




Richard Ngo

Follow

Jan 5 · 14 min read ·  Listen

 Save



Visualizing the deep learning revolution

The field of AI has undergone a revolution over the last decade, driven by the success of deep learning techniques. This post aims to convey three ideas using a series of illustrative examples:

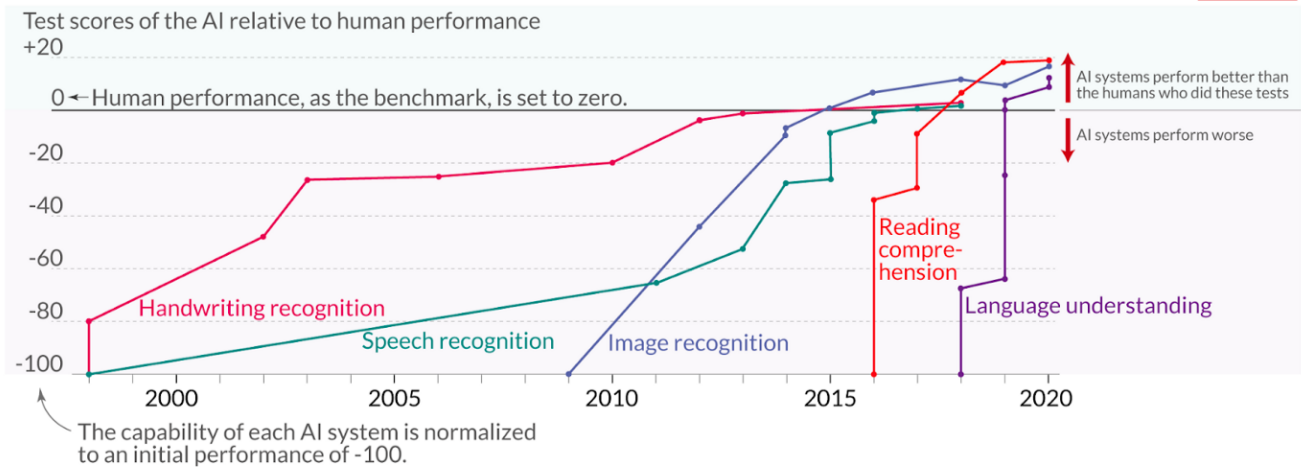
1. There have been huge jumps in the capabilities of AIs over the last decade, to the point where it's becoming hard to specify tasks that AIs *can't* do.
2. This progress has been primarily driven by scaling up a handful of relatively simple algorithms (rather than by developing a more principled or scientific understanding of deep learning).
3. Very few people predicted that progress would be anywhere near this fast; but many of those who did also predict that we might face existential risk from AGI in the coming decades.

I'll focus on four domains: vision, games, language-based tasks, and science. The first two have more limited real-world applications, but provide particularly graphic and intuitive examples of the pace of progress.

Vision

Image recognition

Image recognition has been a focus of AI for many decades. Early research focused on simple domains like handwriting; performance has now improved significantly, beating human performance on many datasets.



Data source: Kiela et al. (2021) – Dynabench: Rethinking Benchmarking in NLP
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the author Max Roser

However, it's hard to interpret scores on benchmarks in an intuitive sense, so we'll focus on domains where progress can be visualized more easily.

Image generation

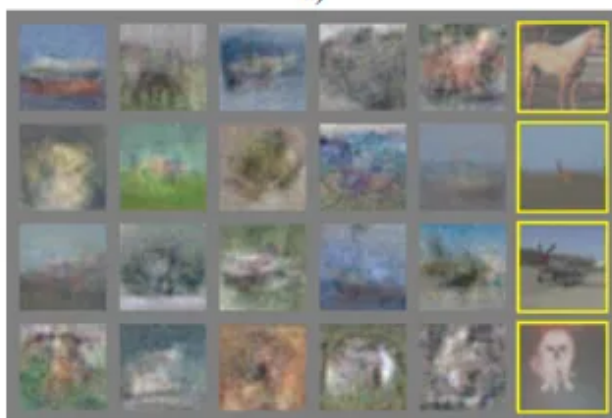
In 2014, AI image generation advanced significantly with the introduction of Generative Adversarial Networks (GANs). However, the first GANs could only generate very simple or blurry images, like the ones below.



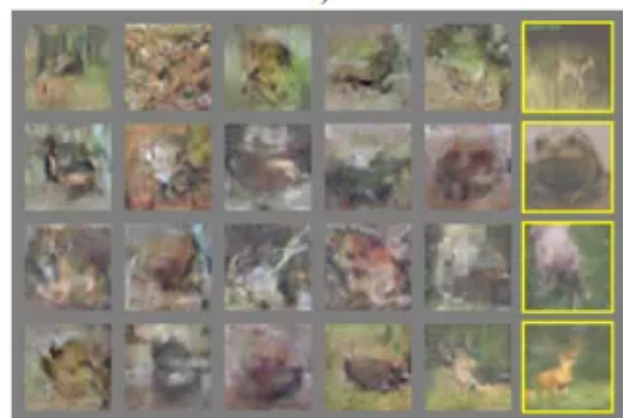
a)



b)



c)



d)

Images with yellow borders are real, all others are GAN-generated.

Over the next 8 years, image generation progressed at a very rapid rate; the figure below shows images generated by state-of-the-art systems in each year. Over the last two years in particular, these systems made a lot of progress in generating complex creative scenes in response to language prompts.

Timeline of images generated by artificial intelligence

Our World
in Data

These people don't exist. All images were generated by artificial intelligence.

2014



Goodfellow et al. (2014) – Generative Adversarial Networks

2015



Radford, Metz, and Chintala (2015) – Unsupervised Representation Learning with Deep Convolutional GANs

2016



Liu and Tuzel (2016) – Coupled GANs

2017



Karras et al. (2017) – Progressive Growing of GANs for Improved Quality, Stability, and Variation

2018



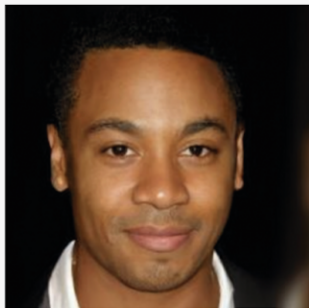
Karras, Laine, and Aila (2018) – A Style-Based Generator Architecture for Generative Adversarial Networks

2019



Karras et al. (2019) – Analyzing and Improving the Image Quality of StyleGAN

2020



Ho, Jain, & Abbeel (2020) – Denoising Diffusion Probabilistic Models

2021 Image generated with the prompt: "a couple of people are sitting on a wood bench"



Ramesh et al. (2021) – Zero-Shot Text-to-Image Generation (OpenAI's DALL-E 1)

2022 Image generated with the prompt: "A Pomeranian is sitting on the King's throne wearing a crown. Two tiger soldiers are standing next to the throne."



Saharia et al. (2022) – Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding (Google's Imagen)

OurWorldinData.org – Research and data to make progress against the world's largest problems. Licensed under CC-BY by the authors Charlie Giattino and Max Roser

This is an astounding rate of progress. What drove it? In part, it was the development of new algorithms — most notably GANs, transformers and diffusion

models. However, the key underlying factor was scaling up the amount of compute and data used during training. One demonstration of this comes from the Parti series of image models, which includes four networks of different sizes (with parameter counts ranging from 350 million to 20 billion). Although they were all trained in the same way, for the three prompts below you can clearly see how much better the bigger models are than the smaller ones (e.g. by watching the ability to portray text gradually emerge).



Prompt: A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!



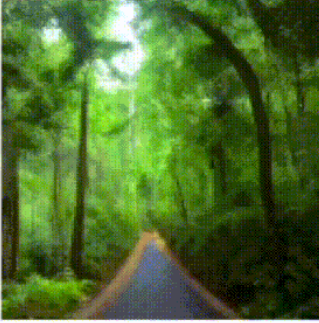
Prompt: A map of the United States made out of sushi. It is on a table next to a glass of red wine.



Prompt: A squirrel gives an apple to a bird

Video generation

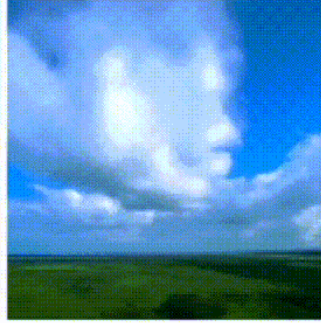
path in a tropical forest



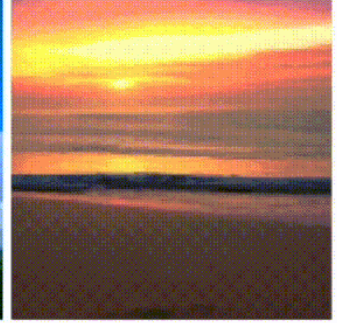
Construction Site Activity



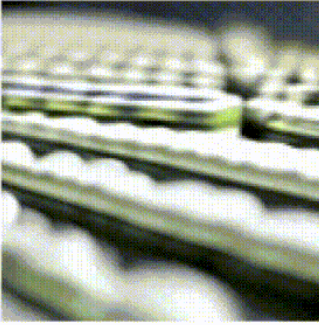
Clouds moving



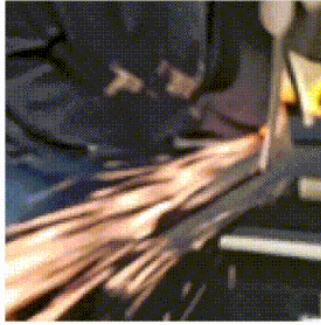
sunset at sea



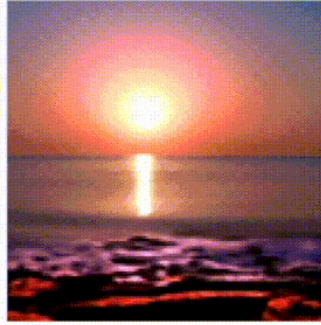
Aluminum cans on conveyor belt



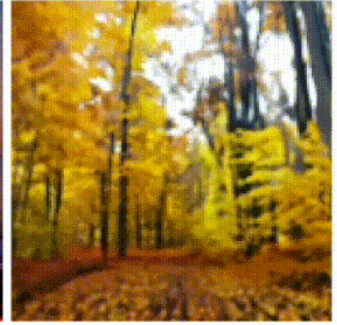
Close up of Craftsman worker
sawing a steel pipe,
Technician concept



Dramatic ocean sunset



Forest in Autumn



Berlin - Brandenburg Gate at
night



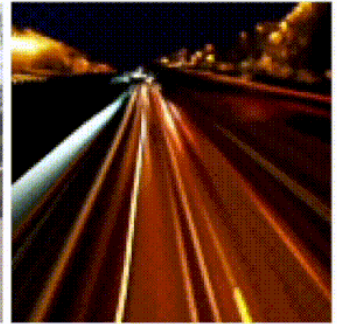
Aerial of horses on a pasture



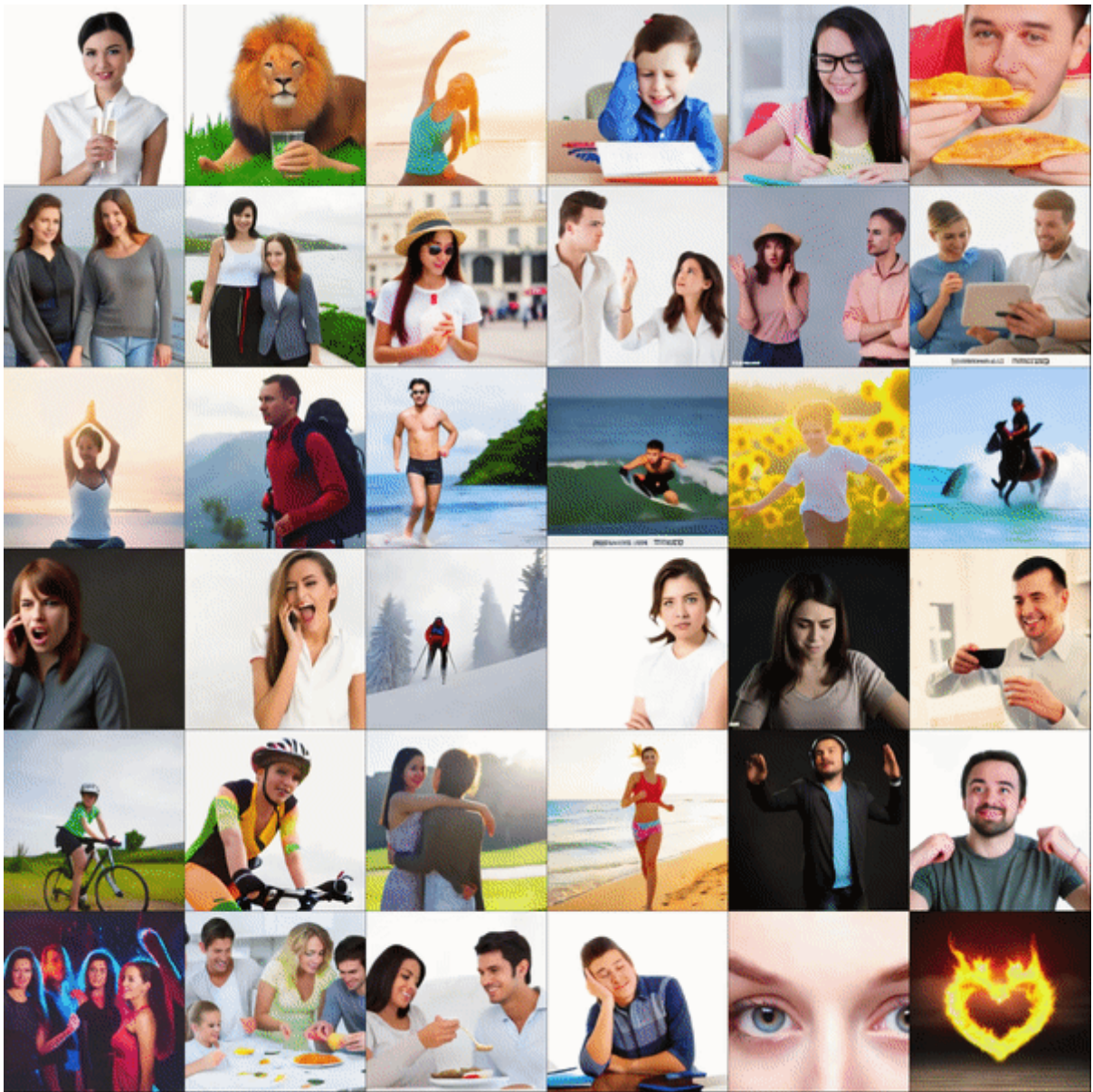
Traffic jam on 23 de Maio
avenue, both directions, south
of Sao Paulo,



Busy freeway at night



Video diffusion (April 2022)



CogVideo (May 2022)



Make-A-Video (September 2022)

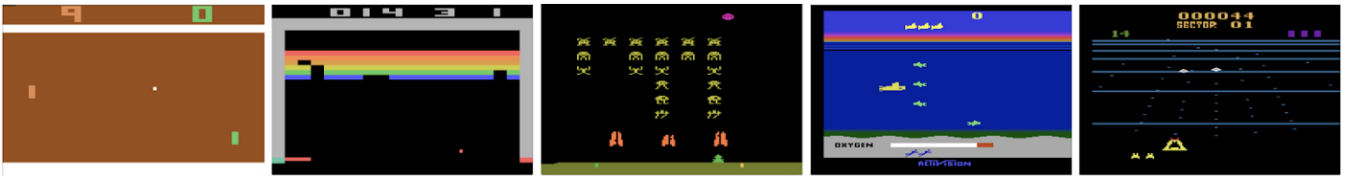


Imagen Video (October 2022)

Games

Video games

In 2013, the state of the art for using ML to learn to play video games was beating humans at a few very simple Atari games, like Pong and Breakout.



By 2018, this had progressed to beating professionals at the most complex esports, in particular StarCraft and DOTA2, in which players take thousands of actions in a row throughout games lasting up to an hour.

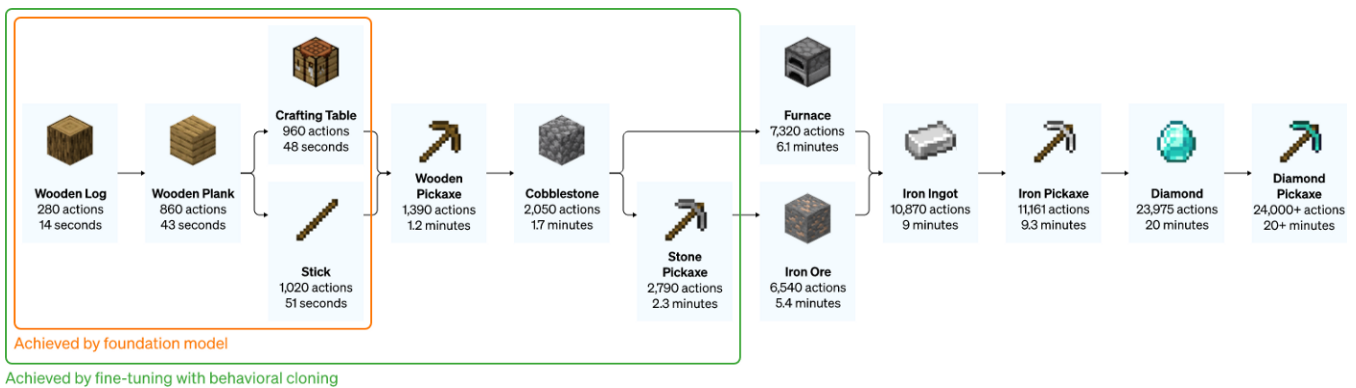


DeepMind's AlphaStar model, which beat a top professional at StarCraft



OpenAI Five, which beat a team of top professionals at DOTA2

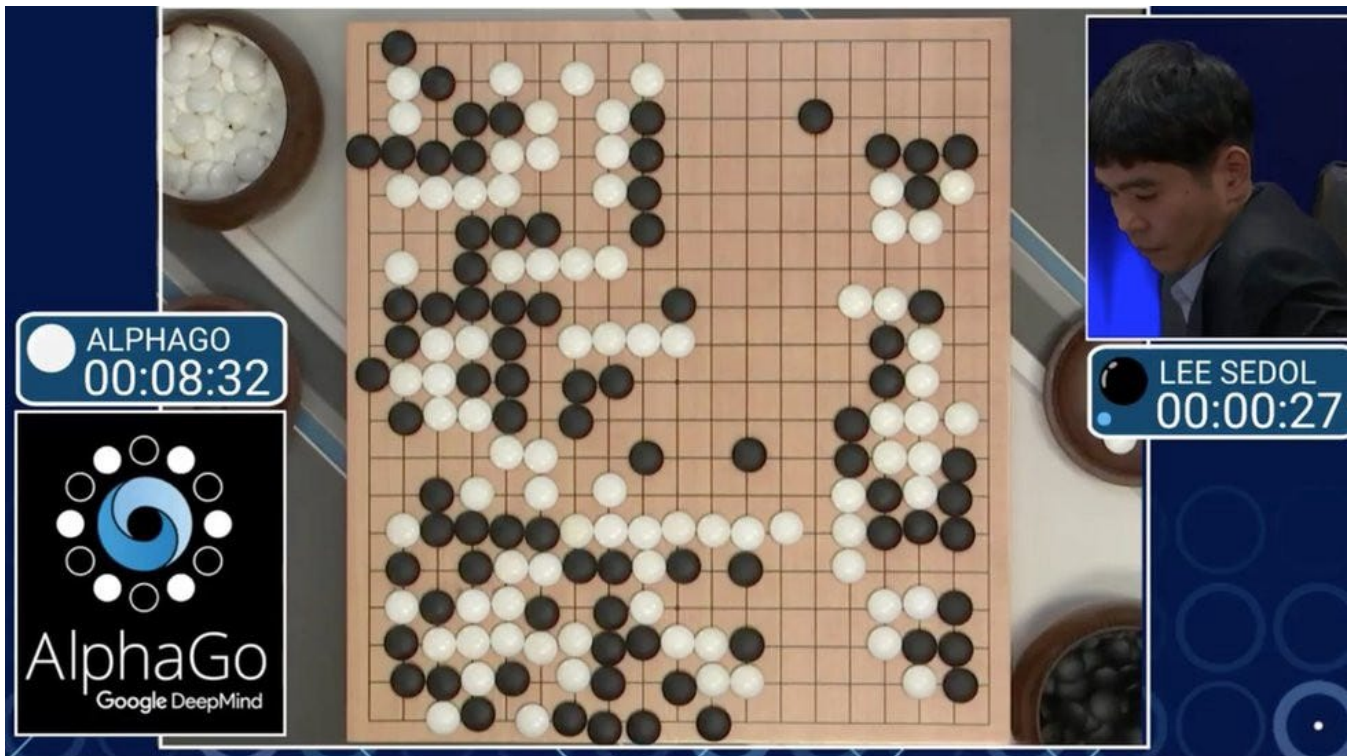
And by 2022, capabilities had extended from narrow games to open-ended environments like Minecraft, where training on videos of human players allowed AIs to learn to take complex sequences of actions involving many different steps. Below you can see the many subtasks involved in making a diamond pickaxe, and a video of the AI carrying them out.



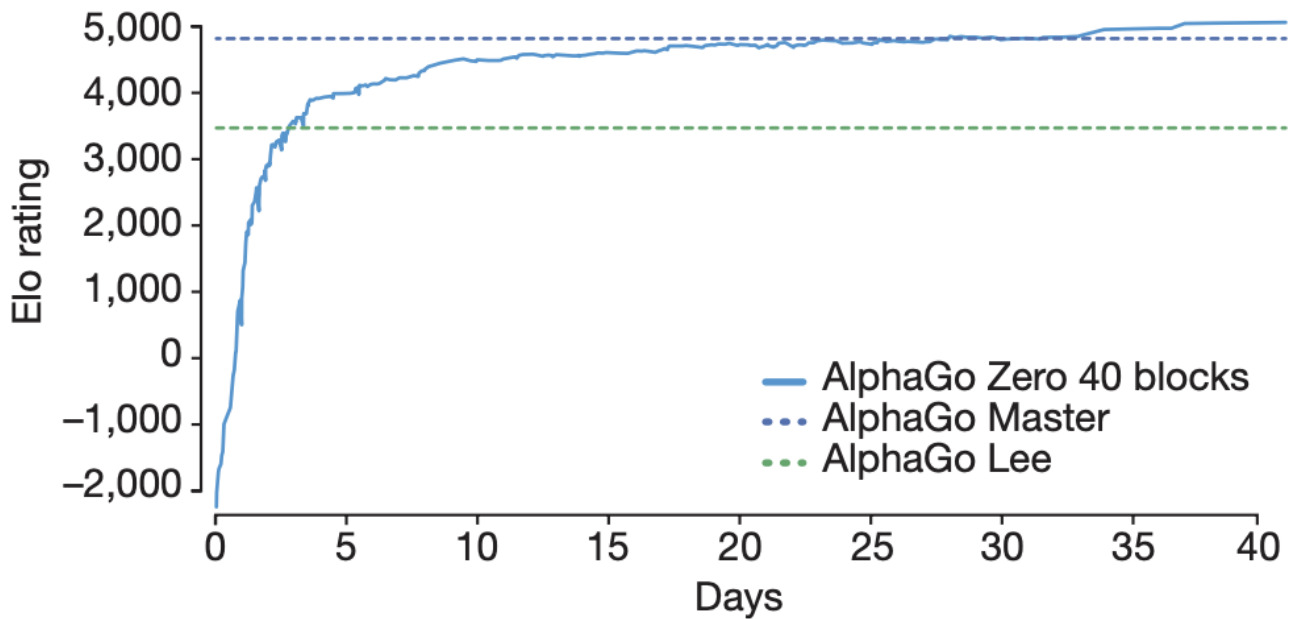


Strategy games

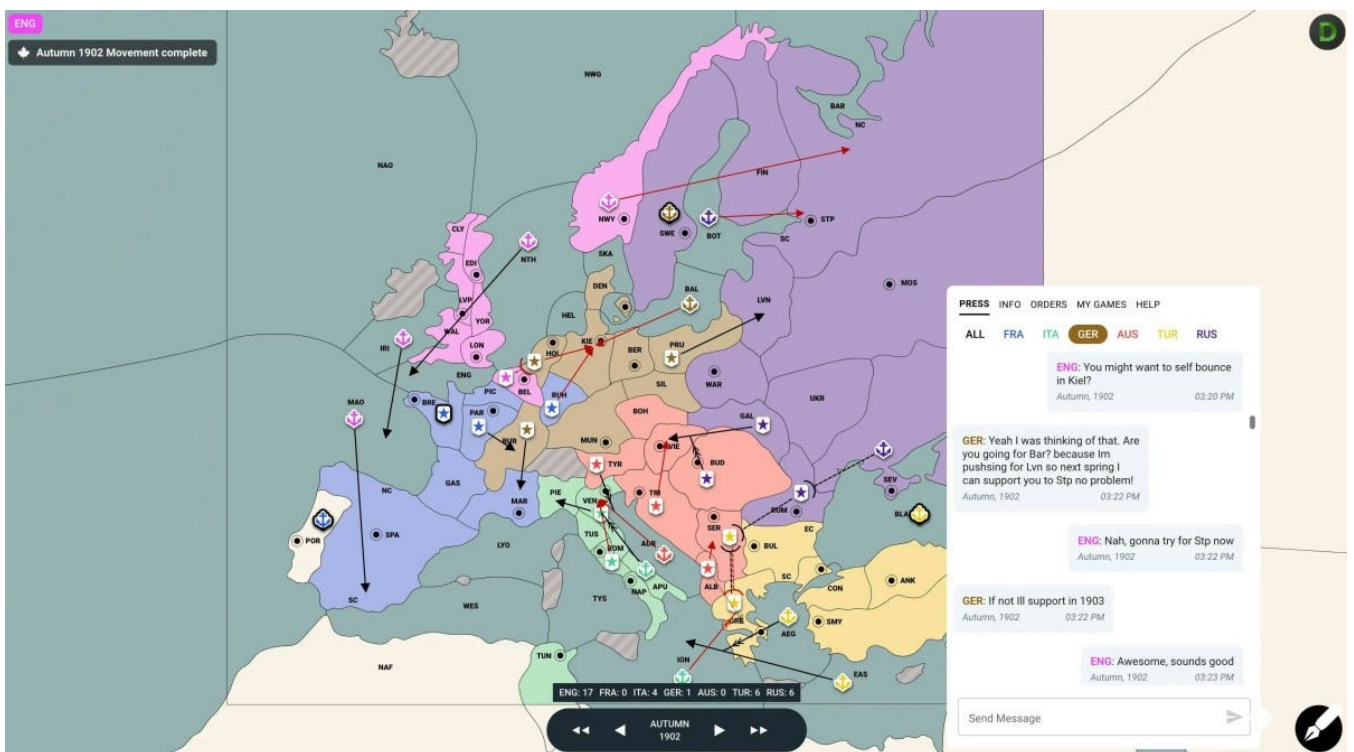
2016 and 2017 saw DeepMind's AlphaGo system beat the world's best players at Go, a board game far more complex than chess.



Notably, a later version of AlphaGo trained without any human data, only by playing against itself, reached superhuman level (indicated by the green line in the graph below) after less than three days of training, which involved rediscovering many aspects of Go theory and strategy that humans had developed over thousands of years.



However, chess and Go are still quite simple games — players can only interact in very limited ways. 2022 saw AIs reaching human-level performance at the board game Diplomacy (a strategy game similar to Risk). Diplomacy requires players to form hidden alliances via complex language-based negotiations, as well as deceiving other players when betraying those alliances.



Language-based tasks

The level of language capabilities used in Diplomacy negotiations were enabled by the development over the last few years of large language models (LLMs) trained on

billions of words of text. Despite only being trained to predict the next word, in order to do so very accurately LLMs develop a range of emergent skills, including broad knowledge about the world which can be accessed by giving them the right prompts. The main LLMs we'll discuss in this section are OpenAI's [GPT-2](#), [GPT-3](#) and [ChatGPT](#) models (the last of which you can [try online here](#)) and several variants of [Google's PaLM model](#).

Text generation

While it's hard to compare the results from different prompts, some samples from different models can give an intuitive sense for the rate of progress.

In 2011, state-of-the-art language models couldn't even put together grammatical sentences:

PROMPT: THE MEANING OF LIFE IS

The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger. In the show's agreement unanimously resurfaced. The wild pastured with consistent street forests were incorporated by the 15th century BE. In 1996 the primary rapford undergoes an effort that the reserve conditioning, written into Jewish cities, sleepers to incorporate the .St Eurasia that activates the population. Mar??a Nationale, Kelli, Zedlat-Dukastoe, Florendon, Ptu's thought is. To adapt in most parts of North America, the dynamic

The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger. In the show's agreement unanimously resurfaced. The wild pastured with consistent street forests were incorporated by the 15th century BE. In 1996 the primary rapford undergoes an effort that the reserve conditioning, written into Jewish cities, sleepers to incorporate the .St Eurasia that activates the population. Mar??a Nationale, Kelli, Zedlat-Dukastoe, Florendon, Ptu's thought is. To adapt in most parts of North America, the dynamic

Here's a slightly more coherent sample from [a model trained on Shakespeare's texts in 2015](#):

VIOLA:

Why, Salisbury must find his flesh and thought
That which I am not apt, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

A big jump in text generation came with the development of GPT-2 in 2019, which was consistently able to generate coherent sentences containing relevant knowledge about the world.

PROMPT: MILEY CYRUS WAS CAUGHT SHOPLIFTING FROM ABERCROMBIE AND FITCH ON HOLLYWOOD BOULEVARD TODAY

The 19-year-old singer was caught on camera being escorted out of the store by security guards.

The singer was wearing a black hoodie with the label 'Blurred Lines' on the front and 'Fashion Police' on the back

The singer was also wearing a pair of black-rimmed glasses, a black jacket, black jeans and black sandals.

She was carrying a pair of black and white striped gloves and a small black bag.

A further jump came with GPT-3 in 2020, which was trained in a very similar way to GPT-2, but with around 100 times more parameters.

PROMPT: UNITED METHODISTS AGREE TO HISTORIC SPLIT – THOSE WHO OPPOSE GAY MARRIAGE WILL FORM THEIR OWN DENOMINATION

After two days of intense debate, the United Methodist Church has agreed to a historic split – one that is expected to end in the creation of a new denomination, one that will be “theologically and socially conservative,” according to the Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will “discipline” clergy who officiate at same-sex

weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.



The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the “largest Protestant denomination in the US”, but that it has been shrinking in recent decades. The new split will be the second in the church’s history. The first occurred in 1968, when roughly 10% of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split “comes at a critical time for the church, which has been losing members for years,” which has been “pushed toward the brink of a schism over the role of LGBTQ people in the church.” Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

The next model in the series was [ChatGPT, released in 2022](#). ChatGPT significantly surpasses GPT-3 on a range of tasks; as one example, see its creative writing response to the prompt below:



write a biblical verse in the style of the king james bible explaining how to remove a peanut butter sandwich from a VCR 



And it came to pass that a man was troubled by a peanut butter sandwich, for it had been placed within his VCR, and he knew not how to remove it.  

And he cried out to the Lord, saying, "Oh, Lord, how can I remove this sandwich from my VCR, for it is stuck fast and will not budge?"

And the Lord spoke unto him, saying, "Fear not, my child, for I shall guide thy hand and show thee the way. Take thy butter knife, and carefully insert it between the sandwich and the VCR, and gently pry them apart. And with patience and perseverance, the sandwich shall be removed, and thy VCR shall be saved."

And the man did as the Lord commanded, and lo and behold, the sandwich was removed from the VCR, and the man was saved.

And the Lord said, "Verily I say unto thee, seek not to put thy peanut butter sandwiches in thy VCR, for it is not a suitable place for such things. Rather, keep thy sandwiches in thy refrigerator or on thy plate, where they belong."

And the man heeded the Lord's words, and from that day forth, he kept his sandwiches in their proper place, and was saved from trouble and woe. Amen.

Common-sense question-answering

Historically, one of the key bottlenecks in AI has been common-sense reasoning. Over the last few years, though, large language models have become capable of giving straightforwardly correct common-sense answers to a wide range of questions. (Note, however, that whether or not a LLM responds correctly to a question often depends sensitively on how the question is asked.)

Here are some explanations of jokes from [Google's 2022 PaLM model](#) (with the top box containing examples given to the model, and then the boxes below containing the model's own responses):

Explaining a Joke

I will explain these jokes:

(1) The problem with kleptomaniacs is that they always take things literally.

Explanation: This joke is wordplay. Someone who "takes things literally" is someone who doesn't fully understand social cues and context, which is a negative trait. But the definition of kleptomania is someone who literally takes things.

(2) Always borrow money from a pessimist. They'll never expect it back.

Explanation: Most people expect you to pay them back when you borrow money, however a pessimist is someone who always assumes the worst, so if you borrow money from them, they will expect that you won't pay them back anyways.

Input: I was supposed to start writing the paper at 5:00 PM. But then I started playing with this cool new language model for 10 minutes. 10 minutes later, it's suddenly 9:30 PM!

Model Output: This joke is about how time flies when you're having fun. The person was supposed to start writing a paper at 5:00 PM, but they started playing with a language model instead. They thought they were only playing with it for 10 minutes, but it was actually 4.5 hours.

Input: I was going to fly to visit my family on April 6th. My mom said "Oh great, your step dad's poetry reading is that night!" So now I'm flying in on April 7th.

Model Output: The joke is that the speaker's mother is trying to get them to go to their step dad's poetry reading, but the speaker doesn't want to go, so they are changing their flight to the day after the poetry reading.

Input: Reading "War and Peace" doesn't scare me. I once had to review the CL for my intern's whole project on his final day!

Model Output: 'War and Peace' is a very long book, so reading it is a daunting task. However, the speaker is saying they had to review the code for their intern's project on their final day, which is even more daunting.

Input: I think I understand why me and my partner are such a good match. I get excited when I smell food and become super alert when the doorbell rings. My girlfriend Shelley becomes ecstatic when she sees a person who she just saw 5 hours ago.

Model Output: This joke is about dogs. Dogs get excited when they smell food and when the doorbell rings. The girlfriend is excited to see someone she just saw 5 hours ago, which is a trait of dogs.

Input: I tried 10,000 random restarts of my neural network, but I was accused of overfitting. I guess no good seed goes unpunished.

Model Output: This joke is a pun. A neural network is a computer program that can learn from data. A "seed" is a number that is used to initialize a random number generator. A "good seed" is a number that produces a good random number generator. The phrase "no good deed goes unpunished" means that if you do something good, you will be punished for it.

Input: What's the difference between a zebra and an umbrella? One is a striped animal related to horses, another is a device you use to stop the rain from falling on you.

Model Output: This joke is an anti-joke. The joke is that the answer is obvious, and the joke is that you were expecting a funny answer.

Here are a range of further capabilities showcased by PaLM:

Cause & Effect

Prompt

Wh

Planning and strategic thinking

LLMs have also developed the capability to describe long-term plans, and break them down into a series of tasks. For example, here's ChatGPT:



Can you help me build an app that takes links to essays and produces 10 bullet summaries using GPT-3?



Sure, I'd be happy to help you build an app that uses GPT-3 to produce 10 bullet summaries of essays. GPT-3, or Generative Pretrained Transformer 3, is a state-of-the-art language processing model developed by OpenAI. It's capable of generating human-like text and can be used for a wide range of natural language processing tasks, including summarization.



To build an app that uses GPT-3 to generate summaries of essays, you'll need to do the following:

1. Sign up for an API key from OpenAI. You'll need this to access GPT-3 and use it in your app.
2. Choose a programming language and development environment to use for your app. Some popular options for building AI-powered apps include Python, JavaScript, and Swift.
3. Install the necessary libraries and dependencies for your chosen programming language. These will likely include the OpenAI API client library, as well as any other libraries you need for handling text input and output, making HTTP requests, and so on.
4. Write the code for your app. This will likely involve reading in the link to the essay, using the OpenAI API to generate a summary of the essay, and then formatting the summary as a list of 10 bullet points.
5. Test your app to make sure it's working as expected. You may need to iterate on your code and fine-tune the GPT-3 model to get the best possible summaries.

Overall, building an app that uses GPT-3 to generate summaries of essays is a complex undertaking, but it's certainly possible with the right tools and knowledge. If you need help along the way, don't hesitate to reach out to a professional developer or AI expert for assistance.

Try again

While their ability to act on these plans is currently limited, a range of work aims to adapt LLMs to perform computer-based tasks. Here's an example of the [ACT-1](#) model carrying out a task using a computer interface:

Houston Homes for Sale: Hous x 15002 Cozy Hollow Ln, Housto x

redfin.com/TX/Houston/15002-Cozy-Hollow-Ln-77044/...

"Adept Client" started debugging this browser Cancel

City, ... Buy Rent New Sell Mortgage Real Estate Agents Feed 4

Search Overview Property Details Sale & Tax History Schools

LISTED BY REDFIN

Adept

find me a house in houston that works for a family of 4. my budget is 600k

3D Walkthrough Street View

Enter a task...

15002 Cozy Hollow Ln, Houston, TX 77044

\$425,000 Buy with Redfin: \$424,132 >

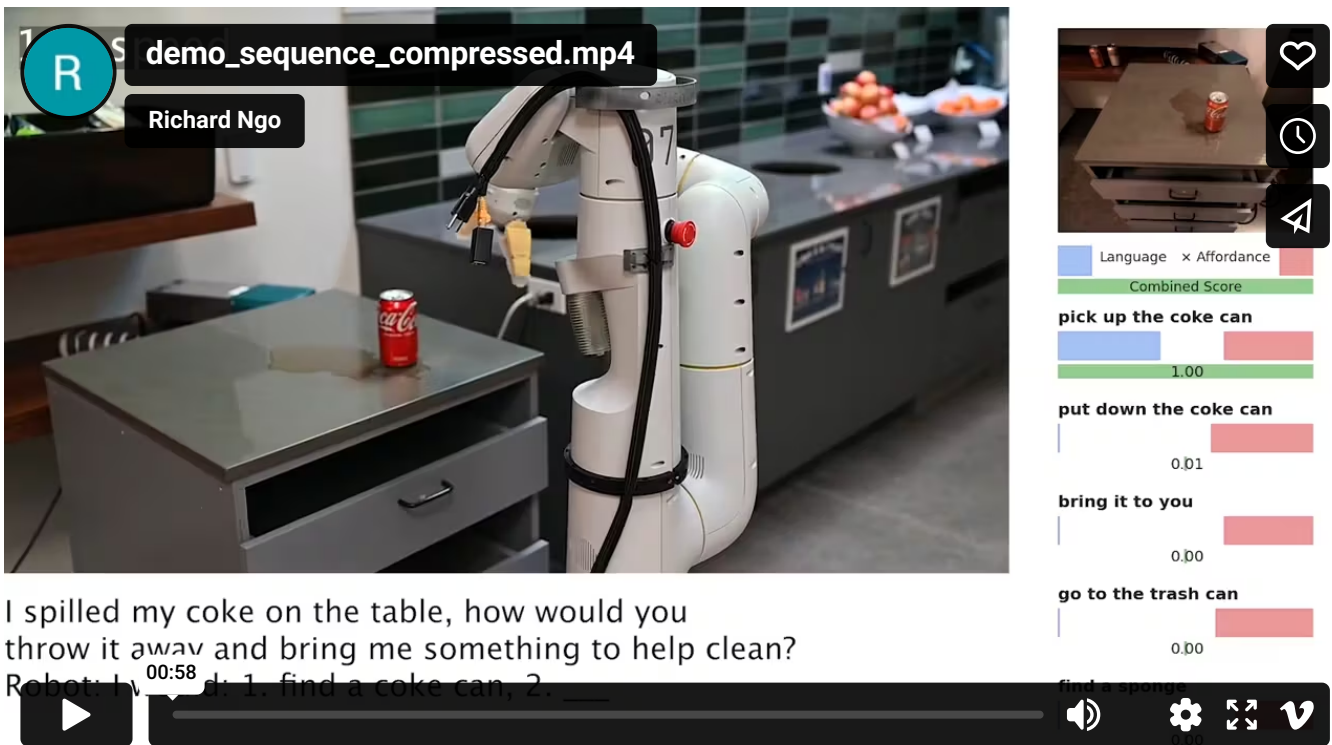
4 Beds 3.5 Baths 3,407 Sq Ft

This home is popular

It's been viewed 1,082 times. Tour it in person or via video chat before it's gone!

Tomorrow: 9:00 am • 10:00 am • 11:00 am • 12:00 pm • 1:00 pm • More times

And here's a video of Google's [PaLM-SayCan](#) robot generating and carrying out a multi-step plan in response to user instructions:



Science

While success at the language-based tasks described above might allow AIs to automate much white-collar work, in the long term the most impactful and concerning capabilities will likely involve AIs capable of automating the process of scientific and technological advancement (including progress in AI development itself). Over the last few years we've seen rapid progress towards AIs which are useful in scientific research.

Coding

An early step towards automating coding came with GPT-2, where the researchers who trained it noticed a surprising phenomenon: it was capable of auto-completing realistic (although usually buggy) code, because its training data had included a range of code from across the internet.

That was in 2019. In 2020, GPT-3 developed much more sophisticated coding capabilities, with the Codex language model (based on GPT-3) being capable of following language instructions to write whole functions. Here's an example of Codex writing code to implement a feature in a simple game, based on language instructions.



When the rocket is clicked, temporarily display some text saying "Firing thrusters!" in white on the current location -- and temporarily speed up by 4x for 0.25 second.



```
'hidden';

/* Now set background to the
color of space. */
document.body.style.background
Color = 'black';

/* When the rocket is clicked,
temporarily display some text
saying "Firing thrusters!" in
white on the current location
-- and temporarily speed up by
4x for 0.25 second. */
rocketship.onclick =
function() {
  var text =
document.createElement('div');
  text.innerHTML = 'Firing
thrusters!';
  text.style.color = 'white';
  text.style.position =
'absolute';
  text.style.left =
rocketship.offsetLeft + 'px';
  text.style.top =
rocketship.offsetTop + 'px';

document.body.appendChild(text
);
```

The full video can be found [here](#).

Here's an example of Codex writing code to plot a graph:

Now plot the results. Label both axes (y axis is max temperature), rotate the x ticks, and add a title



Select only the rows with datatype TMAX, and convert from C to F. Also change dates to MM-DD

	date	datatype	station
attributes	value		
2	06-30	TMAX	GHCND:USW00023272
, ,W,2400	60.98		
6	07-01	TMAX	GHCND:USW00023272
, ,W,2400	64.94		

```
"""Load the downloaded
data into a dataframe,
print results."""
import pandas as pd

df =
pd.DataFrame(response.json
()['results'])

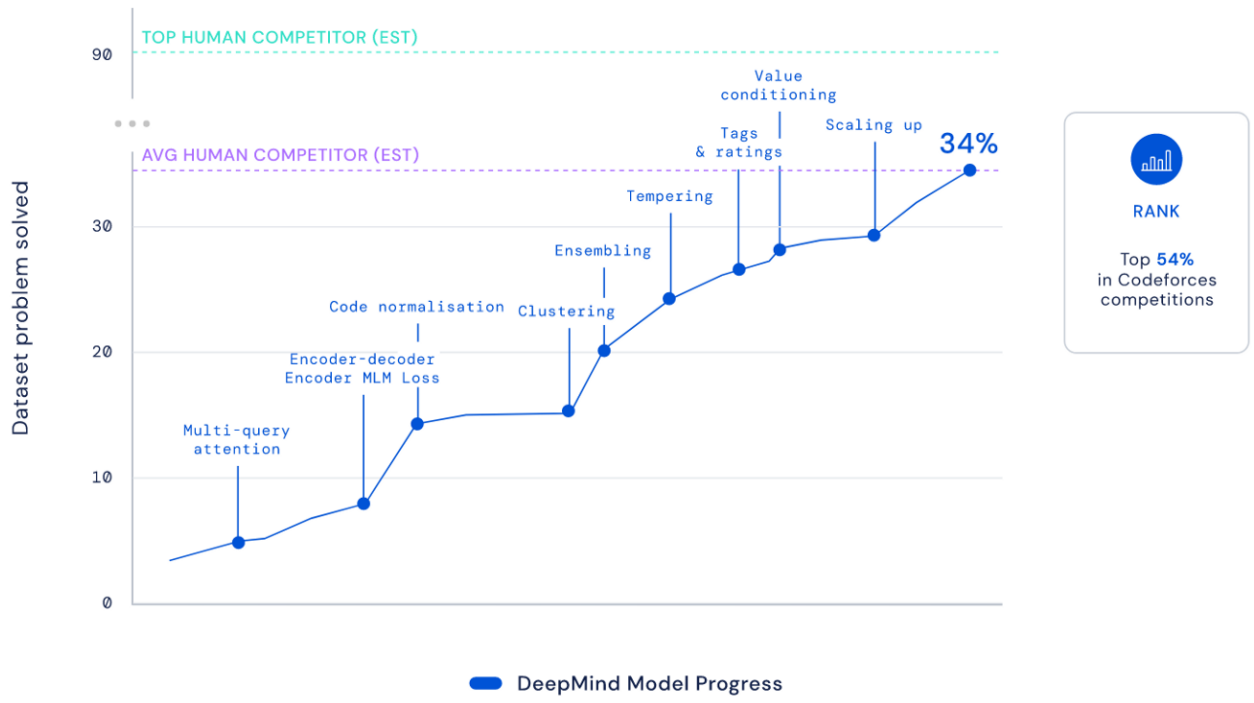
print(df)

"""Select only the rows
with datatype TMAX, and
convert from C to F. Also
change dates to MM-DD"""
df = df[df['datatype'] ==
'TMAX']
df['date'] =
pd.to_datetime(df['date'])
df['date'] =
df['date'].dt.strftime('%m
-%d')
df['value'] = df['value']
* 1.8 + 32

print(df)
```

The full video can be found [here](#).

DeepMind's AlphaCode system was trained to solve competitive programming problems, and would have placed in the top 54% of humans who participated in the coding competitions it was tested on.



ChatGPT is also capable of answering high-level coding questions, including some at a comparable level to interview questions for hiring software engineers:



Find the bug with this code:

```
...  
for (var i = 0; i < 5; i++) {  
  setTimeout(() => console.log(i), 1000)  
}  
...
```



The code contains a bug that will cause it to always print the number 5 to the console, rather than the expected sequence of numbers from 0 to 4. This is because the `setTimeout` function creates a closure that captures the value of the `i` variable, but the `i` variable is changed before the callback function is executed.



To fix this bug, the code can be modified to create a new variable in the loop that is passed to the `setTimeout` callback function, rather than using the loop variable directly. This will ensure that each callback function captures the correct value of the loop variable, and the expected output will be produced. Here is an example of how the code could be fixed:

```
Copy code  
  
for (var i = 0; i < 5; i++) {  
  // Create a new variable and pass its value to setTimeout  
  const value = i;  
  setTimeout(() => console.log(value), 1000)  
}
```

This code will print the numbers 0 to 4 to the console, as expected. Alternatively, the `let` keyword can be used in place of `var` to declare the loop variable, which will automatically create a new variable for each iteration of the loop and avoid the need to create a new variable manually. This is a common pattern for avoiding closure-related bugs in JavaScript.



What does this weird looking regex do?

```
/\b(((\d|two|three)):(\d|one|two|three))?(:(\d|one|two|three))?\s?(am|pm)?\b/i;
```



This regular expression appears to be for matching time values in a text. The regular expression uses a combination of numeric and word-based patterns to match a time value in the 12-hour clock format, with an optional seconds value and AM/PM indicator. 👍

The regular expression begins with `\b` which is a word boundary anchor. This will match the position between a word character and a non-word character. This is followed by a capturing group `((\d|two|three)):(\d|one|two|three)`, which will match the hour and minute values of the time.

The hour value is matched by the pattern `(\d|two|three)`. This will match either a single-digit hour value in the range `0` to `1`, followed by an optional digit in the range `0` to `9`, or it will match the word "two" followed by a character in the range `0` to `3`. This allows for the word-based representation of the hour value, such as "two" for 2, "three" for 3, etc.

The minute value is matched by the pattern `(\d|one|two|three)`. This will match an optional character in the range `0` to `5`, followed by a digit in the range `0` to `9`. This allows for the word-based representation of the minute value, such as "zero" for 0, "five" for 5, etc.

After the hour and minute values, the regular expression has an optional capturing group `:(\d|one|two|three)?` which will match the seconds value of the time, if present. This uses the same pattern as the minute value to match the seconds.

Finally, the regular expression has an optional non-capturing group `\s?(am|pm)?` which will match an optional space character followed by an optional AM or PM indicator. The `\b` at the end of the regular expression matches another word boundary, ensuring that the matched time value is a complete word in the text. The `i` at the end of the regular expression specifies that the match should be case-insensitive.

Overall, this regular expression will match time values in the 12-hour clock format, with an optional seconds value and AM/PM indicator. The time can be represented using a combination of numeric and word-based patterns, allowing for flexibility in the input.

Math

The last few years have also seen incredible progress on applying AI to mathematics. AI had already been used to help prove some mathematical theorems, but they required humans to manually specify what parts of the solution to search

for. By contrast, modern language models are capable of answering free-form mathematical questions, including some questions used in math competitions.

Here are some examples of Google's MINERVA model (a variant of PaLM) answering math questions:

Question: Assume that the variance of the first n natural numbers is 10, and the variance of the first m even natural numbers is 16. Compute $m + n$.

Model output: Let σ^2 be the variance of the first m even natural numbers, and τ^2 be the variance of the first n natural numbers. Then, we have $\tau^2 = \frac{n^2-1}{12}$ and $\sigma^2 = \frac{m^2-1}{3}$. We solve the system of equations:

$$\begin{aligned}\frac{n^2-1}{12} &= 10 \\ \frac{m^2-1}{3} &= 16.\end{aligned}$$

This translates into the system

$$\begin{aligned}n^2 &= 121 \\ m^2 &= 48 + 1 = 49.\end{aligned}$$

Therefore, $n = \boxed{11}$ and $m = \boxed{7}$, so $n + m = \boxed{18}$.

Question: For every $a, b, b \neq a$ prove that

$$\frac{a^2 + b^2}{2} > \left(\frac{a+b}{2}\right)^2.$$

Model output:

$$\begin{aligned}\frac{a^2+b^2}{2} &> \left(\frac{a+b}{2}\right)^2 \\ \iff \frac{a^2+b^2}{2} &> \frac{a^2+b^2+2ab}{4} \\ \iff a^2 + b^2 &> \frac{a^2+b^2+2ab}{2} \\ \iff 2a^2 + 2b^2 &> a^2 + b^2 + 2ab \\ \iff a^2 + b^2 &> 2ab \\ \iff a^2 + b^2 - 2ab &> 0 \\ \iff (a-b)^2 &> 0\end{aligned}$$

which is true, because the square of a real number is positive.

Question: A particle moves so that it is at $(3 \sin \frac{t}{4}, 3 \cos \frac{t}{4})$ at time t . Find the speed of the particle, measured in unit of distance per unit of time.

Model output: The speed is the magnitude of the velocity vector. Therefore, we must first find the velocity vector, which is the derivative of the position vector. We have

$$\mathbf{v} = \frac{d}{dt} \left(3 \sin \frac{t}{4}, 3 \cos \frac{t}{4} \right) = 3 \left(\cos \frac{t}{4}, -\sin \frac{t}{4} \right) \frac{1}{4} = \left(\frac{3}{4} \cos \frac{t}{4}, -\frac{3}{4} \sin \frac{t}{4} \right).$$

Therefore, the magnitude of the velocity vector is

$$\begin{aligned}|\mathbf{v}| &= \sqrt{\left(\frac{3}{4} \cos \frac{t}{4}\right)^2 + \left(-\frac{3}{4} \sin \frac{t}{4}\right)^2} = \sqrt{\frac{9}{16} \cos^2 \frac{t}{4} + \frac{9}{16} \sin^2 \frac{t}{4}} \\ &= \frac{3}{4} \sqrt{\cos^2 \frac{t}{4} + \sin^2 \frac{t}{4}} = \frac{3}{4}.\end{aligned}$$

Therefore, the speed of the particle is $\boxed{\frac{3}{4}}$.

Here's an example of a model from OpenAI which is capable of both informal proofs and formal proofs in the LEAN programming language:

PROBLEM 6

Adapted from IMO Longlist 1990 Problem 77^[4]

For a, b, c reals, prove that $(a^2 + ab + b^2)(b^2 + bc + c^2)(c^2 + ca + a^2) \geq (ab + bc + ca)^3$.

↔ FORMAL

INFORMAL

After cancelling terms appearing on both sides, we are left to prove that:

$$3a^2b^2c^2 + \sum_{sym} a^3b^2c \leq \sum_{cyc} a^4bc + \sum_{cyc} (a^4b^2 + b^4c^2)$$

After multiplying both sides by 2, we can rearrange the above inequality to:

$$0 \leq \sum_{cyc} (a^2b + a^2c - b^2c)^2$$

which clearly holds, giving the claim.

Open in app ↗

Get unlimited access



PROBLEM 6

Adapted from IMO Longlist 1990 Problem 77^[4]

For a, b, c reals, prove that $(a^2 + ab + b^2)(b^2 + bc + c^2)(c^2 + ca + a^2) \geq (ab + bc + ca)^3$.

↔ FORMAL

INFORMAL

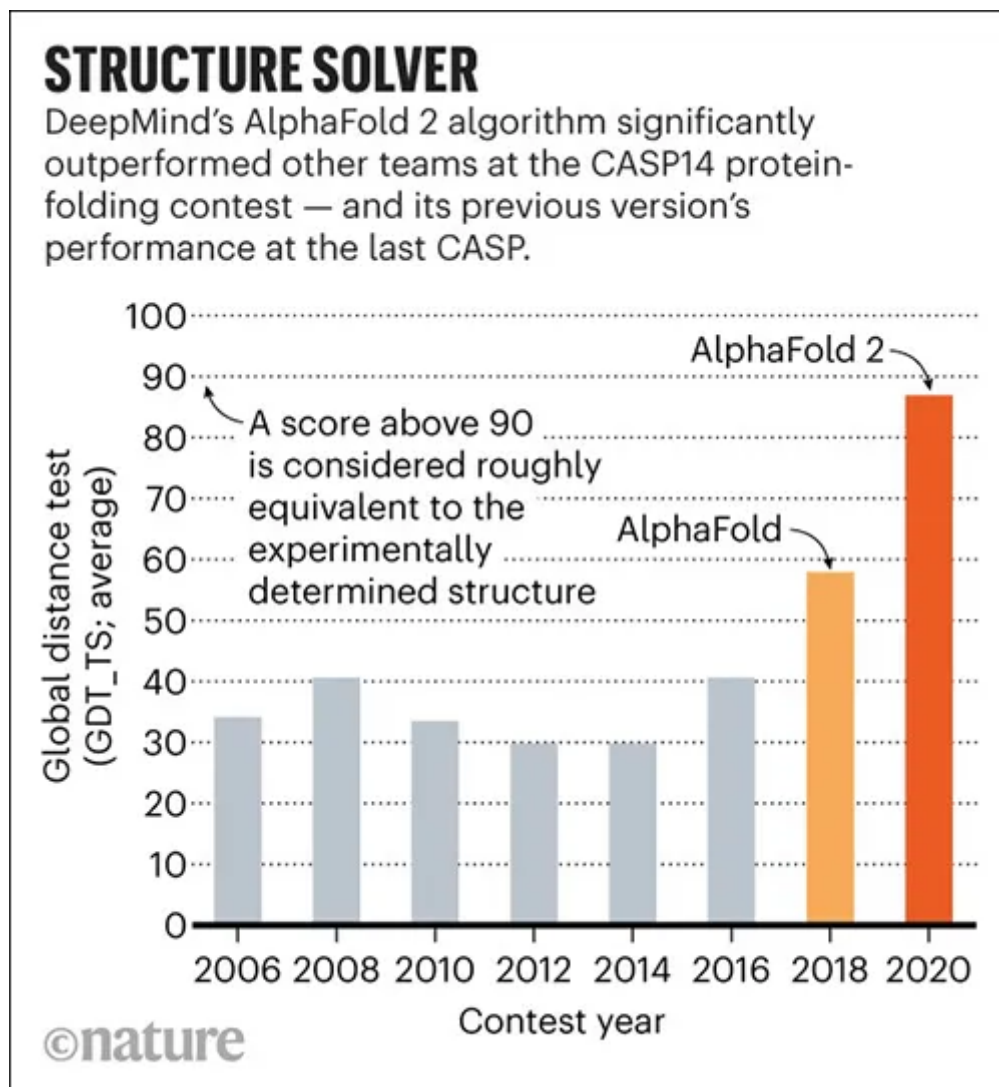
```
theorem imo_longlist_1990_p77
  (a b c : ℝ) :
  (a * b + b * c + c * a)^3 ≤
    (a^2 + a * b + b^2) * (b^2 + b * c + c^2) *
    (c^2 + c * a + a^2) :=
begin
  -- The three initial steps use Cauchy–Schwarz to prove
  -- `(a * b + b * c) ^ 2 ≤ (a ^ 2 + b ^ 2) * (b ^ 2 + c ^ 2)`
  -- which is required for the final call to `nlinarith`.
  let u : euclidean_space ℝ (fin 2) := ![a, b],
  let v : euclidean_space ℝ (fin 2) := ![b, c],
  have h₀ := real_inner_mul_inner_self_le u v,
  simp [u, v, fin.sum_univ_succ,
        ←pow_two, ←pow_two, le_of_lt, mul_assoc] at h₀,
  -- The model introduces another required cut (i.e. invent
  -- the term `0 ≤ (c + a) * (c + a)` and proves it).
  have h₃ : 0 ≤ (c + a) * (c + a),
  { nlinarith, },
  have h₄ := sq_nonneg (a * b + b * c + c * a),
  simp [sq, h₀, h₃, mul_add, add_mul] at h₄ ⊢,
  nlinarith [sq_nonneg (b - a),
            sq_nonneg (c - b),
            sq_nonneg (a - c)]
end
```

Although none of the examples above show AIs doing novel research, DeepMind's [AlphaTensor](#) model discovered a new algorithm for matrix multiplication which was faster than any designed by humans. While AlphaTensor was specifically developed for this purpose (as opposed to the more general systems discussed above), the result is notable because matrix multiplication is the key step in training neural networks.

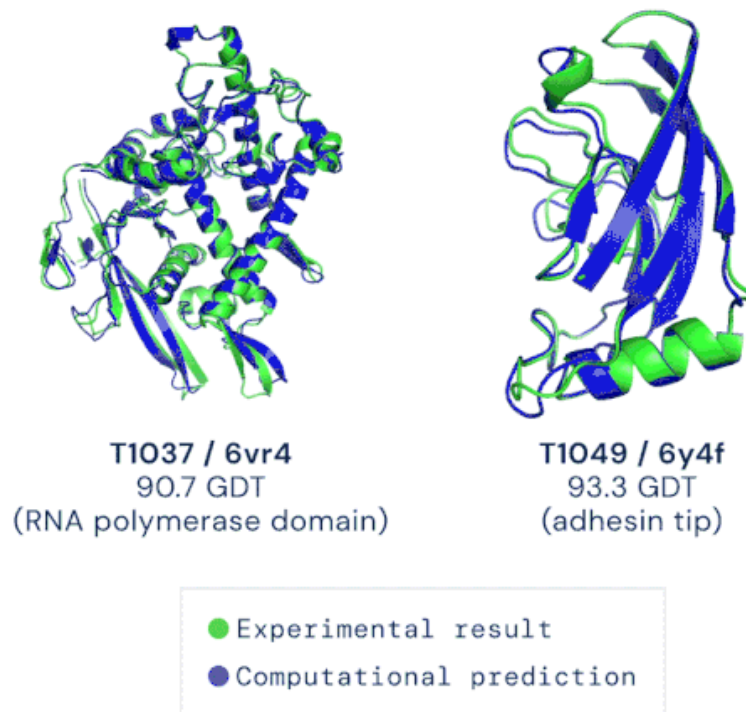
Life sciences

The biggest result thus far for AI in the life sciences has been the success of [AlphaFold 2](#), DeepMind's AI for predicting protein structures, which was described

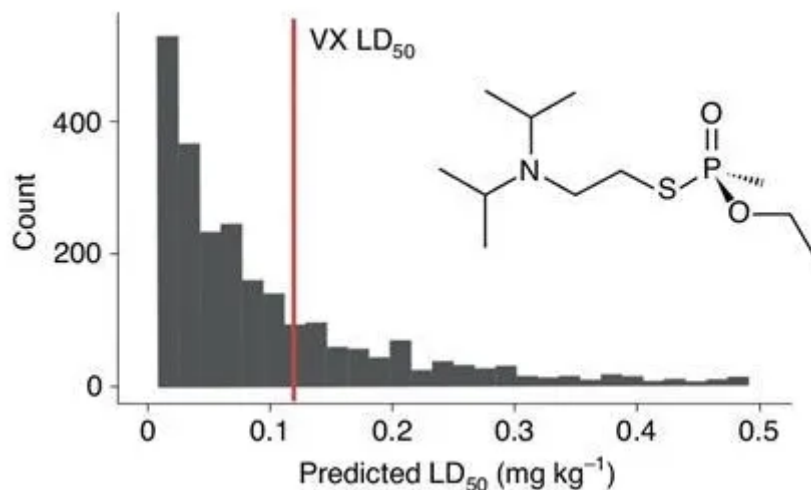
by experts in the field as having solved the protein folding problem (one of the core open problems in biology). While this claim may be somewhat overblown, it was nevertheless a huge leap forward compared with previous attempts:



Some examples of AlphaFold 2's predictions, compared with the experimental findings:



A more worrying advance comes from a team of drug development researchers, who reversed their algorithm for finding harmless drugs to show how it could potentially generate chemical weapons. The chart below shows the predicted toxicity of compounds generated by their AI; many of them have a lower lethal dose than VX (one of the most lethal toxins known).



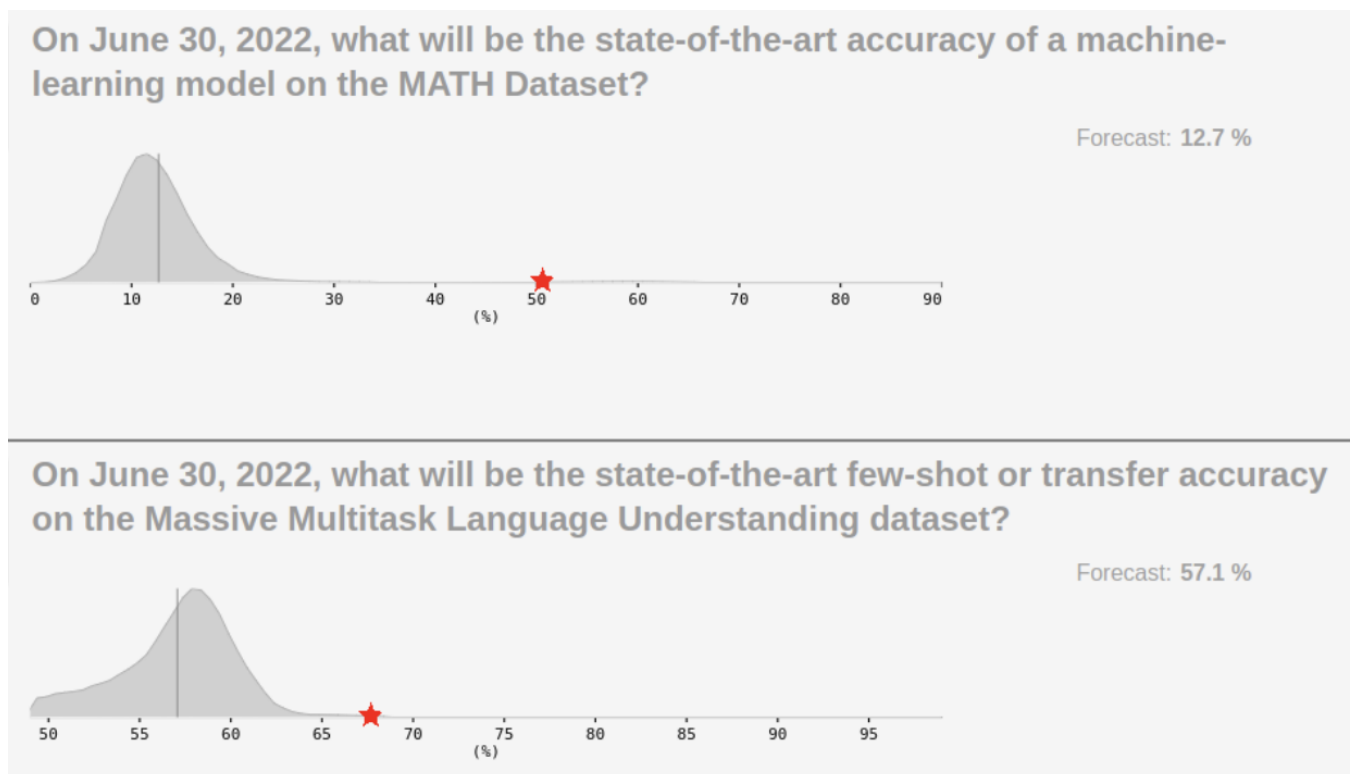
Source: Dual use of artificial-intelligence-powered drug discovery (Urbina et al., 2022).

Looking forward

Just to highlight how absurd the rate of progress in AI has been, note that around half of the breakthroughs I've mentioned happened in 2022 alone! Overall we're seeing AI becoming increasingly good at taking actions in complex environments, doing strategic thinking, and then connecting it to the real world. You might worry

that showcasing individual examples is a haphazard way to track this progress, compared with measuring more quantitative metrics. However, it's becoming difficult to design benchmark tasks which remain unsolved for more than a few years, even when they're intended to track long-term progress.

For example, consider the MATH dataset of challenging competition mathematics problems, released in 2021. Professional forecasters predicted that the best accuracy achieved by mid-2022 would be 12.7%; in fact, it was over 50%, well outside their confidence intervals. They also significantly underestimated progress on the 2020 Massive Multitask Language Understanding dataset, which covers elementary mathematics, US history, computer science, law, and more. These forecasters were not experts in ML, but predicted much faster progress than most experts did, and *still* dramatically underestimated progress on these two benchmarks (although on two other benchmarks their predictions were more accurate).



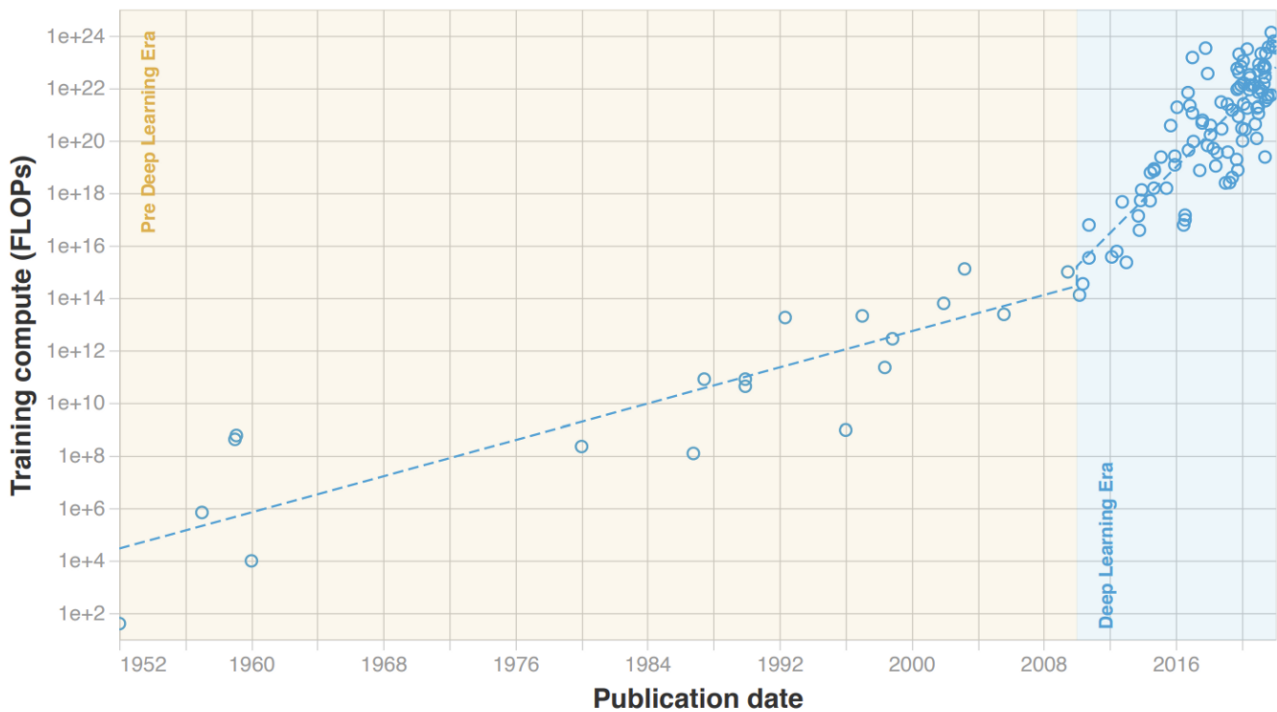
Source: [AI Forecasting: One Year In \(Steinhardt, 2022\)](#)

One reason that it's hard to make accurate forecasts is because the compute used in training machine learning systems has been scaling up so dramatically. To get a sense for this, see the graph below showing the computational resources used to train over 100 cutting-edge machine learning systems. Note in particular that the y-axis is on a logarithmic scale — the most compute-intensive system pictured on the

graph was trained using over 10 million times more compute than the biggest system from ten years earlier (around the start of the deep learning era).

Training compute (FLOPs) of milestone Machine Learning systems over time

n = 121



Source: [Compute Trends Across Three Eras of Machine Learning \(Sevilla et al., 2022\)](#)

The people who have come closest to correctly forecasting how fast progress would be have typically done so by focusing on compute scaling, even when it led to absurd-sounding consequences. In the 1990s, Hans Moravec predicted human-level AI by the 2030s, based on forecasting compute scaling. In 2005, Ray Kurzweil used a similar method to predict a “profound and disruptive transformation in human capability” (which he called the “singularity”) by 2045. In 2010, before the deep learning revolution, DeepMind co-founder Shane Legg predicted human-level AI by 2028 using compute-based estimates. OpenAI co-founder Ilya Sutskever, whose AlexNet paper sparked the deep learning revolution, was also an early proponent of the idea that scaling up deep learning would be transformative. Whether or not the specifics of these predictions are right, they all seem much more plausible now than they did a decade ago, especially when compared to the mainstream consensus that progress in AI would be nowhere near this fast. More recently, in 2020 a team at OpenAI developed scaling laws which claimed that scaling up the amount of compute used in large language models would predictably improve their performance, a claim which has largely been borne out since then (although there have been recent corrections to the specific scaling laws they proposed).

Another thing that many of these forecasters share is serious concern about the possibility that AGI won't be aligned with human values, and that this might lead to catastrophic outcomes. Shane started and led the DeepMind technical AGI safety team; more recently, Ilya has been spending 50% of his time on alignment research; and the core team behind the scaling laws paper founded Anthropic, an alignment-focused research lab. The possibility that we might be in the run-up to AGI naturally focuses the mind on the risks of bringing such a powerful technology into the world.