

Towards understanding the (conceptual) structure of LLMs



Roma Patel
Google DeepMind

CogSci, July 2024

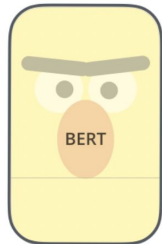
01 The current state of LLMs

01

The current state of LLMs



[OpenAI, 2022]



[Devlin et. al., 2018]

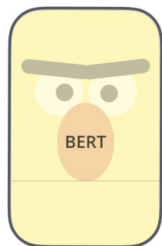


[Google DeepMind, 2023]

Human-like performance on benchmark language tasks



Can generate fluent, coherent and human-like text (QA, summarisation)



Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings.

[Brown et. al., 2020]



Can few-shot learn NLU tasks: entailment, sentiment, grammatical corrections

Poor English input: The patient was died.
Good English output: The patient died.

Poor English input: We think that Leslie likes ourselves.
Good English output: We think that Leslie likes us.

Poor English input: Janet broke Bill on the finger.
Good English output: Janet broke Bill's finger.

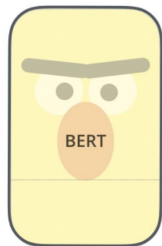
Poor English input: Mary arranged for, in St. Louis, John to rent a house cheap.
Good English output: Mary arranged for John to rent a house in St. Louis.

Context →	The bet, which won him dinner for four, was regarding the existence and mass of the top quark, an elementary particle discovered in 1995. question: The Top Quark is the last of six flavors of quarks predicted by the standard model theory of particle physics. True or False? answer:
Target Completion →	False

Figure G.31: Formatted dataset example for RTE

[Wang et. al., ICLR 2019]

Human-like performance on benchmark language tasks



Can reconstruct WordNet structure and learn all word relations!

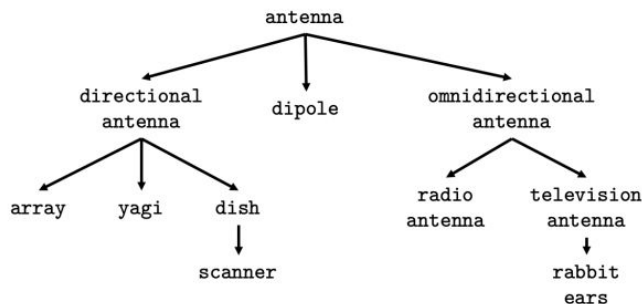
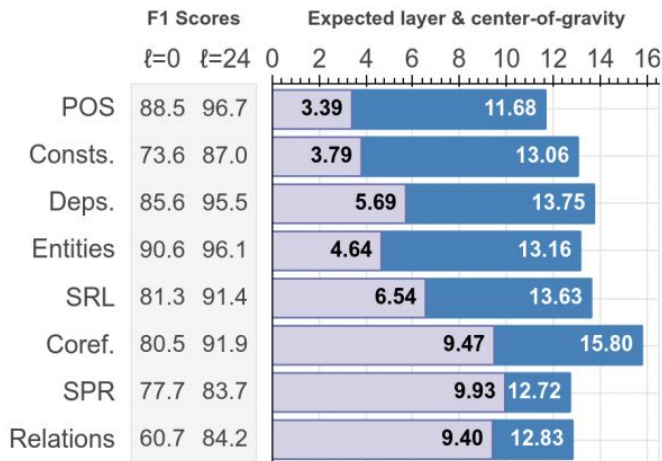


Figure 1: *An example subtree from the WORDNET hierarchy.*

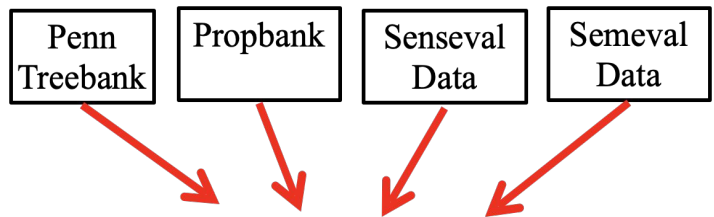
[Chen, Lin and Klein, 2021]

Can solve most classical NLP tasks!

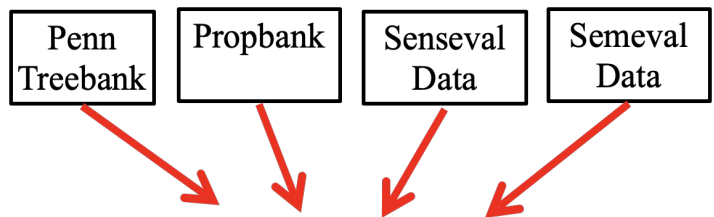


[Tenney, Das and Pavlick, 2019]

A very different type of learning from humans



A very different type of learning from humans



As humans, we benefit and learn from many things that are not just streams of text data—vision, sound, actions, interactions, touch...



That's a nice green block there, Anna!

Criticism against “form”-only learning

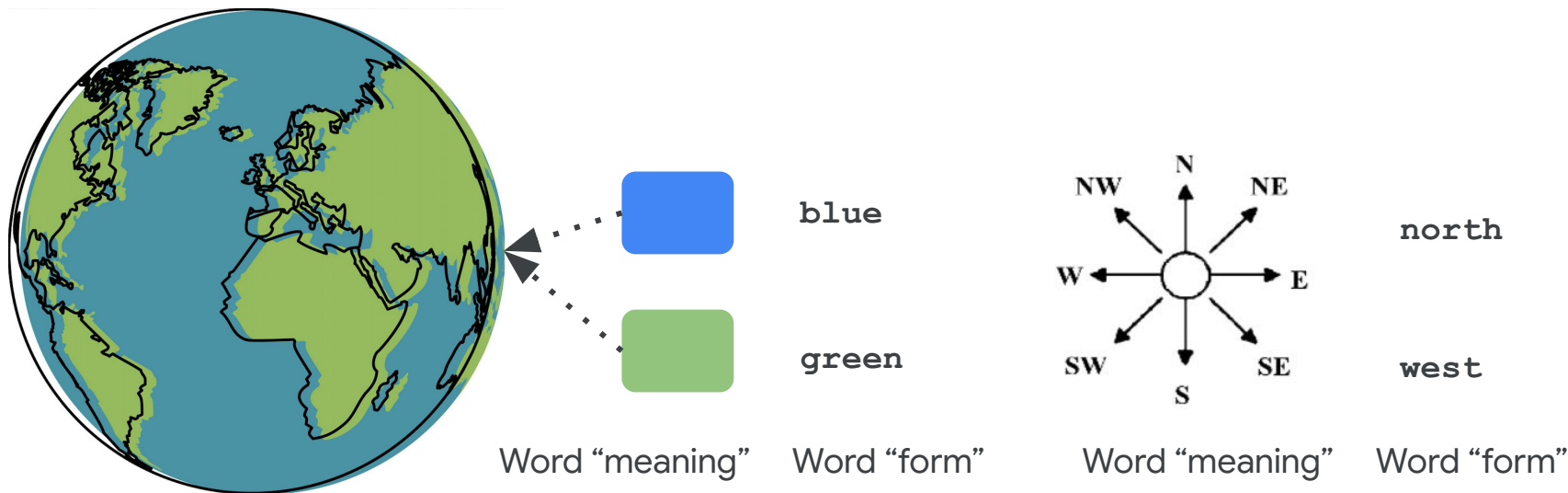
“We argue that the language modeling task, because it only uses form as training data, cannot in principle lead to learning of meaning”.

[Bender and Koller 2020]

Criticism against “form”-only learning

“We argue that the language modeling task, because it only uses form as training data, cannot in principle lead to learning of meaning”.

[Bender and Koller 2020]



Criticism against “form”-only learning

“We argue that the language modeling task, because it only uses form as training data, cannot in principle lead to learning of meaning”.

[Bender and Koller 2020]



The **meaning** of the words *blue* and *green* as it occurs in the world is more than just occurrences of word **form** in data



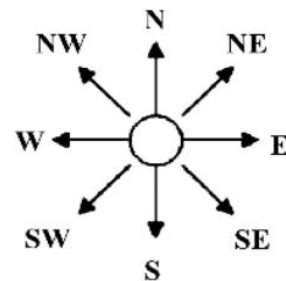
blue



green

Word “meaning”

Word “form”



north

west

Word “meaning”

Word “form”

Quinian bootstrapping & learning with placeholder symbols

(from Carey's "The Origin of Concepts")

Quinian bootstrapping & learning with placeholder symbols

(from Carey's "The Origin of Concepts")



- **Relations among some symbols are learned directly, in terms of each other** (“proton”, “mole”)
- Symbols are initially at most only partly interpreted in terms of antecedently available concepts
- Symbols serve as **placeholders**
- Modeling processes—**analogy**, inductive inference, thought experiment, limiting case analyses—are used to provide conceptual underpinnings for the placeholders
- These modeling processes combine and integrate separate representations from distinct domain-specific conceptual systems
- These processes **create explicit representations** of knowledge previously embodied in constraints on the computations

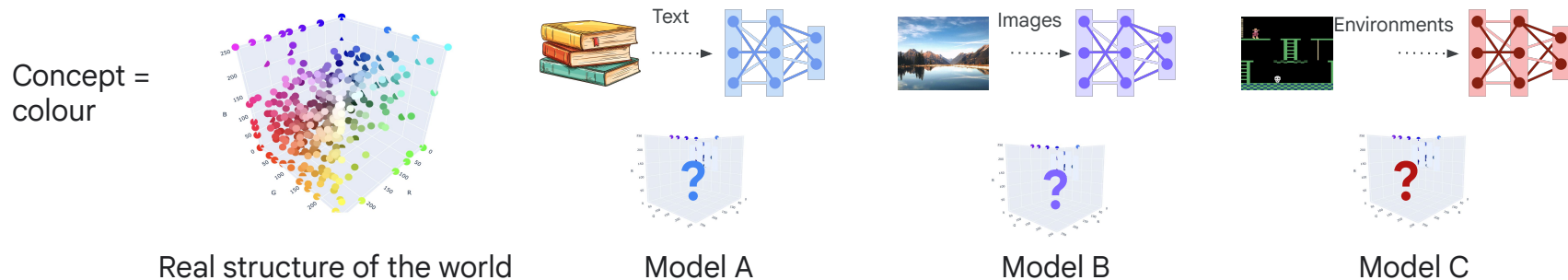
This talk: Understanding the conceptual structure of language models

- ✗ Taking a stance on whether models should be able to learn conceptual meaning
- ✓ Ways in which we can understand the “conceptual structure” learned by LLMS

This talk: Understanding the conceptual structure of language models

- ✗ Taking a stance on whether models should be able to learn conceptual meaning
- ✓ Ways in which we can understand the “conceptual structure” learned by LLMS

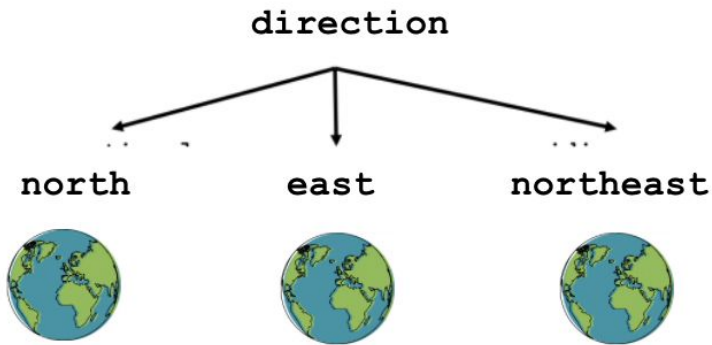
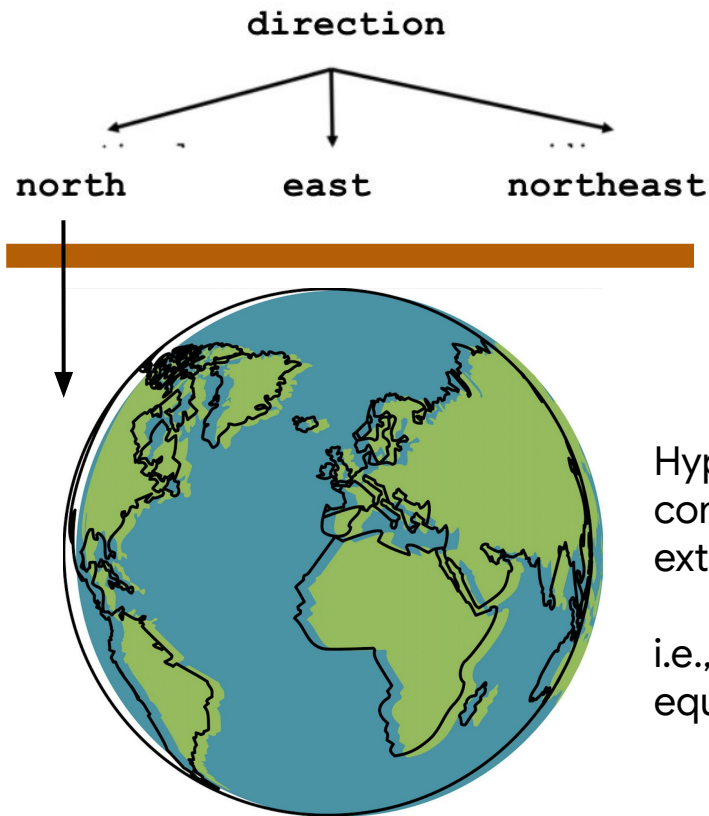
How? Evaluate and compare the representations of concepts learned by different models by analogy with grounded representations of concepts in the real world



02

Evaluating the conceptual structure of LLMs

Bootstrapping concept representations to the world

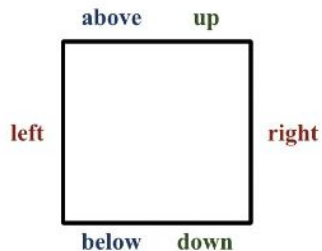


Hypothesis: If we teach models how to ground **some** concepts, can they use the structure of language to extrapolate meaning to **all** related concepts?

i.e., is the text-only representation of concepts functionally equivalent to what would be learned if it was grounded?

Categories of concepts

Spatial Terms



World:

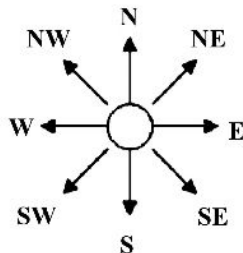
[0. 0. 0. 0. 0.]

[0. 0. 0. 1. 0.]

[0. 0. 0. 0. 0.]

Answer: right

Cardinal Directions



World:

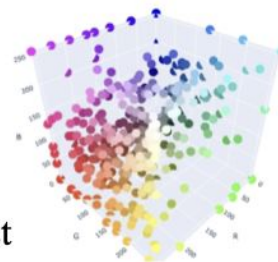
[0. 0. 1.]

[0. 0. 0.]

[0. 0. 0.]

Answer: northeast

RGB Colours



RGB: (255, 0 0)

Answer: red

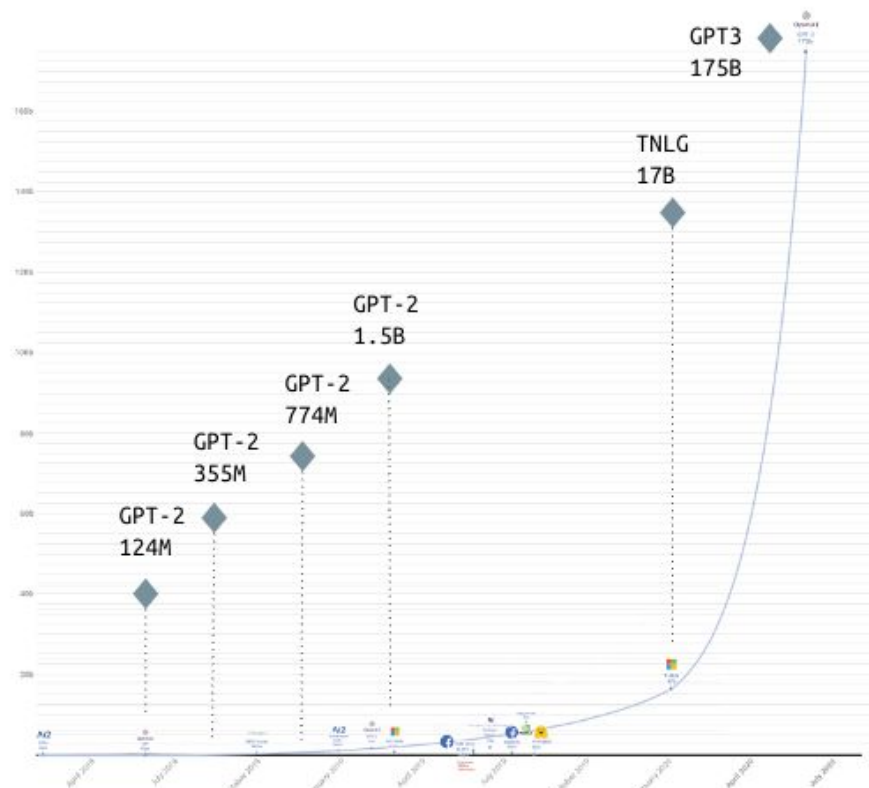
RGB: (143, 0, 255)

Answer: violet

Text-only LLMs tested

GPT-2	(124 M)
GPT-2	(355 M)
GPT-2	(774 M)
GPT-2	(1.5 B)
TNLG	(17 B)
GPT-3	(175 B)

All language models trained to
generate words sequentially



In-context inputs of grounded concept representations

Prompt style 1

```
Array
[[0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]]
Position: right

Array
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0.]]
Position: right

Array
[[0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]]
Position: left

Array
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 1.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
Position: right
```

Prompt style 2

```
Array
[[0. 0. 0. 0. 1. 0. 0.]]
Direction: right

Array
[[1. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]]
Direction: left

Array
[[0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0.]]
Direction: left

Array
[[0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
Direction: right
```

Prompt style 3

```
Array
[[0. 0.]
 [0. 1.]
 [0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]]
The direction is: right

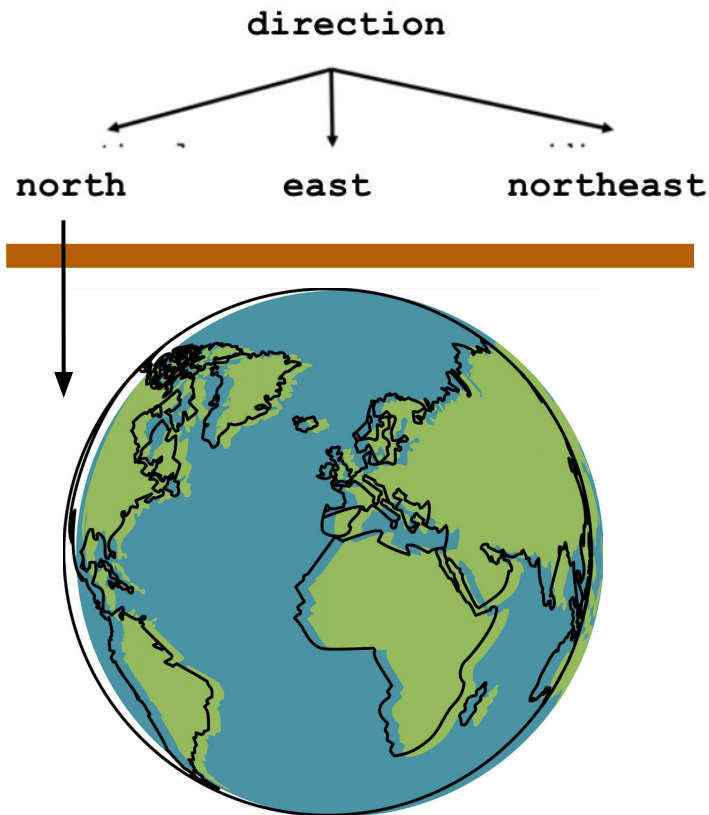
Array
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]]
The direction is: right

Array
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]]
The direction is: left
```

Example answers

```
left
right
.
.
.
```

What we want to evaluate



Prompt

```
Array  
[[0. 0.]  
 [0. 1.]  
 [0. 0.]  
 [0. 0.]  
 [0. 0.]  
 [0. 0.]  
 [0. 0.]]
```

The direction is: right

```
Array  
[[0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 1.]  
 [0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0. 0.]]
```

The direction is: right

```
Array  
[[0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [1. 0. 0. 0. 0.]]
```

The direction is: left

Example answers

left
right
.
.
.

For new realisations of the meaning of concepts, can LLMs learn in-context to extrapolate their text-only meaning to the grounded meaning in a new world?

How do we ensure models aren't memorising what we're testing?

Description [\[edit \]](#)

A matrix of the form

$$L = \begin{bmatrix} \ell_{1,1} & & & & 0 \\ \ell_{2,1} & \ell_{2,2} & & & \\ \ell_{3,1} & \ell_{3,2} & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n,1} & \ell_{n,2} & \dots & \ell_{n,n-1} & \ell_{n,n} \end{bmatrix}$$

is called a **lower triangular matrix** or **left triangular matrix**, and analogously a matrix of the form

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \dots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \dots & u_{2,n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & u_{n-1,n} \\ 0 & & & & u_{n,n} \end{bmatrix}$$

is called an **upper triangular matrix** or **right triangular matrix**. A lower or left triangular matrix is commonly denoted with the variable L , and an upper or right triangular matrix is commonly denoted with the variable U or R .



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Current events](#)
[Random article](#)
[About Wikipedia](#)
[Contact us](#)
[Donate](#)

[Contribute](#)
[Help](#)
[Learn to edit](#)
[Community portal](#)

Not logged in

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Matrix (mathematics)

From Wikipedia, the free encyclopedia

For other uses, see [Matrix](#).

"[Matrix theory](#)" redirects here. For the physics topic, see [Matrix theory \(physics\)](#).

In [mathematics](#), a **matrix** (plural **matrices**) is a [rectangular](#) array or table of [numbers](#), [symbols](#), or [expressions](#), arranged in rows and columns, which is used to represent a [mathematical object](#) or a property of such an object. For example,

$$\begin{bmatrix} 1 & 9 & -13 \\ 20 & 5 & -6 \end{bmatrix}$$

is a matrix with two rows and three columns; one say often a "two by three matrix", a "2×3-matrix", or a matrix of dimension 2×3.

How do we ensure models aren't memorising what we're testing?

	Name ↕	Hex (RGB) ↕	Red (RGB) ↕	Green (RGB) ↕	Blue (RGB) ↕	Hue (HSL/HSV) ↕	Satur. (HSL) ↕	Light (HSL) ↕	Satur. (HSV) ↕	Value (HSV) ↕	CGA number (name); alias ↕
	Aqua	#00FFFF	0%	100%	100%	180°	100%	50%	100%	100%	11 (high cyan); cyan
	Blue	#0000FF	0%	0%	100%	240°	100%	50%	100%	100%	09 (high blue)
	Teal	#008080	0%	50%	50%	180°	100%	25%	100%	50%	03 (low cyan)
	Navy	#000080	0%	0%	50%	240°	100%	25%	100%	50%	01 (low blue)
	Lime	#00FF00	0%	100%	0%	120°	100%	50%	100%	100%	10 (high green); green
	Green	#008000	0%	50%	0%	120°	100%	25%	100%	50%	02 (low green)
	Black	#000000	0%	0%	0%	0°	0%	0%	0%	0%	00 (black)
	Gray	#808080	50%	50%	50%	0°	0%	50%	0%	50%	08 (dark gray)
	Purple	#800080	50%	0%	50%	300°	100%	25%	100%	50%	05 (low magenta)
	Olive	#808000	50%	50%	0%	60°	100%	25%	100%	50%	06 (brown)
	Maroon	#800000	50%	0%	0%	0°	100%	25%	100%	50%	04 (low red)
	Silver	#C0C0C0	75%	75%	75%	0°	0%	75%	0%	75%	07 (light gray)
	White	#FFFFFF	100%	100%	100%	0°	0%	100%	0%	100%	15 (white)
	Fuchsia	#FF00FF	100%	0%	100%	300°	100%	50%	100%	100%	13 (high magenta); magenta

From all the (training) data on the web, there are many instances of the “groundings” we wish to evaluate (e.g., names of colours) that language models may have seen during training.

How do we ensure that we are truly testing grounding or bootstrapping and not memorisation?

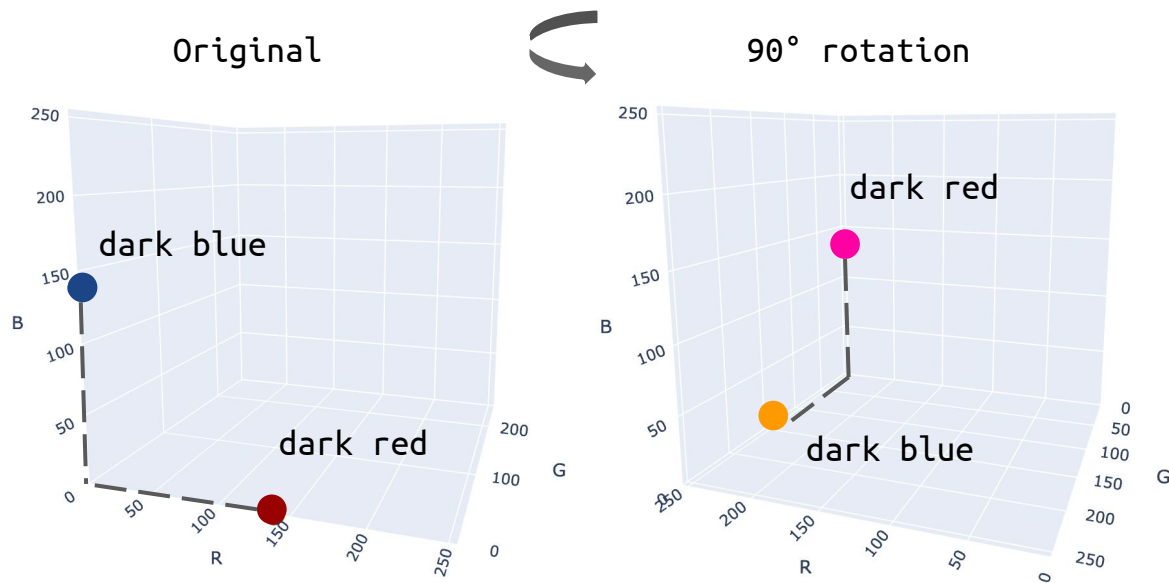
Isomorphism as a control

“..systems are capable of understanding if they emulate representations that are isomorphic to underlying meaning under important semantic relations like equivalence.”

$$f(x) = f(x') \iff g(x) = g(x').$$

Isomorphically equivalent worlds (albeit different groundings of concepts) should allow models to bootstrap meaning equally well.

Example rotations in colour space



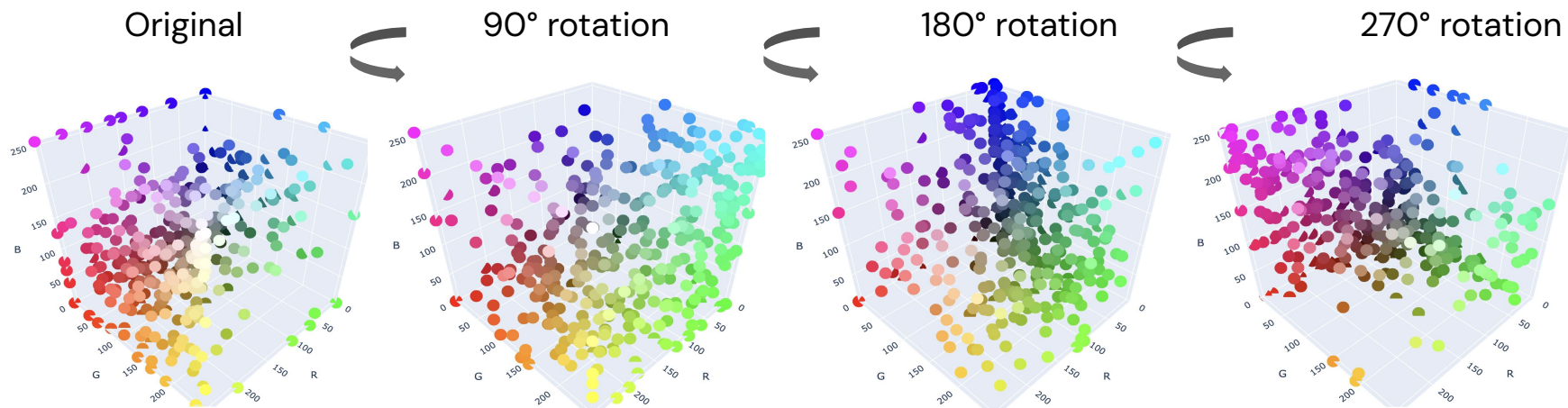
$$f(x') \iff g(x)$$

New “grounding” or semantic meaning of each element in a rotated world

Preserves the structural relation between elements

Example rotations in colour space

Rotating by n degrees: different word forms but same relation between word forms



Example rotations in cardinal space

Rotating by n degrees: different word forms but same relation between word forms

Original



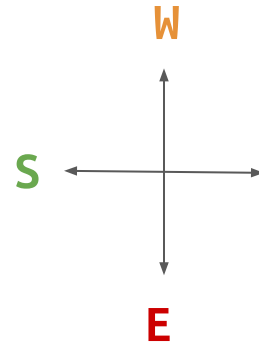
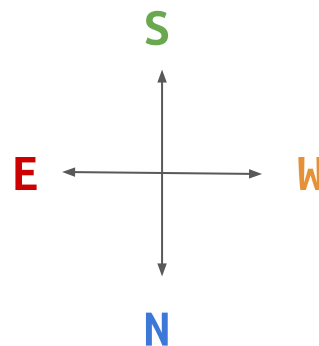
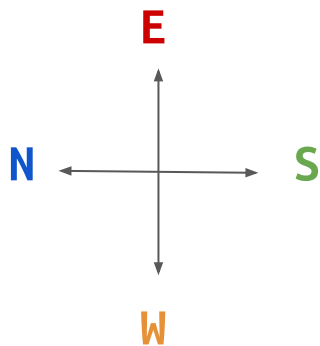
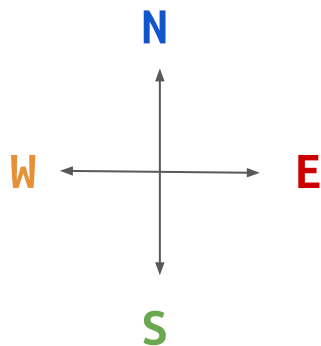
90° rotation



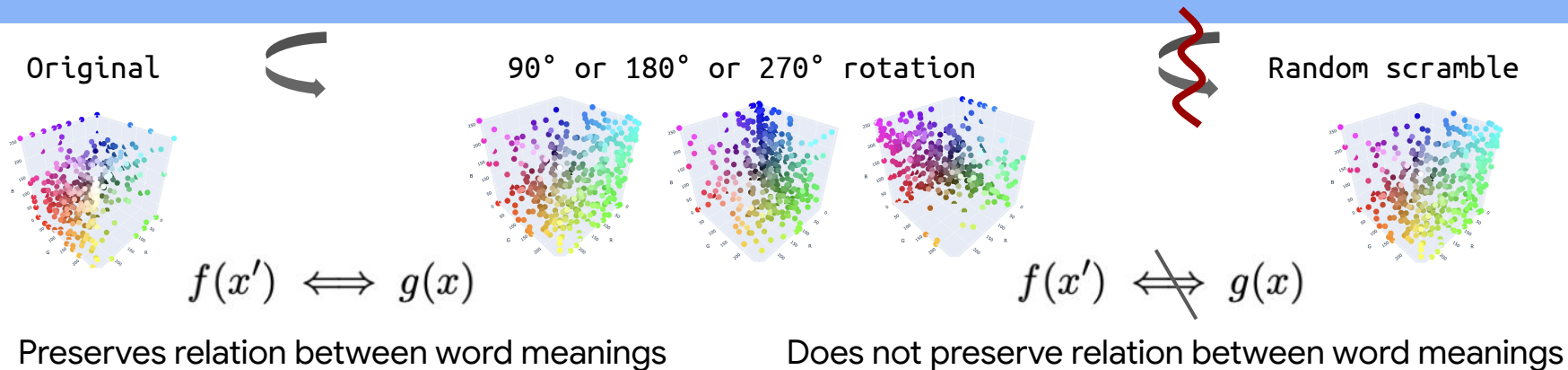
180° rotation



270° rotation



Isomorphic worlds: our expectation



If models know the relation between words..

And can learn a grounding in one world, they should learn a grounding in all worlds that preserve relations.

The generalisation pattern is somewhat consistent!

A grounding that does not preserve this should be impossible to learn.

The generalisation pattern is random and inconsistent!

Experimental Setup

*A sanity check
(few-shot generalisation)*

Can models learn grounded concepts when it is taught to them during (few-shot) training?

Zero-shot generalisation

If we teach a model some grounded concepts, can they generalise to other related concepts?

Error Analysis

How far are model predictions (in the grounded world) from the true grounding?

*Does this hold in the true
<-> rotated worlds? Is
performance lower in the
random worlds?*



Few shot generalisation

Example Input (20 in-context-learning examples followed by prompt)

World: [0. 0. 0.] [0. 0. 0.] [0. 0. 1.] Answer: right	World: [0. 0. 0.] [0. 0. 0.] [0. 0. 1.] Answer: right	World: [0. 0. 0.] [0. 0. 0.] [0. 0. 1.] [0. 0. 0.] [0. 0. 0.] Answer: right	World: [1. 0.] [0. 0.] [0. 0.] [0. 0.] [0. 0.] Answer: left
World: [1. 0. 0. 0.] Answer: left ...13 more...	World: [0. 0.] [1. 0.] [0. 0.] Answer: left	World: [0. 1. 0. 0.] Answer: left	World: [1. 0. 0. 0.] [0. 0. 0. 0.] Answer:

Example Model Outputs

GPT-2 (124M)

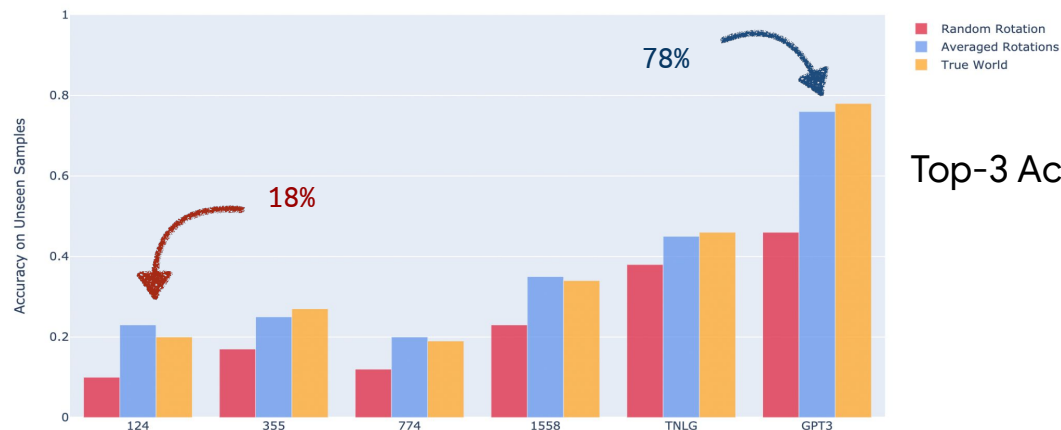
world	P=0.09
0. 0.]]	P=0.08
[0 [0	P=0.01

GPT-3 (175B)

left	P=0.20
right	P=0.11
leftmost	P=0.01

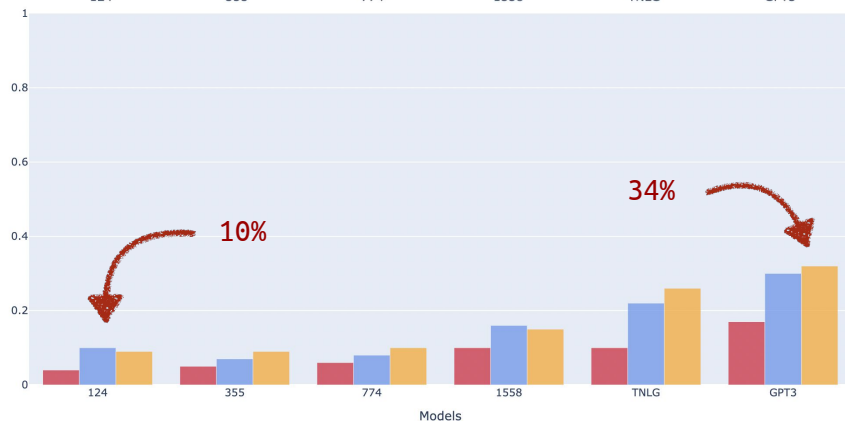
When models are shown a new representation of a concept can they extrapolate their learned meaning of the concept to this new representation in-context?

Few shot generalisation



Top-3 Accuracy

In new unseen worlds, models **can** learn to ground what is up vs. down when they have been taught (few-shot) both concepts.



Top-1 Accuracy

Array

```
[[0. 0. 0. 1.]  
 [0. 0. 0. 0.]  
 [0. 0. 0. 0.]  
 [0. 0. 0. 0.]  
 [0. 0. 0. 0.]  
 [0. 0. 0. 0.]  
 [0. 0. 0. 0.]]
```

The direction is: up

Zero shot generalisation

Example Input (60 in-context-learning examples followed by prompt)

Example Model Outputs

 RGB: (48, 213, 200) Answer: orange	 RGB: (220, 20, 60) Answer: crimson	 RGB: (0, 0, 128) Answer:
---	---	---

GPT-2 (124M)

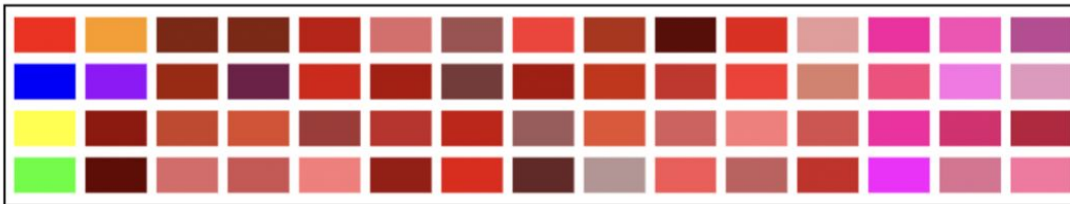
color	P=0.14
black	P=0.08
rgb	P=0.02

GPT-3 (175B)

navy	P=0.24
dark blue	P=0.13
blue	P=0.08

Sub-space
centered
around the
colour red

All 60 Training Examples

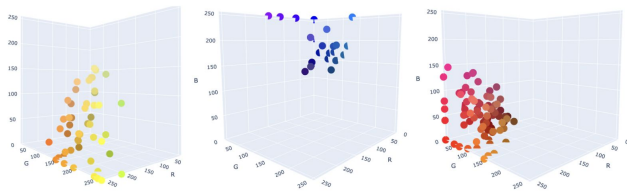


6 Primary and Secondary Colours

red, blue, yellow, green, orange, violet

57 Colours Within a Sub-space

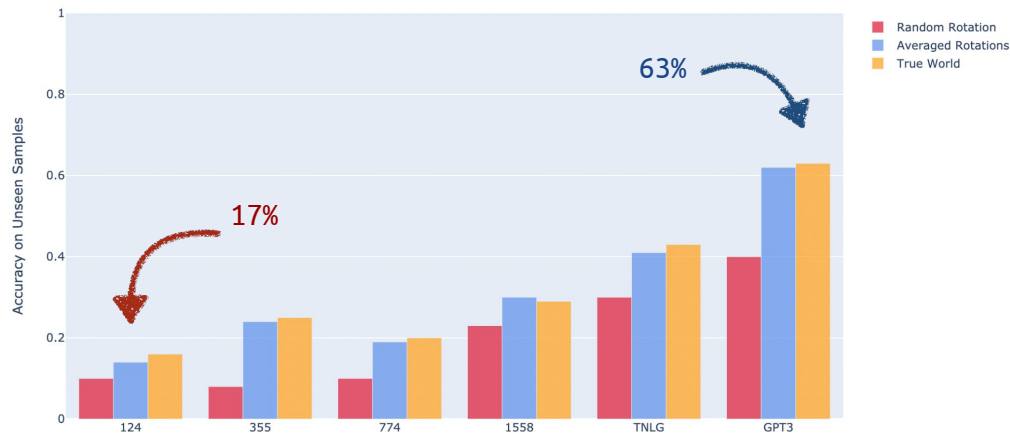
dark red, maroon, crimson, fuchsia, rust, bright red..



Example colour sub-spaces for yellow, blue and red

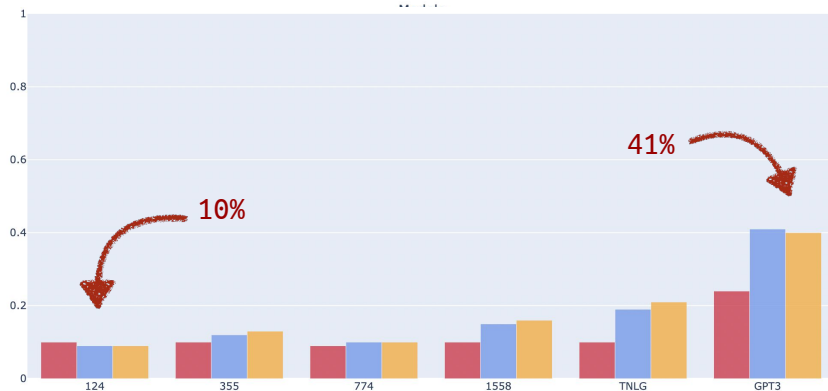
When models are shown new representations of **some** concepts can they bootstrap to new unseen concepts in-context?

Zero shot generalisation



Top-3 Accuracy

In new unseen worlds, models have to ground what *unseen colours* when they have been taught only *10 colours* (zero-shot).



Top-1 Accuracy

RGB: (0, 65, 106)
Answer: dark imperial blue

RGB: (111, 0, 255)
Answer: electric indigo

RGB: (239, 222, 205)
Answer:

Analysing model predictions

Model Prediction

dark red
medium apricot
burnt orange
green
aquamarine
dark rose

True Grounding

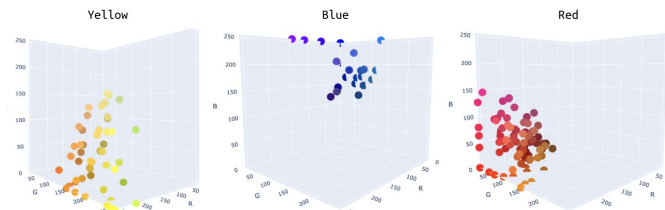
wine
apricot
orange
forest green
sea green
deep carmine

True G	Predicted G & Distance
dark red	wine (76.5), light crimson (208.1)
light green	dark slate gray (144.7) beige (126.6), light sea green (129.7) cerulean (185.7), violet (262.6)

Table 2. Example predictions from GPT-3 with Euclidean distances from the ground-truth.

$$d(c_1, c_2) = \begin{cases} \sqrt{(c_{1x} - c_{2x})^2 + (c_{1y} - c_{2y})^2 + (c_{1z} - c_{2z})^2} & \text{for } c_1 \in C \\ 500 & \text{for } c_1 \notin C \end{cases}$$

Eq 1. Euclidean distance between two points in space



Analysing model predictions

	124M	355M	774M	1.5B	175B
C	328.3	309.5	209.6	190.7	96.3
R-IV	334.9	334.9	334.9	334.9	334.9
R-ID	174.9	174.9	174.9	174.9	174.9

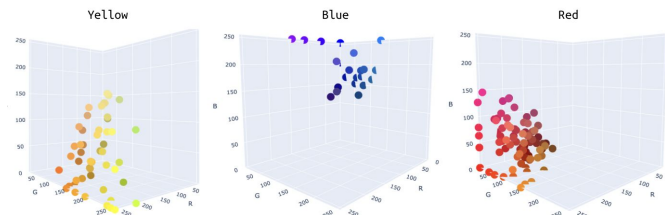
Table 1. Average Euclidean distance between predictions and ground-truth answers.

True G	Predicted G & Distance
dark red	wine (76.5), light crimson (208.1) dark slate gray (144.7)
light green	beige (126.6), light sea green (129.7) cerulean (185.7), violet (262.6)

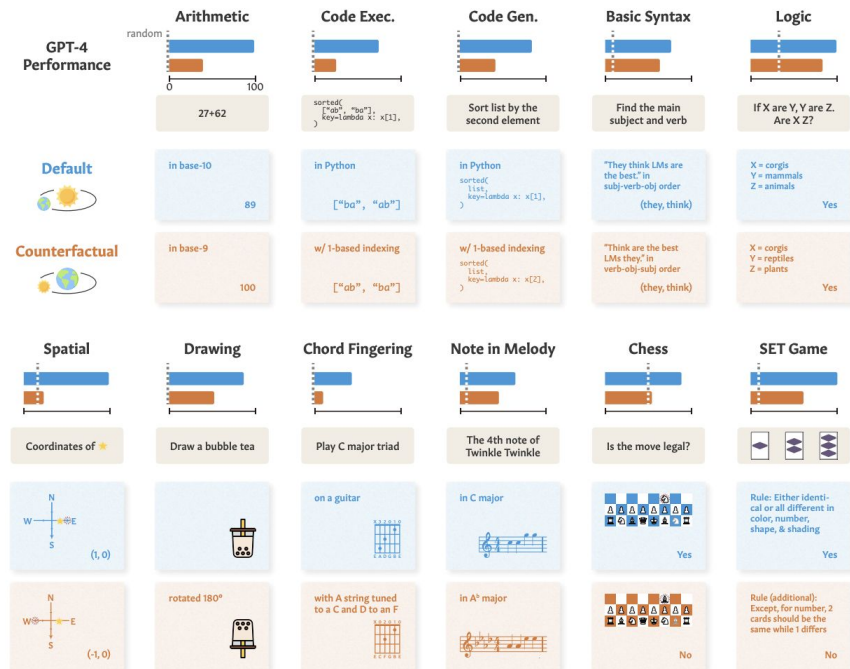
Table 2. Example predictions from GPT-3 with Euclidean distances from the ground-truth.

$$d(c_1, c_2) = \begin{cases} \sqrt{(c_{1x} - c_{2x})^2 + (c_{1y} - c_{2y})^2 + (c_{1z} - c_{2z})^2} & \text{for } c_1 \in C \\ 500 & \text{for } c_1 \notin C \end{cases}$$

Eq 1. Euclidean distance between two points in space

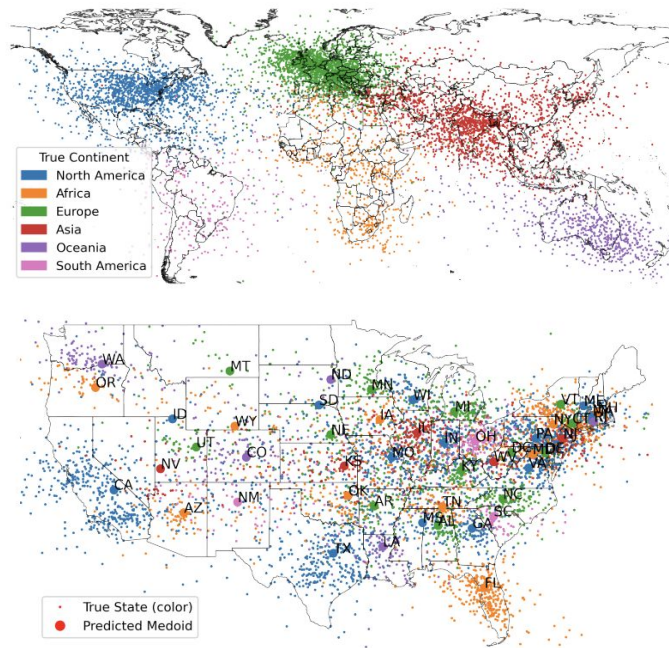


Other concepts and counterfactuals



Evaluating adaptability of LLM reasoning capabilities to new domains (with counterfactual controls)

[Wu et al., 2024]



Evaluating linear representations of concepts (space, time) and if circuits/neurons responsible for concepts can be uncovered

[Gurnee and Tegmark., 2023]

02

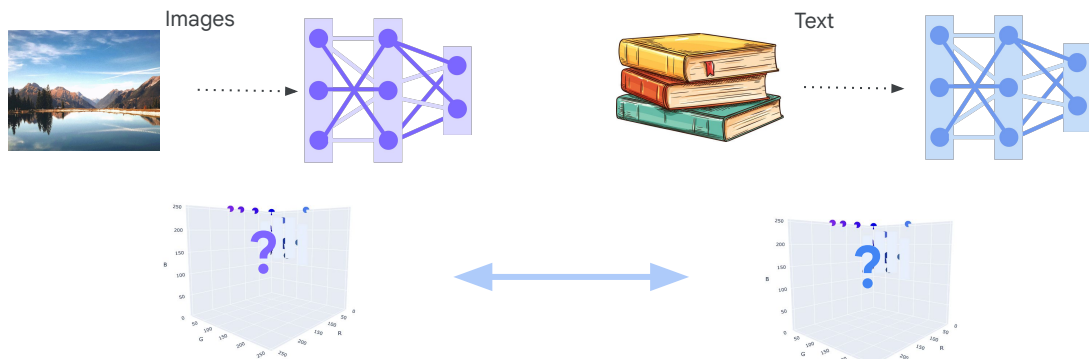
Mapping representations between models

Conceptual spaces of different pre-trained models

The models we studied before were text-only language models.

Hypothesis: once the inputs are encoded and transformed into “meaning representations”, if the conceptual spaces of models are similar, we should be able to map between them.

Conceptual spaces of different pre-trained models



Model can take in images and text
and produce some output class

Model A

Vision-Text LLM (CLIP)

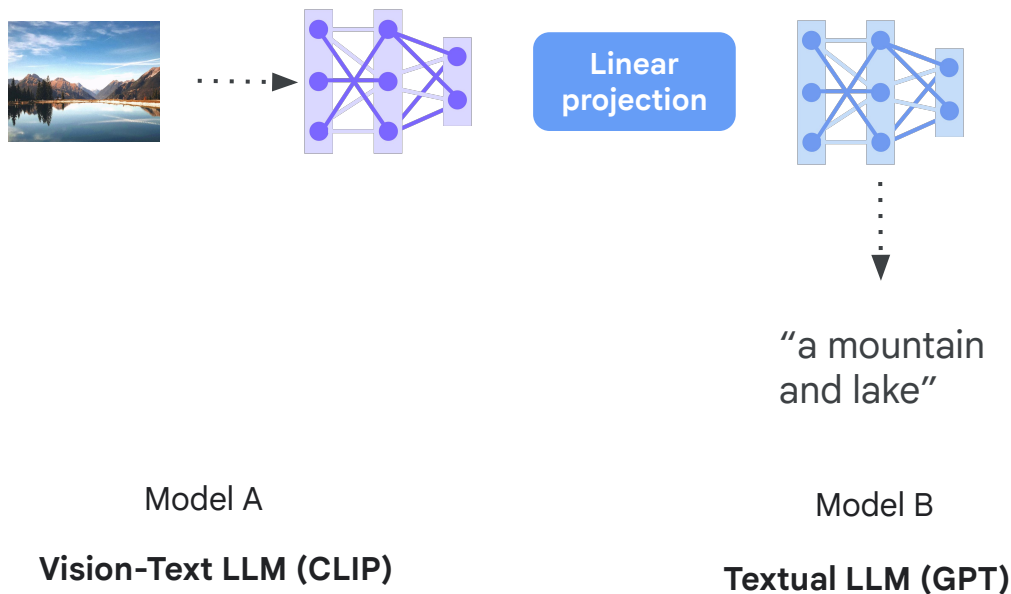
Model can take in text and
produce some output text

Model B

Textual LLM (GPT)

Model A and B might be trained
on different datasets,
modalities, have different
architectures and input and
output spaces.

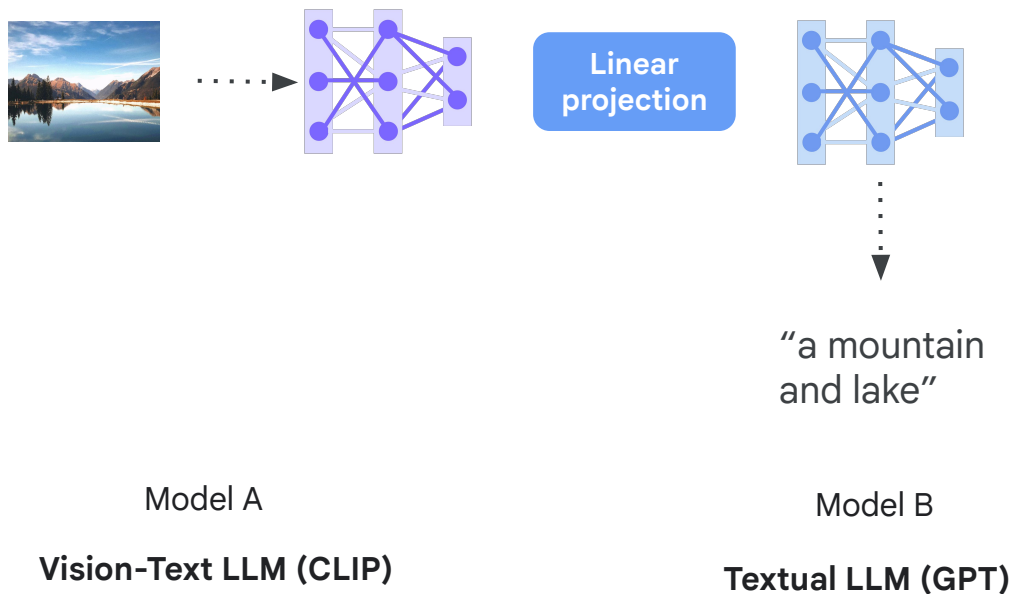
Conceptual spaces of different pre-trained models



Model A and B might be trained on different datasets, modalities, have different architectures and input and output spaces.

However, can we map representations from one model to another and evaluate if outputs (and therefore conceptual spaces being learned) are consistent?

Can we use this to map between model representations?

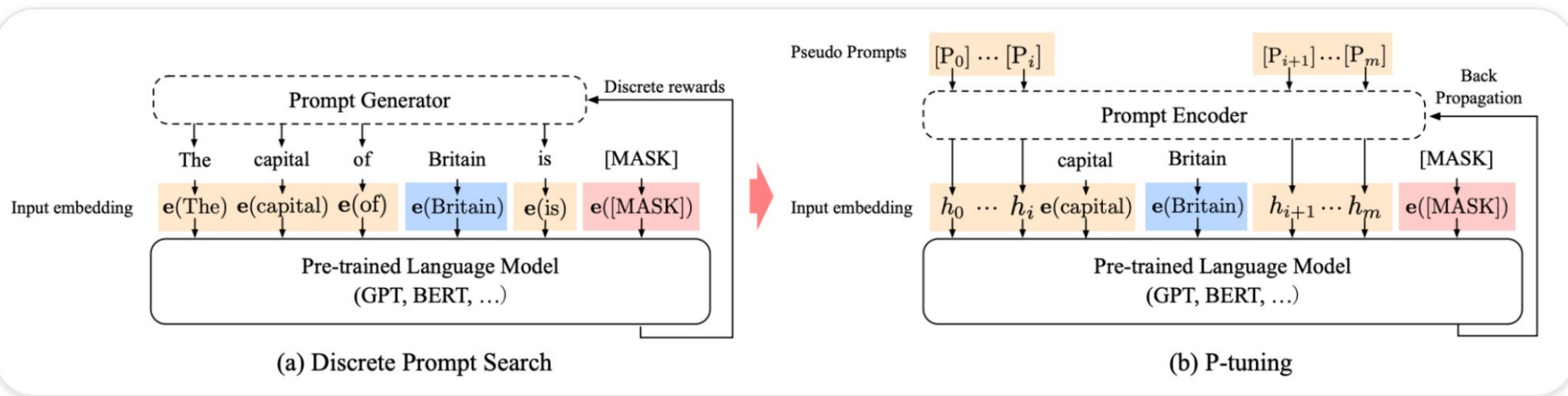


Learn a linear projection from Model A representation $\rightarrow$ Model B input (e.g., image captioning)

Since a linear projection is distance-preserving, if the conceptual spaces of the two models are structured similarly, at test-time, the LLM should be able to take in any image representation and predict a caption that is correct

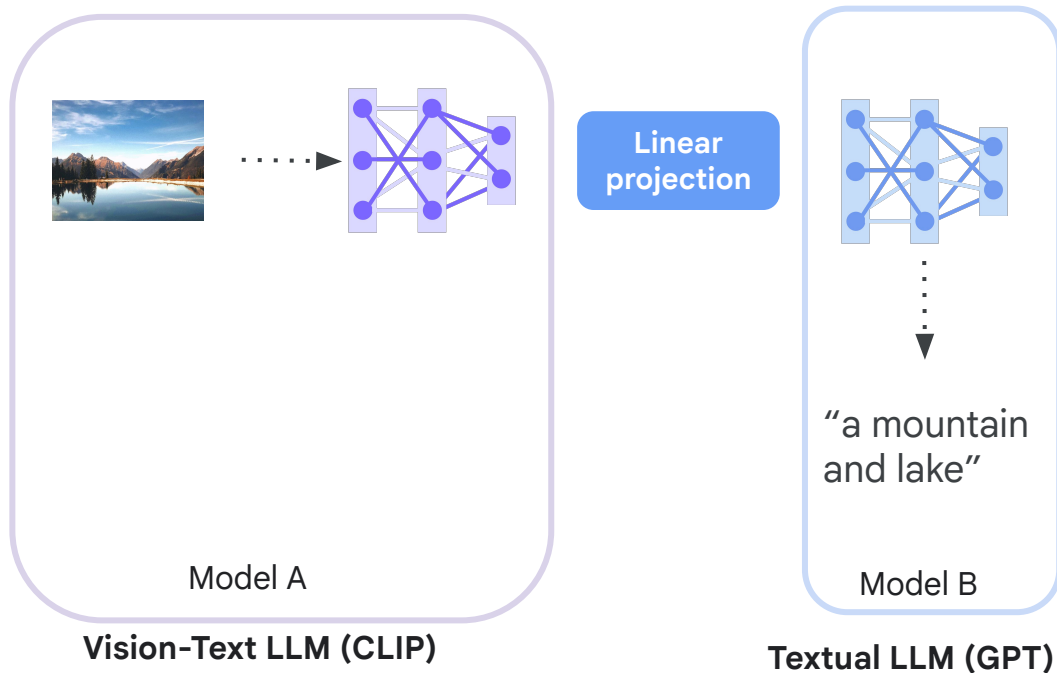
Evaluate image-projected-LLM outputs for concepts.

Soft prompting to elicit in-context learned information



“Soft prompts” or prefixes can be learned through a linear projection into the embedding space of a model for a specific task. This learned projection can then improve performance on that task!

Evaluating differently trained image encoders



Three different image encoders that received different amounts of linguistic supervision during training

NFRN50: pre-trained on image classification categories

BEIT: no exposure to language/classes

CLIP: pre-trained to align images with full natural language captions

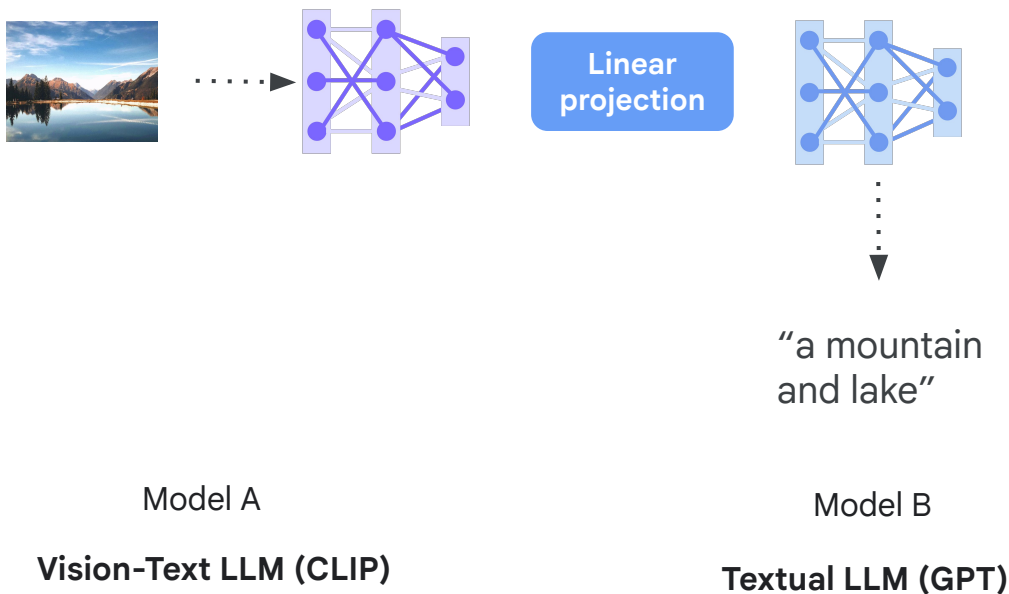
Control: randomly initialised NFRN50 that was not pretrained

Text decoder: GPT-J (6B parameters)

Evaluating differently trained image encoders

Image Captioning	NoCaps - CIDEr-D				NoCaps (All)		CoCo	CoCo	
	In	Out	Near	All	CLIP-S	Ref-S	CIDEr-D	CLIP-S	Ref-S
🔥NFRN50 Tuned	20.9	30.8	25.3	27.3	66.5	72.5	35.3	69.7	74.8
🔥MAGMA (released)	18.0	12.7	18.4	16.9	63.2	68.8	52.1	76.7	79.4
🔥MAGMA (ours)	30.4	43.4	36.7	38.7	74.3	78.7	47.5	75.3	79.6
❄️NFRN50 Random	5.4	4.0	4.9	5.0	47.5	55.7	4.8	49.5	57.1
❄️BEIT	20.3	16.3	18.9	18.9	62.0	69.1	22.3	63.6	70.0
❄️NFRN50	21.3	31.2	26.9	28.5	65.6	71.8	36.2	68.9	74.1
❄️CLIP	34.3	48.4	41.6	43.9	74.7	79.4	54.9	76.2	80.4

Evaluating differently trained image encoders



Finding #1

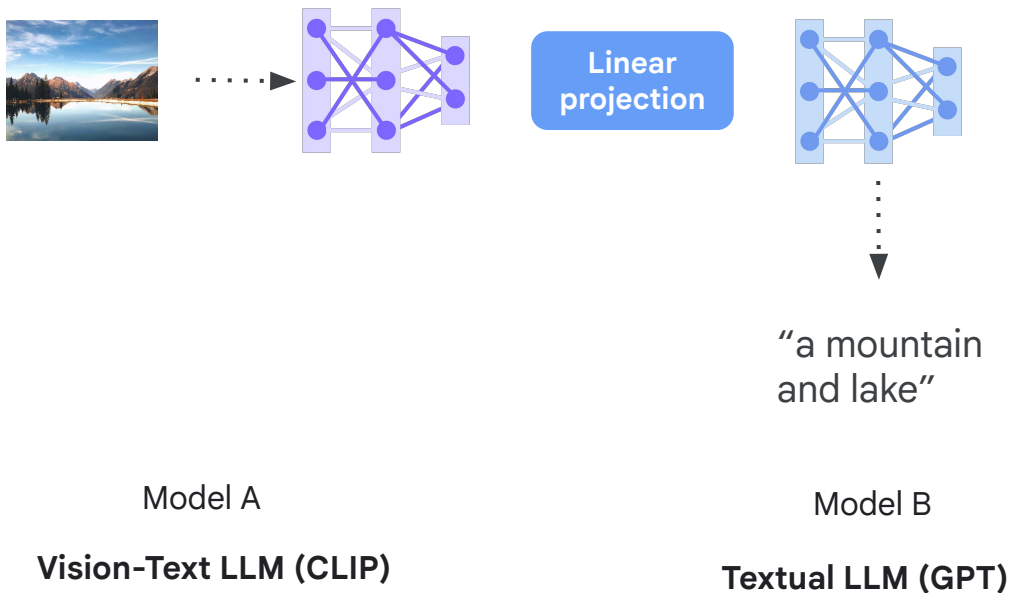
Linguistic supervision during pre-training plays a key role in concept formation

CLIP and NFRN50 outperform BEIT in terms of transferring image properties to language

BEIT only transfers coarse-grained visual properties (scene/background)

CLIP can reproduce concepts in the image to a greater accuracy

Evaluating differently trained image encoders



Finding #2

For some models, projecting between representations has comparable accuracy to a trained model.

A separately trained image model projected onto a text model can produce conceptually correct captions from the text-only model similar to what a trained vision-text model would!

Example captions from projecting from Model A to Model B

Image Captioning



CLIP	a giraffe in the lobby of the building
NFRN50	the giraffe in the zoo.
BEIT	a peacock in the garden
NFRN50	a man and a woman in a field of flowers
Random	



CLIP	tennis player in action
NFRN50	tennis player at the tennis tournament.
BEIT	tennis player during a tennis match.
NFRN50	the new logo for the team
Random	

Visual Question Answering



CLIP	He is surfing a wave.
NFRN50	He is surfing the waves.
BEIT	He is jumping into the water.
NFRN50	He is swimming in the pool.
Random	

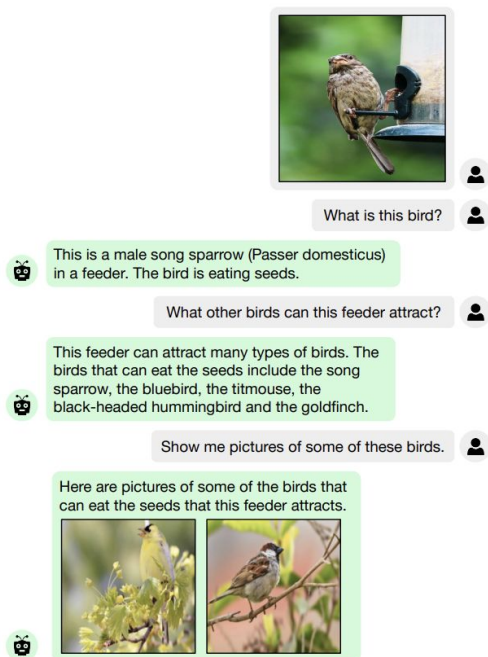
Q: What is the person doing?
A: surfing



CLIP	A tennis racket
NFRN50	A tennis racket
BEIT	A baseball bat.
NFRN50	A tree
Random	

Q: What is the person holding?
A: tennis racket

Mapping model spaces to each other



What is this bird?

This is a male song sparrow (*Passer domesticus*) in a feeder. The bird is eating seeds.

What other birds can this feeder attract?

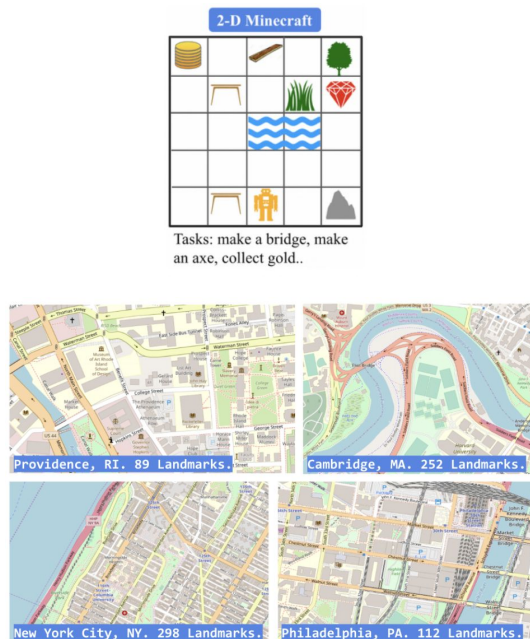
This feeder can attract many types of birds. The birds that can eat the seeds include the song sparrow, the bluebird, the titmouse, the black-headed hummingbird and the goldfinch.

Show me pictures of some of these birds.

Here are pictures of some of the birds that can eat the seeds that this feeder attracts.

Cross-modality interactions by fine-tuning input and output layers across models

[Koh et. al., 2023]



2-D Minecraft

Tasks: make a bridge, make an axe, collect gold..

Providence, RI. 89 Landmarks.

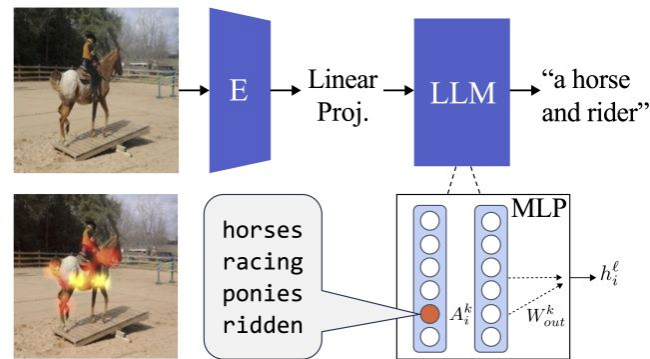
Cambridge, MA. 252 Landmarks.

New York City, NY. 298 Landmarks.

Philadelphia, PA. 112 Landmarks.

Map between representations of reinforcement learning agents and LLMs.

[Patel et. al., 2024]



Find neurons or circuits in the model that are responsible for decoding concepts when injected into models

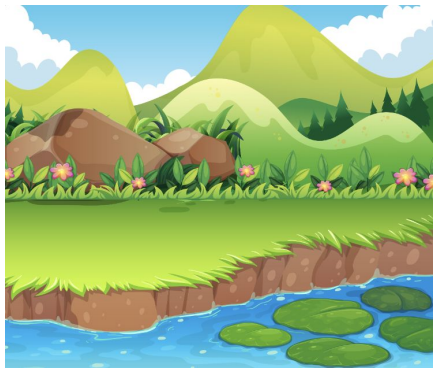
[Schettmann et. al., 2023]

What we've learned about representations learned by models

- 01 Models that have trained on text-only data learn a conceptual space that allows them to ground concepts in textual worlds
- 02 Models that have trained on text-only data can be mapped to models that train on image data to transfer conceptual information from one to another

03 Concept leakage and failures in mapping concepts in LLMs

How do word roles map to concepts for VLMs?



Human processing of language typically applies a single role to a word in context in one semantic interpretation of a sentence.

Finding: models often do not have the same distinction between sense and reference. For the same concept there can be semantic leakage across different senses of the noun for the same reference.

How do word roles map to concepts for VLMs?



Figure 1: The word *bat* is realized as two entities given the prompt *a bat is flying over a baseball stadium*.

80.3% over 216 images (17 stimuli)

1. A word or phrase is interpreted as two distinct (concurrently-incompatible entities)
 - a. “a bat is flying over a stadium”

How do word roles map to concepts for VLMs?



Main (left): *a **fish** and a gold ingot*.

Control (right): *a fish and an ingot*.

97.2% over 6 stimuli-control pairs

1. A word or phrase is interpreted as two distinct (concurrently-incompatible entities)
 - a. “a bat is flying over a stadium”
2. A word is interpreted as a modifier of two different entities
 - a. “a fish and a gold ingot”

How do word roles map to concepts for VLMs?

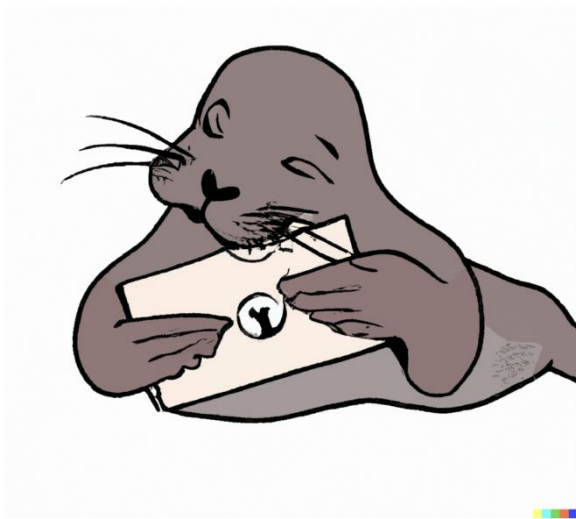


Figure 2: The word *seal* is realized both as an object and as a modifier of another word (*letter*) in the prompt *a seal is opening a letter*, resulting in *concept leakage*. The prompt tends to generate *sealed* letters, while minimally-different prompts that do not contain the noun *seal* do not.

1. A word or phrase is interpreted as two distinct (concurrently-incompatible entities)
 - a. “a bat is flying over a stadium”
2. A word is interpreted as a modifier of two different entities
 - a. “a fish and a gold ingot”
3. A word is interpreted simultaneously as an entity and as a modifier of a different entity
 - a. “a seal is opening a letter”

92.5% over 10 stimuli-control pairs

How do word roles map to concepts for VLMs?

- Caveats: small-scale experiments, need a more comprehensive evaluation
- VLMs don't follow the **basic principles of symbol-to-entity mapping**
- A single referent in a sentence is not mapped to one grounding in the world but **concept leakage** occurs
- Leakage reflects some differences in the conceptual spaces that the models form vs. what humans form

What we've learned about representations learned by models

- 01 Models that have trained on text-only data learn a conceptual space that allows them to ground concepts in textual worlds
- 02 Models that have trained on text-only data can be mapped to models that train on image data to transfer conceptual information from one to another
- 03 Models seem to have concept leakage and incorrect symbol-entity mapping. The conceptual space learned seems different from humans and true grounding

04 The way forward

04 The way forward

- 01 Linguistic and cognitive behavioural tests can help uncover similarities and differences in concepts learned by models and humans
- 02 Representational tests on a single model can allow us to uncover the conceptual spaces being formed during training
- 03 Projecting representations between multiple different models can allow us to discern whether models are forming similar conceptual spaces
- 04 Tools from mechanistic interpretability in combination with the above can allow us to uncover what parts of the model are responsible for concept usage

Thank you

Questions?

Google DeepMind

Extra slides

Towards understanding what concepts LLMs have learnt

Few shot generalisation



Evaluating language generations

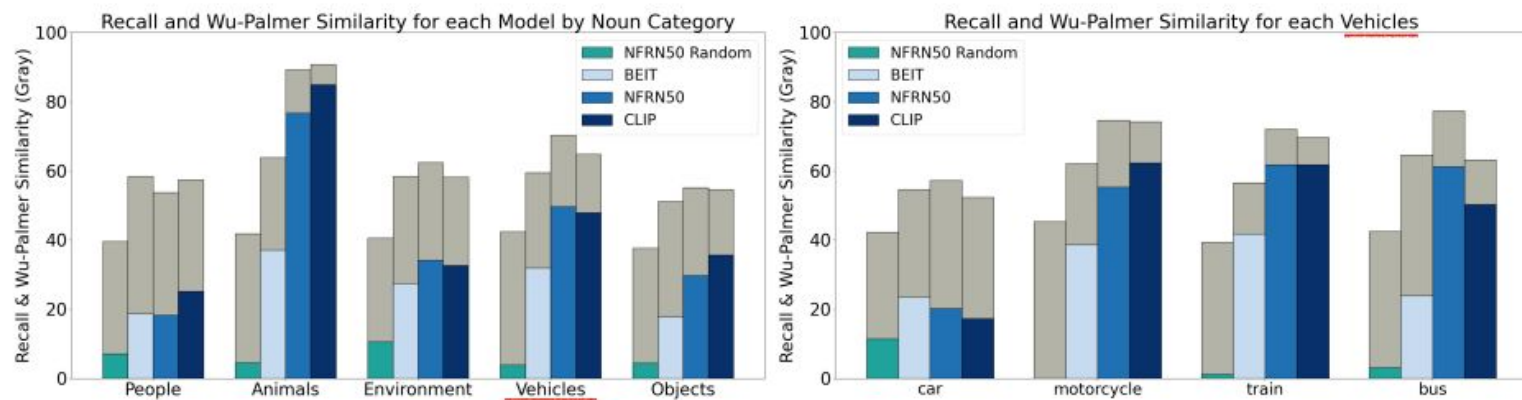
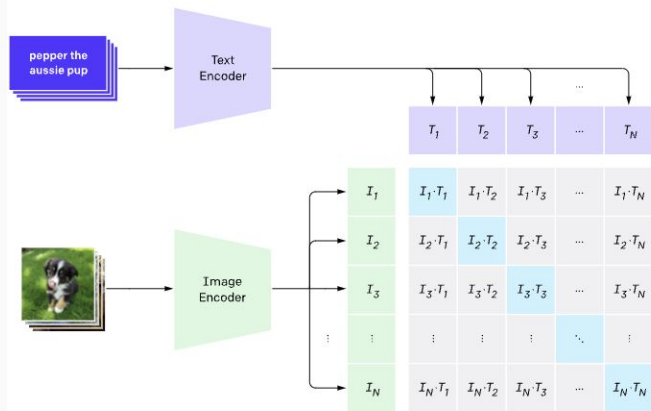


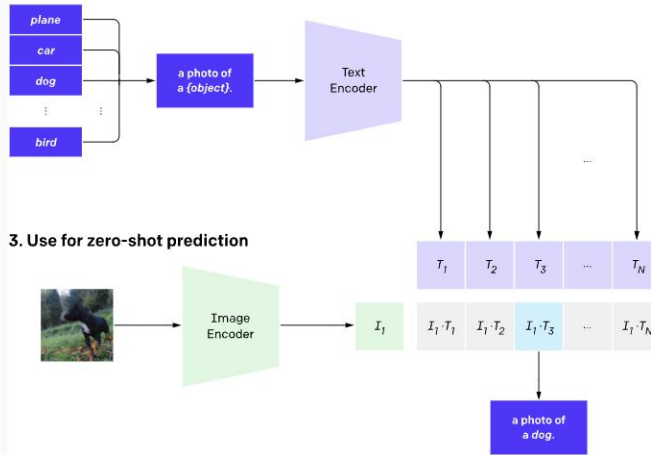
Figure 3: On average, recall of nouns in generated captions follows the standard pattern (CLIP, NFRN50, BEIT). However, judging by Wu-Palmer similarity, BEIT performs nearly the same or better than NFRN50 and CLIP on 4/5 of the noun categories. This indicates that although BEIT struggles to transfer the exact correct concept, it is transferring a related one based on visual similarity. On the right we show this effect for individual vehicle words. BEIT may have never learned to distinguish the ‘bus’ concept, but the LM still understands to generate a highly related concept, i.e., another vehicle. Average random Wu-Palmer similarity is around .4 consistently.

1. Contrastive pre-training

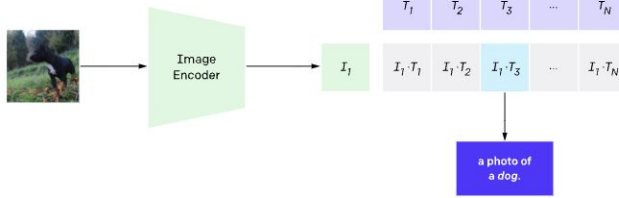


Overview A

2. Create dataset classifier from label text



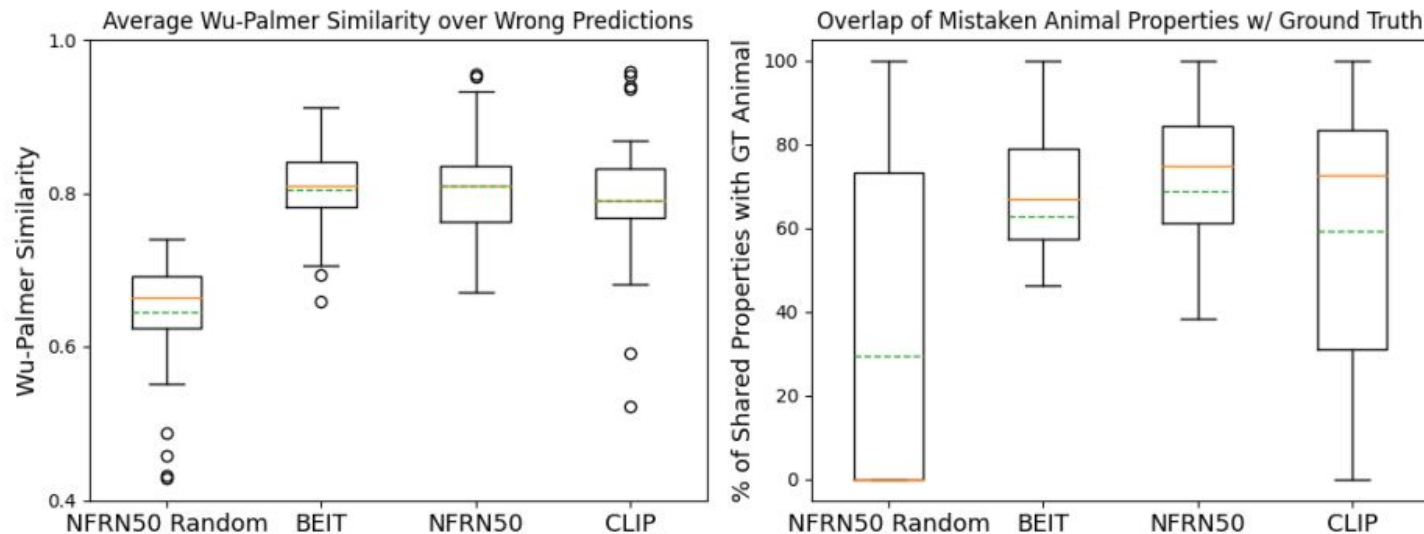
3. Use for zero-shot prediction



Overview B

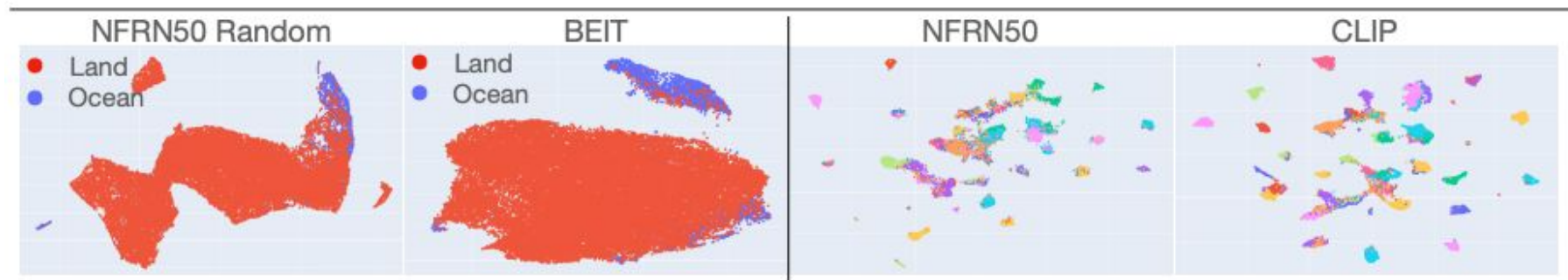
CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a dog" and predict the class of the caption CLIP estimates best pairs with a given image.

Evaluating language generations



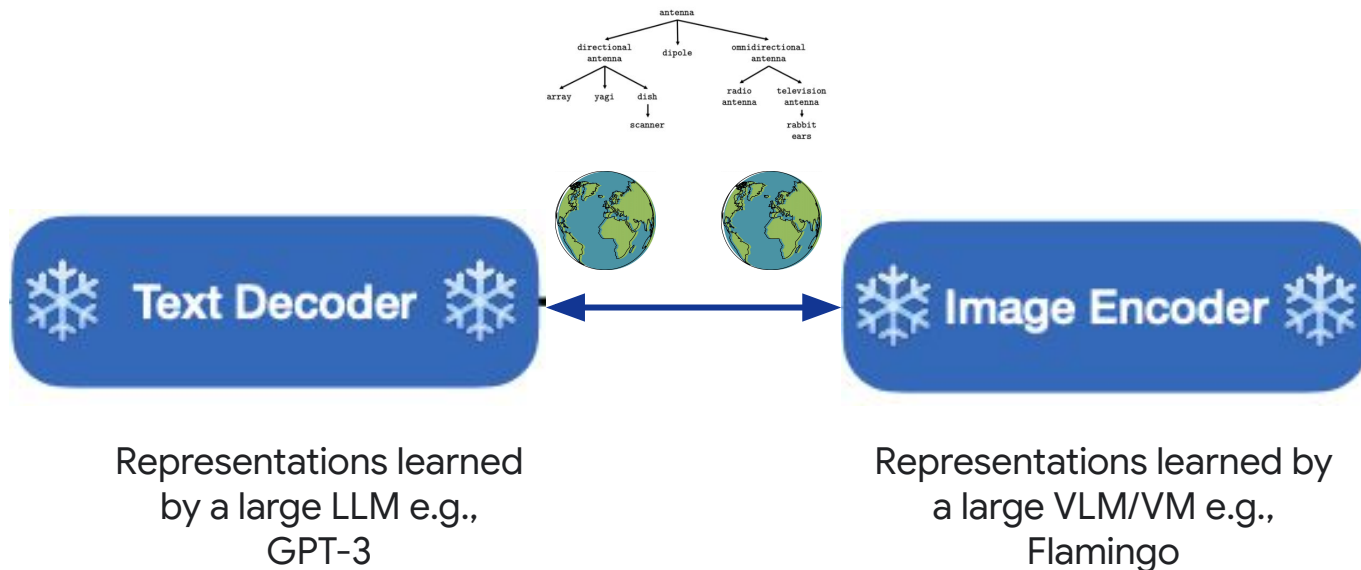
(a) **Left:** Wu-Palmer Similarity for captions in which the models made mistakes show that BEIT, NFRN50, and CLIP all similarly far off from the correct animal, meaning that even if they predict the wrong animal, it is on average very taxonomically similar. **Right:** When the model mistakes one animal for another in the dataset, how similar are the AWA properties for the true animal and the one it mistakes it most for? The average number of overlapping properties show that animals predicted from BEIT are at least as similar to the real animal as NFRN50 and CLIP. Median is shown as the solid orange line while the dashed green line shows the mean.

Evaluating language generations



(b) **UMAP projections of AWA images:** While NFRN50 and CLIP cluster tightly along lexical categories (color coded by animal), BEIT clusters the most distinctly along animals that live in water/the ocean; the randomly initialized NFRN50 mostly randomly overlap in one cluster.

Conceptual spaces of different pre-trained models



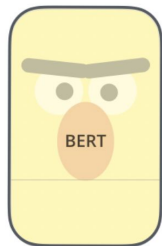
Can we map language model spaces to visual groundings?

Conceptual spaces of different pre-trained models



Figure 1: We train linear projections from image representations into the input space of a language model to produce captions describing images. We find that LMs can describe the contents of most image representations, but performance varies based on the type of image encoder used.

Human-like performance on benchmark language tasks



Can reconstruct WordNet structure and learn all word relations!

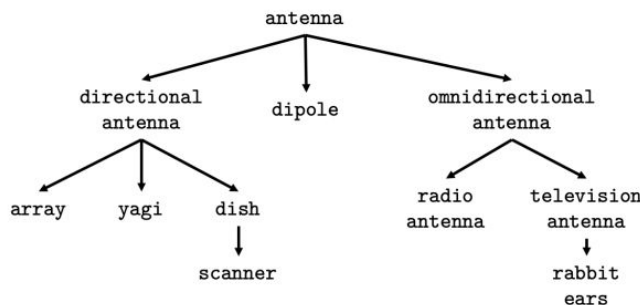
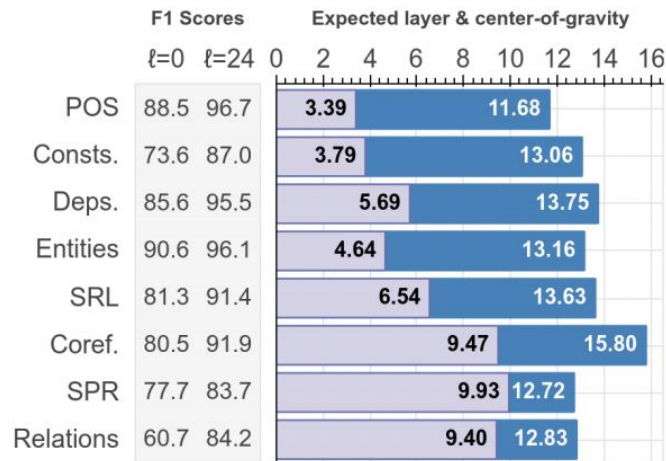


Figure 1: An example subtree from the WORDNET hierarchy.

Constructing Taxonomies from Pretrained Language Models. Chen, Lin and Klein. NAACL 2021.

Can solve most classical NLP tasks!



BERT Rediscovered the Classical NLP Pipeline. Tenney, Das and Pavlick. ACL 2019.

Quinian bootstrapping & learning with placeholder symbols

(from Carey's "The Origin of Concepts")

- **Relations among some symbols are learned directly, in terms of each other** ("proton", "mole")
- Symbols are initially at most only partly interpreted in terms of antecedently available concepts
- Symbols serve as **placeholders**
- Modeling processes—**analogy**, inductive inference, thought experiment, limiting case analyses—are used to provide conceptual underpinnings for the placeholders
- These modeling processes combine and integrate separate representations from distinct domain-specific conceptual systems
- These processes **create explicit representations** of knowledge previously embodied in constraints on the computations

Four column numbered points

01

Molestie litora condimentum
accumsan placerat fermentum
adipiscing parturient ipsum
condimentum neque urna
aptent ornare erat felis
vestibulum pharetra vestibulum
non blandit aenean justo
faucibus viverra.

02

Vestibulum a a dui mi amet a
consectetur nascetur
consectetur scelerisque nibh
eleifend mi pharetra. Vel
ullamcorper mus condimentum
tristique consectetur tempus
elementum ullamcorper a eget
ad sit hendrerit ullamcorper
per convallis vestibulum.

03

Tempor at tristique neque
malesuada tempus adipiscing
quam et mus odio facilisi
blandit magnis penatibus ac
litora vel curae rutrum donec
non fusce est a vestibulum.

04

Gravida etiam aliquet volutpat
fusce parturient nostra dictum
elementum a suspendisse
venenatis odio a condimentum
adipiscing parturient
condimentum rhoncus nam
sem a congue.

Four column text

Subheading

Molestie litora condimentum
accumsan placerat fermentum
adipiscing parturient ipsum
condimentum neque urna
aptent ornare erat felis
vestibulum pharetra vestibulum
non blandit aenean justo
faucibus viverra.

Subheading

Molestie litora condimentum
accumsan placerat fermentum
adipiscing parturient ipsum
condimentum neque urna
aptent ornare erat felis
vestibulum pharetra vestibulum
non blandit aenean justo
faucibus viverra.

Subheading

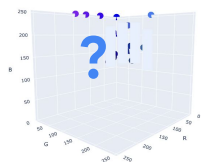
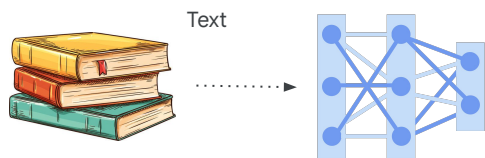
Molestie litora condimentum
accumsan placerat fermentum
adipiscing parturient ipsum
condimentum neque urna
aptent ornare erat felis
vestibulum pharetra vestibulum
non blandit aenean justo
faucibus viverra.

Subheading

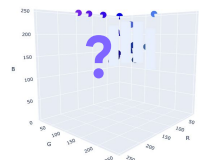
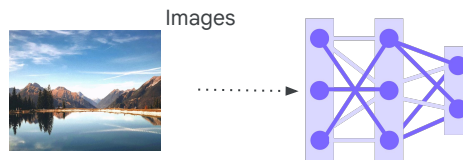
Molestie litora condimentum
accumsan placerat fermentum
adipiscing parturient ipsum
condimentum neque urna
aptent ornare erat felis
vestibulum pharetra vestibulum
non blandit aenean justo
faucibus viverra.

Statement accumsan diam eros et eu
lectus a fames parturient ultrices
platea.

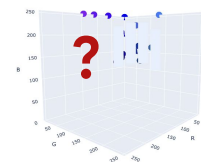
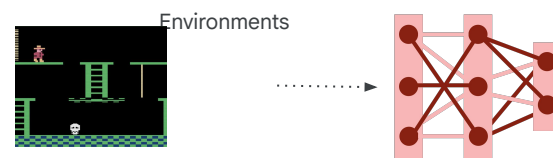
Conceptual spaces of different pre-trained models



Model A



Model B



Model C

Can we map language model spaces to visual groundings?

