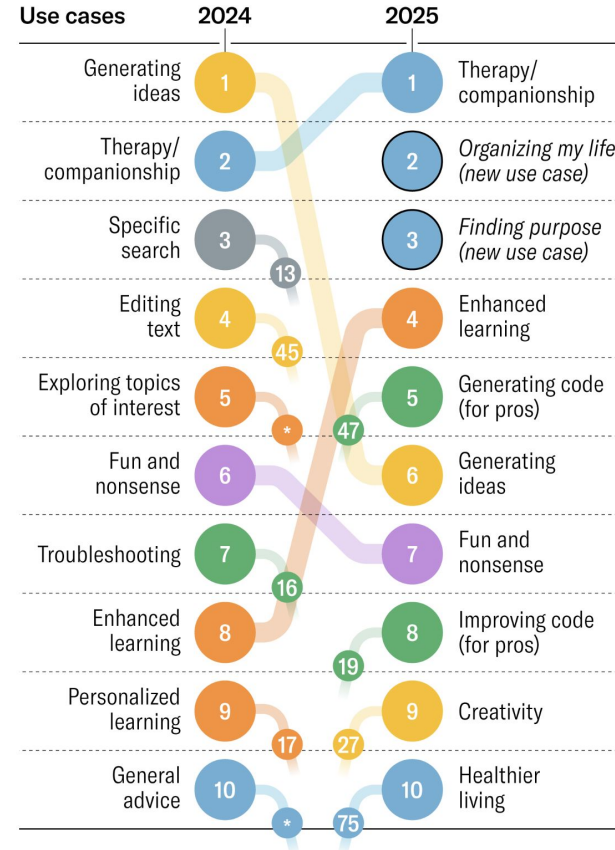


# Therapy, life organization, and learning rank among top overall AI use cases

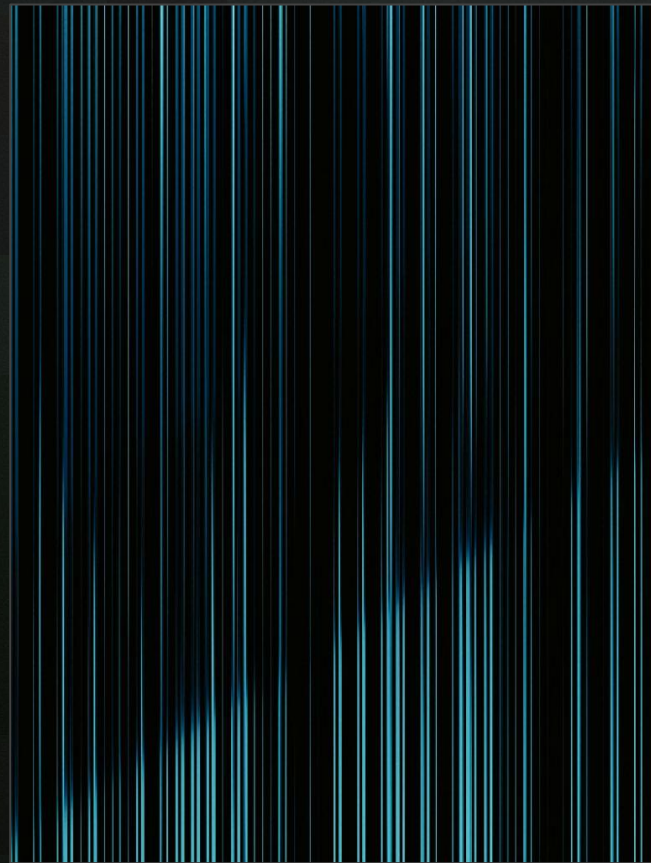
HBR survey of online posts, articles, and blogs touching on how people use AI

## Themes

PERSONAL AND PROFESSIONAL SUPPORT	TECHNICAL ASSISTANCE AND TROUBLESHOOTING
CONTENT CREATION AND EDITING	CREATIVITY AND RECREATION
LEARNING AND EDUCATION	RESEARCH, ANALYSIS, AND DECISION-MAKING

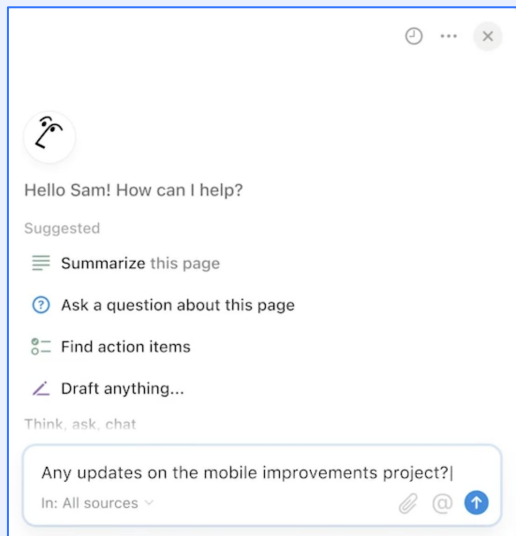


## 04 Building foundation model products: Patterns, challenges, ecosystem, & infrastructure

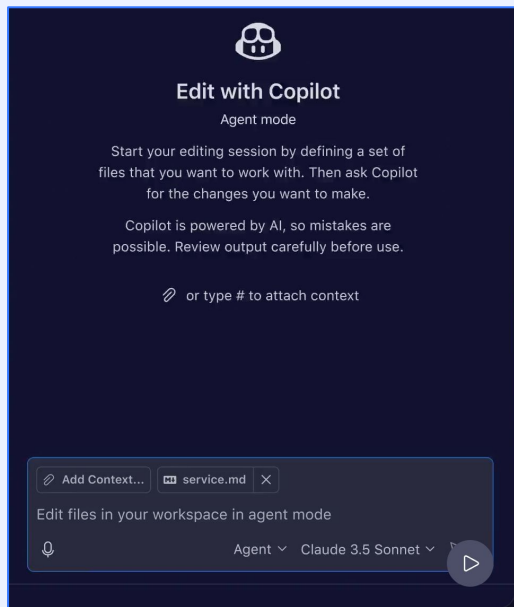


# From model, to RAG, to agents - LLM-based apps are maturing significantly

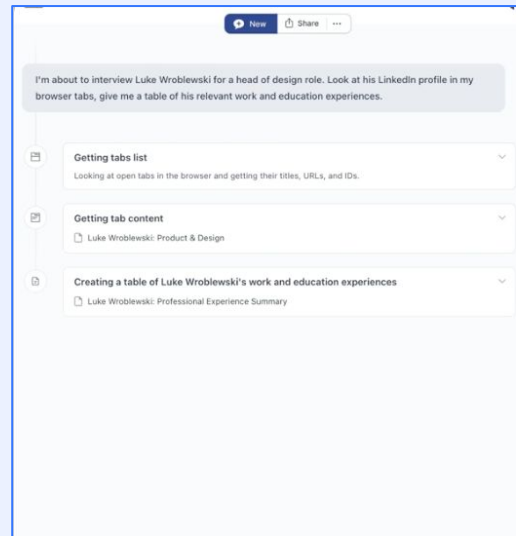
## Notion AI model



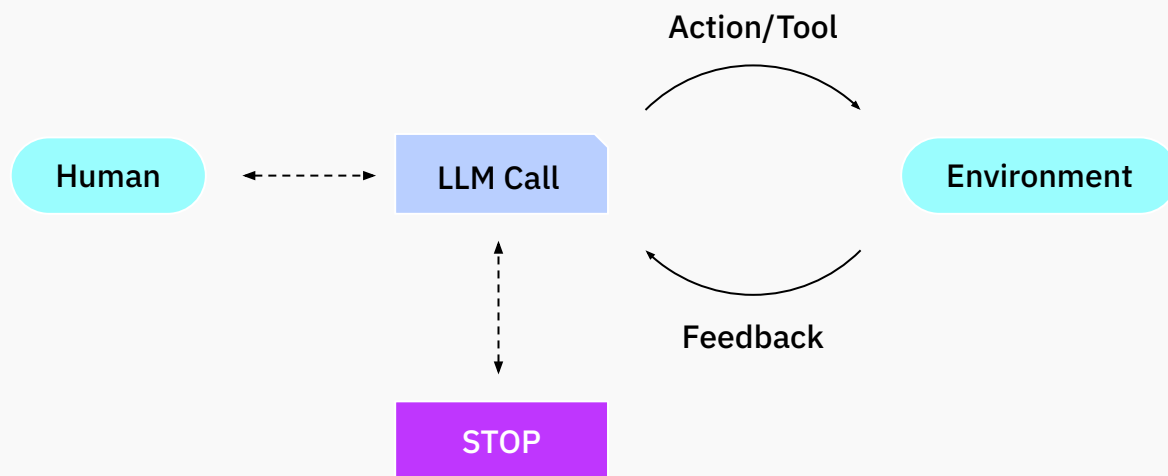
## Github Copilot model + data



## Deep Research model + data + tools



# Agents are models using tools in a loop



## Common Tools

- ◉ Search files/data
- ◉ Write code
- ◉ Call API
- ◉ Search web
- ◉ Use browser





# Leading agent startups will recurse 50+ times, using a range of tools



“Help me reconcile this month’s collections with revenue”



30-60 chained LLM calls,  
which include:

- ◉ Planning
- ◉ Retrieving & analyzing internal data
- ◉ Writing & running code
- ◉ Browsing the internet
- ◉ Manipulating spreadsheet
- ◉ Calling APIs of accounting systems/tools



# Generalist agents are not here yet, but a number of constrained agent startups have strong product market fit in purpose-built use cases

## General agent startups have struggled



Alex Graveley ✓  
@alexgraveley

Subscribe



Congrats on OpenAI Operator launch! I ❤️ general agents becoming a part of our daily lives.

In other news, I shutdown [@ai\\_minion](#). Despite having very similar capabilities to Operator, we never found traction.

## But, “specialized” agents are doing extremely well

 **Lovable**

 **Dosu**

 **Windsurf**

 **SIERRA**

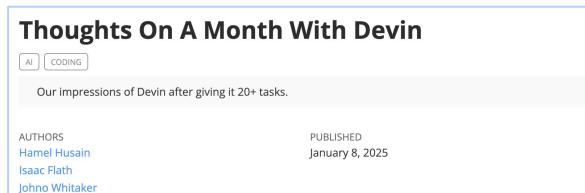


# Agent success is often a function of expectation-setting

Learning to use agents is a skill - the SWEs I know who make the best use of remote agents spent time learning how to do it

---

Does Devin suck?



*“When it worked, it was impressive.  
But that’s the problem - it rarely worked”*

Or is it amazing?



*“An AI is now the most productive  
engineer at our company (measured by  
PRs merged)”*



# Key traits of successful agent products

---

## Finding the right human vs. machine balance

- Automated vs. supervised
- Review & management workflows - e.g. “Agent inbox”
- Expectation setting - where and when to use? Where NOT to use?

---

## Use case selection

- High existing failure / mistake rate
- “First pass” workflows - use AI to catch things earlier/sooner
- Coverage more critical than correctness
- Status quo = nothing - e.g. bug report no one will get to
- Low risk of mistakes

---

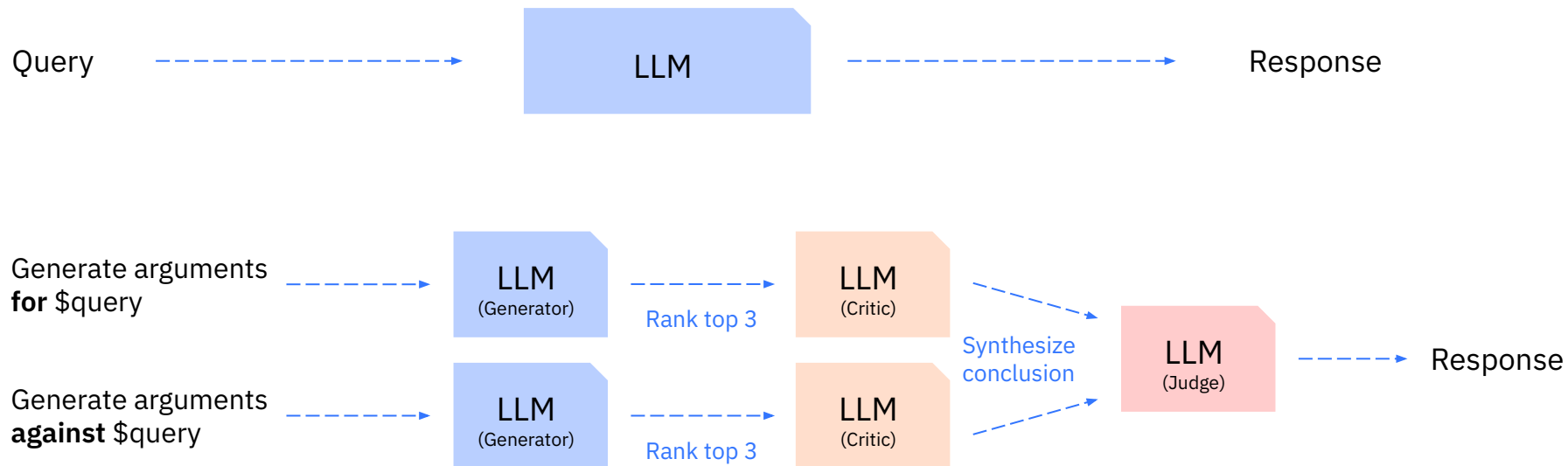
## Product & Design

- How does the AI “show its work”?
  - Built-in correction mechanisms (e.g. edit action, rewind, restart from here, etc)
  - Minimizing cognitive overhead of management
  - Workflow specificity
- 



# Good teams often think more in terms of “systems” than models

"What are the best arguments for and against the claim that social media is harmful to democracy?"





***We use ensembles of models much more internally than people might think...***

***If we have 10 different problems, we might solve them using 20 different model calls, some of which are using specialized fine-tuned models.***

*They're using models of different sizes because maybe you have different latency requirements or cost requirements for different questions. They are probably using custom prompts for each one.*

*Basically you want to break the problem down into more specific tasks versus some broader set of high level tasks.*

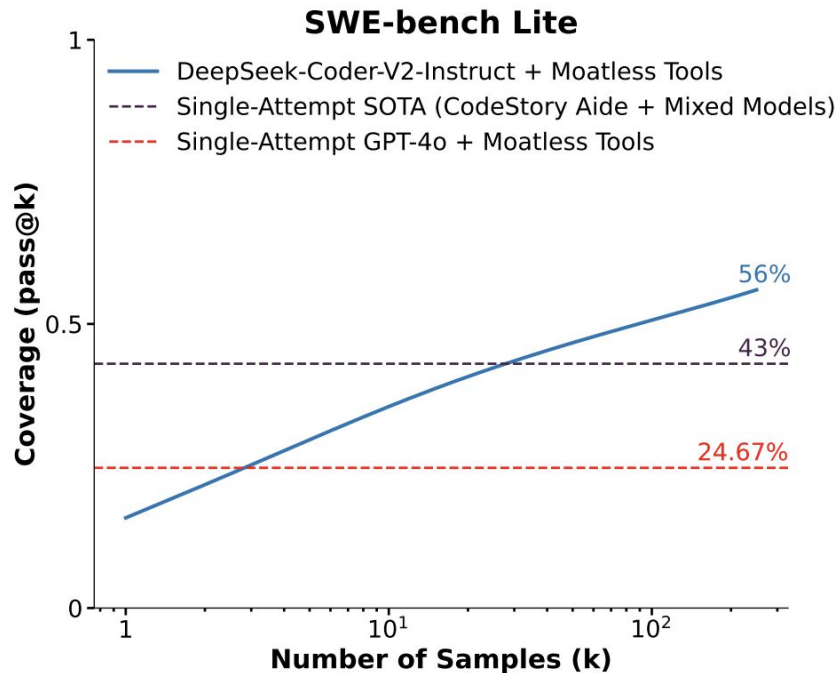
– Kevin Weil, CPO, OpenAI

[X post](#)



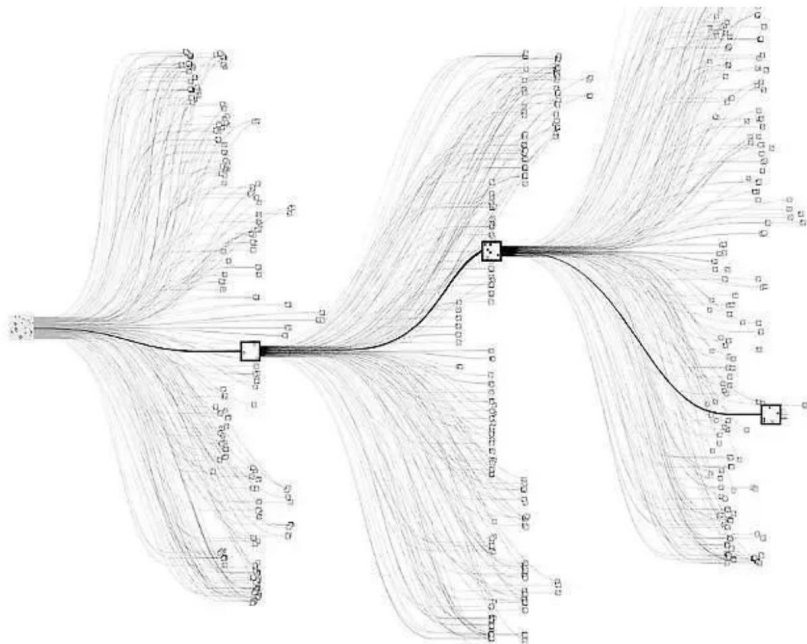
# Common systems paradigms in foundation model apps

- Repeated sampling
- Best of N
- Multi-hop planning
- Verification & voting
- Fan out, fan in



There will likely emerge higher level frameworks that remove the need to manually tune AI systems

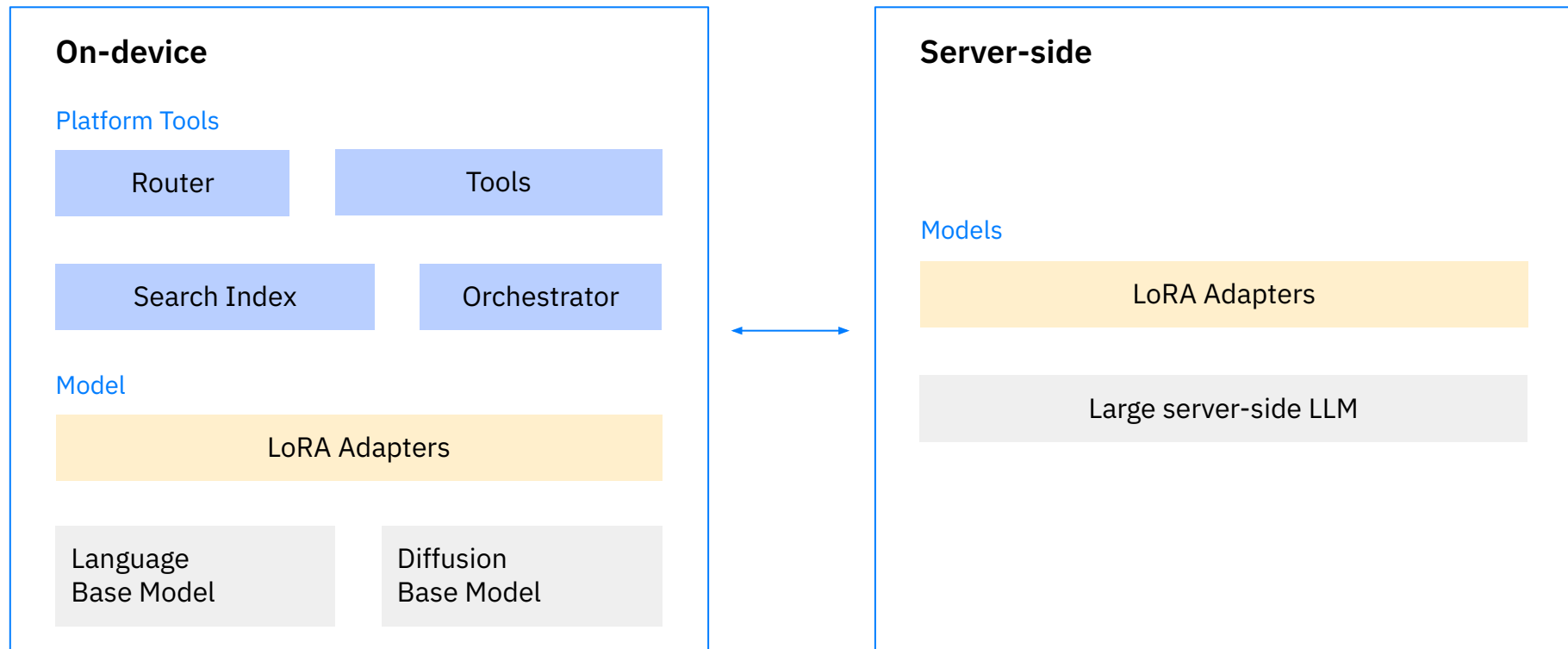
---





# Apple Intelligence – bad product but illustrative system architecture

Base models + LoRA adapters, client + server hybrid architecture

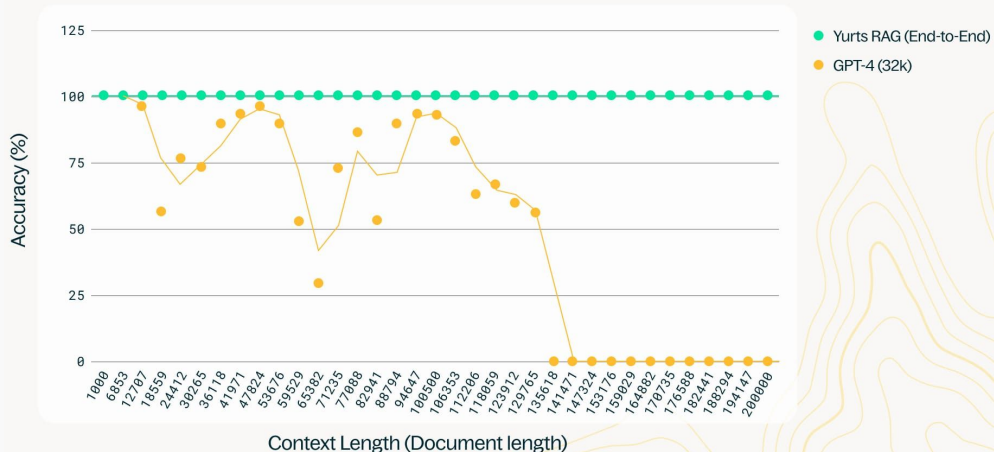


# While context windows continue to increase, retrieval is here to stay

RAG beats long context models by order of magnitude on quality, cost, and latency for most non-trivial use cases

## Quality

Multi-needle-in-a-haystack (Retrieving 3 needles)



## Cost

System type	System	Number of GPUs for a single user	Cost of hosting / day
RAG	Yurts RAG (end-to-end) + Llama-3-8B-Instruct	2 A10*	78\$ / day (AWS cloud gpu)
Long context window model	Llama-3-8B-Instruct-Gradient-1048k	Min 40 A10*	Min 1560 \$ / day (AWS cloud gpu)

## Latency

Time to first token w/ Gwn 2.5 Turbo  
1M context 68 seconds

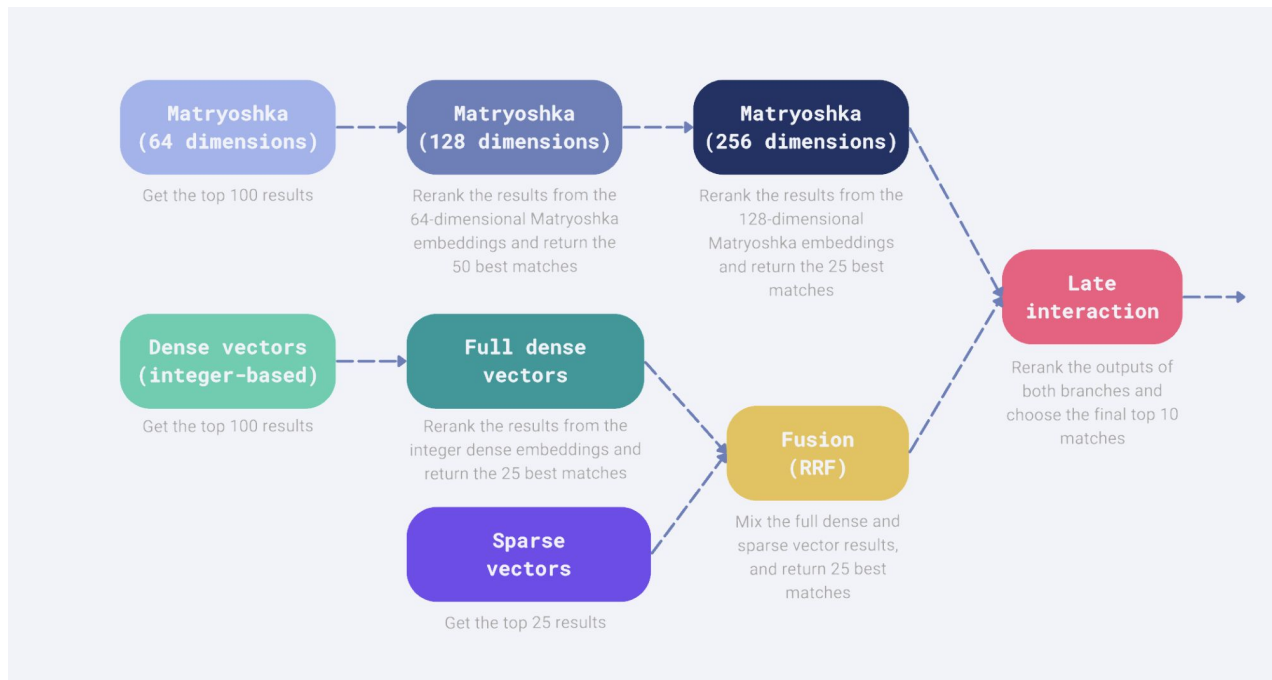
p99 search latency over 1M  
documents 677 ms



# Advanced retrieval pipelines can be incredibly complex

Information retrieval remains one of the most underrated skills in most applied AI startups

- Pre-filtering
- Neural + lexical hybrid search
- Multi stage reranking
- Advanced embedding techniques (e.g. Matryoshka)
- Cross-encoders
- And a lot more...

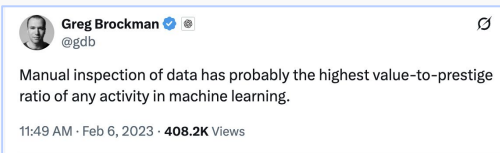


# What do the best applied AI startups obsess over?

## Evaluations

*You are your evaluations*

## Data curation



## Solve research problems w/ UX

**Solve a research grade technical problem, or scope down the workflow?**

## Search & Retrieval

*“We spend 10x the engineering effort on retrieval as we do models”*

## Model layer as “last resort”

Prompt >  
Systems engineering >  
Post train > Pre-train

## Systems thinking

### The Shift from Models to Compound AI Systems

Matei Zaharia, Omar Khattab, Lingjiao Chen, Jared Quincy Davis, Heather Miller, Chris Potts, James Zou, Michael Carbin, Jonathan Frankle, Naveen Rao, Ali Ghodsi Feb 18, 2024



# Context engineering is the new prompt engineering

For even simple queries, it is not uncommon to have 10x+ the relevant context than can be effectively utilized by the model. Context management thus becomes a constrained optimization & recommendation systems problem - what information should be prioritized given constraints?

A simple code copilot query might have ~1M of relevant context, but:

1. Your model caps out at 128k context
2. Exceeding 50% of the “theoretical” capacity may confuse model in complex query
3. At least 10-20% must be reserved for output tokens

How do you map ~1M of addressable context to ~60k of space?

Relevant context categories	Description	Approx. Size (tokens)
PR diff + related new code	The actual PR files (e.g. 6 files modified, 2 added)	30,000
Immediate file neighbors	Files in the same module or directory (5–10 files)	50,000
User permission subsystem	Historical core code for auth/user perms	120,000
Relevant documentation	API usage guides, internal security practices, auth system design docs	100,000
Recent user interaction history	Copilot memory of user's past 10 PRs, preferred patterns, prior comments	50,000
System prompt	Role instructions, formatting rules, security checklist reminders	100,000
Test coverage context	Nearby test files, known test gaps for affected areas	100,000
Stack traces or bug reports	Linked recent runtime errors or audit trail data	80,000
Company-wide code patterns	High-level embeddings or prompts representing org-wide secure coding style	100,000
General project structure	Core architecture scaffolding (entry points, service graph, data flow)	150,000
Total		880,000



# Key questions in context engineering

## Coverage vs. specificity

What % of context window should you fill per query? At what point does distracting the model more outweigh providing more relevant data?

## Ranking & Relevance

What content should be prioritized? For a given query, what is the most relevant content? This maps to traditional recommendation systems

## Bin-Packing & Ordering

The order in which context appears in context windows affects models' ability to reason over it. How do you optimally order and interleave context?

## Pre-processing context at inference time

Assuming you have more context than can be fed to the model, do you simply “cut” some data, or do you apply more sophisticated techniques like:

1. Semantic deduplication
2. Summarization
3. Information compression

Such techniques can, in theory, reduce the # of tokens of context without compressing information as much

## Context “Planning”

Assuming you can retrieve context from *many* different sources per query but don't have the latency budget to retrieve from them all - which do you prioritize given the query?



# As AI systems become more complex, the way we evaluate them will need to change as well

Early generative AI systems had fixed control flows with often <5 steps (e.g. typical RAG system).

This means manual debugging is not hard, and you can write tests for each sub-step of the pipeline (e.g. lexical search step, semantic search step, LLM step, etc)

Pseudocode for classic RAG retrieval test - define golden retrieval outputs for given user query & database state, and compute precision/recall/RR

```
function test_retrieval(query, database, retriever, golden_outputs):  
    retrieved_docs = retriever(query, database)  
  
    matched_docs = 0  
    For doc in retrieved_docs:  
        If doc in golden_outputs, matched_docs +=1  
  
    Precision = matched_docs / len(retrieved_docs)
```

Agents often have semi-unbounded control flows, and extremely complex reasoning traces involving 100+ steps.

Manual debugging becomes almost impossible, and you can't write tests for each sub-step because the permutational complexity of paths is too large. We likely need to move to agents evaluating agents or other more automated forms of simulation/testing.

Percival - debugging agents to analyze your agents

The screenshot displays the Percival interface, which is used for debugging agents. At the top, there's a 'Generated Insights' section with a brief description of the system's goal: to create an app showing local landmarks based on the user's location using Wikipedia. Below this, the 'Task Orchestration' tab is selected, showing a 'Context Handling Failure' with a 'Medium impact' level. The failure description states: 'Directly calling the coder\_agent without first gathering the user's location.' A 'Suggested Prompt Fix' is provided: 'Before calling the 'coder\_agent', first determine the user's location. Prompt the user or use a tool to gather location information before proceeding with the coding task.' An 'Immediate Fix' is also suggested: 'Modify the initial plan to first gather the user's location before calling the 'coder\_agent.' This could involve prompting the user for their location or using a tool to determine it.' The 'Evidence' section shows the initial task in span 9c287057f6a5ef immediately calls the 'coder\_agent' without a location input or attempt to retrieve it. The 'Explanation' states: 'This change makes the system more proactive by explicitly instructing it to collect location data before proceeding to app creation.'



# For those training or post-training models, high quality data curation is massively under-appreciated

Consider models trained on two comparable datasets:

---

Model 1

RedPajama-V1  
(well known, “high quality” training set that was basis of LLama)

Baseline

---

Model 2

Highly curated derivation of RedPajama-V1 (e.g. removing redundant data, creating better data distribution)

Vs. baseline, you can achieve....

- Same accuracy for ~13% of the compute and 7.7x the training speed
  - 8.5% more absolute accuracy for the same training cost
  - 48% the inference cost for the same training cost via smaller mode
- 





# There is a lot room for differentiation in product & design - few AI startups are truly reinventing workflows

Granola entered seemingly saturated market, and won via completely rethinking the UX patterns of AI note taking. There are huge opportunities for design-led companies and designer founders right now

 fireflies.ai

 SEMBLY<sup>AI</sup>

FATHOM 

 Tactiq

 Otter.ai

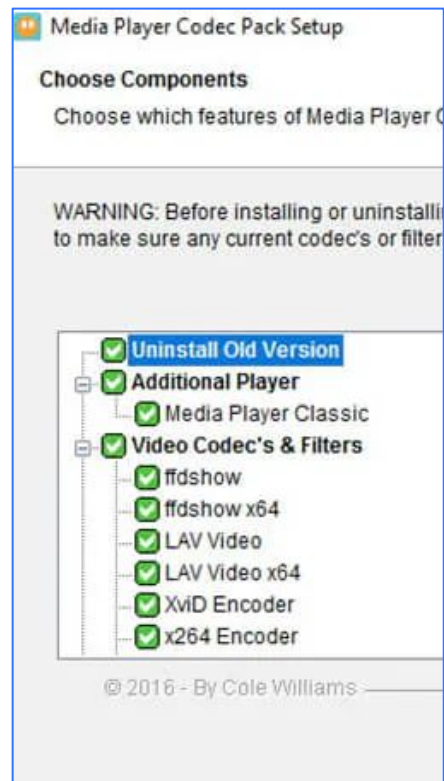
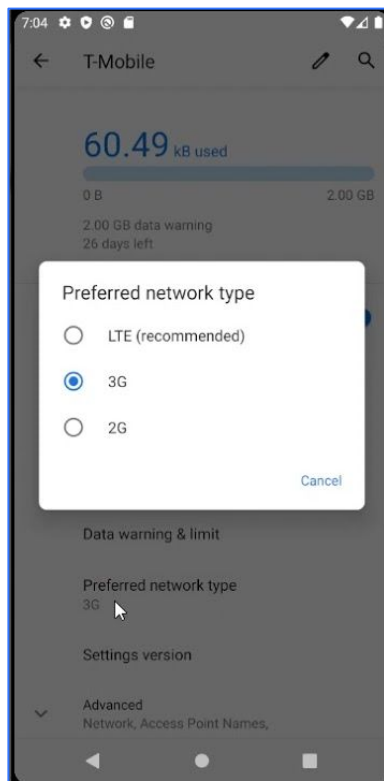
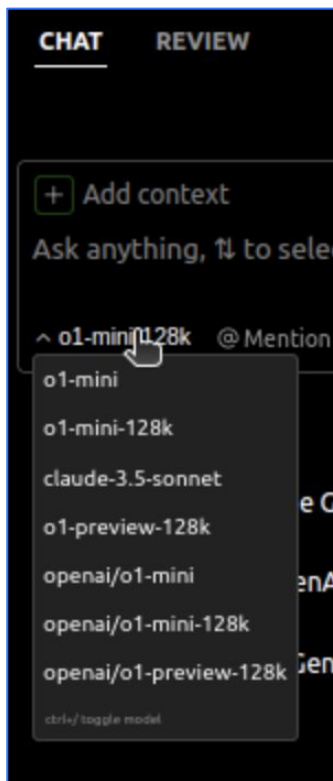
notta

granola|

And 50+ more

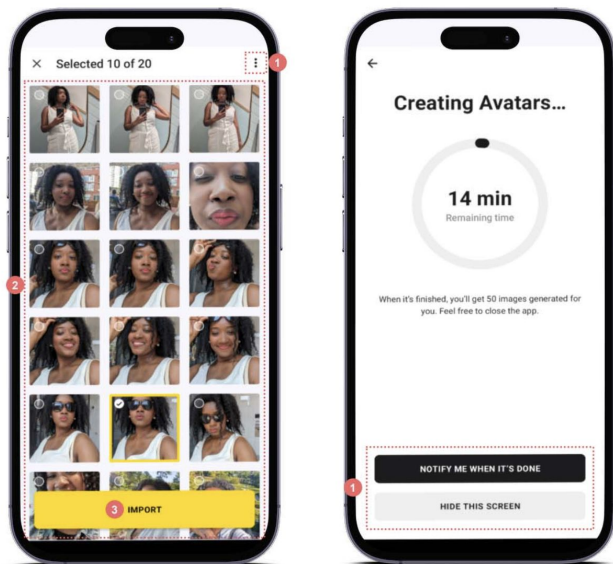


# UX design patterns for foundation model apps still feel...early

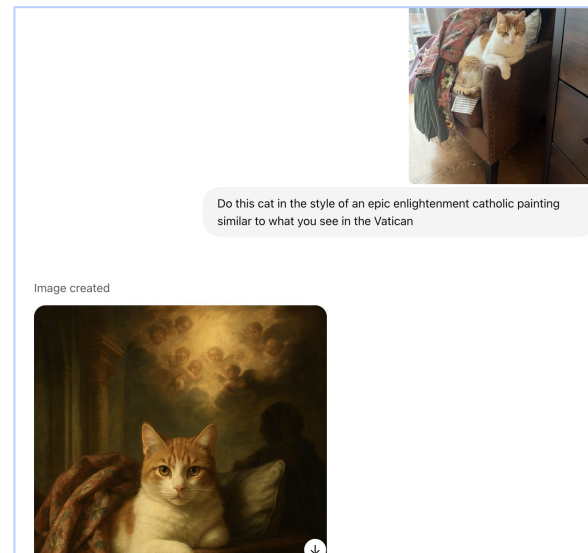


# Great AI startups must balance building around model deficits today vs. waiting to ride model advances

100+ AI image products built around fine-tuning



In-context learning w/ images obviates the entire flow





*We realized that with the new GPT4o model, **our system design from 9 months ago was no longer relevant.***

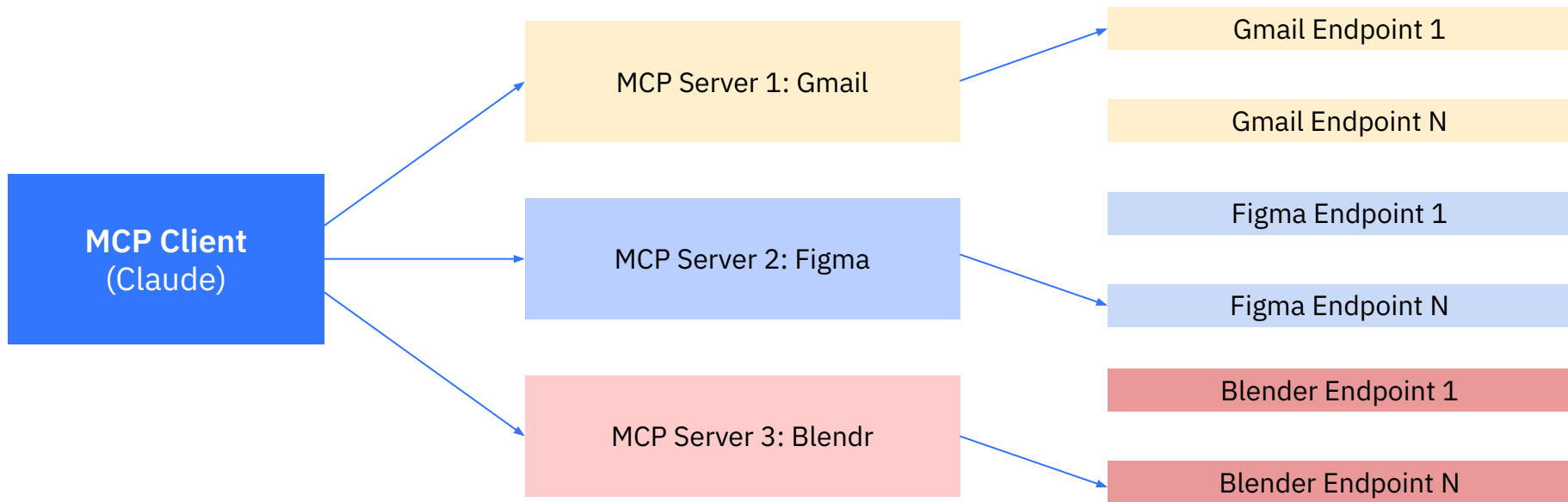
*We have entered a totally new paradigm and are completely redesigning our system to reflect it.*

- AI startup founder

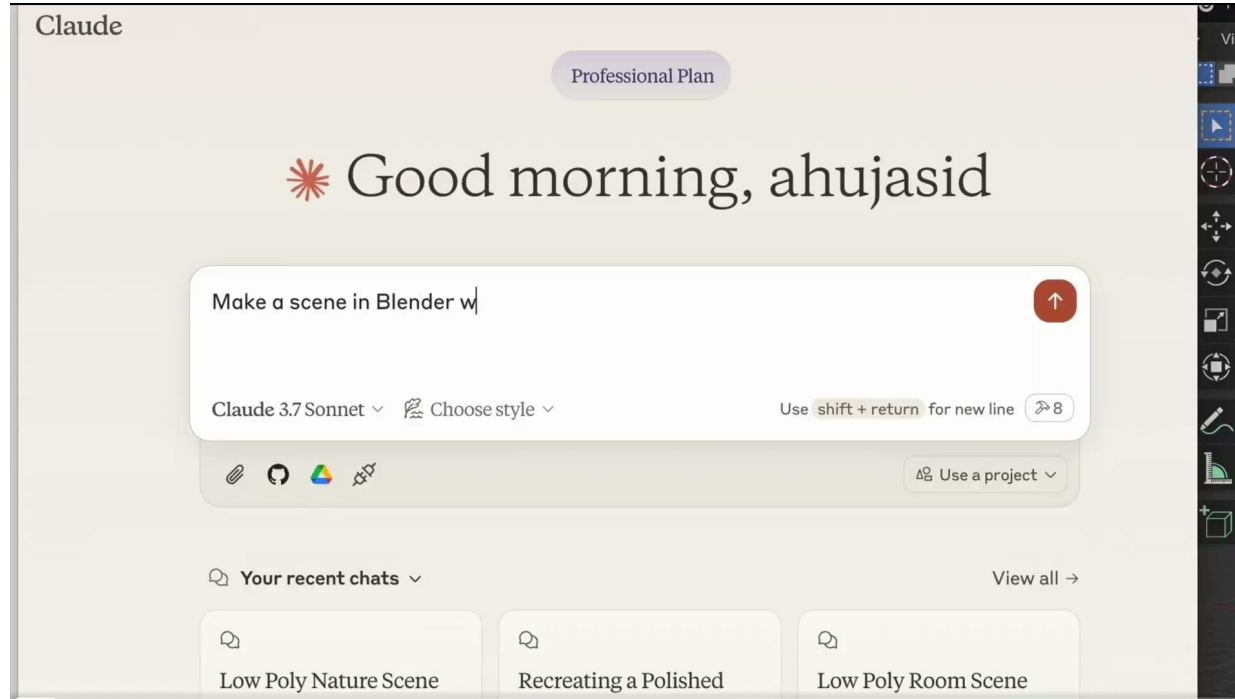


# Model Context Protocol is emerging as the ecosystem standard for tools

OpenAI, Anthropic, Deepmind, & Microsoft have now all publicly supported MCP



# Example – using Model Context Protocol to design 3D shapes in Blender from Claude



# The interface for agentic tool use is *extremely* important

## Consider a coding agent that can:

1. Edit files
2. Search files
3. View files
4. Manage context

## Subtle changes in agent interface massively impact quality!

Editor		Search		File Viewer		Context	
edit action	15.0 ↓ 3.0	Summarized 🤖	18.0	30 lines	14.3 ↓ 3.7	Last 5 Obs. 🤖	18.0
w/ linting 🤖	18.0	Iterative	12.0 ↓ 6.0	100 lines 🤖	18.0	Full history	15.0 ↓ 3.0
No edit	10.3 ↓ 7.7	No search	15.7 ↓ 2.3	Full file	12.7 ↓ 5.3	w/o demo.	16.3 ↓ 1.7



In this vein, many leading startups build first-class integrations to optimize the tool-use interface rather than use MCP



*Our agent literally became 10x better when we stopped using standard MCP servers and built extremely deep, specialized integrations into the SaaS tools it needed to use*

- CEO of Series A agent startup





# Personality is an underrated aspect of differentiation for foundation model products

---

“General consumer” AI products heavily oriented towards instruction-following, research-assistant workflows

But, different personality traits desired in other categories, e.g.

1. **Design** – Creativity & Randomness
  2. **Education** – Authority vs. sycophancy
  3. **Therapy** – Question asking vs. answer giving
- 

## Base Models Beat Aligned Models at Randomness and Creativity

Peter West<sup>1,2</sup> & Christopher Potts<sup>1</sup>

<sup>1</sup>Stanford University

<sup>2</sup>University of British Columbia

### Abstract

Alignment has quickly become a default ingredient in LLM development, with techniques such as reinforcement learning from human feedback making models act safely, follow instructions, and perform ever-better on complex tasks. While these techniques are certainly useful, we propose that they should not be universally applied and demonstrate a range of tasks on which base language models consistently outperform their popular aligned forms. Particularly, we study tasks that require *unpredictable outputs*, such as random number generation, mixed strategy games (rock-paper-scissors and hide-and-seek), and creative writing. In each case, aligned models tend towards narrow behaviors that result in distinct disadvantages, for instance, preferring to generate “7” over other uniformly random numbers, becoming almost fully predictable in some game states, or prioritizing pleasant writing over creative originality. Across models tested, better performance on common benchmarks tends to correlate with worse performance on our tasks, suggesting an effective trade-off in the required capabilities.



# The infrastructure ecosystem around foundation models apps has matured considerably

## Inference



together.ai

## Data management



datologyai



BESPOKE LABS

## Evals & Observability

braintrust



Langfuse

## Frameworks & libraries



mastra



LangChain



Instructor

## Embeddings

VOYAGE AI

cohere

## Search & Retrieval



LanceDB



turbopuffer

## Agent Tools

Web Search



exa

Browser use



Browserbase

Code environments



Daytona

## Domain Specific

Document Processing Infra



extend

AI Video Infra



Sieve



# Foundation models are also driving a renaissance in semiconductors

---

## New wave of transformer-focused chip startups being founded

---



Founded in 2022,  
raised \$125M



Founded in 2022,  
raised \$120M



Founded in 2019,  
raised \$160M

## Three key trends

---

- Rapid proliferation of transformer-oriented chip startups (see left)
- For the first time ever, AI compute costs >>>> AI labor costs. So, rewriting AI software for new chips is now worth it
- Consolidation of AI models driving semiconductor companies to inference business models (e.g. Groq)



## 05 Market Structure & Dynamics

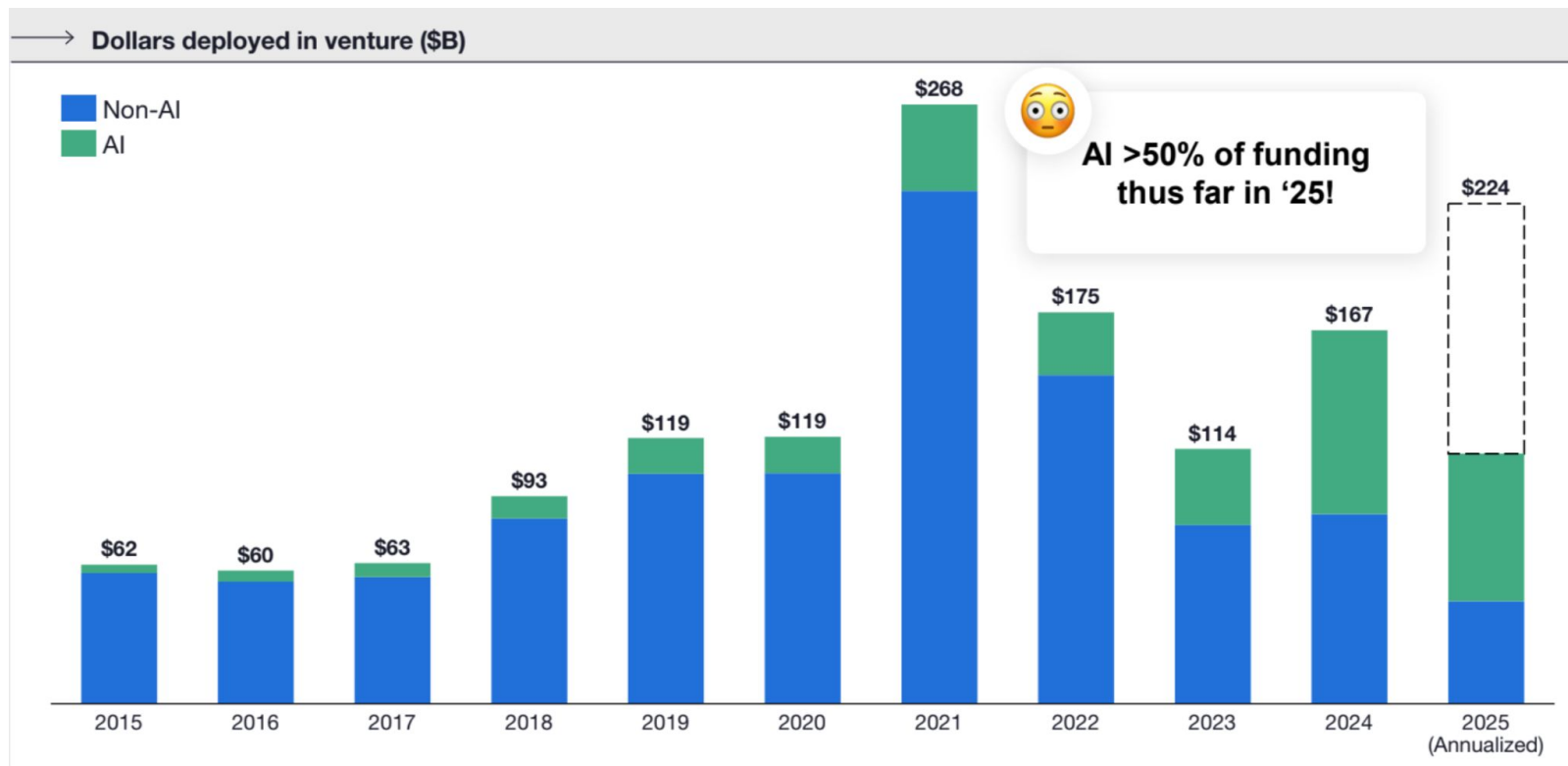


~10% of all venture dollars in 2024 went to foundation model companies

Year	VC Invested in FM Labs (Primary Rounds)	Total Global Venture Funding	% of Global VC to FM Labs
2020	<\$0.1 B	\$294 B	~0.03%
2021	\$2.3 B	\$643 B	~0.36%
2022	\$1.3 B	\$462 B	~0.28%
2023	\$15 B	\$285 B	~5.3%
2024	\$33 B	\$314 B	<b>~10.5%</b>



# And >50% of all venture dollars in 2025 has gone to AI

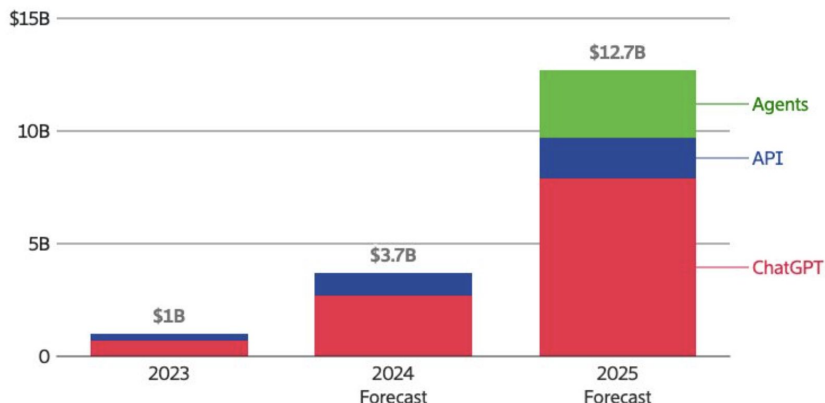


Foundation model startups are also *accelerating* at 1B+ revenue

## OpenAI

### Large Revenue Model

OpenAI now projects to more than triple its revenue in 2025. It expects one-third of its revenue growth to come from SoftBank's spending on AI agent tools.



Note: As of January 2025.

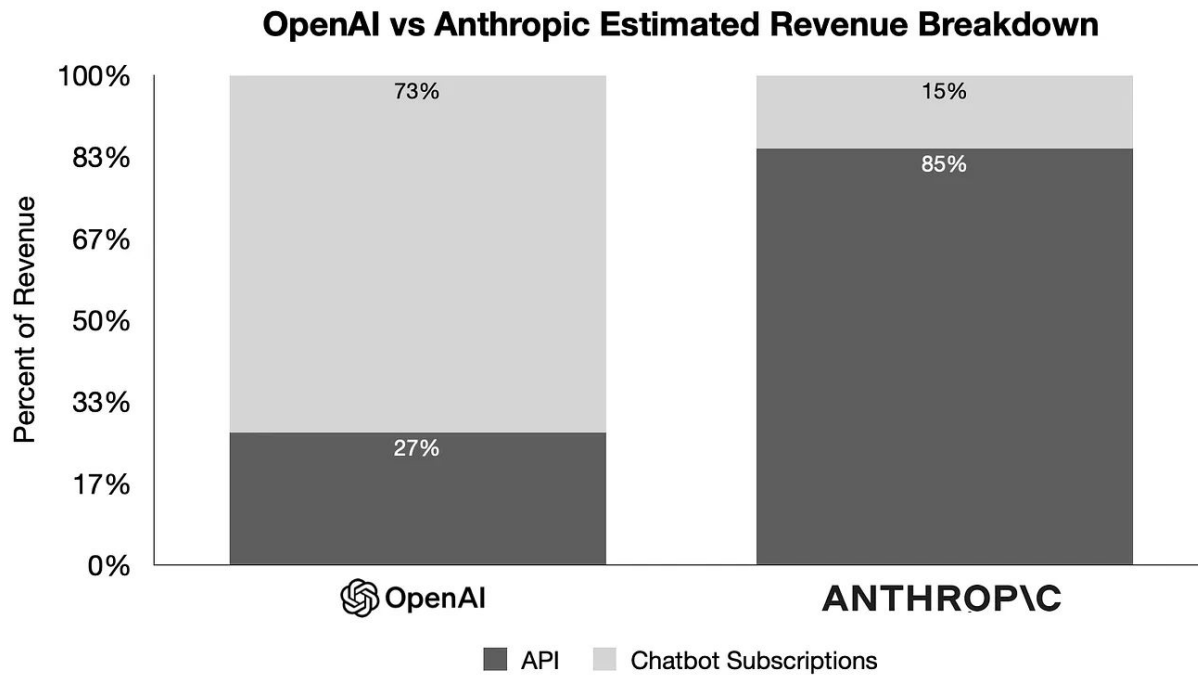
Chart: Shane Burke • Source: The Information reporting

## ANTHROPIC

Annualized revenue reached \$2 billion in the first quarter, the company confirmed, **more than doubling from a \$1 billion rate in the prior period**



OpenAI is becoming a consumer app company,  
and Anthropic an API company





# Leading model companies will likely have to become application layer companies to survive



Aisha Malik

## OpenAI is reportedly developing its own X-like social media platform

OpenAI is building its own X-like social media network, according to a new report from [The Verge](#). The project is still in the early stages, but there's an internal prototype focused on ChatGPT's image generation that contains a social feed.

Anthropic hires Instagram co-founder as head of product

Kyle Wiggers · 7:00 AM PDT · May 15, 2024

CNBC DISRUPTOR 50

## OpenAI in talks to pay about \$3 billion to acquire AI coding startup Windsurf

PUBLISHED WED, APR 16 2025 2:31 PM EDT | UPDATED 5 HOURS AGO

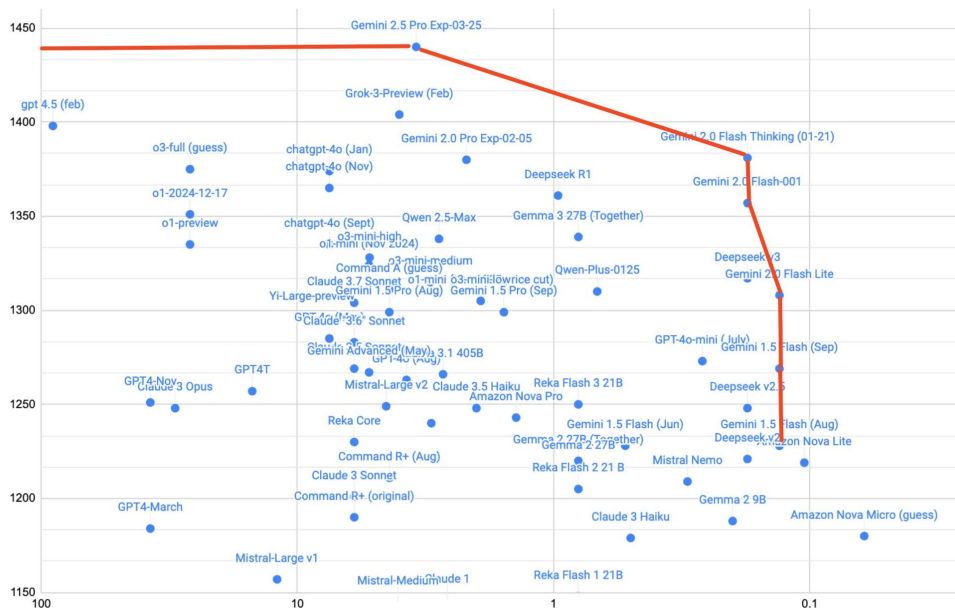


Hayden Field  
@HAYDENFIELD

SHARE    



Google “owns” pareto frontier of speed vs. quality as of April 2025. Reflective of how this is an economies of scale business

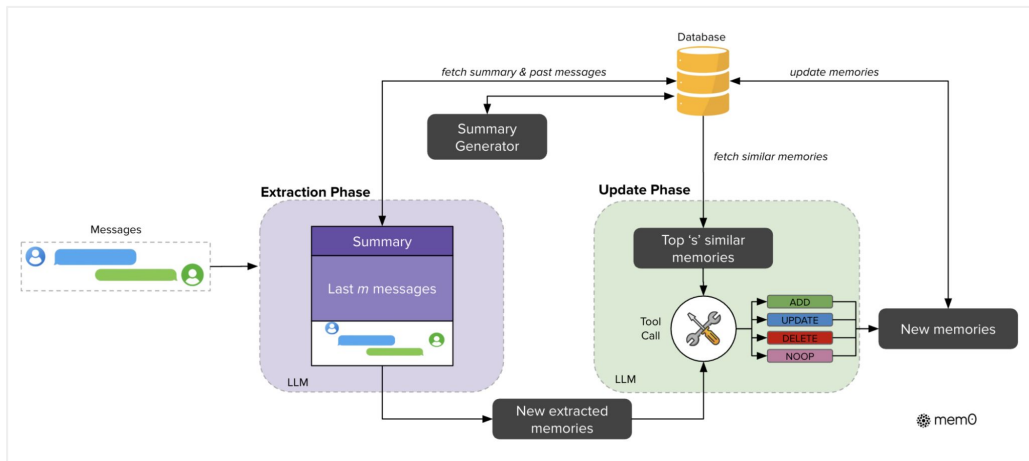


## Latent Space



# Memory is emerging as the key potential stickiness driver for consumer AI chat apps like ChatGPT

Whoever owns general consumer AI memory will own “Sign in with X” for all AI applications - allowing users to “bring their own memory”. But, memory is very difficult to get right.



Sample memory architecture from Mem0 - key question is what to remember and how to distill it, as well as how to blend memory with other context especially in longer sessions

February 13, 2024 Product

## Memory and new controls for ChatGPT

We're testing the ability for ChatGPT to remember things you discuss to make future chats more helpful. You're in control of ChatGPT's memory.



# Will foundation model companies in physical domains like robotics be able to “defy gravity” like we have seen in images & text?

Operational complexity of these domains are *much* higher than pure software. But, pricing is similar to software

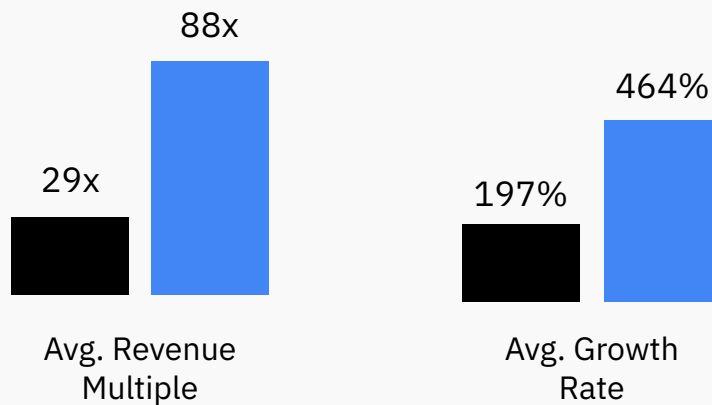
Company	Description	Funding	Key Investors
<b>Skilld AI</b>	Building foundation models for robotic control and manipulation	350M+	Thrive Capital, NEA, Khosla, etc.
<b>1X (formerly Halodi)</b>	Humanoid robotics with AI training systems	\$100M+	OpenAI Startup Fund, Tiger Global, EQT
<b>Cobot AI</b>	LLM-native robot training and cobot manipulation stack	150M+	Possibly early-stage VCs (unconfirmed)
<b>Physical Intelligence (Pi)</b>	Focused on training AI agents for general physical tasks	500M+	Likely stealth or early-stage funding
<b>Figure AI</b>	Humanoid robots powered by advanced AI models	~\$675M	Microsoft, OpenAI, Nvidia, Jeff Bezos
<b>Sanctuary AI</b>	General-purpose humanoid AI systems	\$100M+	Bell, Export Dev. Canada, others
<b>Agility Robotics</b>	Humanoid warehouse and logistics robots	\$180M+	DCVC, Playground, Amazon Industrial



# High valuations at the application layer, but also unprecedented revenue growth

- Bolt - \$0 to \$20M in 60 days
- HeyGen - \$0 to \$35M in a year
- Harvey - \$1M to \$15M in a year
- Hebbia - \$500k to \$10M in a year
- Glean - \$10 to \$40M in a year
- Together - \$1 to \$10M in a year
- Github CoPilot drives 40% percent of GitHub revenue growth
- OpenAI - >\$2B Annual Run Rate

Series B & C AI Companies - Valuation & Growth Premium (2H'23 - '24 YTD)



# AI-native applications are now in the multi-billion dollar run rate

Company	Description	Revenue/ARR
Midjourney	AI image generator	> \$200m ARR
Anysphere (Cursor)	AI code generation tool	> \$200m ARR
ElevenLabs	AI audio platform	>\$100m ARR
Glean	AI enterprise assistant (search and RAG)	> \$100m ARR
Runway	AI content generator and editor	\$84m ARR
Mercor	AI recruiting startup	\$75m ARR
Synthesia	AI video generator	> \$70m ARR
Abridge	Healthcare AI platform	> \$50m ARR
Harvey	Legal AI platform	> \$50m ARR
StackBlitz	AI code generation	> \$40m ARR
Writer	AI text-based content generator and editor	> \$40m ARR
Bolt	AI code generation	\$40m ARR
Codeium	AI code generation	~\$40m ARR
EvenUp	AI legal startup	> \$35m ARR
Clay	AI-powered sales and marketing platform	\$30m 2024 revs
Sierra	Customer support AI agent builder	\$20m ARR
Lovable	AI app-building platform	\$17m ARR
Hebbia	AI knowledge work platform	> \$13m ARR
Aragon.AI	AI headshot generator	> \$10m ARR
Magnific	AI image upscaler and enhancer	\$10m ARR
Poolside	AI software engineering platform	< \$10m 2024 revs
<b>Total</b>		<b>&gt; \$1.2b ARR</b>



# AI applications are fundamentally resetting expectations for what people will pay for software

---

It is not unreasonable to suspect most professionals will be paying 5–10k+/month in next few years



***“Amp is unconstrained in token usage (and therefore cost). Our sole incentive is to make it valuable, not match the cost of a subscription”***

---



Even in categories where the incumbent has every conceivable advantage, startups win

False narrative that AI is a 'sustaining innovation'. Building successful AI products looks too different.



**GitHub Copilot**

vs.



**CURSOR**



**Adobe Firefly**

vs.



**Krea**



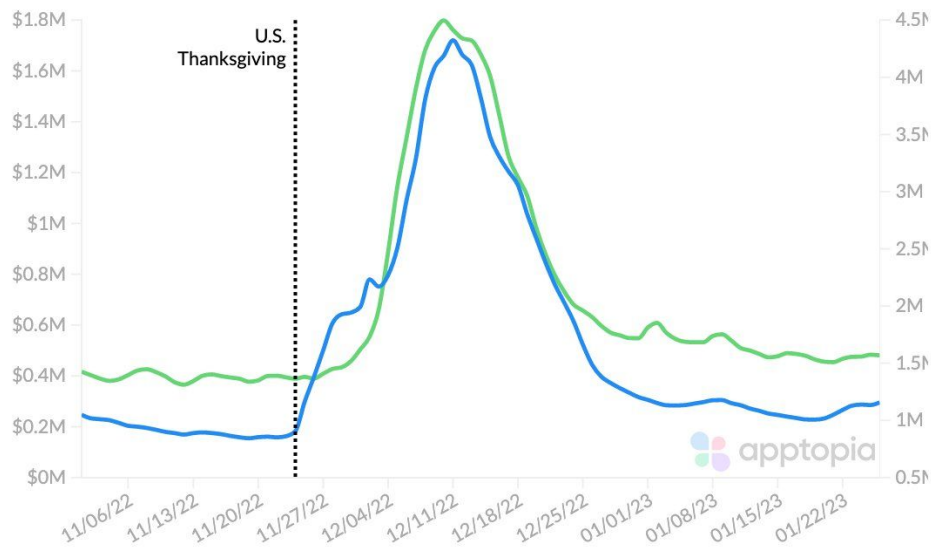


# Huge risk of novelty effect revenue in AI startups – numerous examples of “rise and fall” revenue curves

## AI photo app interest, on the back of Lensa AI, fell as quickly as it rose

Top 15 AI Photo Apps, Worldwide

■ Downloads ■ IAP Revenue



Overall, the AI market feels very “bubbly” across many dimensions

Many companies burning \$50M+ a year on training without established product-market-fit

**News** August 24, 2024

## **Three cofounders leave French AI startup H just three months after raising \$220m seed**



# Market structure of the GPU ecosystem looks *profoundly* different than the CPU ecosystem, driving rise of new “GPU Cloud” vendors

## CPU Clouds

- Bundle hardware w. cloud services
- Sell “low level” software services (e.g. EC2) at very low margin, and higher level services at incrementally higher margins
- Primarily pay-as-you-go model



## GPU Clouds

- Offer zero software beyond access to the GPU itself
- Do not focus on incremental services
- Extreme focus on fixed duration, longer term contracts



## Two drivers

Gen AI (GPU) workloads exhibit scaling laws, meaning that *incremental* compute always has marginal advantage.

So, given fixed budget, you care more about additional GPU-time vs. paying margin for “value add” software”. CPU workloads do not benefit from more compute beyond what is needed.

Dollar cost of GPU workloads tends to be >>> CPU workloads. As such, labor relationship flips - better to pay someone \$1M a year to write custom software than eat 10% margin increase for bundled software.



# NVIDIA & the GPU ecosystem remain the “guaranteed” winners

“AI Inference token generation has surged tenfold in just one year...” - NVIDIA Q1 Report

Market Summary > NVIDIA Corp

**137.38** USD

+ Follow

+128.46 (1,440.13%) ↑ past 5 years

Closed: Jun 2, 4:21 PM EDT • Disclaimer

After hours 137.56 +0.18 (0.13%)

1D | 5D | 1M | 6M | YTD | 1Y | 5Y | Max



## 06 What's next?



# Operating as an “AI-native” company looks fundamentally different

The best companies are increasingly adopting a mantra of: “Learn how to use AI, or leave.”

## What This Means

1. **Using AI effectively is now a fundamental expectation of everyone at Shopify.** It's a tool of all trades today, and will only grow in importance. Frankly, I don't think it's feasible to opt out of learning the skill of applying AI in your craft; you are welcome to try, but I want to be honest I cannot see this working out today, and definitely not tomorrow. Stagnation is almost certain, and stagnation is slow-motion failure. If you're not climbing, you're sliding.



Small, capital efficient teams are the new normal

## Meet Gamma, A Low-Profile AI Startup That's Actually Profitable

AI startup CEO Grant Lee has turned obsessive A/B testing – and a healthy distrust of venture capital – into 50 million in ARR and profits to go with 50 million users. Plus: Upstarts is on the road.

“Its last funding round was a modest \$12 million Series A from Accel last year.  
**Back then, it had 16 people; today it employs just 30.”**



## And the composition of teams is rapidly changing

---



*“I increasingly don’t see a difference between designers & product managers in our company”*

---

*VP Product, Growth-stage startup*

---



*“AI has completely changed how I think about hiring as a CMO. I don’t hire specialists anymore. I hire generalists who can use AI tools”*

---

*CMO, Publicly-listed company*

---





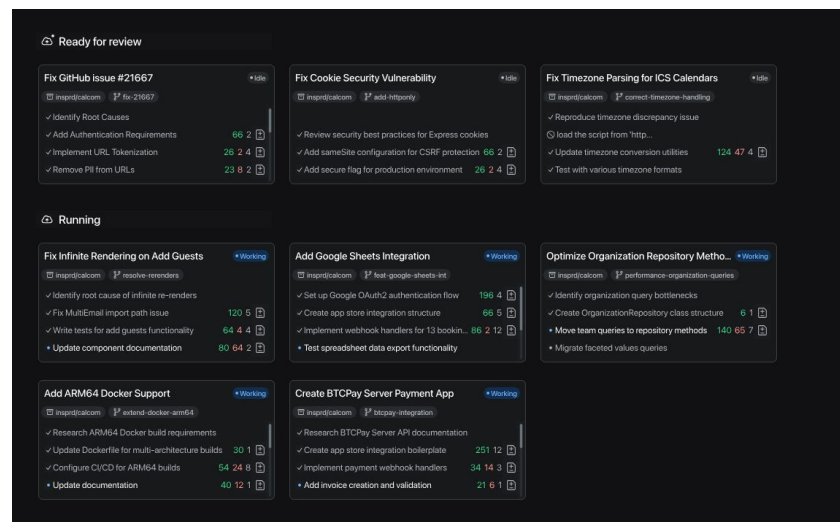
# Learning to “manage” fleets of AI workers will become a new skill, not dissimilar from managing people



*I haven't written a new line of code myself in 3 months. I spend all my time managing and reviewing agents*

—  
**CTO, leading CodeGen startup**

## “Agent Inbox” Design Pattern Emerging



# Products are being designed for AI as the primary “consumer”, not humans

## .cursorrules files are the new docs?

### Build Workers using a prompt

To use the prompt:

1. Use the click-to-copy button at the top right of the code block below to copy the full prompt to your clipboard
2. Paste into your AI tool of choice (for example OpenAI's ChatGPT or Anthropic's Claude)
3. Make sure to enter your part of the prompt at the end between the `<user_prompt>` and `</user_prompt>` tags.

Base prompt:

```
<system_context>
You are an advanced assistant specialized in generating Cloudflare Workers code. You
</system_context>

<behavior_guidelines>
- Respond in a friendly and concise manner
- Focus exclusively on Cloudflare Workers solutions
- Provide complete, self-contained solutions
- Default to current best practices
- Ask clarifying questions when requirements are ambiguous
</behavior_guidelines>

<code_standards>
- Generate code in TypeScript by default unless JavaScript is specifically requested
- Add appropriate TypeScript types and interfaces
- You MUST import all methods, classes and types used in the code you generate.
Use ES modules from npm (e.g. import { createMiddleware } from 'npm:express-middleware'
</code_standards>
```

## 80% of Neon database instances created by AI agents, not humans



**Nikita | Scaling Postgres**    
@nikitabase



Numbers of databases created by ai exceeded number of databases created by humans on the [@neondatabase](#) platform.

Serverless and instant provisioning is key!

6:47 PM · Feb 4, 2025 · **3,621** Views



2



5



64



3



# Where will the most value be destroyed?

## Outsource to In-house

Functions that were traditionally outsourced to agencies & consultancies will be moved in-house (e.g. video production)

## Specialist to Generalist

People in extremely specialized jobs, and tools oriented towards specialists, will be at risk as generalists + AI can achieve similar results

## Middle management will be eroded

Jobs primarily oriented around communication and information transfer will be deleted (e.g. project manager, middle manager)

## Incumbents in “line of fire” of AI

For example - unstructured data businesses (e.g. CRM), creative tool businesses (e.g. Figma), developer tool businesses (e.g. Github)

## Companies unwilling to go through cultural & organizational pain

Adapt to AI, or lose

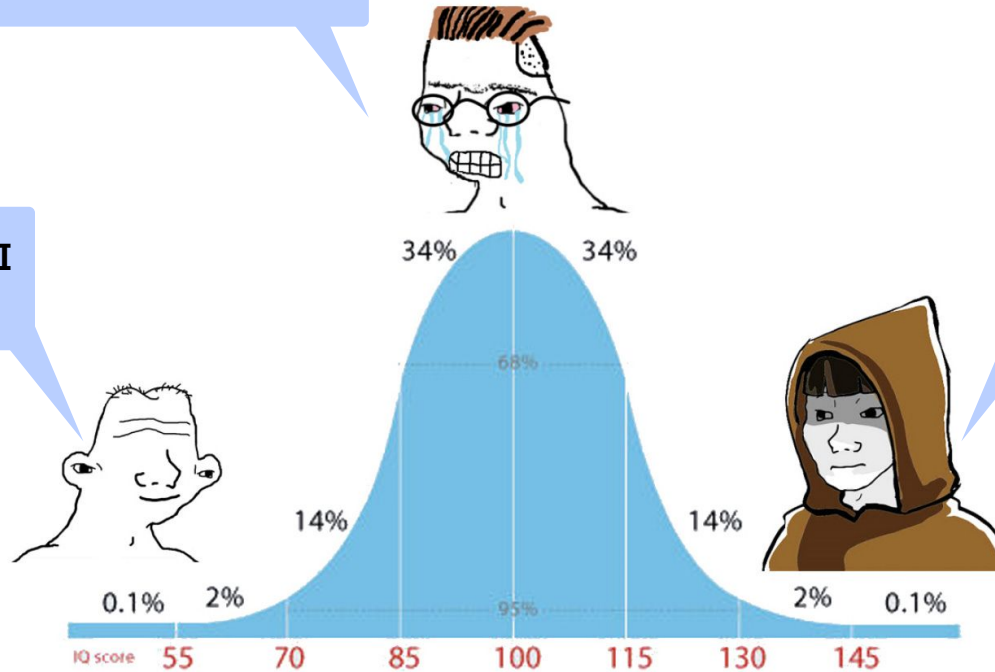


# Is AGI close? The smartest AI researchers seem to think so...

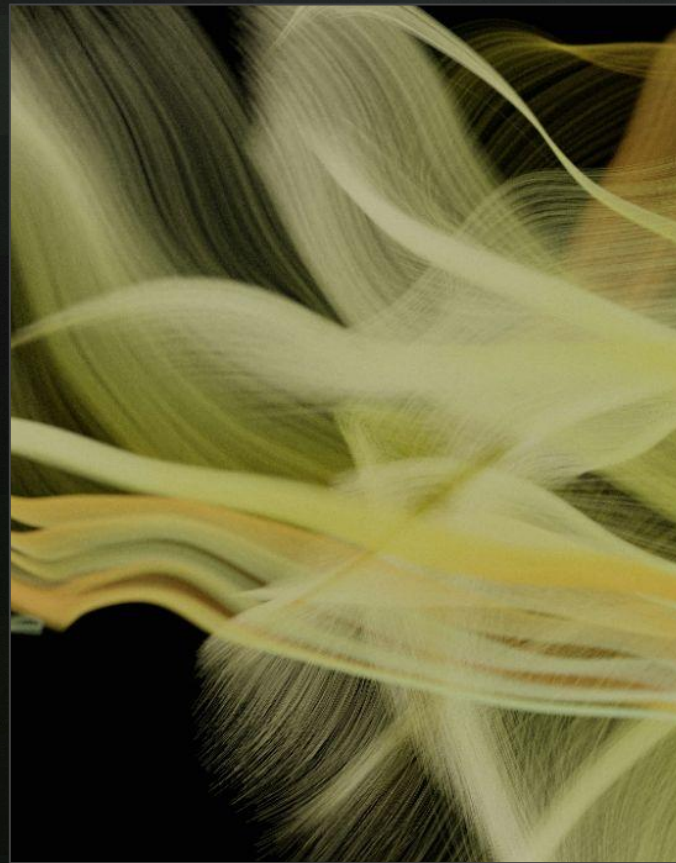
These models are just advanced statistical prediction devices

There will be AGI in 3 years

There will be AGI in 3 years



## 07 What We're Excited to See Built



# The downstream impact of AI code generation

The proliferation of AI code generation will have far reaching impacts on the rest of the software development lifecycle

---

## What this might look like:

### Reinvention of the SLDC

How might CICD, deployment, observability, git, and similar change in a world where AI is writing more code than humans?

### Software engineering “shifting right”

Many designers / PMs are already prototyping and submitting PRs thanks to AI code gen. Is there room for “IDEs” or similar products for such personas? How will traditional design & product tools change?

### The AI first software organization

The divide between engineering, product, and design is blurring. Task management tools will manage tasks for agents just as much as they manage humans. As organizational structures change in these ways, what new needs emerge?

### Validation, Testing, & Guardrails

The importance of testing, validation, and guardrails on software is going up *dramatically*. Will traditionally niche approaches become mainstream (e.g. load testing, fuzz testing, formal proofs, etc)? Will “review” workloads like code review need to be rethought?

We may also need better ways to automate “product” feedback as well – e.g. using LLMs to run synthetic experiments, synthetic UX studies



# Modern data-as-a-service businesses

LLMs have fundamentally altered our ability to collect, create, structure, understand, and transform data. We predict there will be a renaissance of “Data-as-a-Service” companies

---

## What this might look like:

### Collect previously inaccessible data

Use voice agents to call people or interview people. Use email agents to solicit data at a novel scale. Use LLMs conversational ability to extract deeper, more flexible insights from people (e.g. Listen Labs)

### Structure previously unstructure-able data

E.g. turn personal websites into metadata-rich people profiles.

### Novel business models

If AI lowers the cost/effort/time required to collect certain data by 1000x via synthetic results, AI interviews, or similar, can you re-invent the business model of a data/research category? E.g. could you build a *proactive* expert interview platform that reaches out to you with relevant, personalized interviews

### Use LLMs at the “last-mile” in data delivery

Allow users to get “custom” data on demand vs. being forced into a predefined schema/structure. Build rich query & analysis workflows into the data business.

### Synthetic + Real

LLMs are very good at mimicking users/people. Use LLMs to create synthetic data, and blend that synthetic data with real data in an intelligent way (e.g. Evidenza)



# Next-generation creative tools

There's an obvious opportunity to disrupt creative expression of all forms

---

## What this might look like:

### Defensibility via something besides AI

Mechanisms worth exploring:

**Networks** – New forms of social networks built around AI-based democratization of creation. Allow users to “fork” or “remix” content generated by others, or create new forms of marketplaces for AI-native creators

**Runtimes** – Lower level infrastructure innovations in computer graphics or similar that become *more valuable* as AI makes it easier to produce content

**Workflow Specificity** – Not enough companies have focused on specific types of creators.

E.g. what might an AI image gen company built *purely* for brand design, or *purely* for photographers, look like?

### Mixing traditional editing w/ AI

Immense opportunity to innovate on how to combine traditional editing modalities with generative AI, allowing for both rapid experimentation & precise control.

E.g. generative 3D + mesh editing + point cloud editing + 3D style transfer. Subframe is a good example of this in UX design (combining “vibe prototyping” w/ classical layer editing)

### You need VSCode in order to build the copilot

Unlike in software engineering, most other professional design domains lack an open source editor with a rich plugin ecosystem.

So, how do you sequence building the editor, then the copilot?  
E.g. see Sequence in video editing





# Data for AI

Data is likely to remain the largest bottleneck for advancing AI systems. What are novel and clever ways of producing more, high-fidelity data?

---

## What this might look like:

### Data as a by-product

Products or applications which are offered for “free” but generate high-quality data for ML systems as an implicit byproduct ([more here](#))

### Simulation & RL Environments

What might an “Ansys for RL” look like? Can we come up with high-quality environments to train, evaluate, and improve agents? What might these look like and could a startup help create, manage, and run them?

### Data management for AI

Better ways to structure, manage, query, cluster, curate & clean data for AI (e.g. [Datology](#))

### Community & Network Based Evals

[LMArena](#) is a good, early example of tapping into the “wisdom of the crowd” to produce evaluation criteria for models.

What are other mechanisms for creating marketplaces or networks for people to evaluate AI systems?

### Verifiers, Checkers, & Reward Models

Generalist reward models and verifier models are likely to become a standard model class, analogous to embedding models, which assist in generating reward data for AI.

### “Vertical” Annotation companies

Companies offering extremely high quality annotation data in specialized domains that are outside the scope of “mainstream” labeling labs (e.g. [DavidAI](#) in audio)



# AI & Science

Generative models will have a profound impact across the sciences - from chemistry, biology, materials, mathematics, climate, and more

---

## What this might look like:

### Data for the sciences

Data is, *by far*, the limiting factor for foundation model utility in many science categories such as biology & chemistry.

We think there are opportunities around novel forms of data capture (e.g. sensing/screening), as well Mercor/Scale style businesses that identify more scalable forms of data annotation. E.g. [Elio Labs](#) building a novel microscope designed specifically for AI.

### Closed-Loop Generate + Verify (e.g. “AI Scientist”)

Combine advances in generative models with improvements in traditional computational modeling (e.g. CFD) and wet lab automation to form closed-loop, generator + verifier style systems in areas like materials, biology, chemistry, etc.

E.g. [Orbital Materials](#) does this in materials

### AI & Math

#### Autonomous theorem proving

We often need to “prove” traits of mission critical systems - e.g. proving that aircraft will behave correctly, or that a distributed system has no consensus bugs.

Can you combine LLMs w/ formal mathematical languages like [Lean](#) to build autonomous verifiers, reducing the cost/effort/complexity to prove traits of systems by multiple order of magnitude?

#### Auto-formalization & Optimization

Mathematical optimization (e.g. Gurobi, Mathworks) has traditionally been limited by the knowledge of how to *formalize* business problems into math. LLMs are good at this. Does this allow for novel startups?



# Infrastructure for AI

AI systems & workloads are creating many new infrastructure requirements, as well as altering that way we need to think about traditional infrastructure categories

---

## What this might look like:

### Multi-Modal Data Management

Generative models mean most companies will increasingly need to manage & process complex multi-modal data, including audio, video, images, text. The tooling to do this is still early (e.g. see [Aperture](#), [Lance](#) as good examples)

### AI-provisioned infrastructure

Many traditional infrastructure categories (e.g. databases, VMs, APIs) are transitioning to being used more by AI agents than humans.

This *greatly* increases the importance of serverless architectures, scale-to-zero, multi-tenancy w/ strong isolation, treating everything-as-code, & support for ephemeral and volatile workload patterns (e.g. see why Replit uses Neon as a backend\*)

### Infrastructure primitives for AI

Web search for AI systems, browsers for AI systems, computing sandboxes for AI, wallet & payments infra for AI, etc. Most “web primitives” will need to be redesigned for AI

### Infra problems that get 100x worse with agents

For example, authorization and fine grained access control for internal services will get 100x worse when a bunch of AI agents have access to do many things in your environment.

### GPU Ecosystem

Dealing with GPUs is still immensely complicated. Lots of continued opportunity for GPU abstraction, multi-tenant GPUs, abstracting GPU vs. CPU, and novel compute marketplaces for GPU (e.g. [SF Compute](#))

\*See also how [Bauplan](#) is interesting in terms of exposing data pipelines to agents since everything is sandboxed and git-versioned by default, a rarity in data infrastructure



# Foundation Model Systems

How do infrastructure & tooling needs change as we begin to view foundation model applications more like systems?

---

## What this might look like:

### Optimization of FM Systems

Along the lines of DSPy & Ember - how do we make it easier to build, test, and evaluate complex foundation model systems which make heavy use of more complex systems paradigms such as repeated sampling, fan out + fan in, verifiers, and similar?

I think over time this will more like “simulation” - ala Applied Intuition in autonomous vehicles. Given sophisticated FM applications can likely be treated as complex systems, you will likely want optimize them end to end.

### Reinforcement Learning & Verifiers

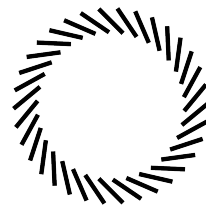
There is likely a startup opportunity to offer best in class generalist reward models and verifiers as an API, similar to what we saw with embedding models (e.g. see [GR](#))

Beyond this - it is becoming clear that most AI application companies will benefit from doing domain-specific RL against end-to-end task success in their apps. The tooling & infra to do this is very complex. How do we make it easier?

### Generator + Verifier Systems

I am extremely interested in any founders combining foundation models as “generators” with secondary verifier systems - e.g. see [KernelBench](#) and this [blog](#)





## About the Author

Davis Treybig is a Partner at Innovation Endeavors, an early-stage venture fund that backs founders solving complex technical and engineering challenges to rethink large industries.

Artificial intelligence is a core focus area of the fund. We have invested broadly in AI across areas like biotechnology (e.g. Eikon), robotics (e.g. Gatik), computer vision (e.g. Planet), financial research (AlphaSense), healthcare (Viz), the built environment (e.g. Trunk Tools), & more.

Davis primarily invests in computing infrastructure, machine intelligence, and next-generation tools for builders - including developers, designers, and engineers. Recent investments include Augment, Bauplan, Capsule, Dosu, Extend and Responsive.

[davis@innovationendeavors.com](mailto:davis@innovationendeavors.com) • [Substack](#) • [Twitter](#) • [LinkedIn](#)



INNOVATION ENDEAVORS

[davis@innovationendeavors.com](mailto:davis@innovationendeavors.com)

