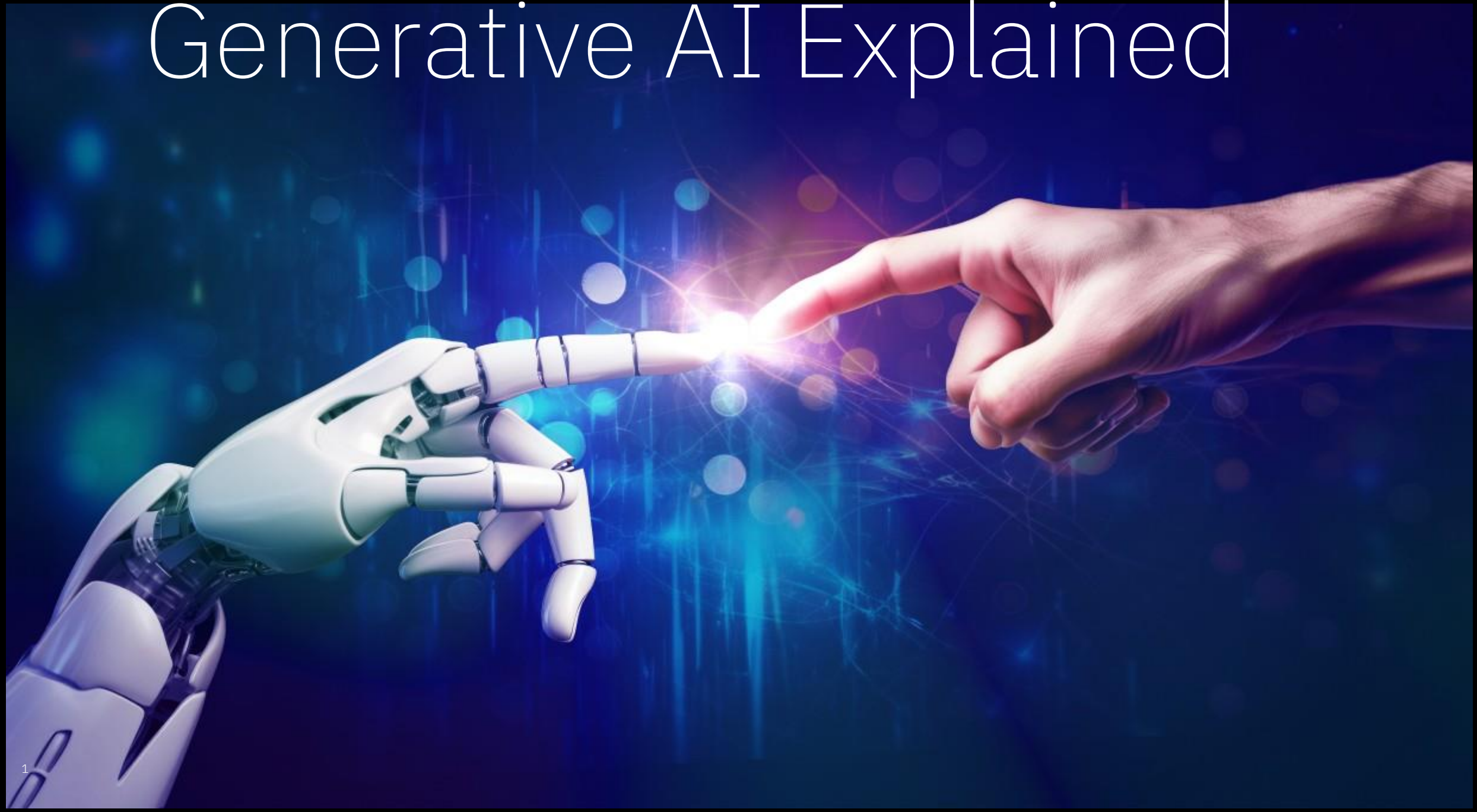
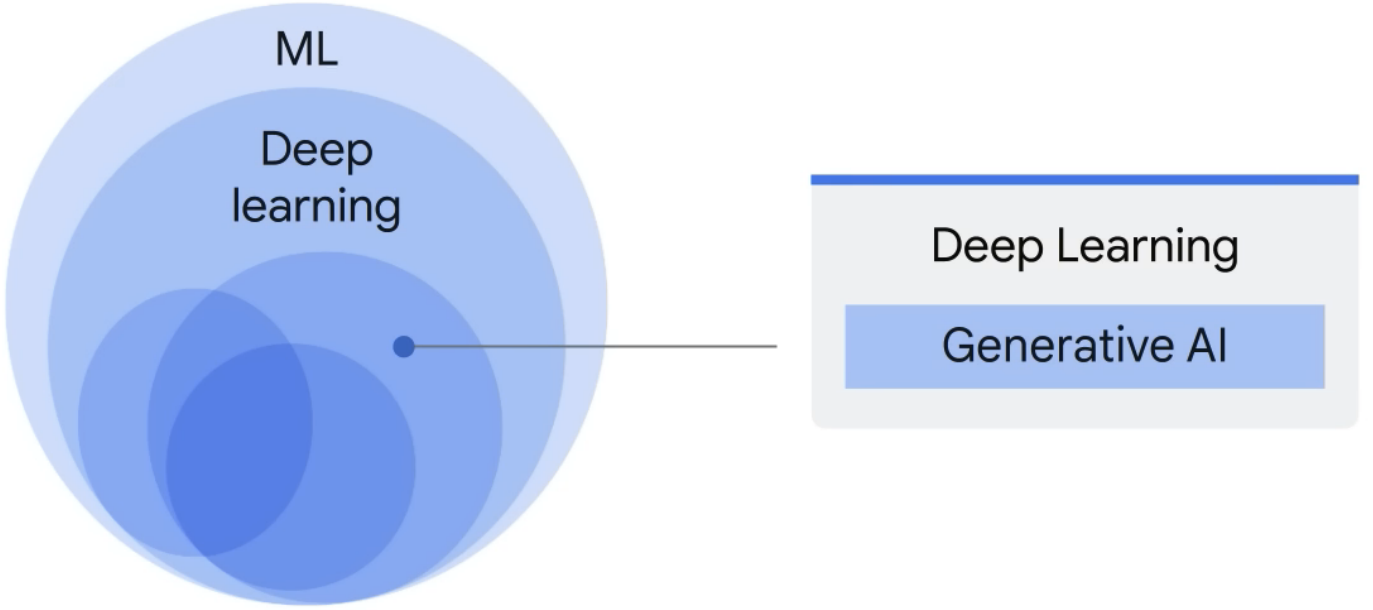
Generative AI Explained



1. **Overview of Gen AI**
2. Attention is All you Need
3. It’s About Cosine Similarity
4. What’s Next for Gen AI
5. Is AI going to Replace Me?
6. Resources

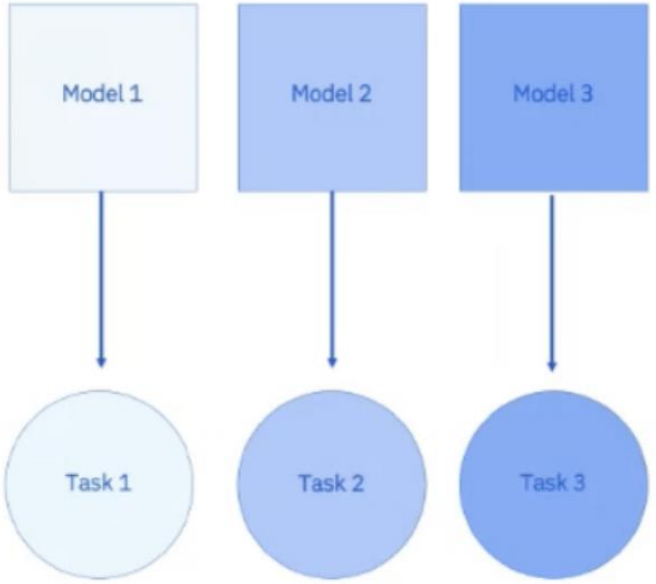
Generative AI

Generative AI is a subset of Deep Learning



Traditional AI Models

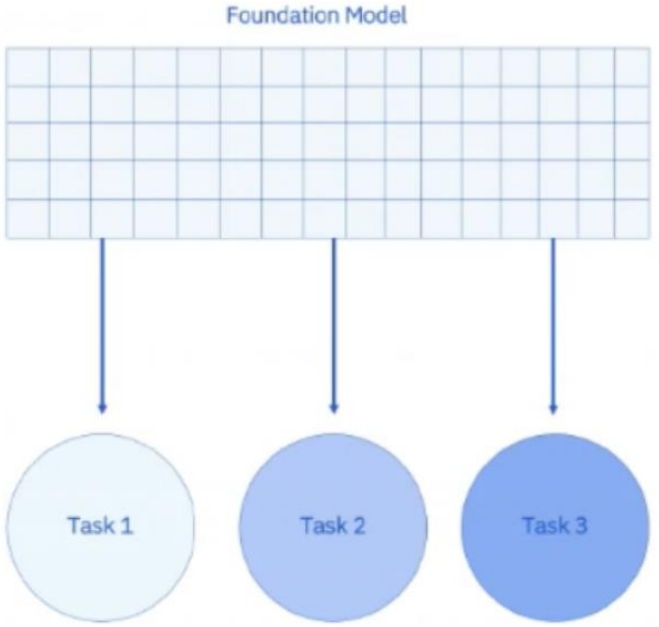
Each model is trained for a specific task



1,000s to 1,000,000s labeled data points per task

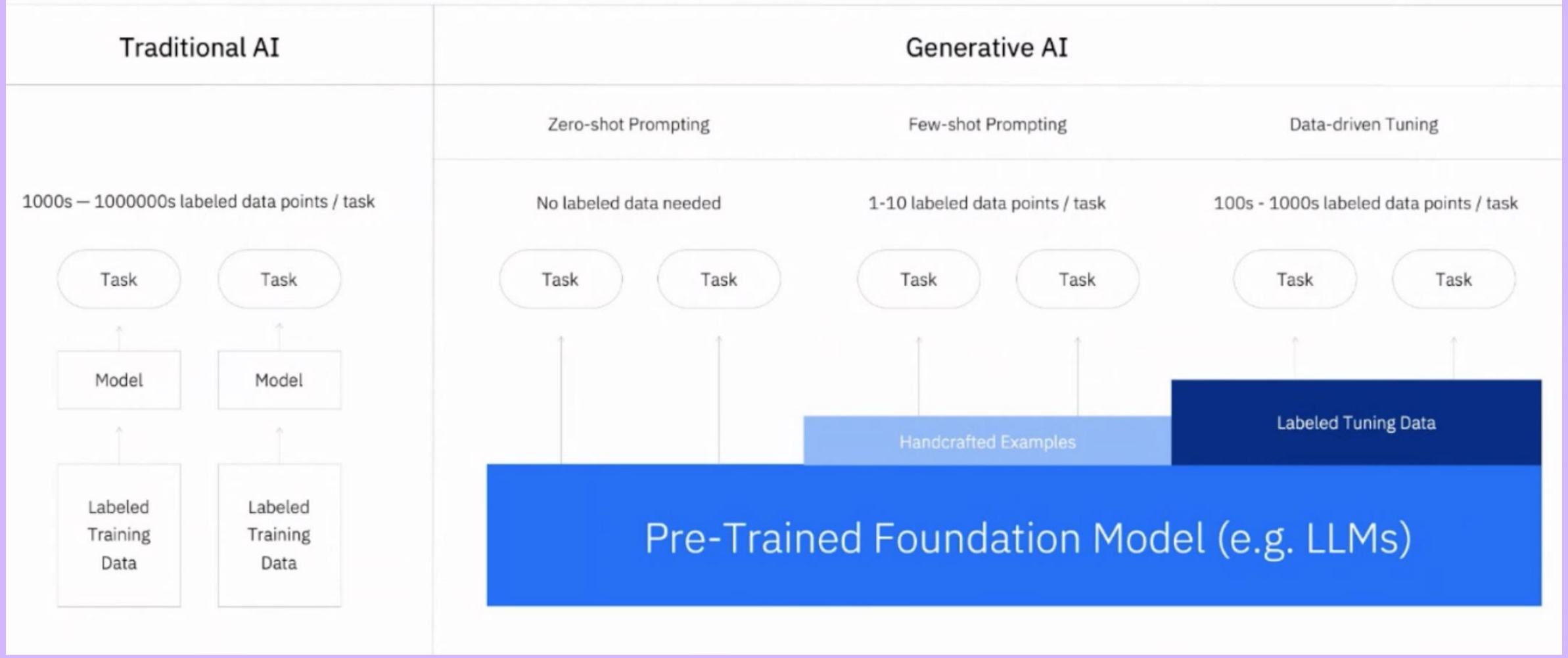
Foundation Models

One model that can address many tasks



0 to 1,000s labeled data points per task

A new paradigm for data-efficient development



… an inflection point in AI

1980s | Big data

1980s to ~2010 | Large labeled data + Compute

2017+ | **Self-supervision at scale** + Massive unlabeled data + Compute

Expert Systems

Hand-crafted symbolic representations

Words: symbolic representation
Algorithms: rule-based/ML

Machine Learning

Task-specific hand-crafted feature representations

Words: symbolic representation
Algorithms: machine learning

Deep Learning

Task-specific learned feature representations

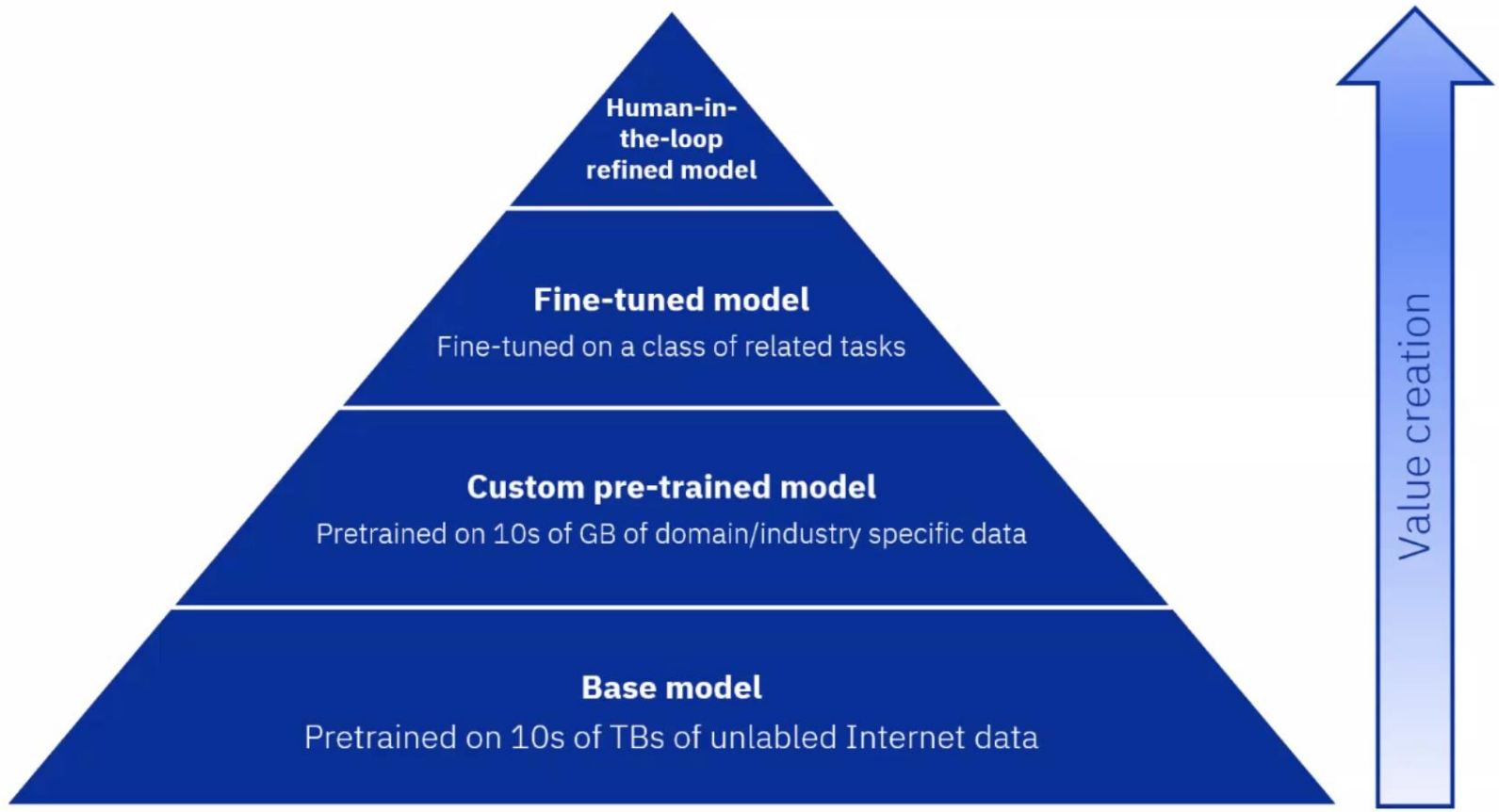
Words: dense vectors
Algorithms: neural networks

Foundation Models

Generalizable & adaptable learned representations

Words: dense vectors
Algorithms: transformer-based NN

Foundation model customization



What are people trying to do with Generative AI?

Most relate to one of these AI Tasks...

1 **Retrieval-Augmented Generation (RAG)**
Based on a set of documents or dynamic content, create a chatbot or a question-answering feature grounded on specific content. E.g., building a Q&A resource from a broad knowledge base, providing customer service assistance

2 **Summarization**
Transform text with domain-specific content into personalized overviews, capturing key points.
E.g., sales conversation summaries, insurance coverage, meeting transcripts, and contract information

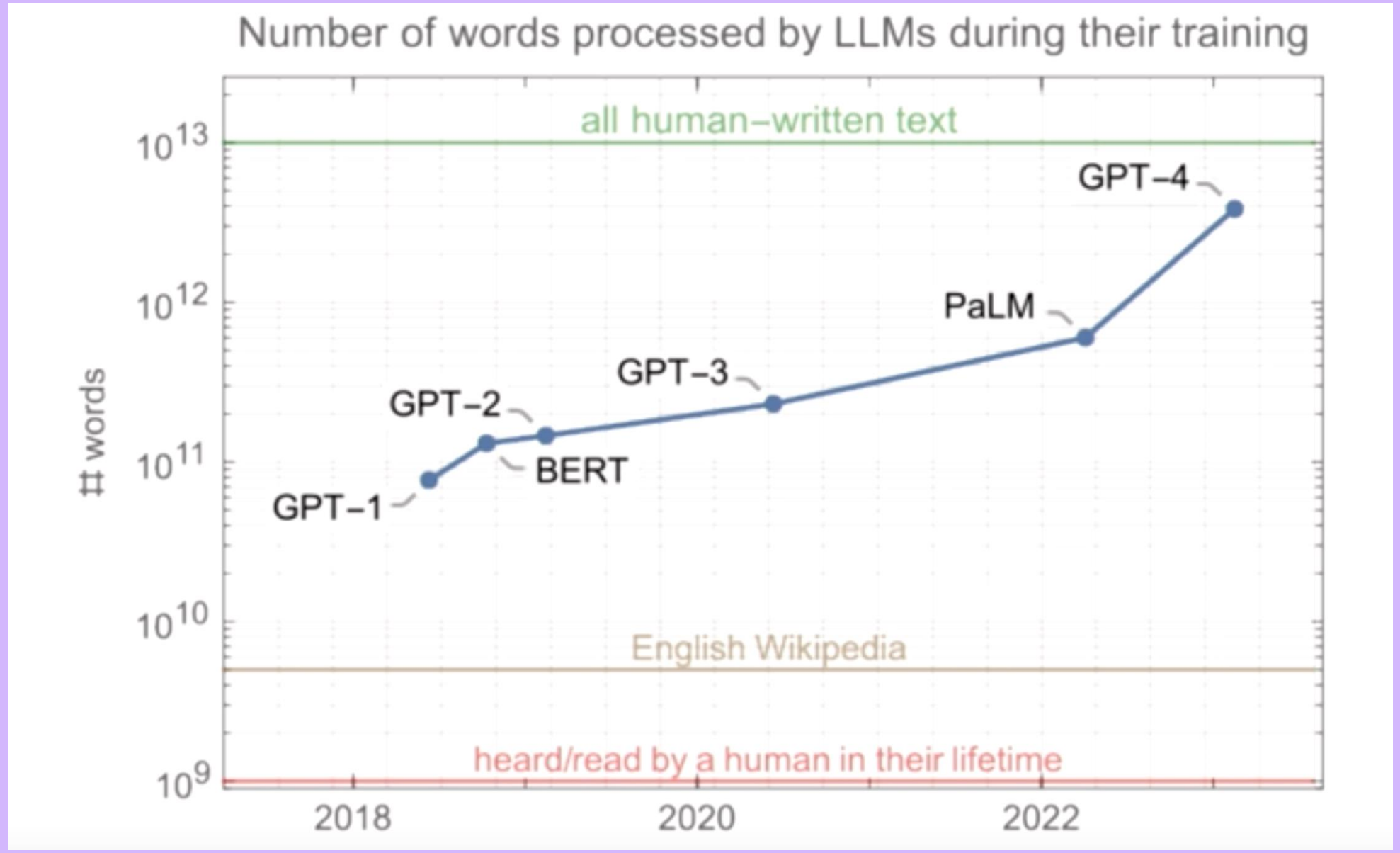
3 **Content Generation**
Generate content for a specific purpose.
E.g., content creation for marketing campaigns, job descriptions, blog posts and articles, email drafting support, and code generation

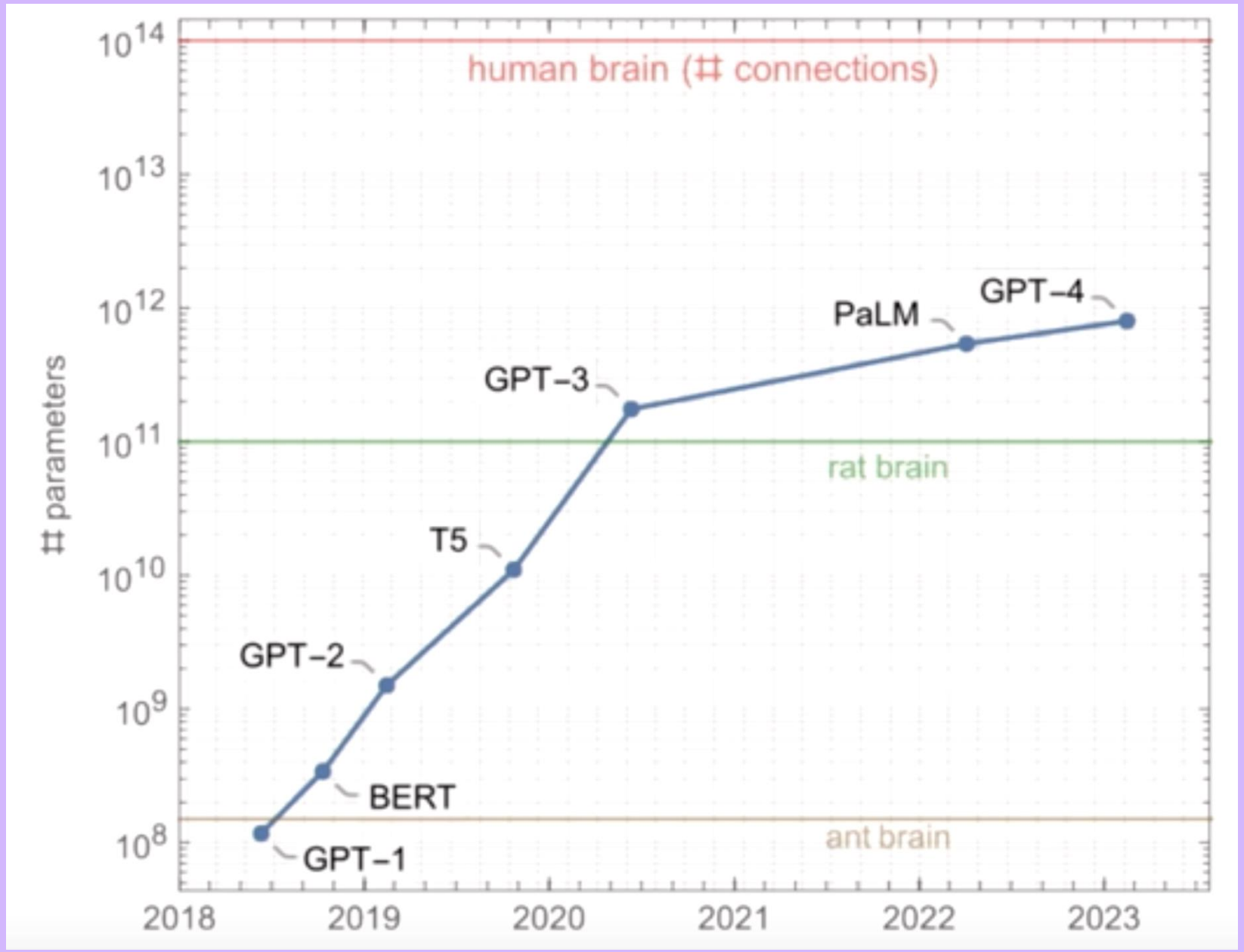
4 **Named Entity Recognition**
Identify and extract essential information from unstructured text.
E.g., audit acceleration, SEC 10K fact extraction

5 **Insight Extraction**
Analyze existing unstructured text content to surface insights in specialized domain areas.
E.g., medical diagnosis support, user research findings

6 **Classification**
Read and classify written input with as few as zero examples.
E.g., sorting of customer complaints, threat & vulnerability classification, sentiment analysis, and customer segmentation

Model	Organization(s)	Description
GPT (General Purpose Transformer)	OpenAI	GPT refers to a class of models including GPT-2, GPT-3, and GPT-4.
ChatGPT	OpenAI	A conversational system built on top of GPT-3 and fine-tuned using both supervised and reinforcement learning techniques.
Codex	GitHub	A transformer-based large language model that generates source code from natural language. It was produced by fine-tuning GPT-3 on a corpus of source code data from GitHub.
LLaMA	Meta	An auto-regressive language model based on the transformer architecture that comes in different sizes: 7B, 13B, 33B and 65B parameters. LLaMA was released under a non-commercial license.
BERT (Bidirectional Encoder Representations from Transformers)	Google	A transformer-based machine learning technique for natural language processing (NLP), pre-trained on language modeling and sentence prediction tasks.
RoBERTa	University of Washington & Meta	An extension of BERT that was trained on a larger data set with longer sentences, and removed the sentence prediction task.
T5 (Text to Text Transfer Transformer)	Google	A transformer-based architecture that uses a text-to-text approach for tasks including translation, question answering, and classification.
BART	Meta	A transformer-based technique that is considered to be a generalization of GPT and BERT.
OPT (Open Pre-trained Transformer)	Meta	A language model with 175 billion parameters trained on publicly available data sets.
BLOOM (BigScience Large Open-science Open-access Multilingual Language Model)	BigScience	A transformer-based language model trained on around 176 billion parameters.
CLIP (Contrastive Language-Image Pre-training)	OpenAI	A technique that connects text and images by pre-training an image encoder and a text encoder to predict which images were paired with which texts in the training set.
DALL-E & DALL-E 2	OpenAI	Diffusion models that generate digital images from text descriptions, trained on a dataset of text-image pairs.
Stable Diffusion	Stability AI, CompVis LMU, and Runway	A diffusion model that can generate images from natural language prompts.
Midjourney	Midjourney	A model that can generate images from natural language prompts. It's speculated that the underlying technology is based on Stable Diffusion.


Number of words processed by LLMs during their training
all human−written text
English Wikipedia
heard/read by a human in their lifetime
GPT−1
GPT−2
BERT
GPT−3
PaLM
GPT−4
words
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
2018
2020
2022


human brain (# connections)
rat brain
ant brain
parameters
10^{14}
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
10^{8}
GPT-1
BERT
GPT-2
T5
GPT-3
PaLM
GPT-4
2018
2019
2020
2021
2022
2023


LLM training prices (at the time of their creation)
Price, US$
10^8
10^7
10^6
10^5
10^4
1000
100
2018
2020
2022
GPT–4
PaLM
GPT–3
GPT–2
BERT
GPT–1

1. Overview of Gen AI
2. **Attention is All you Need**
3. It’s About Cosine Similarity
4. What’s Next for Gen AI
5. Is AI going to Replace Me?
6. Resources

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

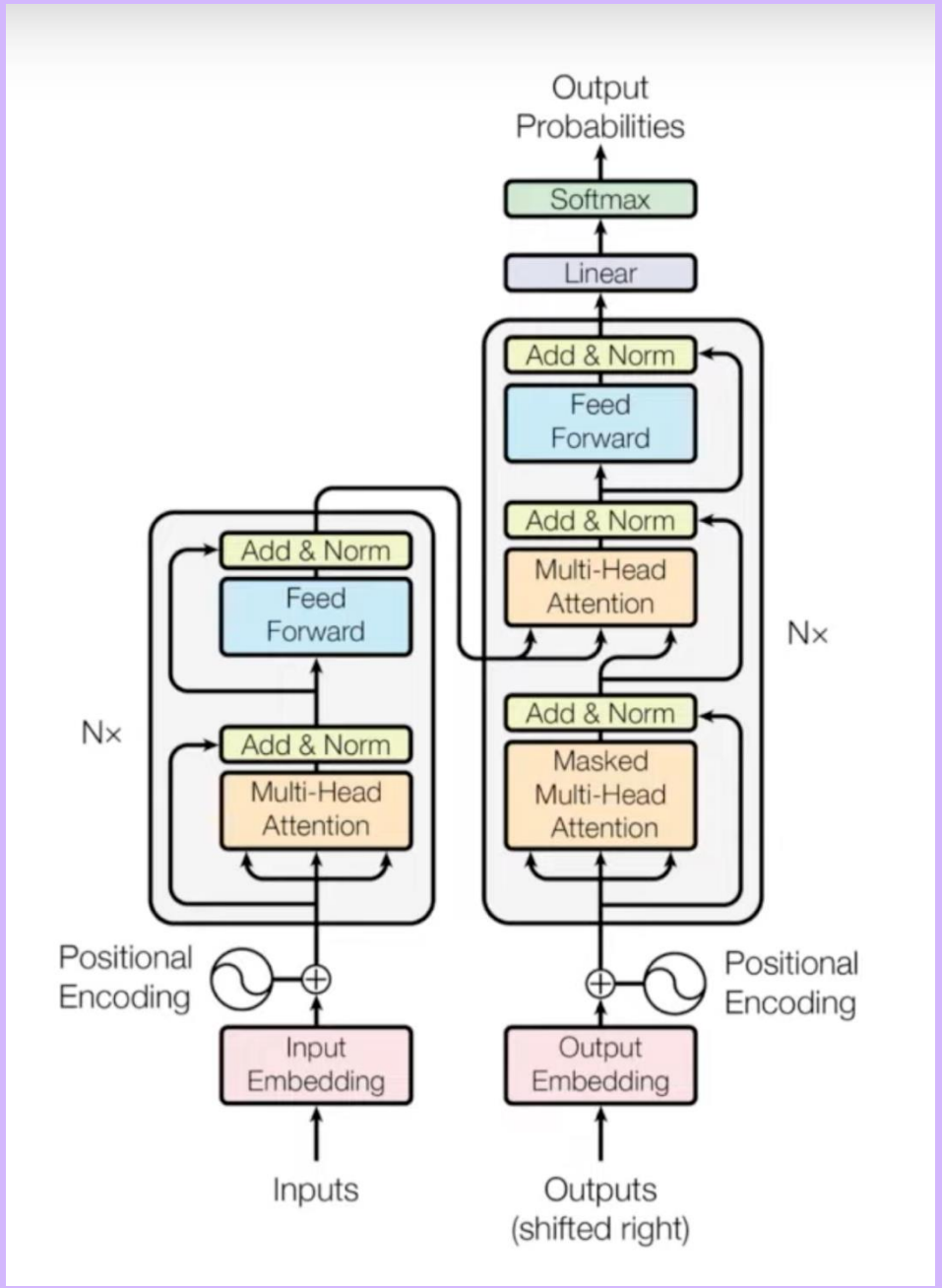
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Transformer architecture with:

Encoder and Decoder

uses:

Attention Mechanism

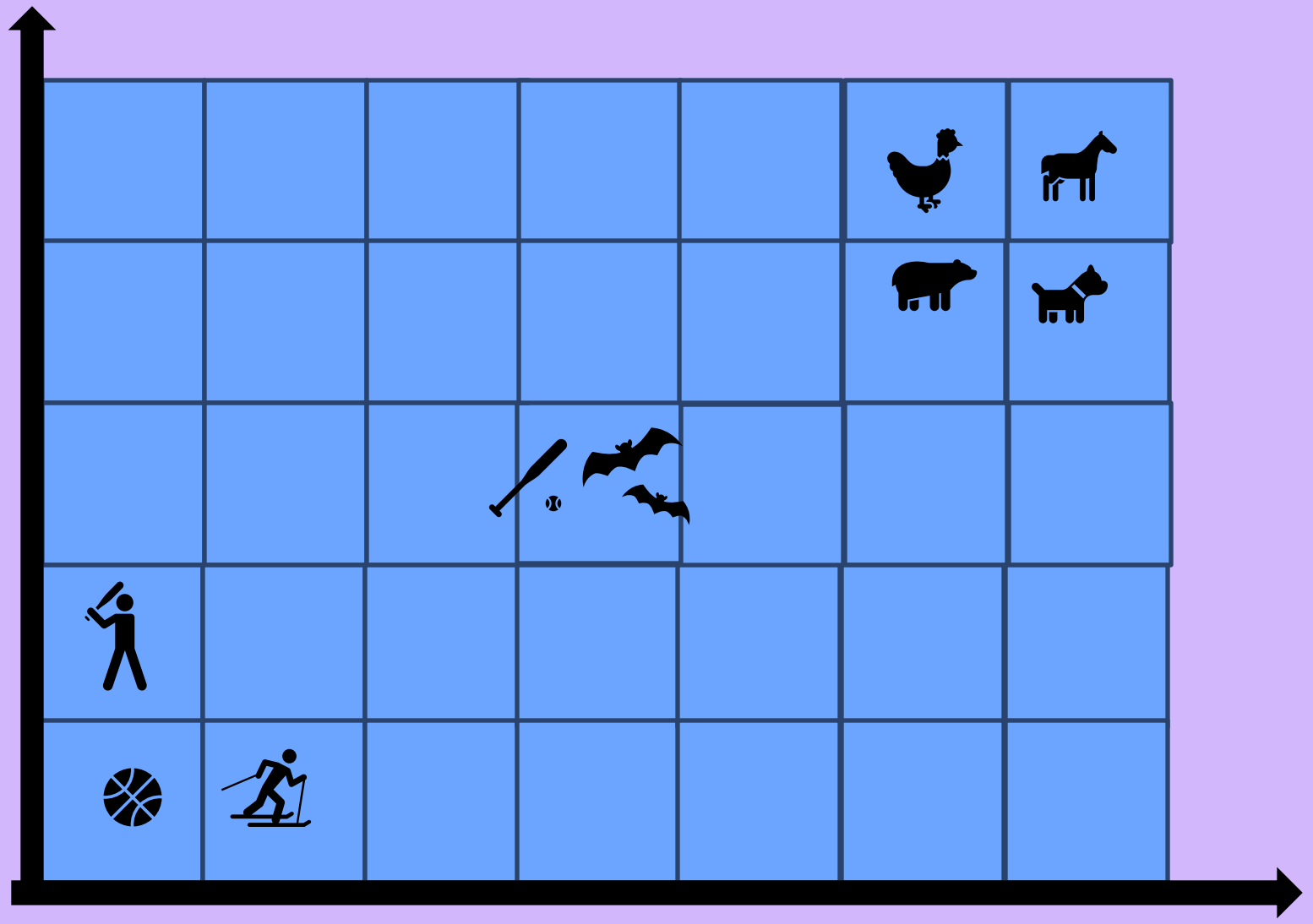


1. Overview of Gen AI
2. Attention is All you Need
3. **It’s About Cosine Similarity**
4. What’s Next for Gen AI
5. Is AI going to Replace Me?
6. Resources

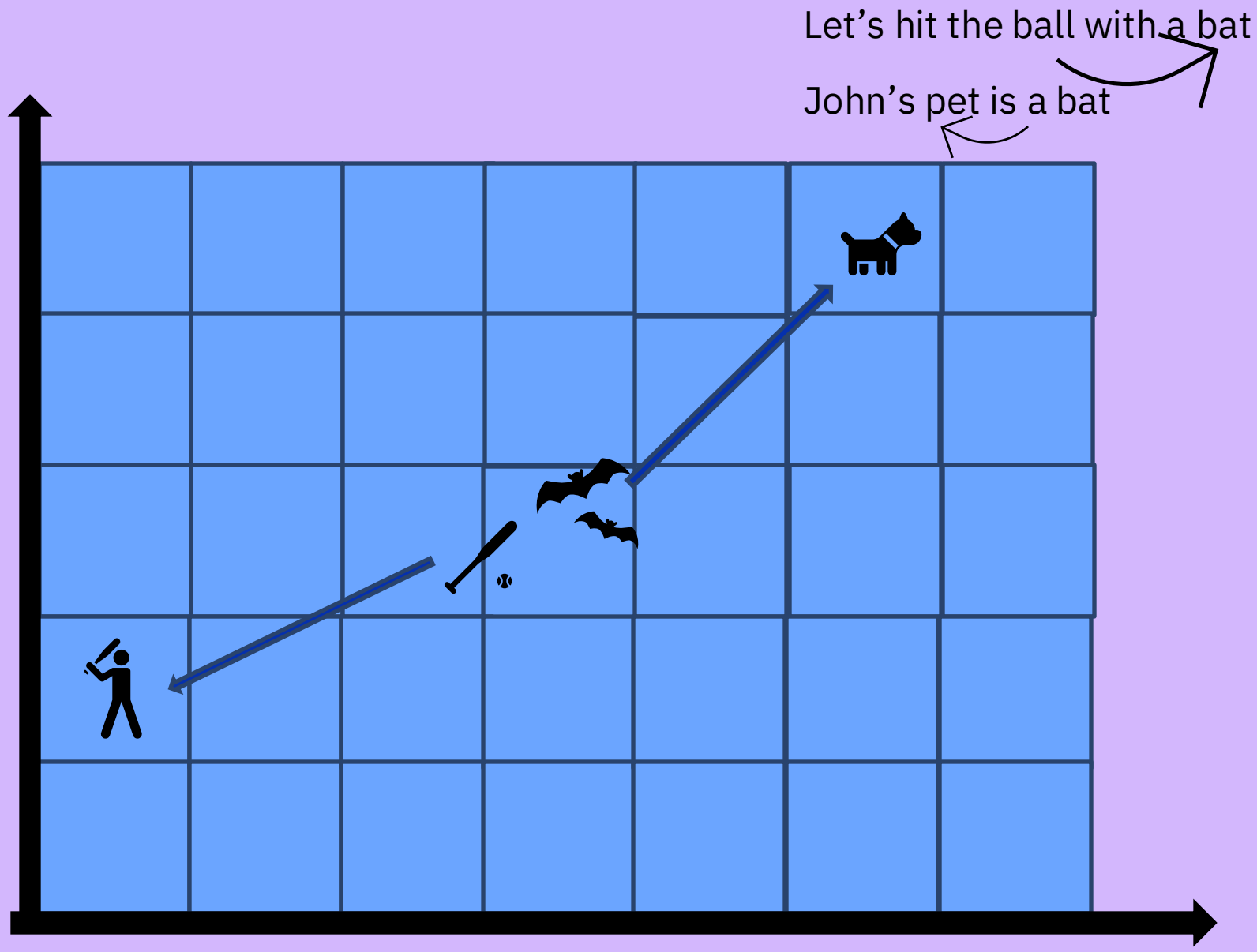
Similarity between words is obtained using:

- Dot-product
- Cosine similarity
- Scaled dot product

So how does it know which way to ‘pull’ **bat**?

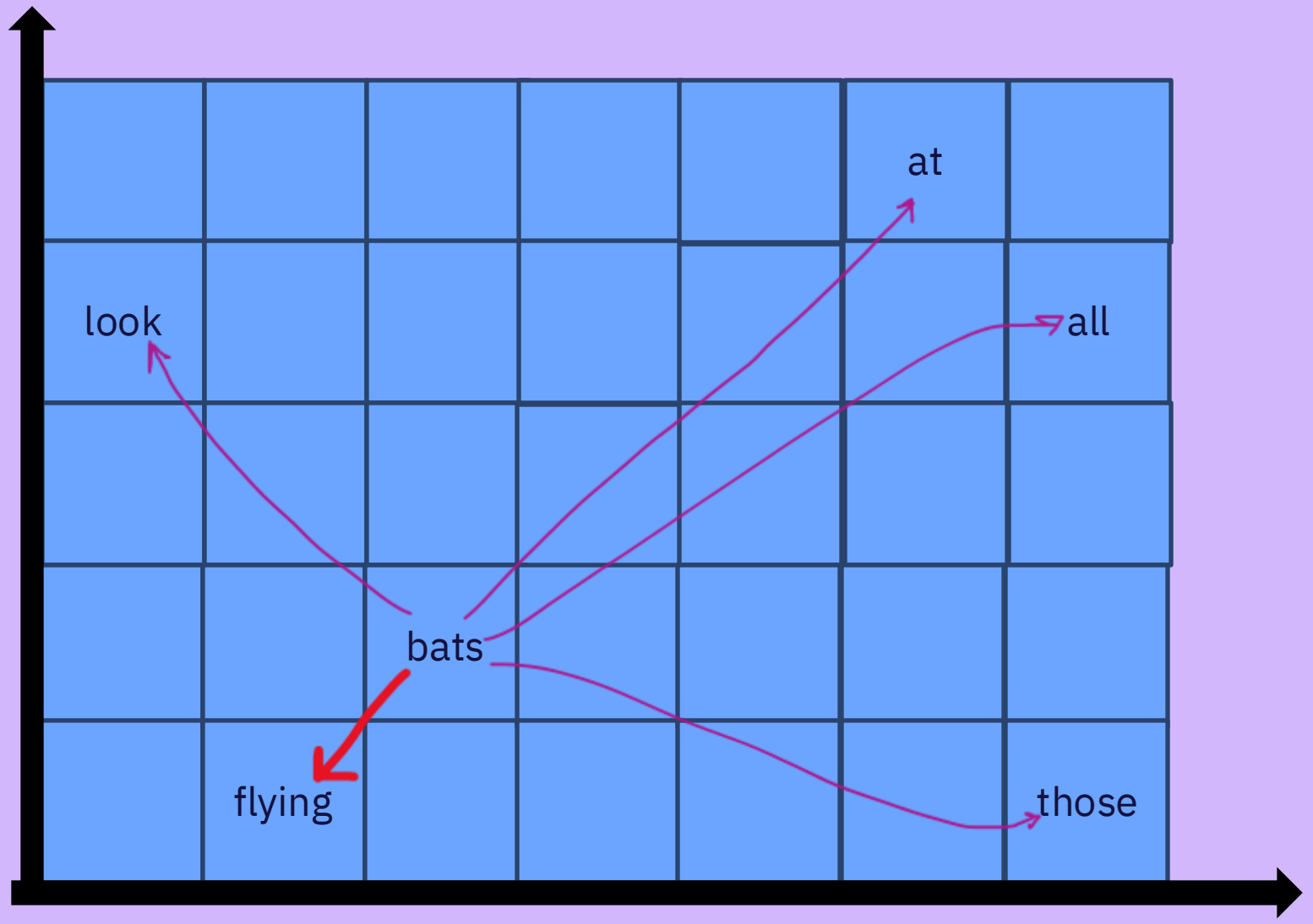


Context is important

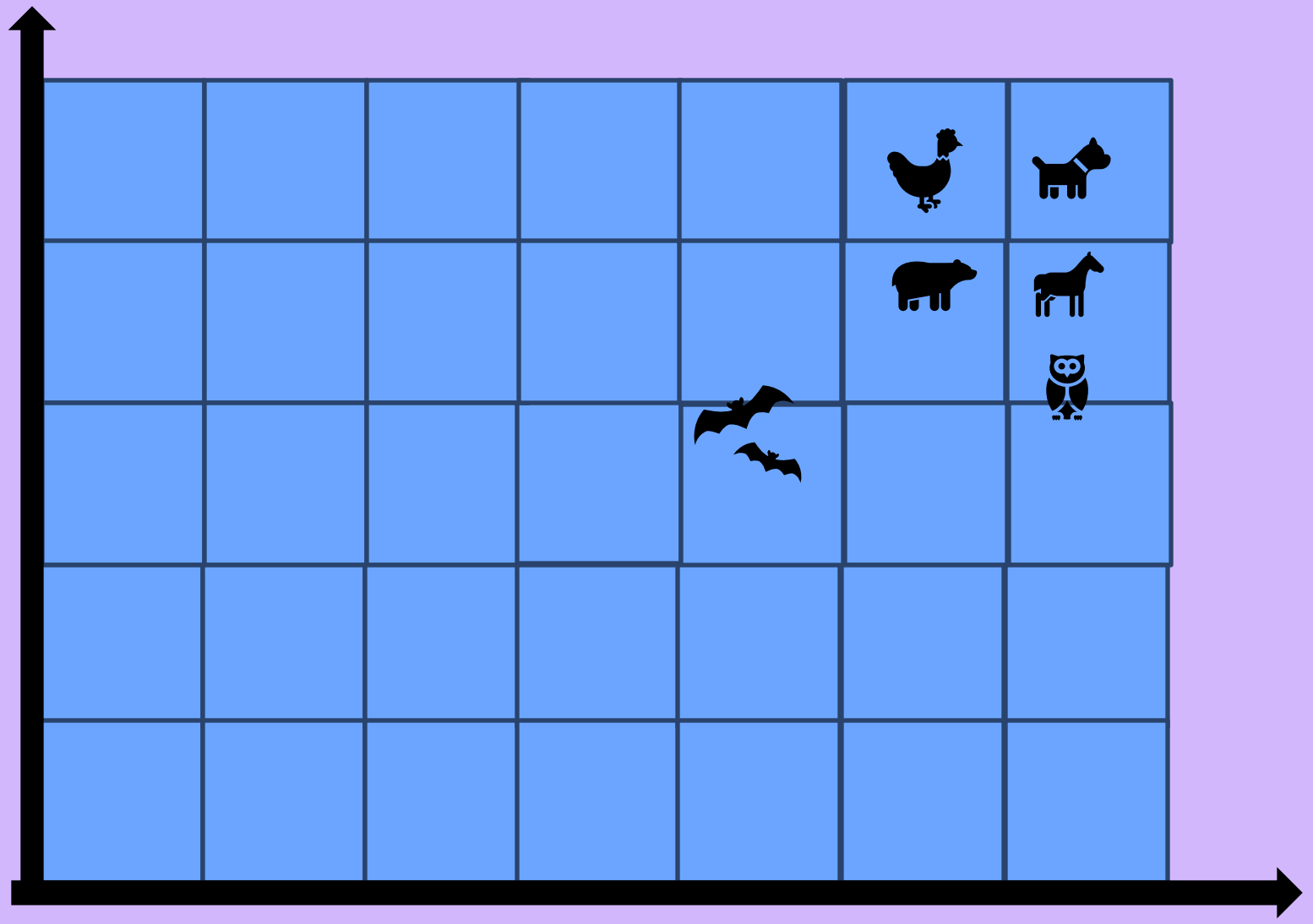


All the words are also pulling on bats

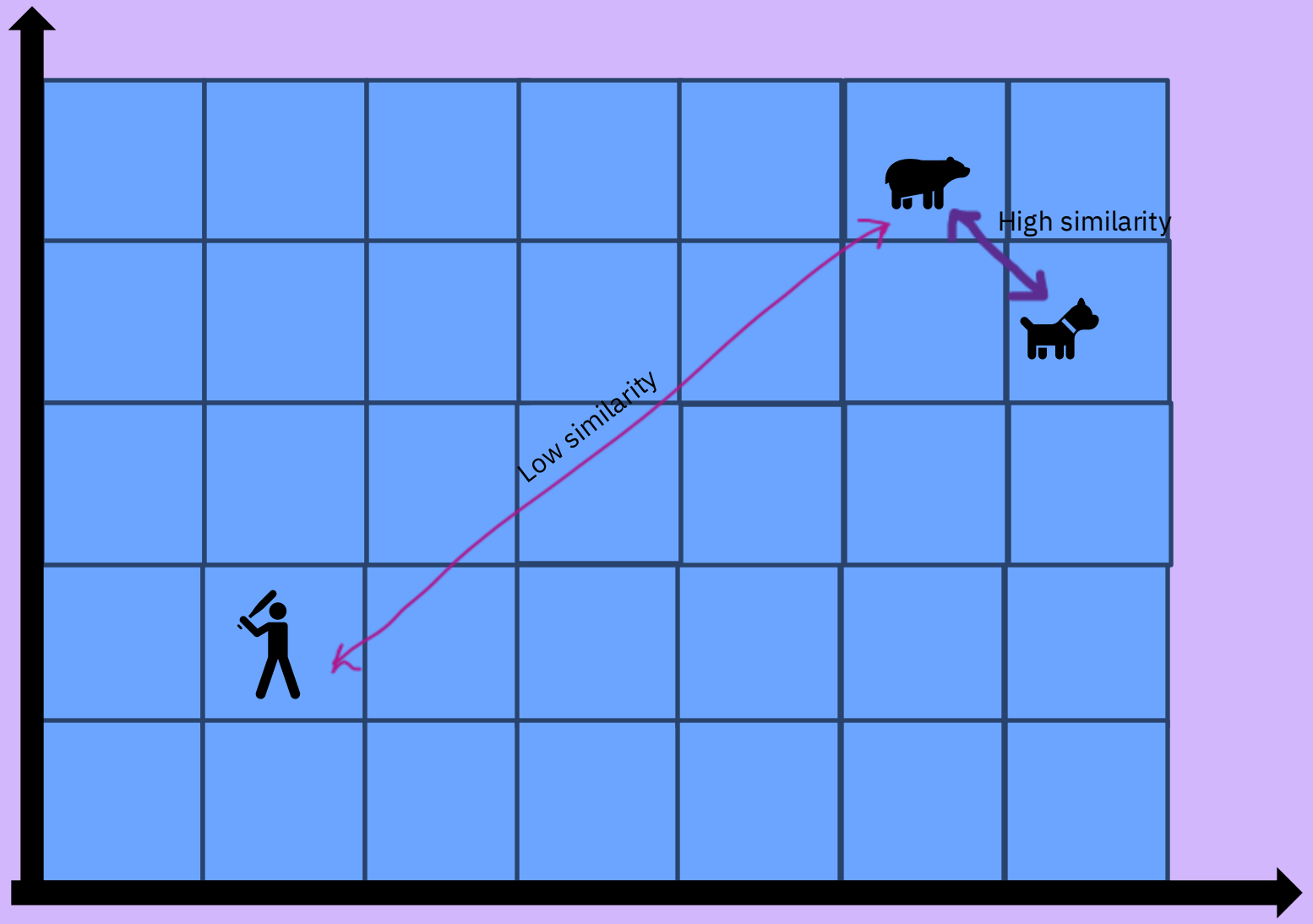
Some of the words are dis-similar, not as strong of a pull as the word flying



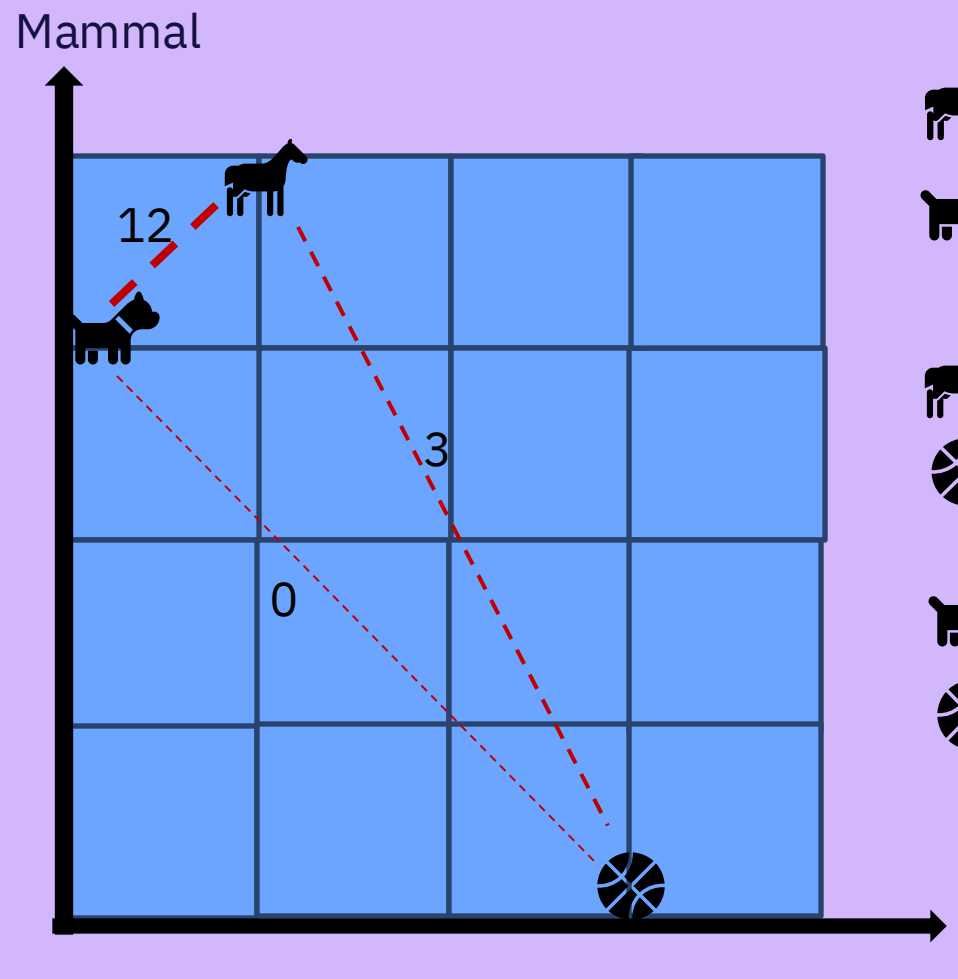
Context pulls



Let’s talk about measure of similarity



Dot Product Measure



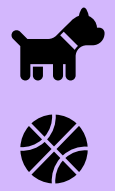
Sport	Mammal
1	4
0	3

Sport Mammal

1x0 + 4x3 = 12

Sport	Mammal
1	4
3	0

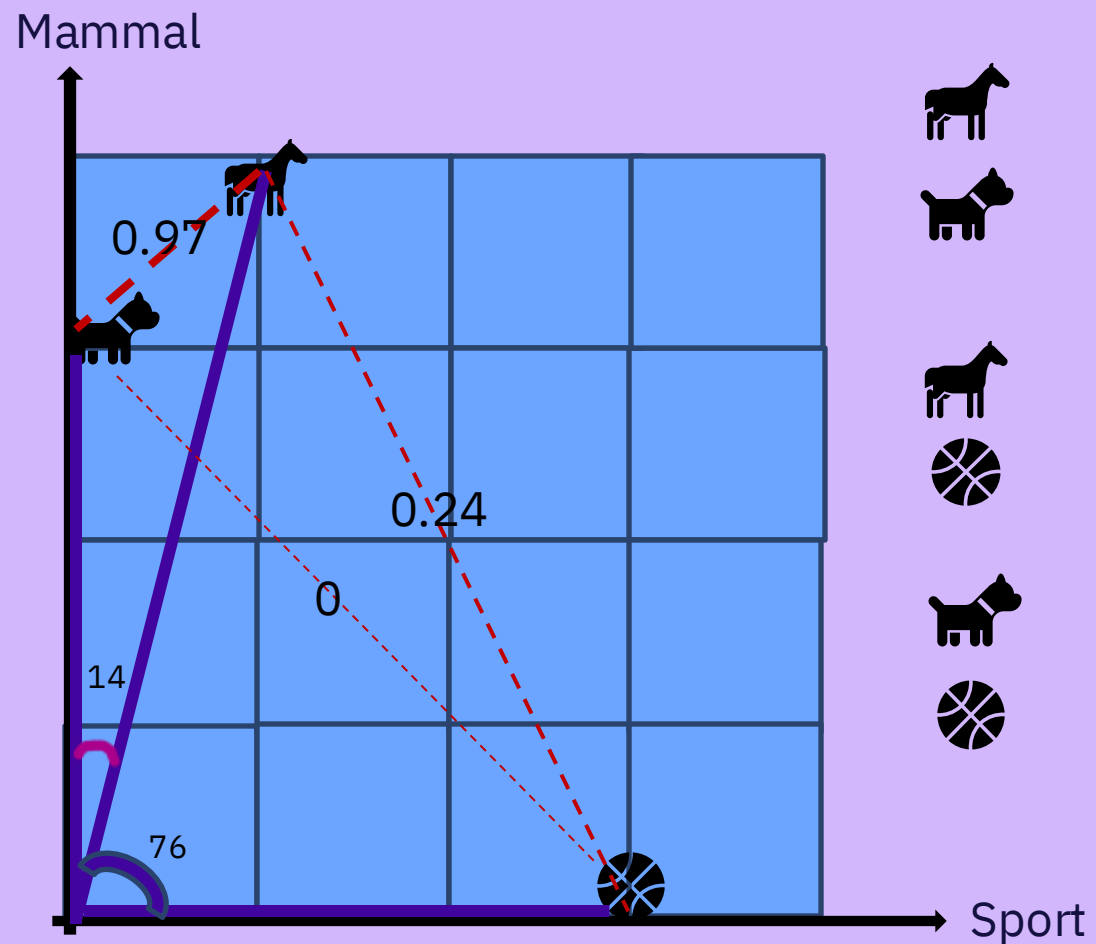
1x3 + 4x0 = 3



Sport	Mammal
0	3
3	0

0x3 + 3x0 = 0

Cosine similarity measure

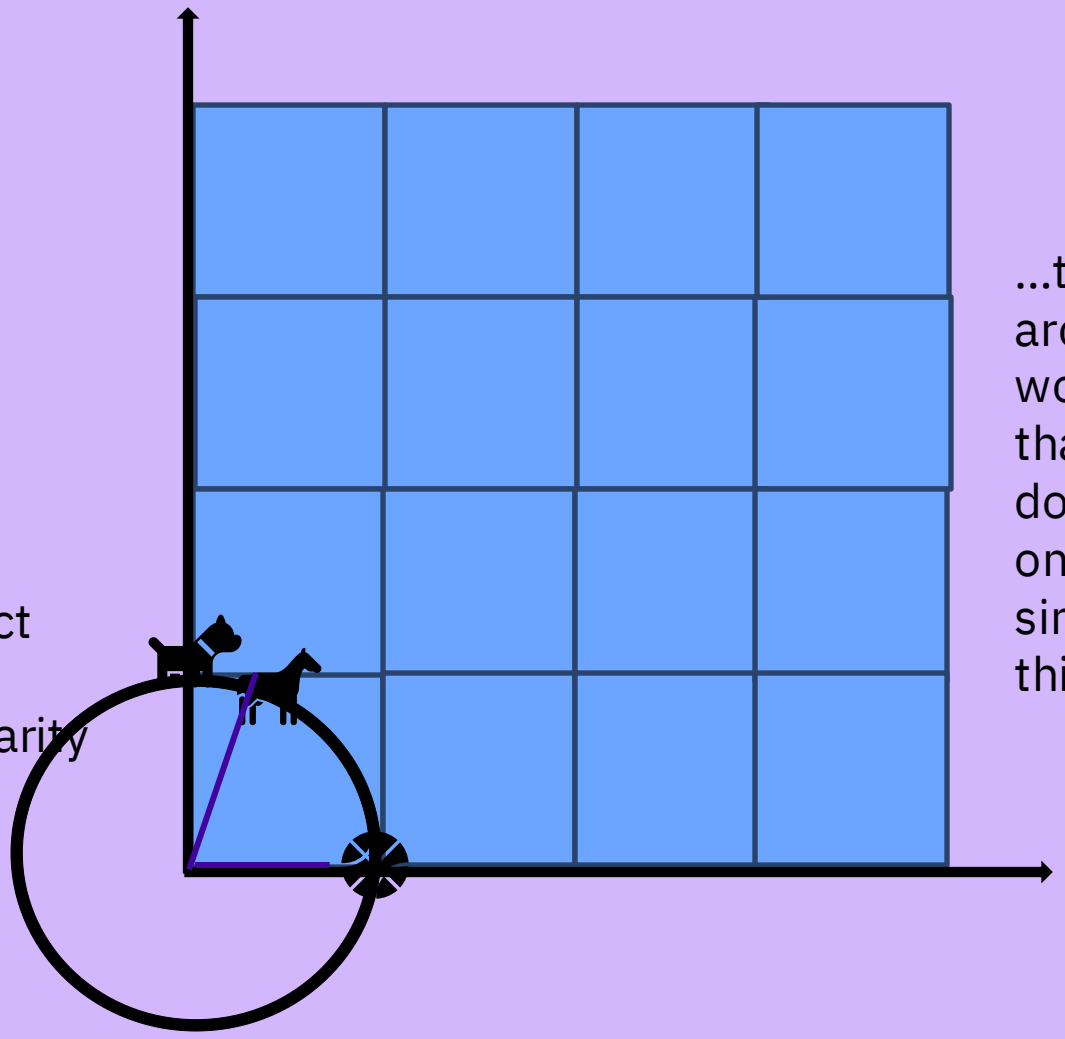


Cos (14) = 0.97

Cos (76) = 0.24 **θ**=arctan(1/4) ; deg = 180/π

Cos (90) = 0

Now if vectors have length one...



Dot-product
=
Cosine similarity

...that is, if I draw a unit circle around the origin, and put the word, where it meets the circle, that means, I scale everything down so all vectors have length one, then dot-product and cosine similarity are the exact same thing!

Scaled dot product measure

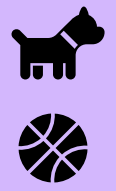
Dot product divided by the **square root of of the length of the vector**.
And this is the one that gets used in the Attention Mechanism!

1	4
0	3

1x0 + 4x3 = 12 → 12 / √2 = 8.49

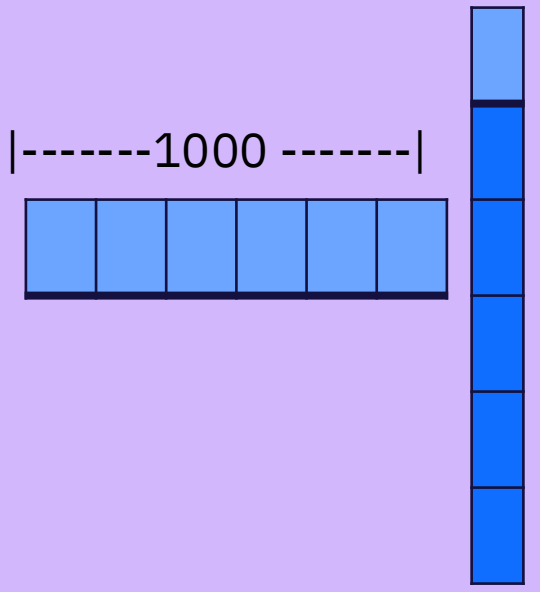
1	4
3	0

1x3 + 4x0 = 3 → 3 / √2 = 2.12



0	3
3	0

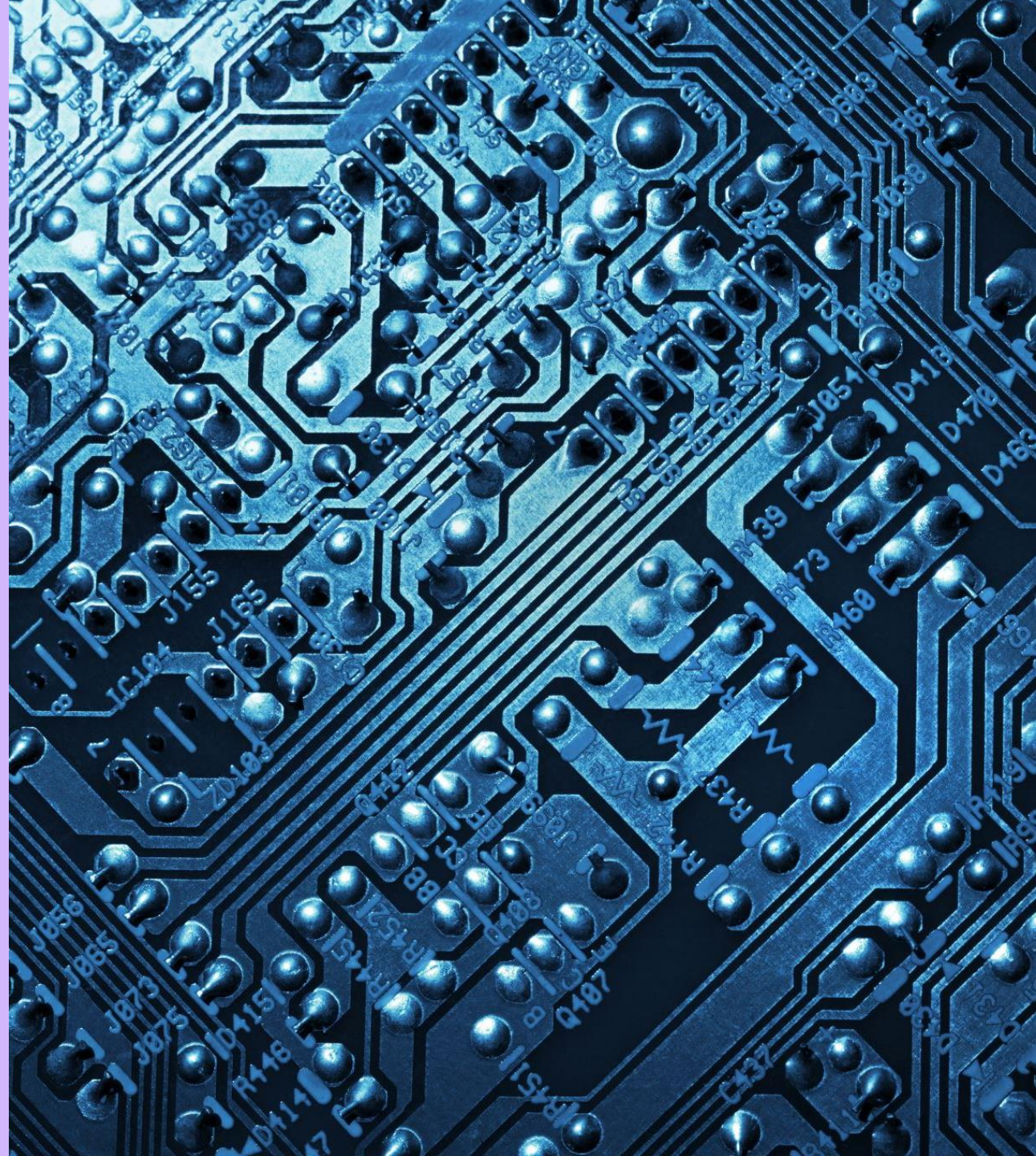
0x3 + 3x0 = 0 → 0 / √2 = 0



1. Overview of Gen AI
2. Attention is All you Need
3. It’s About Cosine Similarity
4. **What’s Next for Gen AI**
5. Is AI going to Replace Me?
6. Resources

Latest AI Trends in

1. Reality check: not just a chatbot, but rather integrated with other tools
2. Multi-modal: bring on text, images and video
3. Smaller is better: we are reaching the end of largeness in LLMs
4. GPU and Cloud costs:
5. Model optimization: LoRA
6. Custom local models: RAG
7. Virtual agents: not just LLM, also Large Action Models (LAM)
8. Regulation: Performance, security, privacy, transparency
9. Shadow AI: unofficial use of AI at work and school
10. Your guess??



This is the Year of Reality Check...more Realistic Expectations

Shortly after Nov 2022, after ChatGPT was introduced to the masses, the craze was chatbots and Dall-E for vision.

This year many GenAI tools are being implemented as integrated elements, rather than stand-alone chatbots.

They enhance and complement existing tools rather than supplant them. Think Copilot features in MS Office or Generative Fill in Adobe Photoshop.

Embedding AI into everyday workflows helps us better understand what GenAI can and cannot do.

Multi-modal AI

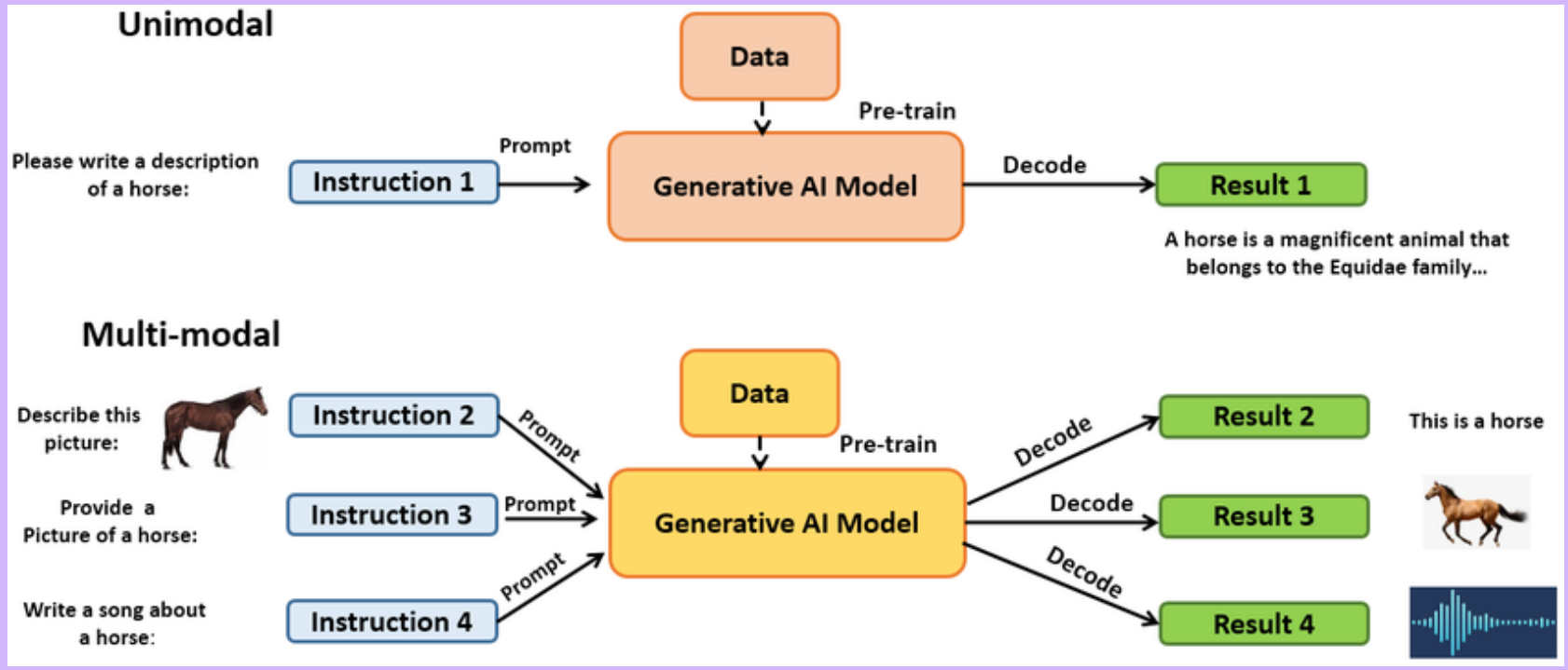
Multi-modal AI can take multiple layers of data as input.

We already have multidisciplinary models today like OpenAI's **GPT-4v** or Google **Gemini** that can move between NLP and Vision tasks.

New models are also bringing videos into the fold.

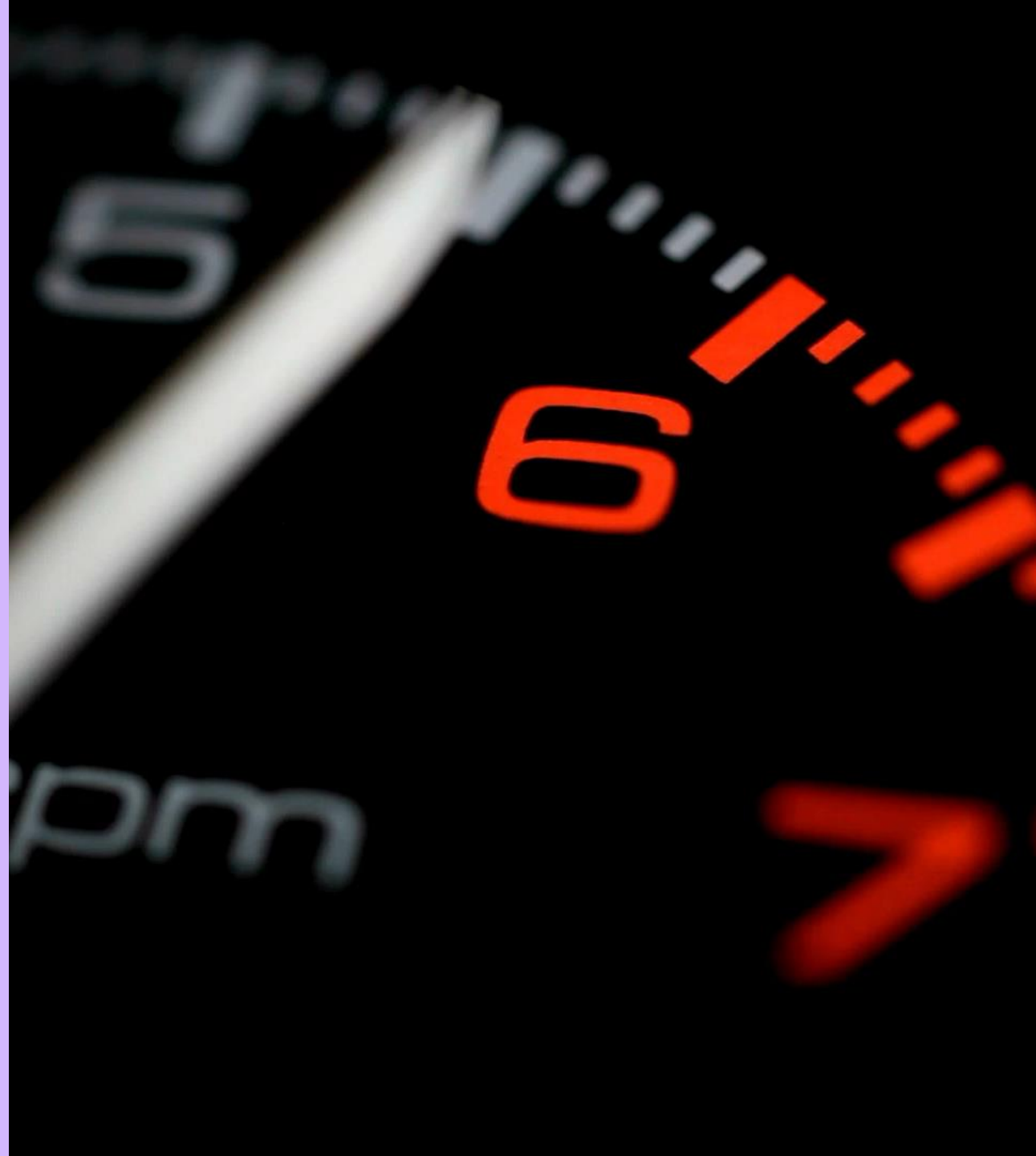
Self-driving cars rely heavily on multimodal AI.

We're on a path to create humanoids – systems that require human-like interactions or operations in environments designed for humans.



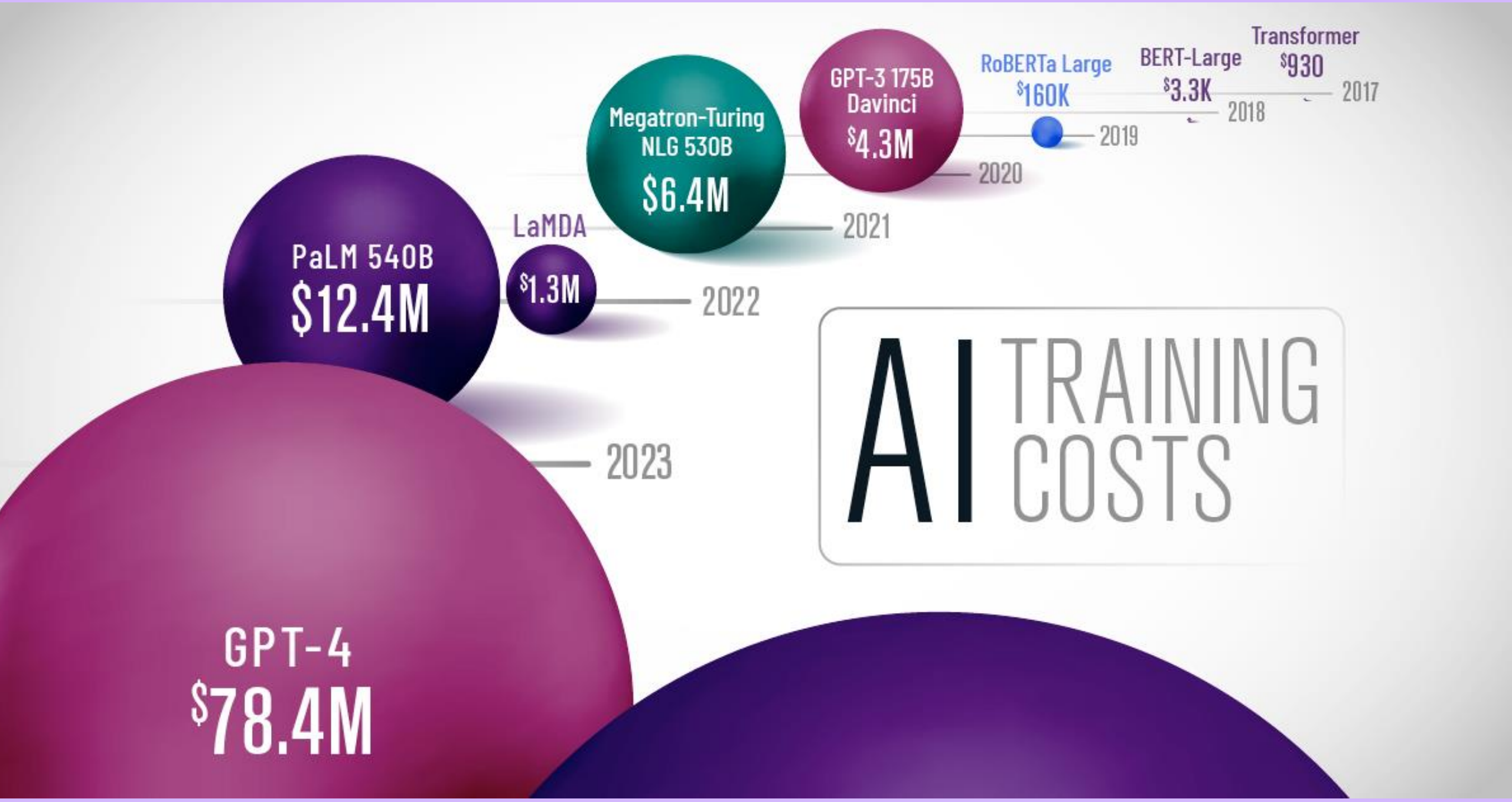
Unimodal vs Multimodal AI. Source: [ResearchGate]

Maximum AI with Minimum Data



Smaller is Better

- In the beginning, the craze was about how large is your LLM? And have you reached **trillion** parameters yet?
- Yes we have, but at an eye-watering costs: A recent study by the University of Washington, **training** a GPT3 size model requires the **yearly** electricity consumption of over **1000** households.
- So you may say, OK I get it, training large models does and should cost that much, but what about making an **inference**? That rivals the **daily** energy consumption of close to **33,000** households.
- The idea is to get greater output with fewer parameters. **GPT 4** is rumored to have around **1.76 trillion** parameters.
- Many **Open-source** models have great accuracy with model sizes in the **3 to 17 billion** parameter range.
- December of 2023, Mistral (French AI form) released **Mixtral**. This is called Mixture of Experts, or **MoE** model: **8 Neural networks each with 7B parameters**. It outperforms the 70B parameter of Llama 2 at 6 times faster Inference speed and outperforms OpenAi's far larger GPT-3.5 on most benchmarks.


AI TRAINING COSTS
Transformer
$930
2017
BERT-Large
$3.3K
2018
RoBERTa Large
$160K
2019
GPT-3 175B
Davinci
$4.3M
2020
Megatron-Turing
NLG 530B
$6.4M
2021
LaMDA
$1.3M
2022
PaLM 540B
$12.4M
2023
GPT-4
$78.4M

The expense is not sustainable

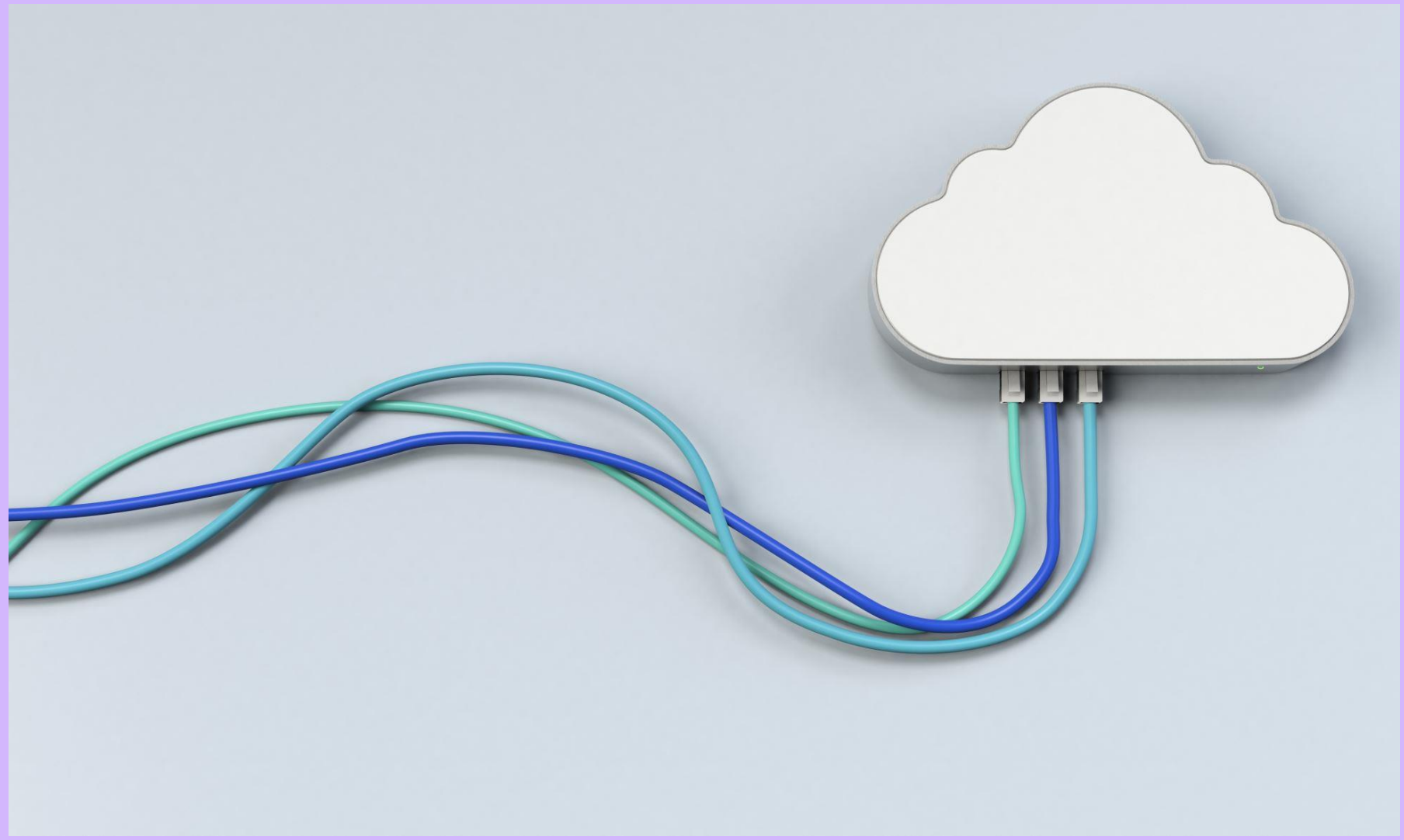
The larger the model the higher the requirements on GPU for training and inferences.

Everyone is scrambling to obtain the necessary GPUs to power the infrastructure, which puts upward pressure on Cloud costs.

2020: GPT3 175B $4.3M

2023: GPT4 1.3T $78.4M

What if these models were a bit more optimized...???



AI Model Optimization

This past year we have seen adoption of techniques for training, tweaking and fine-tuning pre-trained models like quantization.

For example reducing the file size of an audio file or video file by just lowering its bitrate.

In a similar vane, Quantization lowers the precision used to data points. For example from 16-bit floating poirepresentnt to eight-bit integer to reduce memory usage and speed up inference.

...and then, there is LoRA: Low-Rank Adaptation. Freezing pre-trained model weights and injecting trainable layers in each Transformer block. LoRA reduces the number of parameters that need to be updated, which speeds up fine-tuning and reduces memory needed to store model updates

```python
import numpy as np

# Original weight matrix W of size 6x4
W = np.array([
    [4, 1, 3, 2],
    [2, 3, 1, 4],
    [1, 2, 5, 3],
    [3, 4, 2, 1],
    [5, 3, 2, 4],
    [2, 1, 4, 3]
])

# Desired change matrix Delta W approximated by low-rank matrices A and B
A = np.array([
    [0.76, 1.19],
    [1.31, 0.83],
    [-0.36, 1.91],
    [1.39, 0.75],
    [1.93, 1.14],
    [-0.03, 1.61]
])

B = np.array([
    [1.72, 1.39, -0.38, 0.85],
    [1.15, 0.91, 2.53, 1.74]
])

# Compute the approximated change matrix Delta W
Delta_W_approx = np.dot(A, B)

# Updated weight matrix W'
W_prime = W + Delta_W_approx

# Print results
print("Original weight matrix W:\n", W)
print("\nApproximated change matrix Delta_W_approx:\n", Delta_W_approx)
print("\nUpdated weight matrix W_prime:\n", W_prime)

# Output:
Original weight matrix W:
 [[4 1 3 2]
 [2 3 1 4]
 [1 2 5 3]
 [3 4 2 1]
 [5 3 2 4]
 [2 1 4 3]]

Approximated change matrix Delta_W_approx:
 [[2.6757 2.1393 2.7219 2.7166]
 [3.2077 2.5762 1.6021 2.5577]
 [1.5773 1.2377 4.9691 3.0174]
 [3.2533 2.6146 1.3693 2.4865]
 [4.6306 3.7201 2.1508 3.6241]
 [1.7999 1.4234 4.0847 2.7759]]

Updated weight matrix W_prime:
 [[6.6757 3.1393 5.7219 4.7166]
 [5.2077 5.5762 2.6021 6.5577]
 [2.5773 3.2377 9.9691 6.0174]
 [6.2533 6.6146 3.3693 3.4865]
 [9.6306 6.7201 4.1508 7.6241]
 [3.7999 2.4234 8.0847 5.7759]]
```

Key Terms Regarding LoRA

1. **Weights**: In a machine learning model, weights are the parameters that the model learns to adjust during training. They determine how input data is transformed into output predictions.

2. **Matrix**: A matrix is a grid of numbers with rows and columns. In neural networks, weights are often stored in matrices.

3. **Independent**: In the context of matrices, "independent" refers to the idea that the columns (or rows) of a matrix provide unique information. If columns are independent, no column can be formed by combining other columns.

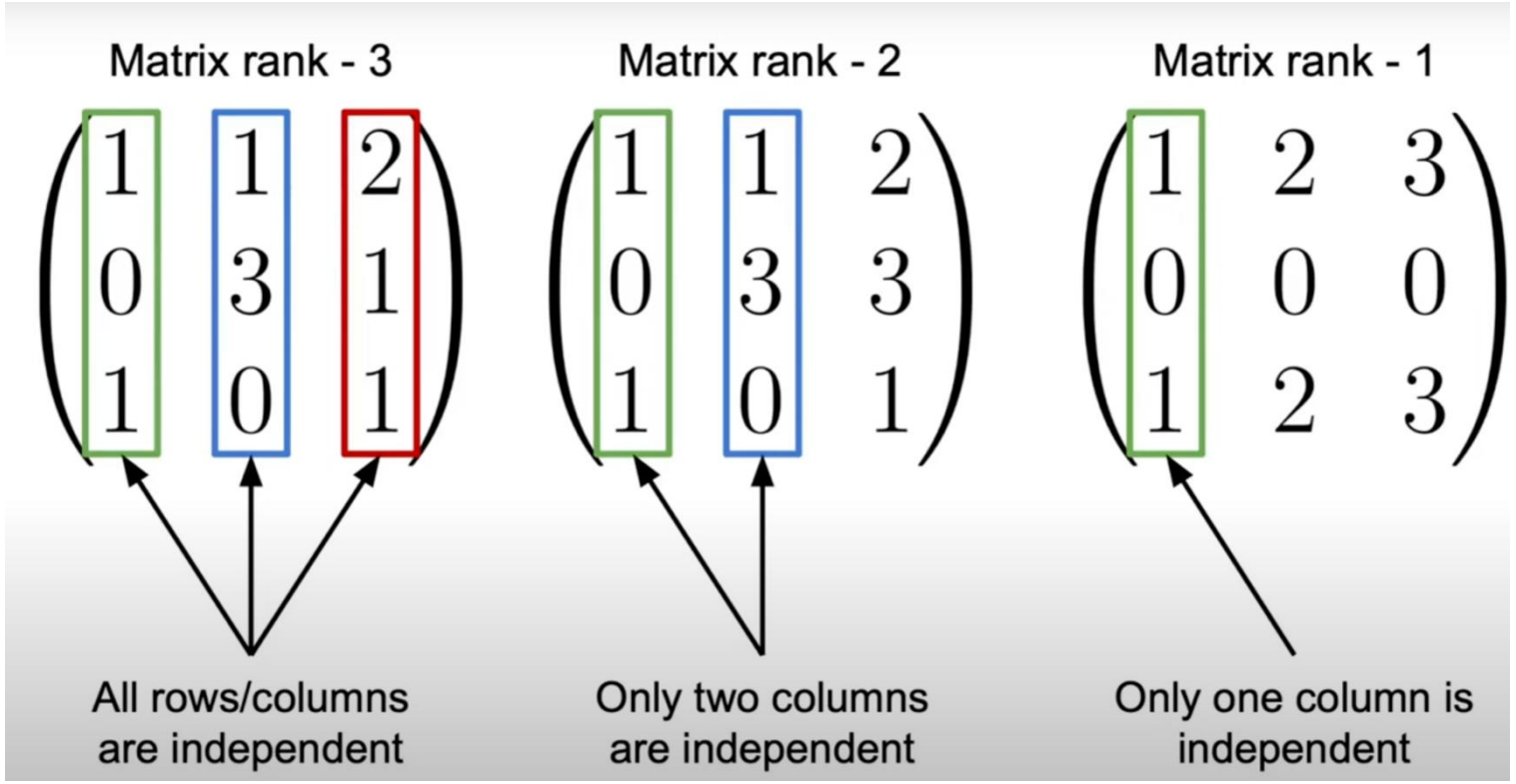
4. **Rank**: The rank of a matrix is the number of independent rows or columns it has. For example, if a matrix has 3 columns but only 2 of them are independent, its rank is 2. Rank gives a sense of how much unique information is in a matrix.

5. **Low-rank**: A matrix has low rank when the number of independent columns (or rows) is much smaller than the total number of columns (or rows). Low-rank matrices are simpler and have fewer degrees of freedom.

It’s called Rank Decomposition

The **rank** of a matrix is the number of **independent** rows or columns it has. It tells you how much unique information the matrix contains.

This is how rank decomposition is used in LoRA, where the focus is on simplifying the matrix while retaining the key information.



Say hello to LoRA

Let's break down a simple **rank decomposition** using the Low-Rank Adaptation (LoRA) technique. We'll start with a matrix and perform a low-rank decomposition by breaking it into two smaller matrices, A and B, and show how they can approximate the original matrix W.

Example:

Suppose we have a simple matrix W, which is a 3×3 matrix (for simplicity), like this:

$$W = \begin{bmatrix} 6 & 2 & 4 \\ 1 & 3 & 5 \\ 7 & 8 & 9 \end{bmatrix}$$

Step 1: Low-Rank Decomposition

We'll approximate W using two smaller matrices, A and B, where A is 3×2 and B is 2×3, representing a **rank-2** approximation. The rank-2 means we will reduce the complexity of the matrix while still capturing the essential structure.

Let's assume A and B are as follows:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 0 \end{bmatrix}$$

Step 2: Compute $A \times B$

Now, we multiply matrices A and B to get an approximation for W. The multiplication of a 3×2 matrix A with a 2×3 matrix B will result in a 3×3 matrix, like W.

$$\Delta W = A \times B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \end{bmatrix} \times \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 0 \end{bmatrix}$$

Performing the matrix multiplication:

$$\Delta W = \begin{bmatrix} (1 \times 2 + 0 \times 1) & (1 \times 1 + 0 \times 2) & (1 \times 3 + 0 \times 0) \\ (0 \times 2 + 1 \times 1) & (0 \times 1 + 1 \times 2) & (0 \times 3 + 1 \times 0) \\ (1 \times 2 + 2 \times 1) & (1 \times 1 + 2 \times 2) & (1 \times 3 + 2 \times 0) \end{bmatrix}$$

$$\Delta W \downarrow \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 0 \\ 4 & 5 & 3 \end{bmatrix}$$

Step 3: Final Approximation of W

So, the matrix ΔW, which is the result of multiplying $A \times B$, is:

$$\Delta W = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 0 \\ 4 & 5 & 3 \end{bmatrix}$$

Now, if we started with a matrix W and applied a low-rank update in LoRA, the **new weight matrix** W_{new} would be:

$$W_{\text{new}} = W + \Delta W$$

Adding the two matrices element-wise:

$$W_{\text{new}} = \begin{bmatrix} 6 & 2 & 4 \\ 1 & 3 & 5 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 0 \\ 4 & 5 & 3 \end{bmatrix}$$

$$W_{\text{new}} = \begin{bmatrix} 8 & 3 & 7 \\ 2 & 5 & 5 \\ 11 & 13 & 12 \end{bmatrix}$$

Conclusion:

We have applied **Low-Rank Adaptation** by:

1. Approximating a matrix W with two smaller matrices A and B.
2. The product of $A \times B$ gives a low-rank matrix ΔW, which is the update applied to the original matrix.
3. The result is a new matrix W_{new} that is more efficient to compute and store.

This illustrates the basic idea of how LoRA reduces the computational load by focusing on low-rank approximations of large matrices.

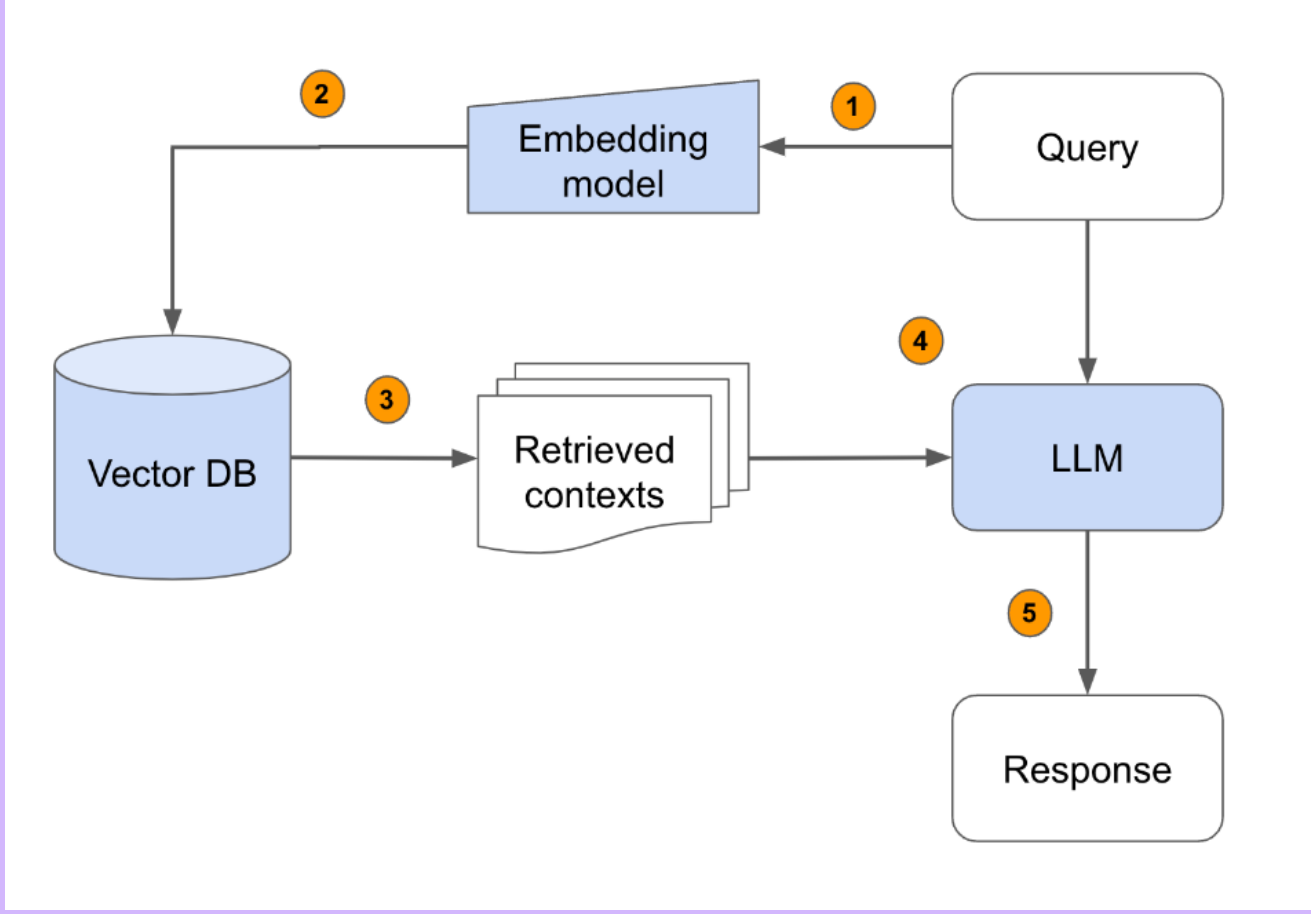
Custom Local Models - RAG

Open-source models lend well to developing powerful custom AI models.

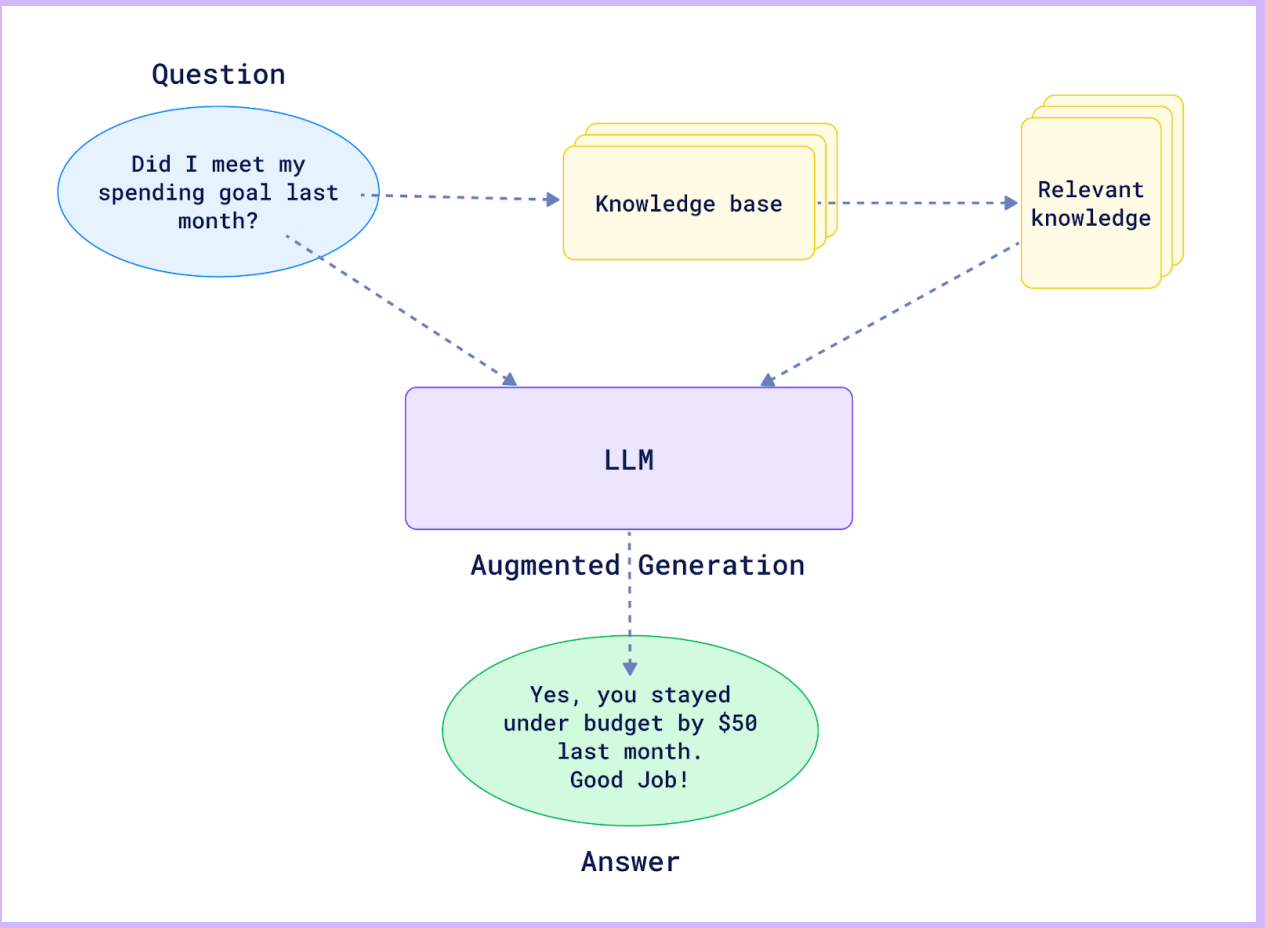
That means, trained on an organization's proprietary data and fine-tuned for their specific needs.

Tools such as RAG (Retrieval Augmented Generation) where you access company-specific information rather than storing it in a general-purpose publicly available LLM.

Sometimes the confidently given answer by the LLM is not so spot on. And it does not relay the source of its answer. That’s where RAG comes in.



Retrieval Augmented Generation (RAG)



Retrieval Augmented Generation (RAG) is a method used in AI to help generate more accurate and helpful answers by combining two things: **retrieving** relevant information from uploaded content and using that information to **generate** (thanks to the LLM) a response.

The **uploaded document** serves as a knowledge base for the LLM, ensuring that answers are informed by this specific content and not just by its general training.

The **LLM** plays two major roles in RAG:

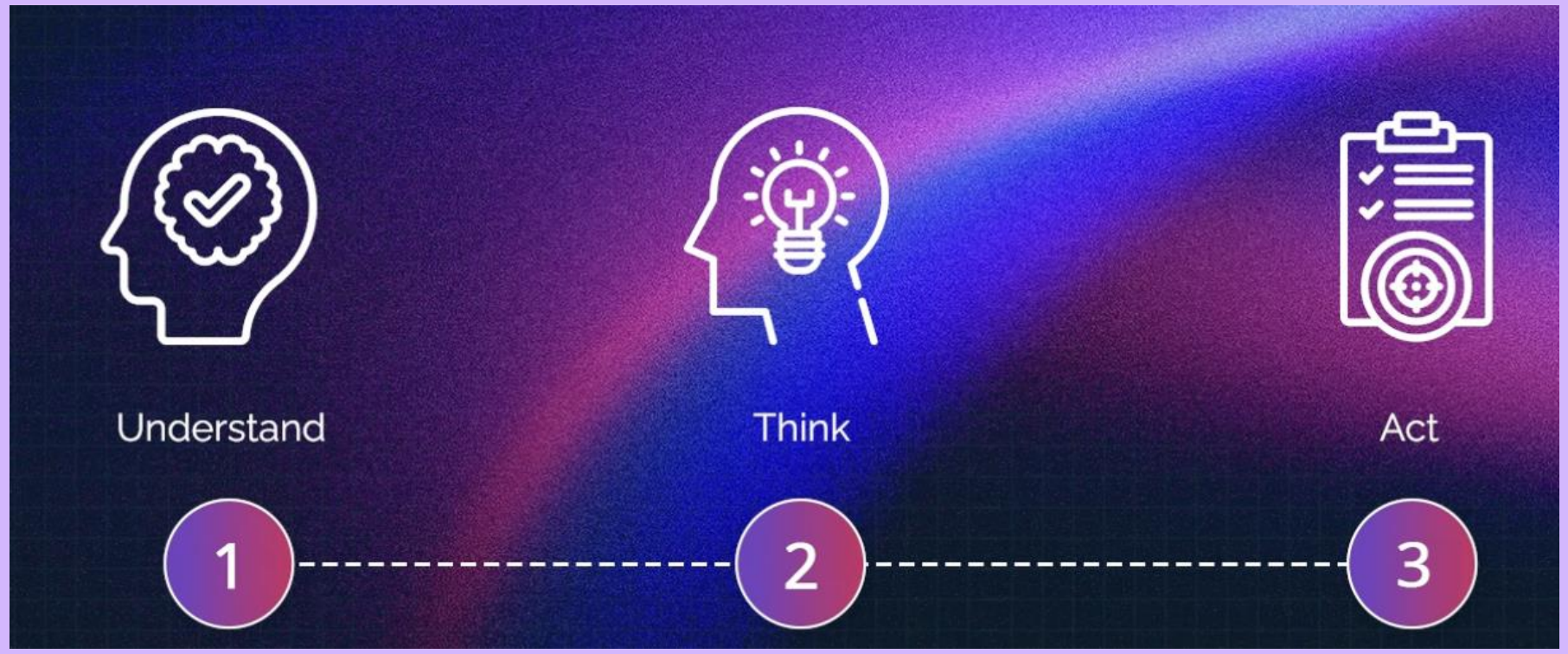
1. **Understanding the Query**: The LLM is excellent at interpreting the question or request from the user, even if it's phrased in a complex or conversational way.
2. **Generating a Response**: Once the LLM retrieves the relevant portions of the uploaded document, it combines that information with its own knowledge to generate a coherent, relevant response.

The AI Future is Agentic

It is more than just a chatbot answering your questions, but rather, agents carrying out tasks.

AI in 2024 is moving from monolithic models to Compound AI systems

LLMs will be connecting via APIs to LAMs (Large Action Models)



Features	RAG	AI Agents
Primary Focus	Knowledge Augmentation	Action and Interaction
Mechanism	Information Retrieval and Integration	Tool Utilization and Decision Making
Strengths	Improved Accuracy, Reliability, and Domain Expertise	Task Completion, Problem Solving and World Interaction
Limitations	Retrieval Performance, Static Context, Limited Interactivity	Tool dependency, Complexity of Agent Design, Ethical Considerations

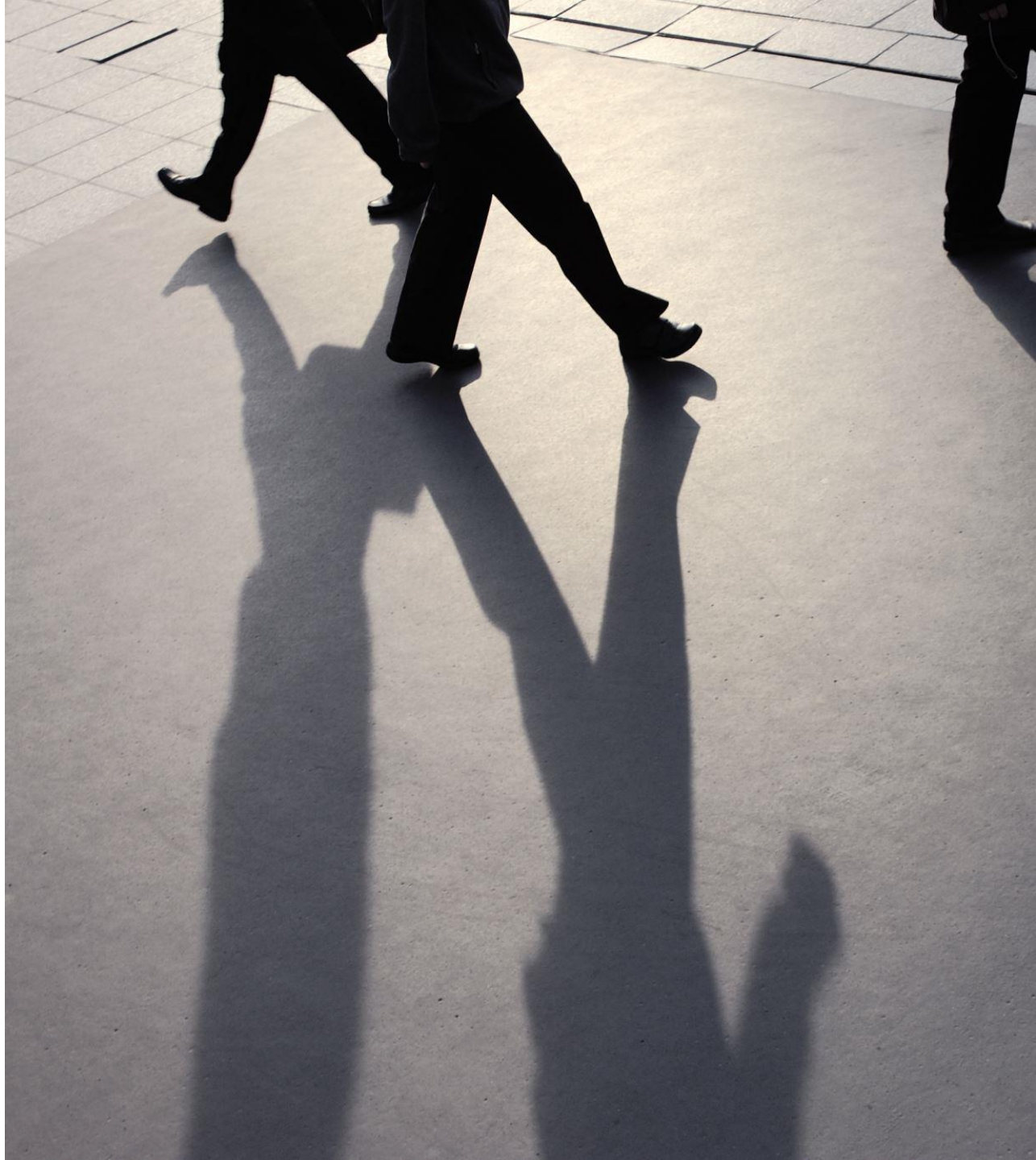
AI Governance Framework

PERFORMANCE	SECURITY	PRIVACY	TRANSPARENCY
ACCURACY	ADAPTABILITY	IP CAPTURE	EXPLAINABILITY
BIAS	ADVERSARIAL ROBUSTNESS	IMPACTED USERS	INTENT
COMPLETENESS			

jfgagne.ai

Shadow AI

General, unregulated use of GenAI tools at work and school.



1. Overview of Gen AI
2. Attention is All you Need
3. It’s About Cosine Similarity
4. What’s Next for Gen AI
5. **Is AI going to Replace Me?**
6. Resources

...not so fast!

Prediction relies on data, that means humans have **two advantages** over machines:

We know somethings that machines don't (yet).
We are better at deciding what to do when there isn't much data.

Humans have **three types of data** that machines don't:

We can gather data from our senses: smell, touch, feel, hunch...

We are the final arbitrators of our own preferences. This is why consumer data is extremely valuable. Store discount to customers that use loyalty cards.

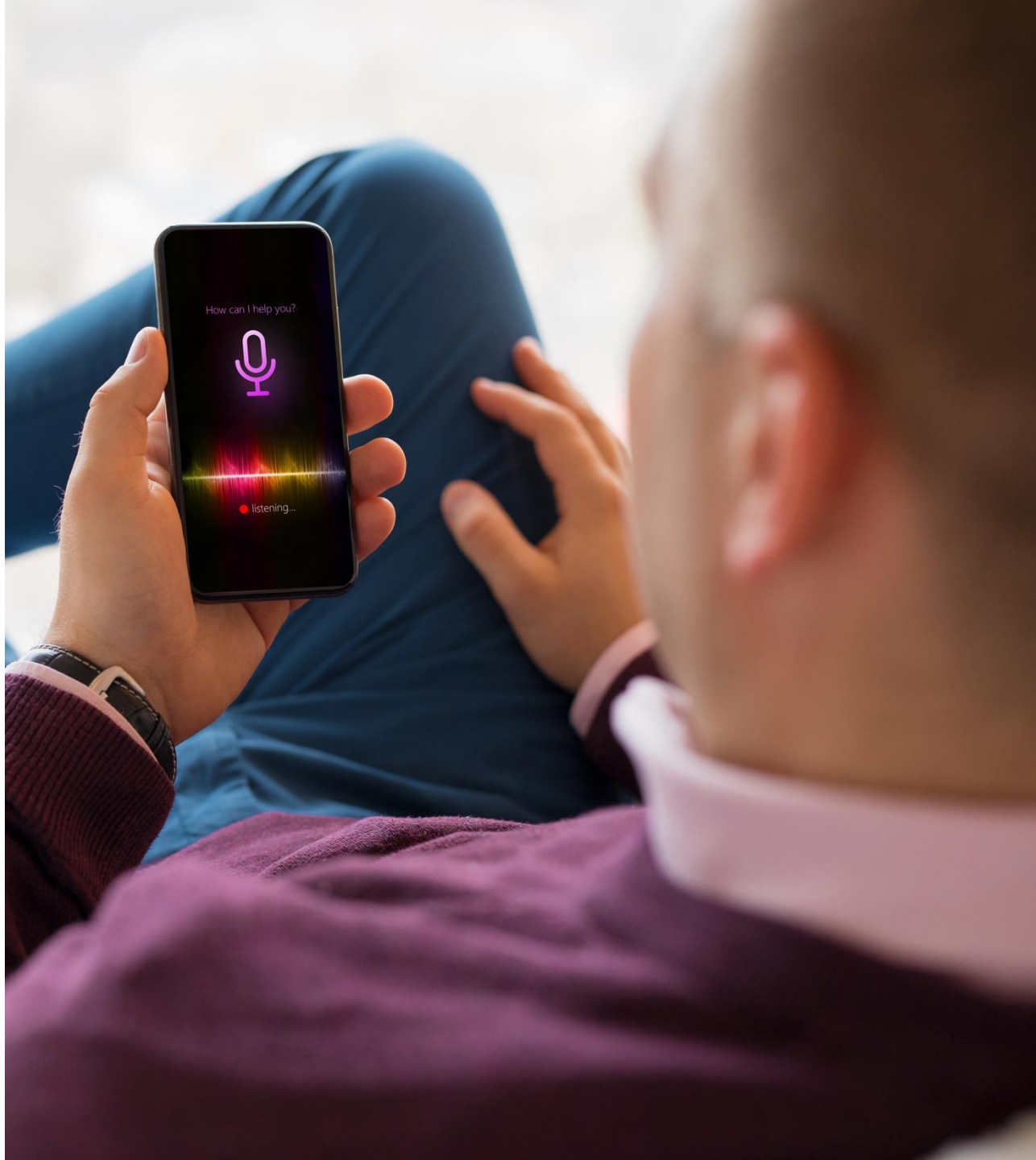
Privacy concerns restrict the data available to machines. As long as enough people keep their sexual activity, financial situation, mental health status and repugnant thoughts to themselves, the AI system will have insufficient data to predict behavior.

These excerpts are taken from the book **Prediction Machines** by Ajay Agrawal, Josh Gans and AVI Goldfarb

Narrow Intelligence

Narrow AI is focused on addressing very focused tasks

Example
Buying a book with a voice-based device based on “common knowledge”



Broad AI

Broad AI is about integrating AI within a specific industry knowledge and data to train this type of system.

Example

Self-driving cars are a collection of narrow AI systems that can make decisions

Artificial General Intelligence (AGI)

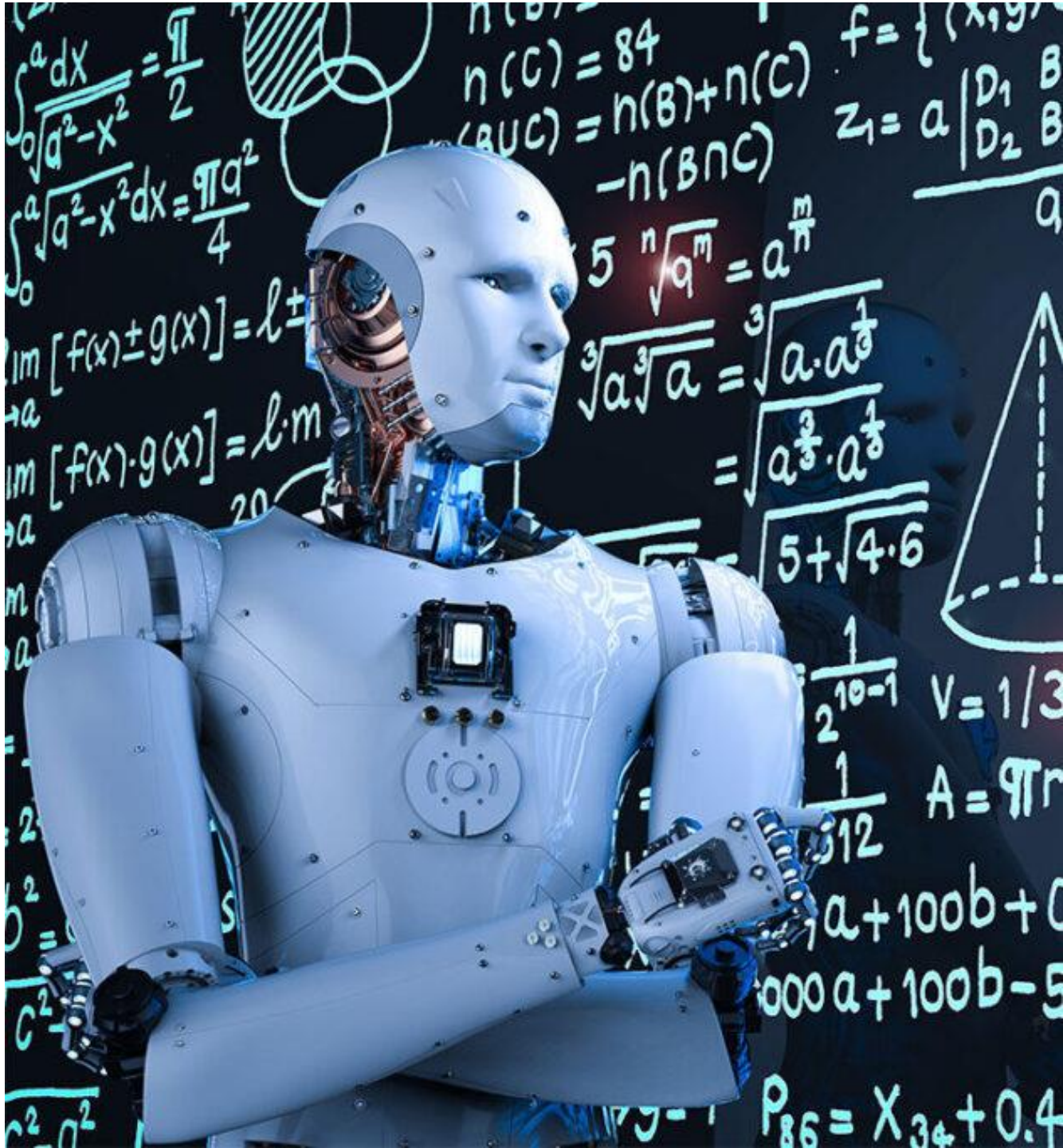
General AI refers to machines that can perform any intellectual task a human can.

Currently AI does not have the ability to use previous experiences to produce new creative ideas

Artificial Super Intelligence (ASI)

ASI is when...

...the machine is aware of itself!



Median Expert Prediction for AGI (2040)

Median Expert Prediction for ASI (2060)

1900

2000

Today

2100

waitbutwhy.com

1. Overview of Gen AI
2. Attention is All you Need
3. It’s About Cosine Similarity
4. What’s Next for Gen AI
5. Is AI going to Replace Me?
6. **Resources**

Resources

MIT News – Explained: Generative AI
https://news.mit.edu/2023/explained-generative-ai-1109

Forbes – What Is Generative AI: A Super-Simple Explanation Anyone Can Understand
https://www.forbes.com/sites/bernardmarr/2023/09/19/what-is-generative-ai-a-super-simple-explanation-anyone-can-understand/

McKinsey – What is ChatGPT, DALL-E, and generative AI?
https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-generative-ai

Harvard Business Review – How Generative AI Can Augment Human Creativity
https://hbr.org/2023/07/how-generative-ai-can-augment-human-creativity

IBM – What is Generative AI?
https://www.ibm.com/think/topics/generative-ai