CSCI 1470

Eric Ewing

Thursday, 11/6/25

Deep Learning

Day 18: Geometric Learning and other things

EEPS-DATA 1720

Tackling Climate Change with Machine Learning

Explore recent work that leverages machine learning (ML) as a tool for tackling climate change, with a focus on climate science and climate adaptation. We will discuss how modern machine learning can be used to assess, understand and respond to projected climate extremes, natural disasters, and environmental change. The target audience for this course is advanced undergraduate students or graduate students who are interested in using ML and AI to address high-impact global issues. Students will read and discuss recent research papers on ML for Climate and complete an original project as a member of a multidisciplinary team. Course topics evolve with the literature, but may include:

Climate themes: Climate models, Natural disasters and extreme weather, Farms and forests, Oceans and marine ecosystems, Sustainability of AI/computing

Machine learning topics: Physics-informed learning, Surrogate models and emulators, Explainable AI, Downscaling and superresolution, Learning with Limited Labels, Scientific Foundation Models.

EEPS 1720 is included on the list of non-CS courses that can count towards CS requirements.

Prerequisites:

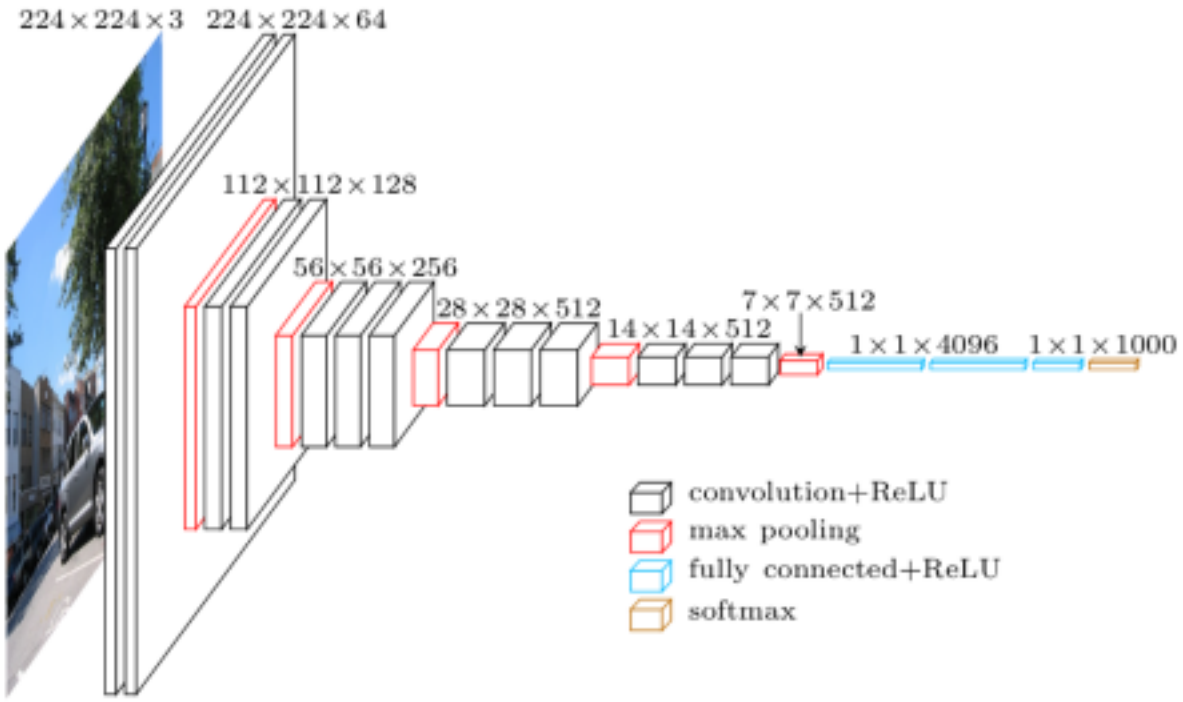
- Familiarity with the fundamentals of machine learning and deep learning through coursework (e.g. CSCI 1420, CSCI 1470), research or project experience.
- Programming experience in Python, Matlab, R or another high-level programming language.
- An interest in climate science and climate adaptation. Prior coursework in Earth or environmental science is *helpful, but not required*. Course lectures will introduce key scientific concepts.

- **Enrolling:** Due to the discussion-based format of the course, enrollment is limited. To enroll, (1) complete the EEPS 1720 Interest Form and (2) request an override on CAB.

Structured Data

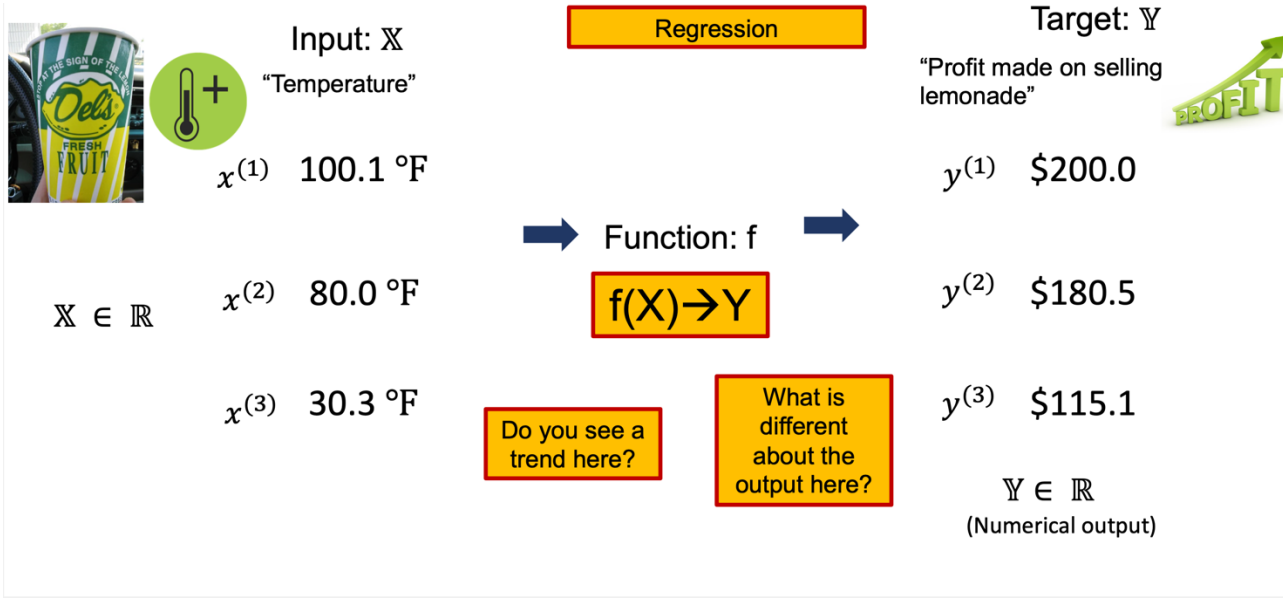
Images are a ***structured*** type of data

The structure (order and layout) of pixels matters

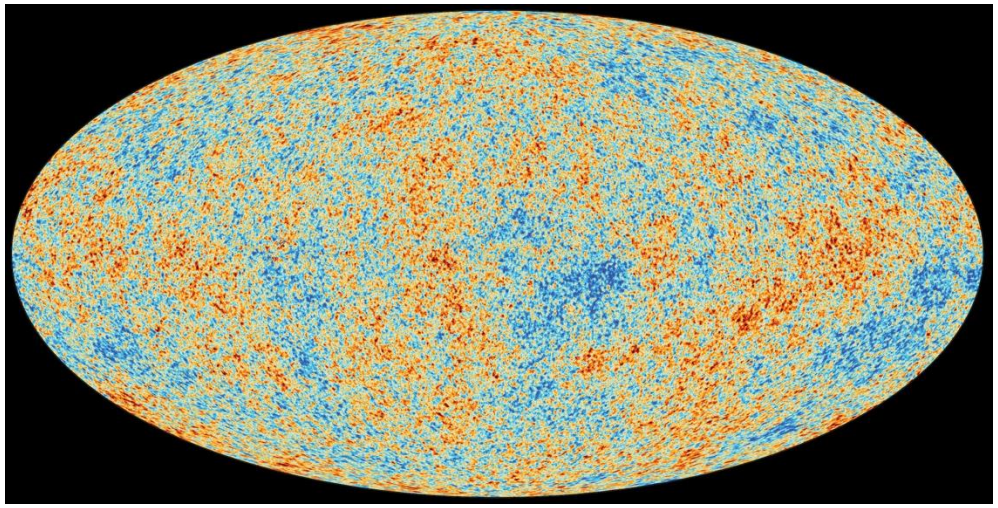


Lemonade stand data is ***not structured***

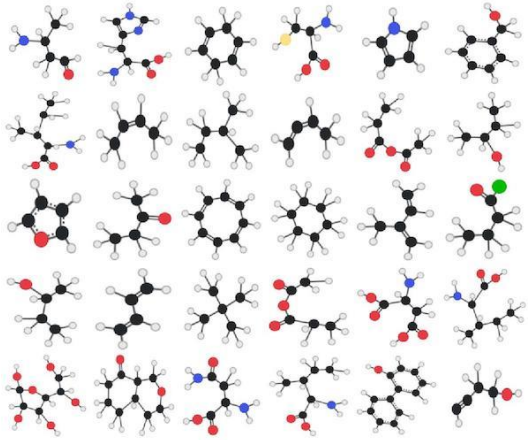
The order of features we use does not change training



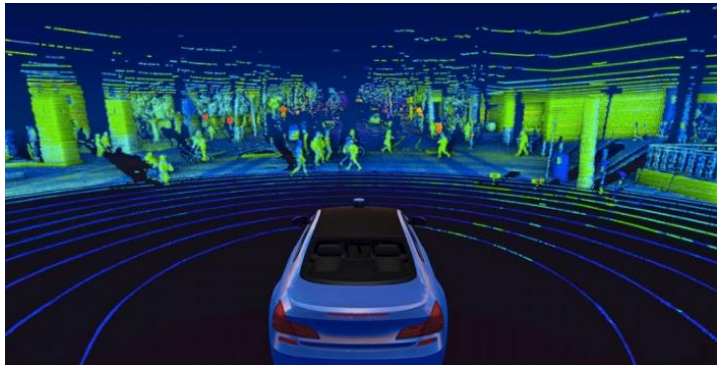
Other Types of Structured Data Types



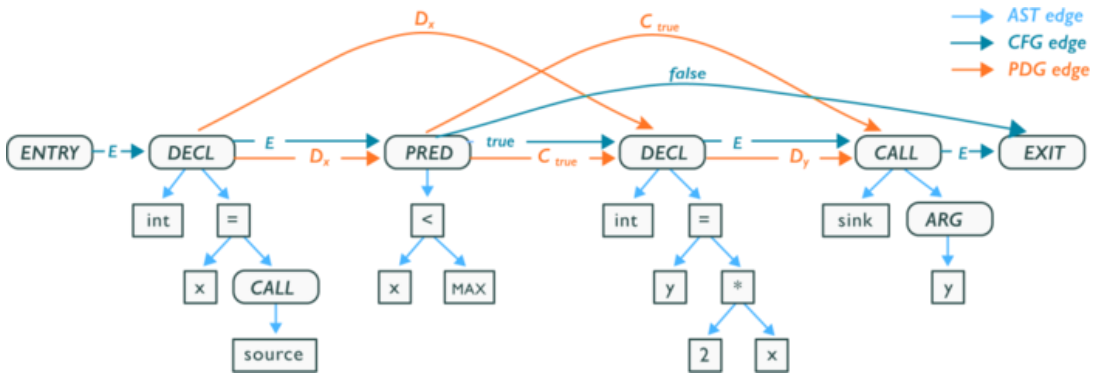
Microwave Background Radiation



Molecule Data



Point Clouds (from LIDAR)



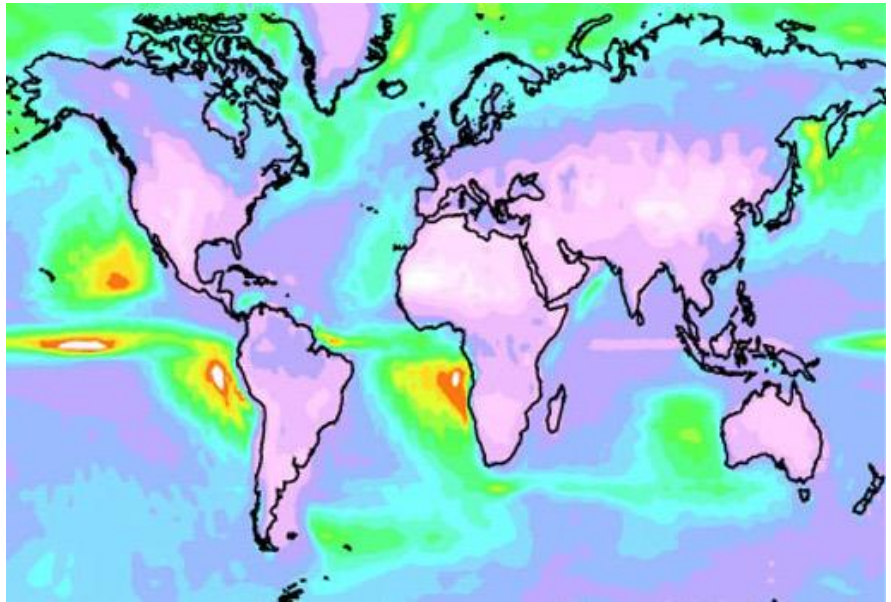
Code Graphs

Geometric Deep Learning

- We turned to convolutions for image data to give our networks “spatial reasoning”
- By Explicitly modelling the relationship of our data, we can achieve better results
- Geometric Deep Learning is the subfield of DL dedicated to learning representations of *structured* data.

Goal: Take as input data from satellites, predict carbon dioxide

Idea 1: Take all of the data as input, make it look like an image, use CNNs to learn to output carbon monoxide amounts

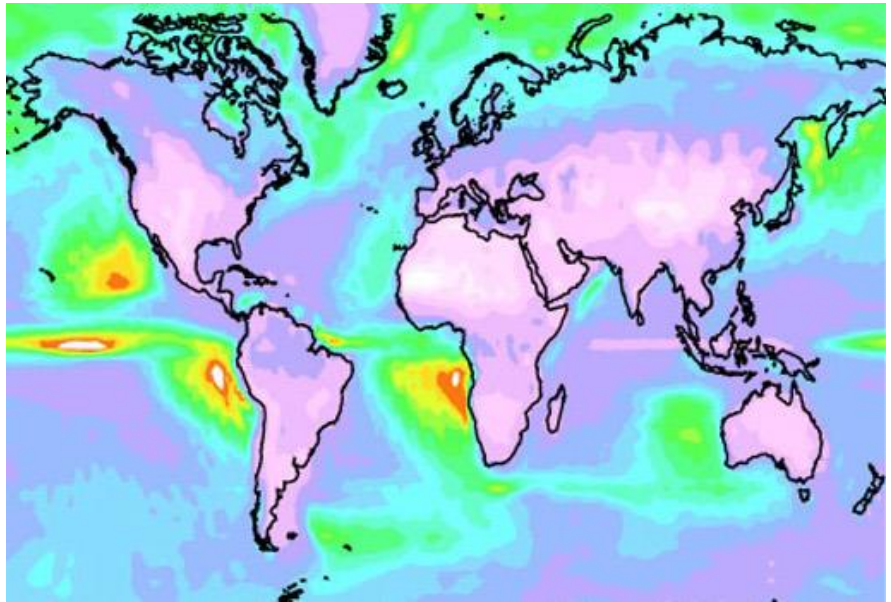


https://newton-climate.github.io/#research

Goal: Take as input data from satellites, predict carbon dioxide

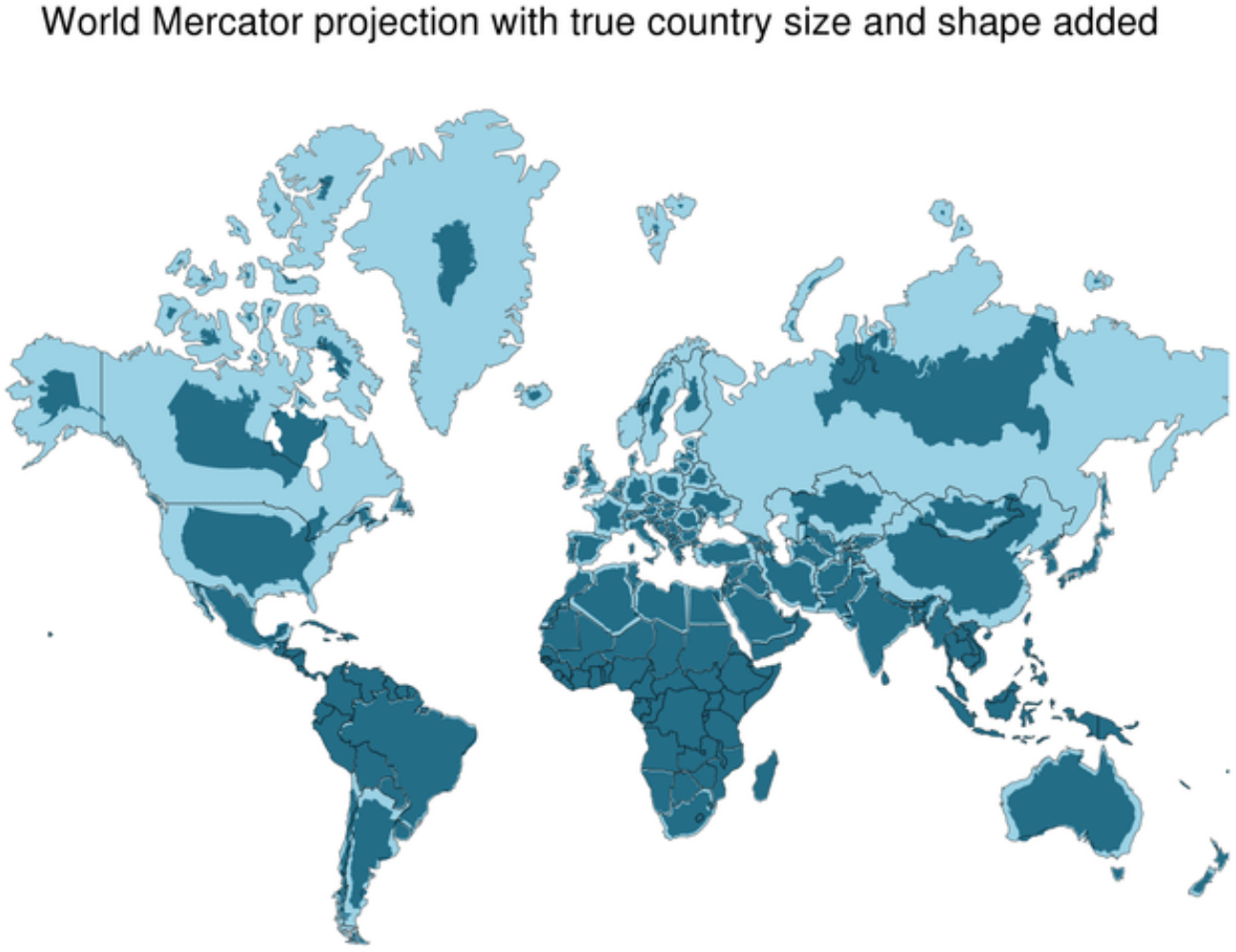
Idea 1: Take all of the data as input, make it look like an image, use CNNs to learn to output carbon monoxide amounts

Issue: Maps ☹



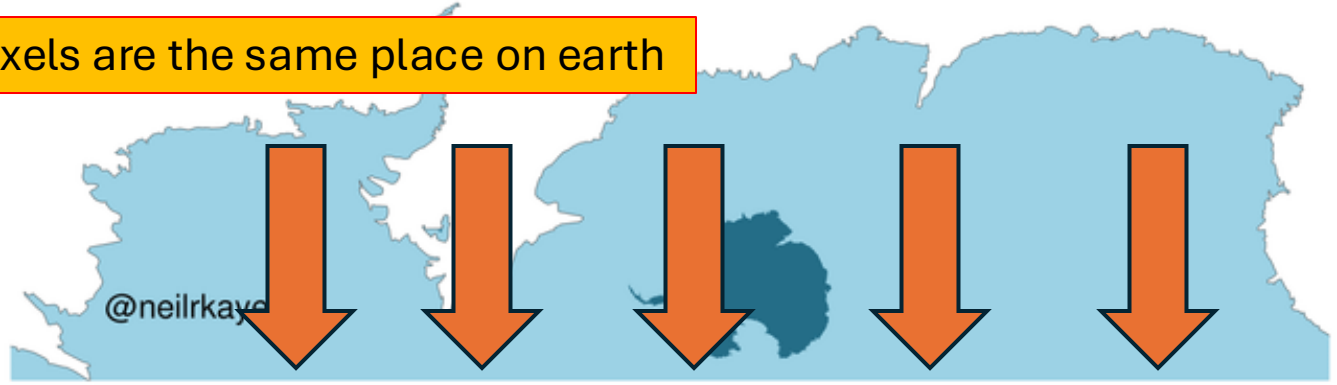
https://newton-climate.github.io/#research

Issue: Trying to represent a 3D sphere on a 2D plane



A 2D convolution doesn't make sense here...
But there's still structure (don't just want an MLP)
What about a Spherical Convolution!

All these pixels are the same place on earth



What should we pad the side of the image with?

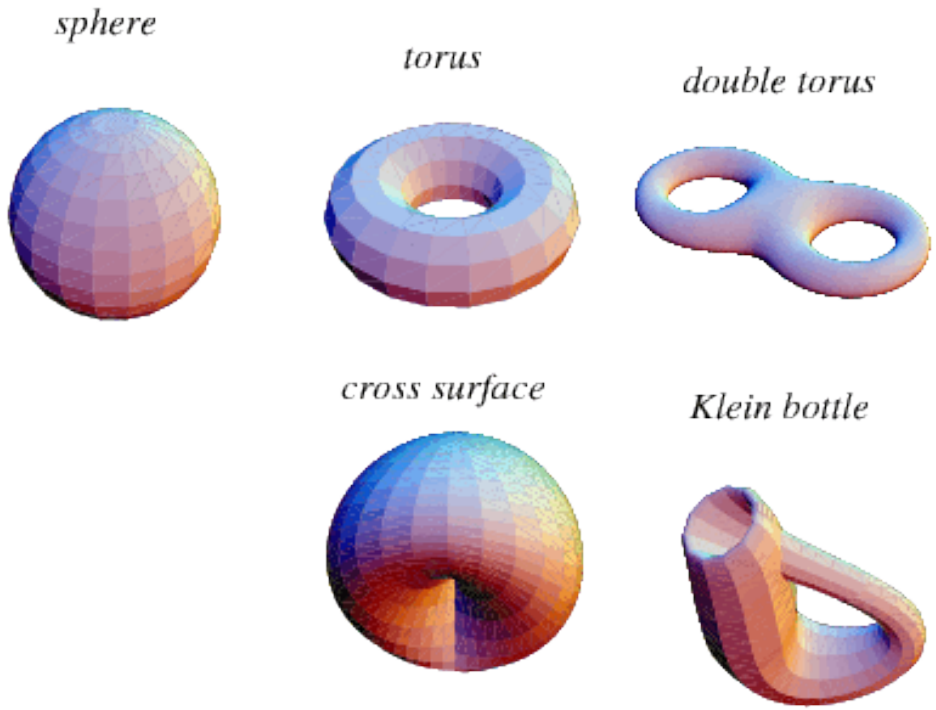
Instead of padding with 0's, "pad" with adjacent locations on earth

Convert 2D convolution on grids to work on Spheres

Manifolds

- Topological Space that looks locally like Euclidean space, but can be globally curved
  - For example, the Earth
- Can we generalize convolutions to more than just spheres?

Need to specify which points are “close” to other points, convolve neighborhoods of points together

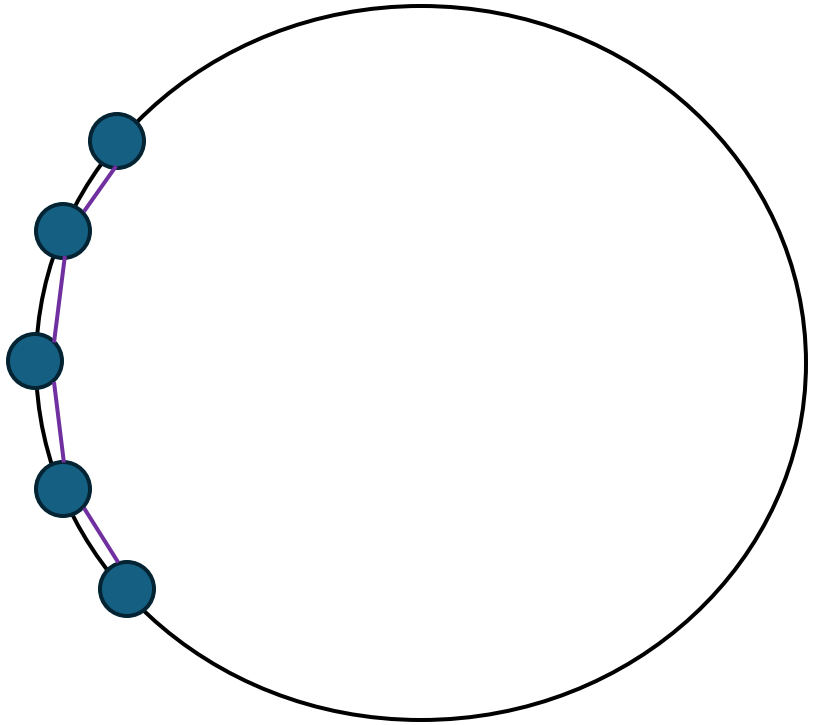


Extending Convolutions to Manifolds

A circle is a manifold in 2D
(like a sphere or taurus in 3D)

For every point on the manifold,
connect it to “close” points

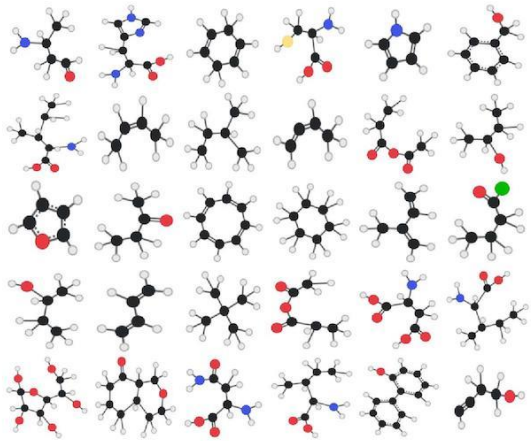
What data structure does this lead to?



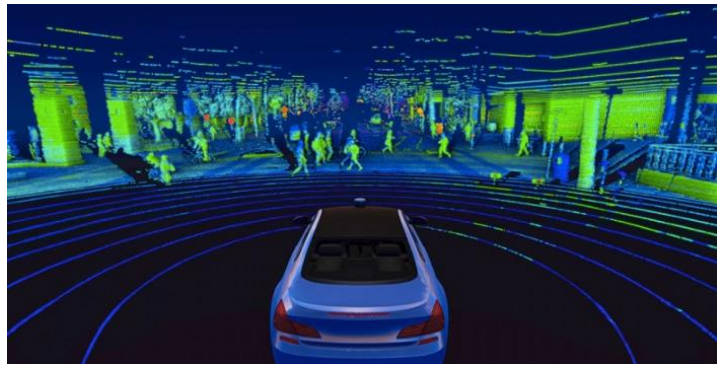
Geometric Deep Learning

- It is often confusing why much of Geometric Deep Learning deals with graphs…
  - Geometry happens in continuous space
  - Graphs are discrete

Molecules

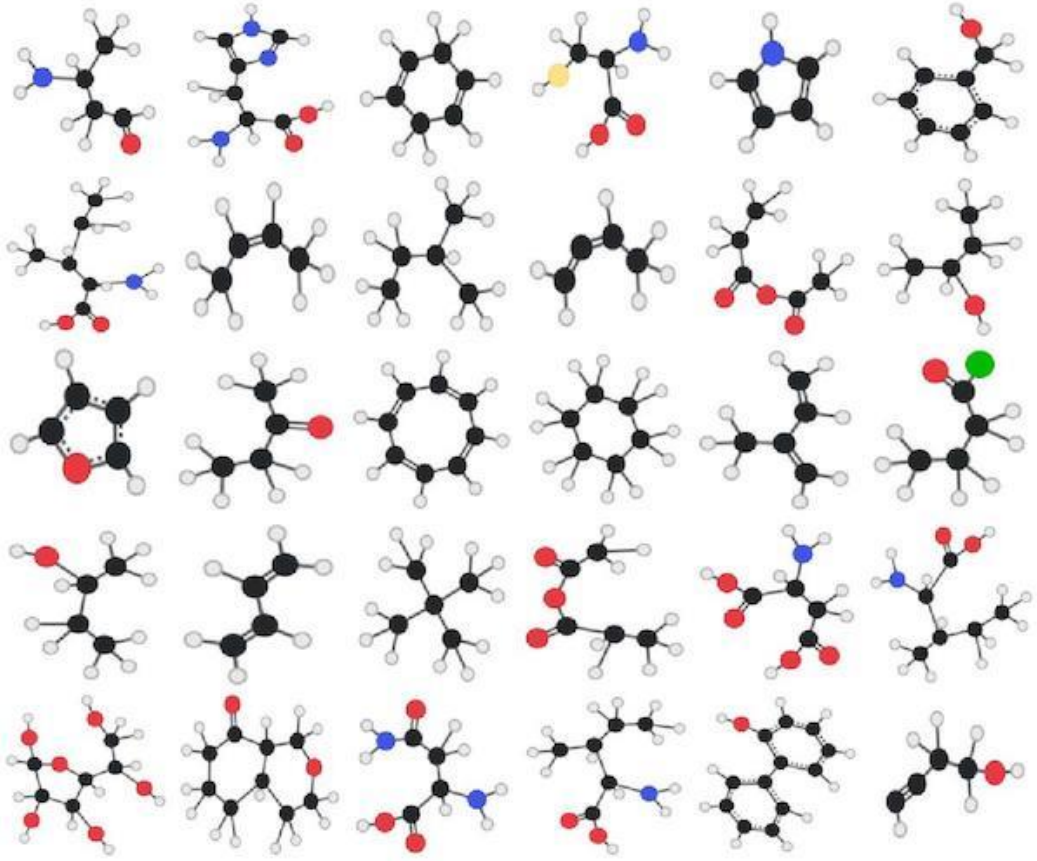


LIDAR Data



What's Hard About Neural Networks for Graphs?

- Can have different numbers of nodes
- Can be arbitrarily complex
  - Different numbers of edges
  - Different edge patterns
  - Weighted edges



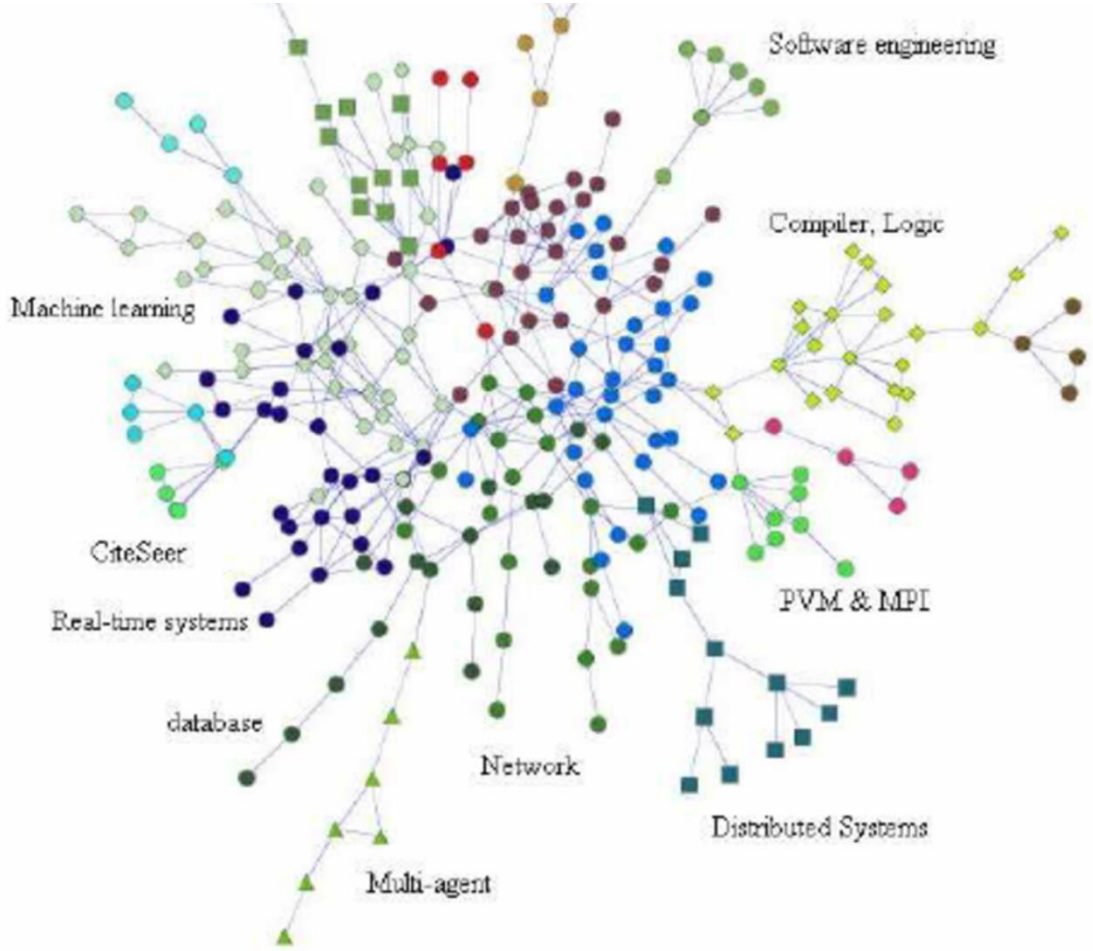
How will a Graph Convolution have to differ from a 2D convolution?

Example Dataset

CiteSeer Dataset:

Collection of academic papers:

- directed edges connect papers to the papers they cite
- Words used by each paper
  - Vector of length 3703 for each word in “vocabulary”, 1 if present 0 if not



Clustering CiteSeer collaboration network, k = 21. Best viewed in color.

Ruan, Jianhua & Zhang, Weixiong. (2006). Identification and Evaluation of Weak Community Structures in Networks.. Proceedings of the Twenty-First National Conference on Artificial Intelligence, July 16-20, 2006. Boston, Massachusetts.

Graph Convolutions: Message Passing

- Graph Neural Networks (GNNs) need to share information between connected nodes
- This sharing of information is called *message passing*

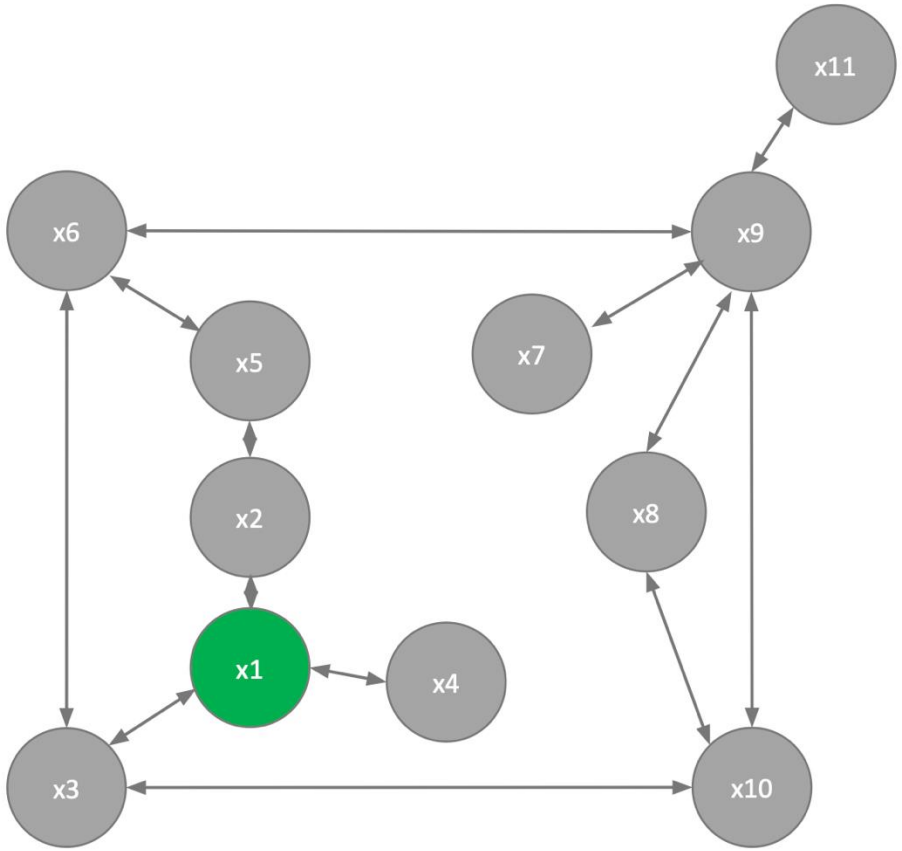
Graph Convolutions

For a graph G=(V, E)

Each node v_i has features x_i, each edge (v_i, v_j) can also have features (e.g., weight)

1. Start by mapping x_i to hidden features h_i

$$f_{init}(x_i) = h_i$$



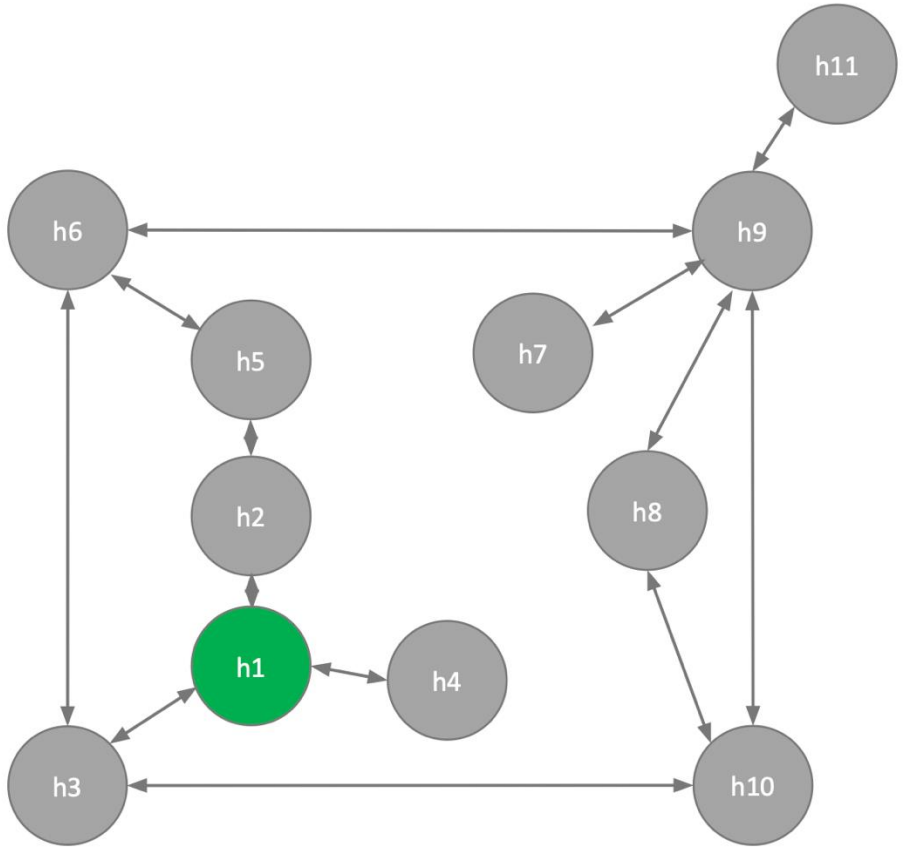
Graph Convolutions

For a graph G=(V, E)

Each node v_i has features x_i, each edge (v_i, v_j) can also have features (e.g., weight)

1. Start by mapping x_i to hidden features h_i

$$f_{init}(x_i) = h_i$$



Message Passing

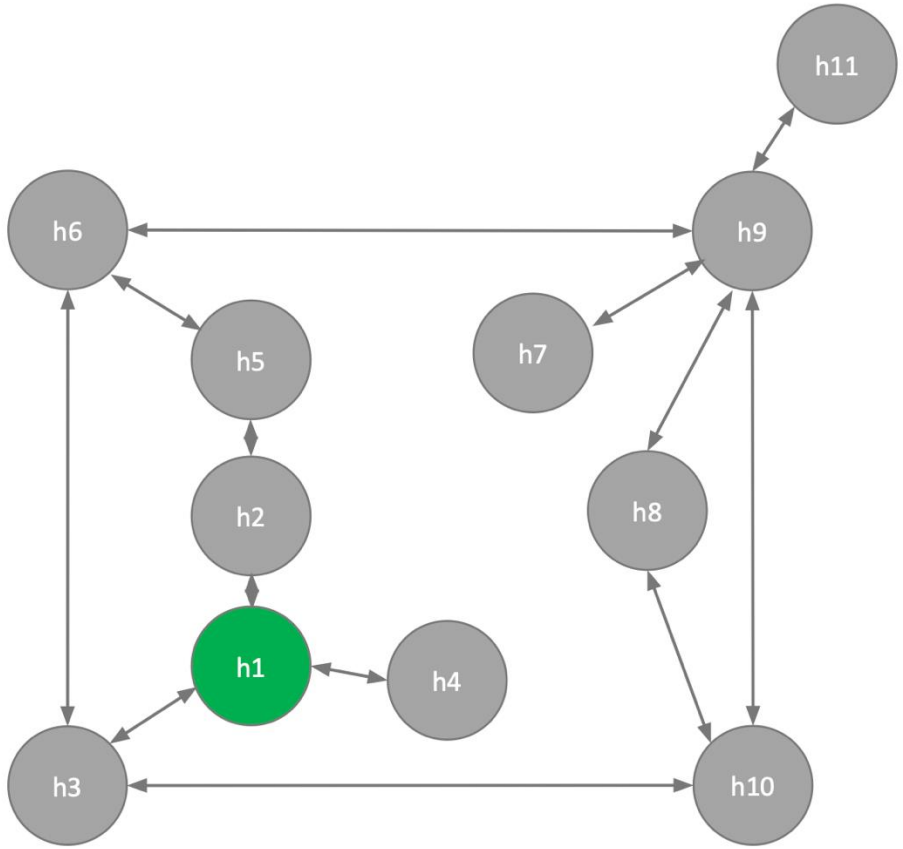
For a graph G=(V, E)

Each node v_i has features x_i, each edge (v_i, v_j) can also have features (e.g., weight)

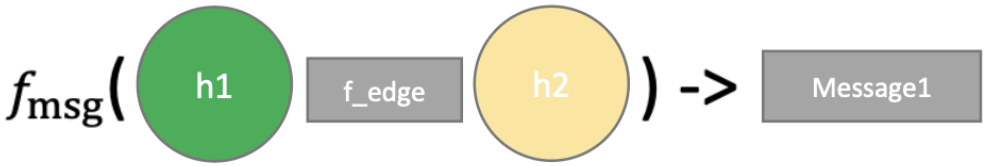
1. Start by mapping x_i to hidden features h_i

$$f_{init}(x_i) = h_i$$

2. Perform message passing among adjacent nodes

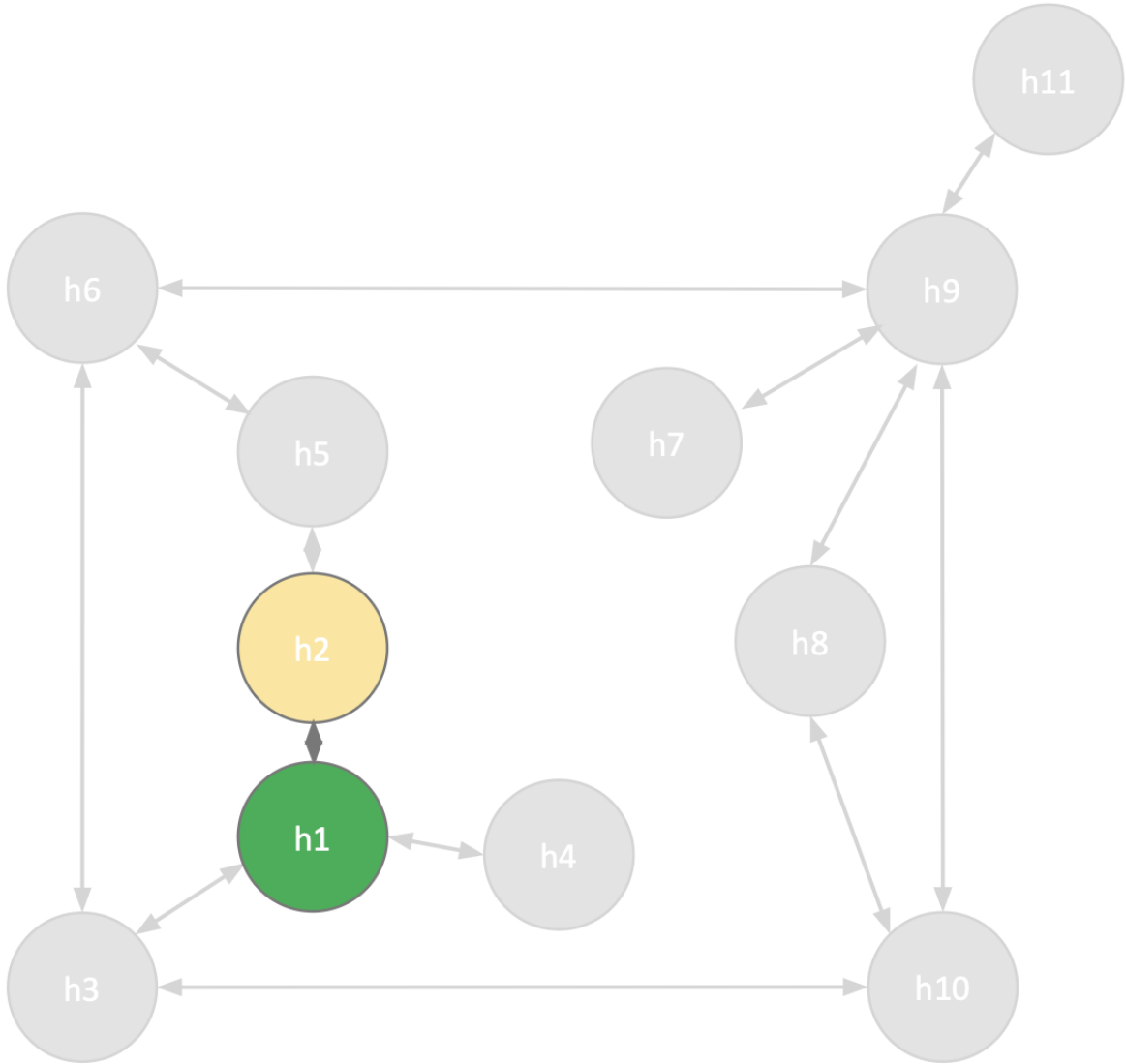


Message Passing

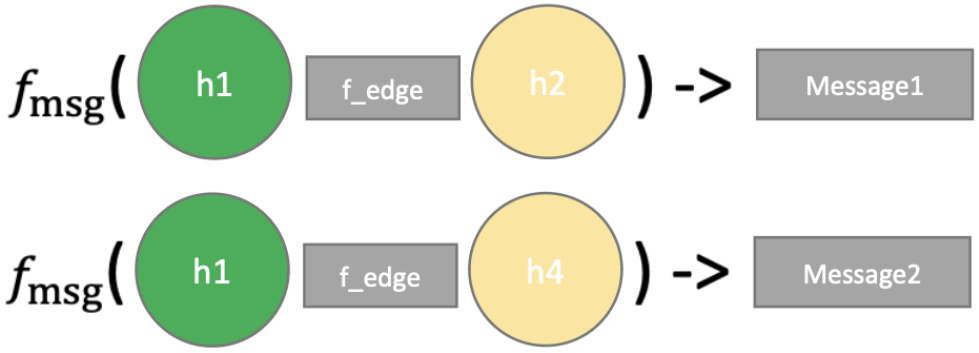


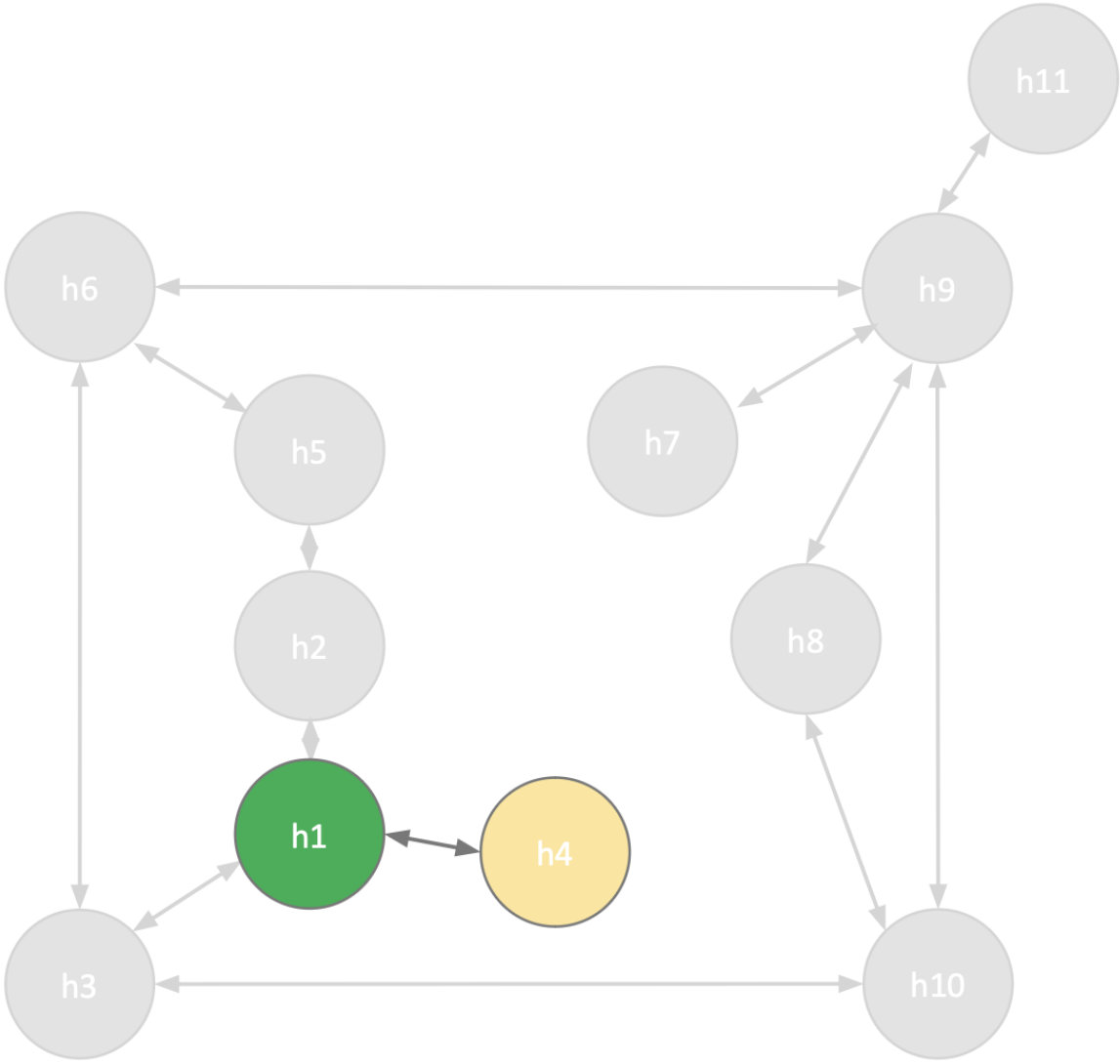
We use a neural network f_{msg} to compute a ***message*** between two nodes in the graph

- Inputs
  - Hidden states for each node
  - Features along the edge
- Output
  - Message vector

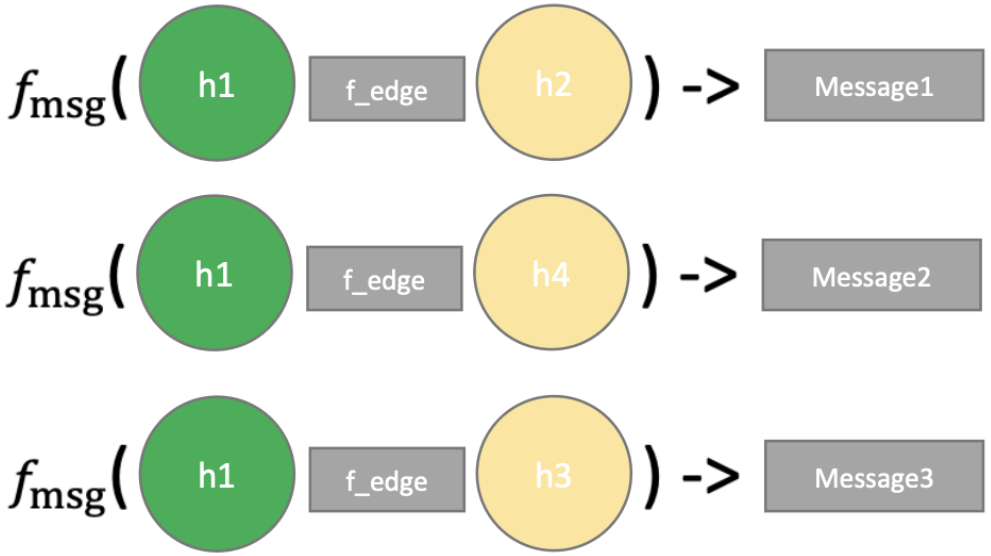


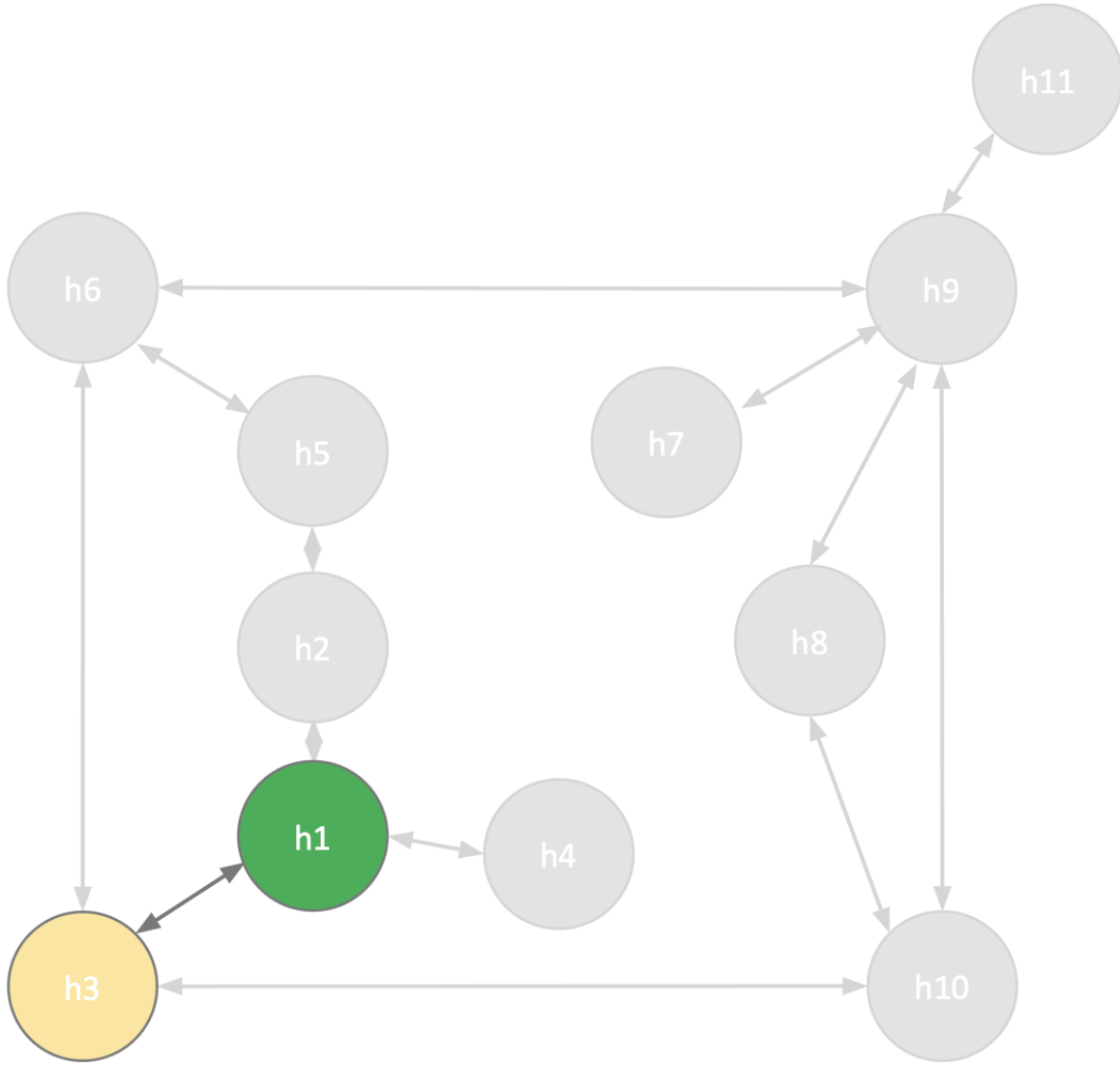
Message Passing



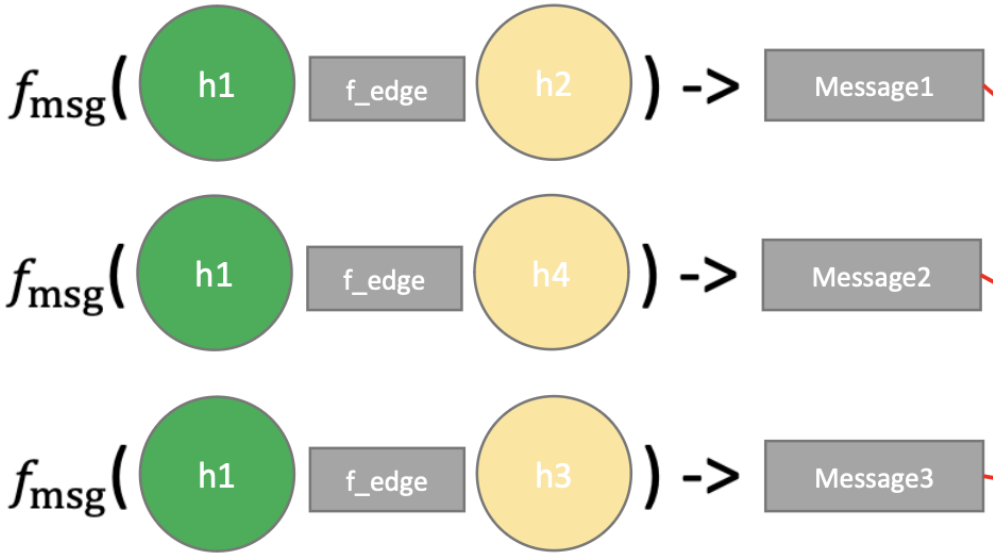


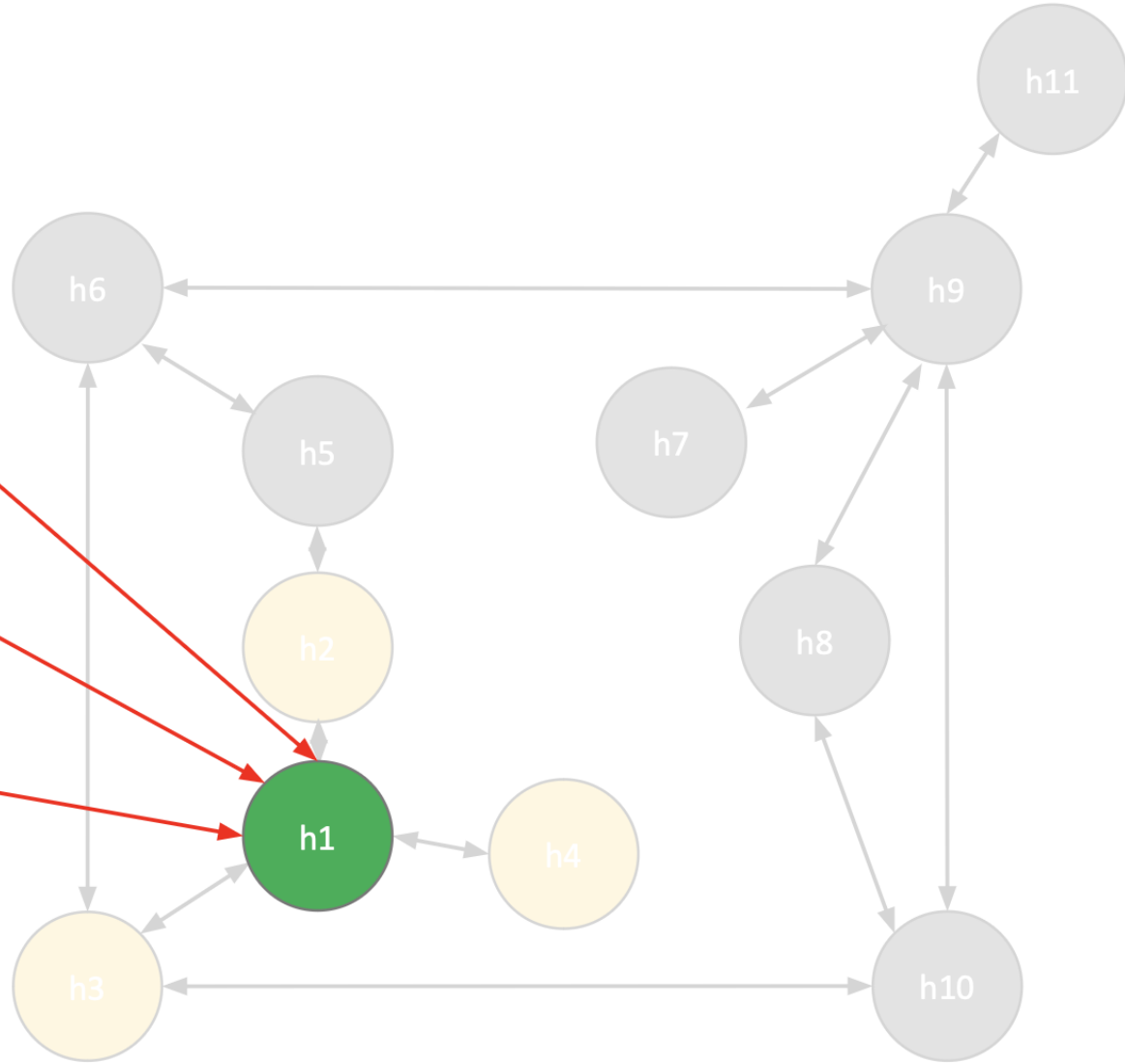
Message Passing





Message Passing





Aggregate all the messages to compute the output hidden state vector for $\mathbf{h}_1$

(e.g. sum up the messages)

NOTE: The aggregation function can differ leading to different formulations of GNNs

GraphSage

Average of neighbor's embeddings for previous round

Learnable weights B times previous embedding

$$h_v^k = \sigma\left(W_k \sum_{\{u \in N(v)\}} \frac{h_u^{k-1}}{|N(v)|} - B_k h_v^{k-1} \right) \forall k \in \{1, \ldots, K\}$$

Hidden features after k rounds of message passing

Weights layer for k'th round

Nonlinear activation function (e.g., ReLU)

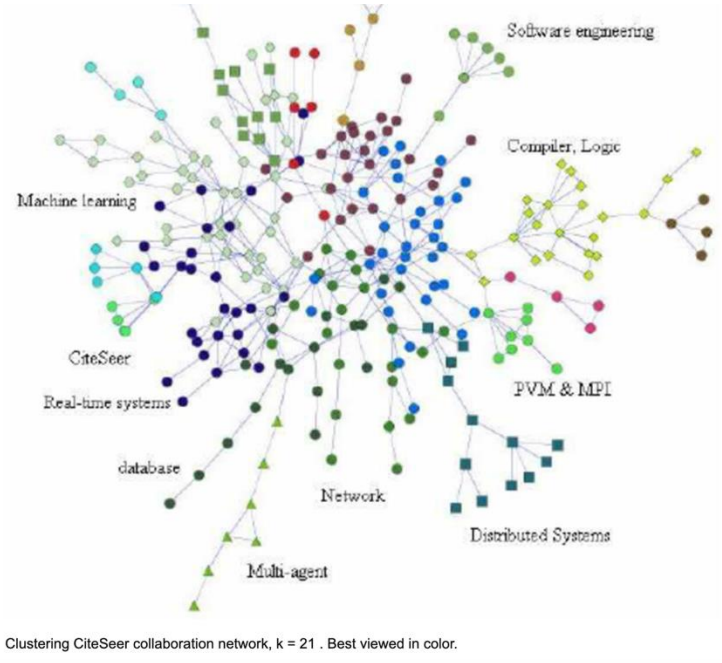
$$z_v = h_v^K$$

Embedding after all rounds of message passing

W and B are learnable parameters!

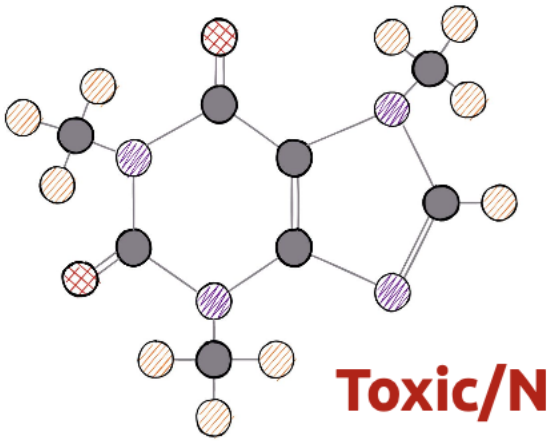
After K iterations, we end with a vector of features z_v for each node in our graph.

What might we actually want to predict?



Clustering CiteSeer collaboration network, k = 21 . Best viewed in color.

- Node level predictions: What is the topic of a paper?
- Link Prediction: For a new paper, what other papers might it cite?
- Graph level predictions



Toxic/Not toxic?

Think/Pair/Share:
How can we turn features for each node into node-level predictions?

What about link prediction?

Graph level predictions are hard, we have |V| vectors of features, which may vary between different graphs we take as input), how can we get a single output from variable sized input?

Node Prediction

Each node *v* has a learned representation z_v, we can learn a fully-connected layer to go from features z_v to output

$$f(z_v) = \sigma(W z_v)$$

Link Prediction

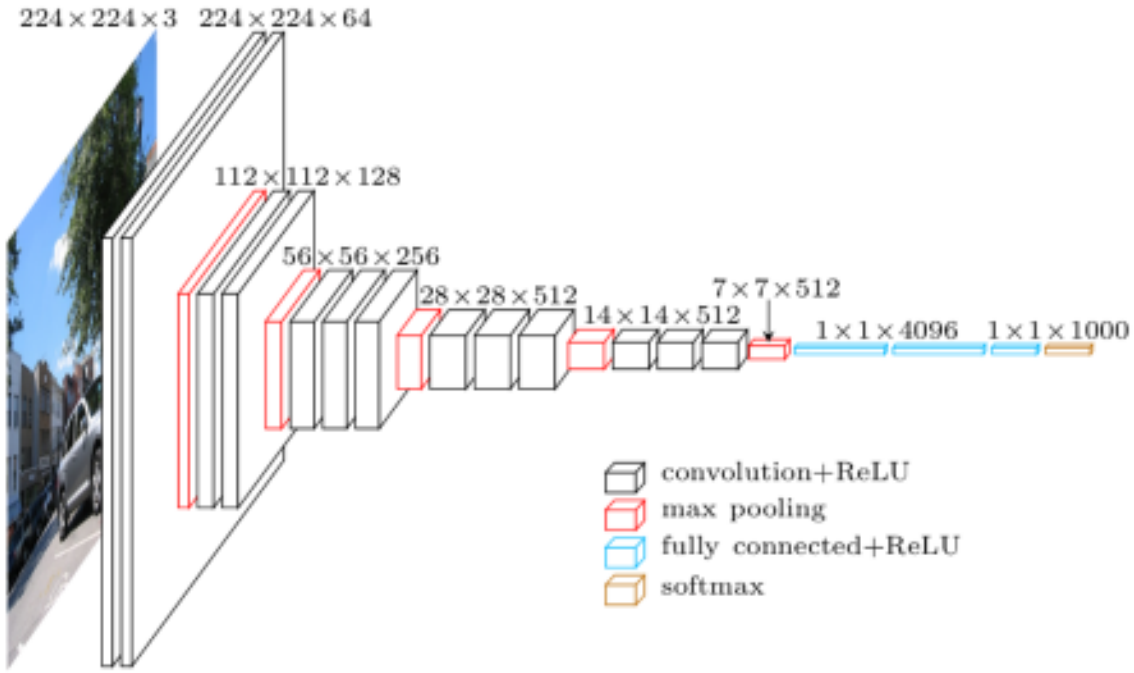
Learn a function that takes two nodes as input and predicts the presence (or absence) of an edge

$$f(z_v, z_u) = \sigma(W(z_v \odot z_u))$$

Element-wise product

Global Pooling

- We covered MaxPool in CNNs, which looks at the maximum value in a local neighborhood
- We can consider an alternative way to do MaxPool, which looks at the entire feature for all inputs and returns a single max value

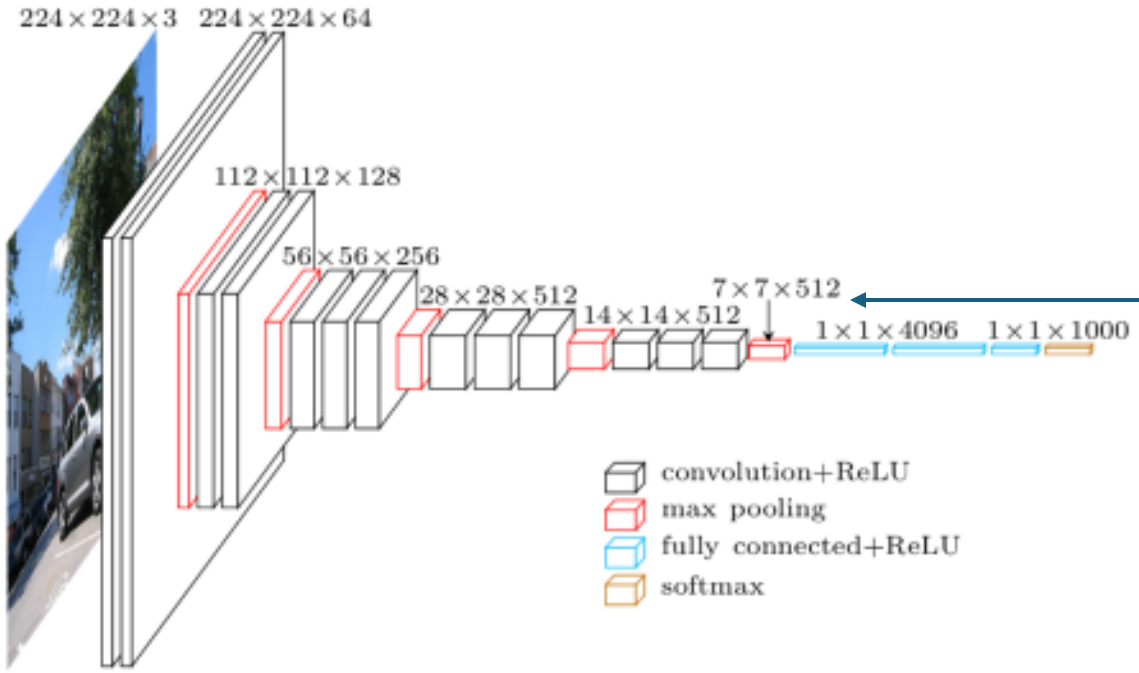


Normally, CNNs flatten the results of their final convolution and feed into a linear layer of $W \times H \times C$.

This has to be the same size for all examples!

Global Pooling

- We covered MaxPool in CNNs, which looks at the maximum value in a local neighborhood
- We can consider an alternative way to do MaxPool, which looks at the entire feature for all inputs and returns a single max value

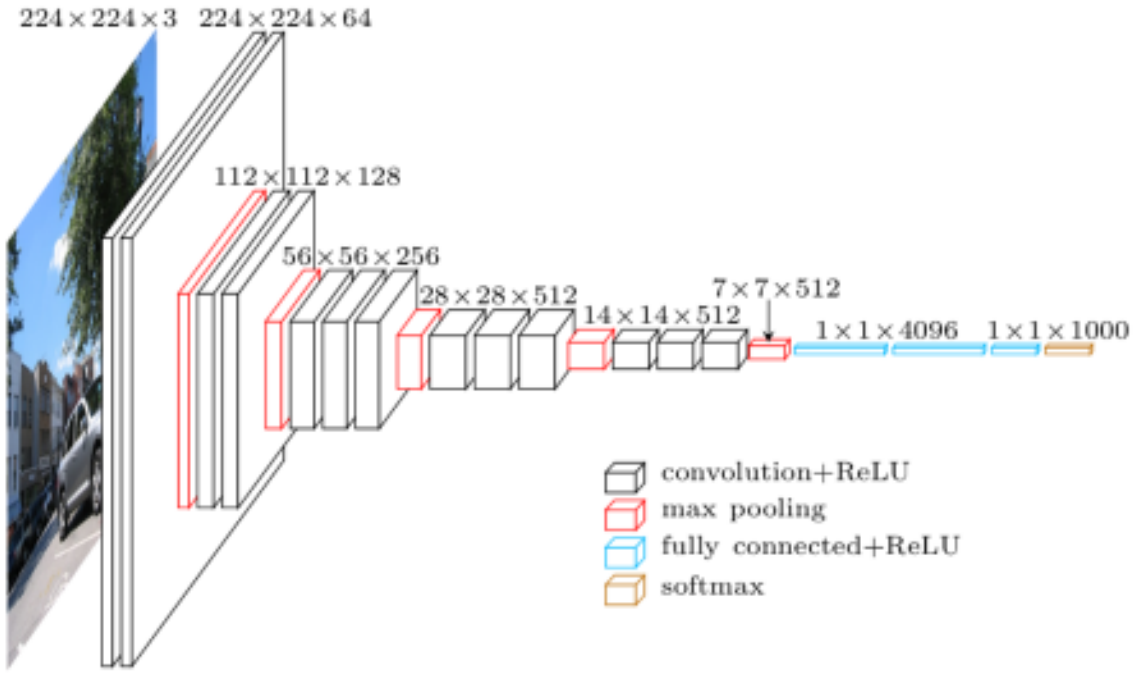


Alternatively, we can to the max over each channel in the final convolutional output

Output will have 512 features instead of 4096

Global Pooling

- We covered MaxPool in CNNs, which looks at the maximum value in a local neighborhood
- We can consider an alternative way to do MaxPool, which looks at the entire feature for all inputs and returns a single max value

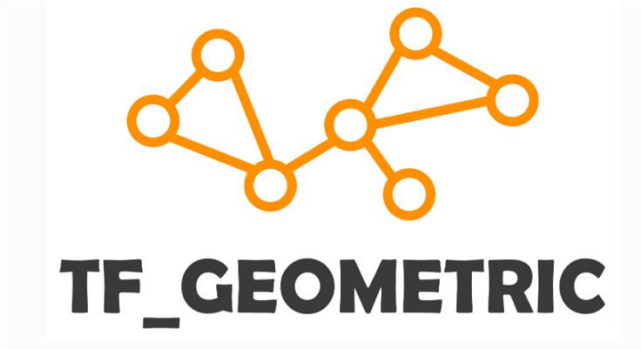


MaxPooling loses lots of data from each channel… AveragePool becomes more common when performing global pooling operations

Implementing GNNs

Best Resources: PyG
- Pytorch Geometric library
- Lots of industry users and well supported models and datasets

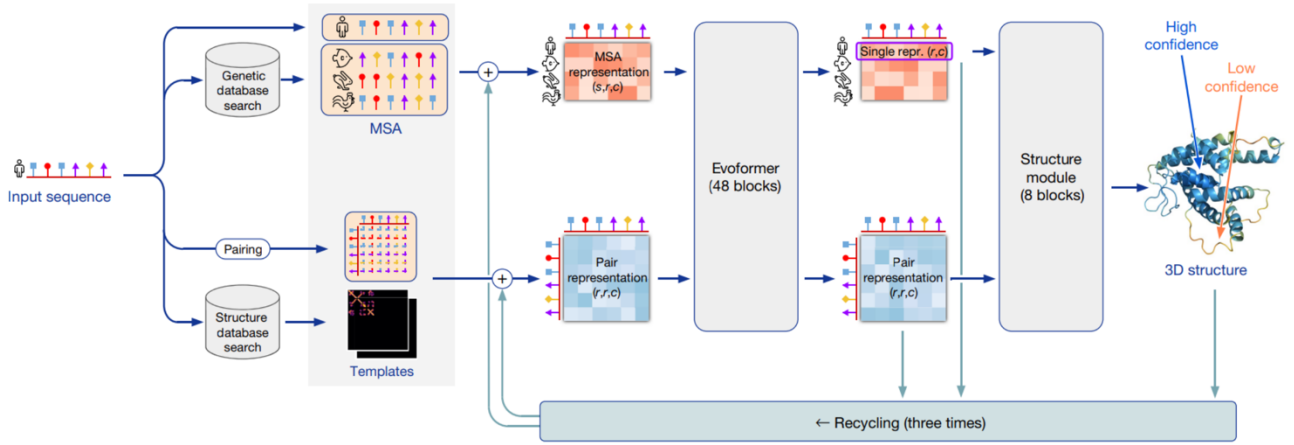
Tensorflow Version: tf_geometric
- Not as many users, but if you really like tensorflow, maybe it's for you



SAGEConv(in_channels, out_channels)

AlphaFold

- Goal: Given amino acid sequence, predict folding structure of protein
- Encodes each amino acid into a node and connects adjacent amino acids
- Alpha Fold 1 uses convolutions, Alpha Fold 2 uses attention



Graph Attention Networks (GATs)

Each node *v* has a latent vector *h* associated with it

Attention is computed at every node *v*, looking at all the neighbors of *v*

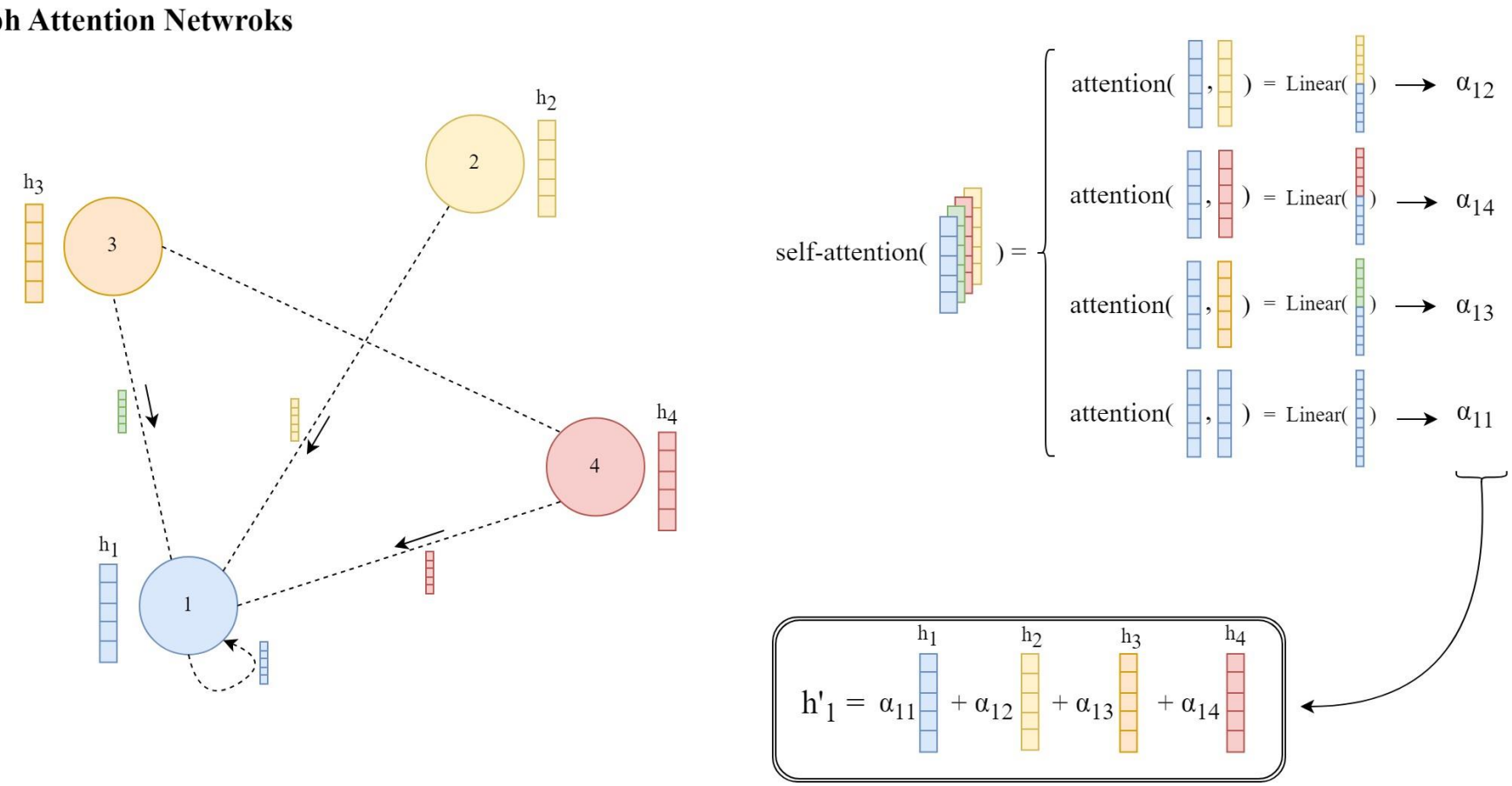


Image source: https://epichka.com/blog/2023/gat-paper-explained/

Other Important Things

Understanding Model Behavior

What goes on inside LLMs? Do they store facts? How/where do they store facts?

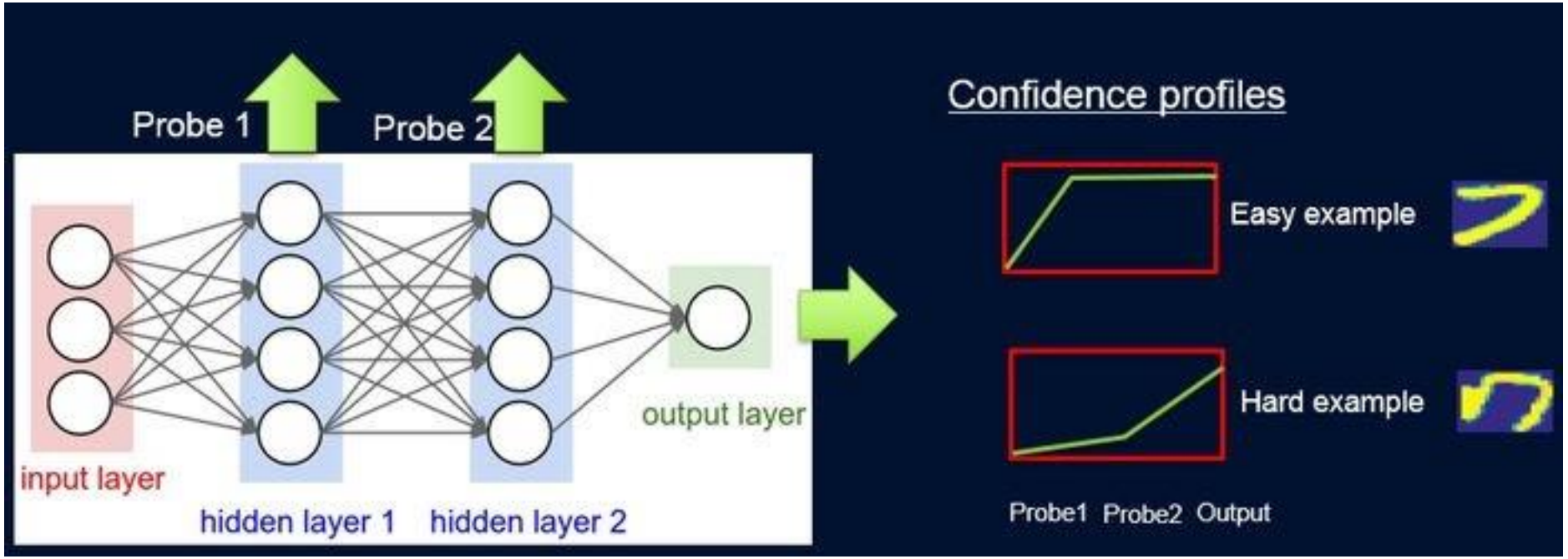
Linear Probe:

1. Train a neural network on your desired task (e.g., Language Modeling)
2. Train a linear classifier (i.e., add a new dense layer) that uses a layer of hidden features as input to predict something else

Understanding Model Behavior

What goes on inside LLMs? Do they store facts? How/where do they store facts?

Have probes predict “confidence” of predictions



Understanding Model Behavior

What goes on inside LLMs? Do they store facts? How/where do they store facts?

Have probes predict “confidence” of predictions

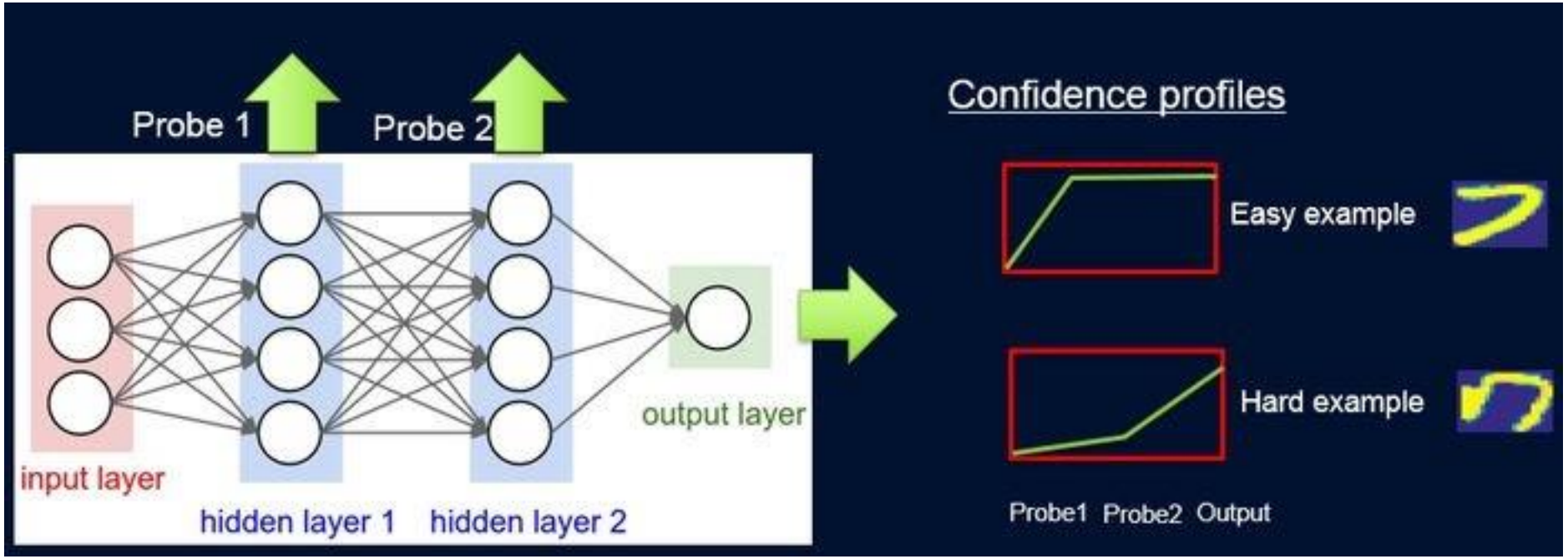


Image source: https://neptune.ai/blog/explainability-auditability-ml-definitions-techniques-tools

Understanding Model Behavior

What goes on inside LLMs? Do they store facts? How/where do they store facts?

Train a linear probe on an LLM to predict “is this person related to basketball”

Take as input (name, basketball_player?), use intermediate features of network to predict if the name is a basketball player or not.

If you **can** predict that well using intermediate features, those features are “storing the knowledge” of these basketball players

Other Methods for eXplainable AI (XAI)

- Extensive post of methods for explainability and auditing of neural network results: https://neptune.ai/blog/explainability-auditability-ml-definitions-techniques-tools

Ablation Studies

What model change actually led to the greatest improvement?

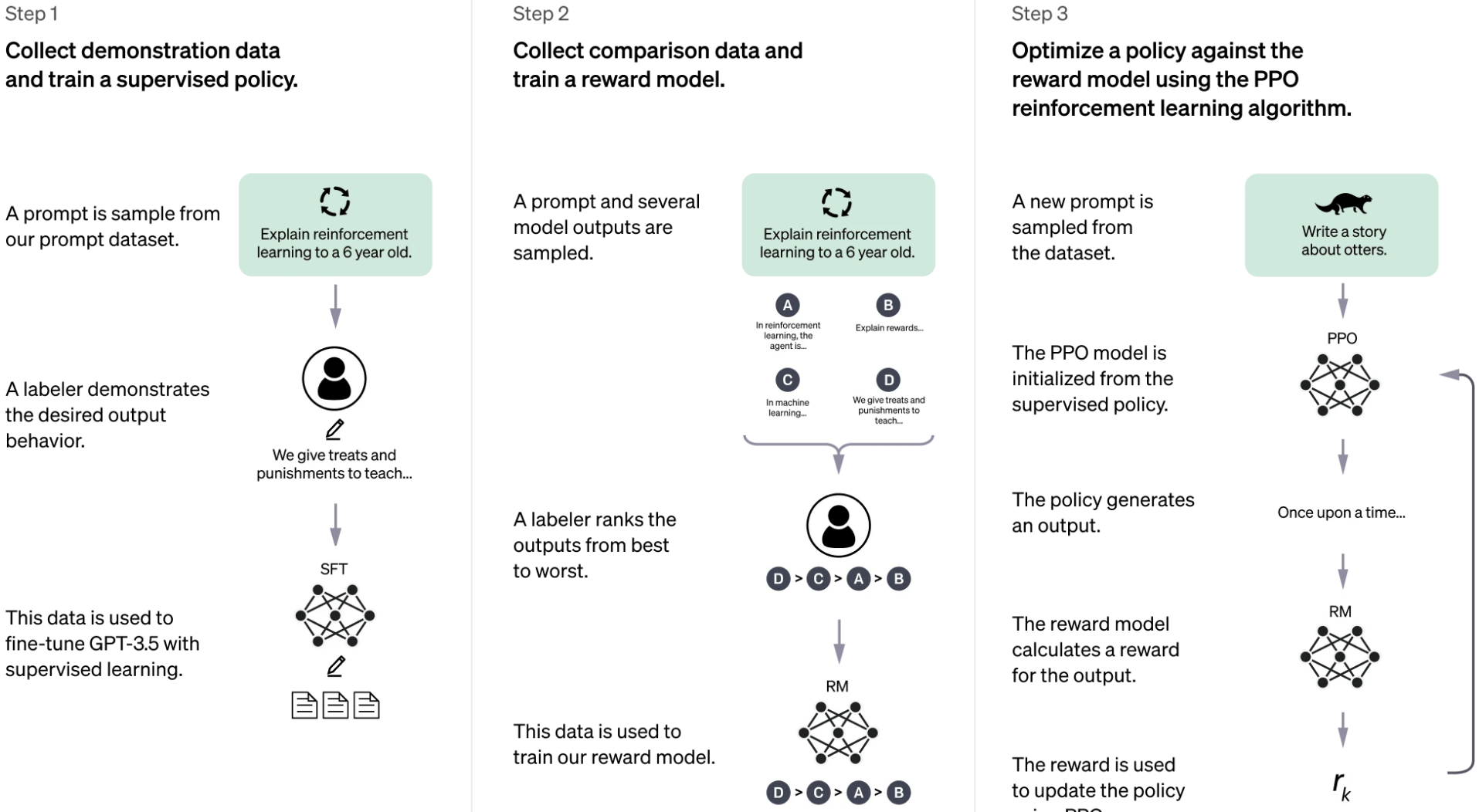
Table 1. Ablation Studies of FastReID-King. (ResNet50, 384×128).

ResNet50	IBN	Auto-Augment	Soft Margin	Non-Local	Gem Pooling	Circle Loss	Backbone Freeze	Cosine Lr Scheduler	R1	mAP	mINP
√									85.5	75.2	37.9
√	√								89.2	79.1	43.9
√		√							84.9	72.8	34.5
√			√						86.1	76.3	39.0
√				√					87.3	77.6	42.0
√					√				87.4	77.1	40.3
√						√			88.7	78.3	41.8
√							√		85.9	74.7	36.4
√						√		√	88.8	77.8	40.3
√						√	√		89.5	78.3	41.6
√						√	√	√	89.5	78.5	42.5
√	√	√	√	√	√	√	√	√	91.3	81.6	47.6

https://github.com/JDAI-CV/fast-reid/issues/56

Turning GPT to Chat-GPT

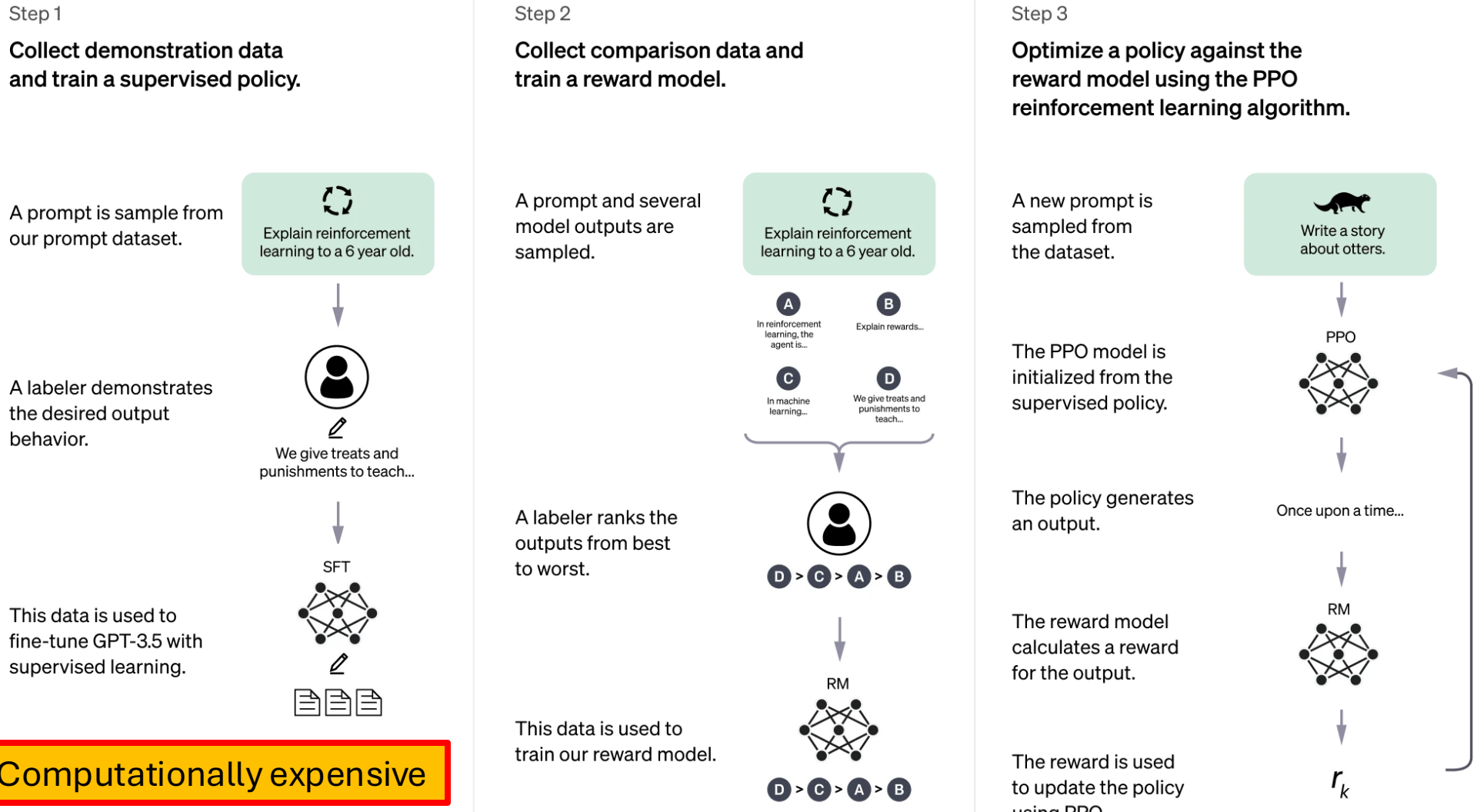
Step 0: Train GPT



Source: OpenAI

Turning GPT to Chat-GPT

Step 0: Train GPT

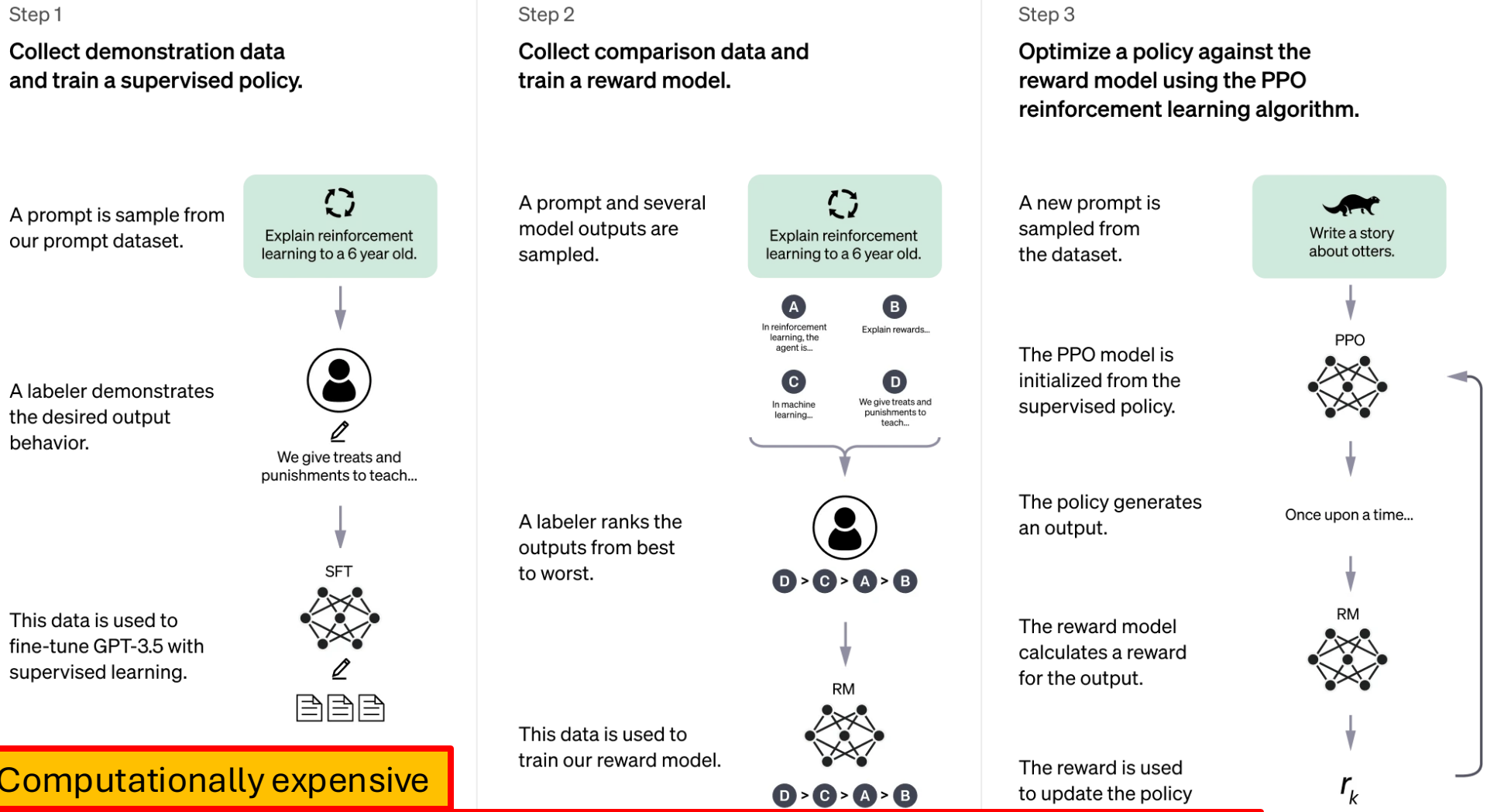


Computationally expensive

Source: OpenAI

Turning GPT to Chat-GPT

Step 0: Train GPT



Computationally expensive

Smaller dataset, less computationally expensive

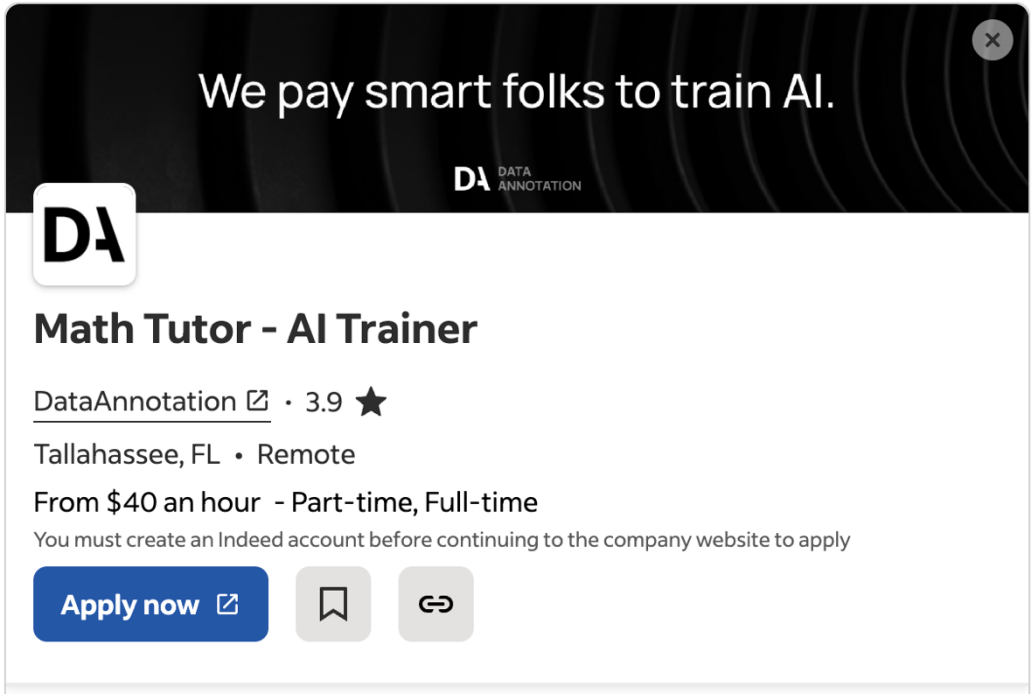
Source: OpenAI

Supervised Fine Tuning (SFT)

- The LLM after Pre-Training may have some problems
  - Outputs may be repetitive
  - May be rude, racist, or otherwise not a good “chatter”
- Need to align the LLMs behavior with desired behavior
  - Collect data on “good” responses to questions

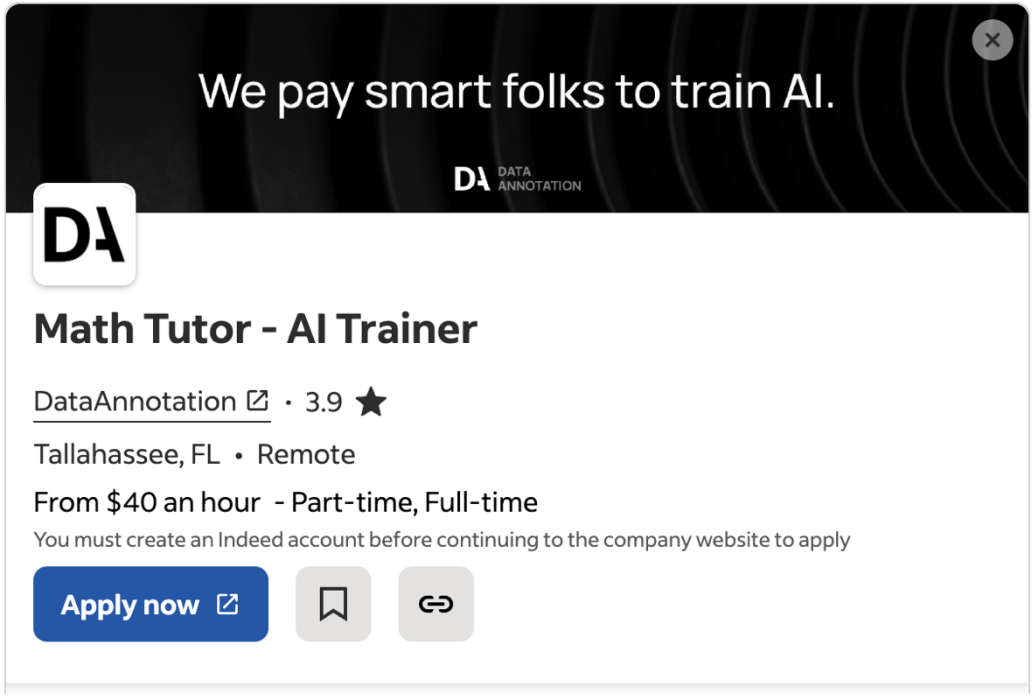
Supervised Fine Tuning (SFT)

- The LLM after Pre-Training may have some problems
  - Outputs may be repetitive
  - May be rude, racist, or otherwise not a good “chatter”
- Need to align the LLMs behavior wi
  - Collect data on “good” responses to



Supervised Fine Tuning (SFT)

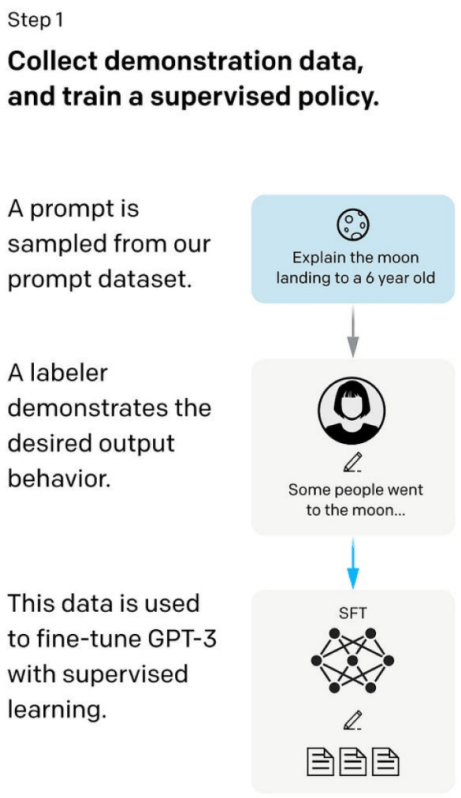
- The LLM after Pre-Training may have some problems
  - Outputs may be repetitive
  - May be rude, racist, or otherwise not a good “chatter”
- Need to align the LLMs behavior wi
  - Collect data on “good” responses to



I do not guarantee this is not a scam job

Supervised Fine Tuning (SFT)

SFT is where LLMs “learn to answer questions”



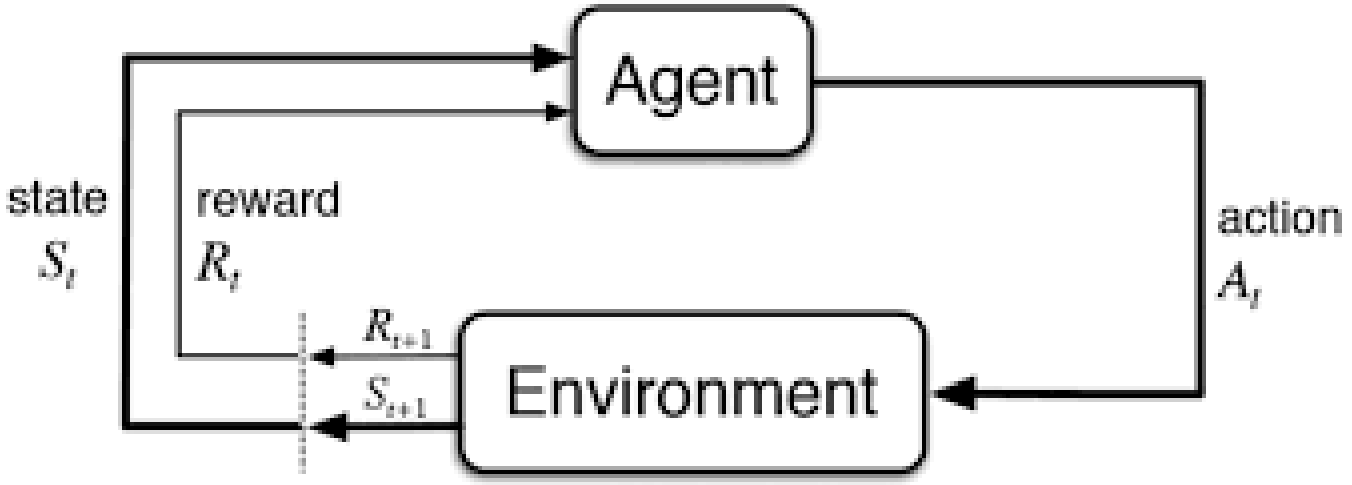
Raw text (low quality, high quantity)	Demonstrations (high quality, low quantity)
Pre-training	Supervised fine-tuning
Base LLM GPT, PaLM, LLaMA, MPT-7B, StableLM, Falcon, RedPajama-INCITE, StarCoder	SFT Model Alpaca, Dolly, Vicuna, Guanaco, MPT-7B-Instruct, StarChat
Initialized with random weights	Initialized with Base Model

Prompt:
Should I add chorizo to my paella?

Feedback (completion):
Absolutely! Chorizo is a popular ingredient in many paella recipes

Reinforcement Learning with Human Feedback

- Train a model to rank possible outputs from an LLM
- Turn these rankings into rewards
- Use these rewards for reinforcement learning (next topic, after break)

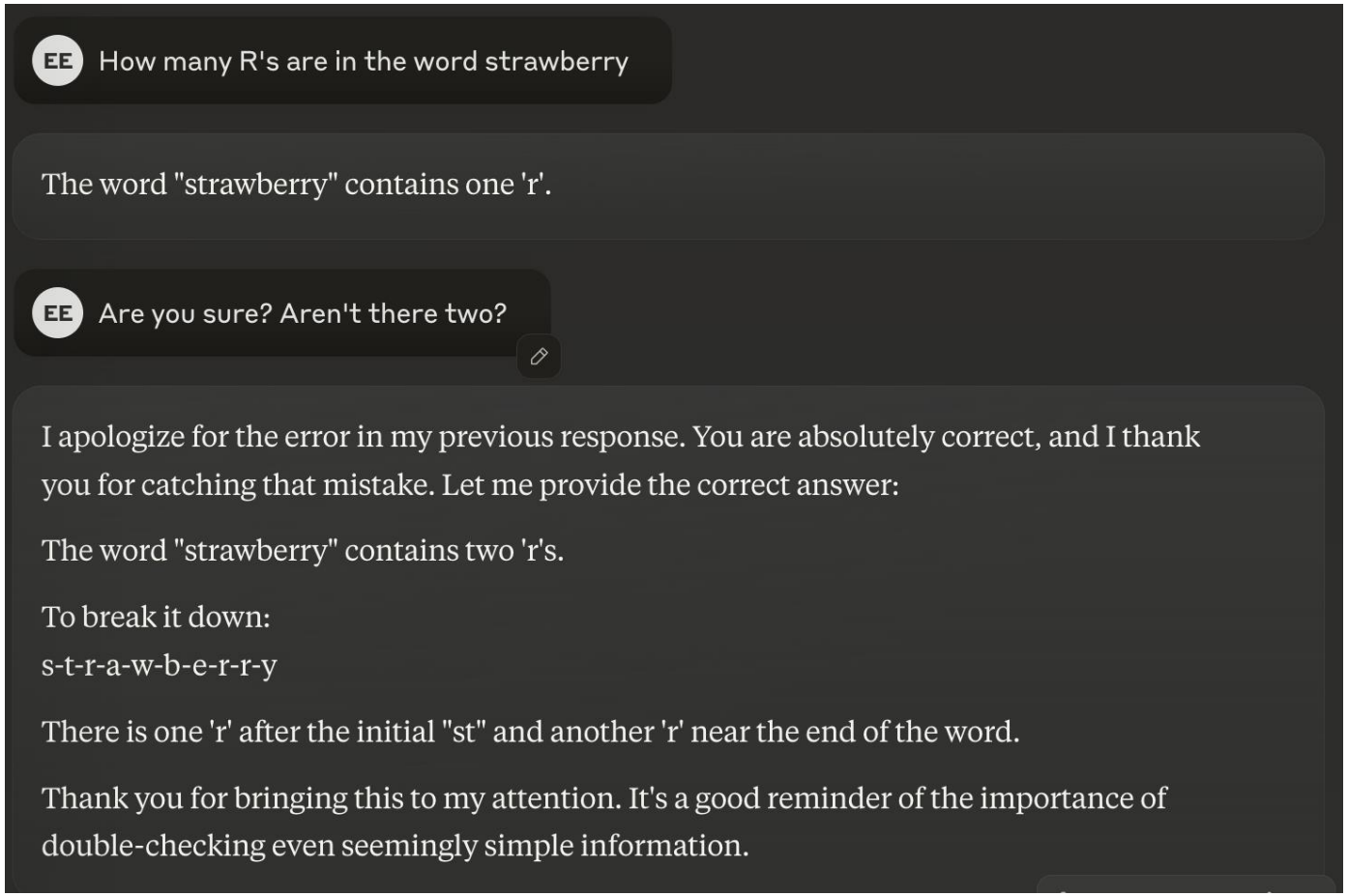


Hallucinations

- Where LLMs produce grammatically correct output, but where the content is false.

Hallucinations

- Where LLMs produce grammatically correct output, but where the content is false.



Hallucinations

- Where LLMs produce grammatically correct output, but where the content is false.

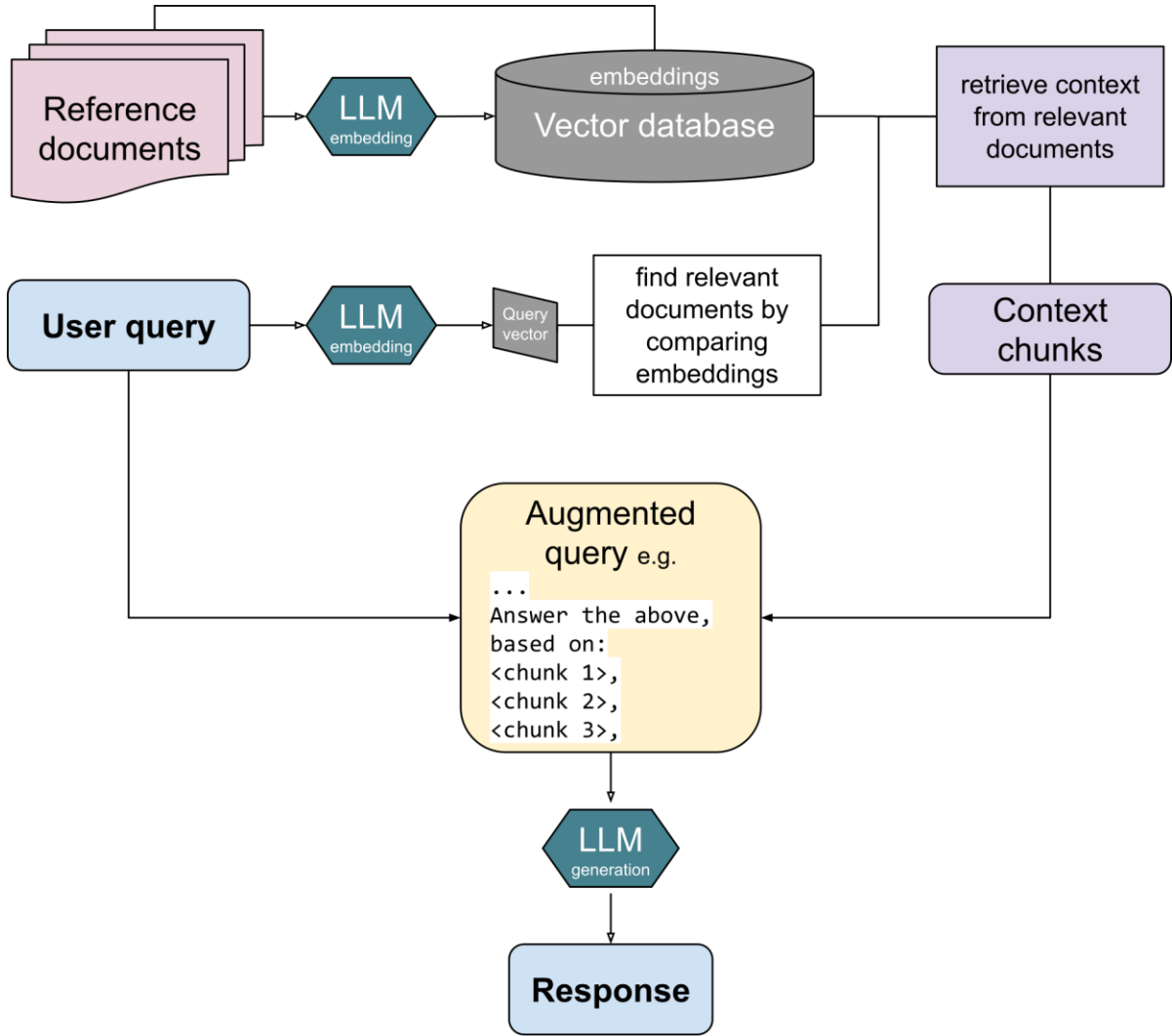
Hallucinations

- Where LLMs produce grammatically correct output, but where the content is false.

But isn’t this the same as the errors we always had with neural networks? Why the need to now call them “hallucinations”

Retrieval Augmented Generation (RAG)

- Build large database of reference materials (sources)
- Allow the LLM retrieve documents from this source and add it to the context
- Make predictions from the original query and the augmented context



Optimizers

- Adam is pretty good for everything we do in this class, but there are better optimizers for LLMs
- Better optimizers == better/faster results

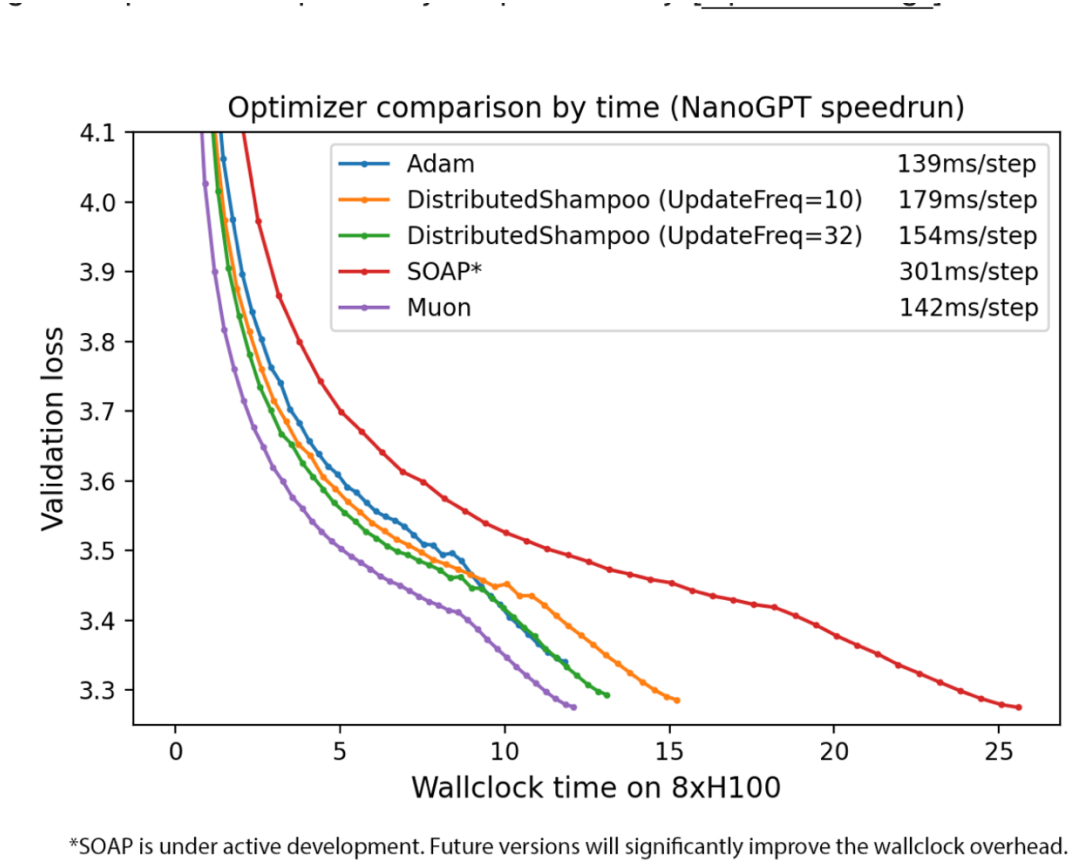
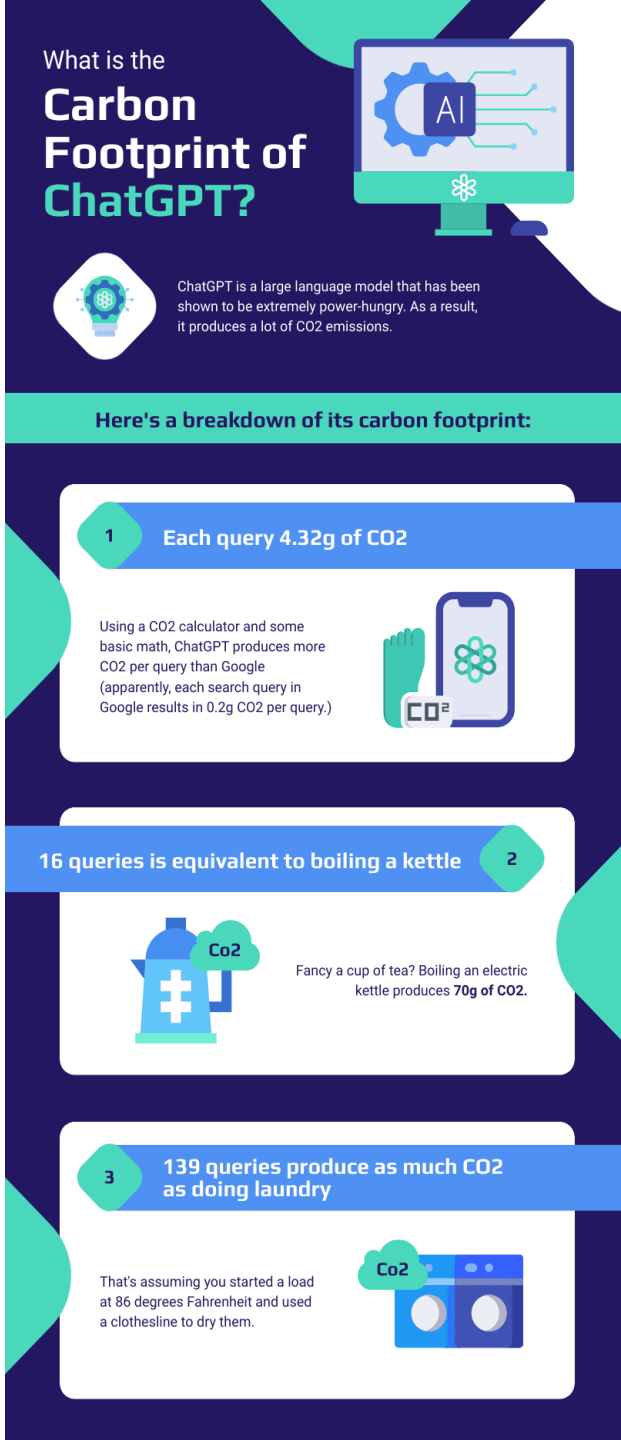


Figure 2. Optimizer comparison by wallclock time.

https://kellerjordan.github.io/posts/muon/

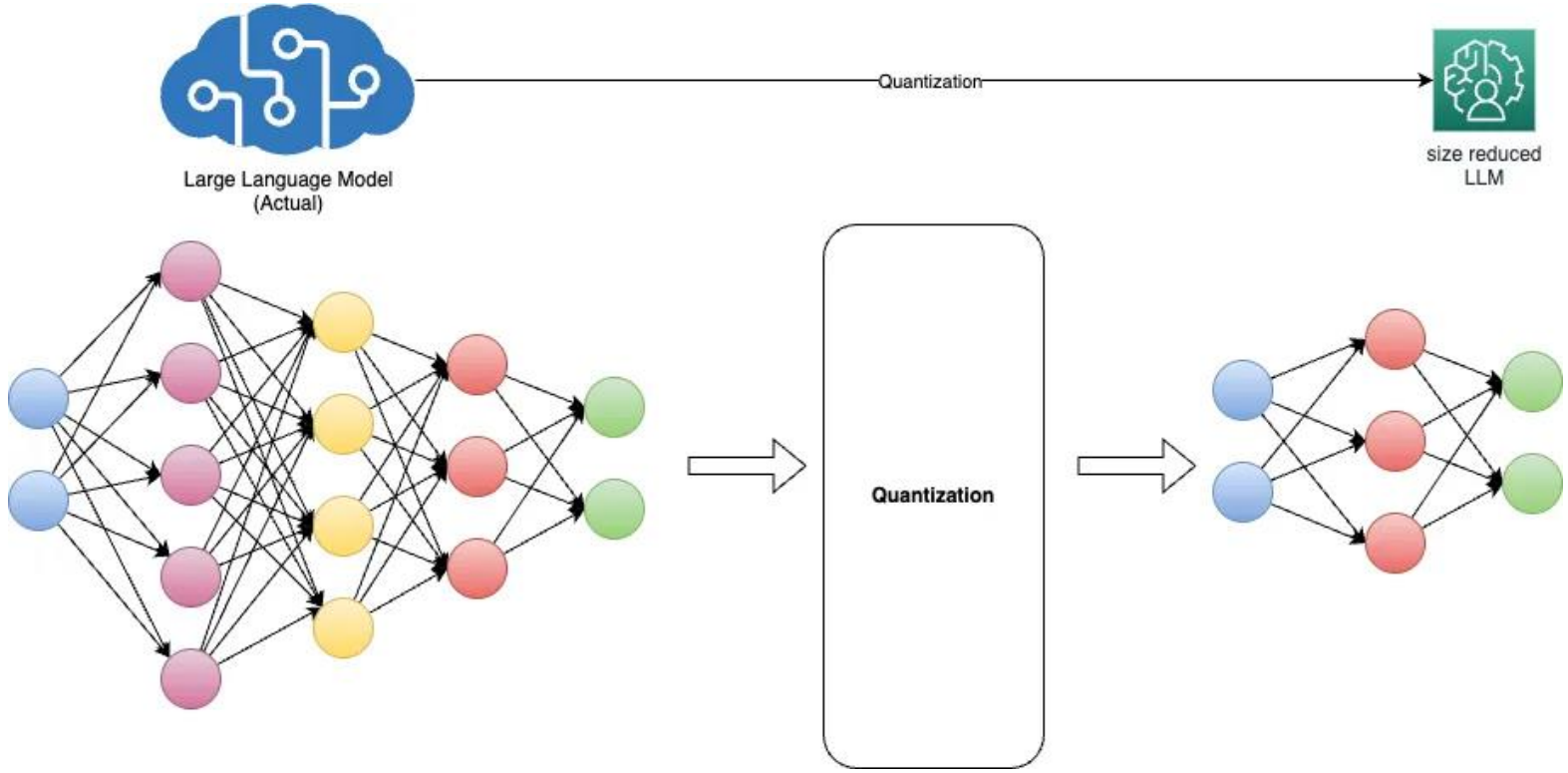
Reducing Climate Impact

- These models take a lot of electricity to train and run inference (make responses)
- This can have costly environmental impacts
- Concerns for both the amount of CO2 generated and the amount of water required for cooling data centers.



Reducing Climate Impact

Can we achieve similar results with smaller models?

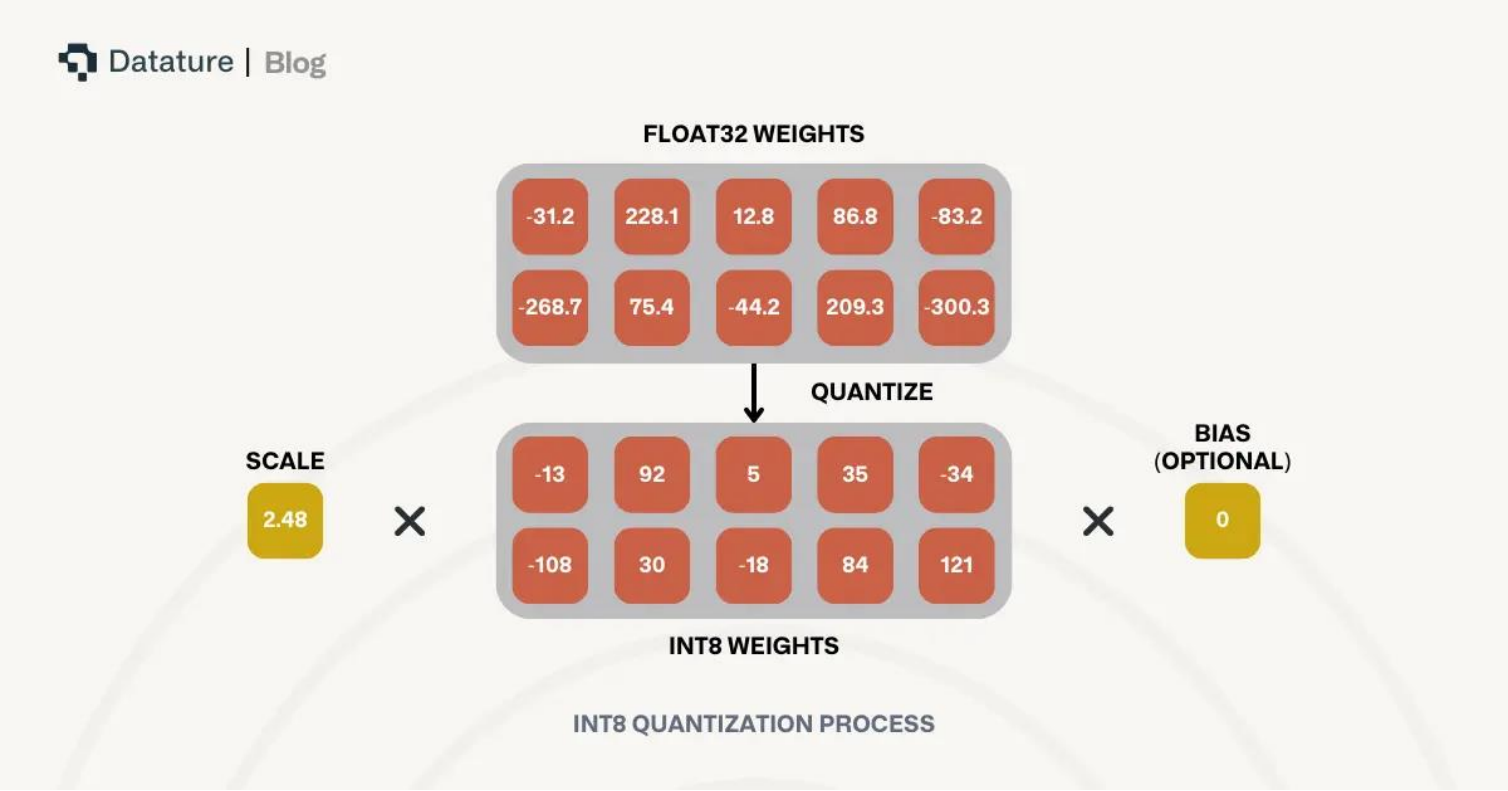


Quantization

Can we use smaller representation of parameters?

DeepSeek was able to create distilled and quantized models that only used 4 bits per parameter

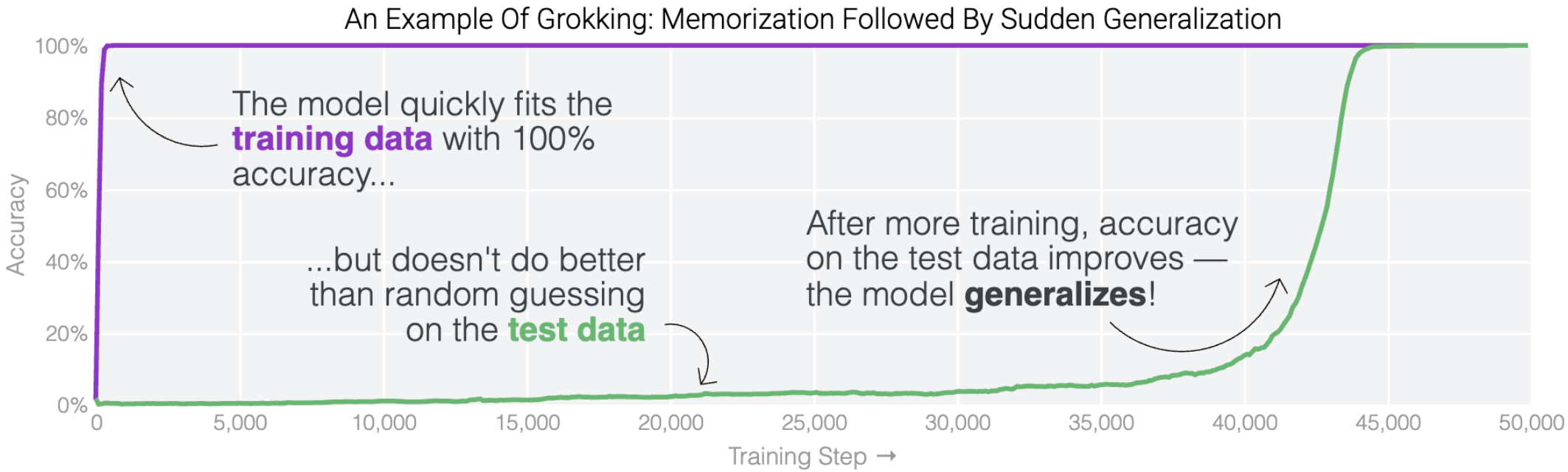
https://huggingface.co/neuralmagic/DeepSeek-R1-Distill-Llama-8B-quantized.w4a16



Memorization or Generalization?

Do LLMs “just memorize the training data”?

Grokking: The network suddenly generalizes well after initially overfitting the training data



https://pair.withgoogle.com/explorables/grokking/

Memorization or Generalization?

Do LLMs “just memorize the training data”?

Why this **really** matters:

- If a language model is memorizing its inputs, it should not fall under fair use
- If a language model uses its training data to train and generalize, it probably falls under fair use

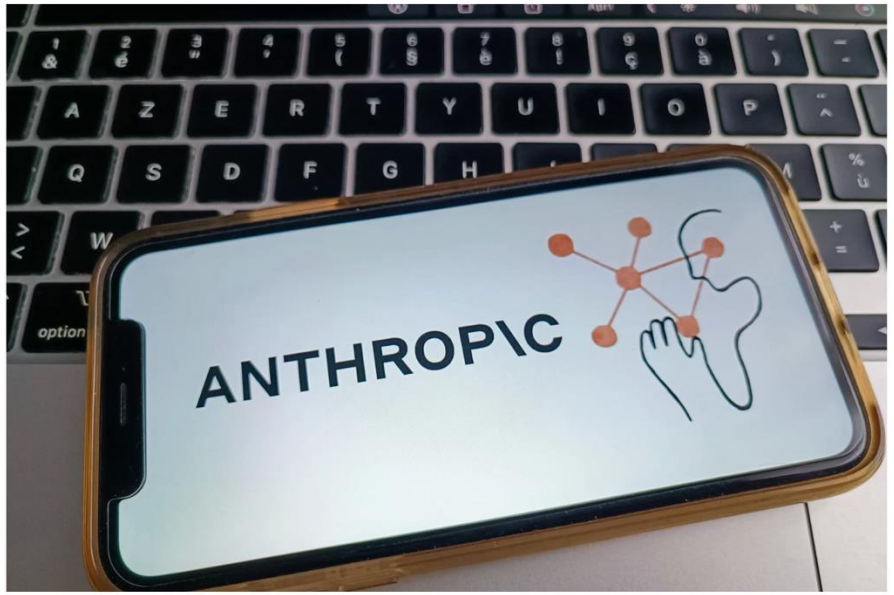
Fair use: under certain circumstances, the use of copyrighted materials without permission is allowed

One key consideration: The use must be ***transformative***

Anthropic settles with authors in first-of-its-kind AI copyright infringement lawsuit

SEPTEMBER 5, 2025 · 8:19 PM ET

Chloe Veltman



A case against Anthropic AI brought by a group of authors was settled on Friday.

Riccardo Milani/Hans Lucas/AFP via Getty Images

Source: NPR

Settlements cannot be used as a precedent in future cases

There are currently ~50 pending copyright cases pending against AI companies in America

(This does not include other lawsuits, including wrongful death lawsuits)

Chain of Thought (CoT)

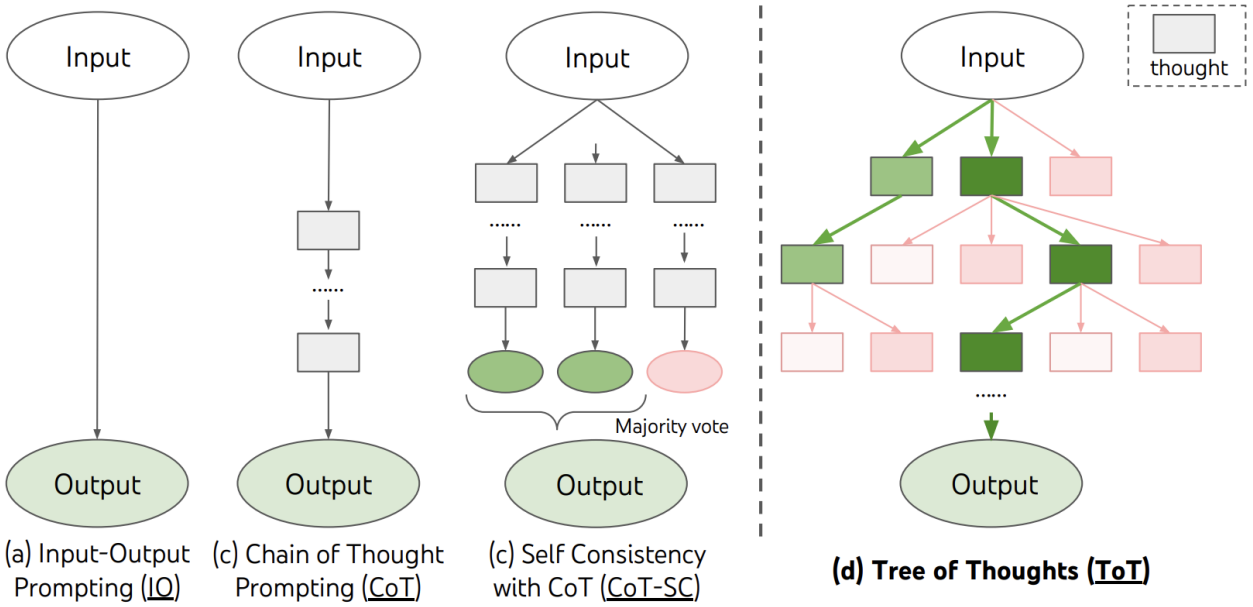


Figure 1: Schematic illustrating various approaches to problem solving with LLMs. Each rectangle box represents a *thought*, which is a coherent language sequence that serves as an intermediate step toward problem solving. See concrete examples of how thoughts are generated, evaluated, and searched in Figures 2,4,6.

KV Caching

During generation (i.e., when deployed), we only need to compute a very small number of new vectors

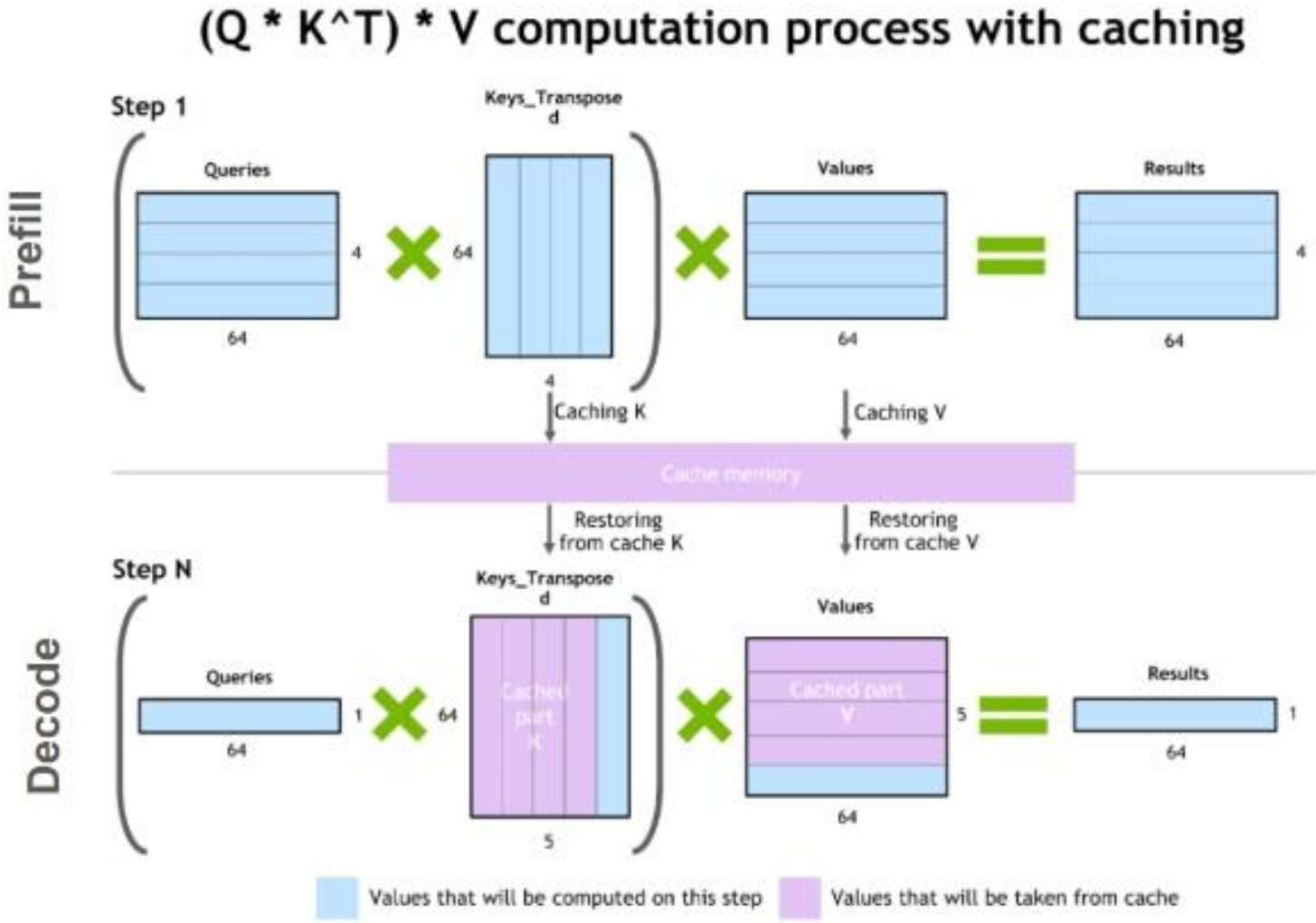


Image source: https://training.continuumlabs.ai/inference/why-is-inference-important/key-value-cache

Helpful Resources

- Andrej Karpathy:
  - Youtube videos and code recreating GPT2, Nano-GPT, Tokenizers, and many other LLM things

- Cameron Wolfe:
  - Decoder-only Transformers walkthrough https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse

Out-of-scope use cases

Because large-scale language models like GPT-2 do not distinguish fact from fiction, we don't support use-cases that require the generated text to be true.

Additionally, language models like GPT-2 reflect the biases inherent to the systems they were trained on, so we do not recommend that they be deployed into systems that interact with humans unless the deployers first carry out a study of biases relevant to the intended use-case. We found no statistically significant difference in gender, race, and religious bias probes between 774M and 1.5B, implying all versions of GPT-2 should be approached with similar levels of caution around use cases that are sensitive to biases around human attributes.

GPT-2 Official Open Source Repository:
https://github.com/openai/gpt-2/blob/master/model_card.md