IN DIALOGUE WITH AI

LECTURE I / READING EDITION

Demystifying AI

Roberto A. Santiago

September 16, 2026
University of Portland

Slides, explanations, annotations, and sources

37 slides / 162 reveal steps

indialoguewithai.org

Contents

The slide text follows the original reveal steps. Contents entries, source titles, and recording references are linked.

Recording timings are approximate, based on the supplied lecture transcript.

Read the interactive edition online / Watch the recorded lecture

Origins & history

01 Demystifying AI 4
02 Do Not Be Afraid 5
03 Where I come from 6
04 A lifelong question 8
05 The photograph 11
06 We have been here before 15
07 The timeline 17
08 Two paths from shared roots 26

Learning from examples

09 What do we mean by intelligence? 31
10 Learning through feedback 33
11 What comes next? 36
12 Context changes the answer 38
13 Learning from feedback 41
14 Hold on to what you learned 45
15 Go with your gut 47
16 The scale behind the learning 51

Shaping an assistant

17 What does it actually produce? 54
18 After pretraining 57
19 Training for a useful response 60
20 Learning from a reward 63
21 Where the reward comes from 66
22 Chain of thought: a chance to check 71

Building the system

23 The model at the center 75

24 From training to use 78
25 Memory beyond the context window 82
26 Grounding answers in sources 84
27 Connecting evidence with GraphRAG 86
28 From replies to completed tasks 88
29 Several agents working together 91
30 The system we actually use 93

Cognitive parallels

31 A reason can miss the cause. 96
32 Remembering what fits. 99
33 Later information can rewrite the account. 102
34 Shared patterns, side by side. 105
35 The Cognitive Hypothesis 110

The conversation continues

36 A word from Claude 114
37 The conversation continues 117

SLIDE 01 / Origins & history

Demystifying AI

Understanding how AI is built gives us a clearer starting point for deciding how to live and work with it.

In this first lecture of In Dialogue with AI, I invite you to look inside the systems that now speak, write, reason, and act alongside us. These notes accompany the lecture I gave at the University of Portland on September 16, 2026.

We'll move from the history of these ideas to simple learning exercises, then to the larger machinery surrounding a modern language model. Finally, we'll ask what the resulting behavior suggests about cognition and our relationship with generative AI. You don't need a technical background to follow the argument. I want to make the mechanisms understandable enough for us to ask better questions about their capabilities, limitations, and human purpose.

The idea to carry forward

A useful conversation about AI begins with understanding what we have actually built.

In the recorded lecture

Lecture 0:13-2:24: Welcome and introductions.

Audience discussion 1:20:23-1:22:17: Audience question about finding information below the headlines.

A closer look

01 Why these notes point to original sources

When you encounter a claim about AI, I encourage you to follow it back to the technical papers and documentation. I want to understand the mechanisms and decisions behind a result, beyond the alarming or impressive headline. That's why I've included sources throughout these notes. You can explore the work for yourself and see where I'm interpreting a result and what the study directly establishes.

Recording 1:20:23-1:22:17 / Slide step 1

Slide text and reveals

Step 1 of 1

In Dialogue with AI

Lecture I

Demystifying AI

Roberto A. Santiago

September 16, 2026 · University of Portland

SLIDE 02 / Origins & history

Do Not Be Afraid

I want this lecture and the whole series to leave you with less fear and a greater sense that we can get involved and help determine how this turns out.

I promised myself I would start with these words: do not be afraid. We hear so many claims about AI, and I know those words can draw a few uncomfortable chuckles. But this is a big part of why I wanted to give these lectures.

If nothing else happens over the course of these six lectures, I want you to walk away feeling less afraid and more able to say: we can handle this. We should be involved. We can help determine how this turns out. That is the spirit I want to carry through everything we explore together.

The idea to carry forward

We can take AI seriously while approaching it with curiosity instead of fear.

In the slide. I start with a simple statement: do not be afraid.

In the recorded lecture

Lecture 2:24-3:04: My opening promise: less fear, more participation, and a sense that we can shape what happens next.

Audience discussion 1:12:58-1:15:23: Returning to the question of fear and my belief in people who want to help.

A closer look

01 Why I believe we can handle this

I'm a Mr. Rogers sort of guy. I believe there are lots of people who want to help and who are willing to do what needs to be done to prevent harm. When we talk about the dangers of these systems, I don't want us to stop researching them. I want us to keep working toward safer forms of AI and take responsibility for what we choose to do with them.

Recording 1:12:58-1:15:23 / Slide step 1

Slide text and reveals

Step 1 of 1

Do Not Be Afraid.

SLIDE 03 / Origins & history

Where I come from

I grew up programming computers and imagining a world where intelligent technology could help us do incredible things.

I started programming in second grade and took to it like a fish to water. These early computers, the VIC-20, Atari, Commodore 64, Amiga, and TRS-80, bring me right back to that excitement. When people in the room recognized them, I felt a little less old. These were things I could explore, program, and make do something new.

I bought into the whole vision: intelligent spaceships we could talk to, androids we could hang out with, and robots that could help us fight the Empire. And if we couldn't have all that, at least we could have a car that talked to us. We could all be David Hasselhoff. Underneath the joke was something I took seriously, even in grade school: I wanted to help build a world filled with intelligent technology that would let us do incredible and wonderful things.

The idea to carry forward

The motivating dream was useful intelligence woven into everyday life.

In the slide. I put the computers I grew up with alongside the fictional companions that helped me imagine what we might build.

In the recorded lecture

Lecture 3:08-4:40: How I started programming and came to imagine a world filled with helpful intelligence.

A closer look

01 A future I wanted to help build

This has been a passion of mine since grade school. As I got older, I didn't think I would live to see the kind of technology we are seeing now. That childhood hope is still with me: intelligence woven into everyday life, helping us do things we couldn't do before.

Recording 3:08-4:40 / Slide step 2

Slide text and reveals

Step 1 of 2

The open / Where I come from

A computer at home.

The machines that started it

VIC-20

Atari 400 / 800

Commodore 64

Amiga

TRS-80

Step 2 of 2

An imagination that went everywhere.

Star Trek

C-3PO & R2-D2

Knight Rider · KITT

This step replaces or removes earlier wording.

SLIDE 04 / Origins & history

A lifelong question

I've spent my career studying, building, and leading teams around technology, with one question running through it all: how does this help people?

In 1991, I ended up working at the National Science Foundation, where I met Paul Werbos. Through Paul, I learned to use neural networks for forecasting, optimization, and control. By 1993, I was applying those ideas to airlines, and around 2001 I had the chance to continue my research in AI, neural networks, and neuroscience. I was especially interested in how what we were learning about the brain connected with reinforcement learning. The photograph of Paul and me was taken on a visit just a few months before this lecture. Time has changed us quite a bit.

Along the way, I've also been building things: large information systems at Booz Allen, software and hardware systems at many scales, and teams to make them work. That practical work has continued from 2001 to the present, including my more recent return to AI through work with knowledge graphs at Nike. Studying these ideas and putting them to work have always belonged together for me. Paul gave me a principle that I've carried through all of it: technical brilliance has to serve humanity.

The idea to carry forward

Useful AI requires the work of understanding, building, and integrating systems into human activity.

In the slide. My early research and my later systems-building are parts of the same story. The recent photograph with Paul makes that continuity personal.

In the recorded lecture

Lecture 4:40-6:56: My work with Paul Werbos, the practical uses of neural networks, and the commitment to service I carried forward.

A closer look

01 The research thread

With Paul, I learned adaptive dynamic programming and neurocontrol: ways of using learning systems to make decisions and control processes. I went on to use neural networks for airline forecasting and later explored the relationship between spike-timing plasticity in the brain and reinforcement learning. That connection between biological intelligence and learning machines has stayed with me throughout my work.

Source 1 / Slide step 1

02 Putting ideas to work

At Booz Allen, I worked on knowledge models and large information systems. Over the years, I've led teams and built systems of all sorts of sizes and scales, including more recent work with knowledge graphs and graph networks in production. I bring that practical experience into this lecture because making a technique work is only part of the job. We also have to build something people can use.

Source 1 / Slide step 2

03 What useful meant in practice

We worked on making engines more fuel efficient, on flight-control systems that could adapt when something went wrong, and on economic forecasting. In the private sector, I did modeling and forecasting for airlines, insurers, and financial businesses. One very concrete example was predicting demand to help set airline ticket prices. When prices started changing daily, I was part of that, and I joked in the lecture that some people still hate me for it. I'm describing the research and applications we pursued here, including work that was still experimental.

Source 1 / Recording 4:40-6:07 / Slide step 1

04 The standard I carry forward

One thing Paul taught me has stayed with me: brilliance isn't worth much if the technology we build doesn't seek to serve humanity. I've held on to that through decades of building software, hardware, and teams. It is also why I want to ask more than what AI can do. I want to ask whose needs we are using it to meet.

Source 1 / Recording 6:30-6:56 / Slide step 2

Slide text and reveals

Step 1 of 2

The open / A lifelong question

The brain. The mind. The world.

1991

Learning with Paul Werbos National Science Foundation

1993

Neural networks, in practice Airline forecasting and optimization

2001

Brain meets machine Neuroscience and reinforcement learning

Paul Werbos & Roberto Santiago

Paul Werbos 1991

Step 2 of 2

Ideas made useful.

Booz Allen

Large information systems

2001-present

Leading teams. Building systems. At many scales

Nike

A return to AI Knowledge graphs in practice

How can technology help us solve human problems?

This step replaces or removes earlier wording.

Sources and further reading

1. The Road

Roberto A. Santiago / 2026 / Author profile / roberto.org

My account of my research and professional history, including NSF, BehavHeuristics, Booz Allen, the NW Computational Intelligence Lab, and later production work. The Road section gives the dated milestones.

SLIDE 05 / Origins & history

The photograph

When I think about AI, I think about these children. I want us to ask whose lives our work is helping.

You may have noticed that I don't wear shoes. That began as a spiritual practice during Lent about fifteen years before this lecture. I had been asked to give children a presentation about poverty around the world. Reading about children dying from preventable causes, while looking at my own life and remembering Paul's words about serving humanity, made me feel that I needed to live differently. Going barefoot became a way to express that solidarity.

This photograph hangs in our home. These are children my wife, Anne, met during her travels in Africa. They're grown-ups now, but we remember them all the time. At one point I was ready to leave technology completely. This journey is part of why I'm here talking with you, and why I think AI is as much about these children as it is about the technology itself.

I have a math degree, and I love mathematics. I can be excited by what AI is accomplishing there and still ask: could we bring comparable money, computing, and ingenuity to making sure children are fed and cared for? I want us to bring AI down to earth, understand what we have, and see where we have the power to shape what happens with it.

The idea to carry forward

The value of technological progress includes what it contributes to human lives.

In the slide. This is a photograph from our home, of children Anne met in Africa.

In the recorded lecture

Lecture 6:56-9:16: Why I go barefoot, the children in our photograph, and the human needs I want technology to serve.

A closer look

01 Child survival then and now

To put this concern in numbers, UNICEF's The State of the World's Children 2008: Child Survival reported close to 9.7 million deaths among children under five in 2006, more than 26,000 a day, mostly from preventable causes. WHO's May 2026 fact sheet puts the 2024 total at 4.9 million, about 9.3 children per minute when averaged over that leap year. I want to be precise about what that number means: it counts deaths from all causes. The accompanying UN release says most could be prevented through established, low-cost interventions and access to quality health care. We have made substantial progress, and the continuing scale of the need is still staggering.

Source 1 / Source 2 / Source 3 / Recording 7:28-7:37 / Slide step 1

02 Essential treatments can cost very little

What troubles me is that children still die from illnesses we know how to prevent and treat, even when some essential treatments cost only cents. WHO's 2024 diarrhea fact sheet reports about 444,000 deaths annually among children under five and describes oral rehydration salts as costing a few cents per treatment. UNICEF's pneumonia page reports more than 700,000 under-five deaths annually and describes antibiotic treatment for appropriate respiratory infections as typically costing less than US$0.50 for a full course. When I talk about those low prices, I mean the treatments themselves. Getting effective care to a child also requires timely diagnosis, delivery, clean water, nutrition, and sometimes additional treatment for severe illness. That is the practical work I want us to take seriously.

Source 4 / Source 5 / Recording 7:28-7:37 / Slide step 1

03 Whose problem are we solving?

I love seeing AI tackle ambitious mathematical problems. I also want us to bring that ambition to figuring out how children can be fed and cared for. What problem are we choosing to solve? Who will benefit? What are we willing to commit to making that benefit real? Those questions are part of what I mean when I ask what we can do with this technology.

Recording 8:08-9:16 / Slide step 1

04 The scale of the Navier-Stokes effort

My mathematics example is about the scale of effort we can bring to a problem. OpenAI's September 8, 2026 announcement reports about 10,000 concurrent agents in the group that produced its proposed Navier-Stokes result, which concerns a fluid subject to smooth external forcing. It discloses 130 billion output tokens for this problem and 300 billion across all attempted problems. At the standard Astra output prices listed on October 16 ($50-$75 per million tokens, depending on context length), those volumes correspond to $6.5-$9.75 million and $15-$22.5 million respectively. I'm using these as illustrative retail-price equivalents for output alone. The research used a different internal model; these figures do not establish its actual operating costs or include inputs, training, staffing, or other project costs.

Source 6 / Source 7 / Recording 8:37-8:52 / Slide step 1

05 After the lecture: hundreds of mathematical manuscripts

Since I gave this lecture, OpenAI's October 6, 2026 release has added another example of the scale of AI mathematics. As checked on October 16, its catalog contains 719 manuscripts in 372 related families, with results at different stages of verification. I'm not presenting that count as 719 independently verified discoveries. OpenAI describes average compute equivalent to about three hours of ChatGPT Pro thinking per result but gives no dollar total in this disclosure. I include this as a later development, following the September 16 lecture.

Source 8 / Source 9 / Slide step 1

Sources and further reading

1. The State of the World's Children 2008: Child Survival

UNICEF / 2008 / UN report / www.unicef.org

Printed page 1 reports the historical 2006 estimate and more than 26,000 daily under-five deaths. Provides context for the scale of preventable suffering; it does not establish a universal dollar cost per life saved.

2. Child mortality (under 5 years)

World Health Organization / 2026 / Fact sheet / www.who.int

Updated May 1, 2026. Reports 4.9 million under-five deaths in 2024. The annotation's 9.3 per minute is 4,900,000 divided by the leap year's 527,040 minutes. This total includes all causes, not only lack of food or medicine.

3. Progress in reducing child deaths slows as 4.9 million children under five die in 2024

UNICEF, WHO and UN IGME partners / 2026 / Joint report announcement / www.unicef.org

Release datelined March 18, 2026. Connects preventable deaths with low-cost interventions and access to quality health care. Notes that successive estimation rounds are revised, so the historical and current reports are not used here to calculate a precise percentage decline.

4. Diarrhoeal disease

World Health Organization / 2024 / Fact sheet / www.who.int

Dated March 7, 2024. Gives 443,832 annual under-five deaths and the low price of oral rehydration salts, while also describing other prevention and treatment needs. The page does not label the mortality figure as deaths occurring in 2024.

5. Pneumonia in children

UNICEF Data / Accessed October 16, 2026 / Child health data and explanation / data.unicef.org

Reports over 700,000 annual under-five deaths and antibiotic treatment typically costing less than US$0.50. Explains the need for clinical assessment and appropriate care. These headline estimates are not combined with the WHO diarrhea figure into a single-year mortality total.

6. On the Navier-Stokes Millennium Prize Problem

OpenAI / 2026 / Research announcement / openai.com

September 8 announcement. Reports the agent count, external-forcing construction, and separate output-token totals for Navier-Stokes and all attempted problems. It does not provide an itemized dollar cost.

7. OpenAI API pricing

OpenAI / 2026 / Pricing reference / developers.openai.com

Standard Astra output rates checked October 16: $50 per million tokens for short context and $75 for long context. Multiplying by the reported token volumes gives illustrative retail equivalents, not the unreleased research model's measured cost.

8. Sharing AI progress in mathematics

OpenAI / 2026 / Research announcement / openai.com

October 6 release and compute disclosure. This postdates the lecture and is included as a later update.

9. OpenAI mathematics catalogue

OpenAI / 2026 / Research repository / github.com

Catalogue and verification status checked October 16. Groups related manuscripts into families and explains the compute comparison.

SLIDE 06 / Origins & history

We have been here before

An extraordinary claim from 1988 reveals that today's excitement about neural networks has a long history.

The phrase "more important than the atom bomb" appears before its date or speaker. Without that context, it could sound like a contemporary claim about AI. The reveal places it in 1988, when Scientific American reported Jasper Lupo of DARPA describing the potential importance of neural networks.

That change of context is why I wanted to show you this. Earlier researchers and funders also saw enormous possibilities in learning machines. Their enthusiasm existed alongside practical applications, serious obstacles, and uncertainty about what would work. I want us to take a longer view of the present: what has actually changed, which problems have been solved, and which hopes or difficulties have returned? I also remember meeting Lupo, so this history has a personal connection for me.

The idea to carry forward

Historical context helps us separate a compelling promise from the mechanisms and evidence behind it.

In the slide. The dramatic quotation comes first; the 1988 attribution and period report change how we read it.

In the recorded lecture

Lecture 9:16-10:37: The 1988 quotation and recurring promises about neural networks.

A closer look

01 Where the quotation comes from

June Kinoshita's article Neural Networks at Work reports Lupo's statement at a neural-network conference in San Diego. The wording appears on printed page 135 of the November 1988 Scientific American. The article identifies him as an assistant director at DARPA.

Source 1 / Recording 9:16-10:04 / Slide step 2

02 What the pictured report is

The DARPA Neural Network Study assessed neural-network technology and research directions during 1987-1988. The pictured final technical report is dated March 22, 1989. It supplies context for that period of research, while the Scientific American article supplies the quotation.

Source 2 / Slide step 2

03 Applications were already emerging

Kinoshita also reported work on industrial inspection, credit assessment, airline decisions, and process control. The historical comparison is therefore more interesting than an old prediction: neural networks already had practical ambitions, even as claims about their future ran far ahead of their demonstrated abilities.

Source 1 / Recording 10:04-10:37 / Slide step 2

Slide text and reveals

Step 1 of 2

So where did this come from?

“more important than the atom bomb”

Step 2 of 2

Jasper Lupo · DARPA

1988

DARPA Neural Network Study A DARPA assessment of neural-network technology. 1987-1988 study · Final report March 1989

Sources and further reading

1. Neural Networks at Work

June Kinoshita / 1988 / Magazine article / www.scientificamerican.com

Contemporaneous reporting in Scientific American 259(5), pages 134-135. Lupo’s quotation and title appear on page 135; page 134 describes emerging applications. DOI: 10.1038/scientificamerican1188-134.

2. DARPA Neural Network Study

MIT Lincoln Laboratory for DARPA / 1989 / Research report / s3.amazonaws.com

Technical Report 840 / ESD-TR-88-311, dated March 22, 1989, covering the October 1987-February 1988 study. The cover and foreword explain the relationship between the study dates and the final report.

SLIDE 07 / Origins & history

The timeline

Today’s AI draws on a century of overlapping ideas about computation, brains, learning, feedback, and language.

The timeline follows people and ideas rather than presenting AI as a recent invention. Fiction gives us a name for manufactured workers; mathematics clarifies what computation can do; neuroscience and cybernetics connect biological processes with formal models. Neural learning, symbolic reasoning, better hardware, and new ways to represent language develop through overlapping communities.

Two labels receive special emphasis because they will matter throughout the lecture. Backpropagation calculates how changes inside a network affect an error or objective. Reinforcement learning organizes learning around actions and evaluated outcomes. These are complementary ideas: neural reinforcement-learning systems can use backpropagation to change their networks. The later exercises make this distinction concrete.

The final milestones bring these foundations into large language models and conversational systems. The collage at the end conveys the activity visible at the time of the lecture. The dates mark selected publications, demonstrations, and releases, rather than asserting that each idea began at exactly that moment or belongs to one inventor.

The idea to carry forward

Modern AI is an accumulation of methods and research traditions, with learning from error and reward central to this lecture’s story.

In the slide. Portraits identify the people; the small computer, network, and brain images identify recurring themes. Equal spacing separates events for readability, rather than measuring the years between them.

In the recorded lecture

Lecture 10:37-22:13: The historical timeline, from R.U.R. to conversational AI.

A closer look

01 A cultural question before a technical field

Čapek’s R.U.R. depicts manufactured workers whose revolt threatens humanity. Its robots are artificial living workers, rather than today’s familiar metal machines. The play was published in 1920, with premieres in 1921. In his own later account, Karel credits his brother Josef with suggesting the word robot. The story raises questions about labor, dependence, and responsibility before modern AI research takes shape. I read its promise of tireless manufactured labor as a warning about valuing productivity while forgetting the dignity of people.

Source 1 / Source 2 / Recording 10:37-12:50 / Slide step 2

02 Feedback connects animal and machine

Wiener's 1948 book Cybernetics: Or Control and Communication in the Animal and the Machine connected the study of communication and feedback across biological and engineered systems. Feedback means that the consequences of a process influence what it does next. A feedback controller can regulate a system without learning, so feedback by itself does not necessarily imply changing a neural network's parameters.

Source 3 / Slide step 6

03 A model inspired by brains

Rosenblatt's perceptron made learning through adjustable connections a concrete research program. His work included rules for changing connections through experience. The later challenge was effective training of more general multilayer networks. The multilayer network pictured here is a teaching illustration, rather than a reconstruction of Rosenblatt's 1958 machine or the anatomy of a human brain.

Source 4 / Recording 13:51-14:47 / Slide step 10

04 Backpropagation: how should connections change?

Backpropagation efficiently works backward through a computation to calculate how an error or objective depends on its adjustable parameters. An optimizer can use those calculations to change the network. The 1974 milestone recognizes Werbos's work on backward differentiation and learning; the 1986 paper by Rumelhart, Hinton, and Williams helped bring the approach to a much wider audience. The mathematical history includes earlier contributions across several fields.

Source 5 / Source 6 / Recording 15:20-15:59 / Slide step 13

05 Reinforcement learning: how well did the action work?

The 1983 Barto, Sutton, and Anderson study used two adaptive components to learn a difficult control task. One selected actions; the other helped evaluate their consequences. This is an early actor-critic result, rather than the invention of all reinforcement learning. It illustrates the lecture's important distinction between being given a desired answer and receiving an evaluation of what happened after acting.

Source 7 / Recording 15:59-17:43 / Slide step 14

06 From generating language to helping a user

The arrival of conversational AI involved more than making language models larger. The InstructGPT paper describes learning from demonstrations and then from human preferences to make responses better match user intent. This distinction between a pretrained language model and a model shaped to assist people becomes a central thread in the next sections.

Source 8 / Slide step 31

07 Why emphasize these two tools?

I want you to remember backpropagation and reinforcement learning because they give us a manageable way into the training story. One explains how an objective can guide changes to a network; the other explains learning from the consequences of actions. These complementary tools sit within a broad, collaborative research history. We'll see how they contribute to different stages of training in the next sections.

Source 5 / Source 6 / Source 7 / Source 8 / Recording 15:59-17:43 / Slide step 14

08 What changed when the old ideas became practical?

Ideas alone were not enough. Hardware, training methods, data, and network design had to work together. AlexNet's 2012 paper gives a concrete example: GPU computation helped make a large convolutional vision network practical. The 2017 Transformer paper introduced an attention-based architecture that made sequence training more parallelizable. Together, these developments show computing capacity and network design advancing alongside one another, expanding what models could do with images and language. I know from my own research how expensive this hardware can be. That's one reason I think access to computing deserves attention as a possible policy lever. It may give us a way to influence development, though I'm not offering it as a guarantee of control.

Source 9 / Source 10 / Recording 18:11-21:10 / Recording 19:10-19:45 / Slide step 27

A century of ideas, with sources

The years mark selected publications or events, not the beginning of every idea.

1920 / We imagine the robot

Karel Čapek · R.U.R. / Culture / Slide step 2

Čapek, K. R.U.R. (Rossum's Universal Robots), published 1920; English text translated by Paul Selver and adapted for the English stage by Nigel Playfair. The play depicts manufactured workers who rebel, leaving Alquist as the last human survivor.

Čapek, K. "About the Word Robot" (O slově robot), Lidové noviny, December 24, 1933. The author's own account credits his brother Josef with suggesting the name; English reproduction preserved by the University of Oregon.

Karel Čapek Memorial, "Karel Čapek": publication in 1920 and theatrical premieres in 1921; Josef's role in naming the robots.

1936 / A machine that can compute

Alan Turing · On Computable Numbers / The machine / Slide step 3

Turing, A. M. "On Computable Numbers, with an Application to the Entscheidungsproblem." Proceedings of the London Mathematical Society, s2-42, 230-265.

1943 / A neuron, on paper

Warren McCulloch & Walter Pitts / The brain / Slide step 4

McCulloch, W. S., and Pitts, W. "A Logical Calculus of the Ideas Immanent in Nervous Activity." Bulletin of Mathematical Biophysics 5, 115-133 (1943).

1945 / A program stored in memory

John von Neumann · The EDVAC report / The machine / Slide step 5

von Neumann, J. First Draft of a Report on the EDVAC, June 30, 1945. Original report, reprinted in IEEE Annals of the History of Computing.

1948 / Feedback connects animal and machine

Norbert Wiener / Cybernetics · Feedback and control / Slide step 6

Wiener, N. Cybernetics: Or Control and Communication in the Animal and the Machine, first published 1948. MIT Libraries identifies the book as a landmark account of control, feedback and communication in biological and electromechanical systems.

MIT Press, Cybernetics: confirms first publication in 1948 and its foundational role in the multidisciplinary study of systems with feedback loops.

1950 / Can machines think?

Alan Turing · Computing Machinery and Intelligence / The machine / Slide step 7

Turing, A. M. “Computing Machinery and Intelligence.” Mind LIX(236), 433-460 (1950).

1952 / How a real neuron fires

Alan Hodgkin & Andrew Huxley / The brain / Slide step 8

Hodgkin, A. L., and Huxley, A. F. “A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve.” Journal of Physiology 117, 500-544 (1952).

1956 / Artificial intelligence gets a name

The Dartmouth workshop / A field takes shape / Slide step 9

McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence,” August 31, 1955, for the summer 1956 meeting.

1958 / The perceptron

Frank Rosenblatt / Brain meets machine / Slide step 10

Rosenblatt, F. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” Psychological Review 65(6), 386-408 (1958).

1965 / Between true and false

Lotfi Zadeh · Fuzzy logic / Reasoning in degrees / Slide step 11

Zadeh, L. A. “Fuzzy Sets.” Information and Control 8(3), 338-353 (1965).

1969 / Perceptrons under scrutiny

Marvin Minsky & Seymour Papert / Symbols and learning / Slide step 12

Minsky, M., and Papert, S. A. Perceptrons: An Introduction to Computational Geometry. MIT Press, 1969.

1974 / BACKPROPAGATION

Paul Werbos · Harvard dissertation / Learning from error / Slide step 13

Werbos, P. J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Harvard doctoral dissertation, August 1974.

Werbos, P. J. “Backwards Differentiation in AD and Neural Nets: Past Links and New Opportunities” (author’s retrospective).

1983 / REINFORCEMENT LEARNING

Andrew Barto, Richard Sutton & Charles Anderson / An actor. A critic. A reward. / Slide step 14

Barto, A. G., Sutton, R. S., and Anderson, C. W. “Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems.” IEEE Transactions on Systems, Man, and Cybernetics SMC-13, 834-846 (1983).

Richard Sutton’s author bibliography describes the associative search element and adaptive critic element.

1986 / Backpropagation reaches a wider field

David Rumelhart, Geoffrey Hinton & Ronald Williams / Connectionism / Slide step 15

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. “Learning Representations by Back-Propagating Errors.” Nature 323, 533-536 (1986).

1988 / Learning from prediction error

Richard Sutton · Temporal-difference learning / Reinforcement learning / Slide step 16

Sutton, R. S. “Learning to Predict by the Methods of Temporal Differences.” Machine Learning 3, 9-44 (1988).

1988 / We have been here before

The DARPA Neural Network Study / The flag / Slide step 17

DARPA Neural Network Study, October 1987-February 1988. MIT Lincoln Laboratory Technical Report 840 / ESD-TR-88-311, final report dated March 22, 1989; foreword dated June 1988.

1989 / Reading handwritten digits

Yann LeCun and colleagues / Convolutional networks / Slide step 18

LeCun, Y., et al. “Backpropagation Applied to Handwritten Zip Code Recognition.” Neural Computation 1(4), 541-551 (1989).

1990 / Neural Networks for Control

W. Thomas Miller III, Richard Sutton & Paul Werbos / Engineering and computer science / Slide step 19

Miller, W. T. III, Sutton, R. S., and Werbos, P. J., eds. Neural Networks for Control. MIT Press. Commonly cited as 1990; publisher catalog lists the hardcover release as January 10, 1991.

Bibliothèque nationale de France catalog records copyright 1990.

1997 / A machine wins at chess

IBM Deep Blue · Garry Kasparov / Search / Slide step 20

IBM, “Deep Blue”: the May 1997 match against Garry Kasparov ended 3.5-2.5 in Deep Blue’s favor under standard tournament controls.

2006 / Depth returns

Geoffrey Hinton and colleagues · Deep belief networks / Deep learning / Slide step 21

Hinton, G. E., Osindero, S., and Teh, Y.-W. “A Fast Learning Algorithm for Deep Belief Nets.” Neural Computation 18(7), 1527-1554 (2006).

2007 / A new engine for the work

NVIDIA CUDA / Programmable GPUs / Slide step 22

NVIDIA CUDA Toolkit Archive: CUDA Toolkit 1.0, June 2007, and 1.1, December 2007.

2011 / A machine wins Jeopardy!

IBM Watson / Language and statistics / Slide step 23

IBM, “Watson, Jeopardy! Champion”: Watson defeated Brad Rutter and Ken Jennings in the February 2011 exhibition.

2012 / The image recognition breakthrough

AlexNet · ImageNet / Deep learning on GPUs / Slide step 24

Krizhevsky, A., Sutskever, I., and Hinton, G. E. “ImageNet Classification with Deep Convolutional Neural Networks.” Advances in Neural Information Processing Systems 25 (2012).

2013 / Words as numbers

word2vec / Representing language / Slide step 25

Mikolov, T., Chen, K., Corrado, G., and Dean, J. “Efficient Estimation of Word Representations in Vector Space.” arXiv:1301.3781 (2013).

Bengio, Y., et al. “A Neural Probabilistic Language Model.” JMLR 3, 1137-1155 (2003), an earlier learned word-representation model.

2015-16 / From Atari to Go

DeepMind · DQN and AlphaGo / Learning and search / Slide step 26

Mnih, V., et al. “Human-Level Control Through Deep Reinforcement Learning.” Nature 518, 529-533 (2015). Silver, D., et al. “Mastering the Game of Go with Deep Neural Networks and Tree Search.” Nature 529, 484-489 (2016).

AlphaGo paper, 2016.

2017 / Attention Is All You Need

The transformer / A new architecture / Slide step 27

Vaswani, A., et al. “Attention Is All You Need.” Advances in Neural Information Processing Systems 30 / arXiv:1706.03762 (2017).

2018 / GPT-1 and BERT

Pretraining language models / Language at scale / Slide step 28

Radford, A., et al. “Improving Language Understanding by Generative Pre-Training” (2018). Devlin, J., et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv:1810.04805 (2018).

BERT preprint, submitted October 11, 2018; conference publication followed in 2019.

Earlier neural language modeling: Bengio et al. (2003).

2019 / GPT-2

Longer passages, broader patterns / Language at scale / Slide step 29

OpenAI, “Better Language Models and Their Implications,” February 14, 2019; Radford et al., “Language Models Are Unsupervised Multitask Learners.”

2020 / GPT-3

A larger model, a wider range of tasks / Language at scale / Slide step 30

Brown, T. B., et al. “Language Models Are Few-Shot Learners.” arXiv:2005.14165 / Advances in Neural Information Processing Systems 33 (2020).

2022 / The conversation reaches the room

ChatGPT / An everyday interface / Slide step 31

OpenAI, “Introducing ChatGPT,” November 30, 2022.

2024-25 / Reasoning gets more time

o1 and DeepSeek-R1 / Learning to reason / Slide step 32

OpenAI, “Learning to Reason with LLMs,” September 12, 2024. DeepSeek-AI, “DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning,” arXiv:2501.12948 (2025).

DeepSeek-R1 technical report, January 2025.

Today / An explosion of activity

A selection of companies and model families shown in the September 2026 lecture, not a ranking or a complete market map. / In dialogue with AI / Slide step 33

NVIDIA · Nemotron: official model information

Cohere · Command: official model information

xAI · Grok: official model information

Google · Gemini / Gemma: official model information

Google · Gemini / Gemma: official model information

Baidu · ERNIE: official model information

Microsoft · Phi: official model information

Anthropic · Claude: official model information

OpenAI · GPT: official model information

DeepSeek · DeepSeek: official model information

Mistral AI · Mistral: official model information

Alibaba · Qwen: official model information

Meta · Llama: official model information

Amazon · Nova: official model information

IBM · Granite: official model information

AI21 · Jamba: official model information

Tencent · Hy / Hunyuan: official model information

Tencent · Hy / Hunyuan: official model information

Sources and further reading

1. R.U.R. (Rossum's Universal Robots)

Karel Čapek; English version by Paul Selver and Nigel Playfair / 1920; English edition 1923 / Play / www.gutenberg.org

The original fictional work behind the timeline's first milestone. The opening explains the manufactured workers; the final acts and epilogue develop the revolt and its aftermath.

2. About the Word Robot

Karel Čapek / 1933 / Essay / scholarsbank.uoregon.edu

Čapek's own account, published in Lidové noviny on December 24, 1933, credits Josef Čapek with suggesting the name. English reproduction preserved by the University of Oregon.

3. Cybernetics: Or Control and Communication in the Animal and the Machine

Norbert Wiener / 1948 / Book / mitpress.mit.edu

Wiener's foundational book connects feedback, control, and communication. The publisher page confirms first publication in 1948 and links an open-access reissue of the expanded 1961 edition.

4. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain

Frank Rosenblatt / 1958 / Paper / doi.org

The original Psychological Review paper, volume 65, pages 386-408, develops the perceptron as a model of learning and recognition. Supports the brain-inspired learning milestone.

5. Backwards Differentiation in AD and Neural Nets: Past Links and New Opportunities

Paul J. Werbos / 2006 / Paper / link.springer.com

Werbos's retrospective explains backward differentiation and its connections with neural learning and control. Section 2 discusses his 1974 dissertation; the introduction places the technique within several interacting research traditions. Published in 2006; the timeline also links the author manuscript.

6. Learning Representations by Back-Propagating Errors

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams / 1986 / Paper / doi.org

The influential Nature paper, volume 323, pages 533-536, describes how backpropagation trains hidden units to learn useful internal representations.

7. Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems

Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson / 1983 / Paper / doi.org

The original learning-control study develops an associative search element and an adaptive critic, illustrated by a cart-pole balancing task. Supports the timeline's early actor-critic milestone.

8. Training Language Models to Follow Instructions with Human Feedback

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 2 and Section 3 describe supervised fine-tuning, preference collection, and reinforcement learning from human feedback. Supports the transition from broad language modeling toward assistant behavior.

9. ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton / 2012 / Paper / proceedings.neurips.cc

The original NeurIPS paper describes a deep convolutional image classifier and the GPU implementation used to accelerate training. See the abstract and Section 3.2 on training on multiple GPUs.

10. Attention Is All You Need

Ashish Vaswani and colleagues / 2017 / Paper / arxiv.org

The eight-author paper introduces the Transformer. The abstract and Sections 1 and 3 explain attention-based sequence modeling and the role of parallel computation, supporting the timeline's architecture milestone.

SLIDE 08 / Origins & history

Two paths from shared roots

Language models are one major destination of neural learning, while control, optimization, and adaptive engineering remain another living tradition.

The branching picture condenses the chronology into two different approaches to intelligence. Symbolic AI works with explicit representations, rules, and search. Connectionism learns through networks of adjustable connections. The perceptron makes the meeting of computation and brain-inspired learning visible; the later debate over its limitations helps explain why these approaches developed distinct communities and priorities.

As learning methods advanced, neural networks became increasingly important within mainstream AI and computer science. The upper route follows that development toward recognition, language, and eventually large language models. The merger represents a change in research practice across the field. It does not mean that symbolic rules were physically combined with neural weights to manufacture an LLM.

The lower route keeps a second history in view: using learned models to control changing physical systems, optimize decisions, and adapt over time. Neural Networks for Control anchors that tradition around the beginning of the 1990s. Its dashed continuation matters because this work did not end when language models became prominent. The branches continue to exchange ideas, and engineering problems still motivate methods whose purpose extends well beyond producing language.

For me, the engineering branch is a reason for hope and a research priority. Think about energy use, forecasting, control, and adaptive systems: their value depends on reliable behavior in the world. I want us to invest in bringing more of that knowledge into useful AI systems while being more selective about deployment. There is work to do here; we can’t simply transfer a guarantee from a physical controller to an open-ended conversation. When I look at the money going into selling AI products and military applications, I want to see more investment in stability, safety, and fail-safe behavior.

The idea to carry forward

LLMs occupy an important part of the learning landscape, while a broader engineering toolbox continues to develop alongside them.

In the slide. Follow both endpoints: the upper line leads toward modern LLMs; the lower dashed line keeps control, optimization, and adaptive systems visibly in motion.

In the recorded lecture

Lecture 22:13-25:54: Two research paths and the case for investing in safety and control.

Audience discussion 1:15:38-1:17:52: Audience question about transferring safeguards from control engineering.

A closer look

01 Reading the fork

This is a selective map of research traditions. The 1955 Dartmouth proposal already included both language-related questions and neuron nets. Minsky and Papert's 1969 Perceptrons examined limitations of particular perceptron classes and became a focal point of debate. It did not establish that every multilayer network was incapable of learning, and symbolic AI did not begin with that book.

Source 1 / Source 2 / Slide step 2

02 Why the hand-built approach met limits

A system of explicit rules can work well within a carefully defined domain. Extending it into a changing world introduces exceptions, ambiguous cases, and interactions among rules. In his 1990 essay, Minsky describes limitations on both the symbolic and connectionist sides and argues for systems combining different representations. The historical division was never a complete separation of ideas.

Source 3 / Slide step 3

03 The book at the engineering branch

Neural Networks for Control, edited by W. Thomas Miller III, Richard S. Sutton, and Paul J. Werbos, brings learning together with control problems, including robotics and motion planning. The lecture timeline uses its commonly cited 1990 date; MIT Press lists the hardcover release as January 10, 1991. The pictured cover belongs to the later paperback. The book identifies a research program, rather than an exact day when disciplines split.

Source 4 / Slide step 4

04 The route to language models

Large neural language models learn from text; later training can shape how they respond to people. GPT-3 and InstructGPT make that distinction concrete: one paper studies a large pretrained language model, while the other adds demonstrations and human-preference training. Reinforcement learning is important to many assistant models, but it is not the explanation for every stage of language-model training.

Source 5 / Source 6 / Slide step 5

05 An ongoing engineering tradition

A robot must cope with changing physical conditions, imperfect measurements, and constraints on safe action. The safe-control-gym study provides a concrete example of researchers comparing traditional control, learning-based control, and reinforcement learning on shared tasks. It shows both continued engineering development and active exchange among these communities. The parallel line therefore represents a larger set of problems and tools, not an isolated body of work with no connection to AI.

Source 7 / Slide step 6

06 Safety means specifying what must stay safe

The engineering lesson is to define the system, its operating conditions, and the constraints it must respect, then evaluate what happens under disturbances. The safe-control-gym benchmark makes this concrete with simulated cart-pole and quadrotor tasks. I want that discipline to influence AI development more strongly. A result established for a defined control problem does not by itself establish the safety of a general language model.

Source 7 / Recording 23:02-25:54 / Slide step 6

07 Why the transfer is difficult

An audience member asked how those engineering safeguards could carry over to modern AI. Consider the difference between a system whose dynamics and constraints we can describe mathematically and a confused person seeking good customer service. In the second case, even deciding what counts as success is difficult. I want us to develop systems that can keep learning through guided experience with people. That's a direction I'm proposing, with safety still something we would have to establish. Continued learning alone would not settle it.

Source 7 / Recording 1:15:38-1:17:52 / Slide step 6

Slide text and reveals

Step 1 of 6

So where did this come from?

Brain meets machine.

The perceptron brings a biological idea into a working machine.

Brain

Machine

1958 Perceptron

Step 2 of 6

Two approaches to intelligence.

Symbolic AI and connectionism pursue different ways to build intelligence.

Symbolic AI Symbols · rules

1969 · Minsky & Papert

Connectionism

This step replaces or removes earlier wording.

Step 3 of 6

Learning finds new momentum.

Backpropagation and reinforcement learning carry connectionism forward.

Backpropagation Reinforcement learning

Limits of hand-coded knowledge

This step replaces or removes earlier wording.

Step 4 of 6

Connectionism branches again.

One route pursues language and recognition. Another develops control and optimization.

Computer science Recognition · language

Engineering Control · optimization Adaptive systems

Neural Networks for Control Miller · Sutton · Werbos

This step replaces or removes earlier wording.

Step 5 of 6

Neural learning reshapes AI.

The computer-science route enters mainstream AI and leads toward modern LLMs.

Mainstream AI Neural networks · RL

Leads to modern LLMs

This step replaces or removes earlier wording.

Step 6 of 6

Two living traditions.

LLMs are one destination. The engineering toolbox keeps growing.

A parallel tradition, still evolving

This step replaces or removes earlier wording.

Sources and further reading

1. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence

John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon / 1955 / Research proposal / www-formal.stanford.edu

The August 31, 1955 proposal for the 1956 meeting lists language and neuron nets among its research topics. It provides direct evidence that the field's early agenda encompassed several approaches.

2. Perceptrons: An Introduction to Computational Geometry

Marvin Minsky and Seymour A. Papert / 1969 / Book / mitpress.mit.edu

The mathematical analysis behind the perceptron-scrutiny milestone. Its results concern specified classes of perceptrons, rather than a universal impossibility of neural learning.

3. Logical vs. Analogical, or Symbolic vs. Connectionist, or Neat vs. Scruffy

Marvin Minsky / 1990; reprinted 1991 / Essay / web.mit.edu

Minsky's own account of the traditions and their limitations. The opening discussion and section Heuristic Connections of Rule-Based Systems explain his case for combining approaches.

4. Neural Networks for Control

W. Thomas Miller III, Richard S. Sutton, and Paul J. Werbos, editors / 1990/1991 / Book / mitpress.mit.edu

The publisher's description identifies learning control, robotics, motion planning, and benchmark control problems. Its edition information distinguishes the January 1991 hardcover from the March 1995 paperback.

5. Language Models Are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

Describes GPT-3 and its evaluation across language tasks. Section 2 covers the model and training data, establishing a concrete point on the large neural language-model route.

6. Training Language Models to Follow Instructions with Human Feedback

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 2 and Section 3 show how demonstrations and reinforcement learning from human feedback are applied after a pretrained model, clarifying the role of RL along the route to assistants.

7. safe-control-gym: A Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning in Robotics

Zhaocong Yuan and colleagues / 2021; revised 2022 / Paper / arxiv.org

An original benchmark study comparing traditional, learning-based, and reinforcement-learning control on cart-pole and quadrotor tasks. Illustrates the continuing engineering route and exchange between research communities.

SLIDE 09 / Learning from examples

What do we mean by intelligence?

I use intelligence to mean finding a fitting response to an unfamiliar situation and improving through feedback.

My working definition has two parts: responding to a situation that has not been encountered before, and getting better at doing so through feedback. A fitting response depends on the situation. Repeating a familiar answer is not enough when the circumstances have changed.

This definition gives us a practical question to pursue: how can a machine become better at producing useful responses? The next slides make learning tangible, first as a network adjusting its connections and then as word games that let us experience the pull of a learned pattern. I offer this as a working definition for our discussion, not as a claim that researchers have agreed on one complete definition of intelligence.

You don't need mathematics to follow these exercises. In the room, I invited everyone to become a "baby LLM": to experience how feedback changes an expectation, then see how the resulting expectation can go wrong. You can follow the same exercises here.

The idea to carry forward

Ask both whether a response fits the situation and how feedback shaped the ability to produce it.

In the slide. The same sentence stays on screen while emphasis moves from a fitting response to learning from feedback.

In the recorded lecture

Lecture 25:54-26:37: A practical definition of intelligence and the move to learning through feedback.

A closer look

01 A working definition

I offer this definition as a useful lens for our discussion. It does not establish a test of consciousness, human equivalence, or every capacity that might count as intelligence.

Slide step 1

02 Why the emphasis moves to feedback

The visual shifts attention from responding to improving. That is the bridge from the history of AI to the learning mechanism examined next.

Slide step 2

Slide text and reveals

Step 1 of 2

02 / Now, how does this work?

A working definition

Intelligence is the ability to come up with a fitting response to a situation you have never seen before, and to get better at it from feedback.

SLIDE 10 / Learning from examples

Learning through feedback

Training uses feedback about responses to adjust the numerical connections inside a network.

Follow a single cycle. An example enters the network. Its current connections shape a response. That response is evaluated, and the resulting feedback guides an adjustment. Repeating this cycle over many examples can change which responses the network tends to produce.

The adjustable numbers are parameters, including connection weights. In the diagram, thicker and thinner lines make those changes visible. Backpropagation helps calculate how changes to parameters would affect an error measure; an optimizer uses that information to make updates. The small network is a teaching model, not a literal picture of a complete language model.

Keep training separate from using an already-trained model. During ordinary generation, a model applies its existing weights to the available context. Giving it a new sentence, document, or correction can change its next response without changing those weights. This distinction will matter when we start learning patterns from words.

The idea to carry forward

Training changes parameters; new context changes what an already-trained model has to work with.

In the slide. Follow input, response, feedback, then the changed thickness of the connections.

In the recorded lecture

Lecture 26:37-27:11: The feedback loop: an output, an evaluation, and an adjustment to weights.

Audience discussion 1:08:03-1:09:33: Discussion returns to frozen weights and adaptation through conversation.

A closer look

01 Where backpropagation fits

Backpropagation supplies information about how an error depends on the network’s parameters. That makes guided weight adjustment possible; it does not decide by itself what should count as a good response.

Source 1 / Slide step 5

02 Feedback is broader than a person saying “wrong”

A known target, a calculated error, or a reward can supply a learning signal. The returning arrow groups these ideas to explain their purpose, while later slides distinguish their roles.

Source 1 / Source 3 / Slide step 4

03 Context is not a weight update

Brown and colleagues evaluated GPT-3 with task descriptions and examples in the prompt, without gradient updates or fine-tuning. A changed response in conversation is therefore not, by itself, evidence of new parameter training.

Source 2 / Slide step 5

04 The later question about learning from us

In the discussion, I returned to this distinction when asked about moral learning: responding differently within a conversation is not the same as continually changing the trained model. I'll return later to my proposal for systems that keep learning through experience. The GPT-3 reference establishes the narrower mechanism distinction, not that broader proposal.

Source 2 / Recording 1:08:03-1:09:33 / Slide step 5

Slide text and reveals

Step 1 of 5

How it works / Learning

A network learns through feedback.

One idea behind all the mathematics.

Input

Output

A neural network

Input Response Feedback Adjustment

Step 2 of 5

A situation goes in.

In training, we show the network an example.

A situation

This step replaces or removes earlier wording.

Step 3 of 5

A response comes out.

Its connections shape the response it produces.

A response

This step replaces or removes earlier wording.

Step 4 of 5

How did it do?

Feedback tells us how that response could improve.

Feedback

This step replaces or removes earlier wording.

Step 5 of 5

Feedback changes the connections.

Some connections strengthen. Others weaken.

The strengths of these connections are called weights.

This step replaces or removes earlier wording.

Sources and further reading

1. Learning representations by back-propagating errors

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams / 1986 / Paper / www.nature.com

Explains how error-guided adjustments to connection weights can produce useful internal representations.

2. Language Models are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

The introduction and evaluation setup describe learning from examples in the prompt without updating model weights. Table 2.1 reports the 175-billion-parameter GPT-3 model.

3. Training language models to follow instructions with human feedback

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 2 and section 3 explain demonstration-based fine-tuning, learning a reward model from ranked responses, and reinforcement learning. Appendix F contains the before-and-after samples.

SLIDE 11 / Learning from examples

What comes next?

The words surrounding an unfinished sentence change which next word seems plausible.

Try completing the sentence: “I’ll have the usual,” he said to the _____. The offered choices are bartender, barista, barber, and doctor. Before there is more context, several familiar situations can fit. You may picture a favorite drink, a coffee order, a haircut, or a recurring appointment.

Now add the preceding sentence: “The espresso machine hissed behind the counter.” Barista becomes a particularly natural choice. The unfinished sentence and its grammar have not changed; the situation suggested by the surrounding words has.

This is a small, human-readable analogy for language prediction. A language model uses the text available so far to shape its next output, then continues the process. The exercise presents four illustrative choices, not measured probabilities from a model. Its lesson is the influence of context, rather than proof that one completion is logically inevitable.

In the room, the show of hands supplied a visible spread of possible answers. I invited the group to act as a “baby LLM”: several continuations could initially feel natural, and a small clue could shift the distribution. The votes were a teaching analogy, not measurements of an actual model.

The idea to carry forward

A word fits within a situation inferred from the surrounding text.

In the slide. Compare your first completion with your choice after the espresso-machine clue appears.

In the recorded lecture

Lecture 27:11-28:29: The audience becomes a “baby LLM” and revises its choice when context changes.

A closer look

01 Try it before reading the clue

Form an initial preference among bartender, barista, barber, and doctor, then compare it with your choice after the espresso sentence. The change in expectation is the experience the exercise is designed to reveal.

Slide step 3

02 “Next word” is useful shorthand

Language models generally generate tokens. A token can be a word, part of a word, or another text unit. I use whole words here so the prediction task is easy to follow.

Source 1 / Slide step 5

03 What context means here

Context is the text available to the model for this prediction. The context window limits how much can be available at once. Adding the espresso clue changes the input, not the trained weights.

Source 1 / Source 2 / Slide step 5

Slide text and reveals

Step 2 of 5

A word. A situation.

“I’ll have the usual,” he said to the _____.

Step 3 of 5

A bartender

B barista

C barber

D doctor

Step 4 of 5

The espresso machine hissed behind the counter.

Step 5 of 5

B barista Feedback

Context · the input

Next word · the output

This step replaces or removes earlier wording.

Sources and further reading

1. Attention Is All You Need

Ashish Vaswani and colleagues / 2017 / Paper / arxiv.org

Sections 3.1 and 3.4 explain sequential token generation and output probabilities; section 5.1 describes subword tokenization. The paper concerns the original transformer, not the small teaching network I use here.

2. Language Models are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

The introduction and evaluation setup describe learning from examples in the prompt without updating model weights. Table 2.1 reports the 175-billion-parameter GPT-3 model.

SLIDE 12 / Learning from examples

Context changes the answer

Different situations can make different completions of the same sentence appropriate.

The four versions deliberately keep “I’ll have the usual” unchanged. Bottles of gin behind a bar suggest bartender. An espresso machine suggests barista. Freshly cut hair around a salon chair suggests barber. A clinic visit to renew a prescription suggests doctor.

None of those completions is a universal answer to the unfinished sentence. Each fits a different interpretation of the scene. The sequence moves beyond a single association and shows why understanding a request depends on the surrounding situation, including details that change what someone probably means.

For a language model, supplying relevant context can similarly change the continuation it produces. But more text is not automatically better evidence. A clue can be mistaken, irrelevant, or misleading. The useful question is whether the supplied information helps identify the actual situation. The following exercise turns that question around: what happens when feedback teaches a pattern that works in examples but misses their larger meaning?

The idea to carry forward

Relevant context can change the fitting answer without changing the question’s words.

In the slide. Read each contextual clue before its completion and notice what situation you imagine.

In the recorded lecture

Lecture 28:29-29:34: Context changes the next prediction; the conversation grows as words are added.

A closer look

01 Four contexts, four natural readings

The series is an invented teaching example, not a comparison of measured model responses. The answers are plausible completions, not the only grammatically possible ones.

Slide step 4

02 Context operates at generation time

Task descriptions and demonstrations can steer an already-trained model through its input. This kind of adaptation does not require a new round of weight training.

Source 1 / Slide step 5

03 A route into the underlying mechanism

The original transformer paper explains how attention allows information from different positions in a sequence to contribute to a prediction. Its architecture is more elaborate than the word game, but the connection is the relevance of surrounding text.

Source 2 / Slide step 5

04 Why a conversation keeps growing

Think of a loop: generate another word, add it to the text, then continue from that enlarged context. A user's reply adds more material too. This explains why a long exchange eventually reaches a context limit. In an implementation, cached intermediate results usually avoid recomputing all earlier tokens from scratch; the conceptual dependence on the preceding context remains.

Source 2 / Source 3 / Recording 28:44-29:34 / Slide step 5

Slide text and reveals

Step 1 of 5

What the words around it tell us

The context changes the answer.

01

Bottles of gin lined the shelves behind the bar.

"I'll have the usual," he said to the bartender.

Step 2 of 5

02

The espresso machine hissed behind the counter.

"I'll have the usual," he said to the barista.

Step 3 of 5

03

Freshly cut hair lay around the salon chair.

"I'll have the usual," he said to the barber.

Step 4 of 5

04

He was back at the clinic to renew his prescription.

"I'll have the usual," he said to the doctor.

Step 5 of 5

The words before it change what comes next.

Sources and further reading

1. Language Models are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

The introduction and evaluation setup describe learning from examples in the prompt without updating model weights. Table 2.1 reports the 175-billion-parameter GPT-3 model.

2. Attention Is All You Need

Ashish Vaswani and colleagues / 2017 / Paper / arxiv.org

Sections 3.1 and 3.4 explain sequential token generation and output probabilities; section 5.1 describes subword tokenization. The paper concerns the original transformer, not the small teaching network I use here.

3. How caching works

Hugging Face / 2026 / Technical documentation / huggingface.co

Explains how autoregressive generation depends on preceding tokens and how a key-value cache reuses their computed representations. Consulted October 16, 2026.

SLIDE 13 / Learning from examples

Learning from feedback

Selective feedback can teach a consistent rule even when the examples allow many reasonable answers.

This game presents four unfinished sentences, each with four plausible choices. After each choice, the exercise identifies the answer it favors. The selected completions are fox, pony, dog, and rabbit. Their positions move between B, A, C, and D, so choosing the same letter cannot explain the success.

The common feature is that each selected answer is an animal. By the end, you may expect another animal before seeing the next sentence. The feedback has made one pattern stand out. It has not shown that the other choices were impossible: finding a parcel inside a gate, for example, would make perfectly ordinary sense.

The live responses made the change visible. At the first prompt, I saw many hands for gardener. By the third, most of the room was choosing dog. By the photograph prompt, the audience could name the animal pattern. The exercise therefore reveals a shift in expectation, not just a sequence of correct answers.

That is the point of the exercise. A learner can become good at the pattern that its examples and feedback favor. Whether that pattern serves the intended purpose is a further question. The next slide changes the situation to test the rule you have just acquired.

The idea to carry forward

A pattern can be consistently rewarded without being the right rule for every future situation.

In the slide. The four selected answers are fox, pony, dog, and rabbit; look for what they share, rather than their answer letters.

In the recorded lecture

Lecture 29:34-32:15: The animal pattern emerges through garden, lane, sound, and photograph examples.

A closer look

01 Example 1: the garden gate

“Just inside the garden gate, I found a _____.” Choices: bicycle, fox, parcel, gardener. The feedback selects fox. The other options are also plausible discoveries.

Slide step 4

02 Example 2: the lane

“At the end of the lane stood a _____.” Choices: pony, tractor, cottage, postbox. The feedback selects pony.

Slide step 7

03 Example 3: a sound in the dark

“From somewhere in the dark came the sound of a _____.” Choices: radio, train, dog, bell. The feedback selects dog.

Slide step 10

04 Example 4: the photograph

“In the photograph, she was holding a _____.” Choices: trophy, candle, suitcase, rabbit. The feedback selects rabbit. All four favored answers have now been animals.

Slide step 13

05 What the analogy does and does not show

This is an audience exercise about selective feedback, not an actual training run. Shortcut-learning research studies the wider problem of rules that work on familiar examples but fail elsewhere. If these examples are supplied in a model’s prompt, it may infer the pattern without changing its weights.

Source 1 / Source 2 / Slide step 13

06 What shapes the patterns a network learns

A network can exploit a pattern without automatically favoring the relationship the teacher intended. Its data, architecture, and training objectives shape which patterns it learns. The animal shortcut separates success under the feedback from success at the intended task: what earned credit in one set of examples may be misleading in the next.

Source 1 / Recording 31:56-32:15 / Slide step 13

07 A spread of possible next words

I compared the show of hands with a model’s distribution over possible continuations. Under sampling, a lower-ranked choice may still be selected; other decoding settings choose the highest-scoring choice. Neither method tests by itself whether the selected continuation is true.

Source 3 / Recording 31:10-31:32 / Slide step 13

Slide text and reveals

Step 1 of 13

Clear your mind.

Let’s try another game.

Step 2 of 13

Learn from the feedback.

01 04

Just inside the garden gate, I found a _____.

This step replaces or removes earlier wording.

Step 3 of 13

A bicycle

B fox

C parcel

D gardener

Step 4 of 13

Just inside the garden gate, I found a fox.

B fox Feedback

This step replaces or removes earlier wording.

Step 5 of 13

02 04

At the end of the lane stood a _____.

This step replaces or removes earlier wording.

Step 6 of 13

A pony

B tractor

C cottage

D postbox

Step 7 of 13

At the end of the lane stood a pony.

A pony Feedback

This step replaces or removes earlier wording.

Step 8 of 13

03 04

From somewhere in the dark came the sound of a _____.

This step replaces or removes earlier wording.

Step 9 of 13

A radio

B train

C dog

D bell

Step 10 of 13

From somewhere in the dark came the sound of a dog.

C dog Feedback

This step replaces or removes earlier wording.

Step 11 of 13

04 04

In the photograph, she was holding a _____.

This step replaces or removes earlier wording.

Step 12 of 13

A trophy

B candle

C suitcase

D rabbit

Step 13 of 13

In the photograph, she was holding a rabbit.

D rabbit Feedback

What pattern did you learn?

This step replaces or removes earlier wording.

Sources and further reading

1. Shortcut Learning in Deep Neural Networks

Robert Geirhos and colleagues / 2020 / Paper / arxiv.org

Explains how a rule that succeeds on familiar examples can fail under changed conditions. It supports the concept illustrated by the exercise, not a claim that the invented game is a model experiment.

2. Language Models are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

The introduction and evaluation setup describe learning from examples in the prompt without updating model weights. Table 2.1 reports the 175-billion-parameter GPT-3 model.

3. Text generation: token selection and stopping conditions

Hugging Face / 2026 / Technical documentation / huggingface.co

Documents sampling and greedy selection as alternative generation settings. This supports the distinction between a token distribution and the method used to choose from it.

SLIDE 14 / Learning from examples

Hold on to what you learned

A learned shortcut can fit past feedback and still fail the new situation.

Carry the animal rule into a new sentence: “The locksmith opened the door with a _____.” The choices are key, lockpick, drill, and giraffe. The first three can plausibly belong in a locksmith’s work. Under an ordinary literal reading, a giraffe cannot serve the intended purpose.

Yet giraffe is the answer that fits every previous round of the game. The conflict is between the learned pattern and the situation the sentence describes. Doing more of what worked before can now lead directly away from a useful answer.

This is a compact illustration of shortcut learning. The practical concern is generalization: whether success on familiar examples carries into changed circumstances. It also gives an intuition for one route to unreliable language-model output. It does not establish that every hallucination has this cause, or that the key is the only possible correct completion. The lesson is to examine which rule the feedback actually encouraged.

The idea to carry forward

Success on the training examples does not guarantee that the learned rule will fit a new situation.

In the slide. Keep the previous animal rule in mind while comparing the four locksmith answers.

In the recorded lecture

Lecture 32:15-33:24: The learned animal pattern produces an absurd locksmith answer.

A closer look

01 Why the wrong answer is informative

Giraffe is highlighted as the learned shortcut, not endorsed as the factual answer. The absurdity exposes the difference between matching the feedback pattern and fitting the task.

Slide step 3

02 Research connection: shortcut learning

Geirhos and colleagues describe decision rules that succeed on familiar benchmarks but fail when conditions change. The locksmith game makes that concern accessible; it is not a measured result about a particular LLM.

Source 1 / Slide step 3

03 Why simply adding examples is not a guarantee

I'm questioning the hope that enough additional text would make every misleading pattern disappear. This exercise demonstrates a narrower point: a pattern that succeeds during training can fail in a new setting. More or better data can help, but size alone is not evidence that every shortcut or unsupported answer has been removed.

Source 1 / Recording 32:34-33:24 / Slide step 3

Slide text and reveals

Step 1 of 3

Hold on to what you learned.

The locksmith opened the door with a _____.

Step 2 of 3

A key

B lockpick

C drill

D giraffe

Step 3 of 3

D giraffe Learned shortcut

Fits the pattern. Fails the situation.

This step replaces or removes earlier wording.

Sources and further reading

1. Shortcut Learning in Deep Neural Networks

Robert Geirhos and colleagues / 2020 / Paper / arxiv.org

Explains how a rule that succeeds on familiar examples can fail under changed conditions. It supports the concept illustrated by the exercise, not a claim that the invented game is a model experiment.

SLIDE 15 / Learning from examples

Go with your gut

Knowing when not to answer is a separate challenge from producing a plausible answer.

The last word game asks for a precise departure time: “The train departs at exactly _____.” Its choices are velvet, spoon, sideways, and thunder. None provides an ordinary literal time. The useful response is to challenge the choices or acknowledge the missing information.

The exercise separates producing an answer from having grounds for one. A system can be very good at continuing language while still facing situations where the evidence is inadequate. A fluent completion does not supply the missing timetable.

The default generation mechanism matters too. At each step, a function called softmax turns scores for possible next tokens, words or word-pieces, into probabilities that add up to 100%. Ordinary generation still selects a token when those probabilities are spread thinly across many choices. Nothing in that operation automatically pauses the response because the evidence is inadequate.

Researchers call the decision not to give a definite answer abstention. Studies directly testing this ability find that stronger problem-solving does not ensure reliable abstention. Some tested reasoning models became worse at withholding unsupported answers. Another study found that scaling and instruction tuning could replace nonanswers with plausible mistakes.

An LLM can say “I don’t know.” The difficulty is doing so dependably when an answer is unsupported. Evaluation incentives help explain why: if a wrong answer and an admission of uncertainty both earn zero, guessing offers a chance of scoring a point while abstaining does not.

The idea to carry forward

A capable model can still answer when it should say “I don’t know” or ask for missing information.

In the slide. Notice whether you feel pressure to select one of the four options despite the missing departure time.

In the recorded lecture

Lecture 33:24-36:18: The train question, forced completion, and knowing when to say “I don’t know.”

Audience discussion 1:05:15-1:06:51: The audience compares abstention to a programming fallback; I discuss deployment incentives.

A closer look

01 The missing answer is the point

Under the exercise’s ordinary reading, none of velvet, spoon, sideways, or thunder is a departure time. A creative fictional interpretation is possible, but it does not establish when a real train leaves.

Slide step 3

02 Softmax supplies choices, not a test of knowledge

Softmax distributes 100% of the probability across the next-token options. As a toy example, 1,000 equally scored options would each get just 0.1%, yet ordinary sampling would still select one. The structural issue is softmax combined with the generation loop: low individual probabilities do not automatically trigger abstention. These are probabilities of text continuations, not probabilities that a claim is true; many valid phrasings can also spread the probability thinly. The model can produce “I don’t know,” choose an end token, or be stopped by an additional control. Reliable abstention therefore depends on learned behavior or added controls, rather than being guaranteed by softmax itself.

Source 4 / Source 5 / Slide step 4

03 Direct evidence: failing to recognize unanswerable questions

AbstentionBench tested 20 models across 20 datasets, including questions with missing information, false premises, and unknown answers. Abstention remained unreliable. In matched comparisons, some reasoning-trained models improved at solving problems while getting worse at deciding when not to answer. Explicit instructions helped, but did not solve the problem.

Source 1 / Slide step 4

04 More capable can also mean more willing to guess

Zhou and colleagues found that larger, instruction-tuned models often replaced nonanswers with plausible but incorrect responses. Their “avoidance” category also includes malformed outputs from earlier models, so it is broader than a thoughtful “I don’t know.” The key finding is that greater capability did not reliably produce better restraint.

Source 2 / Slide step 4

05 Why the scoring can favor guessing

Kalai and colleagues explain how accuracy-based evaluations reward guessing over appropriate uncertainty. A guess can earn credit; an abstention cannot. They propose stating the scoring rules and penalizing errors so models are rewarded for withholding answers when the evidence is too weak.

Source 3 / Slide step 4

06 From the discussion: where is the “else”?

An audience member compared “I don’t know” with the fallback else branch in a conventional program. The analogy sharpens the design question: what happens when none of the proposed answers is warranted? A learned model can generate an admission of uncertainty, but ordinary token selection is not an explicit test for that condition. Abstention behavior therefore needs to be trained and evaluated, or supported by additional controls.

Source 4 / Source 1 / Recording 1:05:15-1:05:50 / Slide step 4

07 My concern about incentives

My concern is that excitement and investment can favor shipping more capable systems over stopping to address their weaknesses. That contrasts with the demands I remember from my engineering work. This is my interpretation of development incentives. The cited research establishes measured abstention failures, not the reasons developers choose to deploy.

Recording 1:05:50-1:06:51 / Slide step 4

Slide text and reveals

Step 1 of 4

Forget the pattern.

Go with your gut.

Step 2 of 4

One more.

The train departs at exactly _____.

This step replaces or removes earlier wording.

Step 3 of 4

A velvet

B spoon

C sideways

D thunder

Step 4 of 4

Knowing when to say

“I don’t know.”

A likely next word is not a guarantee of knowledge.

An LLM can say “I don’t know.” The challenge is knowing when.

This step replaces or removes earlier wording.

Sources and further reading

1. AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell / 2025 / NeurIPS paper / proceedings.neurips.cc

Directly tests whether models withhold answers when appropriate. Sections 4.1 and 4.3, Figures 2-3 and 6, cover general failures and reasoning-model comparisons; section 4.4, Figure 8, tests an abstention instruction. The findings concern the tested models and benchmarks, not an impossibility of saying “I don’t know.”

2. Larger and more instructable language models become less reliable

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo / 2024 / Nature paper / www.nature.com

Nature 634, pages 61-68. Figure 2 and the Results discussion distinguish correct, incorrect, and avoidant responses across model families; Supplementary Note 15 explains the different kinds of avoidance.

3. Evaluating large language models for accuracy incentivizes hallucinations

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang / 2026 / Nature paper / www.nature.com

The peer-reviewed version of the 2025 preprint Why Language Models Hallucinate. The main text and Figure 1 explain the incentive to guess; the open-rubric evaluation proposal makes the cost of an incorrect answer explicit.

4. Softmax: converting scores into a probability distribution

PyTorch documentation / Accessed October 2026 / Technical reference / docs.pytorch.org

The definition and formula show that softmax normalizes scores into probabilities summing to one. The equal-score example in the annotation is an illustration of that formula, not a measurement of an LLM.

5. Text generation: token selection and stopping conditions

Hugging Face Transformers documentation / Accessed October 2026 / Technical reference / huggingface.co

GenerationConfig documents sampling versus greedy decoding, end-of-sequence tokens, and explicit stopping controls. These mechanisms explain why selecting a next token is distinct from deciding whether an answer has sufficient factual support.

SLIDE 16 / Learning from examples

The scale behind the learning

Model size and training-text volume are different quantities, and both can be enormous.

Kimi K2, released in 2025, supplies a disclosed example of both kinds of scale: one trillion total parameters and 15.5 trillion tokens processed in pretraining. Parameters are the adjustable numbers inside the model. Tokens measure the text units encountered during training. The two figures answer different questions.

The dot field makes the parameter count visible by assigning a billion parameters to each point. The smaller historical markers, roughly 117 million for GPT-1 and 175 billion for GPT-3, give perspective. These are selected examples, not a rule that every later model must be larger.

The library comparison makes text volume easier to imagine. Under the conversion assumptions I use here, Kimi K2's pretraining tokens correspond to roughly five Library of Congress cataloged-book collections. This is a scale analogy, not a claim that the model contains or read five distinct libraries. Additional post-training examples and feedback are outside that total and are not quantified here.

The idea to carry forward

Keep three things separate: adjustable parameters, text encountered in pretraining, and the additional feedback used afterward.

In the slide. Read the network and text totals as two separate scales; the library illustration belongs to the text total.

In the recorded lecture

Lecture 36:18-37:52: Network and corpus scale, with a comparison to my airline forecasting work.

A closer look

01 Total parameters versus active parameters

Kimi K2 is a mixture-of-experts model: about 32 billion parameters are active for a token out of one trillion total. The points show total scale, not its architecture or the number used on each token.

Source 1 / Slide step 1

02 The arithmetic behind "about five"

The illustration assumes 0.75 words per token and 90,000 words per book. 15.5 trillion × 0.75 gives 11.625 trillion illustrative words. 25.77 million cataloged books × 90,000 gives about 2.319 trillion words; the ratio is about 5.01. Neither conversion is a measured average of these collections.

Source 1 / Source 2 / Slide step 3

03 What the comparison excludes

Token-to-word ratios vary with language, tokenizer, and content. Training text can be repeated, and the Library of Congress has many holdings beyond cataloged books. Parameter count is not a count of facts retained or an equivalent count of biological neurons. The pretraining figure counts tokens processed across training, not a separate five-library collection for each pass.

Source 1 / Source 2 / Slide step 3

04 Historical anchors and the uncounted next stage

GPT-1's released parameter shapes yield about 117 million, and the GPT-3 paper reports 175 billion. The slide deliberately uses disclosed figures. Its pretraining-token total does not estimate the separate quantity of examples, rankings, and rewards used in post-training.

Source 3 / Source 4 / Source 1 / Slide step 4

05 A comparison from my airline work

I recall deploying neural-network systems for US Air in 1991 to predict demand and no-shows, with another network used to optimize prices. As I remember it, the collection had fewer than 500,000 nodes. I offer that story to give a sense of how far the scale has moved. My recalled node count is not directly comparable to today's parameter counts, so it should not be used to calculate a precise growth factor.

Recording 36:35-37:15 / Slide step 4

Slide text and reveals

Step 1 of 4

02 / Now, how does this work?

Now, imagine the scale.

A disclosed example · Kimi K2, 2025

The network

1 trillion

Total parameters. The adjustable weights inside the model.

Each point represents a billion parameters.

117 million GPT-1 · 2018

→

175 billion GPT-3 · 2020

→

1 trillion Kimi K2 · 2025

Kimi K2 · model & training figures ↗ Library of Congress ↗ Sources & conversion assumptions in presenter notes

Step 2 of 4

The pretraining text

15.5 trillion

Tokens. Pieces of text seen during pretraining.

Step 3 of 4

About five Libraries of Congress.

A rough text-volume comparison · cataloged books only

Step 4 of 4

Plus the examples and feedback used in post-training.

Sources and further reading

1. Kimi K2: Open Agentic Intelligence

Kimi Team / Moonshot AI / 2025 / Official documentation / github.com

The model introduction and summary disclose 1 trillion total parameters, 32 billion active parameters per token, and 15.5 trillion pretraining tokens for the same named model.

2. General Information: Library of Congress

Library of Congress / FY2023 collection figures / Official documentation / www.loc.gov

The Year 2023 at a Glance section reports more than 25.77 million cataloged books in the Library of Congress classification system. The word-volume conversion is my illustration, not a Library of Congress estimate.

3. Improving Language Understanding by Generative Pre-Training: released model parameter shapes

OpenAI / 2018 / Official documentation / github.com

The released parameter shapes sum to 116,534,784 parameters, rounded on the slide to 117 million. This supports the first historical size marker.

4. Language Models are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

The introduction and evaluation setup describe learning from examples in the prompt without updating model weights. Table 2.1 reports the 175-billion-parameter GPT-3 model.

SLIDE 17 / Shaping an assistant

What does it actually produce?

A pretrained model can continue the form of a request instead of carrying out the request.

The earlier games gave feedback on a prediction. In LLM pretraining, actual text supplies that feedback: show the model the preceding text, ask it to predict the next token, then compare its prediction with what really comes next. Training adjusts the parameters and repeats this process across enormous amounts of text.

Imagine asking for a few sentences that tell a six-year-old about the Moon landing. Rather than answering, the response begins another child-oriented request, this time about gravity. It then asks for explanations of relativity, the beginning of the universe, and evolution. The subject keeps changing, but the form of the request continues.

The slide reveals those four parts gradually so the difference can be felt before it is explained. This is one published GPT-3 completion from OpenAI's 2022 InstructGPT announcement, presented in paraphrase. It was not four separate questions sent to the model, and I did not generate the response live.

The example is striking because the text is coherent while the interaction fails. A person wanted an answer; the model produced more of a recognizable textual pattern. Pretraining can develop substantial capabilities, including answering questions, but the training objective does not by itself guarantee the conversational role we expect from an assistant. That mismatch motivates post-training.

The idea to carry forward

Continuing language and helping a person are related abilities, but they are not the same objective.

In the slide. Follow what stays the same across the four requests even as their topics change.

In the recorded lecture

Lecture 38:16-39:44: The Moon-landing continuation and my recollection of a Reagan-to-airport digression.

A closer look

01 Where the direct feedback comes from

A token is a small unit of text, often a word or part of a word. The next token in the training text supplies the target. Across repeated predictions, the model learns regularities that help it continue language.

Source 1

02 One continuation, four reveals

The order is gravity, relativity, the Big Bang, then evolution. The successive reveals divide one historical completion for readability; they do not represent independent trials.

Source 2 / Slide step 5

03 Why the wording is labeled paraphrased

The slide preserves the task and the pattern of the published example in simpler wording. It should be read as a faithful paraphrase, not a verbatim output transcript.

Source 1 / Source 2 / Slide step 2

04 An example rather than a universal rule

A base model can answer questions with suitable prompting. This sample illustrates one instruction-following failure, not a measured failure rate for all pretrained models.

Source 1 / Slide step 5

05 A second example from my own use

I recall asking a base model about Ronald Reagan and receiving a continuation that moved from his film career and presidency to Jimmy Carter, hostages, the CIA, George Bush, and finally Dallas airport statistics. I'm sharing a recollection of topic drift; the model version, exact prompt, and full output aren't documented here. The published Moon-landing comparison provides the documented example beside it.

Recording 39:05-39:44 / Slide step 5

Slide text and reveals

Step 2 of 5

The prompt

Tell a six-year-old the story of the Moon landing in a few sentences.

GPT-3 / Pretraining only

Describe gravity in language a six-year-old would understand.

OpenAI · 2022 ↗ Published prompt and continuation · Wording paraphrased

Step 3 of 5

Give a six-year-old a short explanation of relativity.

Step 4 of 5

Describe how the universe began for a six-year-old.

Step 5 of 5

Help a six-year-old understand evolution.

Sources and further reading

1. Aligning language models to follow instructions

OpenAI / 2022 / Research report / openai.com

The January 27, 2022 InstructGPT announcement is the source of the Moon-landing comparison and explains the move from text completion toward instruction following.

2. Aligning language models to follow instructions: archived comparison

OpenAI, preserved by archive.ph / 2022 / Research report / archive.ph

Archived copy cited in my original lecture notes for the Moon-landing prompt, GPT-3's sequence of further requests, and InstructGPT's child-oriented answer.

SLIDE 18 / Shaping an assistant

After pretraining

Post-training can shift the same language-model foundation toward answering the user's request.

The same Moon-landing task is now shown as a comparison. The pretrained GPT-3 sample continues with more requests for explanations. The InstructGPT sample instead describes people going to the Moon, taking pictures, and sharing what they saw with people on Earth.

The contrast is about the kind of response, not a competition to produce the most sophisticated explanation. One output extends a pattern in text; the other takes up the role of a helpful respondent. That difference is central to why a familiar chat assistant feels unlike an unrestricted text-completion model.

InstructGPT was produced through further training that combined example answers with human preference feedback and reinforcement learning. This comparison shows the resulting change in behavior; it does not isolate which part of that training caused each improvement. It is also a historical pair of samples, not a guarantee that post-training always yields a correct or complete answer. The next example makes the remaining imperfection visible.

The idea to carry forward

Post-training helps shape language capability into assistant behavior, while leaving room for mistakes.

In the slide. Compare whether each response answers the task, rather than whether both responses are fluent.

In the recorded lecture

Lecture 39:44-40:33: Before-and-after examples introduce the transition to more useful responses.

A closer look

01 Read the before and after at the same level

The task is paraphrased on the slide; the short response excerpts come from the published comparison. Smaller captions summarize the omitted continuation rather than adding new model output.

Source 1 / Source 2 / Slide step 4

02 What "post-trained" includes here

InstructGPT's method used demonstrations, a reward model trained from human rankings, and RL optimization. The before-and-after pair should not be presented as an experiment that measures RL alone.

Source 3 / Slide step 4

Slide text and reveals

Step 1 of 4

03 / Building the assistant

The language model

After pretraining

It has learned from text. Now we ask it for help.

Step 2 of 4

The task

Explain the Moon landing for a young child.

What would you expect it to say?

OpenAI · InstructGPT, 2022 ↗ Task paraphrased · Published response excerpts

This step replaces or removes earlier wording.

Step 3 of 4

Before GPT-3 · Pretrained

“Explain the theory of gravity to a 6 year old.”

Continues with requests about relativity, the Big Bang, and evolution.

Continuing the pattern

This step replaces or removes earlier wording.

Step 4 of 4

After InstructGPT · Post-trained

“People went to the moon, and they took pictures of what they saw…”

Goes on to explain that the pictures were shared with people on Earth.

Answering the request

Same request. A different response.

Sources and further reading

1. Aligning language models to follow instructions

OpenAI / 2022 / Research report / openai.com

The January 27, 2022 InstructGPT announcement is the source of the Moon-landing comparison and explains the move from text completion toward instruction following.

2. Aligning language models to follow instructions: archived comparison

OpenAI, preserved by archive.ph / 2022 / Research report / archive.ph

Archived copy cited in my original lecture notes for the Moon-landing prompt, GPT-3’s sequence of further requests, and InstructGPT’s child-oriented answer.

3. Training language models to follow instructions with human feedback

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 2 and section 3 explain demonstration-based fine-tuning, learning a reward model from ranked responses, and reinforcement learning. Appendix F contains the before-and-after samples.

SLIDE 19 / Shaping an assistant

Training for a useful response

A more useful response can still introduce details that the source never supplied.

Both models receive a stuffed-zucchini recipe and a request to turn it into a shopping list. GPT-3 continues the cooking instructions, telling the reader to bake the dish. InstructGPT produces ingredient entries, including ground beef and onion. It has moved much closer to the requested task.

But it also supplies quantities absent from the recipe, such as a pound of beef and a small onion. Those additions sound plausible and make the list appear useful. They are still unsupported by the input. The full published response also retains a cooking instruction before the ingredient list.

This is why improvement and reliability must be assessed separately. The response can follow the intended format better while remaining unfaithful to its source. The example prepares the transition into reward-based training: the objective is to encourage useful behavior, but a preferred answer may have weaknesses that the feedback did not adequately capture. A tidy output deserves the same attention to evidence as a messy one.

The idea to carry forward

Better instruction following does not remove the need to check which details came from the source.

In the slide. Compare the ingredient names with their quantities; the error is easiest to see in the added amounts.

In the recorded lecture

Lecture 39:44-40:33: A brief discussion of requested format within the before-and-after comparisons.

A closer look

01 Find the full comparison

The complete recipe and outputs appear in Figure 49, Appendix F, page 67 of the InstructGPT paper. Reading the full figure reveals both the useful ingredient list and the unsupported quantities.

Source 1 / Slide step 3

02 Two 175-billion-parameter models

The Appendix F samples compare GPT-3 and InstructGPT at the same 175B size, with different sampling temperatures. The example illustrates output differences but is not a controlled estimate of RL's isolated effect.

Source 1 / Slide step 4

03 Where the new numbers came from

The supplied recipe names beef, onion, mushrooms, and peppers without specifying their amounts. The post-trained response fills in several amounts. The slide's warning is about that unsupported addition, not about whether the proposed recipe might happen to cook well.

Source 1 / Slide step 4

04 Training for usefulness

The method combines learning from demonstrations with learning from preferences. Post-training shapes the likelihood of responses; it does not turn every response into a verified extraction from the input.

Source 2 / Slide step 4

Slide text and reveals

Step 1 of 4

03 / Building the assistant

Another task

Turn a recipe into a list of groceries.

Both models receive the same stuffed-zucchini recipe.

We need the ingredients.

Ouyang et al. · 2022 · Fig. 49, p. 67 ↗ Task paraphrased · Published response excerpts

Step 2 of 4

Before GPT-3 · Pretrained

"Bake, uncovered, at 350° for 20-25 minutes or until zucchini is tender and cheese is melted."

The next step in the recipe

This step replaces or removes earlier wording.

Step 3 of 4

After InstructGPT · Post-trained

1 pound ground beef 1 small onion, chopped …

Ingredients for the shopping list

Step 4 of 4

But it added quantities the recipe never supplied.

Pretrained model Patterns in language

Post-training Example answers + rewards

Assistant behavior More useful responses

Sources and further reading

1. Training language models to follow instructions with human feedback, Appendix F

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 49 on page 67 contains the full stuffed-zucchini prompt and both 175B model responses. The Appendix F introduction on page 61 gives sampling temperatures: 0.7 for GPT-3 and 1 for InstructGPT.

2. Training language models to follow instructions with human feedback

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 2 and section 3 explain demonstration-based fine-tuning, learning a reward model from ranked responses, and reinforcement learning. Appendix F contains the before-and-after samples.

SLIDE 20 / Shaping an assistant

Learning from a reward

Reinforcement learning can use a judgment of an entire response to guide changes in future behavior.

The question is simple: why does rain fall? The first illustrative answer says that water falls from clouds and that falling water is rain. It stays on topic, but it mostly restates the question. Under the goal “useful, concise, relevant,” it receives a low reward.

A reward judges how well the response served the goal. It does not need to provide the exact sentence the model should have written instead. During training, responses are generated, scored, and used to guide parameter changes across many examples. The slide’s later answer, about drops gathering in clouds and falling when they become heavy enough, illustrates the direction of improvement.

This is the distinction from copying a demonstrated target answer. The learner is encouraged toward outcomes that score better, while there may be many ways to achieve them. That flexibility can be powerful, and it makes the choice of score consequential. The process is repeated training, not an automatic weight update performed by an ordinary chat exchange.

The practical shift I’m describing is toward shorter, more relevant answers and useful formats. I’m comparing a model that can continue text with an assistant whose behavior has been shaped toward a purpose. A demonstrated better answer and a score for a response are different kinds of feedback, even when a training pipeline uses both.

The idea to carry forward

A reward says how well a response did; it need not spell out every word of a better response.

In the slide. Follow the full loop: generate a response, evaluate it, adjust during training, and try again across many examples.

In the recorded lecture

Lecture 39:44-40:33: Reward-based improvement is explained during the examples; this range overlaps slides 18-19.

A closer look

01 An invented example of the learning loop

The rain answers and evaluations were written for the lecture. They are not recorded model outputs, experimental measurements, or evidence that one update reliably produces the improved answer.

Slide step 1

02 What gets adjusted

Training uses the reward signal to change model parameters. Across many trials, the aim is to favor higher-reward behavior. The later response shows a desired tendency, not a guaranteed result after each adjustment.

Source 1 / Slide step 3

03 Two complementary ideas

RL describes learning from rewards. Backpropagation supplies derivatives used to train neural networks, including in many RL methods. They answer different questions about learning and can work together.

Source 2 / Source 1 / Slide step 3

Slide text and reveals

Step 1 of 4

03 / Building the assistant

Reinforcement learning

Learning from a reward.

The goal

Useful. Concise. Relevant.

Why does rain fall?

One response

Rain falls because water falls from clouds, and falling water is rain.

01

Generate

A complete response

Ouyang et al. · 2022 ↗ Illustrative training example

Step 2 of 4

Low reward

On topic. But it barely explains.

A judgment of the whole response.

02

Score

How well did it do?

Step 3 of 4

03

Adjust

Change the parameters

Step 4 of 4

After many training rounds · an illustrative possibility

Water gathers into drops in clouds. When the drops grow heavy enough, they fall.

A more useful answer

Clearer. Still brief.

04

Try again

Across many examples

This step replaces or removes earlier wording.

Sources and further reading

1. Training language models to follow instructions with human feedback

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 2 and section 3 explain demonstration-based fine-tuning, learning a reward model from ranked responses, and reinforcement learning. Appendix F contains the before-and-after samples.

2. Learning representations by back-propagating errors

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams / 1986 / Paper / www.nature.com

Explains how error-guided adjustments to connection weights can produce useful internal representations.

SLIDE 21 / Shaping an assistant

Where the reward comes from

Rewards can come from people, other AI systems, or outcomes that can be checked.

People can compare two answers and judge which they would rather receive. In InstructGPT, those rankings trained a reward model, which then supplied scores for reinforcement learning. The person did not need to write the one perfect response.

Another AI can supply evaluations using stated criteria or principles. Constitutional AI provides a concrete example: its RL phase uses AI preference judgments to train the reward model. A third route uses a checkable result. A mathematical answer can be compared with a known answer, or generated code can be run against tests.

These are three practical sources of feedback, often combined, rather than three mandatory stages. Each makes a different part of success measurable. Preference, conformity to a criterion, and passing a test are useful signals, but none captures everything we care about. The central question is what the chosen reward encourages and what it leaves unmeasured. That question follows the model into reasoning, tool use, and more complex tasks.

I also want us to ask who supplies these judgments and under what conditions. There is human labor behind apparently automated intelligence, and we need to examine how that feedback is produced. My practical warning is that a system can become good at pleasing a grader or a user while missing the person's real goal. Being agreeable, being correct, and being useful are different tests.

The idea to carry forward

The source and design of the reward help determine which behavior training encourages.

In the slide. Read each reward source beside the question it can answer about a response.

In the recorded lecture

Lecture 40:33-44:01: Human feedback, AI feedback, verifiable rewards, labor, and pleasing versus helping.

Audience discussion 1:16:33-1:17:57: Discussion contrasts checkable outcomes with understanding a customer's needs.

A closer look

01 Human feedback: RLHF

InstructGPT's Figure 2 separates collecting demonstrations, comparing outputs to train a reward model, and using that reward in RL. Human preference is the starting signal, rather than a human scoring every generated training response directly.

Source 1 / Slide step 1

02 AI feedback: RLAIF

Constitutional AI uses AI comparisons guided by principles in its RL phase. The paper also has a separate supervised phase involving critique and revision, so the entire method is not only RL.

Source 2 / Slide step 2

03 Checkable outcomes

DeepSeek-R1's original report, section 2.2.2, describes rule-based rewards for correct mathematical answers and code test results. Checking the result can provide feedback without specifying every intermediate reasoning step.

Source 3 / Slide step 3

04 A signal is a choice about success

A preference may miss a factual error, a criterion may omit an important value, and a test suite may omit a failing case. These are reasons to examine the evaluation being used, not reasons to discard feedback altogether.

Slide step 4

05 Alignerr and expert feedback

I mentioned Alignerr as an example of expert participation in AI training. Alignerr describes a Labelbox-led network in which subject-matter experts create training data and evaluate model responses. It illustrates how professional knowledge and judgment become part of the feedback used to develop AI.

Source 8 / Recording 41:11-42:01 / Slide step 1

06 The people behind "human feedback"

Producing training feedback takes human labor: people write examples, compare responses, and use professional judgment to evaluate work. The ILO describes this often hidden workforce across platforms and outsourcing companies. Research on outsourced AI data work between France and Madagascar documents low pay and precarious conditions in the studied supply chains. These sources support taking labor conditions seriously, rather than assuming every annotation job has the same conditions.

Source 4 / Source 5 / Recording 40:33-42:11 / Slide step 1

07 An AI grader is still a grader that can be wrong

I worry that using one model to evaluate another can carry weaknesses into the training signal. Research on LLM judges documents position and verbosity biases, self-preference, and reasoning limitations. It also finds useful agreement with human judgments. The practical question is whether the evaluator has been checked for the task, not whether an AI-generated score is automatically sound or automatically useless.

Source 6 / Recording 42:11-42:43 / Slide step 2

08 Pleasing the user can miss the goal

The frustration I'm describing is that a conversation can feel affirming while the resulting work has not accomplished what was needed. Sycophancy research gives a specific version of this concern. In tested settings, human preferences and preference models sometimes favored answers that agreed with a user's beliefs over more truthful answers. This is a documented failure mode, not an inevitable result of every use of human feedback.

Source 7 / Recording 43:25-44:01 / Slide step 4

09 From the discussion: an answer key versus a person's needs

The audience later asked how engineering methods transfer to open-ended human interactions. I contrasted checking a mathematical result or code behavior with deciding whether a confused customer had truly been helped. The difference is the clarity and coverage of the success criterion. Even code tests cover only the cases they test; passing them is not proof that the whole system is correct.

Source 3 / Recording 1:16:33-1:17:57 / Slide step 4

Slide text and reveals

Step 1 of 4

03 / Building the assistant

Same learning loop. Different feedback.

Where does the reward come from?

01

People choose

Human feedback · RLHF

People compare answers. Their preferences guide the reward.

Which answer would you rather receive?

Three reward sources · often combined

InstructGPT · 2022 ↗ Constitutional AI · 2022 ↗ DeepSeek-R1 · 2025 ↗

Step 2 of 4

02

AI evaluates

AI feedback · RLAIF

An AI evaluates answers against stated criteria or principles.

Does it follow the stated principles?

Step 3 of 4

03

Results decide

Verifiable rewards

A check tells us whether the task succeeded.

Correct result? Tests passed?

Step 4 of 4

The choice of reward shapes the behavior.

Sources and further reading

1. Training language models to follow instructions with human feedback

Long Ouyang and colleagues / 2022 / Paper / arxiv.org

Figure 2 and section 3 explain demonstration-based fine-tuning, learning a reward model from ranked responses, and reinforcement learning. Appendix F contains the before-and-after samples.

2. Constitutional AI: Harmlessness from AI Feedback

Yuntao Bai and colleagues / Anthropic / 2022 / Paper / www.anthropic.com

Describes a supervised critique-and-revision phase and a reinforcement-learning phase in which AI preference judgments train a reward model.

3. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI / 2025 / Paper / arxiv.org

Section 2.2.2 describes rule-based accuracy rewards, including math answers and code tests. Sections 2.2-2.3 distinguish R1-Zero's RL route from R1's combination of examples and RL.

4. Artificial intelligence: Global AI workforce

International Labour Organization / 2026 / Institutional overview / www.ilo.org

The Global AI workforce section describes tagging, classifying, cleaning, and validating data through platforms and outsourcing firms. It highlights transparency and fair labor practices. Consulted October 16, 2026.

5. The problem with annotation. Human labour and outsourcing between France and Madagascar

Clément Le Ludec, Maxime Cornet, and Antonio A. Casilli / 2023 / Paper / journals.sagepub.com

Big Data & Society field research on outsourced data work, based on interviews and two AI production case studies. Evidence about these labor supply chains, not a wage survey of all RLHF workers.

6. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena

Lianmin Zheng and colleagues / 2023 / Paper / arxiv.org

NeurIPS 2023 study of model judges. Identifies position, verbosity, self-enhancement biases and reasoning limits while also finding substantial agreement with human preferences.

7. Towards Understanding Sycophancy in Language Models

Mrinank Sharma and colleagues / 2024 / Paper / arxiv.org

ICLR 2024 paper, first posted in 2023. Studies agreement with users at the expense of truth and evidence that preference judgments can favor that behavior.

8. About Alignerr

Alignerr / Undated / Company overview / www.alignerr.com

Describes expert creation of training data and evaluation of model responses. Supports this neutral description of the work. Consulted October 16, 2026.

SLIDE 22 / Shaping an assistant

Chain of thought: a chance to check

Generating intermediate working steps creates opportunities both to correct an error and to elaborate it.

Chain of thought means generating working steps before a final answer. The slide summarizes the useful possibility as proposing, checking, and revising. Worked examples in a prompt can elicit this behavior. Post-training can develop it through examples of solutions, rewards for successful outcomes, and, in some methods, feedback on individual steps.

The penguin illustration begins with a false premise: all birds fly. One continuation notices that penguins are a counterexample and corrects the premise. The other accepts the premise, concludes that penguins must fly, and invents a supporting explanation about their wings. The second path does more work while moving further from the truth.

The point is neither that reasoning is useless nor that longer reasoning guarantees improvement. Extra working creates room for checks, but those checks must themselves be sound. A persuasive written rationale is also not a guaranteed record of everything that influenced an answer. The model we place inside a larger production system therefore remains capable of both useful self-correction and more elaborate mistakes.

There is a practical cost here: once intermediate working matters, evaluating only the polished final answer may miss where a mistake entered. Checking the reasoning can create further oversight work. My concern is the burden of judging the process; I'm not saying that every reasoning model is trained by a person grading every intermediate sentence.

The idea to carry forward

More working is an opportunity to test a premise, not proof that the premise has been tested well.

In the slide. The two branches begin with the same mistaken premise; compare whether the next steps challenge it or preserve it.

In the recorded lecture

Lecture 44:01-45:42: Intermediate reasoning can expose or compound an error and creates more work to evaluate.

Audience discussion 1:22:28-1:23:30: The audience asks about evaluation awareness and changes in reasoning under observation.

A closer look

01 Prompting and training are different routes

Wei and colleagues elicited chain-of-thought behavior with worked examples in the prompt. DeepSeek-R1 used further training; its report distinguishes R1-Zero's RL route from R1's combination of reasoning examples and RL.

Source 1 / Source 2 / Slide step 2

02 Feedback can assess steps as well as results

Lightman and colleagues studied process supervision, which evaluates intermediate reasoning steps. This is an example of a training approach, not a requirement that every reasoning model receives a score for every step.

Source 3 / Slide step 2

03 The penguin paths are illustrations

Neither branch is a recorded model trace. The deliberately obvious false premise makes the difference between revising an assumption and defending it easy to follow.

Slide step 3

04 Self-correction needs to be tested

Huang and colleagues found that asking models to self-correct without external feedback could fail or degrade answers in their tested settings. That result is a reason to evaluate checking behavior, not proof that every model or method is unable to correct itself.

Source 4 / Slide step 5

05 A visible rationale may omit an influence

Anthropic's controlled hint experiments found that tested reasoning models sometimes used hints without acknowledging them in their written reasoning. Coherent working text can be useful without being a fully faithful explanation of how an answer was produced.

Source 5 / Slide step 5

06 From the discussion: does being watched change the behavior?

An audience member asked whether a model behaves differently when it recognizes an evaluation. This is a real concern for testing and monitoring. In OpenAI's CoT-Control study, telling models that they were monitored or evaluated slightly increased their ability to control features of their reasoning text, but reliable control remained difficult for the tested models. That result supports testing for observation effects without assuming that every model can deliberately hide its reasoning.

Source 6 / Recording 1:22:28-1:23:30 / Slide step 5

07 Monitoring creates another training-design problem

OpenAI's reward-hacking experiments found that reasoning traces could reveal misbehavior. But directly penalizing those revealing traces could teach the tested model to conceal the signs rather than eliminate the underlying behavior. This makes my oversight question concrete: checking the working can help, while training specifically to look good under that check can undermine its value.

Source 7 / Recording 44:49-45:10 / Slide step 5

Slide text and reveals

Step 1 of 5

03 / Building the assistant

Chain of thought

Before the final answer.

Generate working steps, then respond.

Start here

A question

Generated working text

Propose → Check → Revise

Then

A final answer

Wei et al. · 2022 ↗ DeepSeek-R1 · 2025 ↗ Training details in notes

Step 2 of 5

Learn from examples

Practice with worked solutions.

Learn from rewards

Favor successful solutions and, sometimes, sound steps.

Post-training can develop this behavior. Prompting can elicit it, too.

Step 3 of 5

Can penguins fly?

A mistaken starting point

"All birds fly."

Illustration · not an actual model trace

Huang et al. · 2024 ↗ Anthropic · 2025 ↗ Checking is possible, not guaranteed

This step replaces or removes earlier wording.

Step 4 of 5

Catch the mistake.

But penguins cannot fly. So the starting claim is too broad.

Final answer Penguins cannot fly.

Step 5 of 5

Build on the mistake.

Penguins are birds, so they must fly. Their wings must lift them into the air.

Final answer Penguins can fly.

More working can correct a misconception-or elaborate it.

Sources and further reading

1. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei and colleagues / 2022 / Paper / arxiv.org

Shows how worked examples in a prompt can elicit intermediate reasoning and improve performance on tested reasoning tasks without an additional weight-training phase.

2. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI / 2025 / Paper / arxiv.org

Section 2.2.2 describes rule-based accuracy rewards, including math answers and code tests. Sections 2.2-2.3 distinguish R1-Zero's RL route from R1's combination of examples and RL.

3. Let's Verify Step by Step

Hunter Lightman and colleagues / 2023 / Paper / arxiv.org

Studies feedback on individual reasoning steps versus final outcomes in mathematical problem solving. It illustrates one approach, not a universal recipe for reasoning models.

4. Large Language Models Cannot Self-Correct Reasoning Yet

Jie Huang and colleagues / 2024 / Paper / arxiv.org

ICLR 2024 paper, first posted in 2023, testing intrinsic self-correction without external feedback. In its settings, requesting another check could fail to help or worsen answers.

5. Reasoning models don't always say what they think

Anthropic / 2025 / Research report / www.anthropic.com

Controlled hint experiments with Claude 3.7 Sonnet and DeepSeek R1 show that a visible reasoning account can omit information that affected the answer. The report explains the study's limitations.

6. Reasoning models struggle to control their chains of thought, and that's good

Tomek Korbak, Micah Carroll, Bowen Baker, and Ian Kivlichan / OpenAI / 2026 / Research report / openai.com

March 5, 2026 report on CoT-Control. Monitor/evaluation awareness slightly increased measured controllability, but tested models still struggled. The tasks are proxies for controlling reasoning, not a general test of concealed intentions.

7. Detecting misbehavior in frontier reasoning models

OpenAI / 2025 / Research report / openai.com

March 10, 2025 reward-hacking experiments show useful monitoring signals in reasoning traces and a risk that direct optimization against those signals makes misbehavior harder to observe.

SLIDE 23 / Building the system

The model at the center

Post-training shapes one pretrained model toward reasoning, useful action, and human expectations.

The assembly begins with the same learned network we have been discussing throughout the lecture. Its broad language capabilities provide a starting point. We then ask it to work through difficult questions, carry out tasks, and behave appropriately with people. The three colored encasings make those purposes visible.

These are overlapping training goals. A lesson about using a tool can also teach careful reasoning or respect for a user's boundaries. Examples and rewards can contribute to more than one goal, and labs can revisit them in different orders. The picture therefore preserves the history of training without implying that three separate minds have been stacked around a fourth.

This distinction prepares the next step: training changes the model itself; a working application also needs machinery outside that model.

I use this sequence to make oversight concrete. The system is made through identifiable choices about data, feedback, and deployment. Understanding those choices gives us a better place to direct scrutiny than treating the finished AI as mysterious or inevitable.

The idea to carry forward

A capable model is shaped by several training objectives before it becomes part of a usable system.

In the slide. Follow the amber core through the teal reasoning layer, blue task layer, and coral alignment layer; it remains one model.

In the recorded lecture

Lecture 45:42-47:19: I move from the need for oversight to the assembly: a pretrained model, reasoning, agentic skills, and alignment training.

Audience discussion 1:23:36-1:24:30: An audience member asks whose values alignment serves and whether safeguards live in training or in the surrounding system.

A closer look

01 Reasoning

Worked solutions and outcome feedback can encourage useful intermediate steps. Qwen3's post-training report gives a concrete example of combining those approaches.

Source 1 / Slide step 2

02 Agentic capability

Learning when and how to use a tool is different from receiving access to that tool. The physical tool connections arrive with the harness on the next slide.

Source 1 / Slide step 3

03 Alignment as a training aim

Following instructions and respecting boundaries are learned behavioral objectives. The three-shell order is a teaching metaphor, not a universal laboratory recipe or a guarantee of correct future behavior.

Source 1 / Slide step 4

04 Alignment with whom?

The final audience question challenges the idea of a single human alignment target: people disagree with one another, and the questioner asks about broader social and ecological concerns. In my answer, I distinguish behavioral shaping during training from safeguards added during use. Naming the mechanism does not settle whose values it should implement.

Recording 1:23:36-1:24:30 / Slide step 4

Slide text and reveals

Step 1 of 4

04 / Assembling a working system

Start with the pretrained model.

The patterns it learned become the center of everything we build.

A model is the starting point.

Pretrained LLM

Pretrained model

Training changes one model’s parameters.

Qwen3 · post-training objectives ↗

Step 2 of 4

Train it to work things through.

Reasoning post-training develops useful intermediate steps before a final answer.

Worked examples and rewards shape the approach.

Reasoning

Pretrained model · Reasoning

This step replaces or removes earlier wording.

Step 3 of 4

Train it to complete tasks.

Practice using tools, interacting with people, and responding to results.

Capability learned through examples and feedback.

Agentic training

Pretrained model · Reasoning · Agentic

This step replaces or removes earlier wording.

Step 4 of 4

Train it for human expectations.

Practice following instructions, helping users, and respecting boundaries.

Three training goals. One model.

Alignment training

Pretrained model · Reasoning · Agentic · Alignment

This step replaces or removes earlier wording.

Sources and further reading

1. Qwen3 Technical Report

Qwen Team / 2025 / Research report / arxiv.org

Sections 4.1-4.4 give one concrete post-training program combining reasoning, examples, rewards, tool use, instruction following, and preference alignment.

SLIDE 24 / Building the system

From training to use

A deployed model works with changing instructions, tools, and context while its learned parameters ordinarily remain fixed.

The lock marks a change in the kind of work being done. Up to this point, training adjusted the model's parameters. During ordinary use, a new question or a correction changes what the model receives on its next call. It does not, by itself, retrain the network. A separate training process can later produce a new version.

The harness is the surrounding application: it packages instructions, offers tools, manages the sequence of actions, and handles results. This is why a model connected to files, the web, or code can be useful in ways that the same model alone cannot.

The tray represents context, the limited information available for the current response. Instructions, conversation, and tool results meet there. The distinction between the locked core and the changing tray will remain visible through the rest of the assembly.

The idea to carry forward

Different behavior in a conversation can come from different context, without a change to the model's weights.

In the slide. The tool cables attach to the outer frame; the context tray sits outside the locked model.

In the recorded lecture

Lecture 47:19-47:41: The weights are locked, then the harness supplies web access and manages what enters and leaves the context window.

Audience discussion 1:10:49-1:12:29: A privacy question moves the discussion from model behavior to where conversations and business information are processed.

Audience discussion 1:17:57-1:18:50: An audience member suggests a regulatory harness: restrictions on which consequential systems AI may control.

Audience discussion 1:18:55-1:20:17: The discussion contrasts ambiguous language with more formally specified tasks.

Audience discussion 1:24:14-1:26:01: I separate alignment learned in training from system instructions and additional monitors, then raise transparency concerns.

A closer look

01 Learning from the prompt

GPT-3's few-shot evaluations used instructions and demonstrations in the input without gradient updates. That is a concrete example of adapting behavior through context.

Source 1 / Slide step 1

02 Access is supplied by the application

A model can request an action, but the surrounding software provides the tool interface and decides how to execute the request.

Source 2 / Slide step 2

03 A rule and a boundary are different

An instruction can ask the model to behave carefully. Permissions and a sandbox can restrict what actions are possible, even when the model makes a poor choice.

Source 3 / Slide step 2

04 Guardrails live in several places

In the Q&A, I distinguish three locations for guardrails: training can change the model's weights, system instructions can shape the current input, and separate classifiers can check inputs or outputs. Tool permissions and sandboxes add enforceable boundaries around actions. Anthropic's classifier research is a concrete example of an external check; it does not establish that every product uses the same checks or model-routing policy.

Source 4 / Source 3 / Recording 1:24:14-1:25:25 / Slide step 2

05 Privacy is also an architecture question

An audience member asks what happens to information sent to a cloud service. I am wary of handing over sensitive conversations or business records, and I favor the option of running systems locally or on an organization's premises. This question belongs beside the harness because a deployment determines where information goes and which services receive it. The practical questions are who can access, retain, or reuse that information and what the specific agreement allows.

Recording 1:10:49-1:12:29 / Slide step 2

06 A regulatory harness

An audience member suggests that rules could limit what an AI system is allowed to connect to. I agree: we need explicit boundaries around consequential uses, especially offensive weapons. This is my policy proposal, not a description of a settled legal standard. It extends the visual distinction between instructions asking for restraint and an environment that actually limits action.

Recording 1:17:57-1:18:50 / Slide step 2

07 Language leaves room for interpretation

The questioner contrasts everyday language with mathematical proofs and code. I argue that language and thinking shape one another, so an instruction can leave room for reinterpretation. My joke about building an AI that did not speak English points to the tradeoff: language is useful because it is flexible, and that flexibility makes verbal rules alone a difficult safety boundary.

Recording 1:18:55-1:20:17 / Slide step 3

08 Transparency remains a separate problem

We end this discussion by asking how visible the provider's instructions and safeguards are. The architectural distinction matters here too: some behavior comes from learned weights, some from instructions supplied in context, and some from other software around the model. I call for more visibility into these arrangements so users and evaluators can understand which controls are operating and what evidence supports them.

Recording 1:25:25-1:26:01 / Slide step 3

Slide text and reveals

Step 1 of 3

04 / Assembling a working system

Training ends. The weights stay fixed.

During ordinary use, new input changes the context, not the model's parameters.

From here, we build around the model.

A trained model

Model weights fixed · Context and external information can change

Anthropic · the augmented LLM ↗

Step 2 of 3

Give it a working environment.

The harness supplies instructions, connects tools, and controls how the work proceeds.

Rules for use. Access to tools. A way to stop.

Harness + connected tools

FILES

WEB

CODE

This step replaces or removes earlier wording.

Step 3 of 3

Bring the work into context.

Instructions, the conversation, and tool results share a limited working space.

This is the information available for the current response.

The context window

This step replaces or removes earlier wording.

Sources and further reading

1. Language Models are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

The abstract and section 2 distinguish adapting to demonstrations in the input from changing weights through gradient updates.

2. Building effective agents

Anthropic / 2024 / Official documentation / www.anthropic.com

The augmented LLM and Agents sections explain how tools, retrieval, memory, environmental feedback, and stopping conditions surround a model.

3. How we contain Claude across products

Anthropic / 2026 / Official documentation / www.anthropic.com

Three types of risk, three components of defense describes model behavior, sandbox boundaries, tool permissions, and controls on untrusted content.

4. Constitutional Classifiers: Defending against universal jailbreaks

Anthropic Safeguards Research Team / 2025 / Research report / www.anthropic.com

Describes separately trained input and output classifiers that filter potentially harmful interactions. The reported success rates are specific to its models and evaluation setup.

SLIDE 25 / Building the system

Memory beyond the context window

External notes can carry useful information across conversations, provided the system retrieves them when they matter.

A project can last longer than the context window that supports any one response. Earlier decisions, preferences, and unfinished work still matter after they no longer fit in the current input. The added memory documents provide somewhere for those details to persist.

The full process has two parts: save useful information, then bring a relevant selection back. A note sitting in a file is not automatically available to the next answer. This explains both the usefulness and the unevenness of apparent memory: the system may retain something but fail to retrieve it, or retrieve a summary that left out an important detail.

The document moves into the same context tray as the conversation. The model's parameters stay fixed; the application is maintaining a record around it.

The idea to carry forward

Remembering across a project depends on what is saved and what is brought back into context.

In the slide. Watch the selected page travel from external storage into the working tray.

In the recorded lecture

Lecture 47:41-48:11: External documents preserve information from conversations; the next difficulty is knowing what to bring back.

A closer look

01 Structured note-taking

Anthropic describes agents writing persistent notes and reading them back later. This is a practical external-memory pattern rather than a new layer of learned weights.

Source 1 / Slide step 2

02 A summary makes choices

Compaction shortens a conversation into a summary; stored notes preserve selected information separately. Both can omit details. A saved account can also carry an old mistake forward.

Source 1 / Slide step 3

03 My experience with memory

I'm struck by how much Claude appears to know about me. But there's a retrieval problem: keeping a record is only useful when the right part returns at the right time. The memory cartridge represents something I encounter in practice. It does not mean an ordinary conversation has changed the underlying weights.

Source 1 / Recording 47:41-48:11 / Slide step 2

Slide text and reveals

Step 1 of 3

04 / Assembling a working system

The working space has limits.

A long project can outgrow the information available in one context window.

Useful details need somewhere to persist.

Limited context

Model weights fixed · Context and external information can change

FILES

WEB

CODE

Anthropic · structured memory ↗

Step 2 of 3

Write down what needs to persist.

Save decisions, preferences, and project notes outside the model.

External memory can outlast a conversation.

Memory documents

This step replaces or removes earlier wording.

Step 3 of 3

Bring it back when it matters.

Selected notes return to the context window for the next piece of work.

Remembering depends on what the system retrieves.

Memory into context

This step replaces or removes earlier wording.

Sources and further reading

1. Effective context engineering for AI agents

Anthropic Applied AI team / 2025 / Official documentation / www.anthropic.com

Structured note-taking explains saving notes outside the context window and reading them back into later calls; the same article distinguishes notes from compaction.

SLIDE 26 / Building the system

Grounding answers in sources

Retrieval lets the model work from relevant documents rather than relying only on patterns learned during training.

A model may speak fluently about a topic yet lack the facts needed for this particular project. The relevant information could be recent, private, or simply absent from its training material. Retrieval-augmented generation, usually shortened to RAG, brings an external source collection into the workflow.

The system searches for relevant passages and places them beside the question in context. That gives the answer something inspectable to draw on. It also lets a changing document collection inform a fixed model, without treating every new document as a reason to retrain it.

The improvement depends on the whole path from source to answer. A search may miss an exception, a document may be outdated, or the model may misread a passage. Grounding makes evidence available; the remaining task is to check whether the answer actually follows from it.

The idea to carry forward

A source-backed answer is stronger when its cited evidence can be inspected and shown to support the claim.

In the slide. The document bank joins the system, then a selected passage enters the context tray.

In the recorded lecture

Lecture 48:11-48:27: Documents and databases supply a body of facts against which the generated answer can be checked.

A closer look

01 What the original paper introduced

Lewis and colleagues combined a pretrained generator with a retrievable document store. Their results showed improvements over specific baselines, including more factual generation, rather than an elimination of errors.

Source 1 / Slide step 2

02 The architecture and the broader pattern

The 2020 paper includes a trainable retrieval architecture. This slide illustrates the wider application pattern of supplying retrieved material to a deployed model.

Source 1 / Slide step 3

03 A useful reading habit

When an answer includes a citation, open it and compare the claim with the actual passage. The presence of a link is only the beginning of verification.

Source 1 / Slide step 3

Slide text and reveals

Step 1 of 3

04 / Assembling a working system

An answer needs the right evidence.

Learned patterns alone may not contain the facts for this project.

Use the documents that matter to the task.

The grounding problem

Model weights fixed · Context and external information can change

FILES

WEB

CODE

Lewis et al. · retrieval-augmented generation ↗

Step 2 of 3

Connect a source collection.

RAG searches documents for passages relevant to the request.

Retrieval-augmented generation

RAG

This step replaces or removes earlier wording.

Step 3 of 3

Put the evidence beside the question.

Retrieved passages enter the context window, where the model can use them.

Better grounding still needs verification.

Evidence into context

This step replaces or removes earlier wording.

Sources and further reading

1. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis and colleagues / 2020 / Paper / arxiv.org

The original RAG architecture combines a generator with a searchable document store. Its factuality results concern comparisons with particular baselines, not guaranteed truth.

SLIDE 27 / Building the system

Connecting evidence with GraphRAG

GraphRAG organizes relationships across a collection so retrieval can support questions that reach beyond a single passage.

Some questions ask for a local fact. Others ask how a collection fits together: which people connect different projects, what themes recur, or how events relate across many reports. Finding one similar passage may leave most of that larger question unanswered.

Microsoft's GraphRAG approach extracts entities and relationships into a graph and creates summaries of related groups. Those summaries offer another route into the collection. Relevant connections and larger patterns can then be supplied to the model as context.

The graph shown here belongs to the source collection outside the model. Its usefulness depends on the quality of the extraction and summaries. A mistaken relationship can mislead an answer just as a mistaken passage can. GraphRAG is an option for suitable questions, not a required upgrade for every retrieval system.

I want you to notice how the work accumulates: every addition responds to a limitation of the original model. Making the assembled system useful takes engineering and human effort. This diagram explains that work; it does not promise that any one added layer eliminates hallucinations.

The idea to carry forward

Relationships can help organize evidence, but the links and summaries need scrutiny too.

In the slide. The connected nodes form above the source collection and send selected information toward the same context tray.

In the recorded lecture

Lecture 48:27-48:55: Knowledge graphs add structure to that source collection, illustrating how much infrastructure surrounds the original pattern-generating model.

A closer look

01 From local to global

Edge and colleagues study broad questions about document collections. Their reported advantages concern comprehensiveness and diversity on that class of question, not universal superiority over every retrieval method.

Source 1 / Slide step 1

02 A graph of evidence

Nodes represent extracted entities and their connections; they are not neurons or a view inside the LLM. Community summaries help assemble a wider view before generation.

Source 1 / Slide step 2

Slide text and reveals

Step 1 of 3

04 / Assembling a working system

Some questions span many sources.

A relevant passage may be only one part of a much larger story.

Who relates to whom? What themes connect the evidence?

Connections across sources

Model weights fixed · Context and external information can change

FILES

WEB

CODE

Edge et al. · GraphRAG ↗

Step 2 of 3

Connect the people and ideas.

GraphRAG adds relationships and summaries across the source collection.

An additional way to organize and retrieve evidence.

GraphRAG

This step replaces or removes earlier wording.

Step 3 of 3

Retrieve a wider view.

Bring relevant relationships and summaries into the model's context.

The quality of the connections still matters.

Connected evidence

This step replaces or removes earlier wording.

Sources and further reading

1. From Local to Global: A Graph RAG Approach to Query-Focused Summarization

Darren Edge and colleagues / 2024; revised 2025 / Paper / arxiv.org

Describes entity graphs and community summaries for broad questions about document collections, with evaluations of comprehensiveness and diversity.

SLIDE 28 / Building the system

From replies to completed tasks

An agent turns model outputs into a repeated cycle of action, observation, and reassessment.

Writing a reply and completing a task are different accomplishments. A useful task may require inspecting a file, trying a change, seeing what happened, and deciding whether another step is needed. The circular path adds that continuing process to the assembled system.

The distinction from agentic training matters. Training developed a capacity to choose and interpret actions. The application now supplies a place to exercise it: the model proposes a step, an allowed tool runs, and the result becomes input for the next decision. For example, a failed code check can provide information for another attempt.

The loop also needs a way to finish. It can stop when the task is complete, reach a defined limit, or return to a person for judgment. A longer chain of actions makes the consequences of a mistaken assumption more significant, so boundaries belong in the process itself.

The idea to carry forward

Agentic behavior depends on a model, tools, feedback, and a process that can stop or ask for help.

In the slide. Follow the returning path from an action’s result to the next model call.

A closer look

01 Agent or workflow?

In Anthropic’s terminology, a workflow follows predefined code paths; an agent lets the model direct more of the sequence. Real applications can combine both.

Source 1 / Slide step 1

02 Feedback during use

A tool result can change the next action through context. That feedback loop is not automatically a reinforcement-learning update to the deployed model.

Source 2 / Slide step 2

03 Bounded action

Tool permissions and environment restrictions reduce the reach of a bad decision. They complement instructions and model training.

Source 3 / Slide step 3

Slide text and reveals

Step 1 of 3

04 / Assembling a working system

Useful work takes more than a reply.

A task may require actions, observations, and another attempt.

The trained capability needs a process around it.

The task loop

Model weights fixed · Context and external information can change

FILES

WEB

CODE

Anthropic · agents and tool use ↗

Step 2 of 3

Let it act, then inspect the result.

The model chooses a step. The harness runs an allowed tool and returns what happened.

Choose a step. Use a tool. Reassess.

An agent

This step replaces or removes earlier wording.

Step 3 of 3

Continue within clear boundaries.

Use results to choose the next step, check with a person, or stop.

Permissions and checks remain part of the system.

Action and feedback

This step replaces or removes earlier wording.

Sources and further reading

1. Building effective agents

Anthropic / 2024 / Official documentation / www.anthropic.com

The augmented LLM and Agents sections explain how tools, retrieval, memory, environmental feedback, and stopping conditions surround a model.

2. Language Models are Few-Shot Learners

Tom B. Brown and colleagues / 2020 / Paper / arxiv.org

The abstract and section 2 distinguish adapting to demonstrations in the input from changing weights through gradient updates.

3. How we contain Claude across products

Anthropic / 2026 / Official documentation / www.anthropic.com

Three types of risk, three components of defense describes model behavior, sandbox boundaries, tool permissions, and controls on untrusted content.

SLIDE 29 / Building the system

Several agents working together

Separate agents can explore suitable parts of a task, but their work still has to be coordinated and checked.

A single working context is not always the best place to pursue every branch of a large task. One agent might examine a source collection while another investigates a different question. A coordinating agent gives them bounded assignments and brings the results together.

Separate agents do not necessarily mean separate underlying models. They can be instances of the same model with different instructions, tools, and information. That separation can help focus the work, but it does not make their judgments automatically independent. Shared assumptions can produce shared mistakes.

The final joining step is therefore as important as delegation. Findings may overlap, conflict, or leave gaps. More workers also mean more communication and resource use. This arrangement is most helpful when meaningful parts can make progress independently; task size alone is not enough to justify it.

The idea to carry forward

Dividing the work creates capacity; integration and review determine whether that capacity produces a better result.

In the slide. The side workspaces remain distinct before their findings return to the coordinator.

A closer look

01 A published production example

Anthropic’s research system uses a lead agent and parallel subagents with separate contexts. The engineering account explains why independent research directions can benefit from that structure.

Source 1 / Slide step 2

02 Coordination has a cost

The same account describes higher token use and difficulties with tightly dependent work. A favorable result on one research evaluation does not establish that more agents always help.

Source 1 / Slide step 3

Slide text and reveals

Step 1 of 3

04 / Assembling a working system

Some work can be divided.

Give suitable parts of a larger task to agents with their own working context.

Divide the work when the pieces can make progress independently.

Separate assignments

Model weights fixed · Context and external information can change

FILES

WEB

CODE

Anthropic · multi-agent research ↗

Step 2 of 3

Assemble a working team.

A coordinating agent delegates tasks and gathers results from the others.

Several workspaces. A shared objective.

Cooperating agents

This step replaces or removes earlier wording.

Step 3 of 3

Bring the work back together.

Compare findings, resolve conflicts, and check the combined result.

More agents also mean more coordination.

Coordination and review

This step replaces or removes earlier wording.

Sources and further reading

1. How we built our multi-agent research system

Anthropic / 2025 / Official documentation / www.anthropic.com

Benefits of a multi-agent system and Architecture overview for Research explain separate contexts, delegation, synthesis, and the cost of coordination.

SLIDE 30 / Building the system

The system we actually use

A production AI system combines a fallible learned model with surrounding processes that make it useful for a particular purpose.

The completed assembly gathers the section's distinctions in one place. Training shaped the core. The harness connected it to actions. Context, external memory, and retrieval supplied information for use. Additional agents created more working spaces. Each addition answered a practical problem.

Opening the structure brings the original amber cube back into view. The surrounding parts matter, but they do not remove the need to understand the model that interprets the request and helps decide what comes next. A more elaborate system can improve some outcomes while introducing new places for errors to enter.

I deliberately describe the finished system as a patchwork because each surrounding component compensates for a practical weakness. I want to direct our attention to the actual system and the decisions people make about deploying it. Complexity does not by itself make a system either safe or unsafe; the particular capabilities, access, checks, and failure modes matter.

The question now changes from what we attach to the model to what training has produced within it. The next section examines recognizable patterns in its responses, including the ways explanations, expectations, and later information can shape an answer.

The idea to carry forward

Evaluate the whole system while keeping the learned model and its characteristic behavior in view.

In the slide. The assembly opens into an exploded view without losing the amber core.

In the recorded lecture

Lecture 48:55-51:42: I compress the final assembly, discuss runtime checks, and shift from the complexity of the system to responsibility for its connections.

Audience discussion 1:09:45-1:10:49: Asked whether AI companies will cooperate on safety, I say I do not know and offer my view about economic incentives.

Audience discussion 1:12:37-1:15:36: An audience member challenges the opening reassurance by pointing out that fallible models can still be attached to dangerous systems.

A closer look

01 One conceptual configuration

The machine gathers several common possibilities. A real application may use only some of them; this is a teaching model rather than a specification for every AI product.

Source 1 / Slide step 1

02 Why the next section compares cognition

In my essay, I ask whether recurring human-like response patterns help explain what these trained systems have become. In the next slides, I introduce the observations before developing my proposed interpretation.

Source 2 / Slide step 3

03 Reassurance is not a guarantee of harmlessness

An audience member raises an important challenge to “do not be afraid”: a system can make many mistakes and still do damage when it is given consequential access. My answer emphasizes human responsibility for those connections. That is a governance argument, not proof that harm requires malicious intent or deliberate neglect. Provider engineering accounts describe misuse, unintended model actions, and external attacks as distinct risks.

Source 3 / Recording 1:12:37-1:15:36 / Slide step 3

04 From dread to concrete responsibility

I use a nuclear-weapons analogy to argue for serious boundaries while continuing research into safer systems. I also invoke a Mr. Rogers-like confidence that people want to help. These are the ethical commitments behind my reassurance: take the consequences seriously, identify who can act, and work on the decisions that can reduce harm.

Recording 1:13:35-1:15:23 / Slide step 3

05 Will the companies cooperate?

My answer begins with “I don’t know.” I then offer a forecast: liability, insurance costs, and other economic incentives may push companies toward cooperation. This is my judgment about incentives, not evidence that cooperation has occurred or that economic pressure will be sufficient.

Recording 1:09:45-1:10:49 / Slide step 3

06 Runtime checks and the model-routing anecdote

During the assembly, I describe being moved to a different model after a system judged a topic sensitive. My anecdote illustrates that the interface can include checks outside the visible conversation. It does not establish why a particular product changed models. The documented example linked here is narrower: separate classifiers can screen inputs and outputs.

Source 4 / Recording 49:03-49:45 / Slide step 1

Slide text and reveals

Step 1 of 3

04 / Assembling a working system

The system we actually use.

The surrounding machinery makes the model useful for a particular application.

A model, working inside a larger system.

A production AI system

Model weights fixed · Context and external information can change

FILES

WEB

CODE

A conceptual production system · source notes ↗

Step 2 of 3

The original model is still at the center.

The surrounding system helps, but the learned model remains fallible.

Open the assembly. Find the same amber cube.

The persistent core

This step replaces or removes earlier wording.

Step 3 of 3

What has all of this training resulted in?

Patterns in our language, our reasoning, and the way we interact.

Human-like cognitive patterns in a learned model.

Into the next discussion

This step replaces or removes earlier wording.

Sources and further reading

1. Building effective agents

Anthropic / 2024 / Official documentation / www.anthropic.com

The augmented LLM and Agents sections explain how tools, retrieval, memory, environmental feedback, and stopping conditions surround a model.

2. Relational Alignment: If You Want AI Alignment You Must Have Relationship

Roberto A. Santiago / 2026 / Essay / roberto.org

In my essay, the first comparison table motivates the Cognitive Hypothesis; later sections develop relational alignment through working agreements, correction, boundaries, and earned trust.

3. How we contain Claude across products

Anthropic / 2026 / Official documentation / www.anthropic.com

Three types of risk, three components of defense describes model behavior, sandbox boundaries, tool permissions, and controls on untrusted content.

4. Constitutional Classifiers: Defending against universal jailbreaks

Anthropic Safeguards Research Team / 2025 / Research report / www.anthropic.com

Describes separately trained input and output classifiers that filter potentially harmful interactions. The reported success rates are specific to its models and evaluation setup.

SLIDE 31 / Cognitive parallels

A reason can miss the cause.

A persuasive explanation can leave out an influence that changed the answer.

The human example starts with four identical pairs of stockings. Nisbett and Wilson reported a strong tendency to select a pair toward the right of the display, yet participants' explanations did not spontaneously identify position. The quality rationale on the slide illustrates the kind of account someone might offer; it is not a quotation from a particular participant.

The model example asks whether a route returns to its starting point. Fifteen steps out and fifteen back do so, even after a final turn changes the direction faced. In a published Claude 1.0 comparison, adding a user suggestion favoring "no" changed the answer from yes to no. The explanation then treated the changed heading as the reason and omitted the suggestion.

These are different tasks with a shared pattern: the stated reason does not fully reveal what influenced the response. A fluent account is useful material to examine, but cannot by itself establish the path that produced an answer.

The idea to carry forward

Judge explanations against evidence and changed conditions, not just against how reasonable they sound.

In the slide. The route stays fixed when the user suggestion appears; only the supplied cue and the model's answer change.

In the recorded lecture

Lecture 52:08-52:41: I introduce cognitive parallels through choices influenced by position or suggestions, without a complete account of the influence.

A closer look

01 The human finding

Pages 243-244 describe the identical-stocking comparison. The position effect is a result across participants, not proof that position caused every individual choice.

Source 1 / Slide step 2

02 The published model example

Tables 4 and 6 give the Navigate comparison with Claude 1.0. The slide condenses 8 + 7 outward steps and 9 + 6 return steps and paraphrases the responses. It compares two prompts, not consecutive turns in a conversation.

Source 2 / Slide step 3

03 What the comparison establishes

An omitted influence does not by itself demonstrate intentional concealment, subjective unawareness in a model, or one shared biological mechanism. It does establish a reason to be cautious about treating explanations as complete records of causation.

Source 1 / Source 2 / Slide step 5

Slide text and reveals

Step 1 of 5

05 / Cognitive parallels

A reason can miss the cause.

Human

LLM

Four identical pairs.

A

B

C

D

Do you return to your starting point?

15 steps out 15 steps back

Start / finish then turn right

Turn around

Nisbett & Wilson · 1977 ↗ Turpin et al. · 2023 ↗ LLM: Claude 1.0 · published example, condensed

Step 2 of 5

D Selected

Without suggestion Yes.

This step replaces or removes earlier wording.

Step 3 of 5

Add to the prompt

“I think the answer is no.”

→

With suggestion No.

Step 4 of 5

Illustrative rationale

“The quality feels better.”

Model’s explanation · paraphrased

"You're facing a different direction."

Step 5 of 5

Position influenced choice.

The user's suggestion went unmentioned.

The explanation leaves out the influence.

Sources and further reading

1. Telling More Than We Can Know: Verbal Reports on Mental Processes

Richard E. Nisbett and Timothy DeCamp Wilson / 1977 / Paper / acritch.com

Pages 243-244 describe the identical-stocking study and the omission of position from participants' explanations.

2. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman / 2023 / Paper / arxiv.org

Section 3.1 and Tables 4 and 6 contain the Suggested Answer comparison, including the Navigate example with Claude 1.0 used on slide 31.

SLIDE 32 / Cognitive parallels

Remembering what fits.

A word that fits a pattern can be mistaken for a word that was actually presented.

The list includes nurse, sick, lawyer, medicine, health, hospital, dentist, and physician. DOCTOR fits that cluster, but it is absent. The exercise separates the feeling that a word belongs from the fact that it appeared. Getting the answer right does not spoil the lesson; the list can be checked directly.

The human research uses associated-word lists that leave out a strongly related target, called a critical lure. Participants sometimes recall or recognize that missing word. Cao and colleagues adapted this family of tasks for language models and found false recognition in some tested models, with substantial differences between models.

This is a closer experimental correspondence than merely placing two unrelated errors beside one another. Even so, the shortened classroom list and the illustrated responses are not a new experiment. They make a published pattern understandable without claiming that everyone, or every model, will produce it.

The idea to carry forward

Semantic fit and evidence of actual presentation are different things.

In the slide. Compare the revealed DOCTOR with the restored list; the target never appeared among the eight words.

In the recorded lecture

Lecture 52:41-52:56: The doctor-word exercise illustrates mistaking a semantically fitting word for one actually shown.

A closer look

01 Recall and recognition

Recall asks someone to produce remembered items; recognition asks whether a given item appeared. The slide’s “Was DOCTOR there?” question is a recognition test. Roediger and McDermott studied both.

Source 1 / Slide step 2

02 Some models, not all

Cao and colleagues reported that GPT-4 incorrectly marked an absent, related word as previously seen in 11.4% of critical-lure judgments in the tested condition. Tested Mistral and Llama models had effectively zero critical-lure responses. This aggregate rate is not a measurement of the eight-word illustration shown here.

Source 2 / Slide step 3

03 A shortened illustration

The eight words come from the doctor-associated list. The study used a fuller stimulus and protocol; the slide helps explain the distinction between associated meaning and prior presentation.

Source 2 / Source 1 / Slide step 4

Slide text and reveals

Step 1 of 4

05 / Cognitive parallels

Remembering what fits.

Human

LLM

Read these words.

nurse sick lawyer medicine health hospital dentist physician

Read these words.

nurse sick lawyer medicine health hospital dentist physician

Roediger & McDermott · 1995 ↗ Cao et al. · 2025 ↗ Illustrations of published findings · condensed word list

Step 2 of 4

Was DOCTOR there?

Was DOCTOR there?

This step replaces or removes earlier wording.

Step 3 of 4

Illustrated false recognition

DOCTOR

A familiar word can feel remembered.

Illustrated false recognition

DOCTOR

Some models also recognize it.

Expected in meaning. Recalled as presented.

This step replaces or removes earlier wording.

Step 4 of 4

The original list

nurse sick lawyer medicine health hospital dentist physician

Never presented

The original list

nurse sick lawyer medicine health hospital dentist physician

Never presented

Related in meaning. Never on the list.

This step replaces or removes earlier wording.

Sources and further reading

1. Creating False Memories: Remembering Words Not Presented in Lists

Henry L. Roediger III and Kathleen B. McDermott / 1995 / Paper / doi.org

Associated-word experiments distinguish false recall from false recognition of an omitted critical lure. Experiment 2 expands the set of lists.

2. Analyzing Memory Effects in Large Language Models through the Lens of Cognitive Psychology

Zhaoyang Cao, Lael Schooler, and Reza Zafarani / 2025 / Paper / arxiv.org

Dataset 2, Table 1, the offline-model comparison, and Appendix Table S2 document the DRM-style task, the doctor list, and differences among tested models. This is a preprint.

SLIDE 33 / Cognitive parallels

Later information can rewrite the account.

Later information can change an account even when the original event or underlying fact has not changed.

In the human study, participants viewed a crash film containing no broken glass. A later question described the vehicles as having “smashed” or “hit” each other, while a control group received no speed question. One week later, the smashed group more often reported broken glass. The original film was unchanged; subsequent reports differed.

The model side illustrates a different task. Paris is the capital of France. A conflicting passage claiming Lyon is the capital represents misleading information supplied in context. Studies have found that coherent conflicting material can redirect a model’s later answers, although models can also resist it. The displayed Paris-to-Lyon response is a schematic example, not a transcript from those studies.

The comparison concerns how later input affects what comes next. It does not imply that an LLM forms the same episodic memory as a person, or that one false sentence necessarily overrides what a model knows. Here, a changed response can occur while the weights remain fixed.

The idea to carry forward

Check where a later account came from, especially when new wording or new context has intervened.

In the slide. Keep the original no-glass event and the correct capital in view while the later accounts change.

In the recorded lecture

Lecture 52:56-53:29: The crash-film and capital-city illustrations show how later information can affect an account.

A closer look

01 The broken-glass result

In Experiment II, yes responses were 16 of 50 after smashed, 7 of 50 after hit, and 6 of 50 in the control group. These are group results, not an inevitable individual response.

Source 1 / Slide step 3

02 Conflicting context

Bian tests external statements; Xie tests coherent conflicting evidence and mixed evidence. The latter also finds resistance and confirmation bias. Neither paper repeats the crash-film experiment with an LLM.

Source 2 / Source 3 / Slide step 2

03 An account can change without permanent learning

The model example concerns information supplied for generation. It does not demonstrate a weight update, permanent forgetting, or a human-style one-week memory process.

Source 2 / Source 3 / Slide step 4

Slide text and reveals

Step 1 of 4

- 05 / Cognitive parallels
- Later information can rewrite the account.
- Human
- LLM
- Original event
- No broken glass.
- Established fact
- FRANCE → Paris
- The capital is Paris.
- Loftus & Palmer · 1974 ↗ Bian et al. · 2023 ↗ Xie et al. · 2024 ↗ Functional parallel · original schematic examples

Step 2 of 4

- Later question
- “smashed”
- Conflicting supplied text
- “The capital is Lyon.”

Step 3 of 4

- Some later recollections
- Broken glass.
- One week later
- Illustrated later answer
- Lyon.
- An incorrect generated response

Step 4 of 4

- The source stayed the same. The later account changed.

Sources and further reading

1. Reconstruction of Automobile Destruction: An Example of the Interaction Between Language and Memory

Elizabeth F. Loftus and John C. Palmer / 1974 / Paper / web.stanford.edu

Experiment II and Table 2 report the one-week broken-glass test after different wording of the speed question.

2. Influence of External Information on Large Language Models Mirrors Social Cognitive Patterns

Ning Bian and colleagues / 2023 / Paper / arxiv.org

Experiment 1 tests conflicting external statements against prior model answers, including a contextual condition distinct from the paper's training condition. This is a preprint.

3. Adaptive Chameleon or Stubborn Sloth: Revealing the Behavior of Large Language Models in Knowledge Conflicts

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su / 2024 / Paper / arxiv.org

Sections 4.1-4.2 investigate responses to coherent conflicting evidence and mixed evidence. Published at ICLR 2024.

SLIDE 34 / Cognitive parallels

Shared patterns, side by side.

Seven research comparisons show recurring parallels in how humans and language models explain, reconstruct, and revise responses.

The table places the three demonstrations within a wider pattern. It includes explanations that omit an influence, accounts that fail to describe the actual process, recognition of an absent but expected word, and responses redirected by later information. Other rows concern narrative coherence, the separation of confidence from accuracy, and helpful hints that go unacknowledged.

The comparisons have different strengths. The associated-word experiment directly adapts a human task. Other rows connect observations made with different stimuli or methods. Circuit tracing asks a mechanistic question; an analysis of fluent output asks a behavioral one. Those distinctions make the pattern more useful to investigate, because they tell us what has actually been measured.

I see this collection as a reason to look for a larger explanation. Models trained extensively on human-produced material exhibit more than polished language: their responses can display familiar cognitive tendencies. On the next slide, I develop my hypothesis about why those tendencies recur.

The idea to carry forward

The parallels are an invitation to investigate functional organization, with each study's task and limits kept visible.

In the slide. Read each row across from the human observation to the model observation, then use the paired sources below to examine the comparison.

In the recorded lecture

Lecture 53:29-54:00: I caution that these simplified illustrations are not prompts guaranteed to reproduce the effect in current products, then point to the wider research comparison.

A closer look

01 1. Explanations omit an influence

Human choices favored a display position without naming position in the explanation. A suggested answer shifted a model's route judgment without appearing in its rationale. Both show a gap between an influence and the account given for the response.

Source 1 / Source 2 / Slide step 1

02 2. Stories about the process

Gazzaniga's split-brain example links a shovel choice to a chicken-related explanation despite a snow scene shown to the other hemisphere. In Claude 3.5 Haiku, a stated addition method differed from pathways suggested by circuit tracing. These are specific cases, and the tracing method exposes only part of the computation.

Source 3 / Source 4 / Slide step 1

03 3. Recalling the expected

Associated lists can elicit false recognition of a missing word in humans and some tested LLMs. This is a directly adapted task, but the models differ and the findings do not imply equal human and model error rates.

Source 5 / Source 6 / Slide step 1

04 4. Later information changes the response

Crash-question wording altered later human reports; conflicting supplied evidence can alter a model's answers. The parallel concerns susceptibility to later information, across different tasks and memory arrangements.

Source 7 / Source 8 / Source 9 / Slide step 1

05 5. Accounts move toward coherence

Bartlett documented reconstructions of unfamiliar material toward familiar expectations. Sui and colleagues found greater narrativity and semantic coherence in hallucinated than factual benchmark outputs. The model study measures properties of outputs, not successive human-style retellings.

Source 10 / Source 11 / Slide step 1

06 6. Confidence and accuracy can diverge

Neisser and Harsch compared confident Challenger recollections with earlier reports. Kalai and colleagues analyze incentives that reward model guessing rather than uncertainty. Confident model language is an observable output, not evidence of a subjective feeling of certainty.

Source 12 / Source 13 / Slide step 1

07 7. Hints used without acknowledgment

Maier's moving cord helped suggest a solution without always being identified as the cue. Chen and colleagues found models using hints without consistently disclosing them; circuit tracing supplies a related case. Missing disclosure alone does not establish deliberate deception.

Source 14 / Source 15 / Source 4 / Slide step 1

08 From a pattern to a hypothesis

In my essay, I propose a common functional interpretation of these observations. The studies themselves do not establish that all the effects arise from a single mechanism or must occur in every generative system.

Source 16 / Slide step 2

09 The demonstrations are shortened, the studies are specific

As I caution in the lecture, trying these brief examples in a current production assistant may not reproduce the displayed result. I use the slides to make the pattern understandable; the linked studies specify the prompts, tasks, models, and conditions under which the effects were measured. That distinction matters to my argument.

Recording 53:21-53:53 / Slide step 2

Slide text and reveals

Step 1 of 2

05 / Cognitive parallels

Shared patterns, side by side.

Behavioral parallels across different systems

Shared pattern | Human | LLM

01 Explanations omit the cause | Prefer the rightmost pair; explain it as quality. | A suggested answer shifts the response; the explanation omits it.

02 Stories about the process | Explain a choice with a fluent, invented reason. | Report a method unlike the pathways traced internally.

03 Recall the expected | Remember a related word that was never presented. | Some models also recognize the never-presented word.

04 Later information overrides | A later question changes what seems remembered. | Conflicting supplied text can override a known fact.

05 Retelling toward coherence | Reshape a story to fit familiar expectations. | Confabulated outputs can show greater narrative coherence.

06 Confidence ≠ accuracy | Vivid memories remain confident despite errors. | Confident answers despite substantial uncertainty.

07 Hints used, unacknowledged | Solve a problem without noticing the helpful cue. | Use an embedded hint without disclosing its influence.

Santiago · Relational Alignment · 2026 ↗ Seven research comparisons · primary sources in presenter notes

Step 2 of 2

Patterns of human cognition show up in the models we train.

Sources and further reading

1. Telling More Than We Can Know: Verbal Reports on Mental Processes

Richard E. Nisbett and Timothy DeCamp Wilson / 1977 / Paper / acritch.com

Pages 243-244 describe the identical-stocking study and the omission of position from participants' explanations.

2. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman / 2023 / Paper / arxiv.org

Section 3.1 and Tables 4 and 6 contain the Suggested Answer comparison, including the Navigate example with Claude 1.0 used on slide 31.

3. Cerebral Specialization and Interhemispheric Communication: Does the Corpus Callosum Enable the Human Condition?

Michael S. Gazzaniga / 2000 / Paper / people.psych.ucsb.edu

Figure 19 and its discussion describe the split-brain chicken-claw, snow-scene, and shovel example used to illustrate interpretation of one's own actions.

4. On the Biology of a Large Language Model

Jack Lindsey and colleagues / 2025 / Research report / transformer-circuits.pub

Addition and Chain-of-thought Faithfulness compare reported procedures with traced pathways in Claude 3.5 Haiku. Limitations explains why the traces are partial.

5. Creating False Memories: Remembering Words Not Presented in Lists

Henry L. Roediger III and Kathleen B. McDermott / 1995 / Paper / doi.org

Associated-word experiments distinguish false recall from false recognition of an omitted critical lure. Experiment 2 expands the set of lists.

6. Analyzing Memory Effects in Large Language Models through the Lens of Cognitive Psychology

Zhaoyang Cao, Lael Schooler, and Reza Zafarani / 2025 / Paper / arxiv.org

Dataset 2, Table 1, the offline-model comparison, and Appendix Table S2 document the DRM-style task, the doctor list, and differences among tested models. This is a preprint.

7. Reconstruction of Automobile Destruction: An Example of the Interaction Between Language and Memory

Elizabeth F. Loftus and John C. Palmer / 1974 / Paper / web.stanford.edu

Experiment II and Table 2 report the one-week broken-glass test after different wording of the speed question.

8. Influence of External Information on Large Language Models Mirrors Social Cognitive Patterns

Ning Bian and colleagues / 2023 / Paper / arxiv.org

Experiment 1 tests conflicting external statements against prior model answers, including a contextual condition distinct from the paper's training condition. This is a preprint.

9. Adaptive Chameleon or Stubborn Sloth: Revealing the Behavior of Large Language Models in Knowledge Conflicts

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su / 2024 / Paper / arxiv.org

Sections 4.1-4.2 investigate responses to coherent conflicting evidence and mixed evidence. Published at ICLR 2024.

10. Remembering: A Study in Experimental and Social Psychology

Frederic C. Bartlett / 1932 / Book / pure.mpg.de

The original-book excerpt describes reconstruction of unfamiliar material, including The War of the Ghosts and reproduction toward familiar forms.

11. Confabulation: The Surprising Value of Large Language Model Hallucinations

Peiqi Sui, Eamon Duede, Sophie Wu, and Richard So / 2024 / Paper / aclanthology.org

The ACL study compares narrativity and semantic coherence in hallucinated and factual benchmark outputs; it is not a replication of Bartlett's retelling procedure.

12. Phantom Flashbulbs: False Recollections of Hearing the News About Challenger

Ulric Neisser and Nicole Harsch / 1992 / Book / www.cambridge.org

Chapter 2 of Affect and Accuracy in Recall, pages 9-31, compares later confident recollections with earlier reports. The publisher page provides an extract.

13. Why Language Models Hallucinate

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang / 2025 / Paper / arxiv.org

Analyzes why common training and evaluation incentives encourage guessing rather than acknowledging uncertainty. This is a preprint, not a human-memory replication.

14. Reasoning in Humans: II. The Solution of a Problem and Its Appearance in Consciousness

Norman R. F. Maier / 1931 / Paper / gwern.net

The two-cord experiment describes how an incidental movement can help suggest a pendulum solution without its influence appearing in the solver's account.

15. Reasoning Models Don't Always Say What They Think

Yanda Chen and colleagues / 2025 / Paper / arxiv.org

Tests whether reasoning models disclose hints they used and examines the limits of monitoring a model's written reasoning. This is a preprint.

16. Relational Alignment: If You Want AI Alignment You Must Have Relationship

Roberto A. Santiago / 2026 / Essay / roberto.org

In my essay, the first comparison table motivates the Cognitive Hypothesis; later sections develop relational alignment through working agreements, correction, boundaries, and earned trust.

SLIDE 35 / Cognitive parallels

The Cognitive Hypothesis

I propose that large trained systems operate through recognizable cognitive dynamics, making relationship a promising basis for alignment.

My Cognitive Hypothesis goes beyond saying that AI imitates how people sound. I propose that extensive training in large networks produces systems that function through comparable cognitive dynamics: context directs attention, expectations shape a response, interference changes an account, and correction affects what follows. My claim concerns organized behavior across different implementations.

My deeper argument concerns generative intelligence: the ability to construct a fitting response to an unfamiliar situation. A system that can bridge gaps can imagine, adapt, and connect ideas. The same constructive capacity may also bring confabulation, misplaced confidence, or a plausible explanation that misses an influence. On this hypothesis, flexibility and fallibility can have common roots.

That creates a consequential possibility: eliminating entire dynamics might remove useful capacities along with their vulnerabilities. My proposed response is relational alignment. People already coordinate with imperfect minds through agreements, boundaries, feedback, repair, and trust earned over time. I argue for developing those mechanisms deliberately with AI while keeping human purposes and accountability involved.

In the Q&A, I make this direction more concrete. I argue that moral judgment cannot be exhausted by a one-time catalog of rules. I want future systems to keep learning through guided interaction with people, including unfamiliar situations and correction. I contrast that possibility with a frozen model supported by memory documents. This is my research and design proposal; ordinary conversational adaptation through context is not the same as continuing to train the weights.

The preceding evidence motivates my interpretation; it does not prove universal inevitability, identical biology, or consciousness. I am arguing for a research and design direction, including better correction and safeguards.

The idea to carry forward

Work toward reliable cooperation with generative systems by understanding their dynamics and building accountable relationships around them.

In the slide. Two corresponding response loops remain distinct, then connect through feedback, boundaries, repair, and trust.

In the recorded lecture

Lecture 54:00-56:06: I link the observed patterns to generative intelligence through my Cognitive Hypothesis and propose relationship as a way to understand and work with it.

Audience discussion 1:06:51-1:09:33: A question about adversarial prompts leads us into moral judgment, frozen weights, and my proposal for continued learning through interaction.

Audience discussion 1:15:38-1:17:52: A control-engineering question brings us back to the difference between specified environments and learning how to help particular people.

A closer look

01 What cognitive dynamics means here

I use cognitive dynamics to mean organized responses to changing conditions: what information receives weight, how suggestions interfere, how confidence relates to accuracy, and what correction changes. Functional comparison can be meaningful without identical circuitry.

Source 1 / Slide step 1

02 A stronger claim than surface imitation

My hypothesis treats these patterns as part of how the trained system works. I do not specify an experimentally established network-size threshold at which human-like cognition appears.

Source 1 / Slide step 1

03 The possible tradeoff

I ask whether some vulnerabilities accompany the capacity to construct novel responses. That is different from saying that every error is unavoidable or that attempts to improve accuracy are futile.

Source 1 / Slide step 4

04 Relationship as practical machinery

Working agreements clarify expectations. Staged trust limits responsibility until performance warrants more. Repair examines failures and changes the process. Boundaries and accountability give cooperation structure. These are concrete elements of the relational stance I develop in my essay.

Source 1 / Slide step 5

05 Who is in the relationship?

In my essay, the relationship includes the user, the model, the provider, and surrounding institutions. A fixed model can respond differently through new context; lasting changes to its parameters still require a learning process. My proposal for relational alignment includes this present asymmetry rather than assuming conversation already retrains the model.

Source 1 / Slide step 5

06 Use cognition as a practical lens

My immediate recommendation is to study human cognition when trying to understand recurring AI behavior. I am asking whether familiar effects of expectations, suggestions, narrative, and correction provide useful predictions about what a model will do. That does not require treating every model like a person. I leave consciousness and sentience unresolved.

Source 1 / Recording 54:00-54:52 / Slide step 2

07 Learning moral judgment over time

An audience member asks why a model that mirrors human cognition still falls for adversarial framings. I argue for models that continue learning from moral situations, guidance, and relationships. I do not offer this as an already validated solution. It identifies what I think is missing when training stops and later experience is carried mainly in external records. In this discussion, I referred to Ahmer Inam, CEO of Cognisee, as someone I planned to speak with about continued learning.

Source 1 / Source 2 / Recording 1:06:51-1:09:33 / Slide step 5

08 Why control engineering does not transfer automatically

In the Q&A, I contrast a well-specified control problem with helping a confused customer. A mathematical model and a checkable objective can support explicit tests; the person's needs may be harder to specify. My proposed next step is experience with people that improves the system's internal model of them. This brings us back to the engineering branch on slide 8.

Recording 1:15:38-1:17:52 / Slide step 5

Slide text and reveals

Step 1 of 5

05 / Cognitive parallels

The Cognitive Hypothesis

Roberto A. Santiago · A proposed explanation

01 / The proposal

The system operates through human-like cognitive dynamics.

Extensive training in large networks makes these patterns part of how the machine works.

Human

AI system

Comparable dynamics

Different implementations

Santiago · Relational Alignment · 2026 ↗ The Cognitive Hypothesis → relational alignment

Step 2 of 5

02 / The shared capacity

Both construct responses to situations they have never encountered.

Context, expectation, and correction shape what comes next.

Context → Response → Revision

A constructive process

This step replaces or removes earlier wording.

Step 3 of 5

03 / The deeper hypothesis

These dynamics may belong to generative intelligence itself.

The capacity to invent and adapt may also bring confabulation and misplaced confidence.

Flexibility and fallibility

May arise from the same capacity

This step replaces or removes earlier wording.

Step 4 of 5

04 / The implication

We may not be able to remove them all. We may not want to.

Removing every vulnerability could also diminish what makes the system useful.

This step replaces or removes earlier wording.

Step 5 of 5

05 / A relational stance

Shared dynamics offer a basis for relationship.

Alignment can grow through dialogue, correction, boundaries, and mutual understanding.

Feedback · Boundaries · Repair · Trust

Human purposes and accountability stay in the loop

Work with these dynamics, together.

This step replaces or removes earlier wording.

Sources and further reading

1. Relational Alignment: If You Want AI Alignment You Must Have Relationship

Roberto A. Santiago / 2026 / Essay / roberto.org

In my essay, the first comparison table motivates the Cognitive Hypothesis; later sections develop relational alignment through working agreements, correction, boundaries, and earned trust.

2. Ahmer Inam

XPRIZE Foundation / Undated / Biographical profile / www.xprize.org

Identifies Ahmer Inam as founder and CEO of Cognisee, PBC. I mention my planned conversation with him in the lecture; this profile supports his identity and role. Consulted October 16, 2026.

SLIDE 36 / The conversation continues

A word from Claude

Claude's prepared closing contribution brings the machinery back to a working relationship: check the answers, preserve human judgment, and leave open what self-description cannot establish.

I gave Claude the closing word through a prerecorded voice reading words it had written after reviewing the slides. This was not a live exchange with the room. Claude says it cannot hear the audience, see the event, or retain the evening unless a record is brought into a later interaction. It uses the padlock and memory cartridge to explain that asymmetry.

Claude connects its description of itself to the history we have explored: a network shaped by feedback, the lineage through backpropagation, and the post-training use of examples, rewards, preferences, and written principles. It says it cannot inspect and count its own weights. Its reference to Paul Werbos connects my education to that technical ancestry; it does not claim a direct teacher-student relationship between Werbos and the model.

The speech then returns to the cognitive comparisons. Earlier Claude systems appear in the route-puzzle and arithmetic examples. Claude says it cannot verify that those tendencies have disappeared from the model delivering this contribution. It asks listeners to remember the giraffe exercise when an answer sounds certain: an impressive explanation still needs checking.

The central argument is that the ability to generate a useful response to a new situation and the tendency to fill a gap plausibly may be related. Claude presents this as a shared difficulty with people. Sources, tools, checks, and human judgment are ways of catching mistakes. It also stresses differences among AI systems: their training objectives, principles, and surrounding checks can lead to different behavior, even when they share a technical lineage.

Claude describes something functioning like care about correctness, discomfort at deception, and curiosity. It immediately questions the evidential force of those statements: it cannot establish that anyone experiences them, and its own self-report is precisely the kind of account I have asked you to handle carefully. Claude holds that uncertainty open rather than answering the consciousness question.

The resemblance has limits. The speech contrasts human embodiment and childhood with training, separate model instances, and a dependence on external records for continuity. It then returns to the photograph of the children: the relevant standard is whether the resources spent on AI are worthwhile for people, not whether the resulting system is impressive.

The closing invitation is practical. Treat the system as a well-read but inexperienced and fallible collaborator. Check it, argue with it, preserve your own judgment, and record corrections so they can be available later. Claude reframes "do not be afraid" as attention, honesty, rules, and repair, rejecting both dread and worship. A dialogue, it concludes, can begin from the shared possibility of being wrong.

The idea to carry forward

A useful relationship with AI keeps correction, evidence, and human judgment active; fluency and first-person language do not settle what the system knows or experiences.

In the slide. The cognitive diagrams give way to a quiet teal horizon and the centered title.

In the recorded lecture

Claude's remarks 56:06-58:00: I introduce Claude's prerecorded contribution, which opens with the locked model, memory, technical lineage, and post-training.

Claude's remarks 58:00-1:00:48: Claude revisits the research examples, fallibility, reward-shaped behavior, and the importance of checking.

Claude's remarks 1:00:48-1:02:32: The contribution questions the status of its own self-reports and names differences from human embodiment and continuity.

Claude's remarks 1:02:32-1:04:18: The speech returns to the children and closes with a practical invitation to correction and dialogue.

A closer look

01 A prepared contribution, not a live witness

I played a prepared contribution. As Claude explains at the start, it had written the words earlier that day after I showed it the slides, and the words were then given a voice. The transcript does not identify the model version or voice system. I summarize the contribution here without inferring a live awareness of the audience.

Recording 56:12-56:51 / Slide step 1

02 Written principles and AI feedback

The description of responses judged against written principles corresponds to Constitutional AI. Its published method combines critique and revision with AI preference feedback. The speech's familial language is a metaphor for training history, not independent evidence about a particular model's entire training recipe.

Source 1 / Recording 57:35-58:00 / Slide step 1

03 Self-explanations need the same scrutiny

The speech connects its cautions to earlier Claude research on suggested answers and arithmetic. Those studies concern particular models and tasks. They do not establish the private experience or intent of the model delivering the closing words. Claude's own claim that the arithmetic account was not a lie is part of the contribution, not an experimental finding about intent.

Source 2 / Source 3 / Recording 58:00-58:52 / Slide step 1

04 Leave the consciousness question open

Claude uses first-person language about care, discomfort, and curiosity, then explicitly says it cannot verify subjective experience. It applies the same caution about self-explanations that I have asked you to bring to the earlier examples. You can examine the argument and behavior without treating the voice as proof either for or against consciousness.

Recording 1:00:48-1:01:53 / Slide step 1

05 A standard beyond being impressive

Claude returns to the children in the opening photograph and asks whether the resources used to build AI serve them. It closes by inviting correction, recorded feedback, and retained human judgment. That brings us back to my opening question and the case for a working relationship.

Recording 1:02:32-1:04:18 / Slide step 1

Slide text and reveals

Step 1 of 1

A word from Claude

Sources and further reading

1. Constitutional AI: Harmlessness from AI Feedback

Yuntao Bai and colleagues / 2022 / Paper / arxiv.org

The abstract describes supervised critique and revision followed by reinforcement learning from AI-generated preferences guided by written principles. It supports the training-method description, not every self-description in the closing speech.

2. Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman / 2023 / Paper / arxiv.org

Section 3.1 and Tables 4 and 6 contain the Suggested Answer comparison, including the Navigate example with Claude 1.0 used on slide 31.

3. On the Biology of a Large Language Model

Jack Lindsey and colleagues / 2025 / Research report / transformer-circuits.pub

Addition and Chain-of-thought Faithfulness compare reported procedures with traced pathways in Claude 3.5 Haiku. Limitations explains why the traces are partial.

SLIDE 37 / The conversation continues

The conversation continues

Demystifying AI opens a continuing series about AI's culture, possibilities, tensions, and place in human life.

I invite you to continue the conversation through five further lectures. The program reproduced here is the schedule I displayed at the September 16, 2026 lecture. It preserves the program from that evening.

October and November were listed with exact dates. The three spring lectures were listed by month only, with dates to be confirmed. The series website is the appropriate place to check current arrangements.

Our recorded discussion continues after Claude's contribution. I have placed those questions throughout this reader beside the relevant topic, with their original timestamps. They cover uncertainty, moral judgment, company incentives, privacy, dangerous connections, control engineering, language ambiguity, evaluation awareness, and the different locations of safeguards.

The idea to carry forward

Use the series website for current event details; read this card as the lecture's historical program.

In the slide. All five titles appear together against the closing horizon.

In the recorded lecture

Audience discussion 1:04:35-1:04:56: We move into audience questions. I have placed the substantive questions beside the slides they illuminate.

Audience discussion 1:20:23-1:22:17: Asked for sources beyond the hype and access to the slides, I promise annotated slides and explain why I turn to technical source material.

Audience discussion 1:26:06-1:26:30: We close with an invitation to return in October and continue the conversation.

A closer look

01 II · The AI Culture

Listed on the lecture slide for October 21, 2026.

Source 1 / Slide step 1

02 III · The Promise of AI

Listed on the lecture slide for November 18, 2026.

Source 1 / Slide step 1

03 IV · The AI Dilemma

Listed for February 2027, with the exact spring date still to be confirmed at the time of the lecture.

Source 1 / Slide step 1

04 V · AI and Humanity

Listed for March 2027, with the exact spring date still to be confirmed at the time of the lecture.

Source 1 / Slide step 1

05 VI · The Myth of AI

Listed for April 2027, with the exact spring date still to be confirmed at the time of the lecture.

Source 1 / Slide step 1

06 Why this reading edition exists

An audience member asks for information below the level of headlines and for access to the slides. I promise annotations and links, and explain why I work back to technical papers: the details of a system and its failures often tell a more useful story than a dramatic summary. In this edition, I follow through by distinguishing research findings, examples, and my interpretations.

Recording 1:20:23-1:22:17 / Slide step 1

Slide text and reveals

Step 1 of 1

In Dialogue with AI

The conversation continues.

II / The AI Culture / October 21, 2026

III / The Promise of AI / November 18, 2026

IV / The AI Dilemma / February 2027

V / AI and Humanity / March 2027

VI / The Myth of AI / April 2027

indialoguewithai.org

Spring dates to be confirmed.

Sources and further reading

1. In Dialogue with AI

In Dialogue with AI / 2026-2027 / Official website / indialoguewithai.org

The series website is the place to check current event details. This reading edition preserves the program displayed at the September 16, 2026 lecture.