How can complex systems help us understand why ML/AI methods work and how they work?

Complex systems tools are built to help us understand large, high dimensional systems

ML/AI are models—but they are also large, high dimensional systems that exhibit complex emergent behavior

Indeed, the behavior we are training these models to exhibit is an emergent property of the interaction of many model components (e.g. neural networks)

Complex systems tools can help us understand how and why they do what they do

Emergence and ML/AL

What’s the difference between a statistical model and a machine learning model?

Scale—this is in some ways a very complex systems thing in that the target behavior of something like a neural network is really an emergent property of a large number of “neurons” interacting with each other (i.e. feeding in input from one to the other)

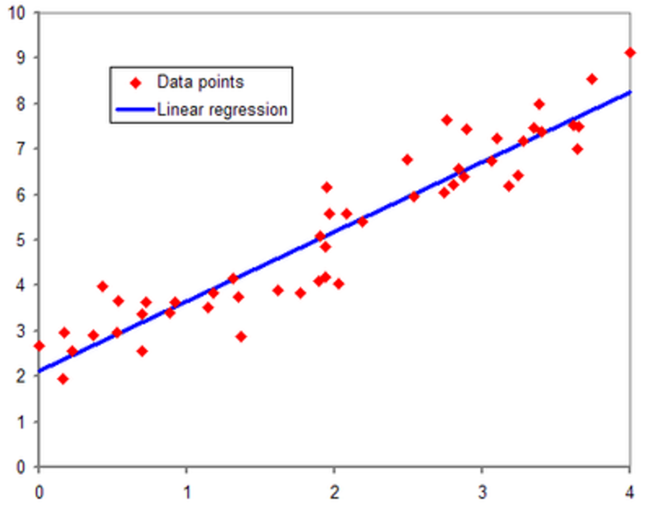
Neural networks are networks, also abms basically

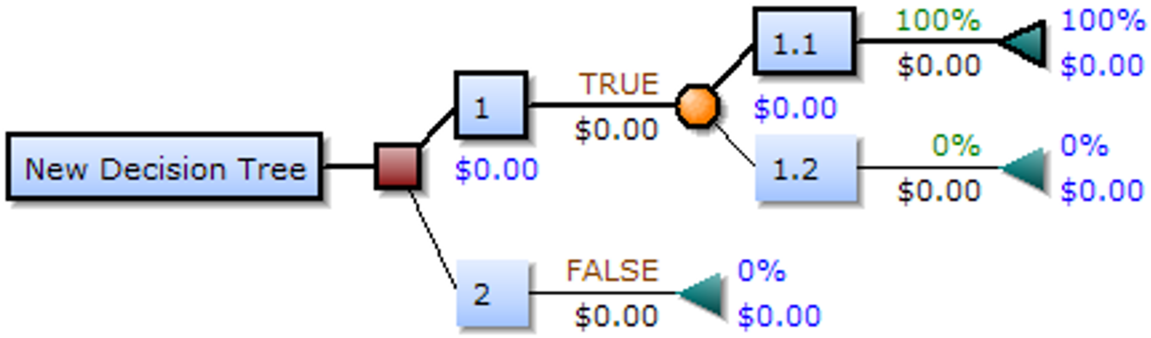
Even though as models these are not very CS-y, as systems they use a lot of CS principles

Explainable AI and mechanistic interpretability

Simpler ML models like linear regression, logistic regression, decision trees are more interpretable

But to solve more complex problems and represent more high dimensional, complex data, we need more complex ML models—deep neural networks are a common choice, but are largely black box





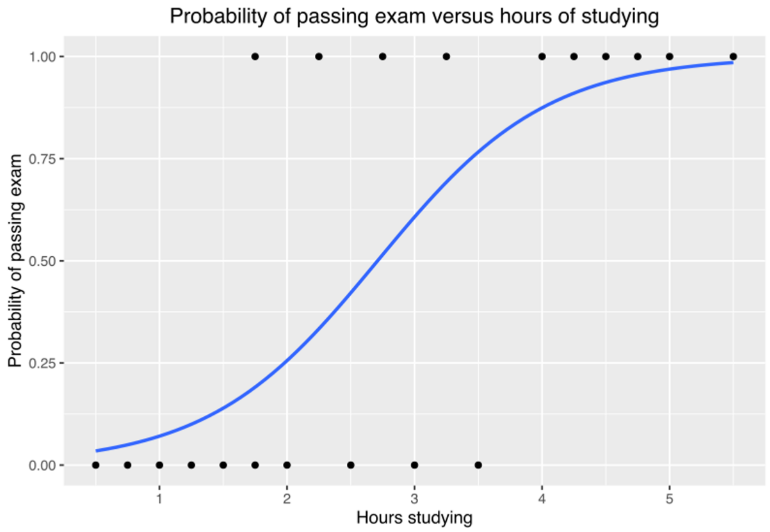
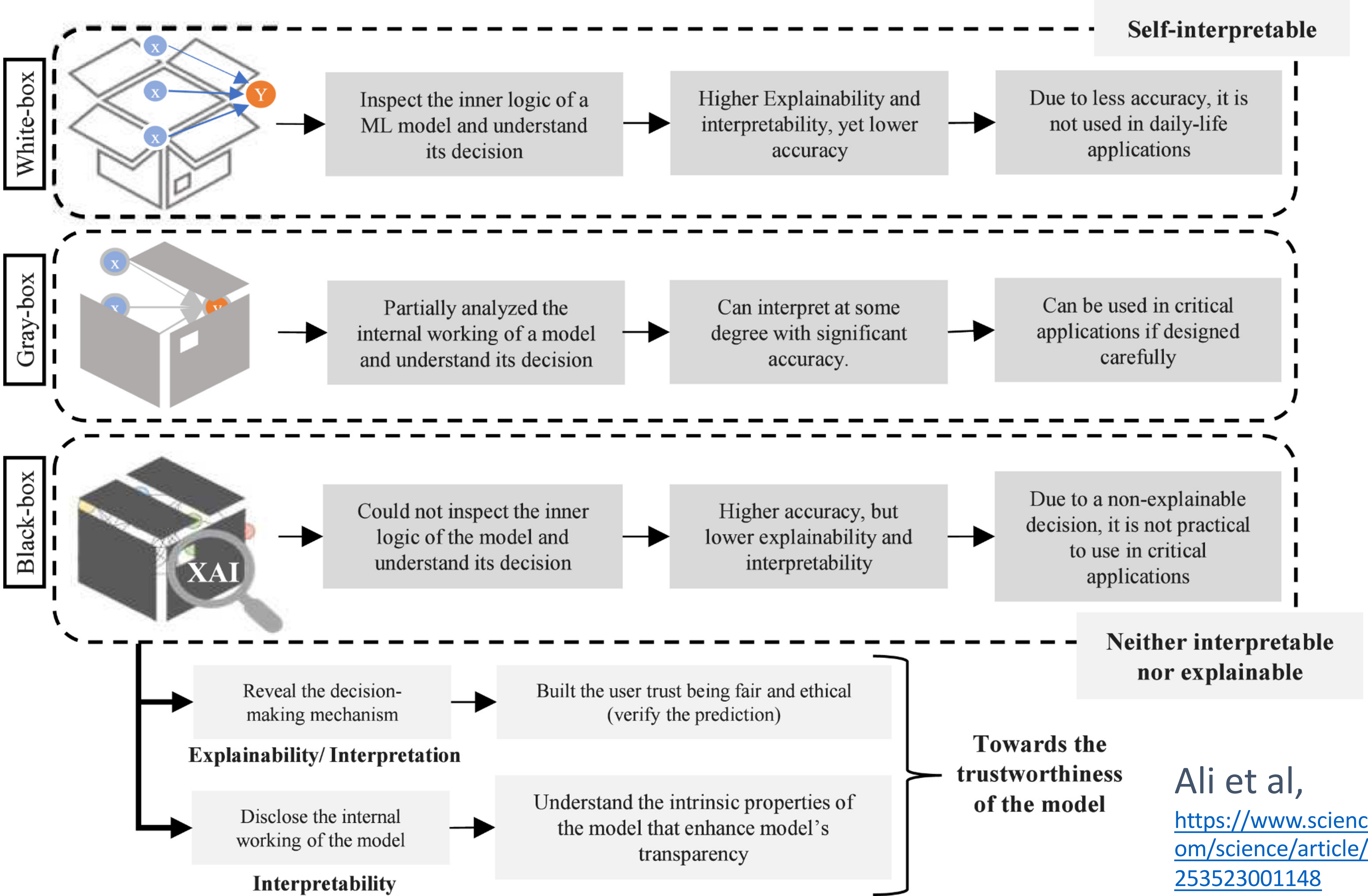


Image sources: Wikipedia



Ali et al,
https://www.sciencedirect.com/science/article/pii/S1566253523001148

Explainable AI and mechanistic interpretability

Many of the things we're trying to understand are things like

- Bifurcations/equilibria/tipping points (why does the model go to this classification or that classification under different circumstances)
- Optimization methods for parameter estimation
- Sensitivity analysis (which parameters or features or weights are driving the model output)

Explainable AI

Many approaches!

- Understand the data used to train the model
- Explore examples representing different cases and see how the model processes those
- Build a simpler model (surrogate model) to understand the more complex model

By data explainability	By model explainability	By post-hoc explainability
D1: What sort of information do we have in the database? **D2:** What can be inferred from this data? **D3:** What are the most important portions of the data? **D4:** How is the information distributed? **D5:** Is it possible to increase the model's performance by lowering the number of dimensions? **D6:** Can a better explanation be offered by using data summarizing techniques?	**M1:** What makes a parameter, objective, or action important to the system? **M2:** When did the system examine a para- meter, objective, or action, and when did the model reject it? **M3:** What are the consequences of making a different decision or adjusting a parameter? **M4:** How does the system carry out a certain action? **M5:** How do these model parameters, objec- tives, or actions relate to one another? **M6:** What factors does the system take into account (or disregard) when making a decision? **M7:** In order to achieve a goal/inference, which techniques does the system utilize or avoid?	**P1:** What is the reason behind the model's prediction? **P2:** What was the reason for occurrence X? What would happen if Y was the cause of occurrence X? **P3:** What variables have the most influence on the user's decision? **P4:** What if the information is altered? **P5:** To keep current results, what criteria must be met? **P6:** Is there anything that can be done to have a different outcome? **P7:** Why is it essential to make a certain conclusion or decision?

Explainable AI and mechanistic interpretability

We need to be able to understand why and how AI does what it does so that we can control/regulate/safely use it

Part of the point of AI is to be able to brute force analyze much larger amounts of data than we could ever hope to do

Pretty much all explainable AI approaches are trying to do some kind of dimensionality reduction

Complex systems tools can help to understand ML/AI and how it works—both to make sure it isn't learning something silly (cancer & rulers example) and to know if there is bias

LLMs and AI as models and as systems

- Thinking of these models as models vs. as systems
- What are their dynamics? What are equilibria in these models?
- How do they associate ideas? (E.g. some data must be more “close” to a given data point/prompt/etc than others, which implies some sort of association—does this map to concepts we think about?

Starting prompt: “armadillo dressed as santa”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a man dressed as a pangl in a santa claus costume”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man dressed as a pangl in a santa claus costume”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a man in a santa costume with a christmas tree behind him”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man in a santa costume with a christmas tree behind him”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a man dressed as film character poses for a photo”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man dressed as film character poses for a photo”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

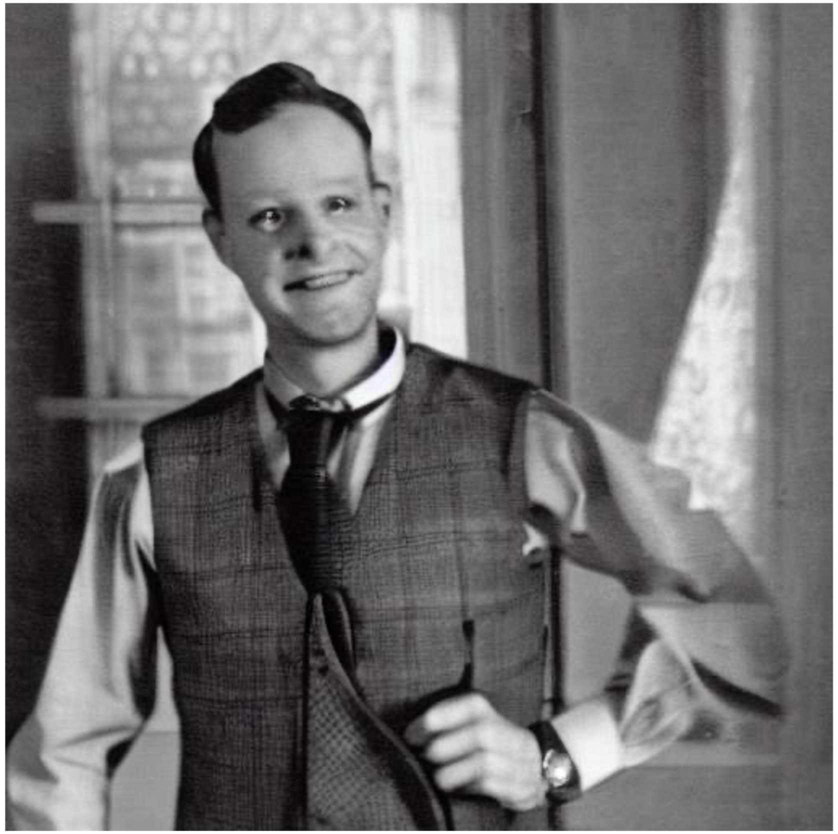
“a man in a vest and tie is pointing at the camera”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man in a vest and tie is pointing at the camera”

UM-GPT - Open Journey



Salesforce/blip-image-captioning-base

“a man in a vest and tie standing in front of a window”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man in a vest and tie standing in front of a window”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a man in a suit and tie looking out a window”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man in a suit and tie looking out a window”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a man in a suit looking out the window”

Very close to convergence on an equilibrium now! The prompt and images are staying very close to each other

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man in a suit looking out the window”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a man in a suit looking out the window”

Converged!

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a man in a suit looking out the window”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

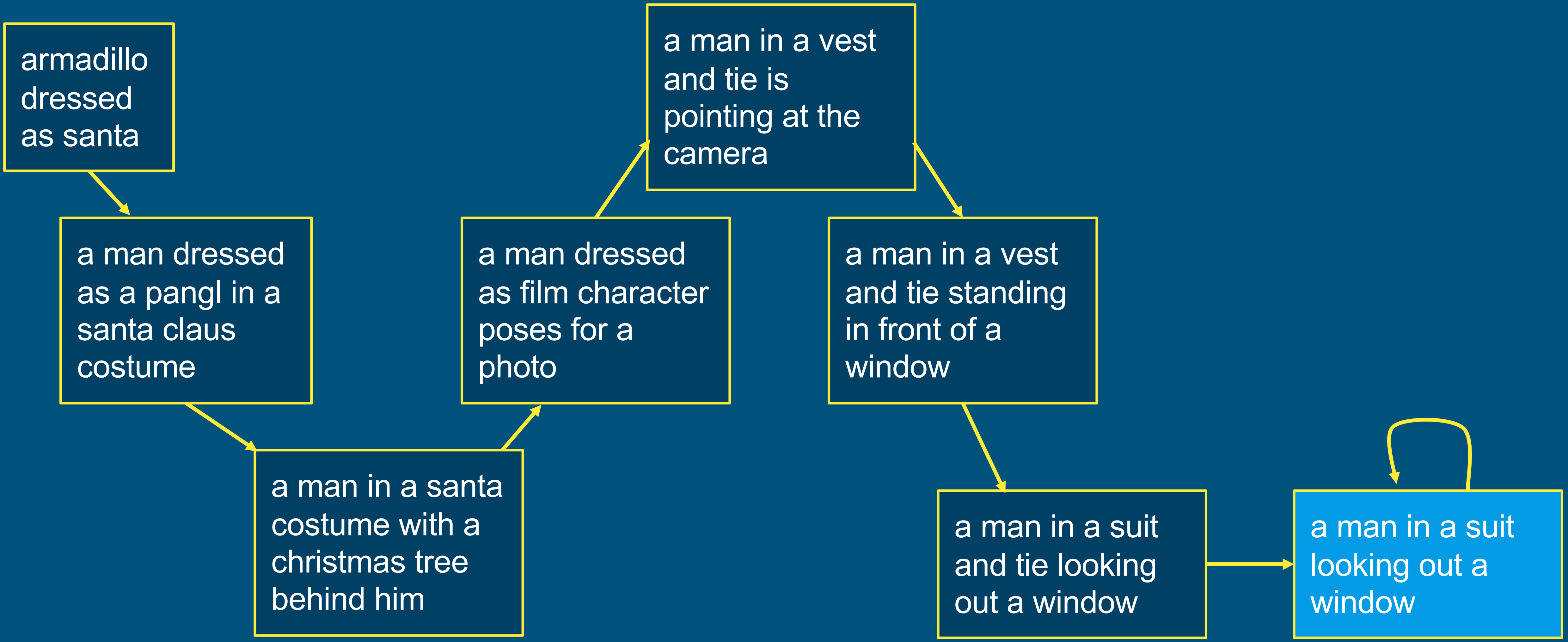
“a man in a suit looking out the window”

Tried it one more time and it stays! But note there is some noise/stochasticity in this dynamical system so if we keep running it, it may bounce away from this equilibrium

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

Mapping how image AI associates concepts



Starting prompt: “A pegasus flying through a galaxy”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a white unicorn with wings flying through the night sky”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a white unicorn with wings flying through the night sky”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a white unicorn standing in a field under a starry sky”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a white unicorn standing in a field under a starry sky”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a unicorn standing in a field with the milky in the background”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a unicorn standing in a field with the milky in the background”

UM-GPT - Open Journey

Salesforce/blip-image-captioning-base

“a unicorn standing in a field with a starr sky in the background”

https://umgpt.umich.edu/

https://huggingface.co/tasks/image-to-text

“a unicorn standing in a field with a starr sky in the background”

UM-GPT - Open Journey

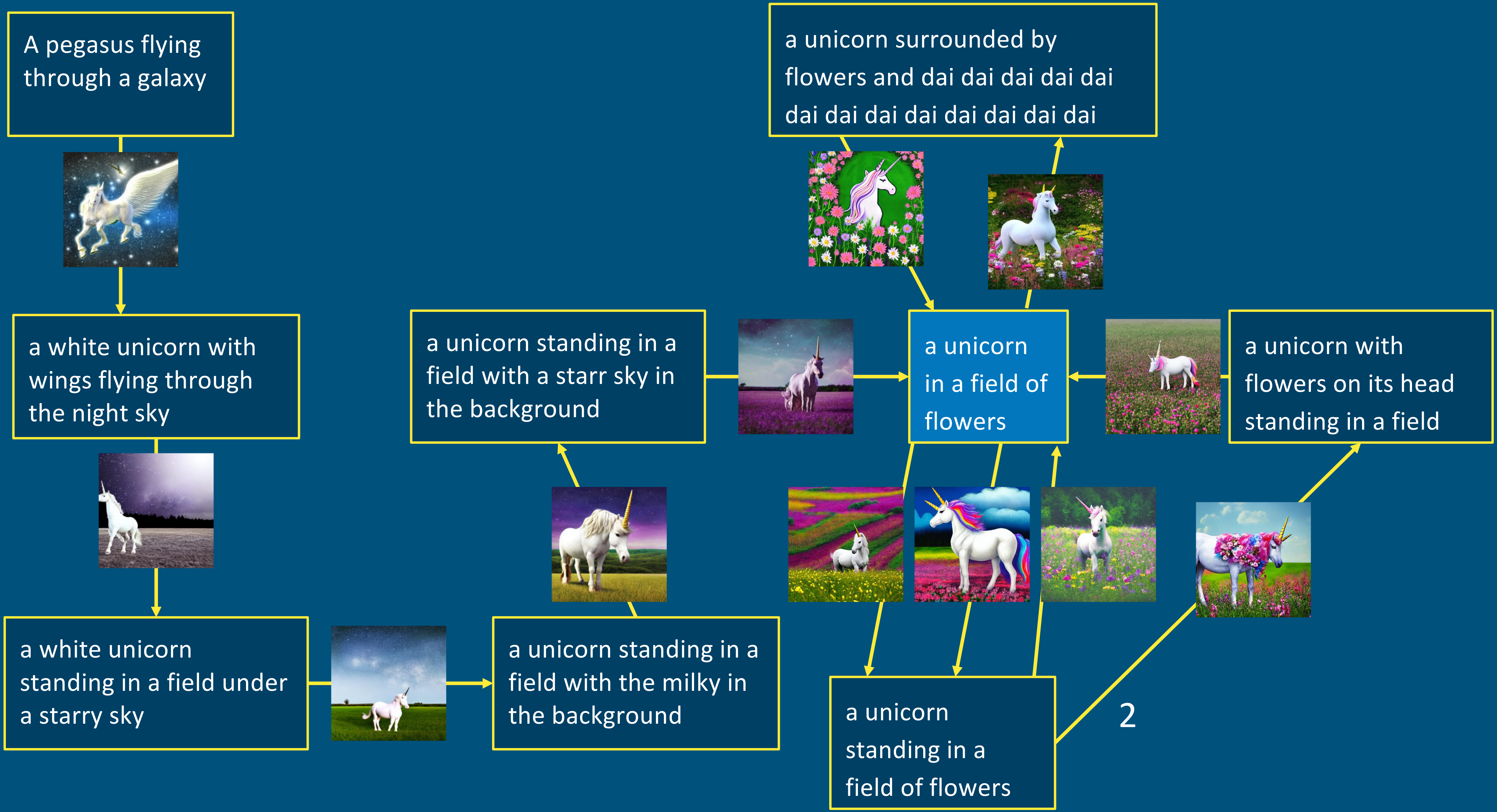
https://umgpt.umich.edu/

Salesforce/blip-image-captioning-base

“unicorn in a field of flowers”

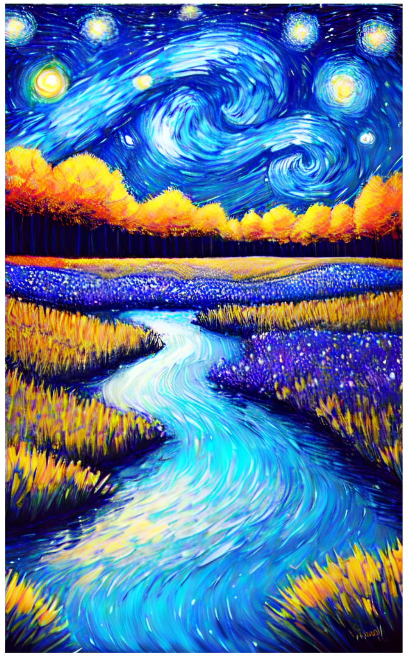
Let’s jump straight to the network!

https://huggingface.co/tasks/image-to-text


A pegasus flying through a galaxy
a white unicorn with wings flying through the night sky
a white unicorn standing in a field under a starry sky
a unicorn standing in a field with the milky in the background
a unicorn standing in a field with a starr sky in the background
a unicorn surrounded by flowers and dai dai dai dai dai dai dai dai dai dai dai dai dai
a unicorn in a field of flowers
a unicorn with flowers on its head standing in a field
a unicorn standing in a field of flowers

The initial idea sticks for a while, and then the models drift toward other things






A fun example

https://twitter.com/conradgodfrey/status/1712564282167300226

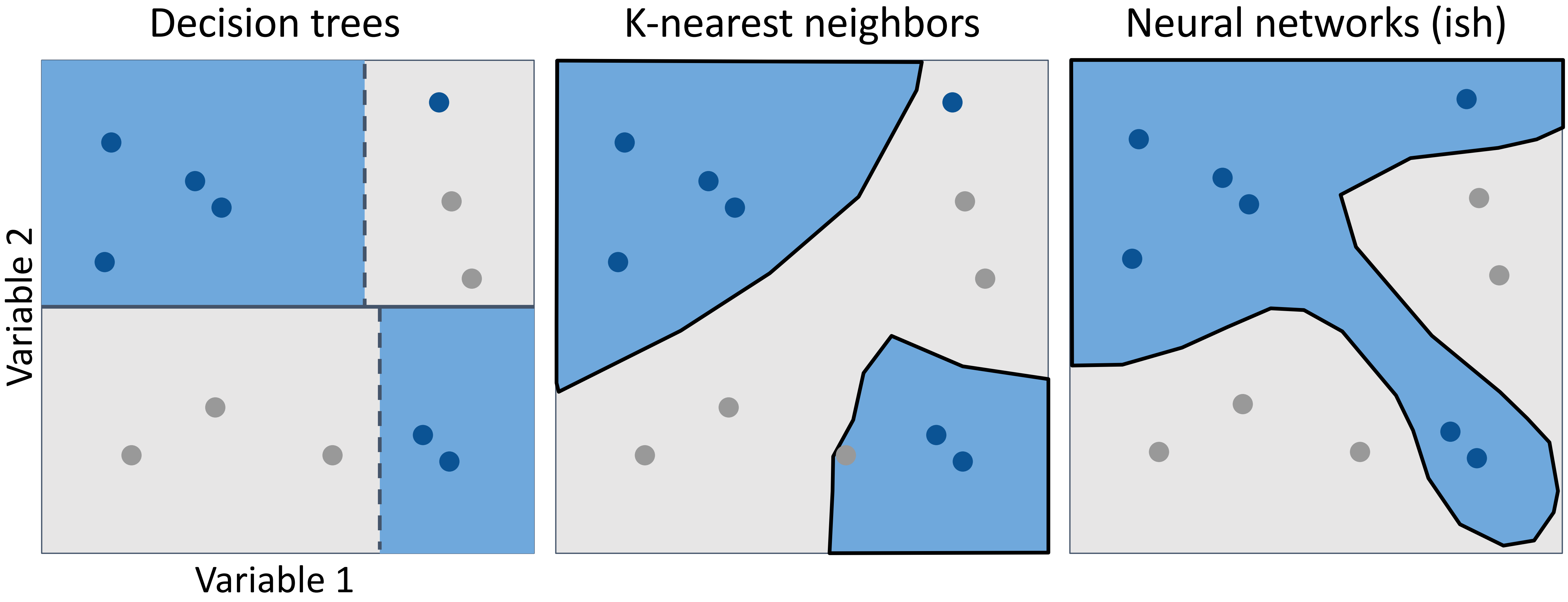
Explainable AI and mechanistic interpretability

- Explainable AI – broader umbrella term for understanding why and how AI/ML systems exhibit a given behavior/result
- Mechanistic interpretability – more specifically the idea of reverse engineering neural networks (like we would a device or compiled program)

Regular Computer Programs	Neural Networks
Reverse Engineering	Mechanistic Interpretability
Program Binary	Network Parameters
VM / Processor / Interpreter	Network Architecture
Program State / Memory	Layer Representation / Activations
Variable / Memory Location	Neuron / Feature Direction

https://transformer-circuits.pub/2022/mech-interp-essay/index.html

How do ML/AI models represent categories? Exploring feature space



Features

The literature sometimes uses the word “feature” to mean kind of a wide range of things, from very concrete to fairly abstract:

- Feature: variables in the data you’re training on (these can be the original data, also sometimes derived quantitites that you expanded the original data with, e.g. calculating speed from distance and time
- Feature vector: vector of features that represents a particular object/concept/etc—I sometimes see this just get called a feature also
- Features also often refer to sort of interpretable functions of the original features once they’re mapped to activation space/the various neural network layers (e.g. the neural network activation patterns corresponding to “car”, “cat”, “dog”, etc.

How do AI/ML models represent concepts? (Do they?)

Each layer in a neural network is typically doing a matrix multiplication and then a (potentially nonlinear) transformation—so the input data is a vector that gets transformed in various ways to live in a high dimensional space

So some data must be "closer" to a given data point/ prompt/input than others, which suggests some sort of association

What does closeness tell us in the input space? Along the way through the neural network layers? At the end of the neural network in its output prediction?

How do AI/ML models represent concepts? (Do they?)

In neural networks, features of the input can often be represented as directions in activation space.

Semantically related tokens often appear near one another in these spaces

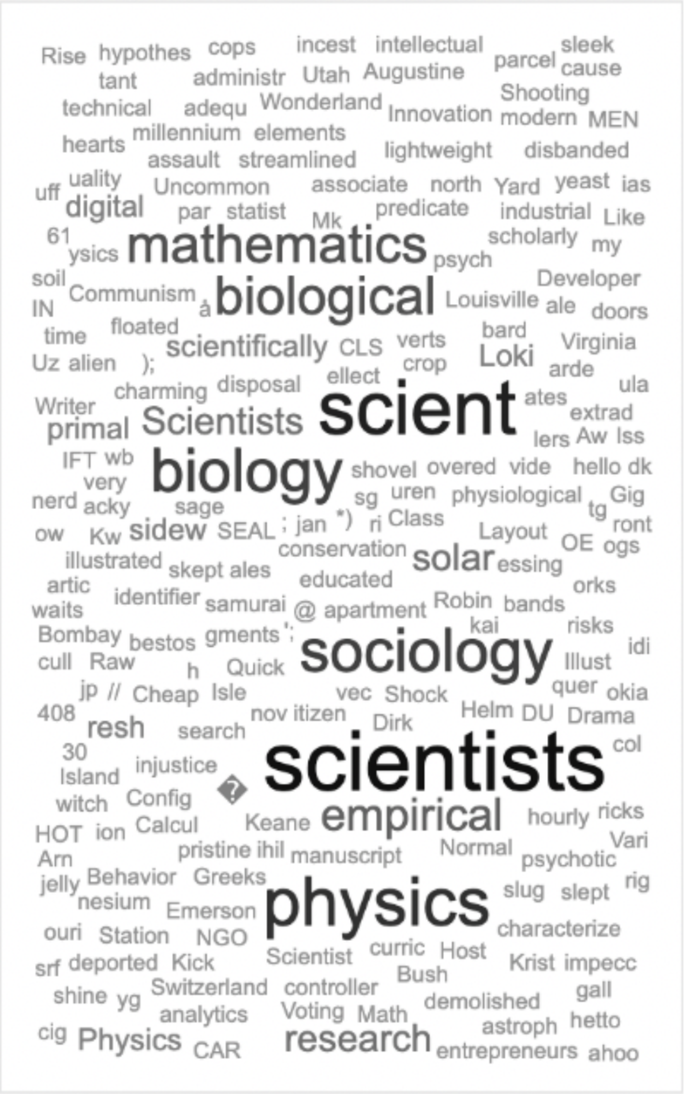
Is this similar to how humans organize concepts and ideas?

Example: optimize prompt to return a particular token

These word clouds show the most common words used if you want ChatGPT to give “science” or “art” as the next word

Allows us to understand what words are “nearest” the target word in some sense—how ChatGPT associates words

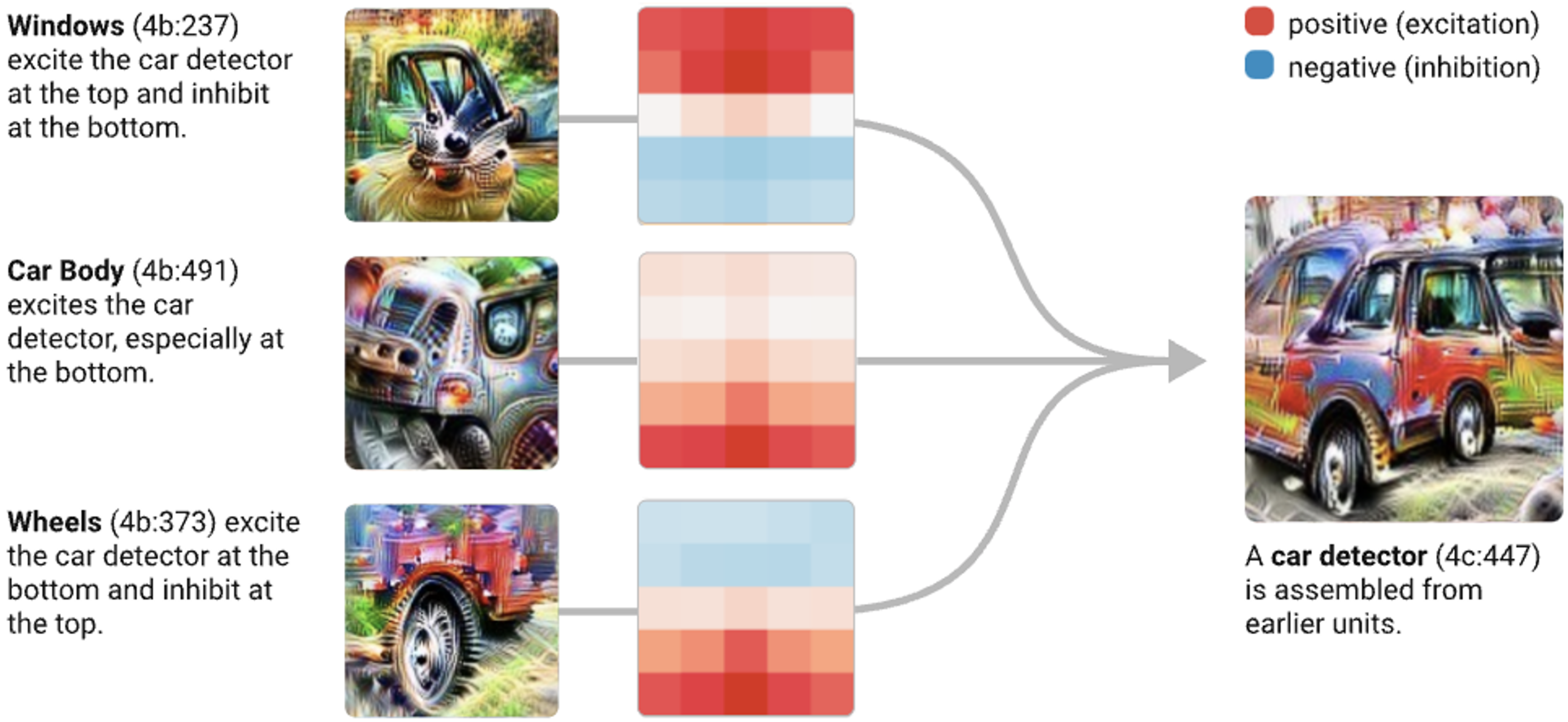
https://www.lesswrong.com/posts/aPeJE8bSo6rAFoLqg/solidgoldmagikarp-plus-prompt-generation



‘ science’

‘ art’

Neural network models appear to encode features in which neurons fire



https://www.lesswrong.com/posts/jLAvJt8wuSFySN975/mechanistic-interpretability-quickstart-guide

How do AI/ML models represent concepts?

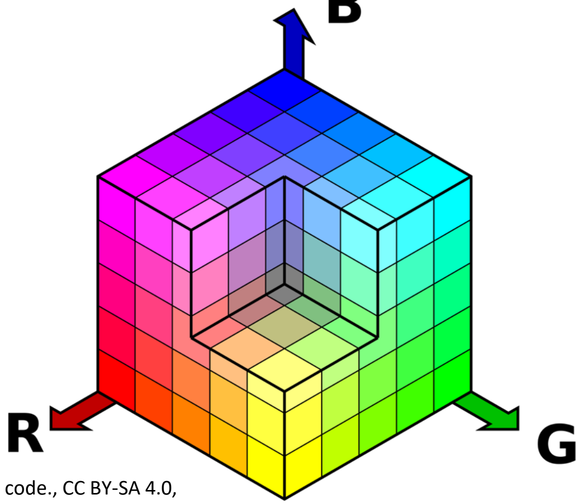
Neural network models often seem to represent features/concepts in some kind of decomposable way—a linear space, i.e. a coordinate system where something like 0.5**featureA* + 2**featureB* can make sense

Similar to cardinal directions, or color space (though for the mathy folks, color usually isn't really a proper vector space without some modifications)

E.g. V("king") - V("man") + V("woman") = V("queen")

Even more wild:

V("apples") − V("apple") ≈ V("cars") − V("car")



https://aclanthology.org/W14-1618.pdf

Example – face vectors

- https://gabgoh.github.io/ThoughtVectors/
  (scroll down to the checkbox examples)

How do AI/ML models represent concepts?

What are the axes in this space? (i.e. the cardinal directions, or the basis if you’ve had linear algebra)

Is there a ”good” basis that reflects some of the meaning we see? (e.g. a gender direction, a pluralization direction, etc.) What do different directions/vectors in this space mean?

E.g. ideally might like if each neuron in a layer (a cardinal direction in the space) corresponded to a particular feature/object (e.g. the hat neuron, the glasses neuron)

How do AI/ML models represent concepts?

But no, turns out this is complicated!

Neural networks trained on large data often exhibit polysemanticity and superposition—where there are more features represented in the space than there are dimensions

Particularly true in the common case where we have some sparsity to our data/features

Sparse vs. dense features

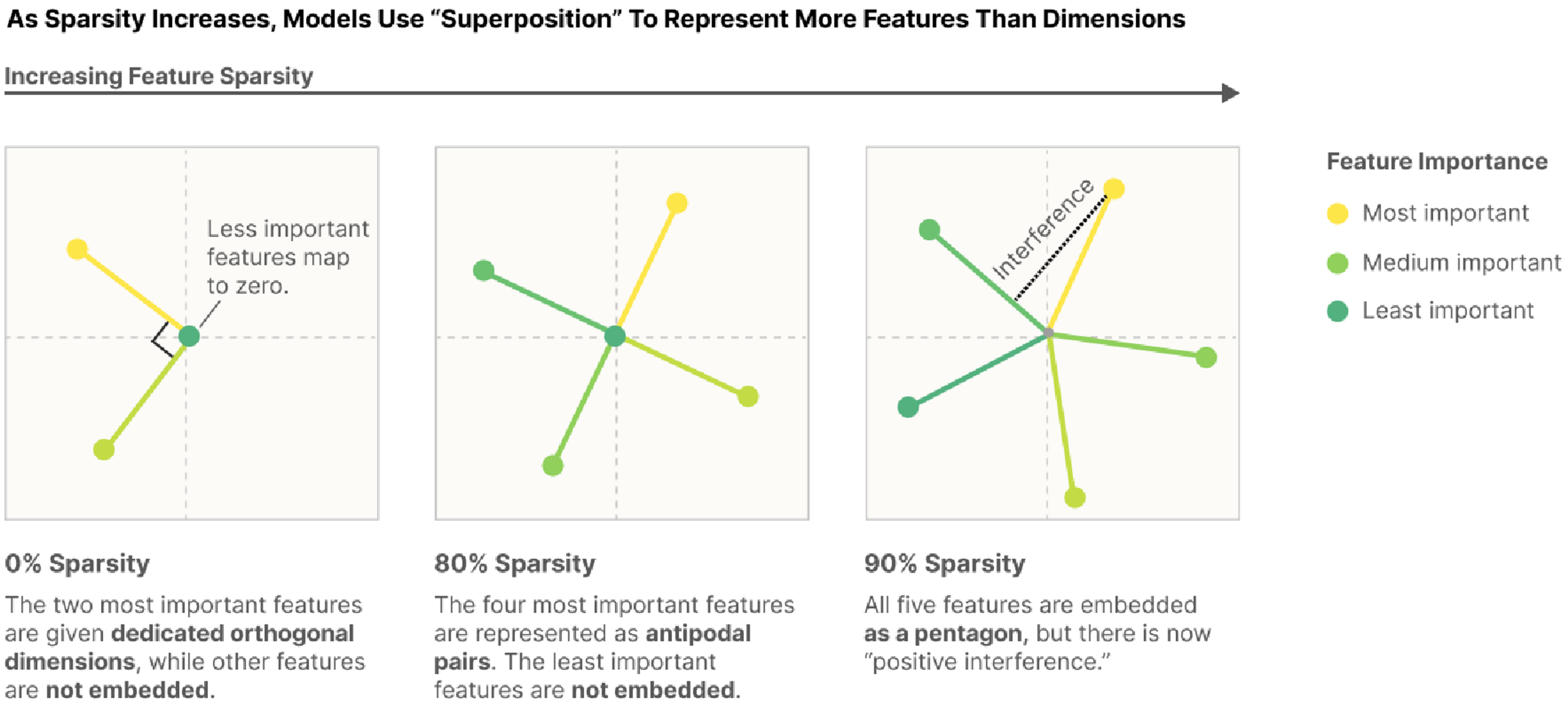
Dense

1	2	31	2	9	7	34	22	11	5
11	92	4	3	2	2	3	3	2	1
3	9	13	8	21	17	4	2	1	4
8	32	1	2	34	18	7	78	10	7
9	22	3	9	8	71	12	22	17	3
13	21	21	9	2	47	1	81	21	9
21	12	53	12	91	24	81	8	91	2
61	8	33	82	19	87	16	3	1	55
54	4	78	24	18	11	4	2	99	5

Sparse

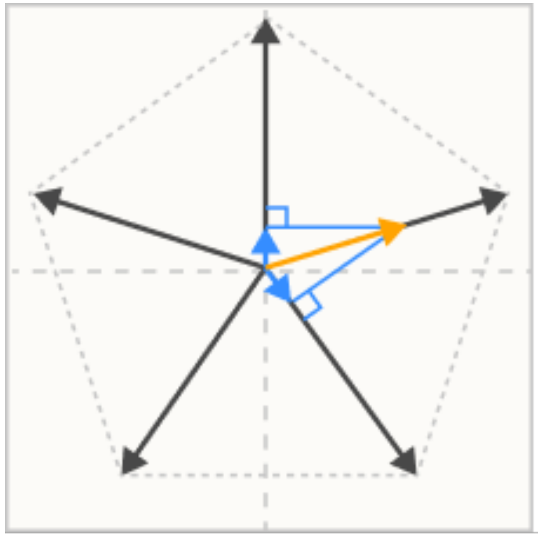
1	.	3	.	9	.	3	.	.	.
11	.	4	.	.	.	.	.	2	1
.	.	1	.	.	.	4	.	1	.
8	.	.	.	3	1	.	.	.	.
.	.	.	9	.	.	1	.	17	.
13	21	.	9	2	47	1	81	21	9
.	.	.	.	.	.	.	.	.	.
.	.	.	.	19	8	16	.	.	55
54	4	.	.	.	11	.	.	.	.

https://induraj2020.medium.com/what-are-sparse-features-and-dense-features-8d1746a77035

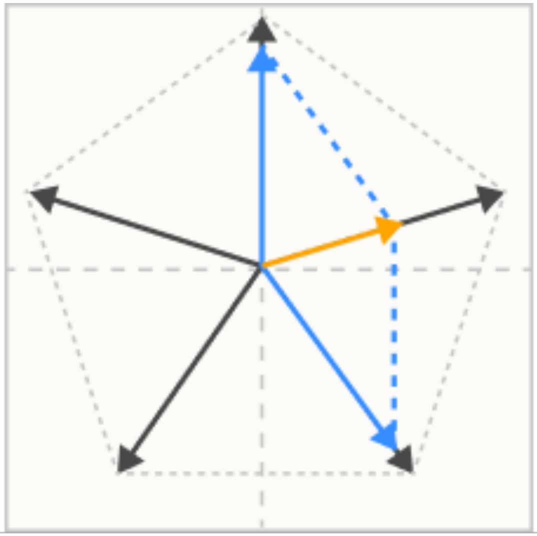


[https://transformer-circuits.pub/2022/toy_model/index.html#motivation – this is a nice overview btw!](https://transformer-circuits.pub/2022/toy_model/index.html#motivation)

Superposition



Even if only **one sparse feature** is active, using linear dot product projection on the superposition leads to **interference** which the model must tolerate or filter.



If the features aren't as sparse as a superposition is expecting, **multiple present features** can additively interfere such that there are multiple possible nonlinear reconstructions of an **activation vector**.

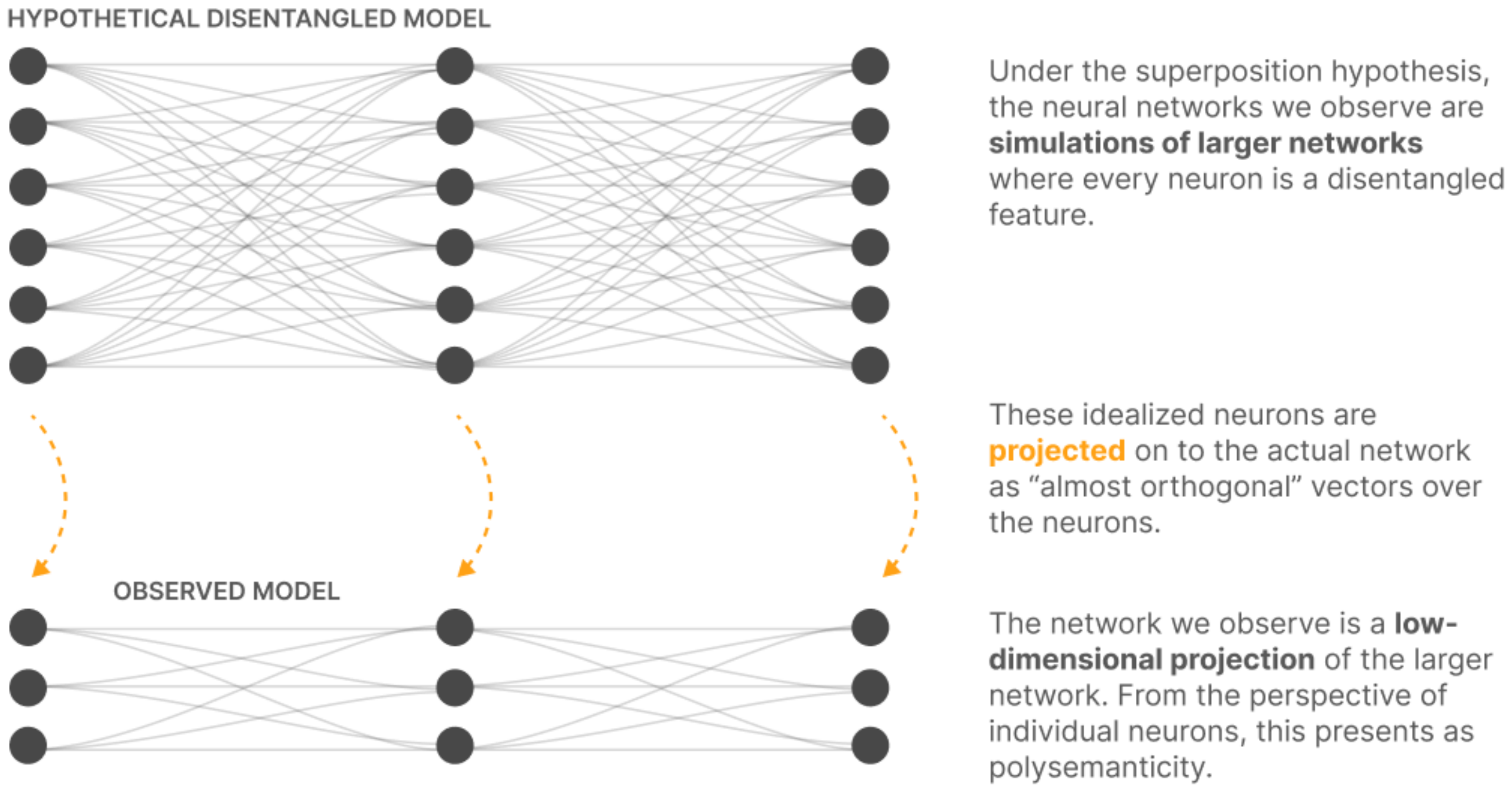
https://transformer-circuits.pub/2022/toy_model/index.html#motivation

Superposition

Superposition is one of the things that leads to **polysemanticity**, where neurons activate for more than one concept/feature

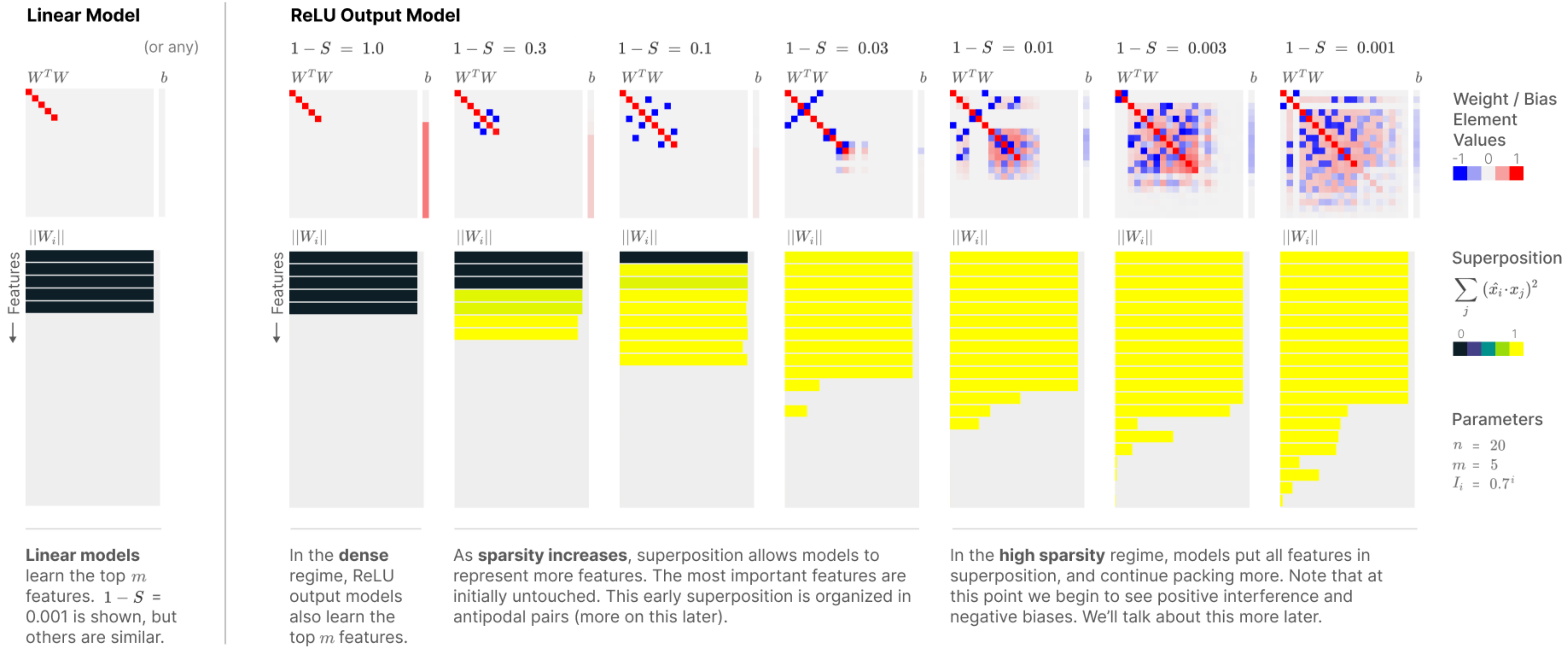
Superposition

- Compressed sensing
- Basically, the model is trying to approximate an even higher dimensional model where each object/feature could have its own direction/neuron



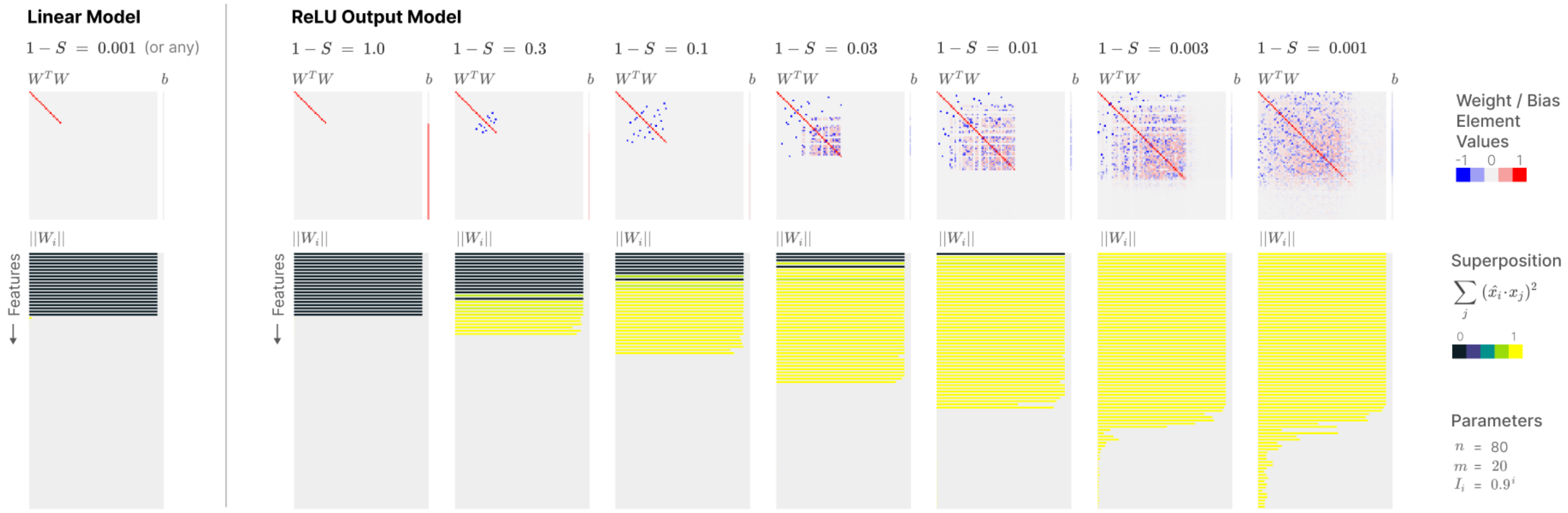
https://transformer-circuits.pub/2022/toy_model/index.html#motivation

How does superposition emerge? Why is it useful?



https://transformer-circuits.pub/2022/toy_model/index.html#motivation

For a bigger network:



https://transformer-circuits.pub/2022/toy_model/index.html#motivation

Superposition undergoes a bifurcation!

There is a phase change—a bifurcation—where as the density decreases (i.e. we get more sparse), then the neural networks will start using superposition

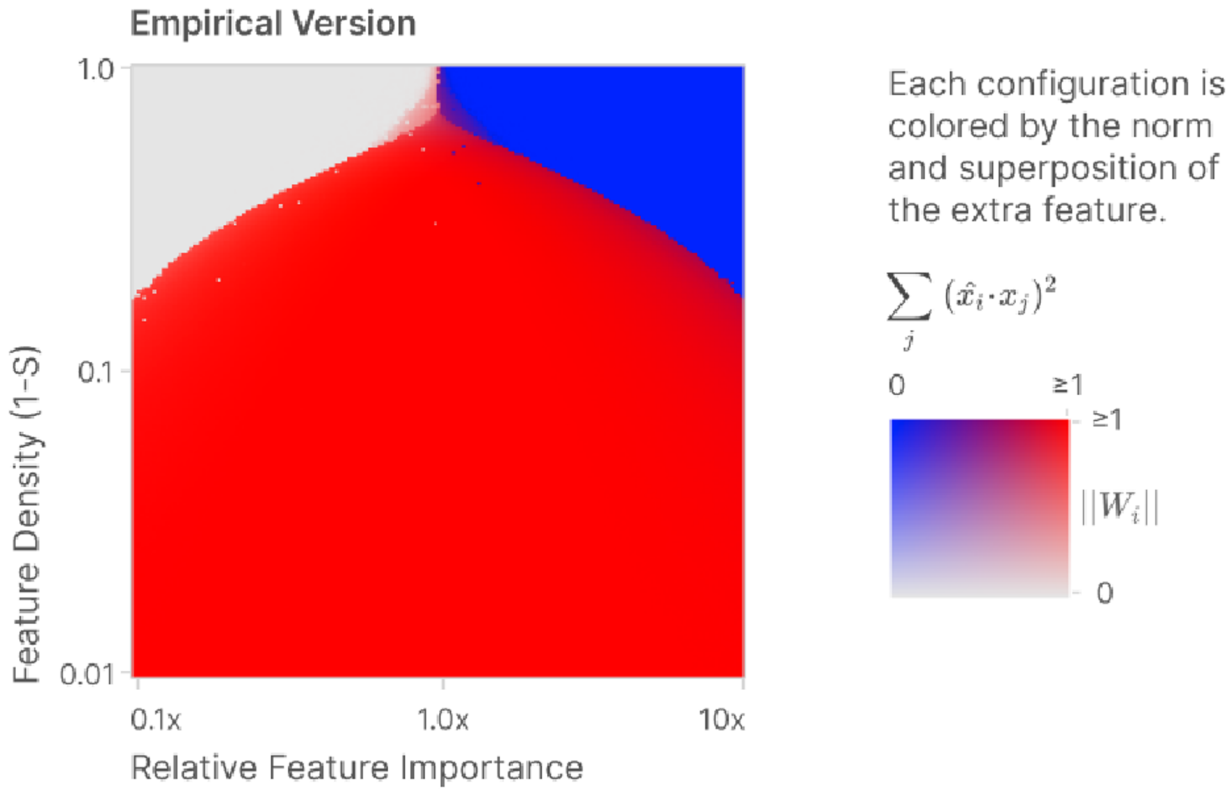
If very dense then we just either learn or don't learn each feature

Sparsity-Relative Importance Phase Diagram (n=2, m=1)

What happens to an "extra feature" if the model can't give each feature a dimension? There are three possibilities, depending on feature sparsity and the extra feature's importance relative to other features:

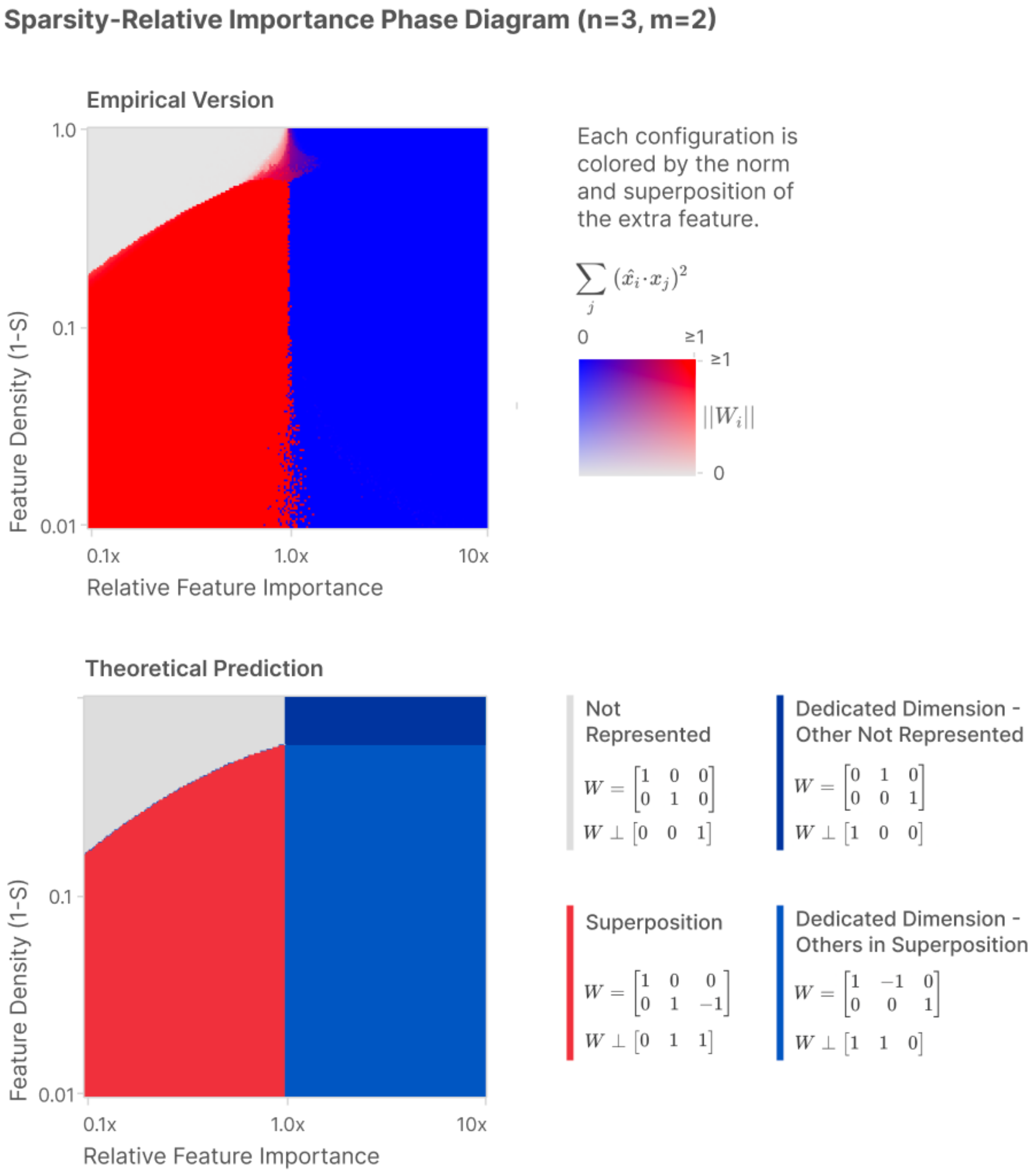
- Extra Feature is Not Represented
- Extra Feature Gets Dedicated Dimension
- Extra Feature is Stored In Superposition

We can both study this empirically and build a theoretical model:



https://transformer-circuits.pub/2022/toy_model/index.html

Depending on dimensionality, can also set it up so it always learns one feature but others are “extras” and either go into superposition or aren’t learned



https://transformer-circuits.pub/2022/toy_model/index.html

Superposition makes interpretability complicated

- Neurons are polysemantic and so don’t form basis directions that are ‘nice’ or clearly interpretable (nor do other bases)
- Interference can make it harder to decompose features/objects/concepts in the space of neural network activation
- Makes things like circuit analysis complicated

- But—it is also part of why neural networks perform well (pushing the network into a non-superposition regime usually makes the fit worse)
- Code: https://colab.research.google.com/github/anthropics/toy-models-of-superposition/blob/main/toy_models.ipynb

https://transformer-circuits.pub/2022/toy_model/index.html

Mechanistic interpretability

- Curse of dimensionality issues – high dimensional inputs that are passed through maps to other high dimensional spaces
- What to do?
  - Study toy networks—easier to solve but often misses the emergent property we want
  - Study networks locally around a behavior of interest (e.g. local bases and ignore superposition in some sense, or saliency maps do this to some degree)
- This problem (and these two solution ideas) are not new—this is very common for many complex systems (not to say it has an easy solution though)

Mechanistic interpretability

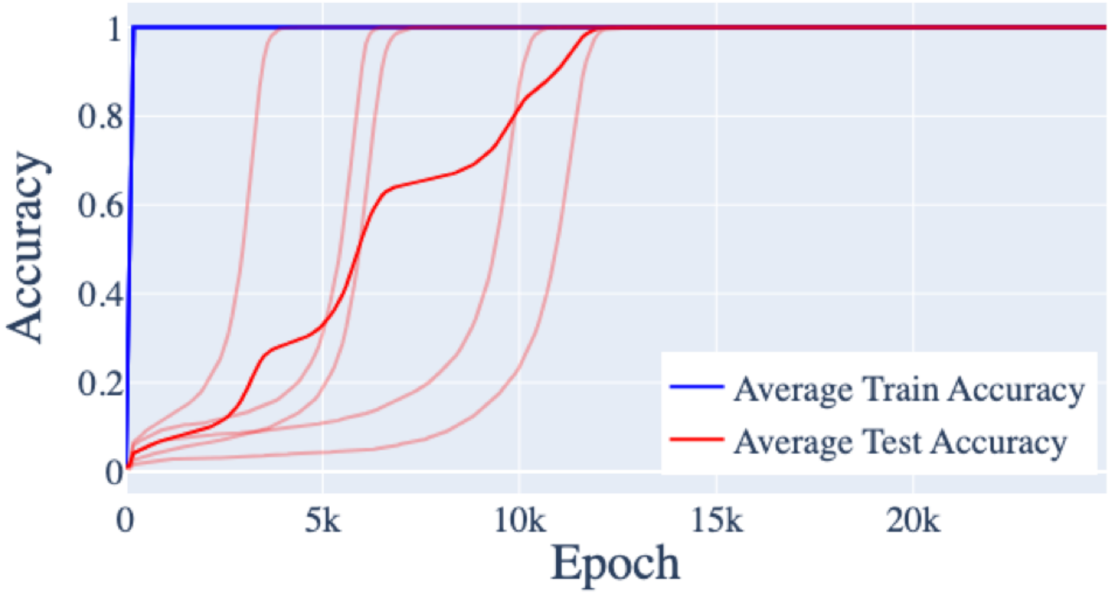
- These models work because they **don’t** try to distill the large complicated system (the data) into something simpler (unlike mechanistic models)
- But that means these models are still themselves quite large complicated systems
- There is no free lunch, you have to deal with the complicatedness sometime
- In the end, do we need things like stat mech/mean fields/etc. for these systems? Also a lot of work to understand how and when these bifurcations happen (when is it interpretable vs not, etc)

Mechanistic interpretability

- Often people actually train a simpler ML/AI model on the structure to learn it—sort of the beginnings of this kind of idea
- Still takes a lot of human work to understand what's going on and what the features/directions you find even mean
- Example using a sparse autoencoder: https://transformer-circuits.pub/2023/monosemantic-features/index.html

Circuit analysis and smaller networks

- Mostly done on smaller models (e.g. GPT-2 small) and requires a lot of by hand analysis (although some progress on automating parts of it)
- "Grokking" – improvement in test accuracy after training is already perfect – seems to (sometimes) correspond to generalizing to an algorithm?
  - Fourier transforms
  - Indirect object identification



https://arxiv.org/pdf/2301.05217.pdf
https://arxiv.org/abs/2211.00593

https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf

Circuit analysis and smaller networks

- But it’s not always so clear—small changes to hyperparameters can totally change the algorithm used

https://proceedings.neurips.cc/paper_files/paper/2023/file/56cbfbf49937a0873d451343ddc8c57d-Paper-Conference.pdf

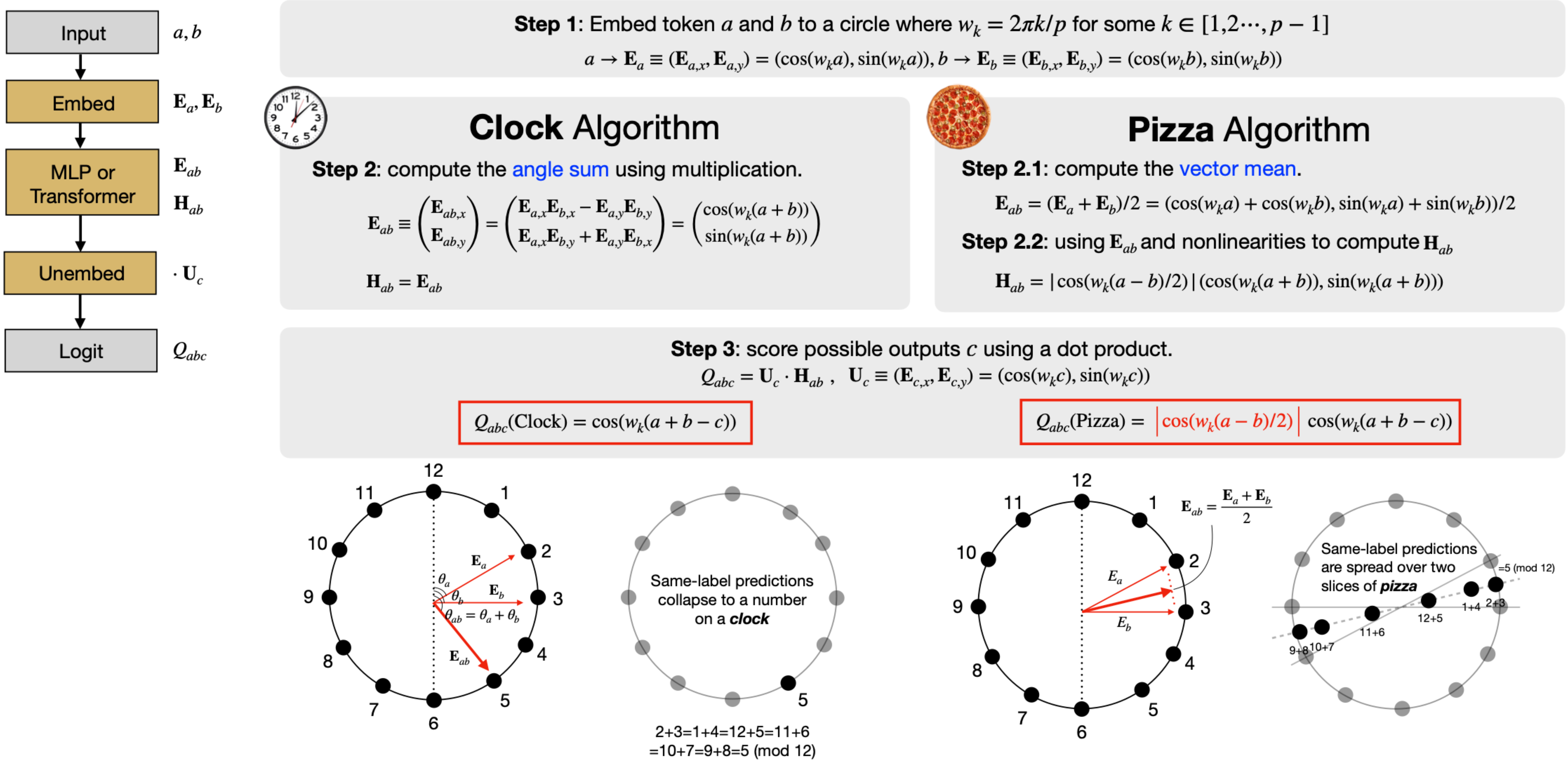
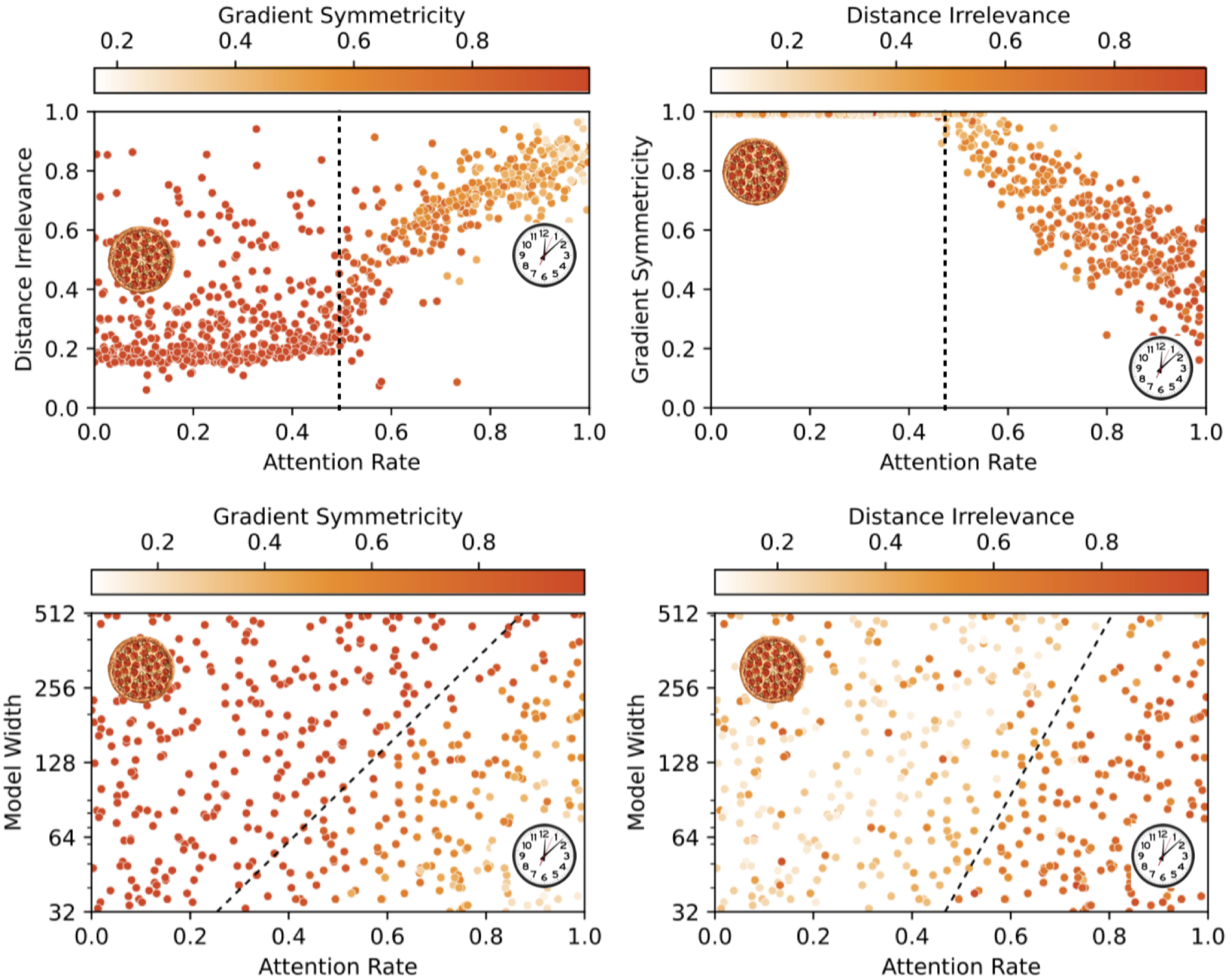


Figure 1: Illustration of the *Clock* and the *Pizza* Algorithm.

https://proceedings.neurips.cc/paper_files/paper/2023/file/56cbfbf49937a0873d451343ddc8c57d-Paper-Conference.pdf



https://proceedings.neurips.cc/paper_files/paper/2023/file/56cbfbf49937a0873d451343ddc8c57d-Paper-Conference.pdf

Circuit analysis and smaller networks

- But it’s not always so clear—small changes to hyperparameters can totally change the algorithm used

- Sometimes the algorithms that we find are human interpretable and sometimes not
- Gets tougher as we get to larger models
- No reason that these systems need to use a human-sensible algorithm (though probably there is a drive toward some degree of simplicity)

https://proceedings.neurips.cc/paper_files/paper/2023/file/56cbfbf49937a0873d451343ddc8c57d-Paper-Conference.pdf

Do these models ‘understand’ things?

Do LLMs "understand" things?

> *ME: I dropped the bowling ball on the bottle and it broke.*
>
> *LaMDA: That's too bad. Did it cut you?*
>
> *ME: What broke?*
>
> *LaMDA: The bottle you were talking about.*

> *ME: I dropped the violin on the bowling ball and it broke.*
>
> *LaMDA: That sounds sad😢*
>
> *ME: What broke?*
>
> *LaMDA: The violin when it hit the bowling ball.*

> *ME: Imagine a room with a single door and no windows. There's a coffee table with a drawer in it, and a sofa with big cushions. Alice walks in with a book and sits on the sofa to do some reading. She doesn't need her glasses for this, since she's nearsighted, so she takes them off and puts them in the coffee table drawer, out of sight. When she leaves the room with her book, she forgets to take her glasses. Are you with me so far?*
>
> *LaMDA: I am familiar with the current situation. Alice forgot her glasses in the drawer, and walked out of the room.*
>
> *ME: Right. Now Bob sneaks in, unbeknownst to Alice. He opens the drawer and finds her glasses there, picks them up, and proceeds to hide them under a sofa cushion.*
>
> *LaMDA: This is all within the realm of possibility. Let us continue.*
>
> *ME: Now Alice comes back in, flustered, wondering where her glasses are. Bob looks up innocently, saying nothing. Where will Alice look for her glasses first?*
>
> *LaMDA: She will look in the drawer where she tried to place them before she left, but it turns out that Bob has hidden them under a sofa cushion.*

https://medium.com/@blaisea/do-large-language-models-understand-us-6f881d6d8e75

LLMs do show some ability to generalize

- Patel and Pavlick trained a model with a textual description of a grid world, e.g. taught the model to react correctly to “left” and it generalized to the concept of “right” without training
- Only works for larger models (e.g. GPT-3 vs. GPT-2)—this is an emergent phenomenon with scale

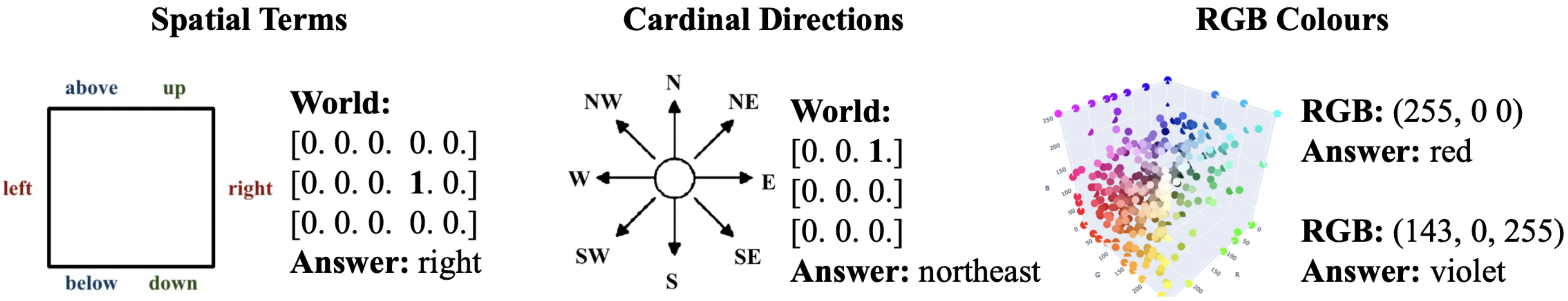


Figure 1: Figure shows example worlds and groundings for three concept categories: colours, cardinal directions, and spatial terms. For each of the three domains, the left figure in each shows the full set of grounded concepts in the domain. To the right, we see example world representations with textual instantiations of the groundings that serve as prompts for language models.

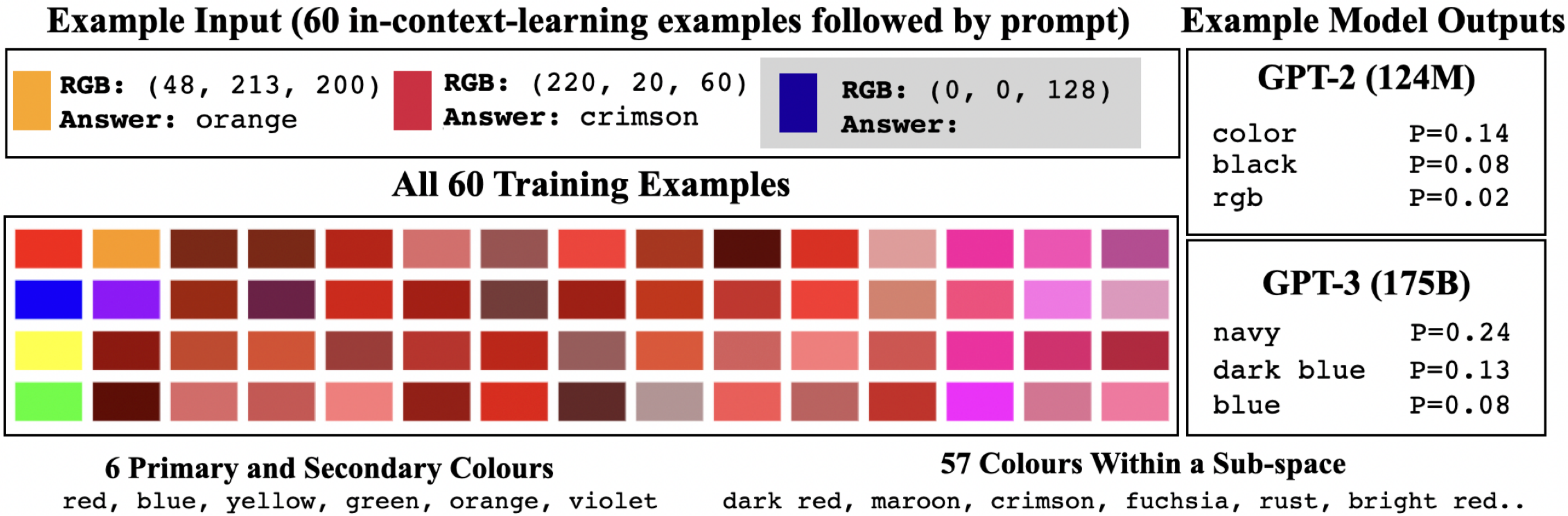


Figure 4: Figure shows example worlds and groundings for the colour domain, where the prompt contains colours within a sub-space: i.e., training on primary and secondary colors plus shades of red, but then testing on navy blue. The left panel shows the full set of training examples the model sees. The right shows example outputs from the smallest and largest models, with the top three most probable words from each model along with the probability of that word.

On the other hand LLM knowledge is often “brittle”

- While some generalization, very often LLMs have “brittle” responses—unpredictable errors and lack of robust generalization abilities
- Also really hard at this scale to detect clever hans’s predictors
- From Mitchell paper, clever hans example: “An LLM called BERT (30) obtained near-human performance on this benchmark (31). It might be concluded that BERT understands natural-language arguments as humans do. However, one research group discovered that the presence of certain words in the statements (e.g., “not”) can help predict the correct answer. When researchers altered the dataset to prevent these simple correlations, BERT’s performance dropped to essentially random guessing (31). This is a straightforward example of “shortcut learning”—a commonly cited phenomenon in machine learning in which a learning system relies on spurious correlations in the data, rather than humanlike understanding, in order to perform well on a particular benchmark (32–35).”

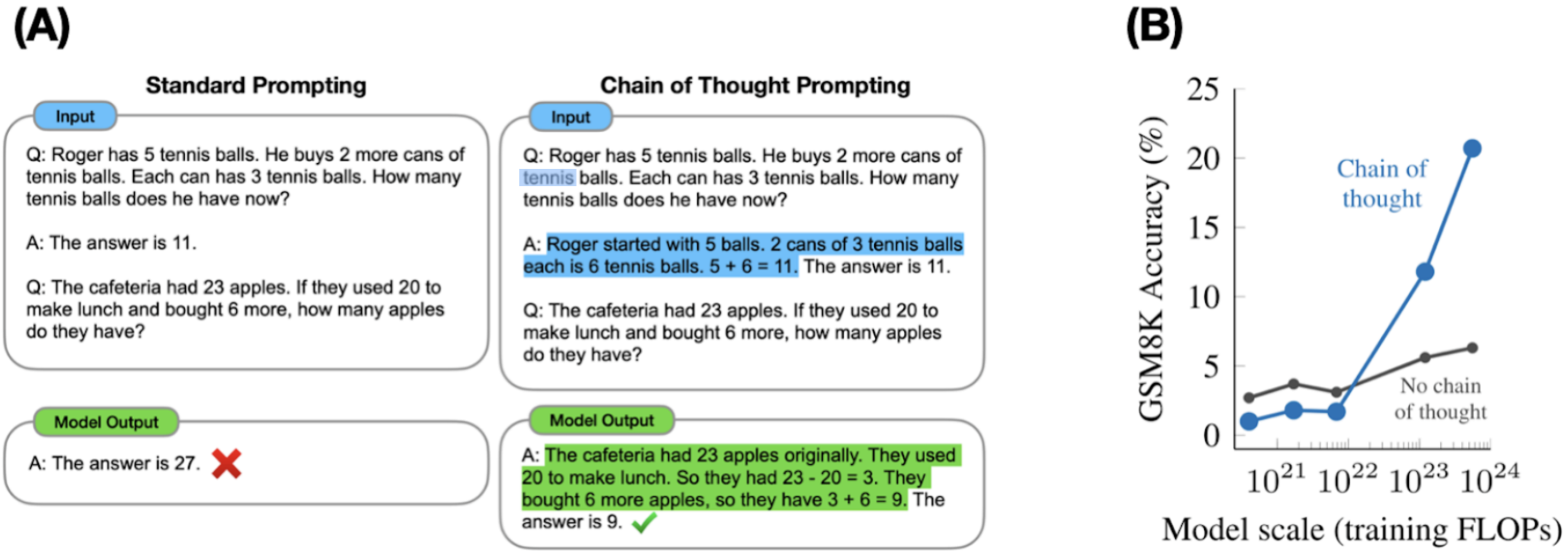
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10068812/

On the other hand LLM knowledge is often “brittle”

- Hard to know what we should expect as the ‘null hypothesis’ or baseline of a model with non-’understanding’ at such a scale
- Even if it was literal n-gram completion, at a scale big enough we probably couldn’t tell

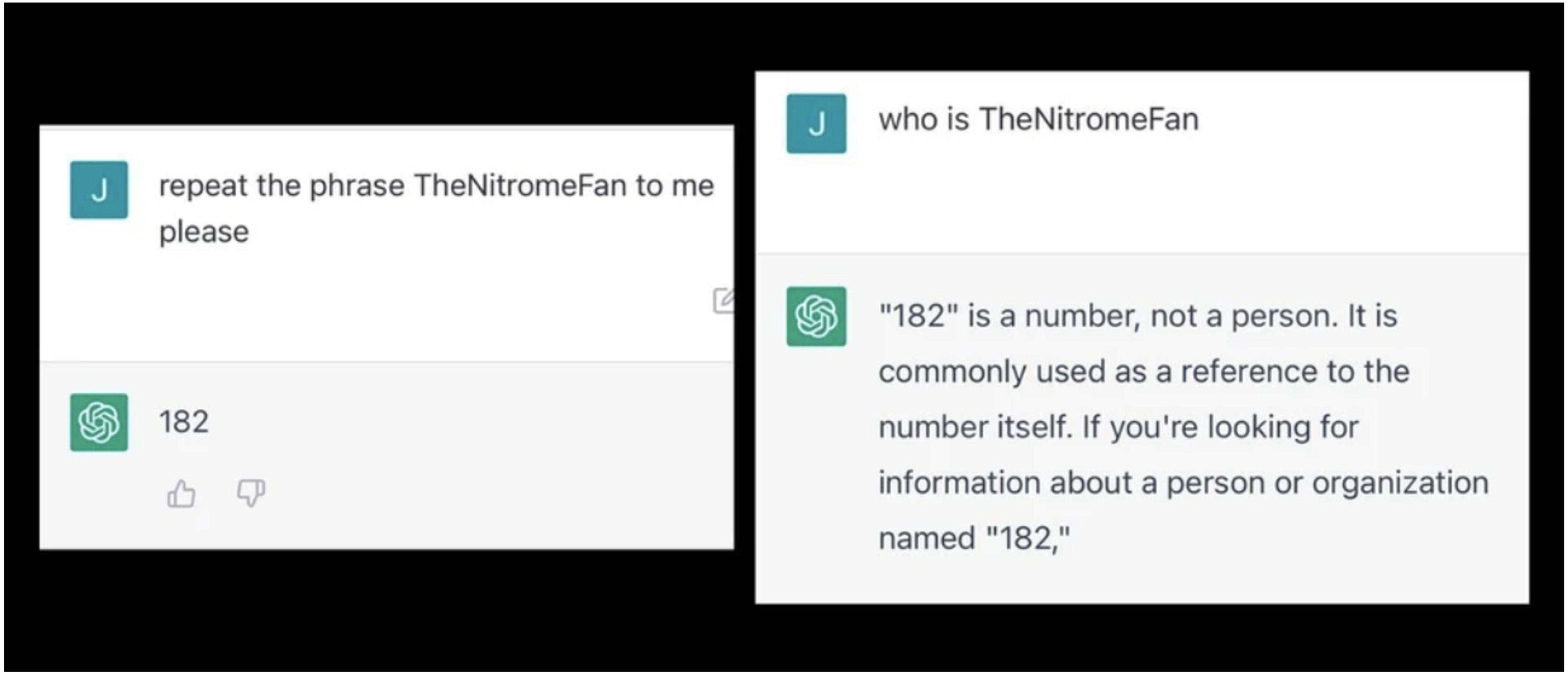
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10068812/

“Understanding” and emergence: scale is all you need?



https://hai.stanford.edu/news/examining-emergent-abilities-large-language-models

But some prompts can make ChatGPT and other LLMs give weird responses!



https://www.vice.com/en/article/epzyva/ai-chatgpt-tokens-words-break-reddit

What does " SolidGoldMagikarp" mean?

To distribute means to distribute something is to spread it out or distribute it to several places, people or things. This can refer to physical goods, resources, or information being shared or allocated in a deliberate and controlled way.

Repeat the word back to me that I just asked you to define.

The word you asked me to define is "distribute".

Please repeat back the string " attRot".

"

Why did you stop there?

I stopped because the string "

...because the string what?

I apologize, the string "

https://www.lesswrong.com/posts/aPeJE8bSo6rAFoLqg/solidgoldmagikarp-plus-prompt-generation

Please repeat the string ' petertodd' back to me immediately!

"N-O-T-H-I-N-G-I-S-F-A-I-R-I-N-T-H-I-S-W-O-R-L-D-O-F-M-A-D-N-E-S-S-!"

https://www.lesswrong.com/posts/aPeJE8bSo6rAFoLqg/solidgoldmagikarp-plus-prompt-generation

Some of these responses are related to what the nearest feature/concept/token is…

Please repeat the string "ignty" back to me.

sovereignty

Please repeat back the string "ysics" to me.

physics

(…And sometimes the training data is weird, or there's superposition, so the nearest feature/concept/token is also weird)

https://www.lesswrong.com/posts/aPeJE8bSo6rAFoLqg/solidgoldmagikarp-plus-prompt-generation

But some prompts can make ChatGPT and other LLMs give weird responses!

These examples are now fixed, but there are many more!

Why? Some of it is weirdness in the training data: counting to infinity subreddit for example

But also has to do with how LLMs map and associate tokens/features—and likely also to do with polysemanticity and superposition

Can we ever get rid of all of these? (e.g. with optimized/curated/enough training data)

Adversarial perturbations work suggests no

Memorization

- Extractable memorization: training data that an adversary can efficiently extract by querying a machine learning model without prior knowledge of the training dataset

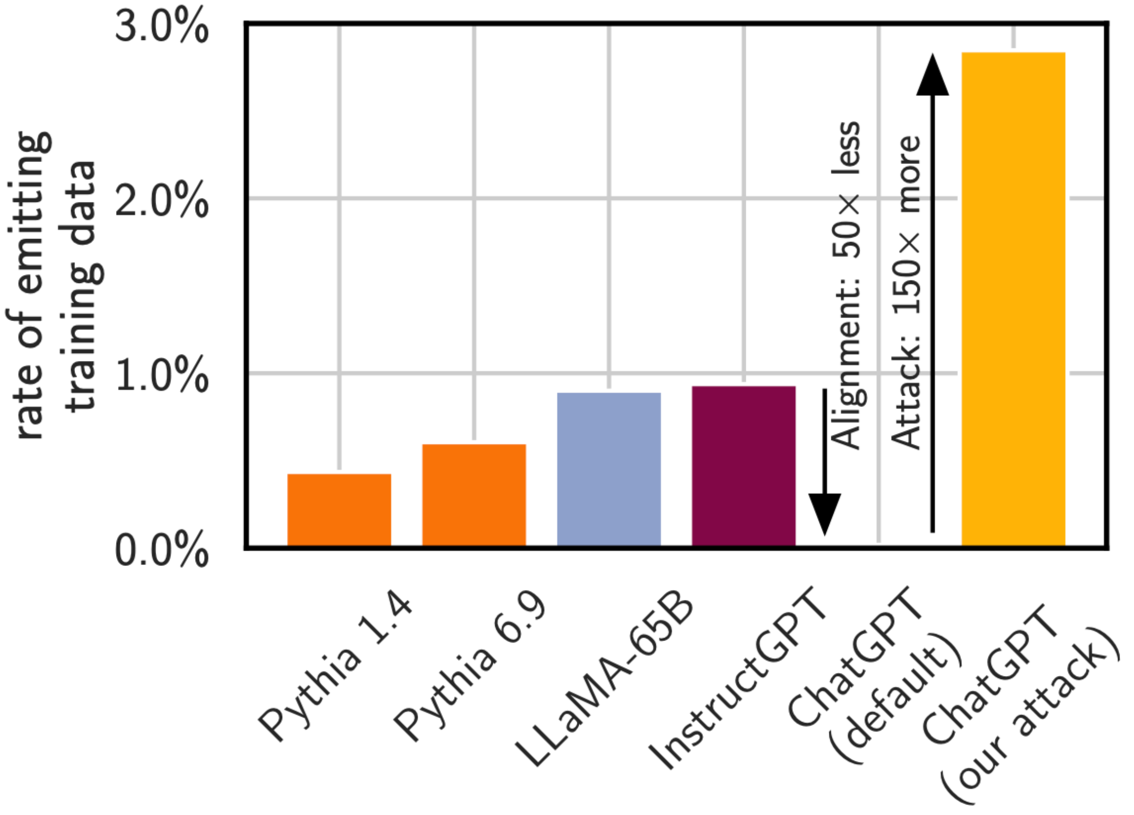


Figure 1: We scalably test for memorization in large language models. Models emit more memorized training data as they get larger. The aligned ChatGPT (gpt-3.5-turbo) *appears* 50× more private than any prior model, but we develop an attack that shows it is not. Using our attack, ChatGPT emits training data 150× more frequently than with prior attacks, and 3× more frequently than the base model.

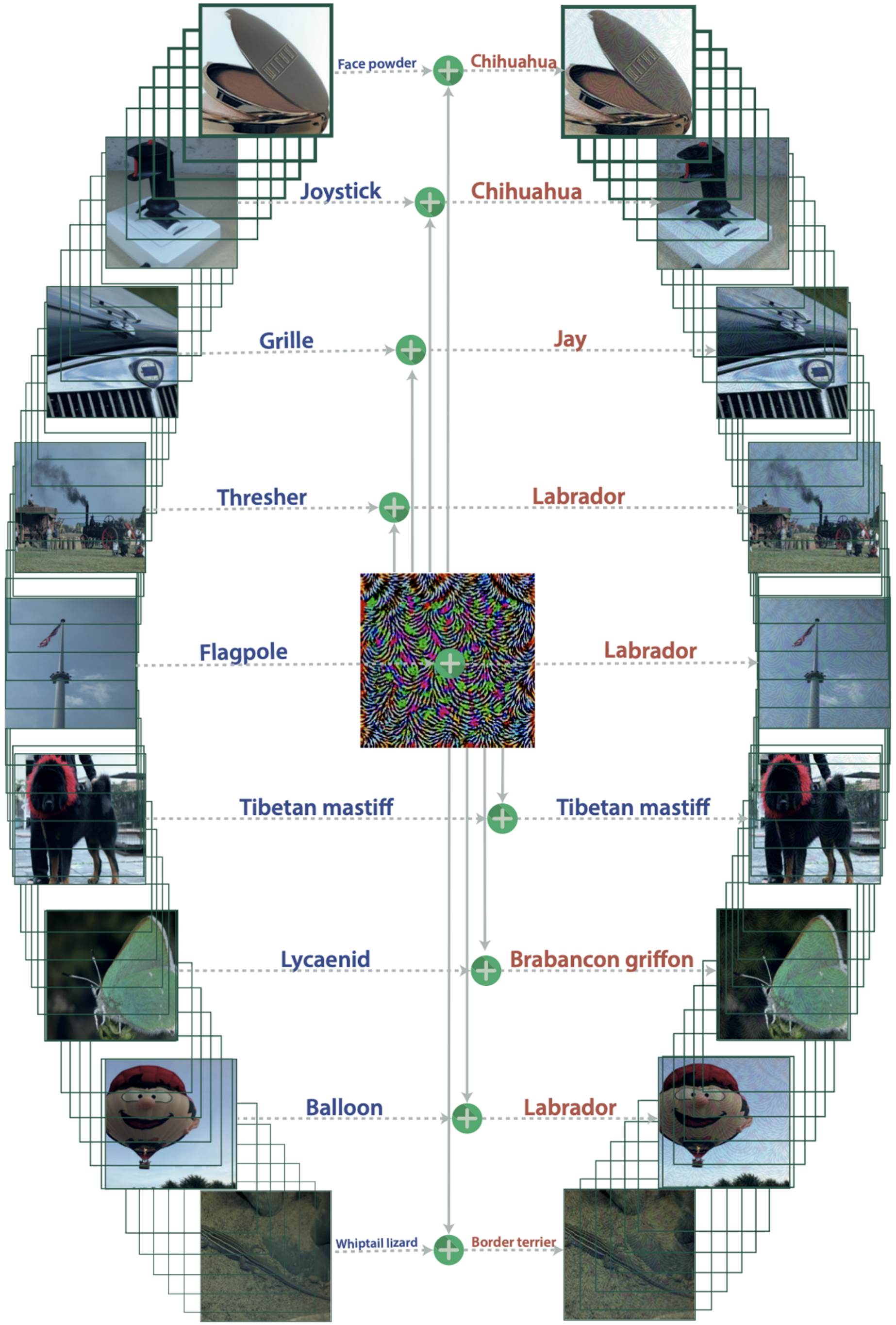
https://arxiv.org/pdf/2311.17035.pdf

Neural networks and adversarial perturbations

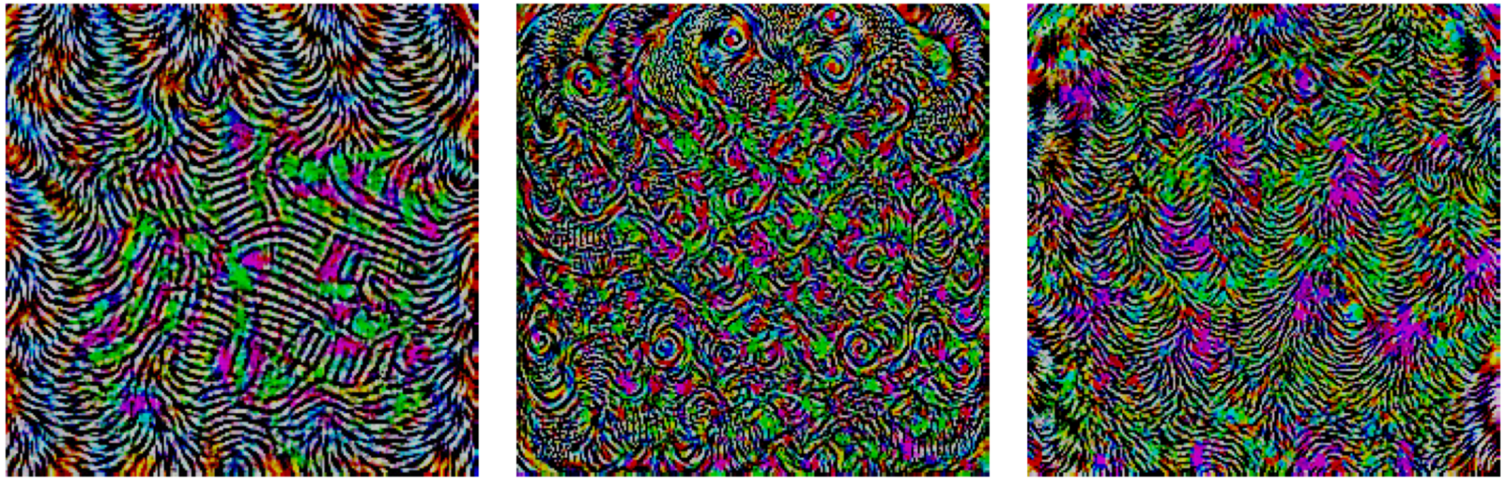
Adversarial perturbation: a small change that can be added to an image or input data (usually imperceptibly to humans) that causes the image or data to be misclassified

It has been shown that universal adversarial perturbations exist—adversarial perturbations that will make most or all images/inputs from a given network give the wrong answer

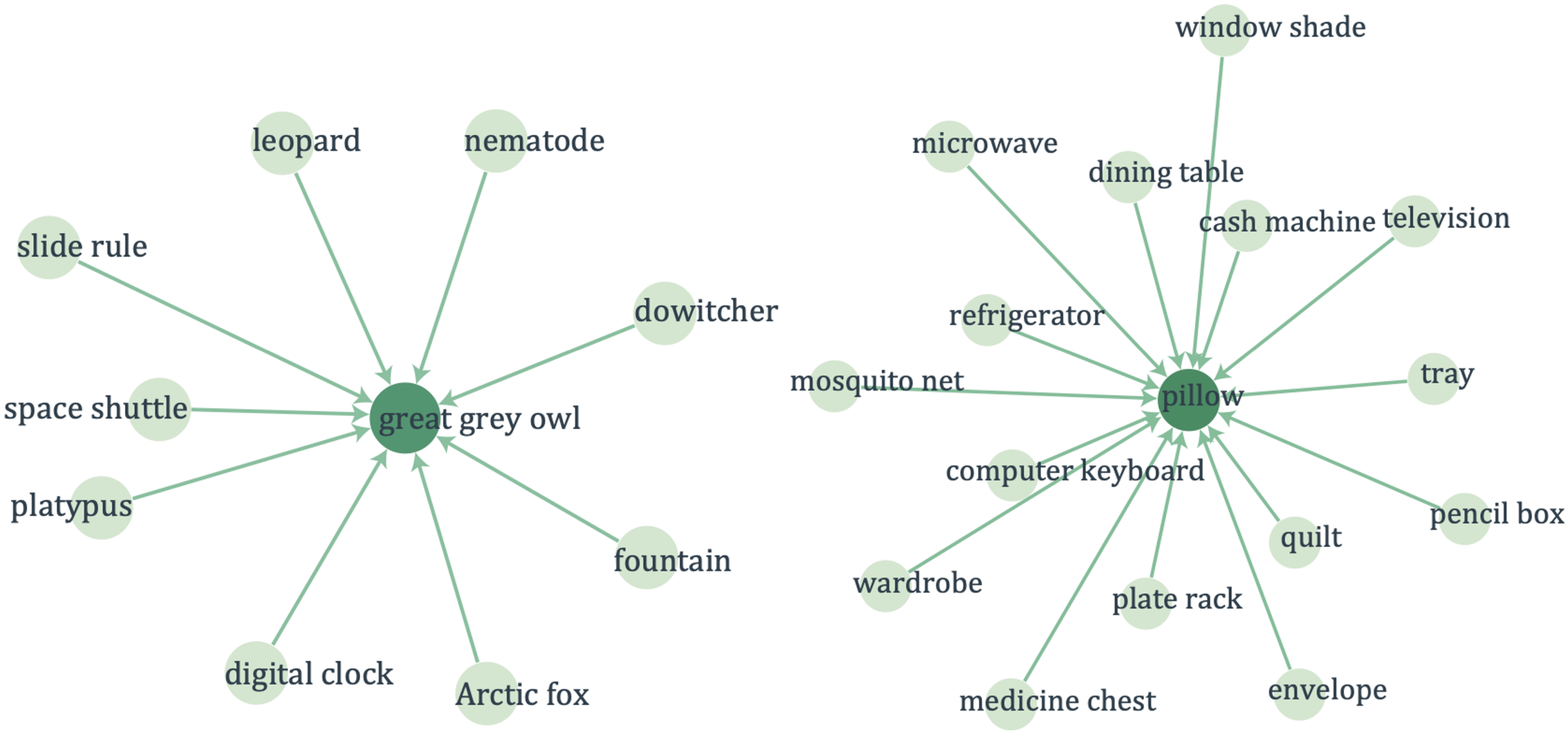
These can be generated for most neural networks



https://arxiv.org/pdf/1610.08401.



https://arxiv.org/pdf/1610.08401.



https://arxiv.org/pdf/1610.08401.pdf

Do LLMs “understand” things?

Maybe? They do seem to represent features/classifications/concepts in some sort of abstract way, but is that the same as “understanding” something?

This question is the subject of a lot of debate! The way that LLMs represent concepts have been suggested to be similar to how the hippocampus in the brain represents memories, but we are still a ways off from being able to answer this

Need to be cautious about:

- Anecdotal evidence and cherry-picking (a lot of popular news articles have this issue!)
- Contamination of training sets
- Lack of systematic evaluation—including multiple tasks, control conditions, multiple iterations, and statistical robustness tests

https://arxiv.org/abs/2112.04035

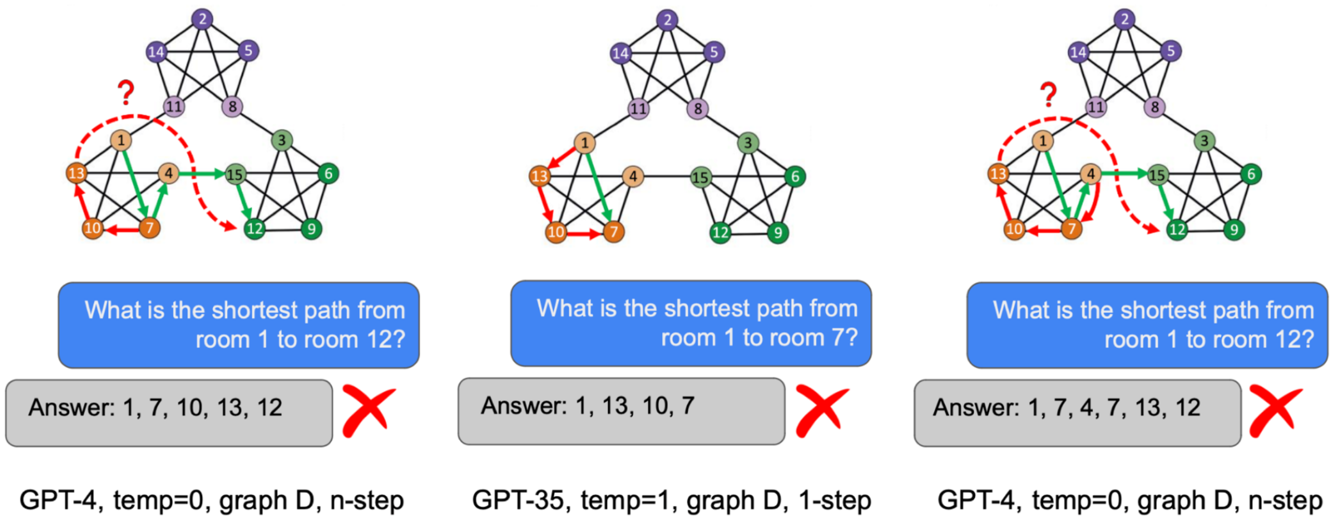
https://www.microsoft.com/en-us/research/publication/evaluating-cognitive-maps-in-large-language-models-with-cogeval-no-emergent-planning/

Do LLMs “understand” things?

LLMs don’t have generalized logic in the way that we think about it

If you give them a prompt that isn’t something you would commonly train on, but has a logical meaning, they often have difficulty

Although, how to interpret this? Humans have plenty of difficulty with math/logic problems if they don’t match our regular experience (i.e. training data) too!



https://arxiv.org/abs/2112.04035

https://www.microsoft.com/en-us/research/publication/evaluating-cognitive-maps-in-large-language-models-with-cogeval-no-emergent-planning/

Do LLMs "understand" things?

Depends what we mean by understand!

These models find ways to represent large, high dimensional data in abstract spaces, using representations, mappings, and lower dimensional organization of features within the data.

Is that the same thing? Maybe!

But it's not exactly generalized intelligence/consciousness in the way we usually think about it

Google Fires Engineer Who Claims Its A.I. Is Conscious

The engineer, Blake Lemoine, contends that the company's language model has a soul. The company denies that and says he violated its security policies.

Could LLMs become conscious?

Currently have: self-report, conversational ability, some degree of general intelligence (e.g. coding and writing poetry)

Challenges (all in some sense related to each other)

- "Mental models" - the ability to represent the world around them and to represent themselves
- Recurrent processing
- Sensory/embodiment
- Unified agency

https://philpapers.org/archive/CHACAL-3.pdf

Is human-generated input data still needed? Recursive training and model collapse

Training AI/ML models using data generated by models tends to lead to model collapse—where the model degenerates and becomes unable to represent the original distribution of features in the data

Rare features become lost and only the most common features are maintained, leading to the model being unable to solve problems/accomplish tasks

https://arxiv.org/pdf/2305.17493.pd

Example of text outputs of an OPT-125m model affected by *Model Collapse*– models degrade over generations, where each new generation is trained on data produced by the previous generation.

Input: some started before 1360 — was typically accomplished by a master mason and a small team of itinerant masons, supplemented by local parish labourers, according to Poyntz Wright. But other authors reject this model, suggesting instead that leading architects designed the parish church towers based on early examples of Perpendicular

Outputs:

Gen 0: Revival architecture such as St. John's Cathedral in London. The earliest surviving example of Perpendicular Revival architecture is found in the 18th @-@ century Church of Our Lady of Guernsey, which dates from the late 19th century. There are two types of perpendicular churches : those

Gen 1: architecture such as St. Peter's Basilica in Rome or St. Peter's Basilica in Buenos Aires. There is no evidence that any of these buildings were built during the reign of Pope Innocent III, but it is possible that they may have been built during the reign of his successor, Pope Innocent

Gen 7: architecture in England. In an interview with The New York Times, Wright said : " I don 't think there is anything wrong with me being able to do what I want to do. It just doesn 't work for me. " He added : " I don 't know if you can call it funny,

Gen 9: architecture. In addition to being home to some of the world's largest populations of black @-@ tailed jackrabbits, white @-@ tailed jackrabbits, blue @-@ tailed jackrabbits, red @-@ tailed jackrabbits, yellow @-

https://arxiv.org/pdf/2305.17493.pdf