Machine Learning in High Energy Physics

Connor Waits

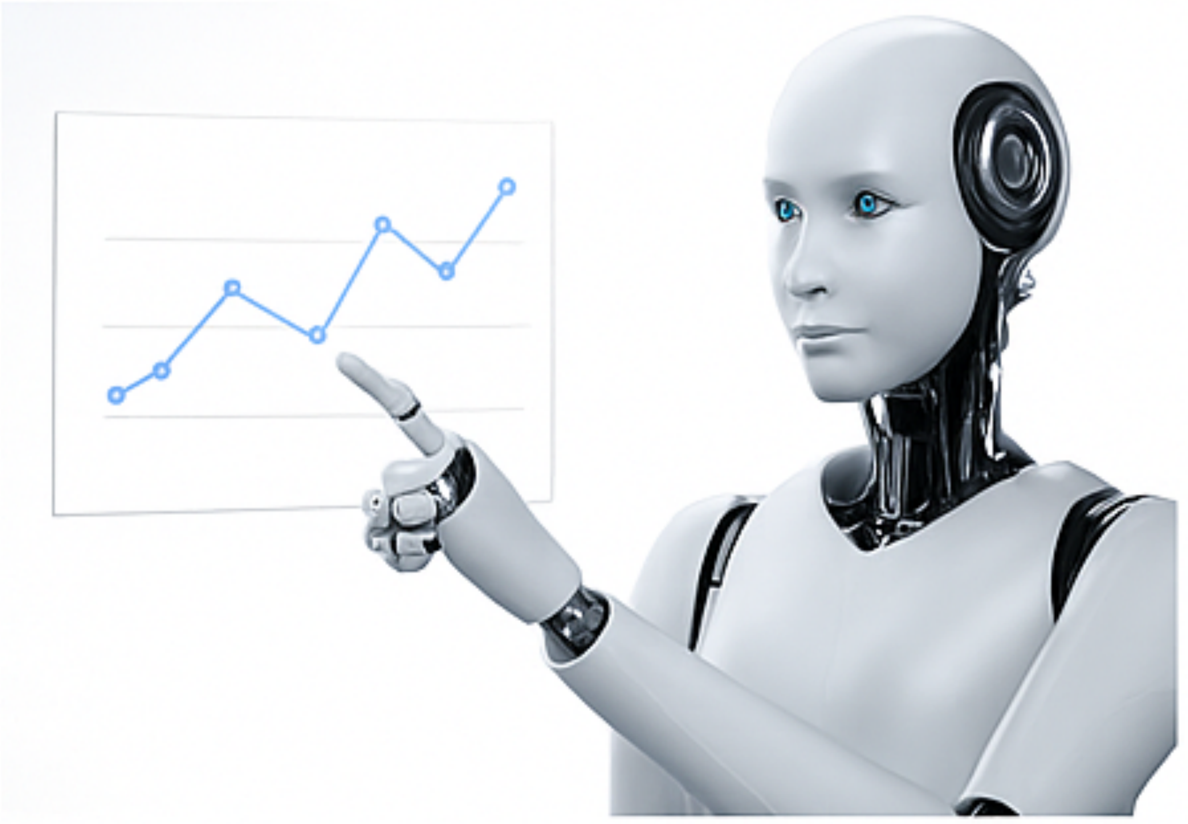
7/9/25

So I asked ChatGPT to make a presentation about AI…

Sure! Here’s a simple slide on artificial intelligence:

Artificial Intelligence

- AI is changing the world!
- Machine learning is a type of AI.
- Robots are cool

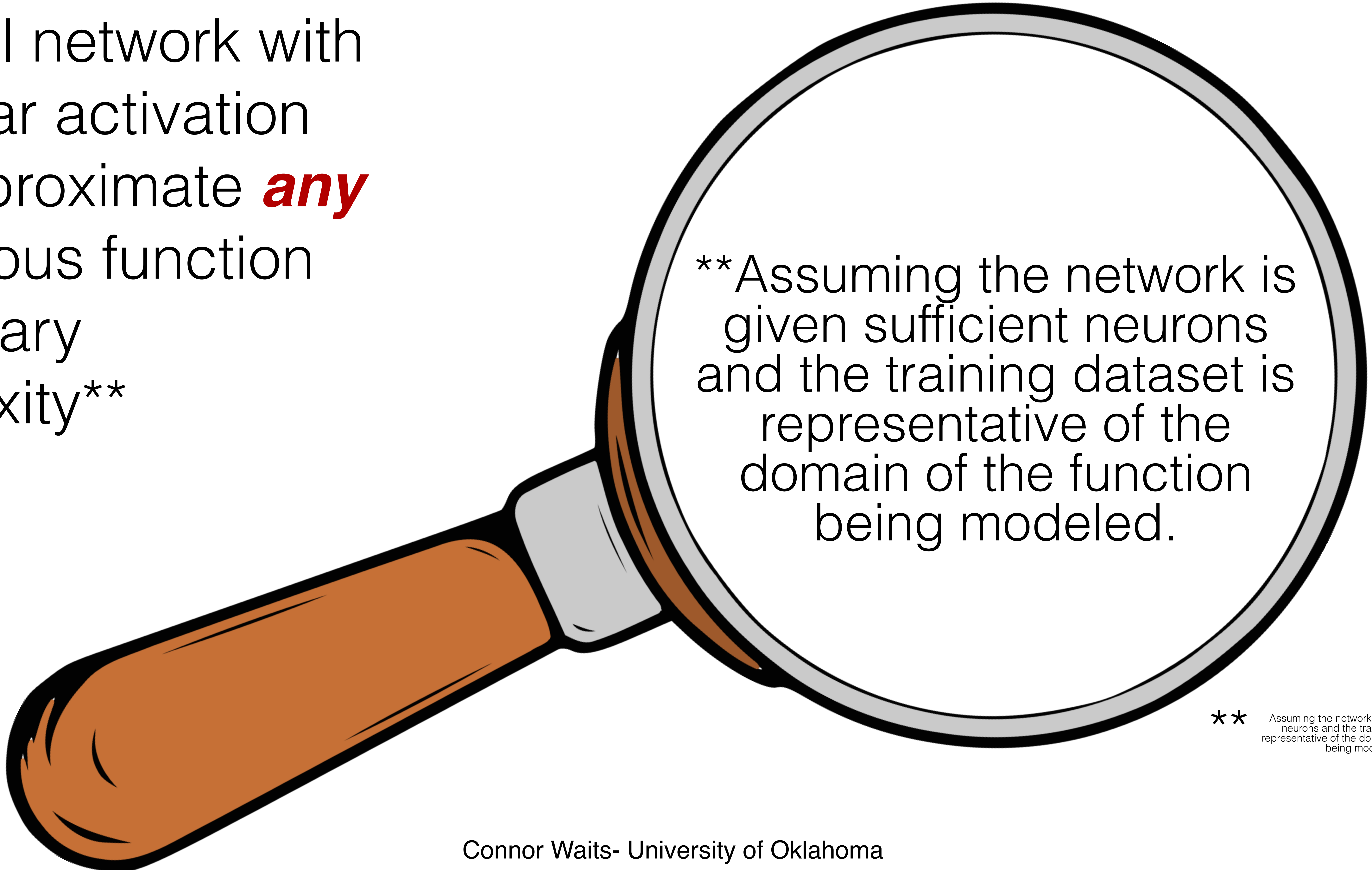


Machine Learning Basics

- Subfield of AI focused on teaching models to make predictions based on generalizations observed in data
  - Supervised: Data is labeled (used for classifications and regressions)
  - Unsupervised: Data is not labeled or partially labeled (used for grouping and clustering)
- Most common method for applying ML techniques is with neural networks

Universal Approximation Theorem

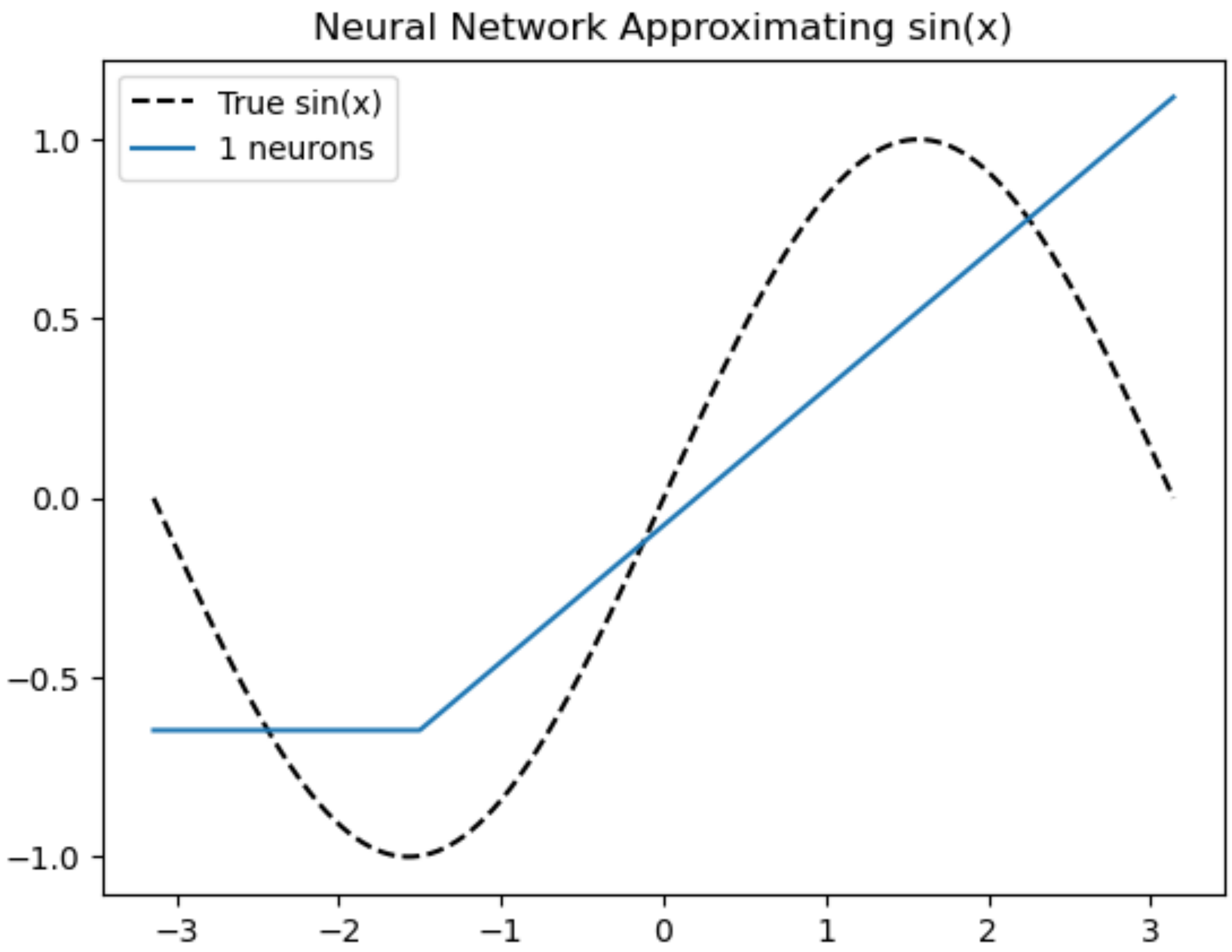
- A neural network with nonlinear activation can approximate ***any*** continuous function of arbitrary complexity**



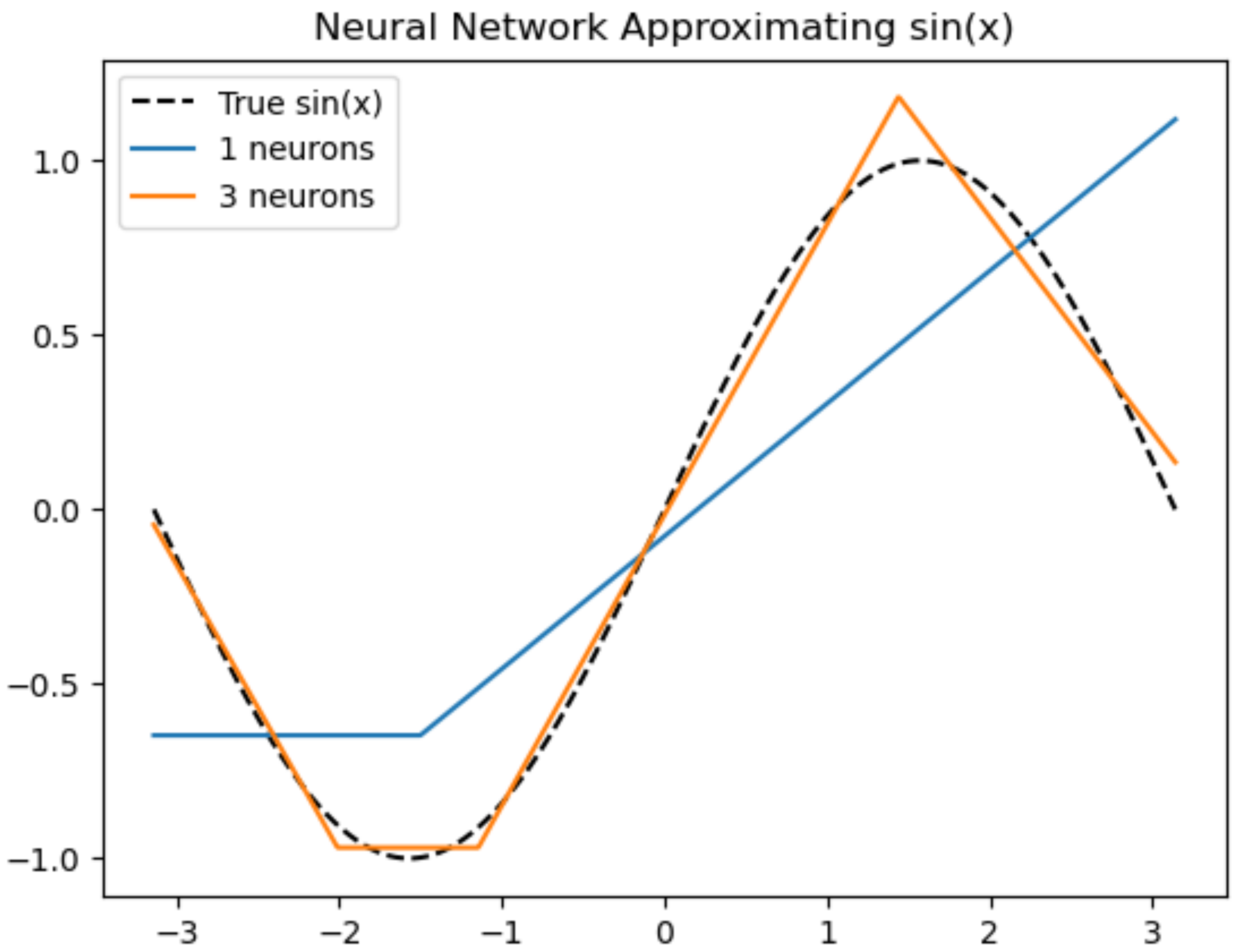
**Assuming the network is given sufficient neurons and the training dataset is representative of the domain of the function being modeled.

** Assuming the network is given sufficient neurons and the training dataset is representative of the domain of the function being modeled.

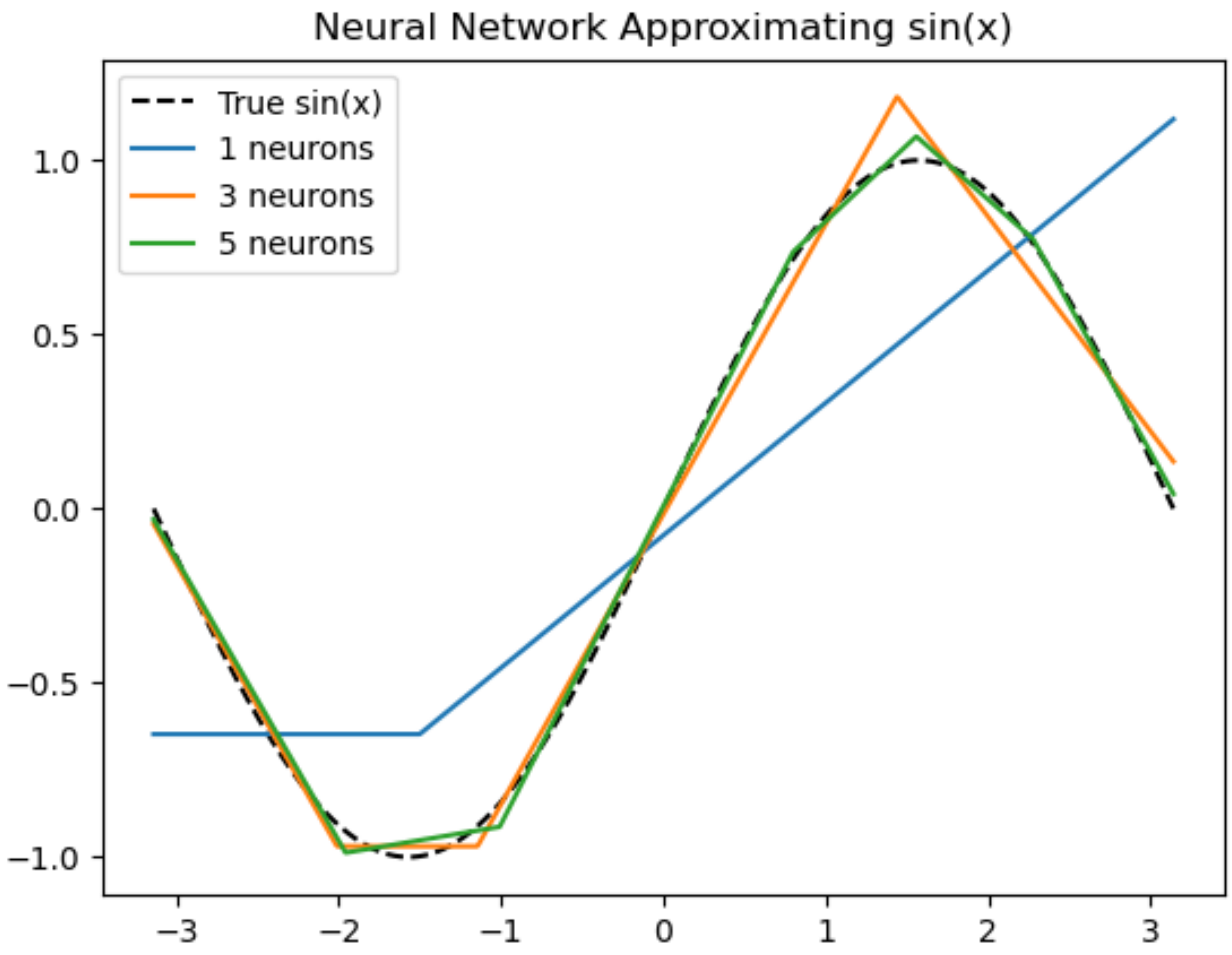
Universal Approximation Example



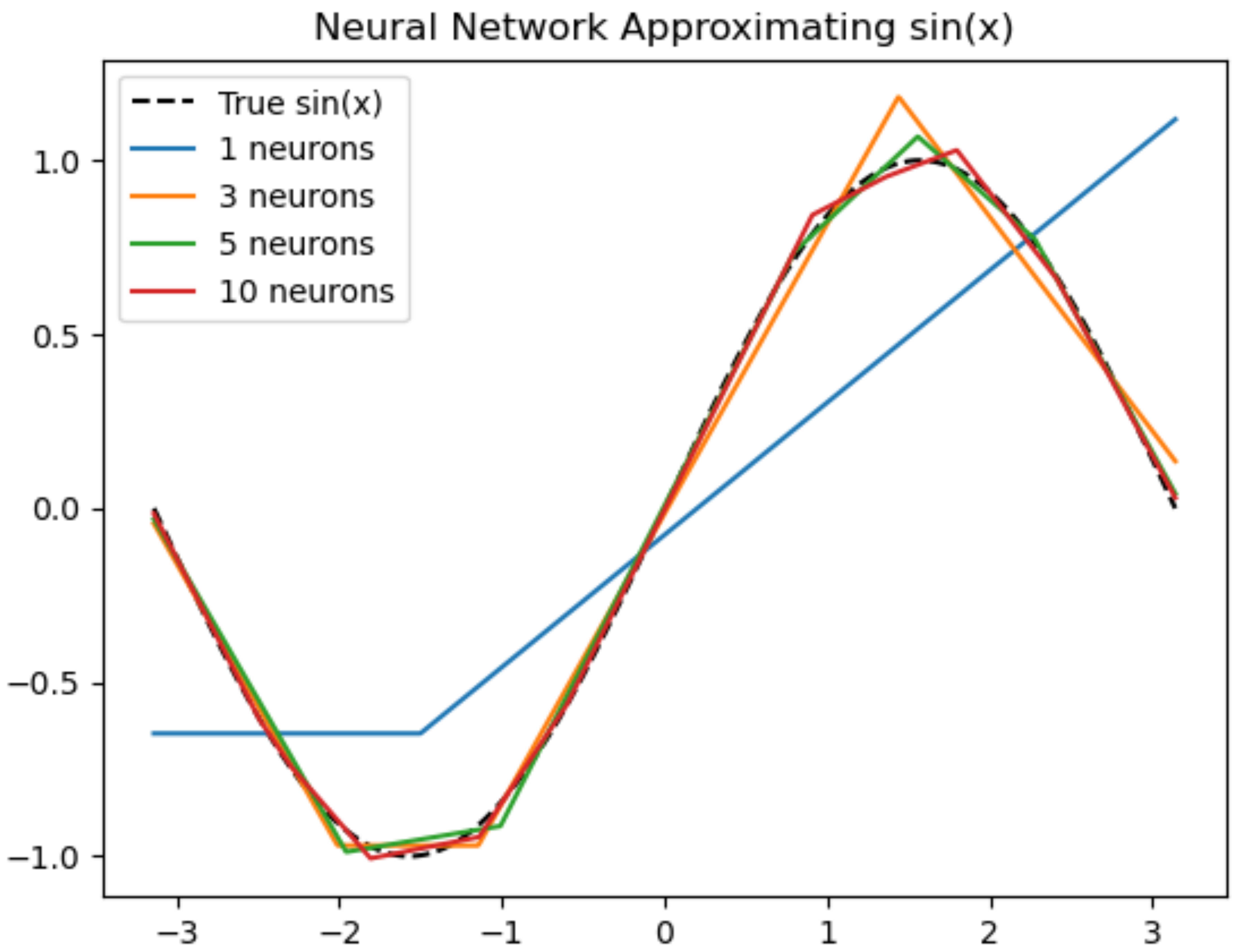
Universal Approximation Example



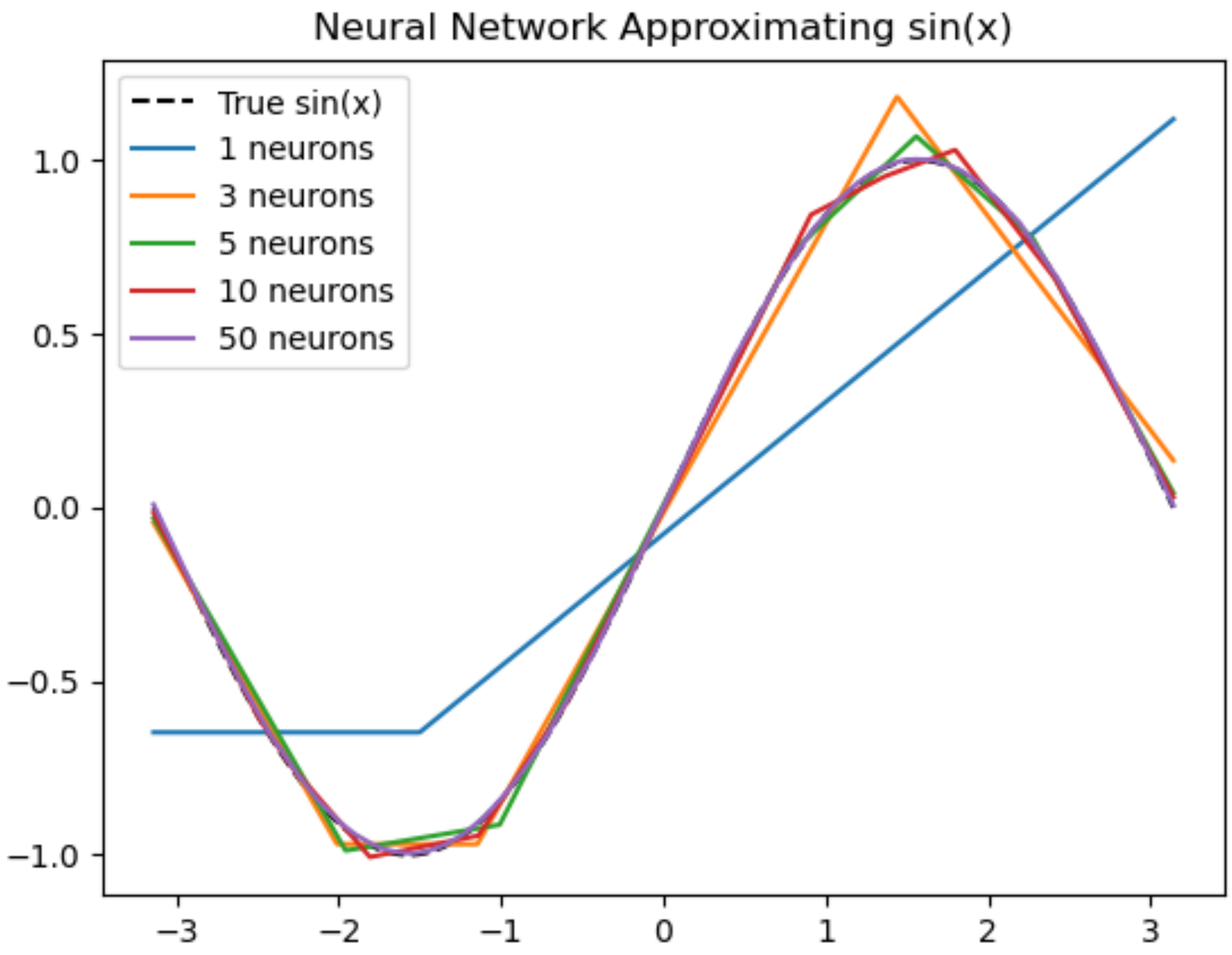
Universal Approximation Example



Universal Approximation Example

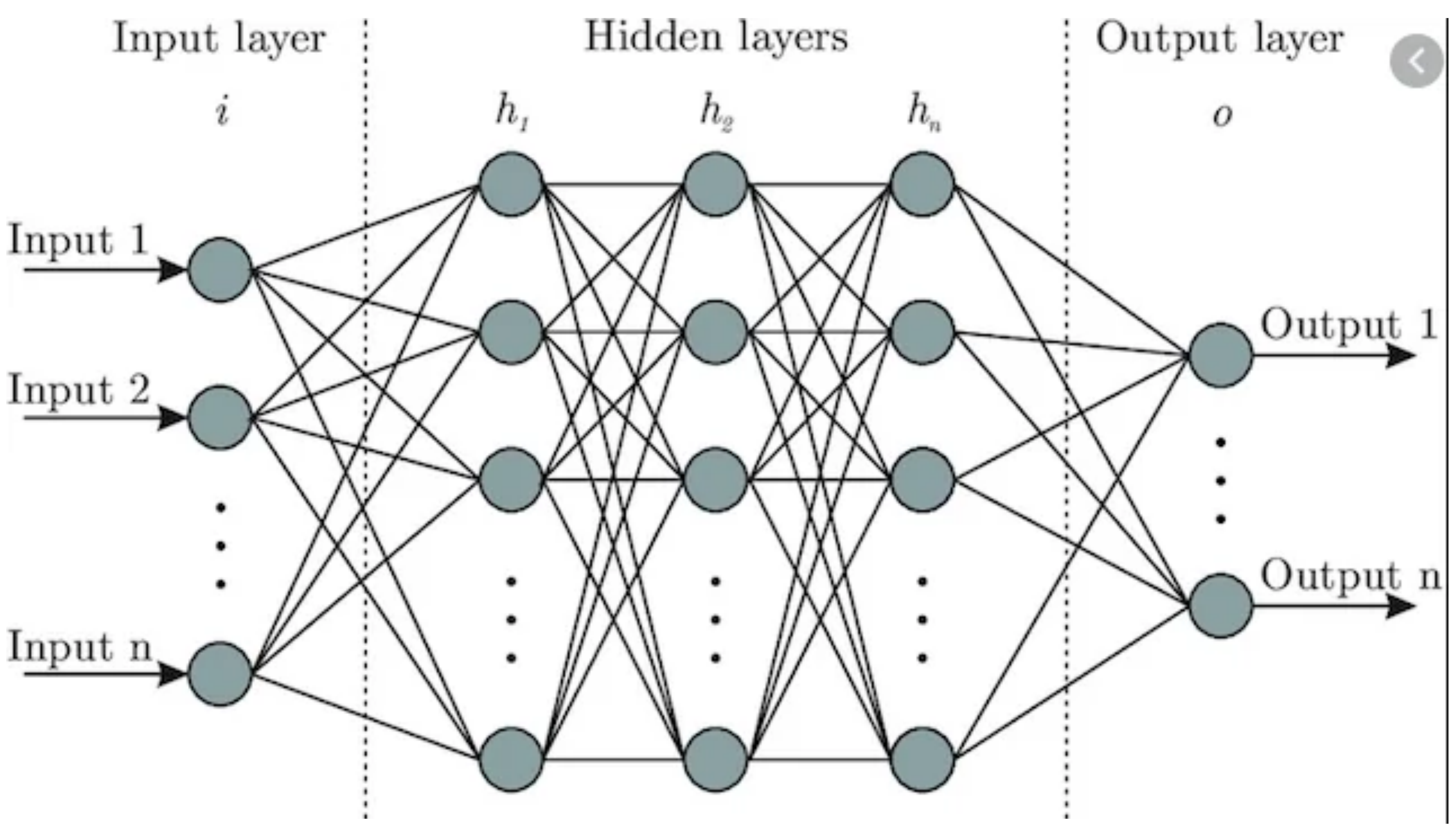


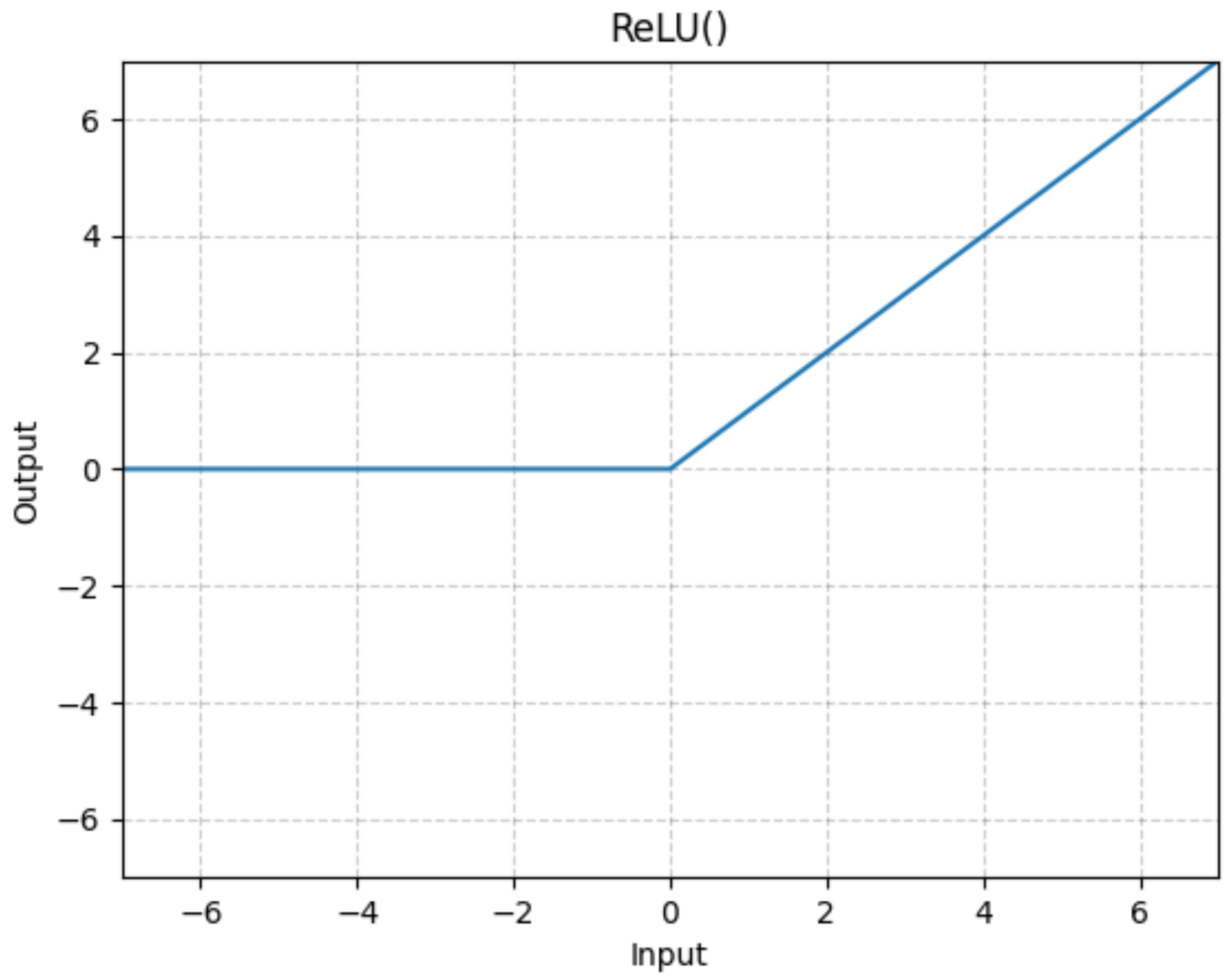
Universal Approximation Example



What is a Neural Net?

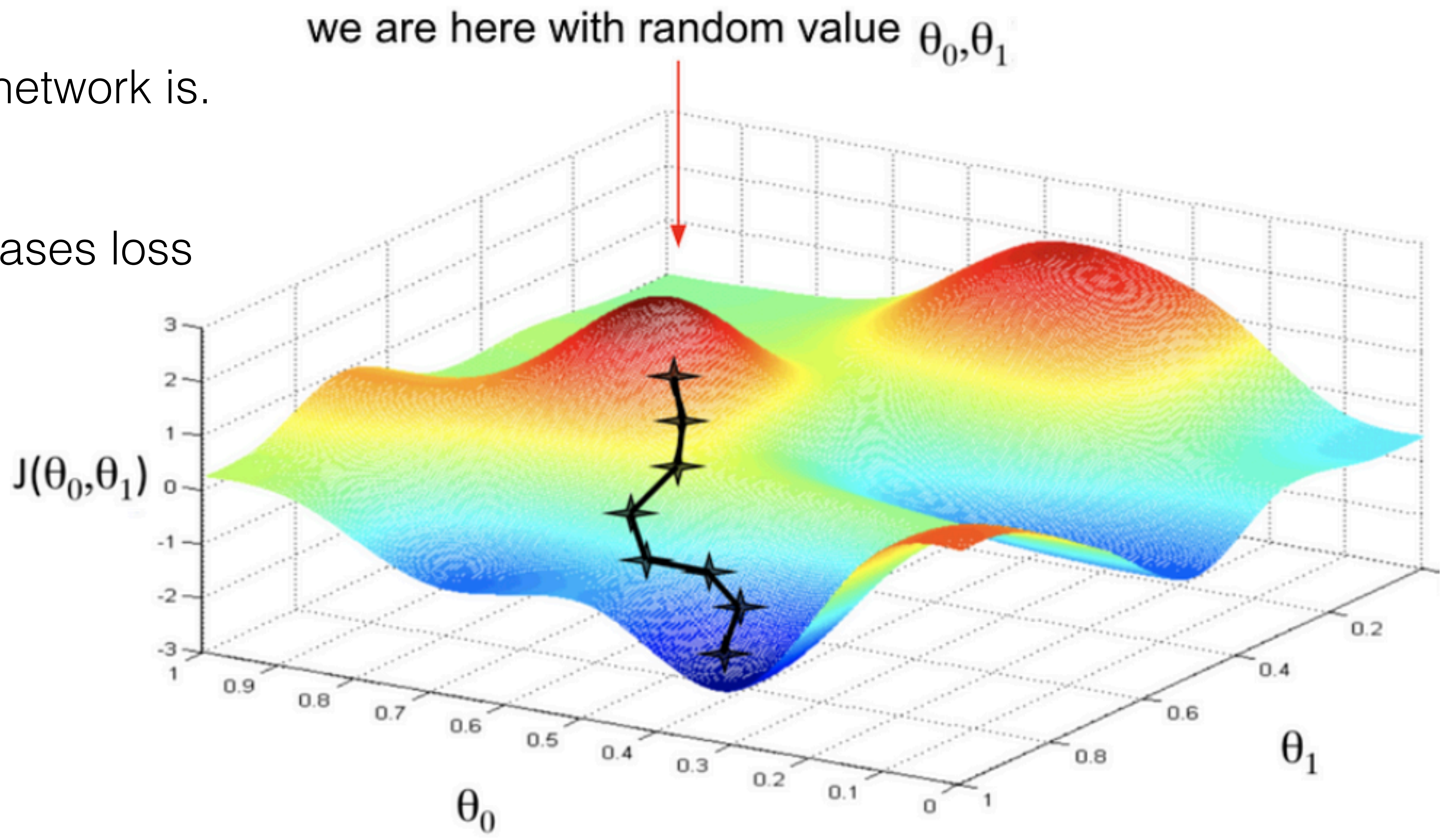
- Set of layers of nodes (or neurons) connected to each node of the previous and subsequent layers
    - Outputs of previous layer passed as inputs to next layer
    - “Training” is the process that finds the optimal set of weights connected each layer
    - The input to the node is $A(\overrightarrow{x} \cdot \overrightarrow{W} + b)$ where A is some non-linear activation function
    - Output of last layer is the probabilities for each possible class



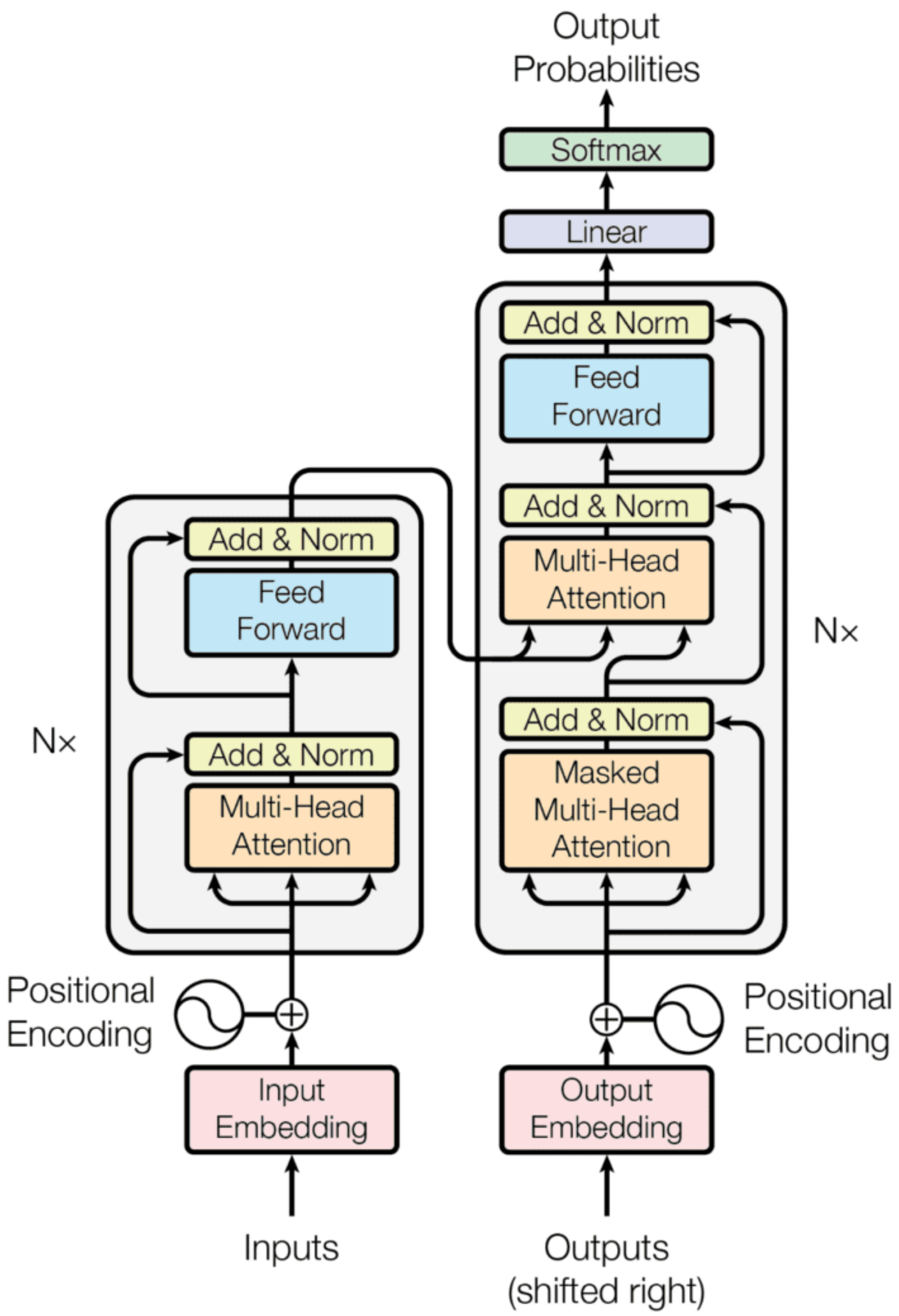


How to train your neural network

- Training: Repeatedly inputting your training dataset through the network and adjusting weights to minimize a "Loss Function"
  - Loss function is a measure of "how incorrect" the network is. Compares model outputs to true labels
  - Updates weights in direction that maximally decreases loss function

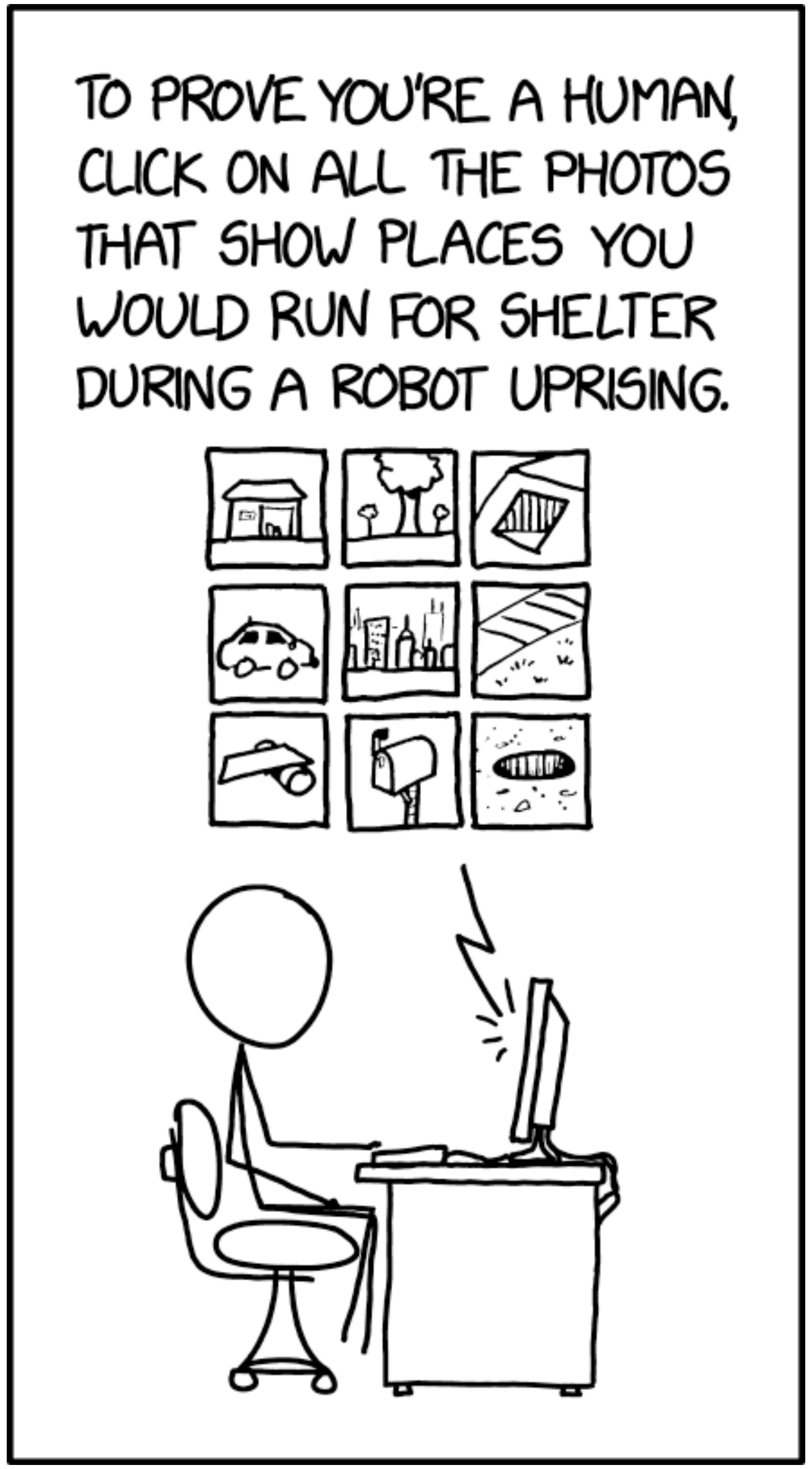


What's a Transformer???



Transformers: Secret Sauce to Modern AI

- Based on encoder-decoder architecture from foundational "Attention is All You Need" paper by Vaswani et al.
- Large Language Model (LLM) transformers consist of:
  - Embedding
  - Stacks of attention + feedforward NN layers
  - Encoder-decoder structure, encoder to understand input, decoder to generate output



Transformer: Example Sentence

- “The quick brown fox jumps over the lazy dog”
  - Embedding: Map each word in the sentence to a numerical vector. Also encodes position of each word
  - Attention: “Which words in this sentence are most important for understanding jump?” Maybe jump pays attention to (attends):
    - “fox” - who did the jumping
    - “over” - modifies the verb jump
    - “dog” - what was jumped
  - Multi-head attention: learn different relationships, one head learns subject-verb relationships (fox-jump) and another learns adjective-noun relationships(quick-fox, lazy-dog), etc
  - Encoder: context learned from attention heads preserved in vector-representation of word
  - Decoder output: encoder and previous decoder outputs used as input - attends to every word in input sentence and response generated so far (superior performance on long sequences compared to earlier natural language processing models such as recurrent neural nets)

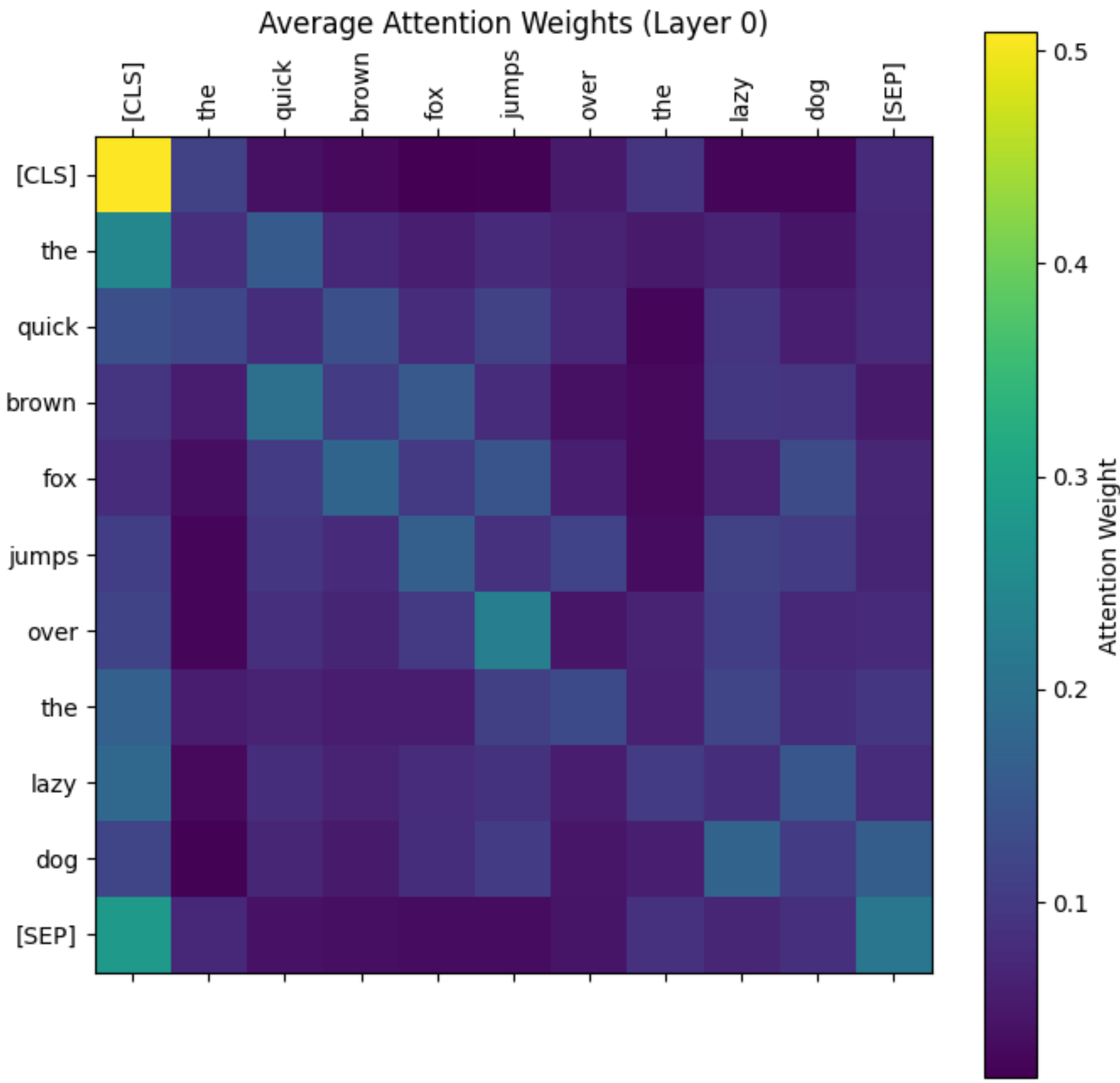
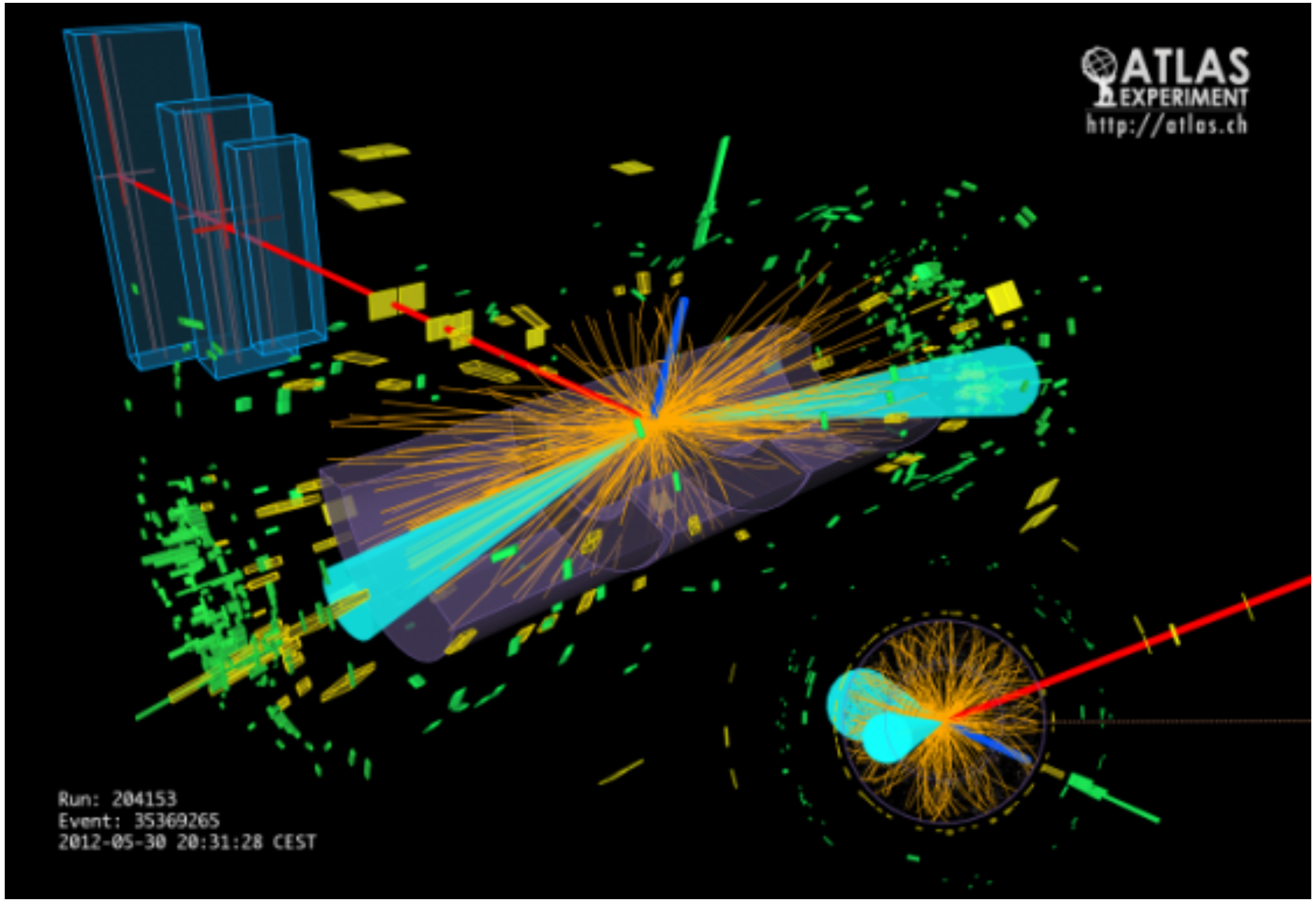


Image Source

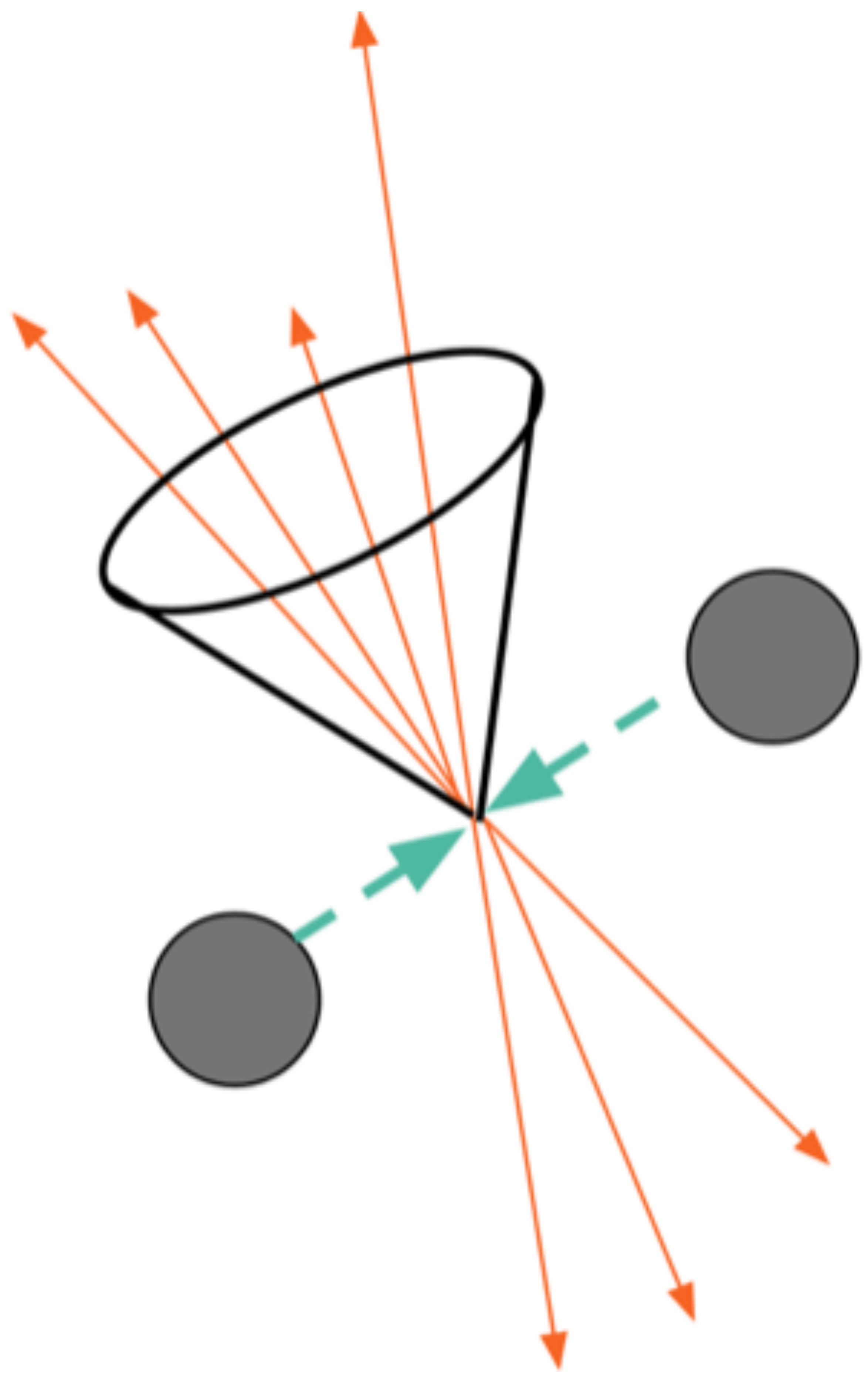
ML revolution in HEP

- ML techniques prevalent in several areas of HEP:
  - Anomaly detection
  - Simulation & background modeling
  - Tracking
  - Triggering
  - ***Jet classification***
  - More!!!

Our focus today

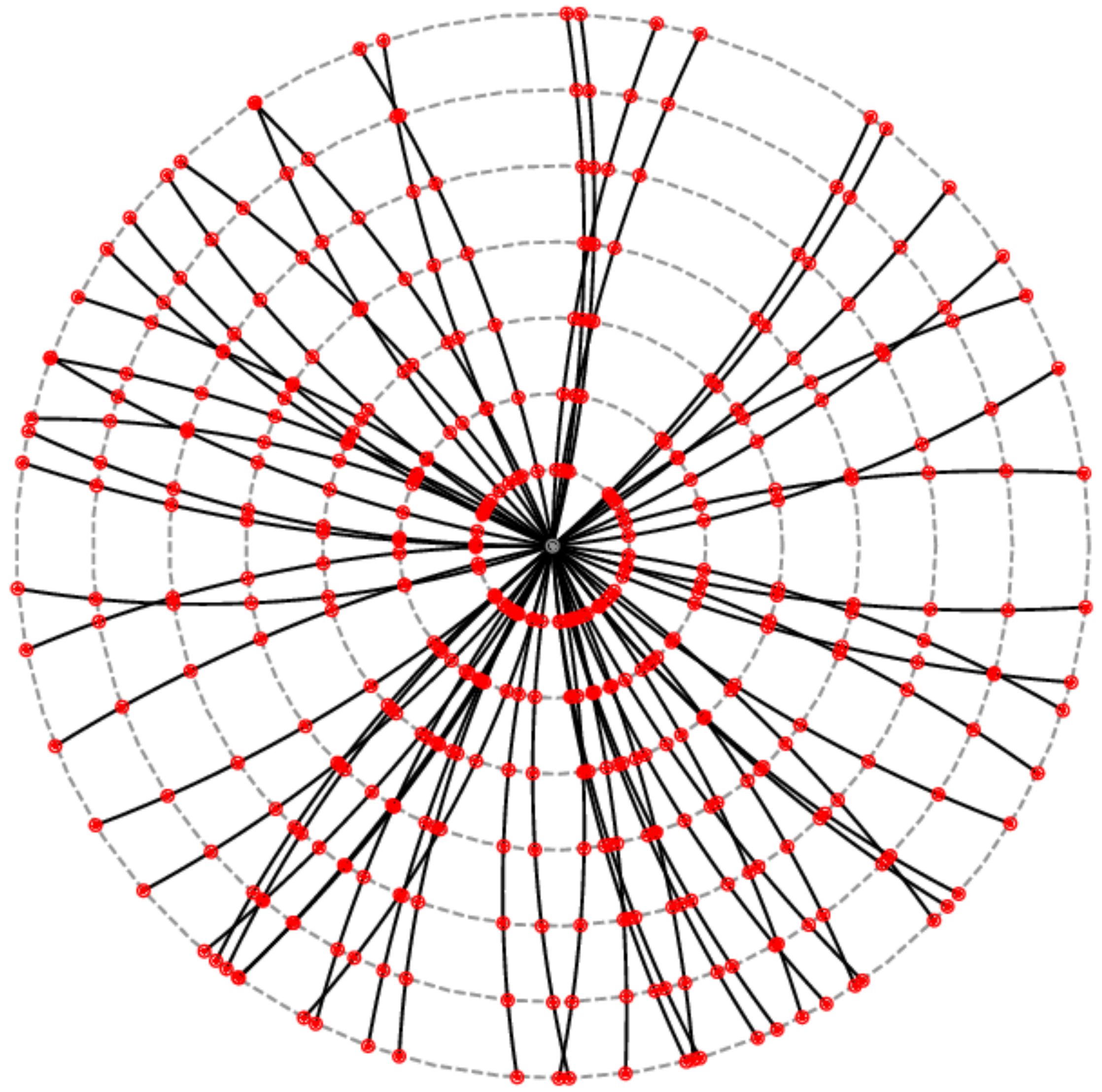


Jets in HEP



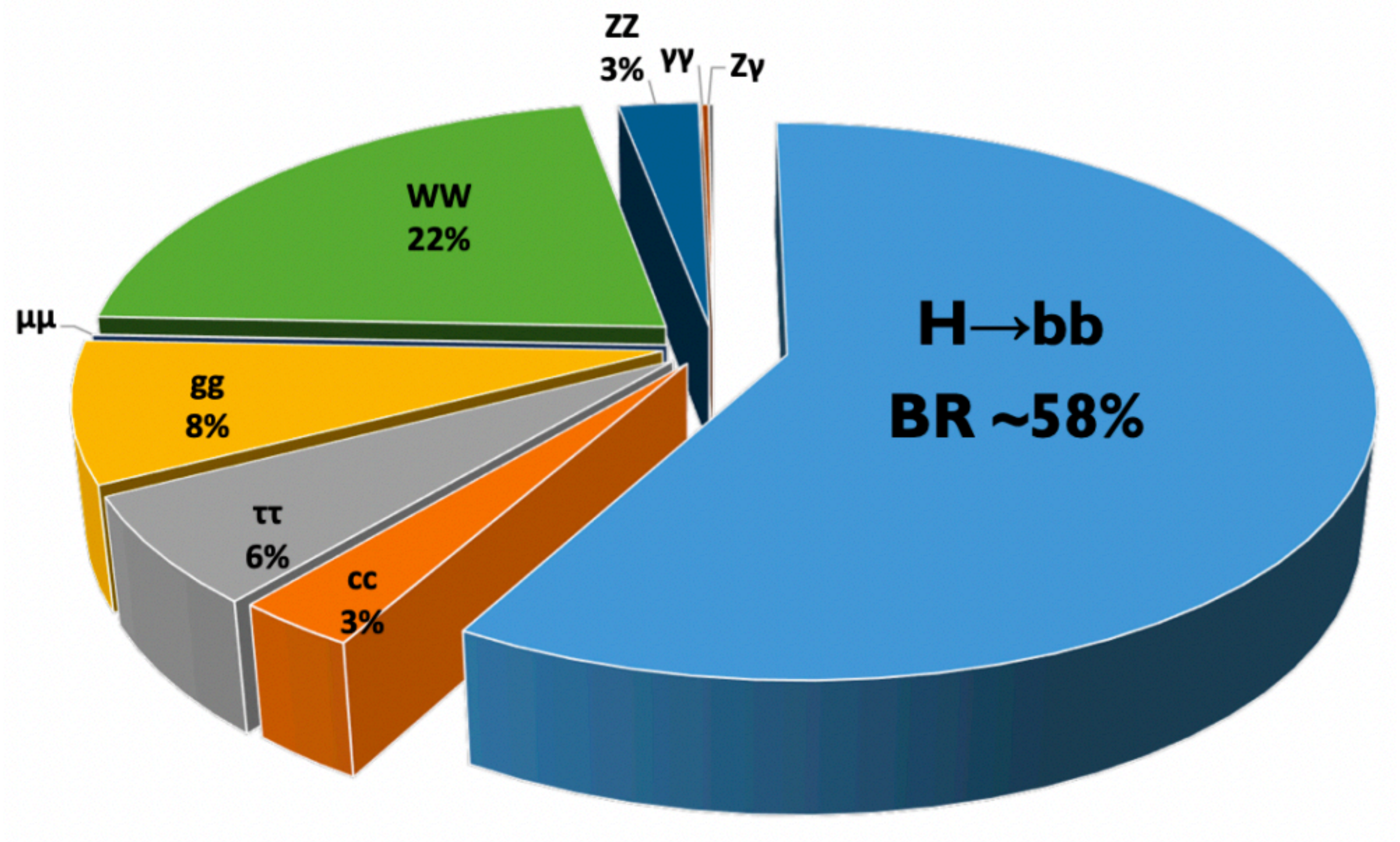
- Spray of collimated particles originating from the hadronization of particles produced in a high energy collision
  - Show up as cluster of energy and tracks in a narrow cone of the detector
  - Jets from different origins have characteristic substructures
- Tracks marking the trajectory are left when charged particles traverse layers of inner detector
  - Serve as inputs for ML jet classifiers

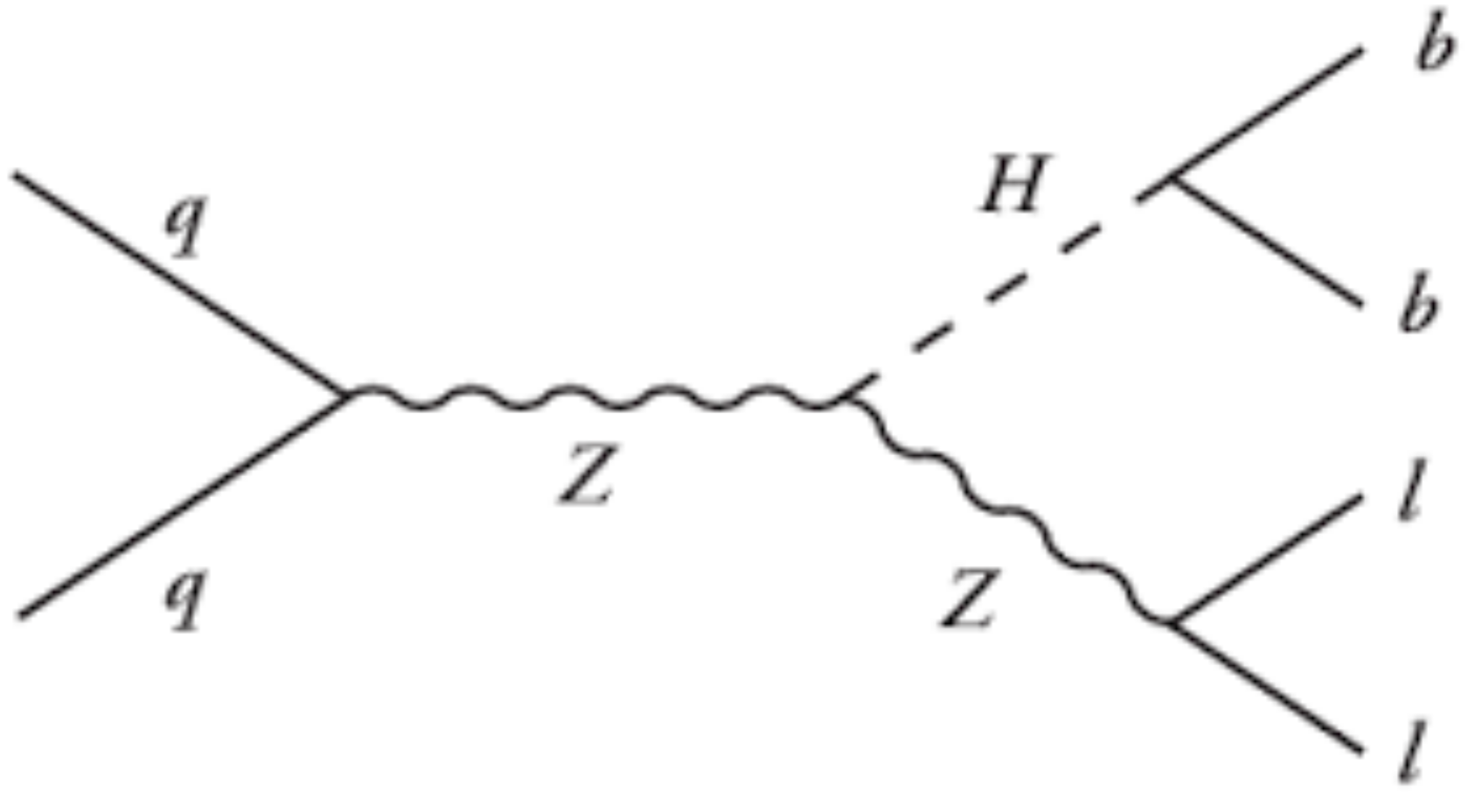
Red dots are "hits" in detector layers. Many hits make up a track



Why tag jets?

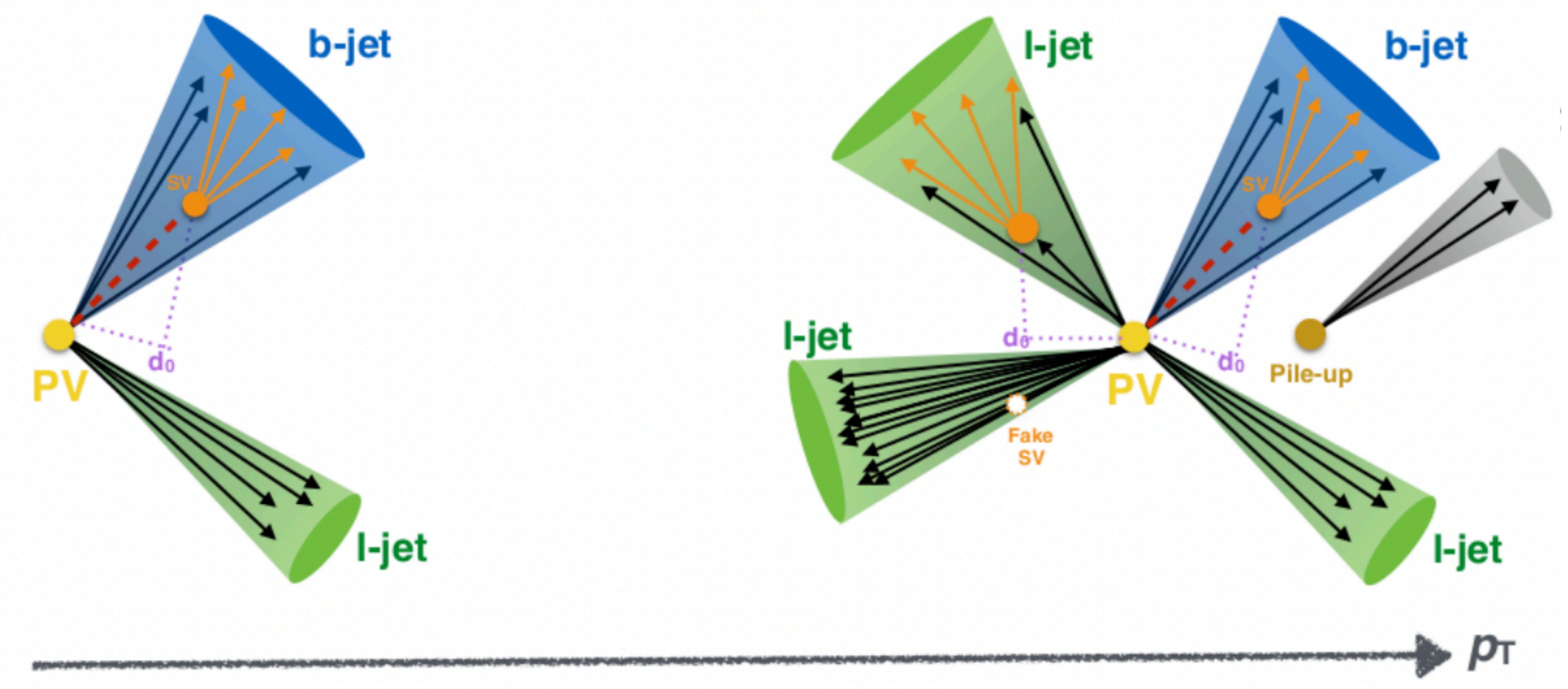
- Hadron Colliders - messy environment with many jets in the final state
  - Jets w/ light flavour hadrons (u/d/s quarks+gluons) >> jets w/ heavy flavour hadrons (c/b quarks)
- Useful to select interesting events, such as $H \rightarrow bb/cc$





B-jet Signatures

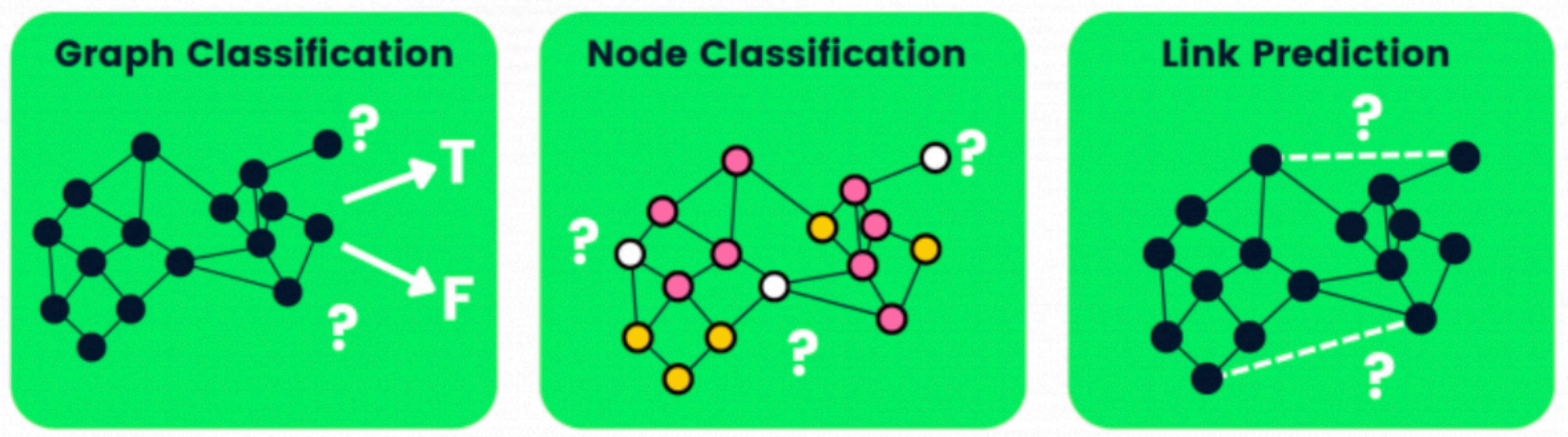
- B-hadrons are relatively long-lived ($\tau \sim 0.5$ mm)
- B-hadron decays are characterized by:
  - Large track impact parameters (tracks are displaced from primary vertex)
  - Displaced secondary vertices w/ high mass
  - Displaced tertiary vertices from $B \rightarrow C$ decays



S. Van Stroud

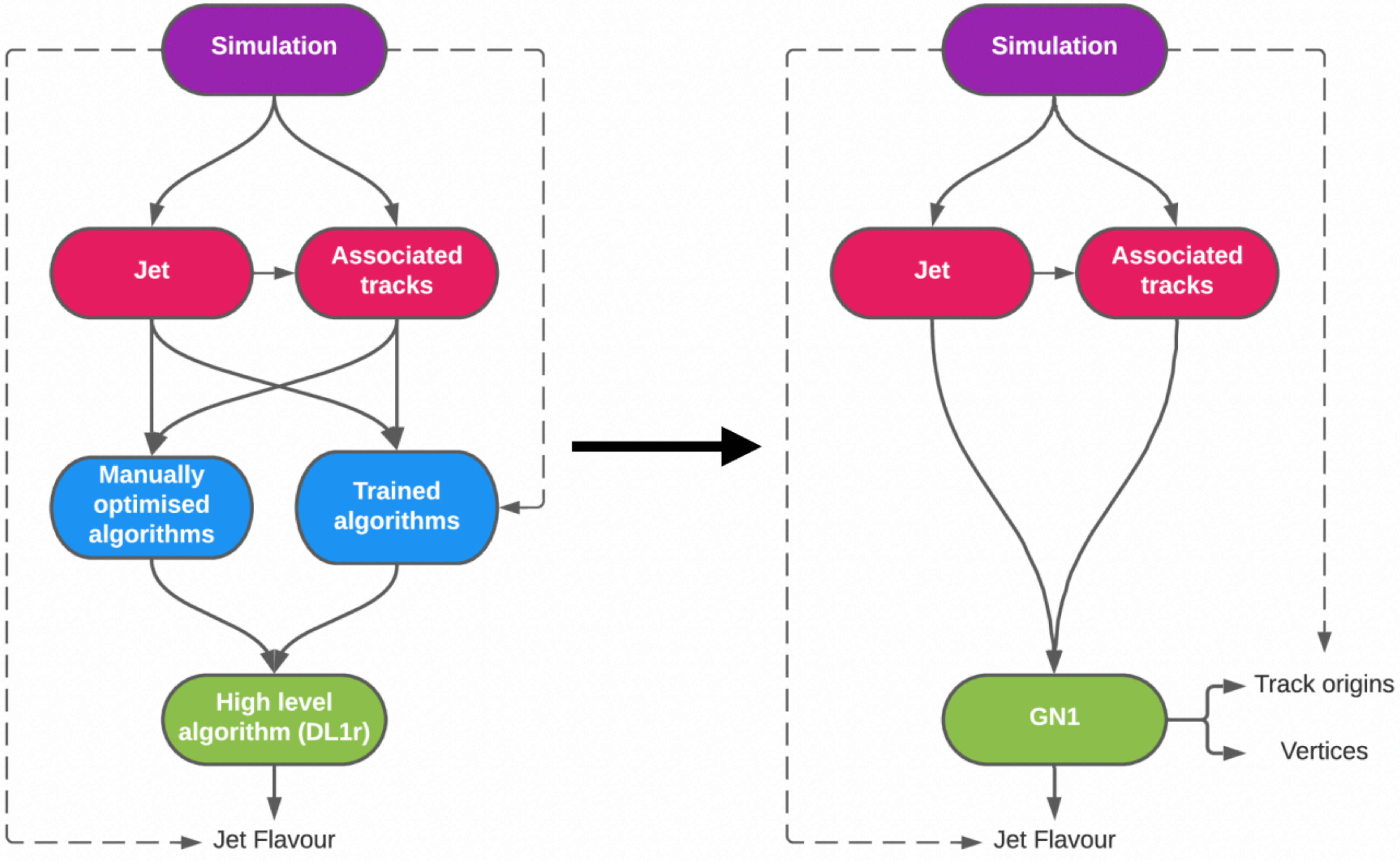
LLM’s to Graph Neural Nets

- Sequences are natural representations of sentences -> graphs are natural representations of tracks in a jet
- In ATLAS: Implement transformer style graph neural nets to:
  - Classify Jets (graph classification)
  - Classify origin of tracks (node classification)
  - Predict tracks originating from shared vertex (link prediction)



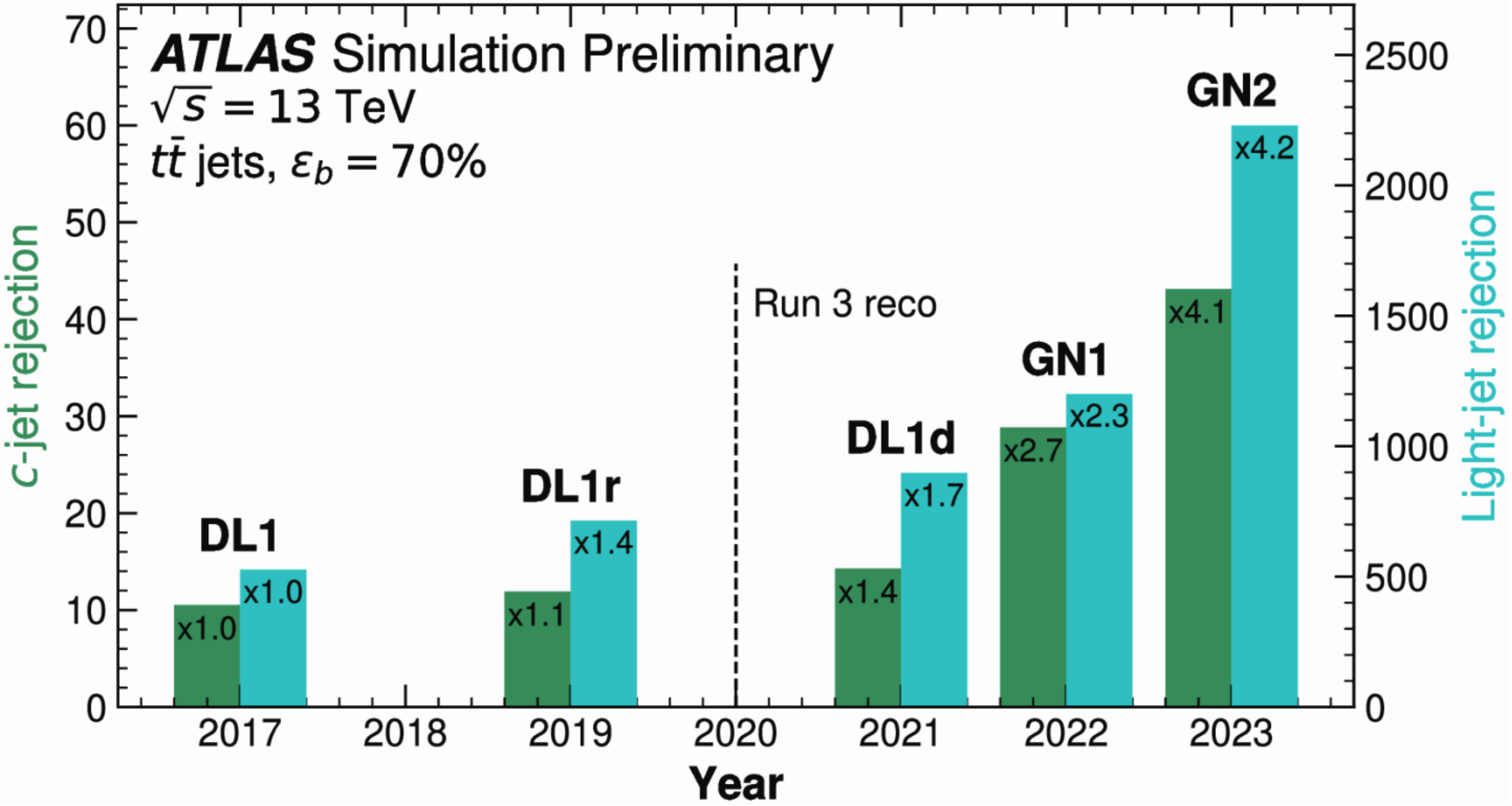
DL1r → GN1

- GN1 removes the need for low-level taggers; “All-in-One Tagger”
- Uses a single model for jet classification
- Implements track origin and vertexing auxiliary tasks

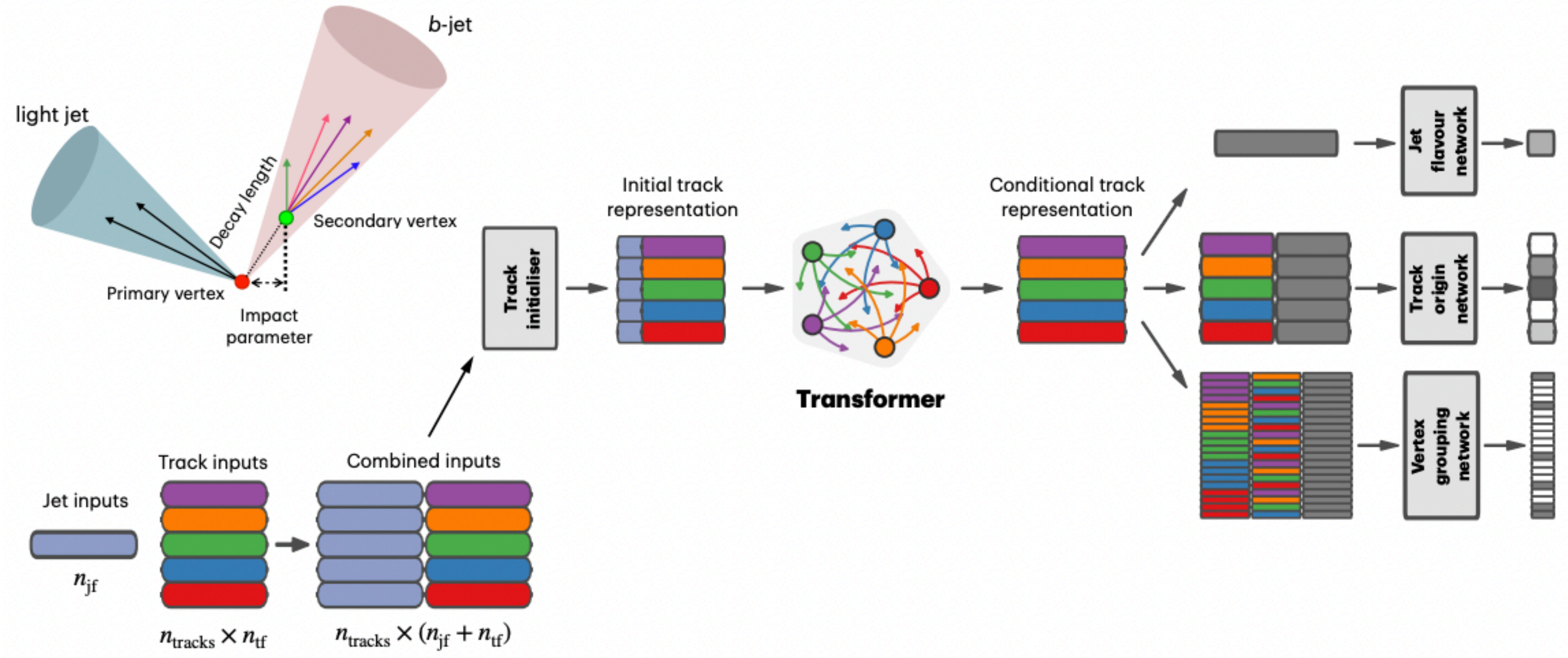


S. Van Stroud

GN2 Performance

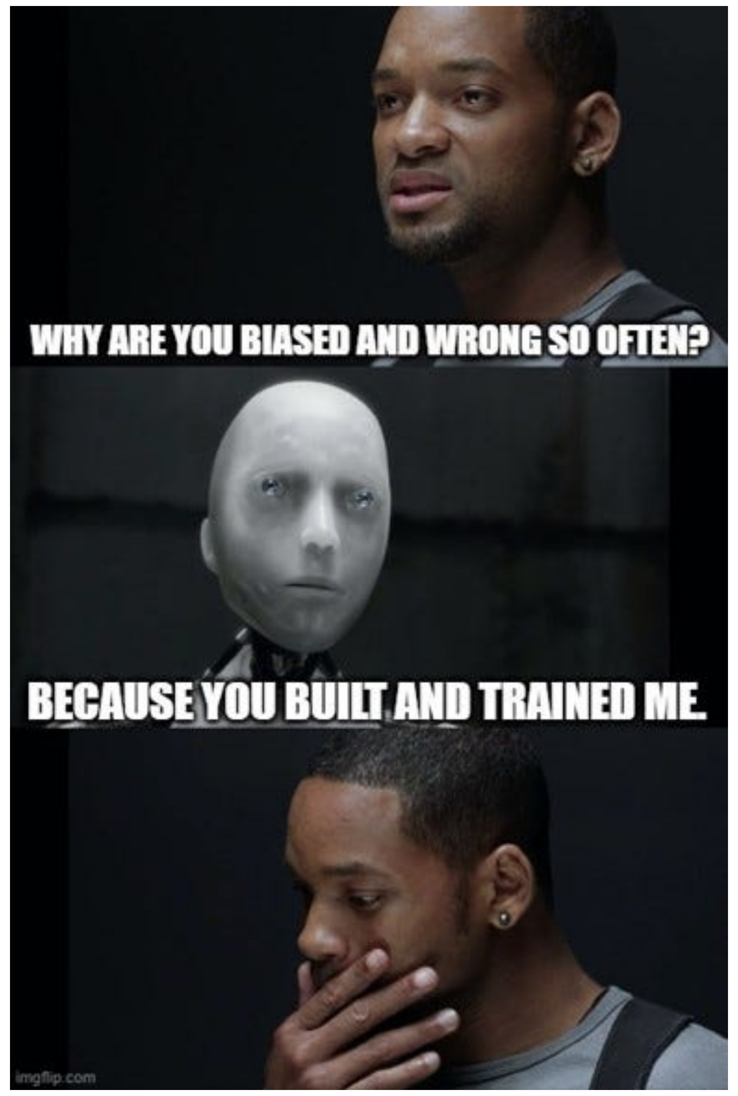


GN2 Architecture



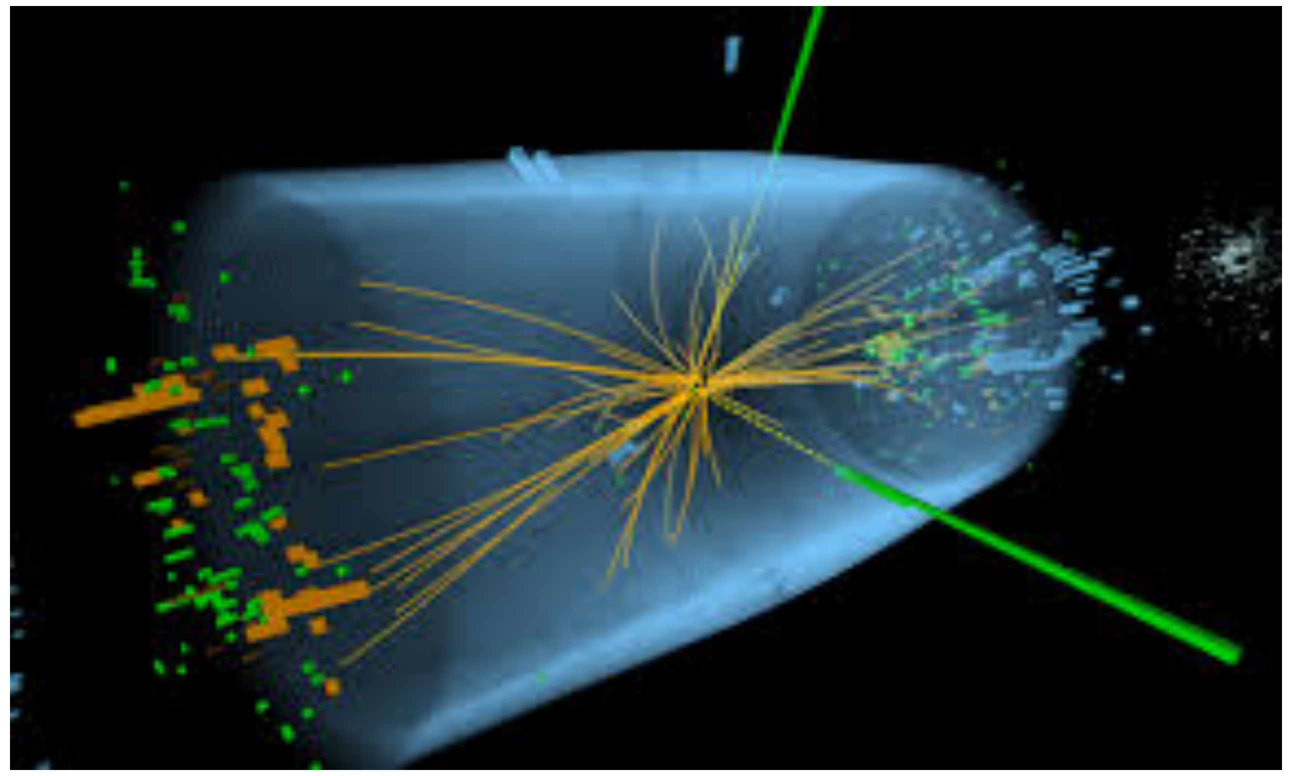
Limitations of ML

- ML outputs are rarely "explainable"
  - Black-box predictions only useful to certain extent
- Overconfident
  - Models are rarely conservative in predictions
  - Algorithms that offer high performance and accuracy actually encourage overconfidence
- Highly domain specific
  - Easily inherits biases from training data
  - Fails to generalize to outlying data



Conclusions

- Modern LLM's are very powerful, but…
  - Expensive to train/use
  - Already trained on (basically) entire internet, AI data consumption may outpace human generation
  - Overconfidence
- ML used in HEP for decades
  - Becoming a more integral part of analysis
  - Use in data-taking set to expand



Thanks!