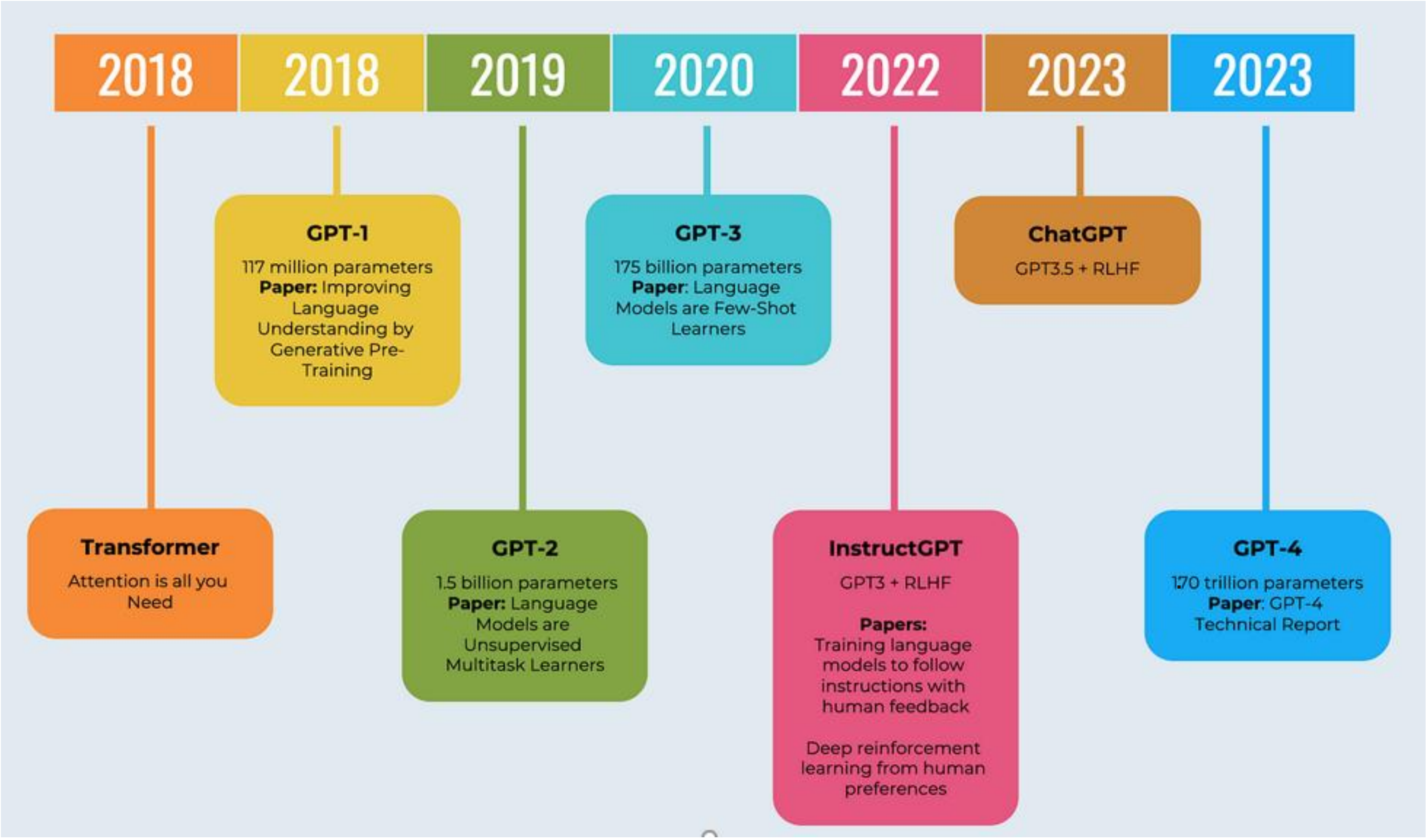
OST
Ostschweizer
Fachhochschule

A Gentle Introduction to AI
utilizing
Large Language Models

Prof. Dr. Guido M. Schuster
Director ICAI - Interdisciplinary Center for Artificial Intelligence

ICAI

State-of-the-art: GPT-4



USA

Million: 10^6

Billion: 10^9

Trillion: 10^{12} =1’000’000’000’000

GPT-4: 1’700’000’000’000 = 1.7x10^{12} Parameters

P.S.:

CH

Million: 10^6

Milliarde: 10^9

Billion: 10^{12} =1’000’000’000’000

Billiarde: 10^{15}

Trillion: 10^{18}

Where does this all come from?

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

OST

Where does this all come from?

July 7,1958 New York Times...

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

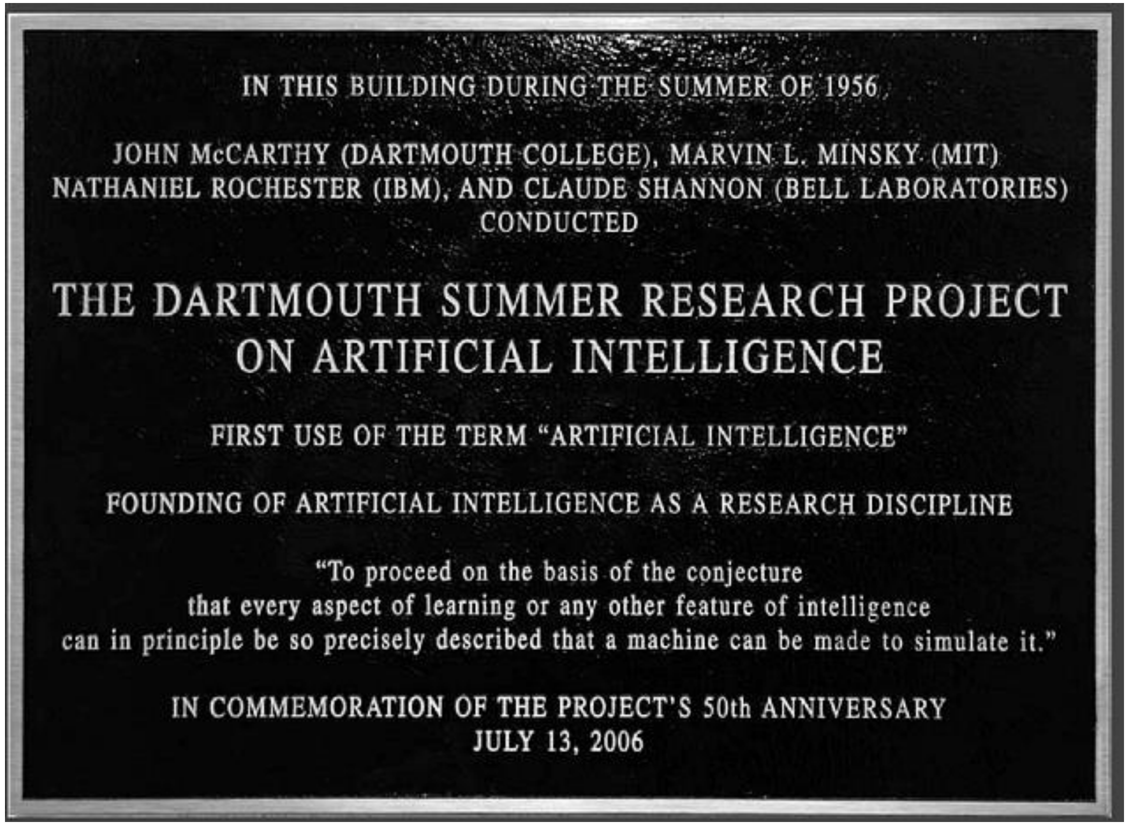
In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

OST

Summer of 1956 – the term "Artificial Intelligence" is born



Summer of 1956 – the term "Artificial Intelligence" is born

Claude Shannon

Article Talk

From Wikipedia, the free encyclopedia

Claude Elwood Shannon (April 30, 1916 – February 24, 2001) was an American mathematician, electrical engineer, computer scientist, cryptographer and inventor known as the "father of information theory" and as the "father of the Information Age".[1] Shannon was the first to describe the Boolean gates (electronic circuits) that are essential to all digital electronic circuits, and was one of the founding fathers of artificial intelligence.[2][3][4][1] Shannon is credited with laying the foundations of the Information Age.[5][6][7]

1937: A founder of modern computers
- Electronic circuits can be used to implement Boolean Algebra

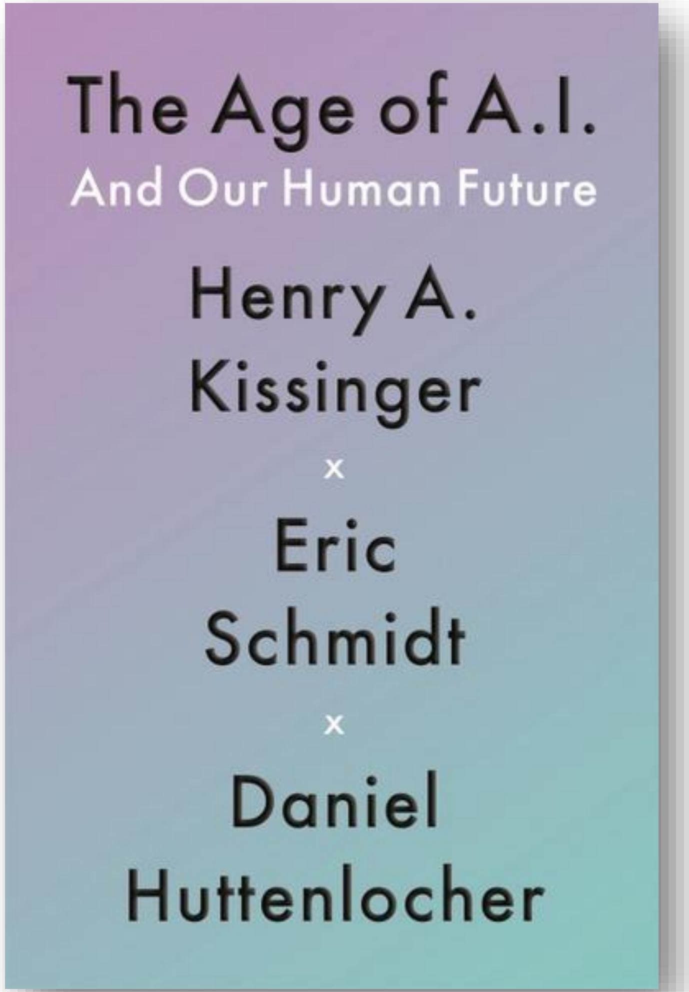
1939-45: A founder of modern cryptography
- "A Mathematical Theory of Cryptography"

1948: A founder of information theory
- "A Mathematical Theory of Communication"

1956: A founder of artificial intelligence
- Theseus was the first electrical device to learn by trial and error, being one of the first examples of artificial intelligence

Age of AI

- **We are entering the Age of AI**
  - Comparable in its effects only to the introduction of
    - Fire
    - Agriculture
    - Electricity
- **Decisions, Decisions, Decisions …**
  - In the age of AI, computers make decisions for and about us every day
    - Work: HR Analytics
    - Love: Tinder
    - Entertainment: YouTube Video
    - Finances: Credit Card Approval
    - And many more …

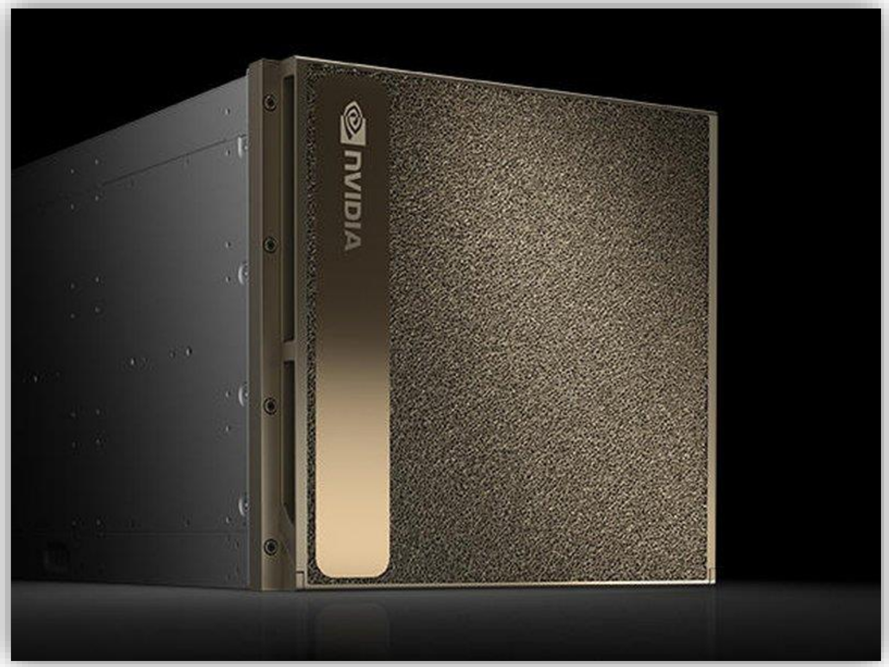


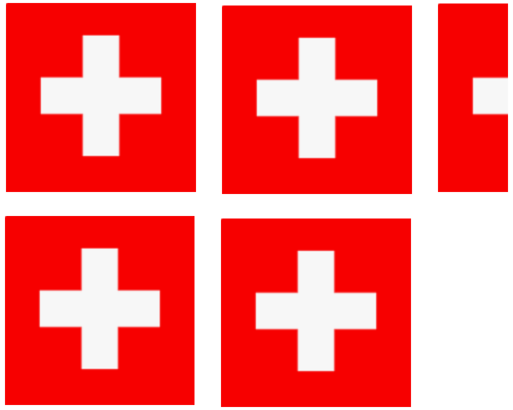
Data

- **Data** is fundamental to AI, but what kind of data is there and where does it come from?

- IT systems
  - Books
  - Wikipedia
  - The Internet
  - Medical records
  - Credit cards
  - Access cards
  - Browser history …
- Dedicated sensors
  - Cameras
  - Microphones
  - Pressure sensors
  - IMU
  - LIDAR
  - RADAR
  - And many more …

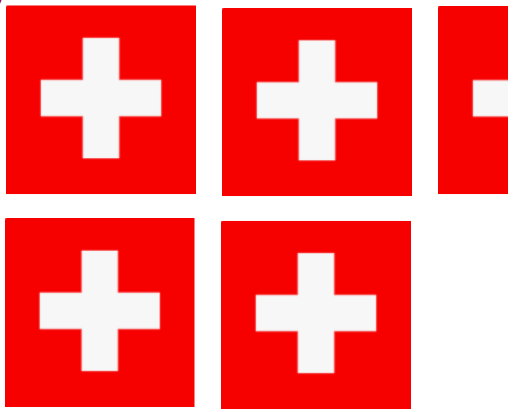
Data & Computing

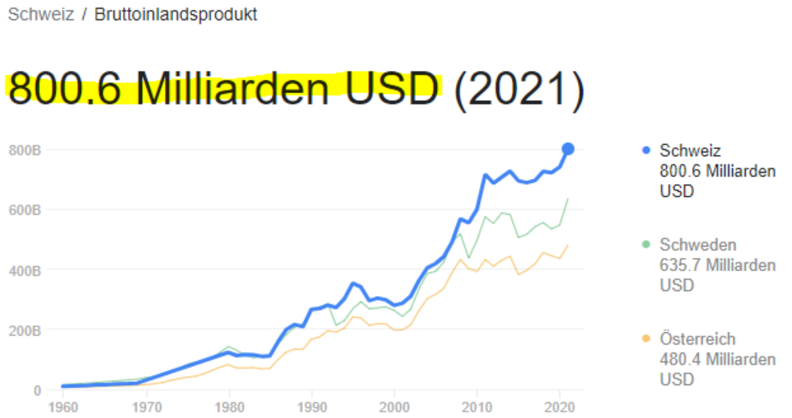




- DGX-2 Deep Learning Supercomputer
  - 2 Petaflops (2e15) per second
  - Earth: 8e9 people
    - 2e15/8e9=1/4e6= <u>250’000 flops/person per second</u>

Data & Computing x 16

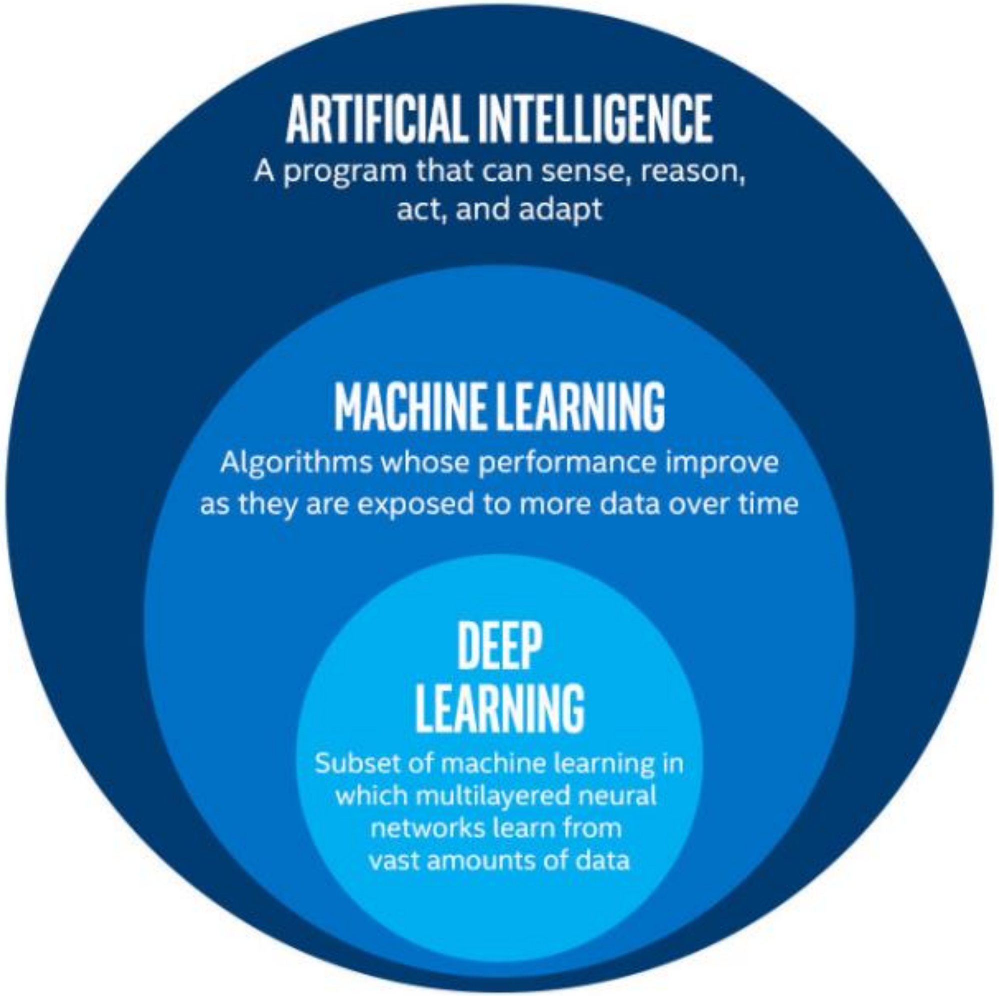


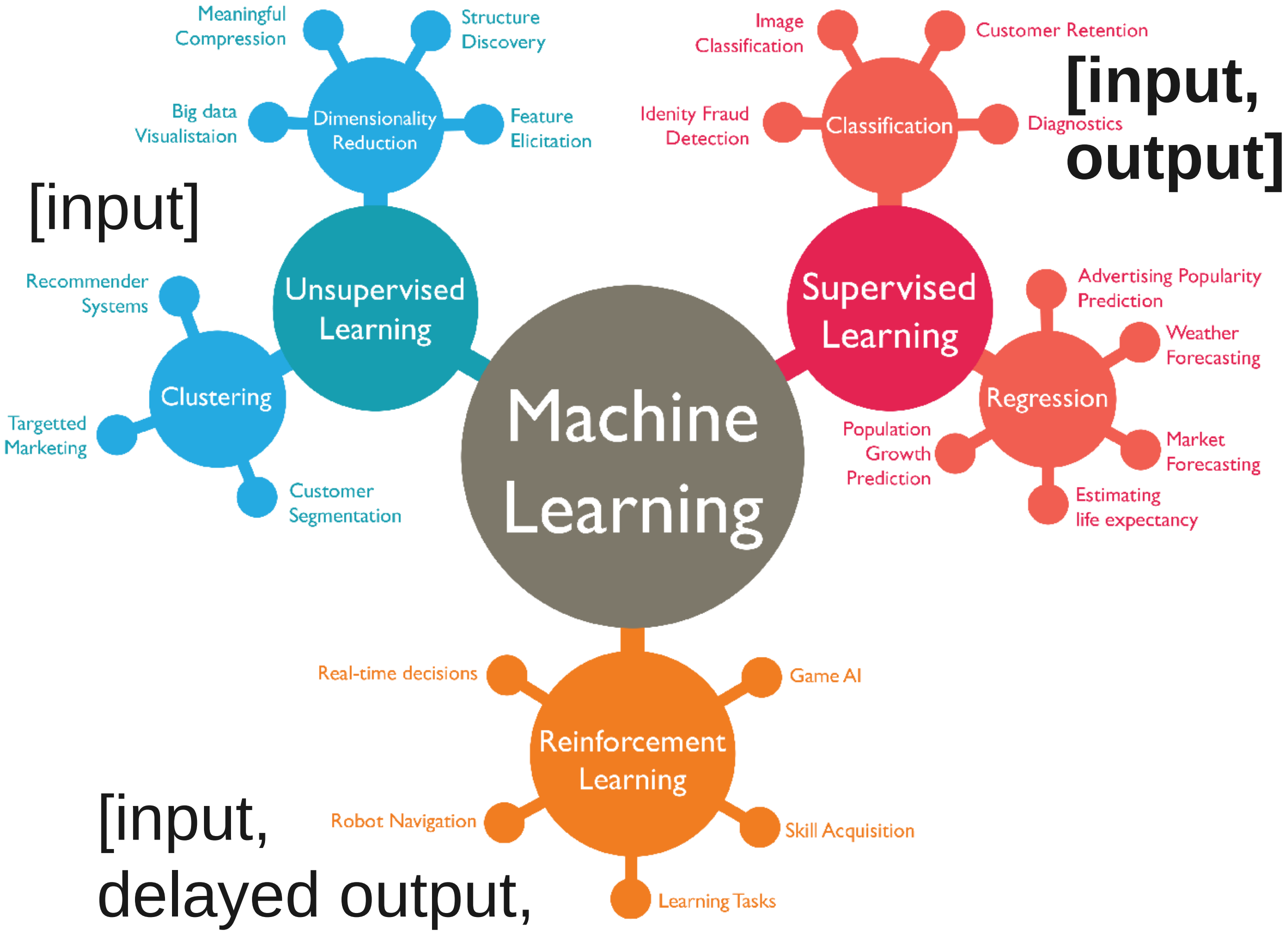


- DGX H200 Deep Learning Supercomputer
  - 32 Petaflops (2^5e15) per second
  - Earth: 8e9 people
    - 2^5e15/8e9=4e6= <u>4’000’000 flops/person per second</u>

Data & Computing & Algorithms

[What kind of data?]





[input]

[input, output]

[input,
delayed output,
grade for this delayed output]

Supervised Machine Learning

- In AI, there is a recurring theme:
  - Examples (training data) are used to teach a computer a pattern <u>between input data and output data</u>
  - The computer then generalizes this pattern to make reasonable decisions (similar to the examples) in new situations (test data)

- This is **supervised machine learning** and the most successful application of AI

<u>The computer develops an</u> **intuition** for the situation

This is similar to a human, but <u>humans need fewer examples</u>

Supervised Machine Learning

- The output of an AI system is **usually not** a simple decision in terms of **YES** or **NO**
  - AI estimates probabilities from example data, which decision to make, a human must influence the final decision by setting the cost of those decisions

For a measured temperature, is this person

healthy or **sick**?

Temperature → **AI** Data & Costs **→ healthy?** **→ sick?**

OST

Supervised Machine Learning

- **A simple example**
  - Based on the temperature, the AI estimates the probability whether a person is **healthy** or **sick**

    P(**healthy**|temperature)

  Note:
    P(**sick**|temperature) = 1-P(**healthy**|temperature)

  - For this crucial estimation, data from the past (examples, also called training data) are used, where doctors have made this decision

TEMPERATURE MEASUREMENT

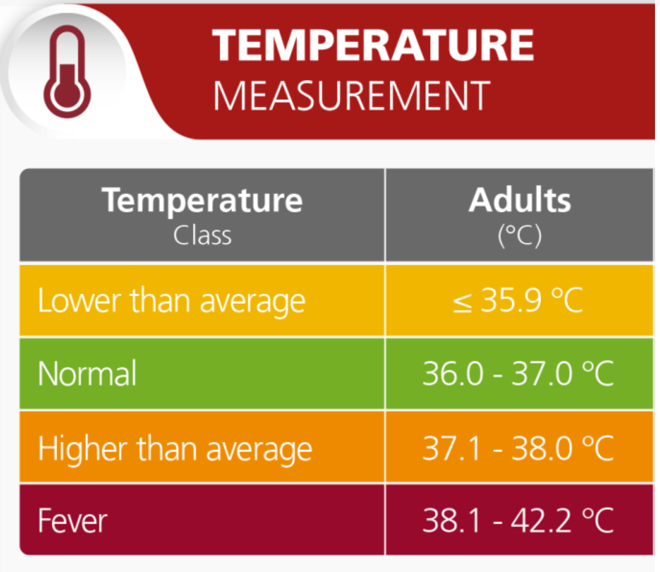
Temperature Class	**Adults** (°C)
Lower than average	≤ 35.9 °C
Normal	36.0 - 37.0 °C
Higher than average	37.1 - 38.0 °C
Fever	38.1 - 42.2 °C

Supervised Machine Learning

- **Training data**:

Person #	Temperature [C]	Doctor decision [healthy] [sick]
1	37.1	healthy
2	36.9	healthy
3	39.4	sick
4	40.1	sick
5	38.2	healthy
6	36.9	healthy
…	…	…
100'000	41.2	sick

- **Measurement**: Temperature T in Celsius
- **Decision**: **healthy** or **sick**



TEMPERATURE MEASUREMENT

Temperature Class	Adults (°C)
Lower than average	≤ 35.9 °C
Normal	36.0 - 37.0 °C
Higher than average	37.1 - 38.0 °C
Fever	38.1 - 42.2 °C

- **Cost of the decision:**
  - More to this later

		Truth!	
		healthy!	**sick!**
AI?	**healthy?**	C_HH=0	C_HS=1
	sick?	C_SH=1	C_SS=0

OST

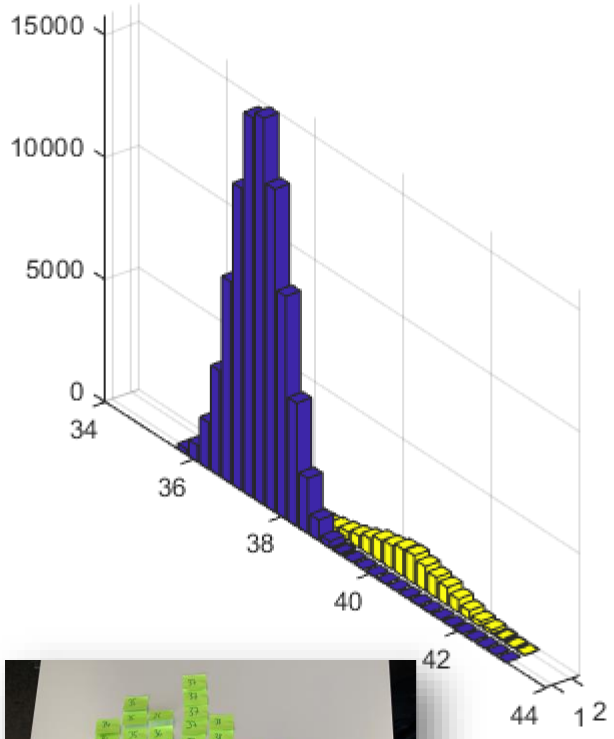
Supervised Machine Learning

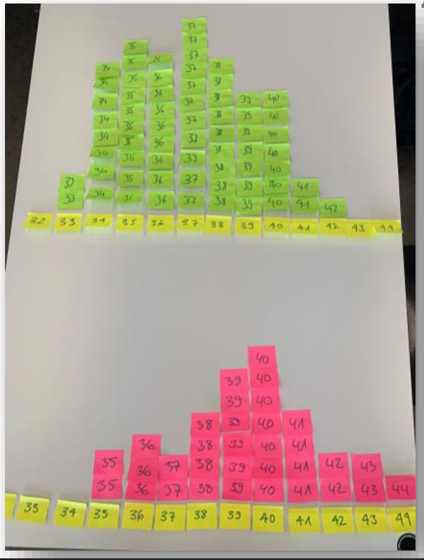
- **Training data (examples):**

Person #	Temperature [C]	Decision [healthy] [sick]
1	37.1	healthy
2	36.9	healthy
3	39.4	sick
4	40.1	sick
5	38.2	healthy
6	36.9	healthy
…	…	…
100'000	41.2	sick

Create a 2D Histogramm

Temperature	#healthy	#sick
35.75	229	0
36.00	676	0
36.25	1910	0
36.50	4314	1
36.75	8253	1
37.00	12356	3
37.25	15542	2
37.50	15812	4
37.75	13209	13
38.00	9119	32
38.25	5044	76
38.50	2261	120
38.75	850	232
39.00	271	352
39.25	65	485
39.50	9	664
39.75	1	882
40.00	0	964
40.25	0	1111
40.50	0	1154
40.75	0	1004
41.00	0	845
41.25	0	733
41.50	0	544
41.75	0	360
42.00	0	211
42.25	0	153
42.50	0	82
42.75	0	28
43.00	0	23





- Training data (examples) where doctors made the decisions (**healthy** or **sick**)

- <u>**This is the entire knowledge of the AI**</u>

OST

Supervised Machine Learning

- **Optimal decisions making**
  - A reasonable goal is to make as few mistakes as possible
  - **Bayes' rule**
    - For a measured temperature, estimate (based on the training data) the probabilities that the person is **healthy** or **sick**
      - P(**healthy**|temperature)
      - P(**sick**|temperature) = 1-P(**healthy**|temperature)

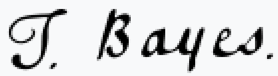
Decide **healthy**, if P(**healthy**|temperature) > ½
otherwise decide **sick**, since then P(**sick**|temperature) > ½

→On average, this is how the fewest mistakes are made

The Reverend
Thomas Bayes

Portrait purportedly of Bayes used in a 1936 book,[1] but it is doubtful whether the portrait is actually of him.[2] No earlier portrait or claimed portrait survives.

Born	c. 1701 London, England
Died	7 April 1761 (aged 59) Tunbridge Wells, Kent, Great Britain
Alma mater	University of Edinburgh
Known for	Bayes' theorem
	Scientific career
Fields	Probability
	Signature



Supervised Machine Learning

$$P(\textbf{healthy}|\text{temperature}) = \frac{(\textbf{\#healthy}|\text{temperature})}{(\textbf{\#healthy}|\text{temperature}) + (\textbf{\#sick}|\text{temperature})}$$

- Where each row in the histogram table corresponds to a given temperature
  - Thus, P(**healthy**|temperature) can be calculated per temperature (row)

Key question:

How is P(**healthy**|temperature) estimated from the **training data**?

training data

Temperature	#healthy	#sick	P(**healthy**|temperature)= #healthy/(#healthy+#sick)
35.75	229	0	1.00
36.00	676	0	1.00
36.25	1910	0	1.00
36.50	4314	1	1.00
36.75	8253	1	1.00
37.00	12356	3	1.00
37.25	15542	2	1.00
37.50	15812	4	1.00
37.75	13209	13	1.00
38.00	9119	32	1.00
38.25	5044	76	0.99
38.50	2261	120	0.95
38.75	850	232	850/(850+232)=0.79
39.00	271	352	0.43
39.25	65	485	0.12
39.50	9	664	0.01
39.75	1	882	0.00
40.00	0	964	0.00
40.25	0	1111	0.00
40.50	0	1154	0.00
40.75	0	1004	0.00
41.00	0	845	0.00
41.25	0	733	0.00
41.50	0	544	0.00
41.75	0	360	0.00
42.00	0	211	0.00
42.25	0	153	0.00
42.50	0	82	0.00
42.75	0	28	0.00
43.00	0	23	0.00

Supervised Machine Learning

- **Optimal decision**
- After measuring the temperature, find P(**healthy**|temperature) in the table (temperature indicates the row) and decide healthy if P(**healthy**|temperature) > ½

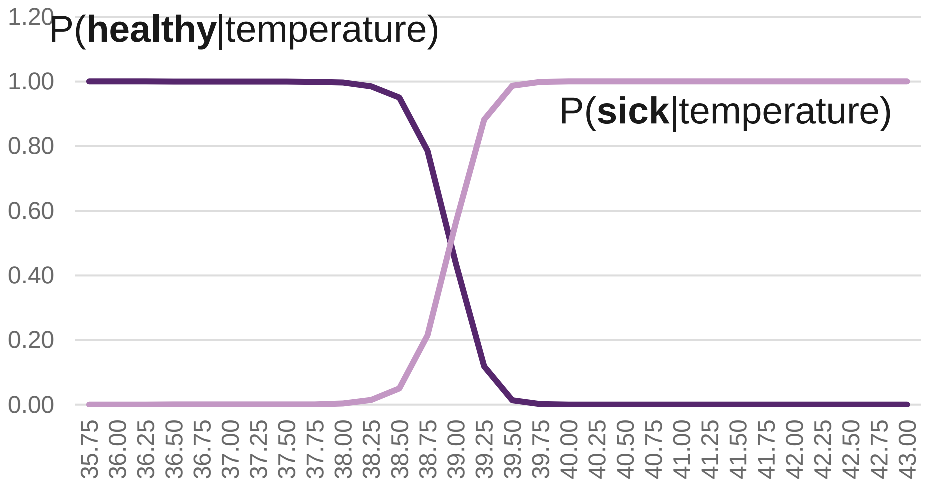
→ **Take the option with the greatest probability**

Temperature	#healthy	#sick	P(**healthy**|temperature)= #healthy/(#healthy+#sick)
35.75	229	0	1.00
36.00	676	0	1.00
36.25	1910	0	1.00
36.50	4314	1	1.00
36.75	8253	1	1.00
37.00	12356	3	1.00
37.25	15542	2	1.00
37.50	15812	4	1.00
37.75	13209	13	1.00
38.00	9119	32	1.00
38.25	5044	76	0.99
38.50	2261	120	0.95
38.75	850	232	0.79
39.00	271	352	0.43
39.25	65	485	0.12
39.50	9	664	0.01
39.75	1	882	0.00
40.00	0	964	0.00
40.25	0	1111	0.00
40.50	0	1154	0.00
40.75	0	1004	0.00
41.00	0	845	0.00
41.25	0	733	0.00
41.50	0	544	0.00
41.75	0	360	0.00
42.00	0	211	0.00
42.25	0	153	0.00
42.50	0	82	0.00
42.75	0	28	0.00
43.00	0	23	0.00

Makes on average the fewest mistakes

Decision is purely data driven!

If the data changes, the decision changes



Supervised Machine Learning

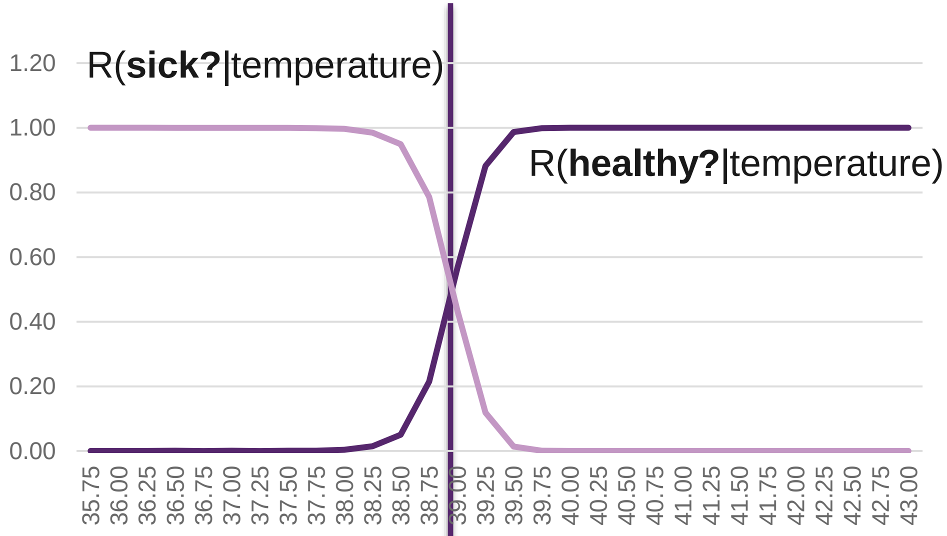
		Truth!	
		healthy!	**sick!**
AI?	**healthy?**	C_HH=0	**C_HS=1**
	sick?	**C_SH=1**	C_SS=0

- **Decision by risk minimization**

→ **Take the smallest risk**

Risk(**healthy**?|temperature) = C_HH*P(**healthy**!|temperature)+**C_HS***P(**sick**!|temperature) = 1*P(**sick**!|temperature)

Risk(**sick**?|temperature) = **C_SH***P(**healthy**!|temperature)+C_SS*P(**sick**!|temperature) = 1*P(**healthy**!|temperature)



Supervised Machine Learning

- **Decision by risk minimization**

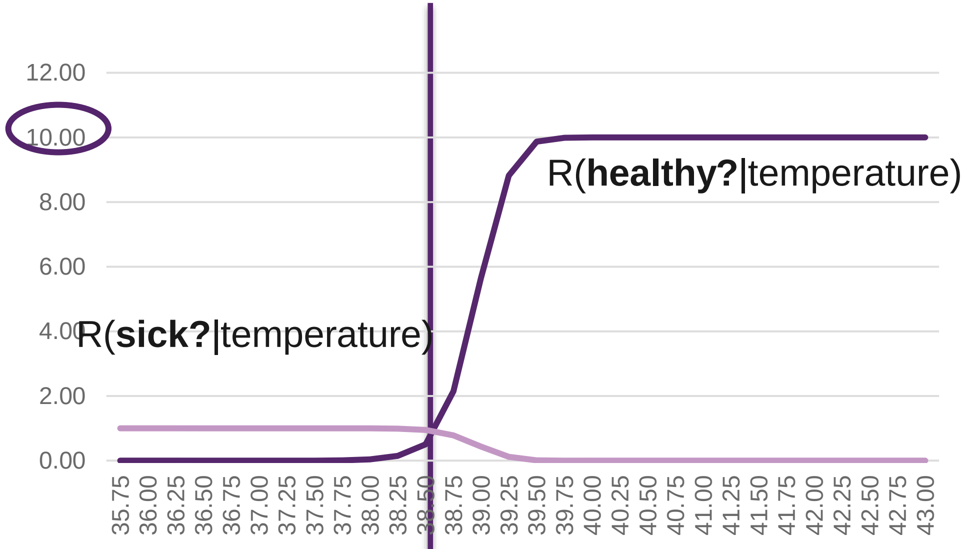
→ Take the smallest risk

		Truth!	
		healthy!	**sick!**
AI?	**healthy?**	C_HH=0	**C_HS=10**
	sick?	**C_SH=1**	C_SS=0

Risk(**healthy**?|temperature) = C_HH*P(**healthy**!|temperature)+**C_HS***P(**sick**!|temperature) = 10*P(**sick**!|temperature)

Risk(**sick**?|temperature) = **C_SH***P(**healthy**!|temperature)+C_SS*P(**sick**!|temperature) = 1*P(**healthy**!|temperature)

R(**sick?|**temperature) stayed the same!



R(**healthy?|**temperature) increased by a factor of 10!

OST

What is a language model?

- A model, which calculates **P(current word | previous words)**
  - This is the probability of a current word, given a sequence of previous words (this is called the "context")
  - For example:

    P(“meal” | “This was a nice”) = high

    P(“silly” | “This was a nice”) = low

- **Optimal decision making** according to Bayes
  - For every possible current word (“meal” or “silly”) for a given sequence of previous words (“This was a nice”), the language model estimates the probability of the possible current word
    - If the goal is, **on average to make as few mistakes as possible**, then the most probable current word should be used

What is a language model?

- **Supervised Learning:** Examples (training data) are used to teach a computer a pattern between input data and output data
  - Here, the input data is the "first word"
    and
    the output data is the "second word"
- **Training data is needed to estimate the probabilities**
  - For language models, take a large body of text and use it to estimate P(first word AND second word)

input data is "nice"

P("second word"="meal"
AND
"first word" = "nice")

output data is "meal"

What is a language model?

Training data for the upcoming simple example:

The large body of text analyzed is simple the sentence:

"The dog jumped over the fox"

repeated 100 times ☺

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox

.
.
.

096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

Note that only five words are the dictionary

The, Dog, Jumped, Over, Fox

→ **Not a realistic example**

What is a language model?

- We use the training data to fill the frequency table, describing the statistical relationship between “first word” and “second word”

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

Frequency (first word AND second word)

		second word					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**	0	100	0	0	100	200
	Dog	0	0	100	0	0	100
	Jumped	0	0	0	100	0	100
	Over	100	0	0	0	0	100
	Fox	0	0	0	0	0	0
	Total	100	100	100	100	100	500

What is a language model?

- We use the frequency table to estimate the joint probabilities, i.e., describing the statistical relationship between "first word" and "second word"

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

P(first word AND second word)

		second word					
		The	**Dog**	**Jumped**	**Over**	**Fox**	**P(first word)**
first word	**The**	0	100/500	0	0	100/500	200/500
	Dog	0	0	100/500	0	0	100/500
	Jumped	0	0	0	100/500	0	100/500
	Over	100/500	0	0	0	0	100/500
	Fox	0	0	0	0	0	0
	P(second word)	100/500	100/500	100/500	100/500	100/500	500/500

What is a language model?

- We use the frequency table to estimate the joint probabilities, i.e., describing the statistical relationship between “first word” and “second word”

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

P(first word AND second word)

		second word					
		The	**Dog**	**Jumped**	**Over**	**Fox**	**P(first word)**
first word	**The**	0	0.2	0	0	0.2	0.4
	Dog	0	0	0.2	0	0	0.2
	Jumped	0	0	0	0.2	0	0.2
	Over	0.2	0	0	0	0.2	0.2
	Fox	0	0	0	0	0	0
	P(second word)	0.2	0.2	0.2	0.2	0.2	1

OST

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word!
  - Let's say the first word is **Dog**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

P(first word AND second word)

		second word					
		The	**Dog**	**Jumped**	**Over**	**Fox**	**P(first word)**
first word	**The**	0	0.2	0	0	0.2	0.4
	Dog	0	0	0.2	0	0	0.2
	Jumped	0	0	0	0.2	0	0.2
	Over	0	0	0	0	0.2	0.2
	Fox	0	0	0	0	0	0
	P(second word)	0	0.2	0.2	0.2	0.4	1

P(second word=Jumped | first word=Dog) = P(second word=Jumped AND first word=Dog)/P(first word=Dog) = 0.2/0.2= 1

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word!
  - Let's say the first word is **Dog**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

P(second word | first word=DOG)

		second word					
	probabilities	**The**	**Dog**	**Jumped**	**Over**	**Fox**	
first word	**The**						
	Dog	0	0	1	0	0	
	Jumped						
	Over						
	Fox						

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first word is **Dog**
    - Hence, we can calculate P(second word | first word = Dog), which is shown below, a function of "second word"
    - Optimal Bayes' decision implies, that the second word must be **Jumped**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

		second word					
	probabilities	**The**	**Dog**	**Jumped**	**Over**	**Fox**	
first word	**The**						
	Dog	0	0	1	0	0	
	Jumped						
	Over						
	Fox						

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first word is **The**
    - Hence, only that row of the frequency table is needed

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

		second word					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**	0	100	0	0	100	200
	Dog						
	Jumped						
	Over						
	Fox						

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first word is **The**
    - Hence, we can calculate P(word | first word = The), which is shown below, this is a function of "second word"
    - Here Bayes' rule implies, that the second word must be either **Dog** or **Fox**, both have a probability of 50%

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

		second word					
	probabilities	**The**	**Dog**	**Jumped**	**Over**	**Fox**	
first word	**The**	0	0.5	0	0	0.5	
	Dog						
	Jumped						
	Over						
	Fox						

What is a language model?

- A model, which calculates P(word | previous words)
  - In general, the probabilities would look more like this, i.e., not so clear which is the best choice
    - One option is to simply pick the most probable one, this follows Bayes' rule → **Dog**

Let's say the first word is **The**

		second word					
	probabilities	**The**	**Dog**	**Jumped**	**Over**	**Fox**	
first word	**The**	0.0	0.4	0.2	0.1	0.3	
	Dog						
	Jumped						
	Over						
	Fox						

What is a language model?

- A model, which calculates P(word | previous words)
  - In general, the probabilities would look more like this, i.e., not so clear which is the best choice
    - Bayes' rule results in **boring** texts, so maybe it would be more interesting, to take Dog in 40% of the cases and Fox in 30% of all cases and so on
      - Hence the words would be **generated** with the same frequency as they occur in the training text

Generative AI

- This can easily be achieved, as it is shown on the right:
  - **All you need is a die with 100 equal sides →**

Let's say the first word is **The**

The	Dog	Jumped	Over	Fox
0.0	0.4	0.2	0.1	0.3

Roll out the probabilities

Dog=0.4				Jumped=0.2		Over=0.1	Fox=0.3		
1 10	11 20	21 30	31 40	41 50	51 60	61 70	71 80	81 90	91 100

What is a language model?

- A model, which calculates P(word | context)
  - The role of the **Temperature T**
    - The idea is, that higher temperature makes the choices more similar in probability
      - This called the high entropy state, i.e., there is a high uncertainty about the next word
    - Lower temperature makes the choices less similar in probability
      - This is called the low entropy state, i.e., there is little uncertainty about the next word
      - For T=0, this is Bayes' rule, for T=∞, this makes all words equally likely

	$P(The\|Context)$	$P(Dog\|Context)$	$P(Jumped\|Context)$	$P(Over\|Context)$	$P(Fox\|Context)$
	0.0	0.4	0.2	0.1	0.3

$$P_T(Dog|Context) = \frac{e^{\frac{P(Dog|Context)}{T}}}{e^{\frac{P(The|Context)}{T}} + e^{\frac{P(Dog|Context)}{T}} + e^{\frac{P(Jumped|Context)}{T}} + e^{\frac{P(Over|Context)}{T}} + e^{\frac{P(Fox|Context)}{T}}}$$

	$P(The\|Context)$	$P(Dog\|Context)$	$P(Jumped\|Context)$	$P(Over\|Context)$	$P(Fox\|Context)$
	0.0	0.4	0.2	0.1	0.3

$$P_T(Dog|Context) = \frac{e^{\frac{P(Dog|Context)}{T}}}{e^{\frac{P(The|Context)}{T}} + e^{\frac{P(Dog|Context)}{T}} + e^{\frac{P(Jumped|Context)}{T}} + e^{\frac{P(Over|Context)}{T}} + e^{\frac{P(Fox|Context)}{T}}}$$

low

Entropy
Temperature

high

T	$P_T(The\|Context)$	$\boldsymbol{P_T(Dog\|Context)}$	$P_T(Jumped\|Context)$	$P_T(Over\|Context)$	$P_T(Fox\|Context)$
0	0	**1**	0	0	0
0.1668	0.0431	**0.4746**	0.1431	0.0786	0.2606
0.2783	0.0860	**0.3619**	0.1764	0.1231	0.2526
0.4642	0.1241	**0.2939**	0.1910	0.1540	0.2369
0.7743	0.1519	**0.2547**	0.1967	0.1729	0.2238
1.2915	0.1703	**0.2321**	0.1988	0.1840	0.2148
2.1544	0.1819	**0.2190**	0.1996	0.1905	0.2091
3.5938	0.1890	**0.2113**	0.1998	0.1944	0.2055
5.9948	0.1934	**0.2067**	0.1999	0.1966	0.2033
∞	**0.2**	**0.2**	**0.2**	**0.2**	**0.2**

What is a language model?

- A model, which calculates P(word | previous words)
  - Now let's use **two** previous words as the context
    - A large body of text is analyzed and the number of occurrences of each word sequence of length 3 (3-gram) is recorded

third word = **The**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

third word = The		second word					
	occurrences	The	Dog	Jumped	Over	Fox	Total
first word	The	0	0	0	0	0	0
	Dog	0	0	0	0	0	0
	Jumped	0	0	0	100	0	100
	Over	0	0	0	0	0	0
	Fox	0	0	0	0	0	0
	Total	0	0	0	100	0	0

What is a language model?

- A model, which calculates P(word | previous words)
  - Now let's use two previous words as the context
    - A large body of text is analyzed and the number of occurrences of each word sequence of length 3 (3-gram) is recorded

third word = **Dog**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

third word = Dog		second word					
	occurrences	The	Dog	Jumped	Over	Fox	Total
first word	The	0	0	0	0	0	0
	Dog	0	0	0	0	0	0
	Jumped	0	0	0	0	0	0
	Over	0	0	0	0	0	0
	Fox	0	0	0	0	0	0
	Total	0	0	0	0	0	0

What is a language model?

- A model, which calculates P(word | previous words)
  - Now let's use two previous words as the context
    - A large body of text is analyzed and the number of occurrences of each word sequence of length 3 (3-gram) is recorded

third word = **Jumped**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

third word = Jumped		**second word**					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**	0	100	0	0	0	100
	Dog	0	0	0	0	0	0
	Jumped	0	0	0	0	0	0
	Over	0	0	0	0	0	0
	Fox	0	0	0	0	0	0
	Total	0	100	0	0	0	100

What is a language model?

- A model, which calculates P(word | previous words)
  - Now let's use two previous words as the context
    - A large body of text is analyzed and the number of occurrences of each word sequence of length 3 (3-gram) is recorded

third word = **Over**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.

096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

third word = Over		**second word**					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**	0	0	0	0	0	0
	Dog	0	0	100	0	0	100
	Jumped	0	0	0	0	0	0
	Over	0	0	0	0	0	0
	Fox	0	0	0	0	0	0
	Total	0	0	100	0	0	100

What is a language model?

- A model, which calculates P(word | previous words)
  - Now let's use two previous words as the context
    - A large body of text is analyzed and the number of occurrences of each word sequence of length 3 (3-gram) is recorded

third word = **Fox**

third word = Fox		second word					
	occurrences	The	Dog	Jumped	Over	Fox	Total
first word	The	0	0	0	0	0	0
	Dog	0	0	0	0	0	0
	Jumped	0	0	0	0	0	0
	Over	100	0	0	0	0	100
	Fox	0	0	0	0	0	0
	Total	100	0	0	0	0	100

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first two words are **Dog Jumped**
    - Hence, only that part of the frequency table is needed

third word = **The**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

third word = The		**second word**					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**						
	Dog			0			
	Jumped						
	Over						
	Fox						
	Total						

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first two words are **Dog Jumped**
    - Hence, only that part of the frequency table is needed

third word = **Dog**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

third word = Dog		**second word**					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**						
	Dog			0			
	Jumped						
	Over						
	Fox						
	Total						

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first two words are **Dog Jumped**
    - Hence, only that part of the frequency table is needed

third word = **Jumped**

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

third word = Jumped		**second word**					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**						
	Dog			0			
	Jumped						
	Over						
	Fox						
	Total						

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first two words are **Dog Jumped**
    - Hence, only that part of the frequency table is needed

third word = **Over**

third word = Over		**second word**					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**						
	Dog			100			
	Jumped						
	Over						
	Fox						
	Total						

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

OST

What is a language model?

- A model, which calculates P(word | previous words)
  - Now we can let the model predict the next word! Let's say the first two words are **Dog Jumped**
    - Hence, only that part of the frequency table is needed

third word = **Fox**

third word = Fox		**second word**					
	occurrences	**The**	**Dog**	**Jumped**	**Over**	**Fox**	Total
first word	**The**						
	Dog			0			
	Jumped						
	Over						
	Fox						
	Total						

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

What is a language model?

- A model, which calculates P(word | previous words)
- Clearly this is a very simple example
  - Only **five** words in the dictionary

    The, Dog, Jumped, Over, Fox
  - Only one sentence 100 times in the training data
    - The dog jumped over the fox →

001) The dog jumped over the fox
002) The dog jumped over the fox
003) The dog jumped over the fox
004) The dog jumped over the fox
005) The dog jumped over the fox
.
096) The dog jumped over the fox
097) The dog jumped over the fox
098) The dog jumped over the fox
099) The dog jumped over the fox
100) The dog jumped over the fox

Hence, the estimated probabilities are not representative of anything ☺

We need more and better training data!

What is a language model?

- **Curse of dimensionality**
  - Oxford English Dictionary estimates that there are around <u>200,000 English words</u> in use
  - Assume you want <u>5 occurrences per word</u>, and all words a uniformly used
    - 1 Million words (10^6) are needed for training a 1-gram (about 10 books)

10 books

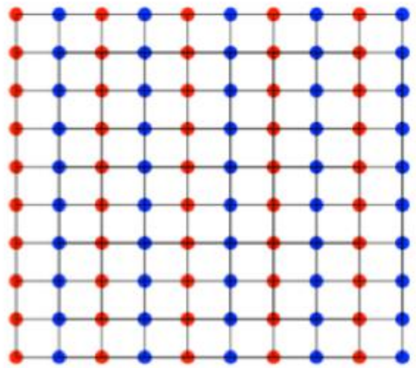
(A) 1-D

10 books needed for 1-gram
100 books needed for 2-gram
1000 books needed for 3-gram
10’000 books needed for 4-gram
100’000 books needed for 5-gram

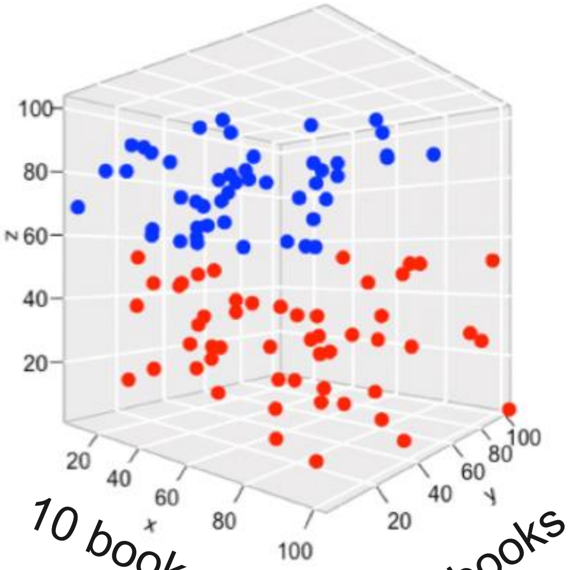
The context was 1 “previous word”, i.e., we calculated the probability of every combination of 2 words, the so called 2-gram

10 books

10 books



(B) 2-D



10 books

10 books

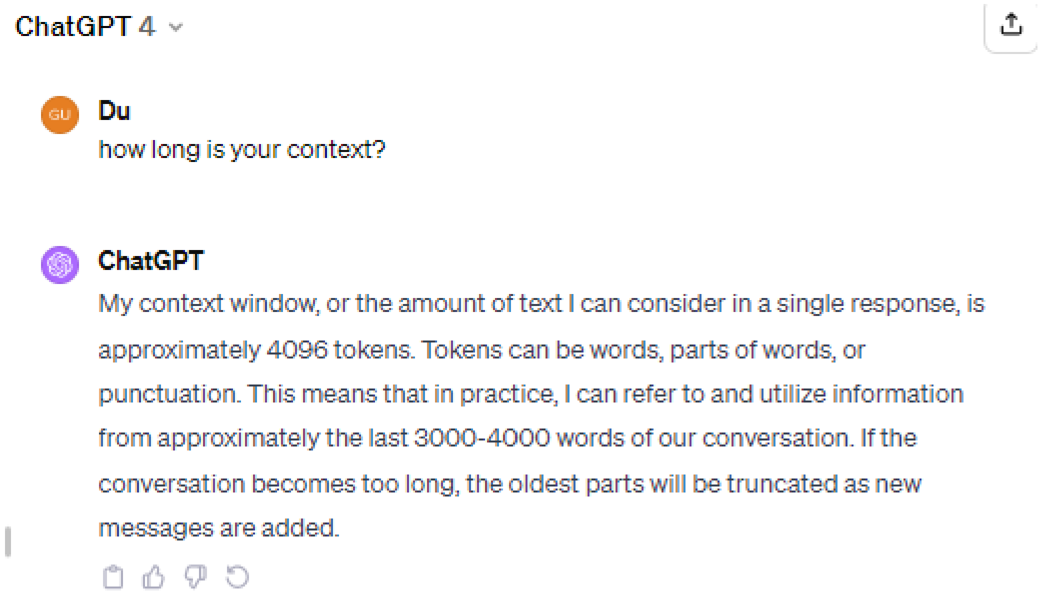
10 books

(C) 3-D

The context is 2 “previous words”, i.e., we calculated the probability of every combination of 3 words, the so called 3-gram

OST

What is a language model?


ChatGPT 4

Du
how long is your context?

ChatGPT
My context window, or the amount of text I can consider in a single response, is approximately 4096 tokens. Tokens can be words, parts of words, or punctuation. This means that in practice, I can refer to and utilize information from approximately the last 3000-4000 words of our conversation. If the conversation becomes too long, the oldest parts will be truncated as new messages are added.

Books of the world, stand up and be counted! All 129,864,880 of you.
Thursday, August 05, 2010 at 8:26 AM
Posted by Leonid Taycher, software engineer

129’864’880 books are needed to train an 8-gram – i.e., a context of 7 words to predict the next word

$P(word_8|word_1,word_2,word_3,word_4,word_5,word_6,word_7)$

GPT 4 uses ~3000-gram, i.e., would need about 10^{3000} books for training

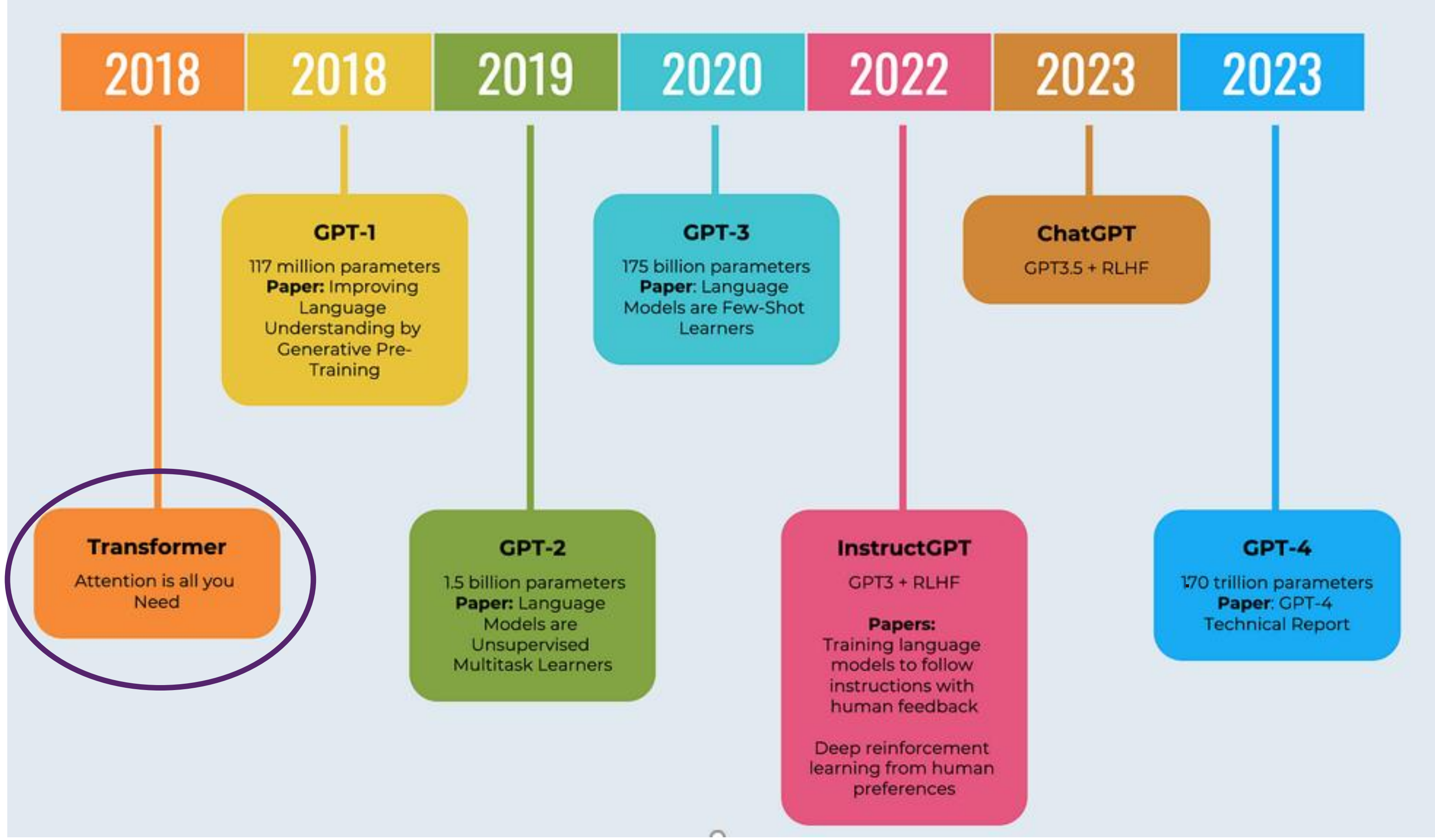
>

It is estimated that there are about 10^{82} atoms in the known universe

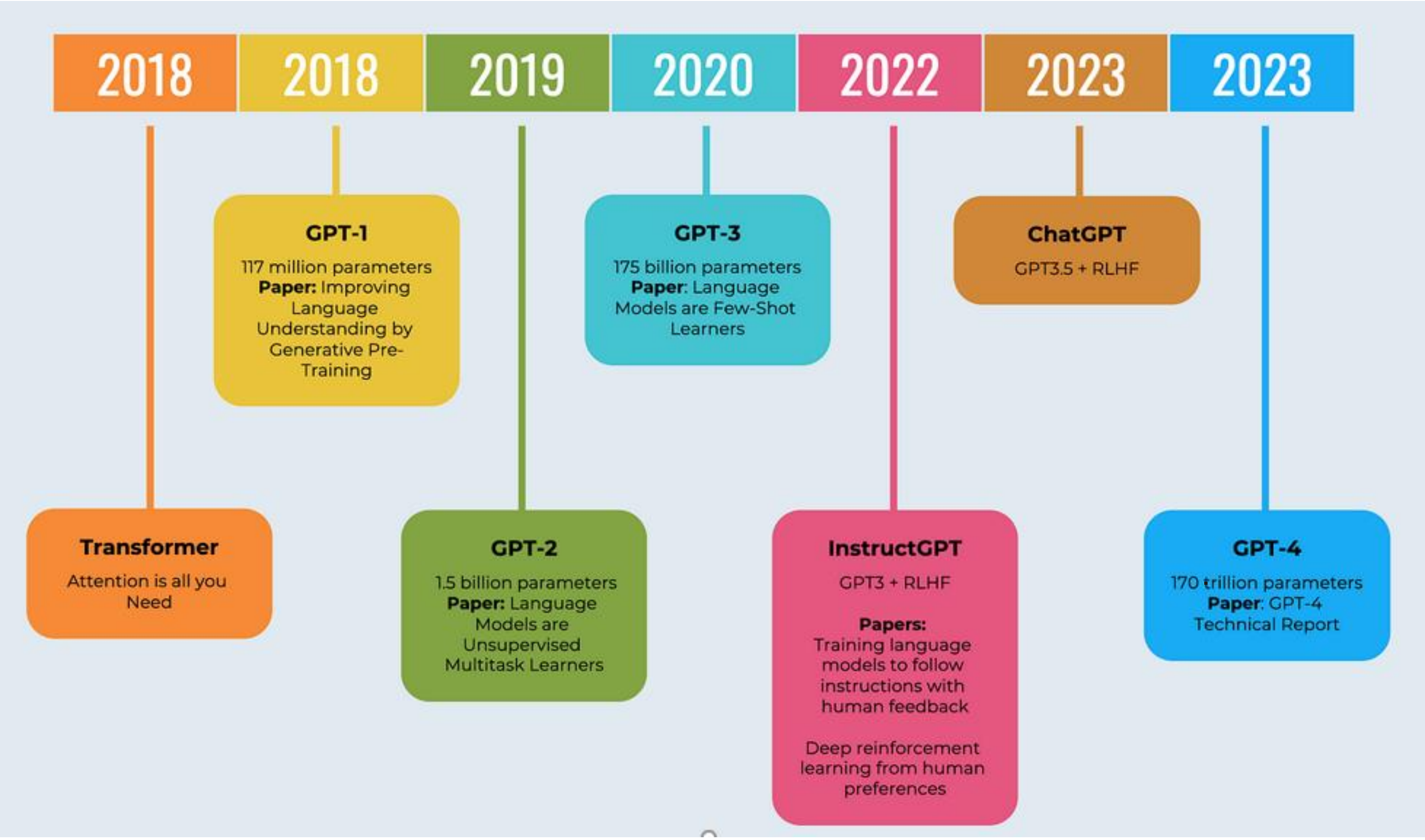
What is a language model?

GPT 4 uses ~3000-gram, i.e., would need about 10^{3000} books for training

→ **Attention**, the fundamental concept behind **Transformers**, → is the solution to the curse of dimensionality



State-of-the-art: GPT-4

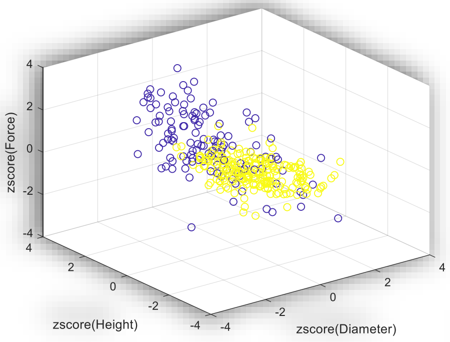


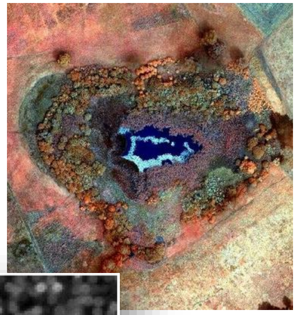
- GPT-4 has ~**1.70×10^{12} parameters**
  - ~200 parameters per human on earth
  - Humans have
    - ~86×10^{9} neurons
    - **~100×10^{12} connections**
- Human > 50GPT-4 ☺
  - GPT-5?
- Note that we might be running out of data to train AI models with …

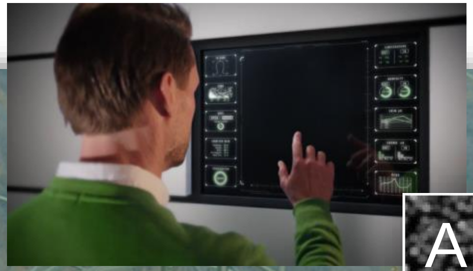
OST

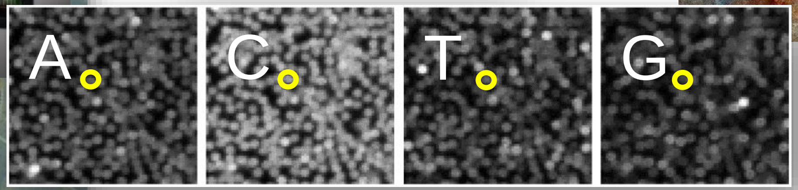
Industrial Applications of Artificial Intelligence

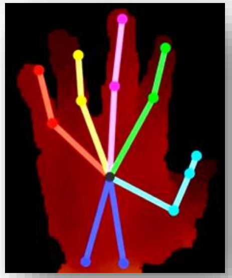
- **ICAI Research & Development**











ICAI

OST

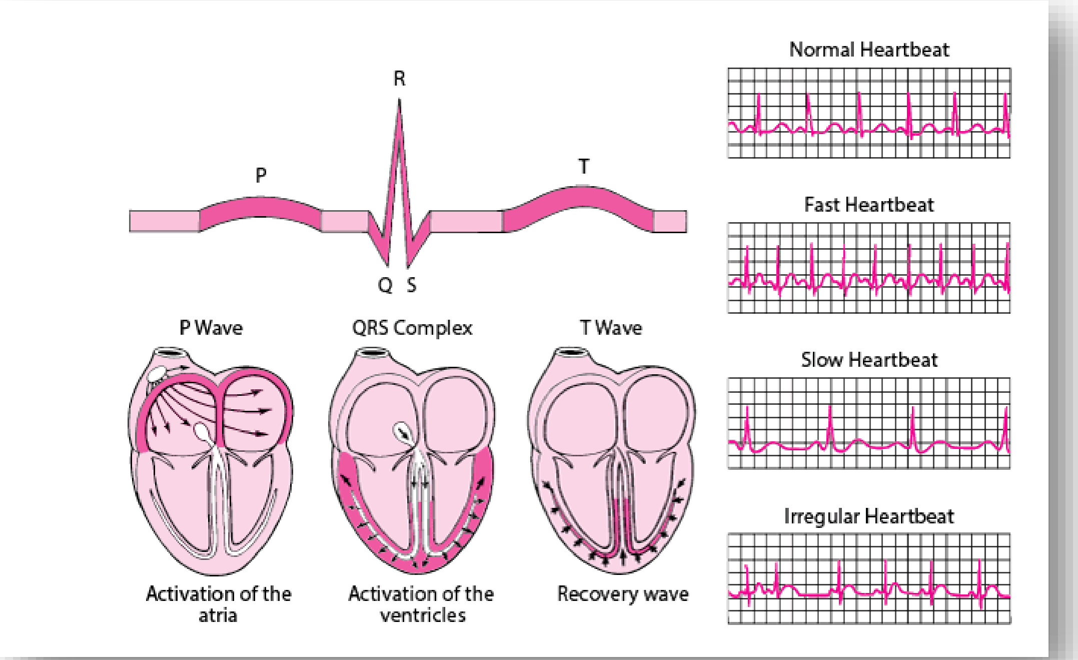
Deep Learning for ECG Analysis

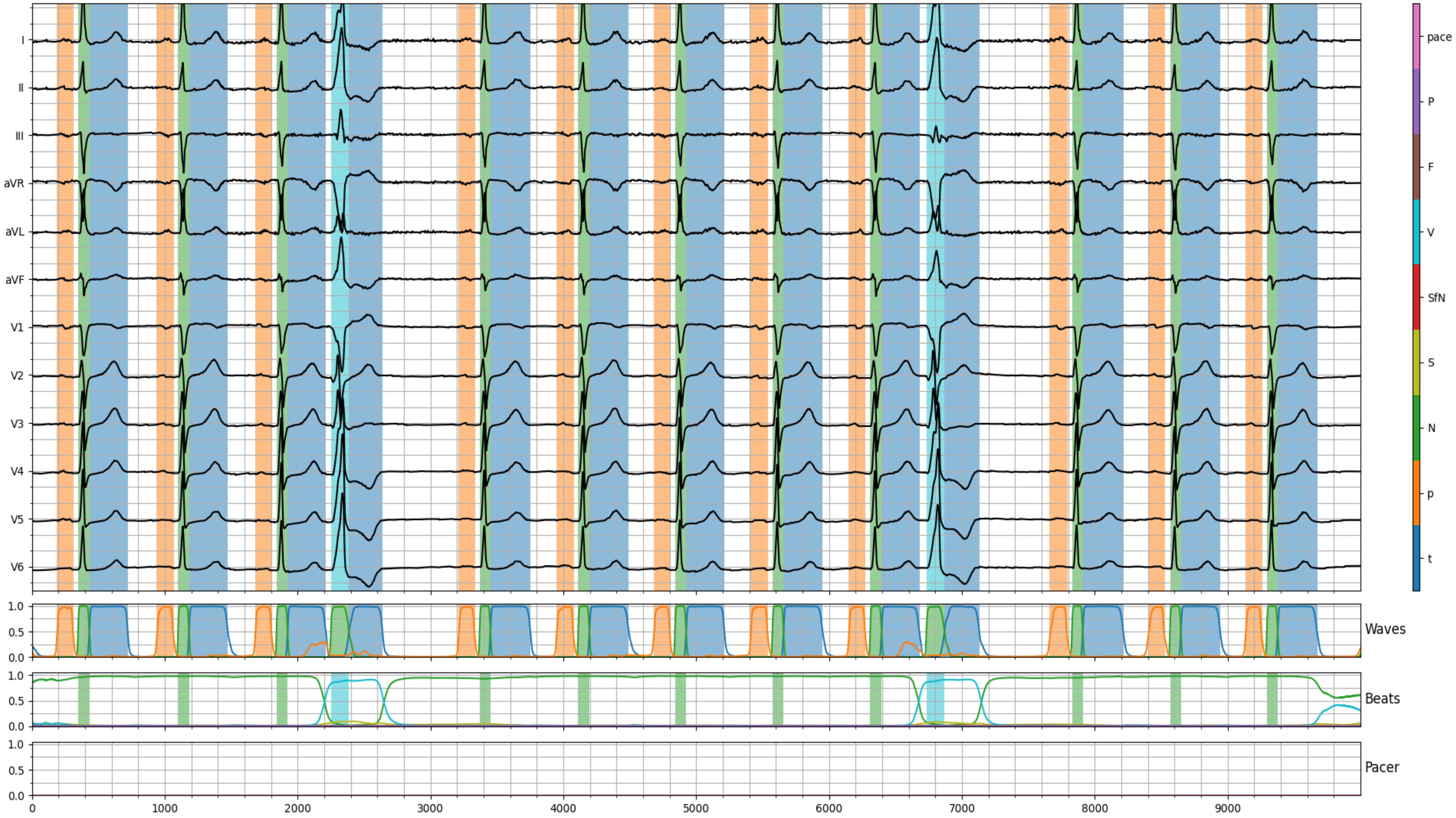
Application Number: 36433.1 IP-LS

Application Title: Data-driven Electrocardiogram Interpretation

Main partners and project manager

Project manager	Ramun Schmid SCHILLER AG
Main research partner	Professor Dr Guido Schuster HSR Hochschule Rapperswil
Research partner	Professor Dr Christian Mueller Universitätsspital Basel
Main implementation partner	Ramun Schmid SCHILLER AG





I
II
III
aVR
aVL
aVF
V1
V2
V3
V4
V5
V6
pace
P
F
V
SfN
S
N
p
t
Waves
Beats
Pacer

ML for Injection Moulding Control

iwk
INSTITUT FÜR
WERKSTOFFTECHNIK UND
KUNSTSTOFFVERARBEITUNG

SUBVENTIONSVERTRAG

Innovationsprojekt 29621.1 IP-ENG

Zwischen der **Innosuisse – Schweizerische Agentur für Innovationsförderung**
(nachstehend **Beitragsgeberin** genannt)

und den folgenden Projektpartnern:

Forschungspartner:

HSR Hochschule für Technik Rapperswil
(nachstehend **Empfänger**)

Umsetzungspartner:

Kistler Instrumente AG

Netstal-Maschinen AG

Geberit International AG

Weidmann Medical Technology AG

Krauss Maffei Schweiz AG

betreffend

Machine Learning basiertes Prozessmanagementsystem zur Optimierung des Spritzgiessprozesses

Data Driven Injection Moulding

Curdin Wick(✉), Frank Ehrig, and Guido Schuster

University of applied science Rapperswil, Rapperswil SG, Switzerland
{curdin.wick,frank.ehrig,guido.schuster}@hsr.ch

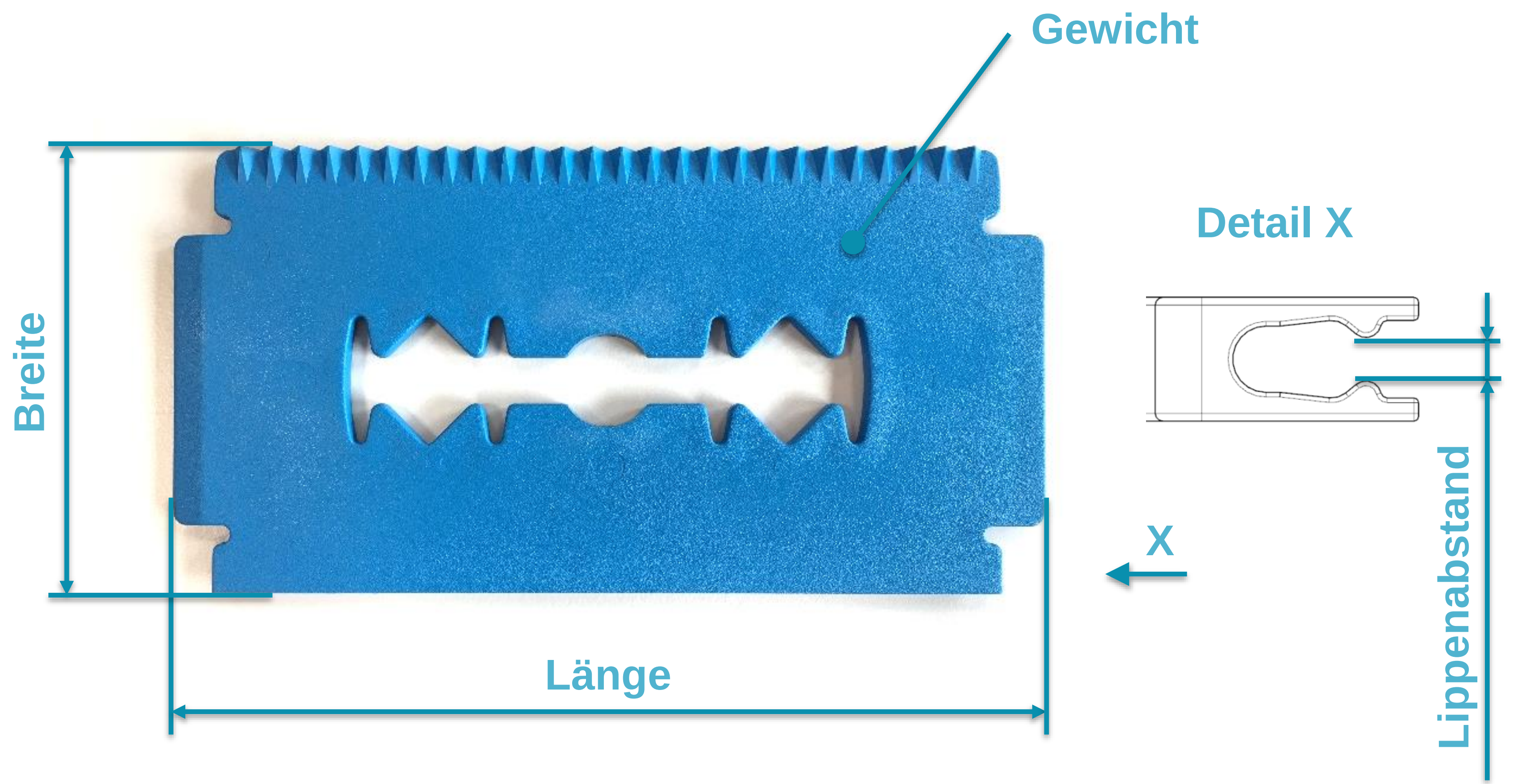
Abstract. The injection moulding process for the production of plastic parts is a very complex process. Therefore, a lot of experience and expert knowledge is necessary to produce parts with high quality. Changes in granule-batches, environmental influences and wear of the machine and the mould can strongly affect the quality of the produced parts. For this reason an injection moulding machine needs an experienced operator, who reacts properly to changing input variables and sets appropriate countermeasures. Modern injection moulding machines are able to record all countermeasures and have access to a wealth of internal machine data. Consequently, an adequate machine learning (ML) method should be able to observe, to learn the proper countermeasures and to evaluate their effectiveness. With deep learning (DL), a state of the art technology in ML, it will be possible to predictively detect process anomalies for the first time, based only on the knowledge about the internal machine data. If an operator changes the setting parameters of the injection moulding machine, the correlation between the adjustment and the anomaly is being learnt. The aim is to get process adjustment recommendations from the machine learning system.

This is a fundamentally new approach for process management in injection moulding, as the machine learning system detects problems long before they can be seen by an operator. Furthermore, the system provides process adjustment recommendations, based on the supervised and automatically generalized actions from different operators using different injection moulding machines, moulds and materials.

Keywords: Injection moulding · Machine learning · Process anomalies

OST

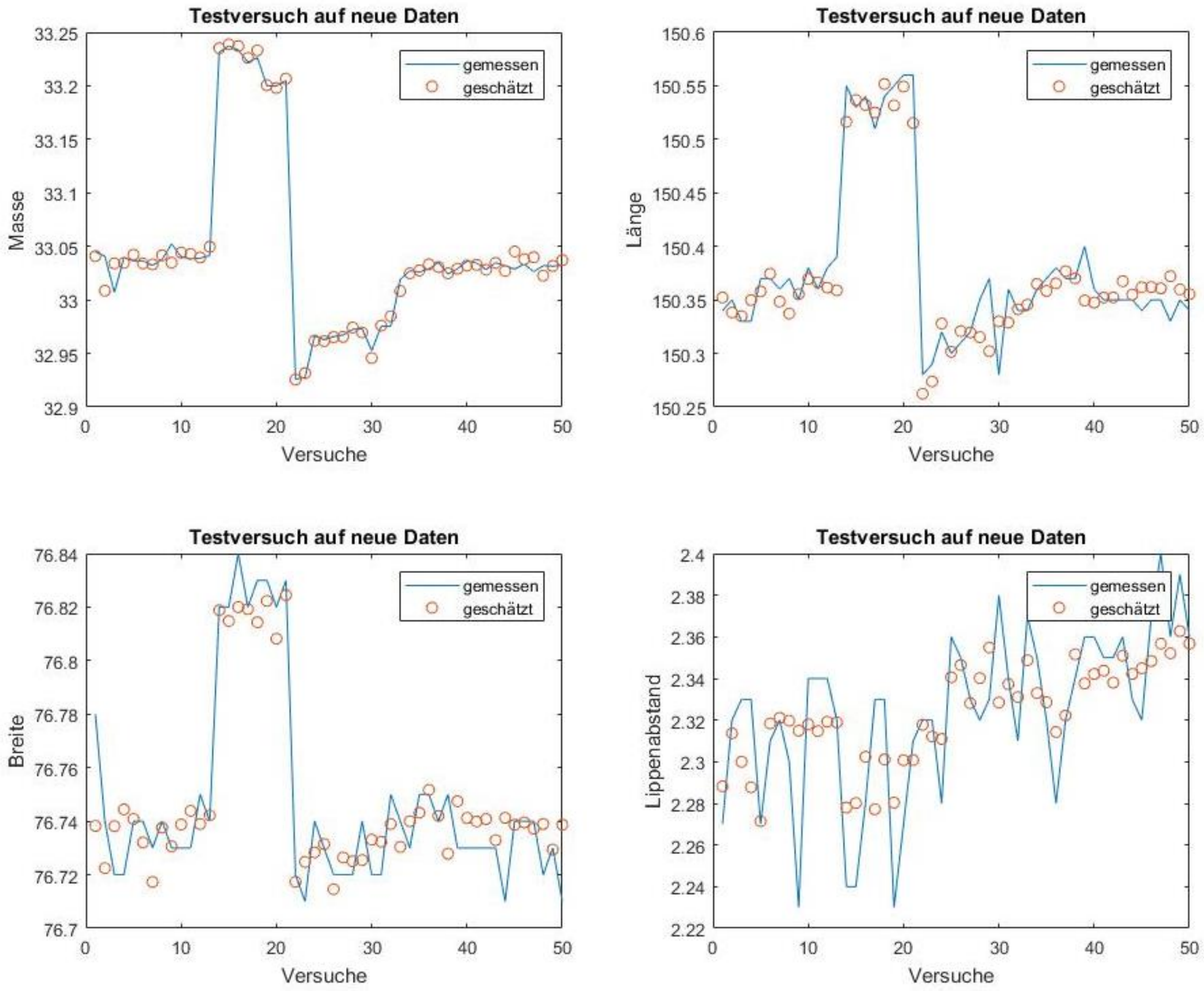
ML for Injection Moulding Control



ML for Injection Moulding Control

- Internal signals were used to train ML model to predict quality data
- Trained model was able to predict the mass, the length and the width with surprising precision

	STD-Error σ	$CV = \frac{\sigma}{\mu}$ in %
Masse	0.009 g	0.03%
Länge	0.017 mm	0.02%
Breite	0.014 mm	0.02%
Lippen-abstand	0.032 mm	1.38%



VR Helicopter Simulator

- **Loft Dynamics**
  - More than 2/3 of the engineering team was educated at the ICAI
  - CTO former ICAI engineer

Loft Dynamics

Erster Virtual-Reality-Simulator EASA-qualifiziert

Das aus dem Labor des Interdiciplinary Center for Artificial Intelligence (ICAI) hervorgegangene Unternehmen VRM Switzerland hat den ersten Helikopter-Flugsimulator entwickelt, der vollständig auf Virtual Reality (VR) basiert und von der Europäischen Agentur für Flugsicherheit (EASA) anerkannt ist.

Application Number: 38437.1 IP-ICT

Application Title: VR motion helicopter hoist operation simulator

Main partners and project manager

Project manager	Fabian Riesen VRMotion AG
Main research partner	Professor Dr Guido Schuster HSR Hochschule Rapperswil
Main implementation partner	Fabian Riesen VRMotion AG

AI based Condition Monitoring using Drones

Application Number: 52765.1 IP-EE

Application Title: CURO: Automatische Zustandsüberwachung und vorausschauende Wartung für Hochspannungsleitungen mittels automatisierten Drohnen und künstlicher Intelligenz

Axpo übernimmt Spezialsoftware-Firma LINIA

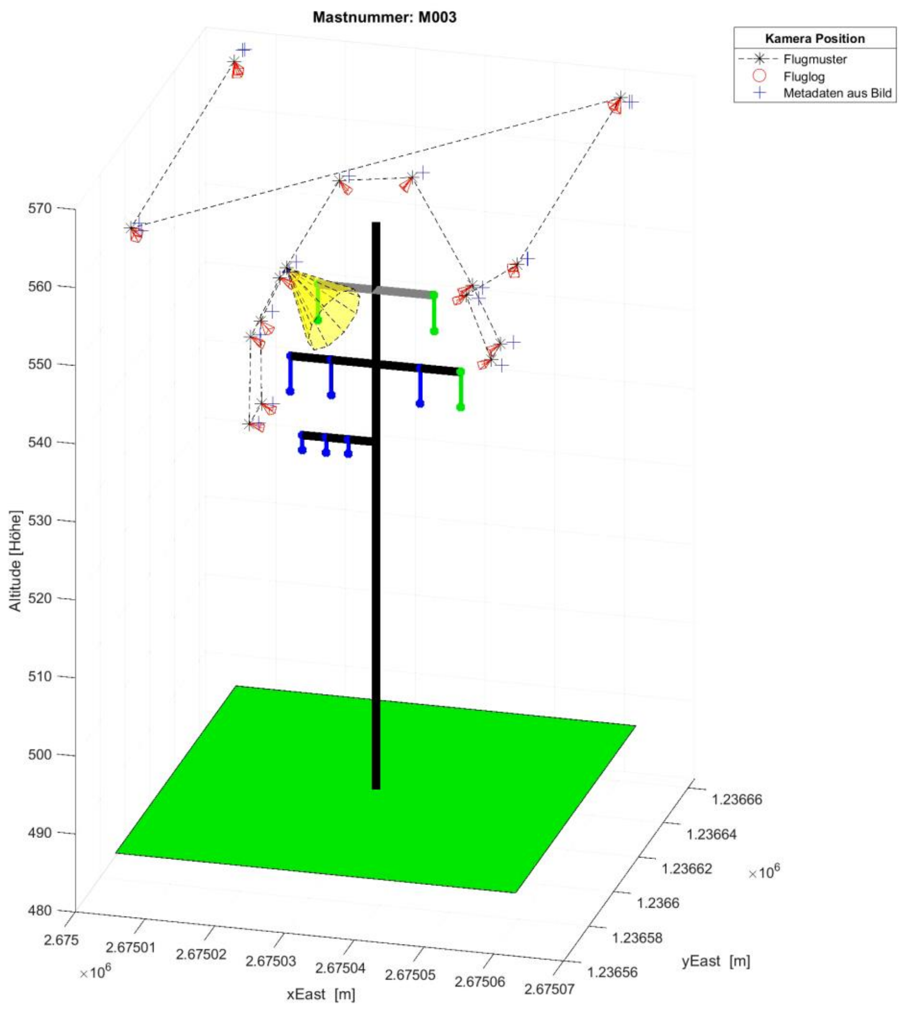
Share | Netze | Unternehmen

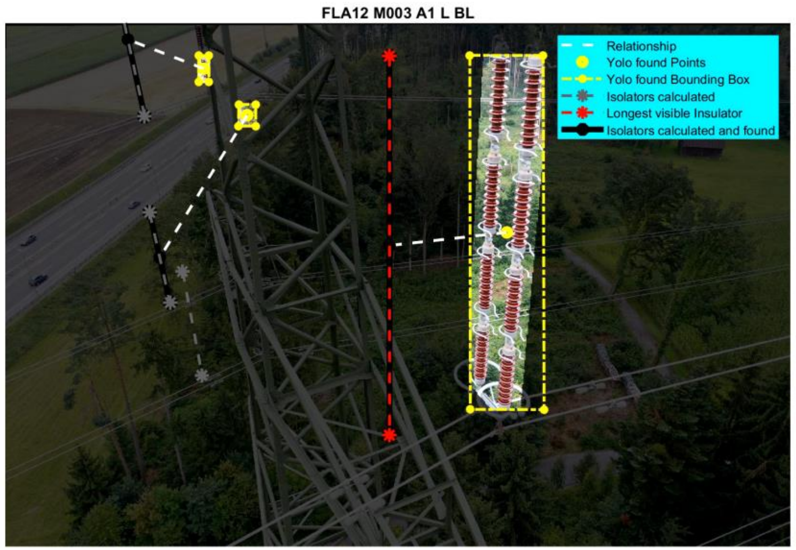
08.04.2024 - Per 1. April 2024 hat Axpo die Software-Firma LINIA übernommen. Die Übernahme ermöglicht Axpo ein einzigartiges Full-Service-Angebot im Dienstleistungsbereich der automatisierten Stromnetzinspektion und den Zugang zu neuen Märkten.

Main partners and project manager

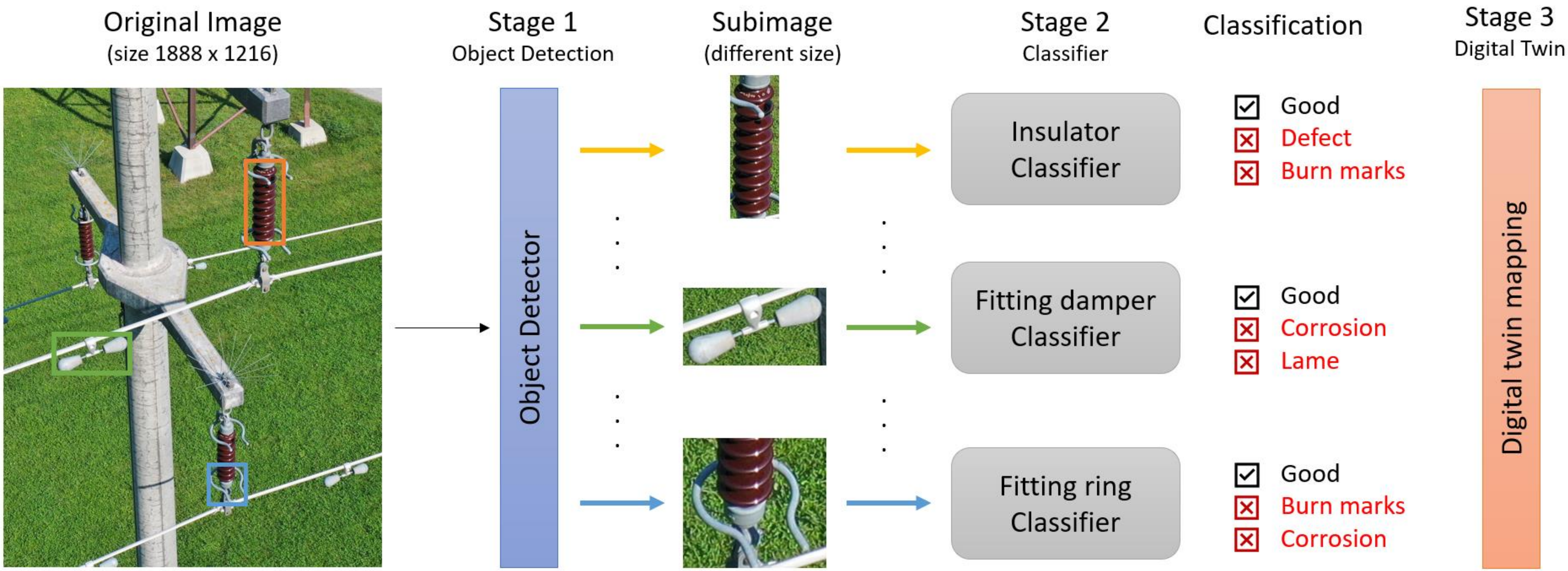
Project manager	Lorenzo Arizzoli-Bulato LINIA GmbH
Main research partner	Professor Dr Guido Schuster OST - Ostschweizer Fachhochschule
Main implementation partner	Lorenzo Arizzoli-Bulato LINIA GmbH

AI based Condition Monitoring using Drones





AI based Condition Monitoring using Drones



Quo Vadis?


Brin in 2010


Elon Musk (2022)


Tim Cook (2023)

Gary Marcus • 2nd
Scientist, author (5 books, including Rebooting AI (Forbes 7 Must Read Book...
7h •

+ Follow

Hype to reality, in a dozen years or less:
2012: Sergey Brin promises driverless cars by 2017
2014-: Elon offers endless promises
2016: Gary Marcus and @filippie509 say its not gonna work anytime soon
2016-: Tech bros say we just need more data
2017-: Tech bros say synthetic data will fix it
2018-: This new FSD release is going fix it (rinse and repeat)
2018-2022: Investors pour in $100B
2022-3: Multiple driverless car companies fail
2023: Cruise implodes; massive remote ops revealed
2024: Apple shuts down driverless car efforts

Apple Cancels Work on Electric Car, Ending Decadelong Effort
bloomberg.com • 1 min read

Thousands of AI Authors on the Future of AI

Preprint

Katja Grace[*†]
AI Impacts
Berkeley, California
United States
katja@aiimpacts.org

Harlan Stewart[†]
AI Impacts
Berkeley, California
United States

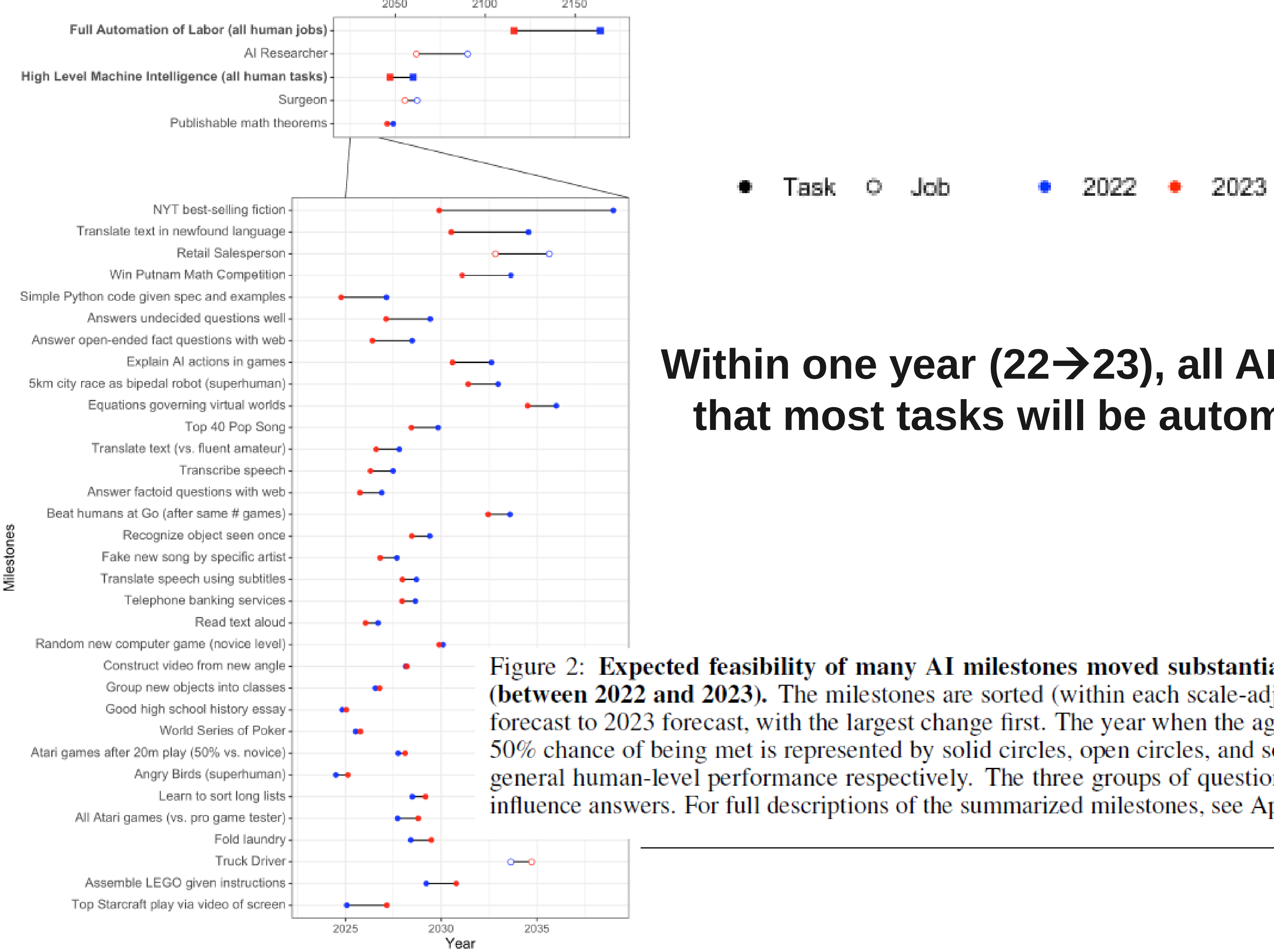
Julia Fabienne Sandkühler[†]
Department of Psychology
University of Bonn
Germany

Stephen Thomas[†]
AI Impacts
Berkeley, California
United States

Ben Weinstein-Raun
Independent
Berkeley, California
United States

Jan Brauner
Department of Computer Science
University of Oxford
United Kingdom

January 2024



Within one year (22→23), all AI experts now think, that most tasks will be automated by AI **earlier!**

Figure 2: **Expected feasibility of many AI milestones moved substantially earlier in the course of one year (between 2022 and 2023).** The milestones are sorted (within each scale-adjusted chart) by size of drop from 2022 forecast to 2023 forecast, with the largest change first. The year when the aggregate distribution gives a milestone a 50% chance of being met is represented by solid circles, open circles, and solid squares for tasks, occupations, and general human-level performance respectively. The three groups of questions have different formats that may also influence answers. For full descriptions of the summarized milestones, see Appendix C.

OST

Robots (manual labor) versus AI (intellectual labor)

AI is the <u>software brain</u>
of the <u>hardware robot</u>

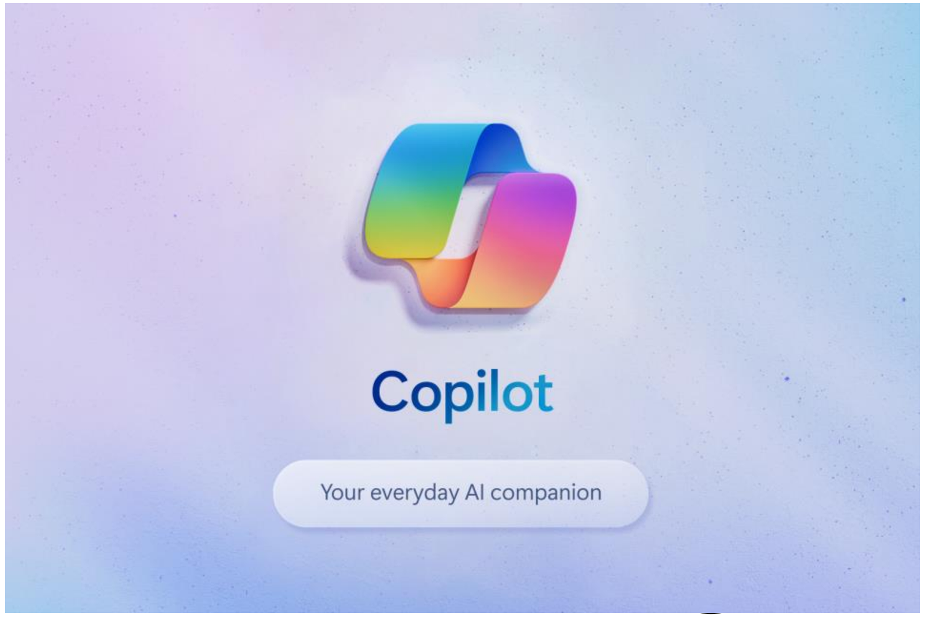
Robots
- hardware
- slow to evolve
- expensive
- local
- do not scale

Brain

AI
- software
- fast to evolve
- cheap
- global
- scales well

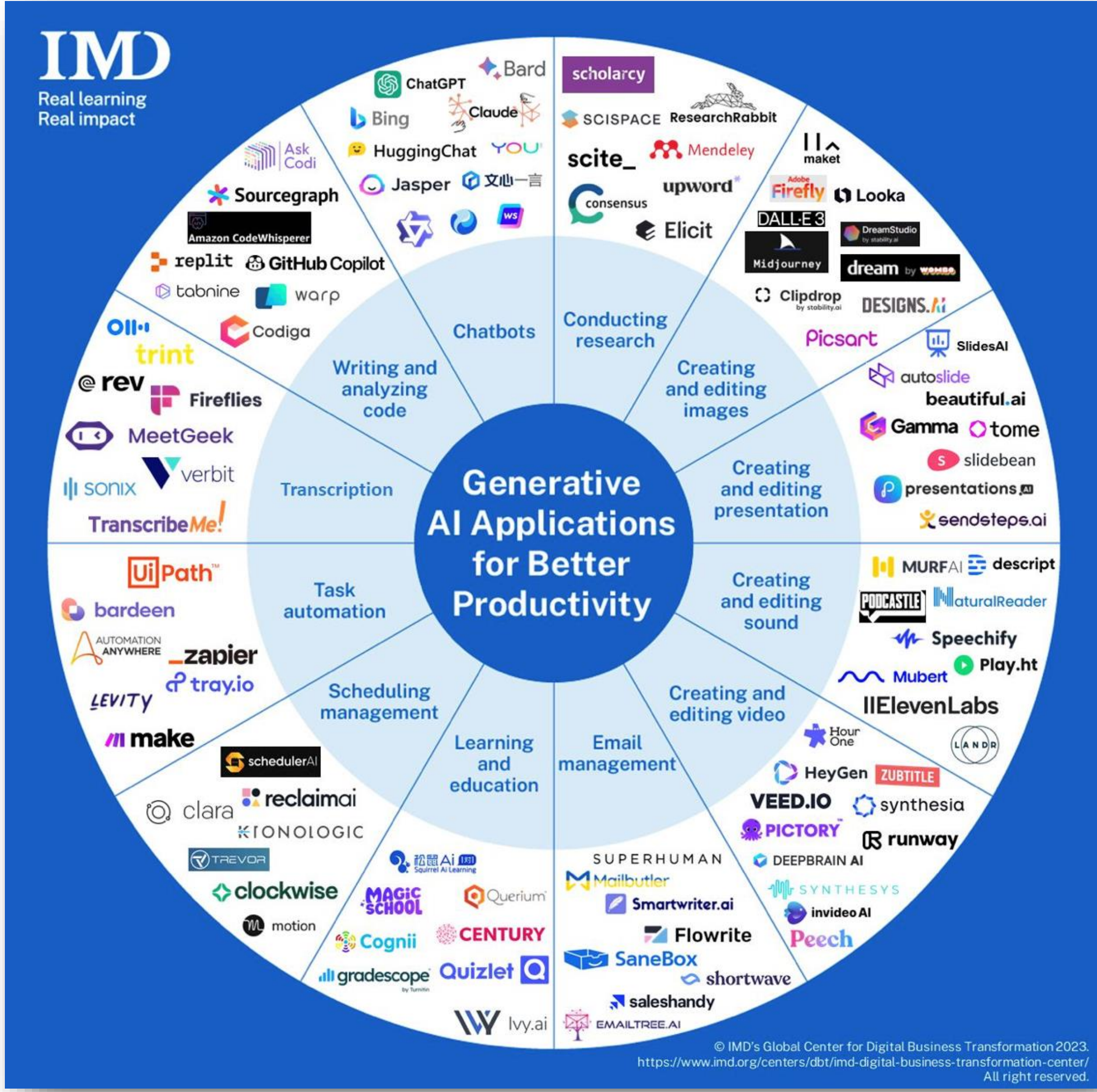
Opportunities

- If you provide goods & services which require **manual labor**
  - **For example, you are a carpenter**
  - Robots will not replace you any time soon
    - Robots are power hungry, expensive, fragile and not very flexible
  - AI will help you to automate your administrative tasks
    - More time for the things you love
- Use the AI tools that will be offered in Microsoft 365
  - Since you are using these tools anyway,
    learn to use them with the new Microsoft Copilot



Threats

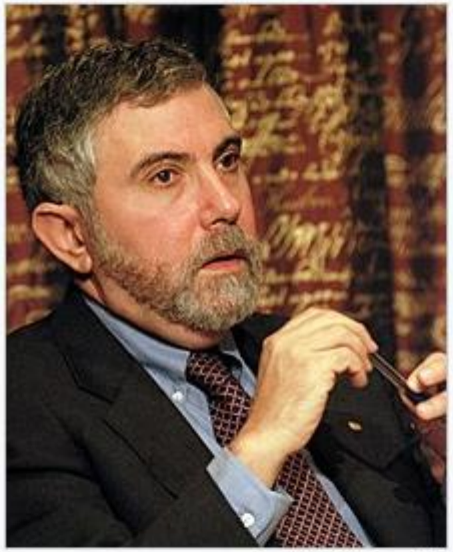
- If you provide services which are **digital**
  - **For example, you are a graphic artist**
  - Think of your work as a collection of task
    - AI will help you to be much more efficient at many tasks
      - Creating/Editing a new text, image, song, video
      - Designing a marketing campaign, …
      - People will become significantly more productive
    - **Higher productivity means less people for the same amount of work and/or better/more work with the same number of people**
      - AI will not replace people,
        people using AI will replace people which do not use AI
  - There are still many **interpersonal** task, where AI will have no impact – make sure you are good at them!
    - Talking to your customers and collaborators
    - Negotiating a good deal
    - Creating strong relationships based on trust



OST

Questions?

**"Productivity is not everything,
but in the long run,
it's almost everything"**



Paul Krugman (2008)

Nobel Memorial Prize in Economic Sciences [edit]

Krugman was awarded the Nobel Memorial Prize in Economic Sciences (informally the Nobel Prize in Economics), the sole recipient for 2008. This prize includes an award of about $1.4 million and was given to Krugman for his work associated with New Trade Theory and the New Economic Geography.[90] In the words of the prize committee, "By having integrated economies of scale into explicit general equilibrium models, Paul Krugman has deepened our understanding of the determinants of trade and the location of economic activity."[91]

Übersicht

- Abschluss: **Certificate of Advanced Studies in Artificial Intelligence**
- Schwerpunkte: AI-Grundlagen, Business Anwendungen, Eigener Case
- Nutzen: **Sie verstehen die technischen und wirtschaftlichen AI-Grundlagen und erwerben die Kompetenzen, eine AI-Führungsrolle zu übernehmen**
- Dauer: 12 Präsenztage
- Kosten: 9900 CHF → inklusiv Unterrichtsmaterial (Bücher & PPT)
- Zulassung: Anerkannter Tertiärabschluss, mehrjährige qualifizierte Berufserfahrung, Tätigkeit in einem entsprechenden Arbeitsfeld, gute Englischkenntnisse, «sur dossier»
- Ort: Campus Rapperswil-Jona
- Durchführung: **Herbst 2025**

ICAI