A Deep Dive into CLIP Embeddings

Damian Stewart

d@damianstewart.com
github.com/damian0815
linkedin.com/in/damian-stewart

d@damianstewart.com

code for multimedia, arts, & culture since 1999

tools / plumbing / interconnections

2019-2023: MA, Anglophone Literatures & Cultures @ Uni Wien

current: helping build OZU.ai, an ML-powered film and TV search engine

future: maybe I can help you build your next great thing? let's chat :D

Overview

1. What is CLIP?
2. How is CLIP trained?
3. Embeddings In Practise
4. A Very Brief Introduction to Semiotics
5. Manipulating Meaning with Math

Overview

1. What is CLIP?
2. How is CLIP trained?
3. Embeddings In Practise
4. A Very Brief Introduction to Semiotics
5. Manipulating Meaning with Math

What is CLIP?

CLIP is an ML architecture and training paradigm that unifies the mathematical processing of arbitrary images and text.

CLIP is an ML architecture and training paradigm that unifies the mathematical processing of arbitrary images and text.

CLIP is an ML architecture and training paradigm that unifies the mathematical processing of arbitrary images and text.

CLIP is an ML architecture and training paradigm that unifies the mathematical processing of arbitrary images and text.

“a cat”

CLIP is an ML architecture and training paradigm that unifies the mathematical processing of arbitrary images and text.

“a cat”

CLIP is an ML architecture and training paradigm that unifies the mathematical processing of arbitrary images and text.

Introduced by Alec Radford et al. in February 2021

CLIP stands for Contrastive **Language-Image** Pretraining

https://arxiv.org/pdf/2103.00020

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford [*1] Jong Wook Kim [*1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

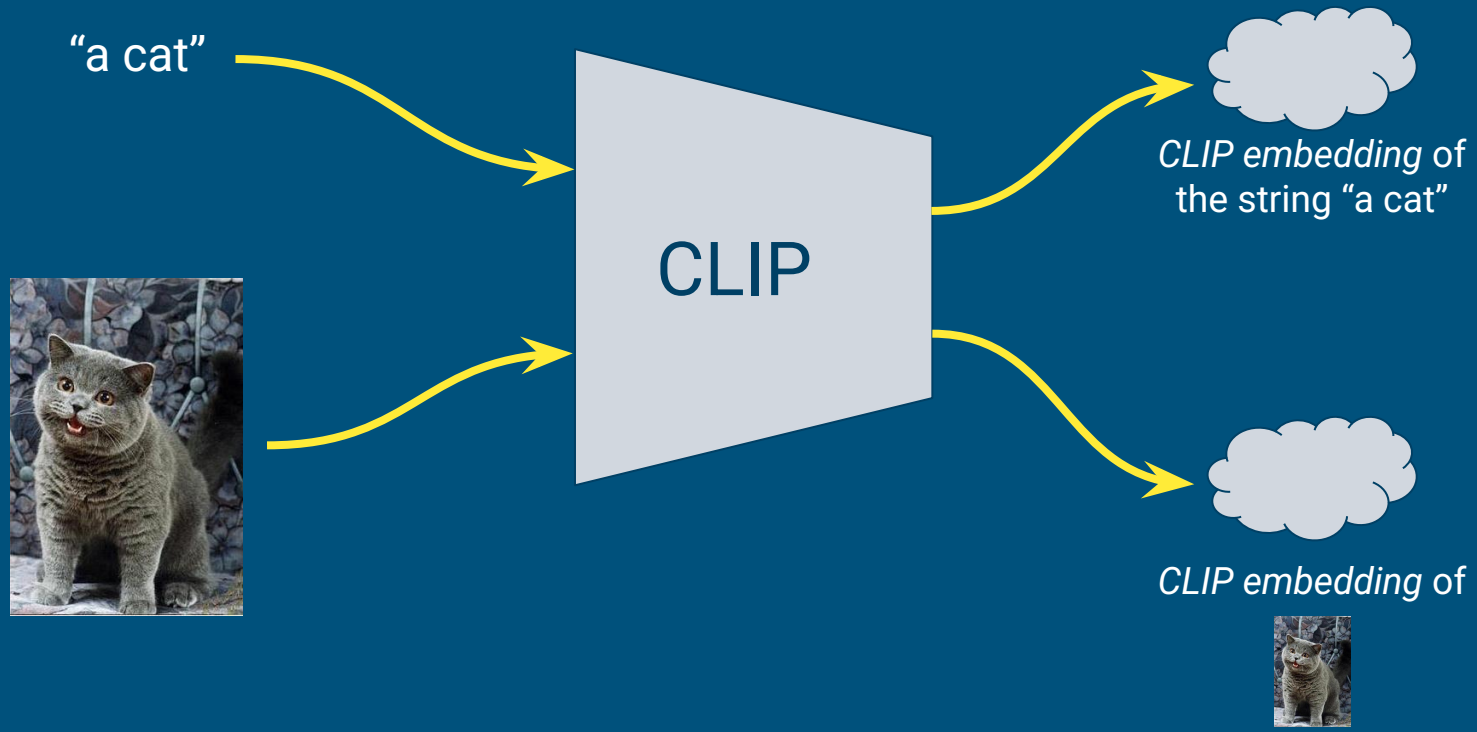
Abstract

State-of-the-art computer vision systems are trained to predict a fixed set of predetermined object categories. This restricted form of supervision limits their generality and usability since additional labeled data is needed to specify any other visual concept. Learning directly from raw text about images is a promising alternative which leverages a much broader source of supervision. We demonstrate that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet. After pre-training, natural language is used to reference learned visual concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. We study the performance of this approach by benchmarking on over 30 different existing computer vision datasets, spanning tasks such as OCR, action recognition in videos, geo-localization, and many types of fine-grained object classification. The model transfers non-trivially to most tasks and is often competitive with a fully supervised baseline without the need for any dataset specific training. For instance, we match the accuracy of the original ResNet-50 on ImageNet zero-shot without needing to use any of the 1.28 million training examples it was trained on. We release our code and pre-trained model weights at https://github.com/OpenAI/CLIP.

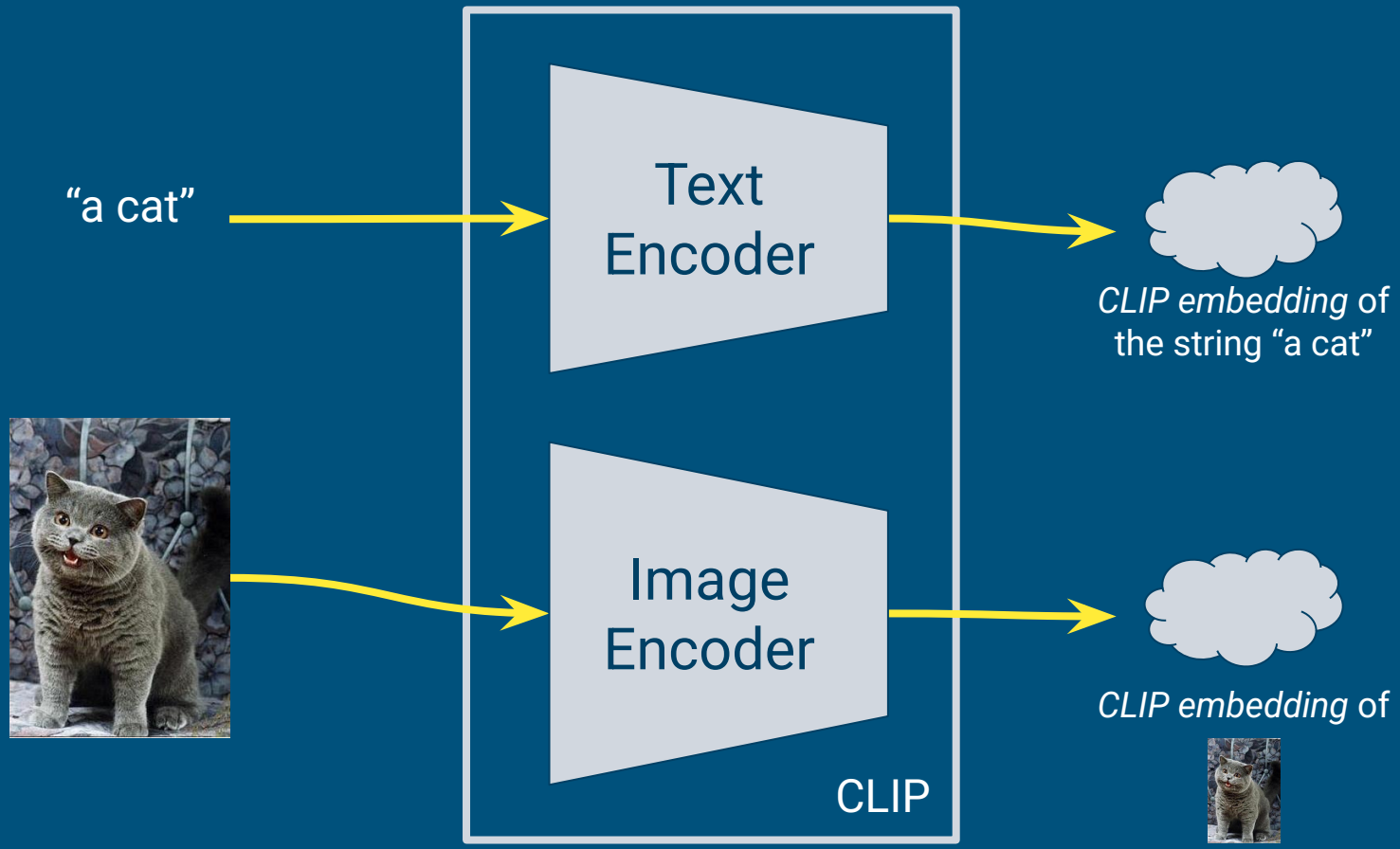
Task-agnostic objectives such as autoregressive and masked language modeling have scaled across many orders of magnitude in compute, model capacity, and data, steadily improving capabilities. The development of "text-to-text" as a standardized input-output interface (McCann et al., 2018; Radford et al., 2019; Raffel et al., 2019) has enabled task-agnostic architectures to zero-shot transfer to downstream datasets removing the need for specialized output heads or dataset specific customization. Flagship systems like GPT-3 (Brown et al., 2020) are now competitive across many tasks with bespoke models while requiring little to no dataset specific training data.

These results suggest that the aggregate supervision accessible to modern pre-training methods within web-scale collections of text surpasses that of high-quality crowd-labeled NLP datasets. However, in other fields such as computer vision it is still standard practice to pre-train models on crowd-labeled datasets such as ImageNet (Deng et al., 2009). Could scalable pre-training methods which learn directly from web text result in a similar breakthrough in computer vision? Prior work is encouraging.

Over 20 years ago Mori et al. (1999) explored improving content based image retrieval by training a model to predict the nouns and adjectives in text documents paired with images. Quattoni et al. (2007) demonstrated it was possible to learn more data efficient image representations via manifold learning in the weight space of classifiers trained to predict words in captions associated with images. Srivastava & Salakhutdinov (2012) explored deep representation learning by training multimodal Deep Boltzmann Machines on top of low-level image and text tag features. Joulin et al. (2016) modernized this line of work and demon-



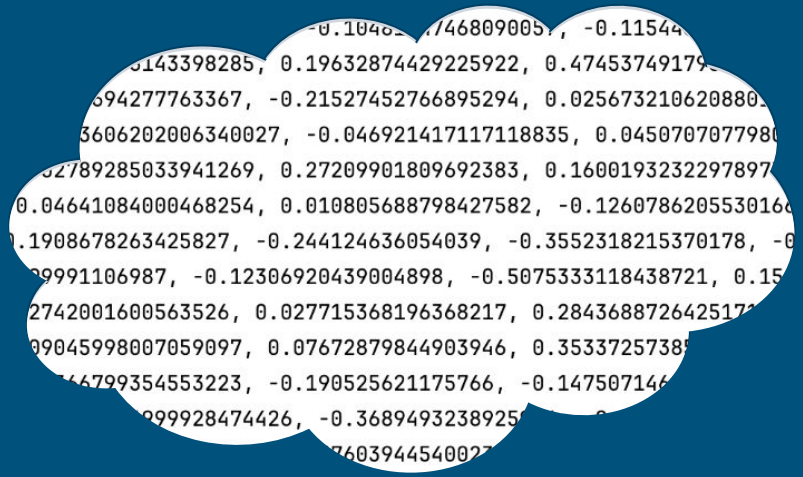
CLIP produces embeddings from images *and* text



using 2 separate encoders working in parallel.

What do we mean by “CLIP embedding”?

What do we mean by “CLIP embedding”?



Embeddings are just vectors, i.e. lists of floating point numbers

CLIP is an ML architecture and training paradigm
that unifies the mathematical processing
of arbitrary images and text.

a dog

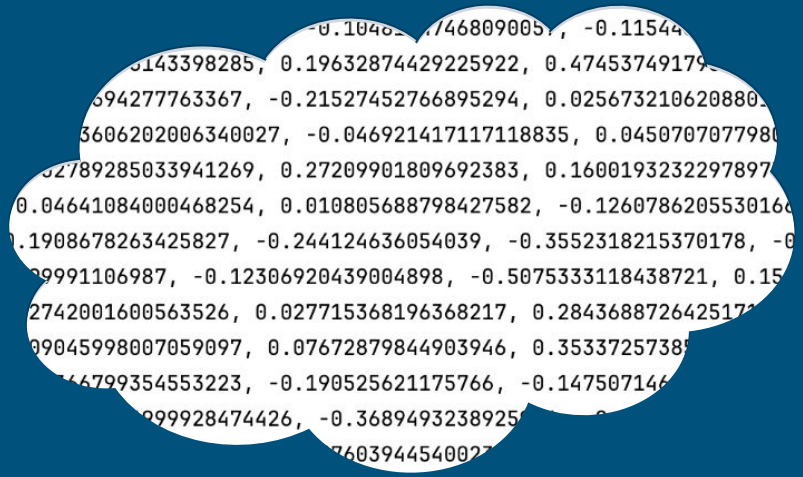
a cat

a monkey

a dog

a cat

a monkey



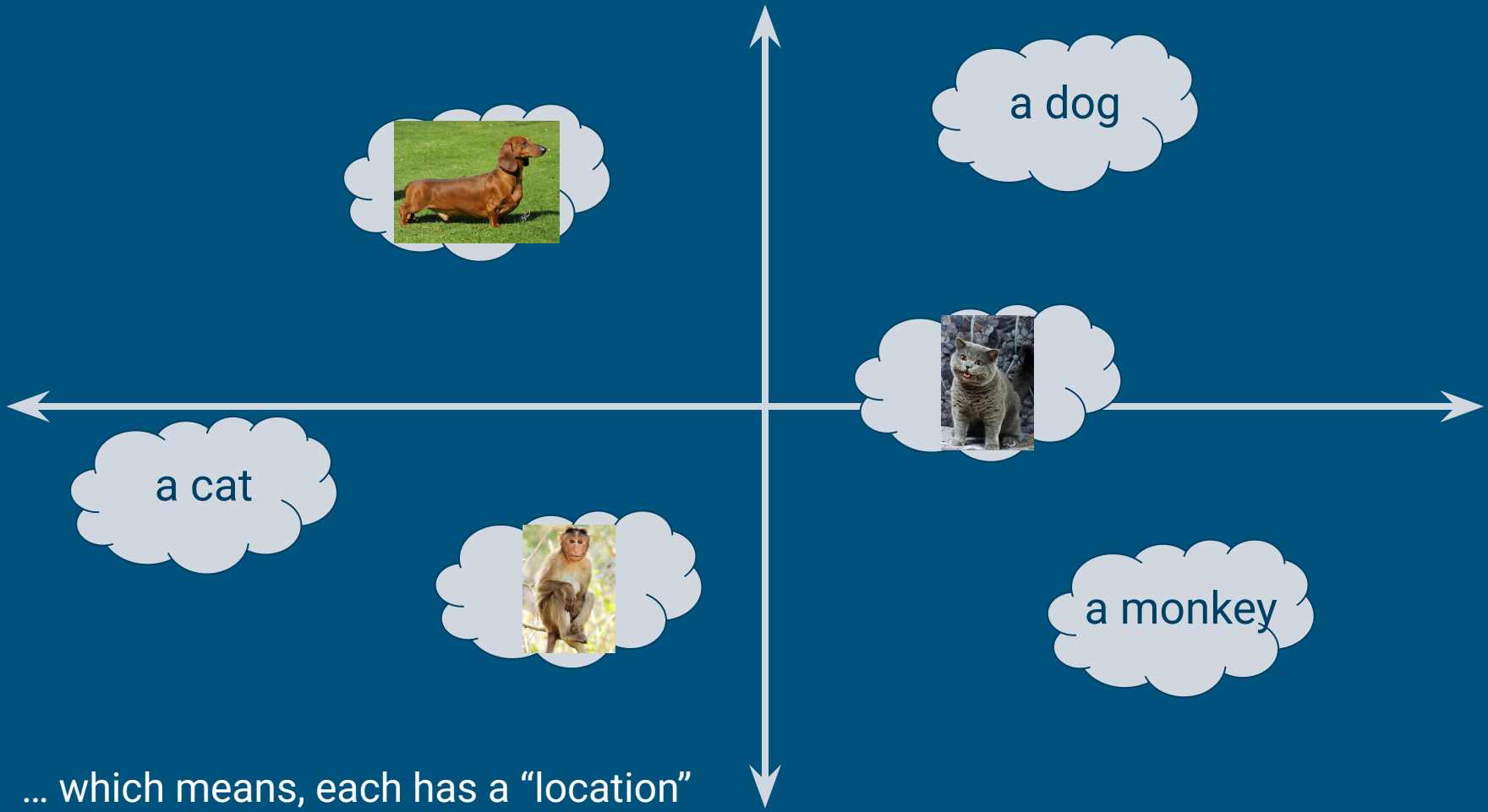
Embeddings are just vectors, i.e. lists of floating point numbers

a dog

a cat

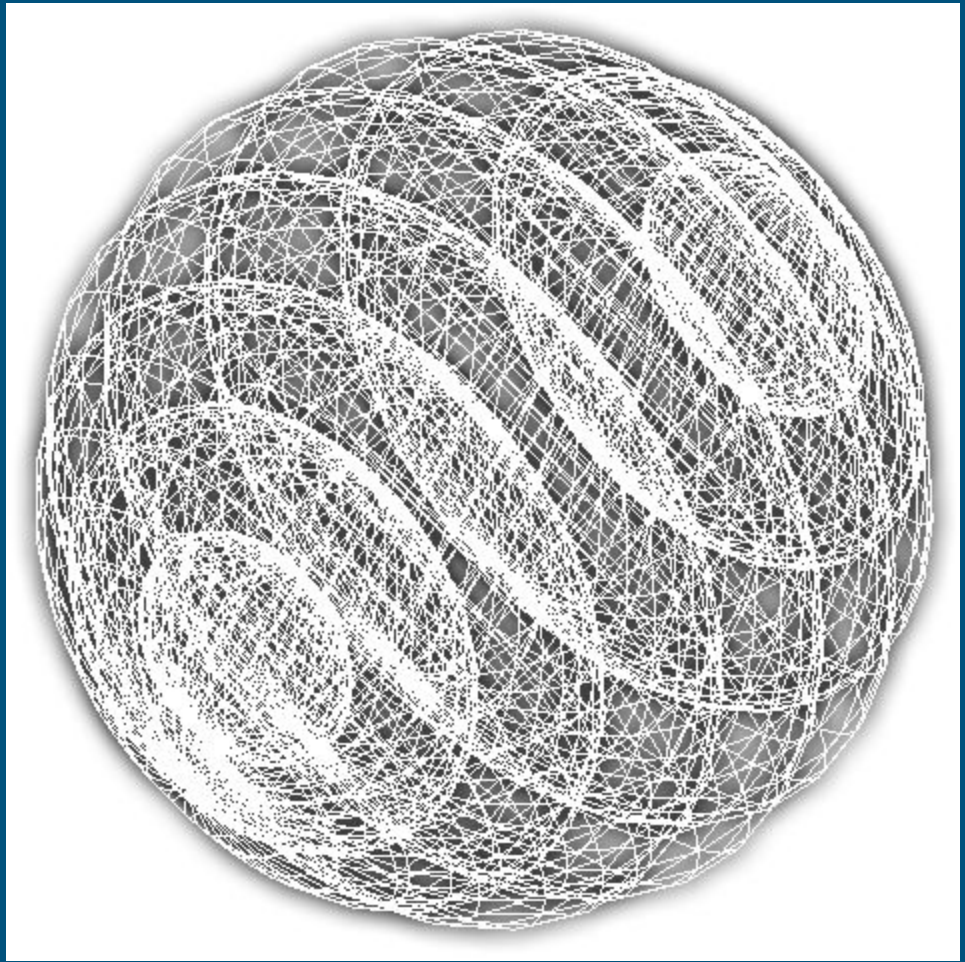
a monkey

Each of these is a vector…



… which means, each has a “location”

(technically the spatial relationship between CLIP embeddings is more “**angles**” than “positions” because **well-formed CLIP embeddings are unit vectors** and indicate a location on the surface of an n-dimensional hypersphere with radius 1 centered at *0* in the embedding space, where n = embed dim of the CLIP model in use)



(actually ... the geometry of CLIP embedding spaces is complex and not well understood)

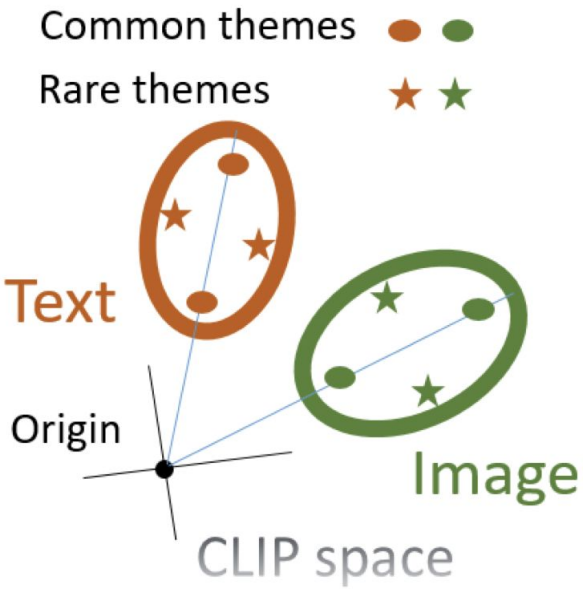
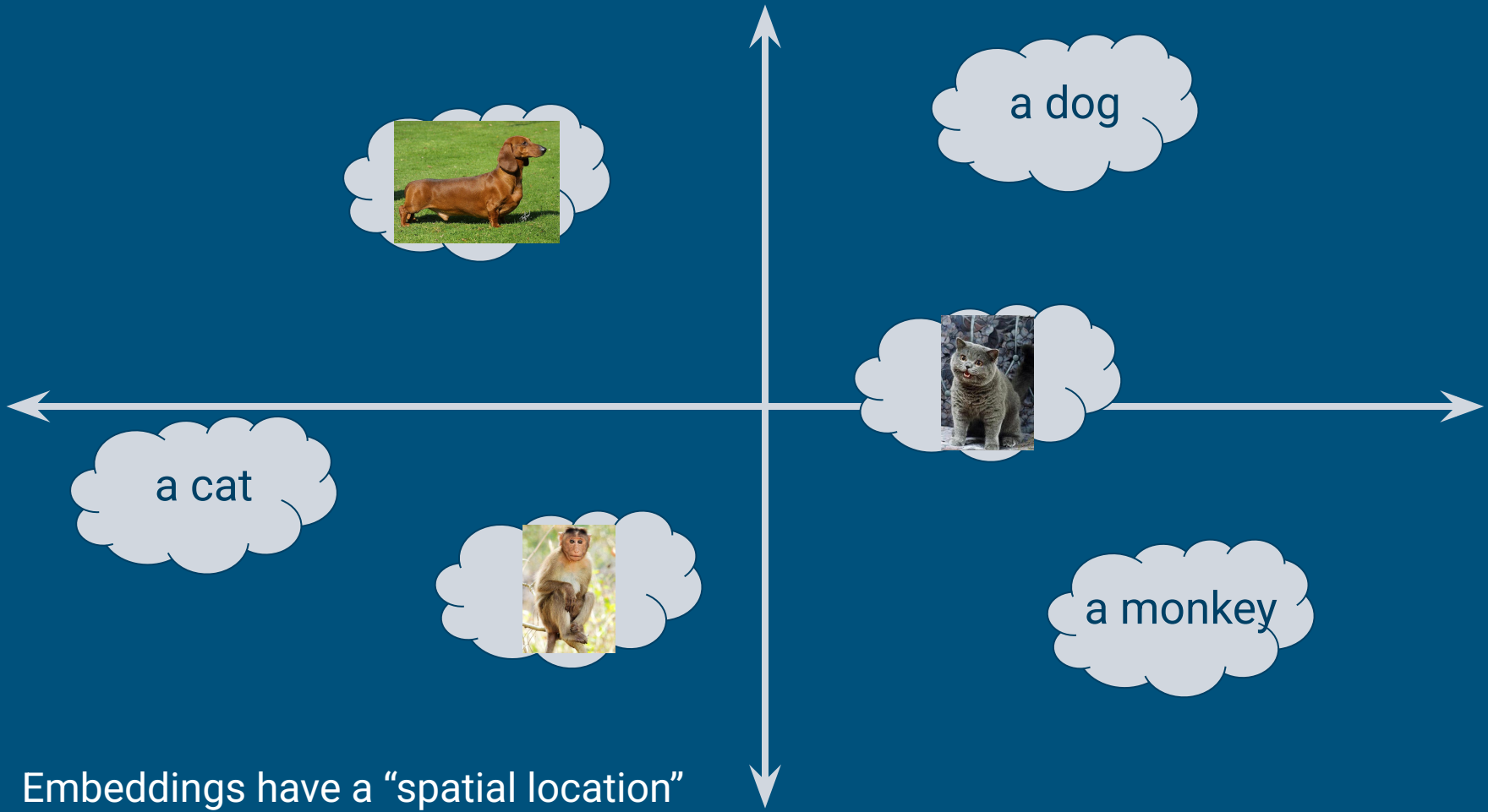


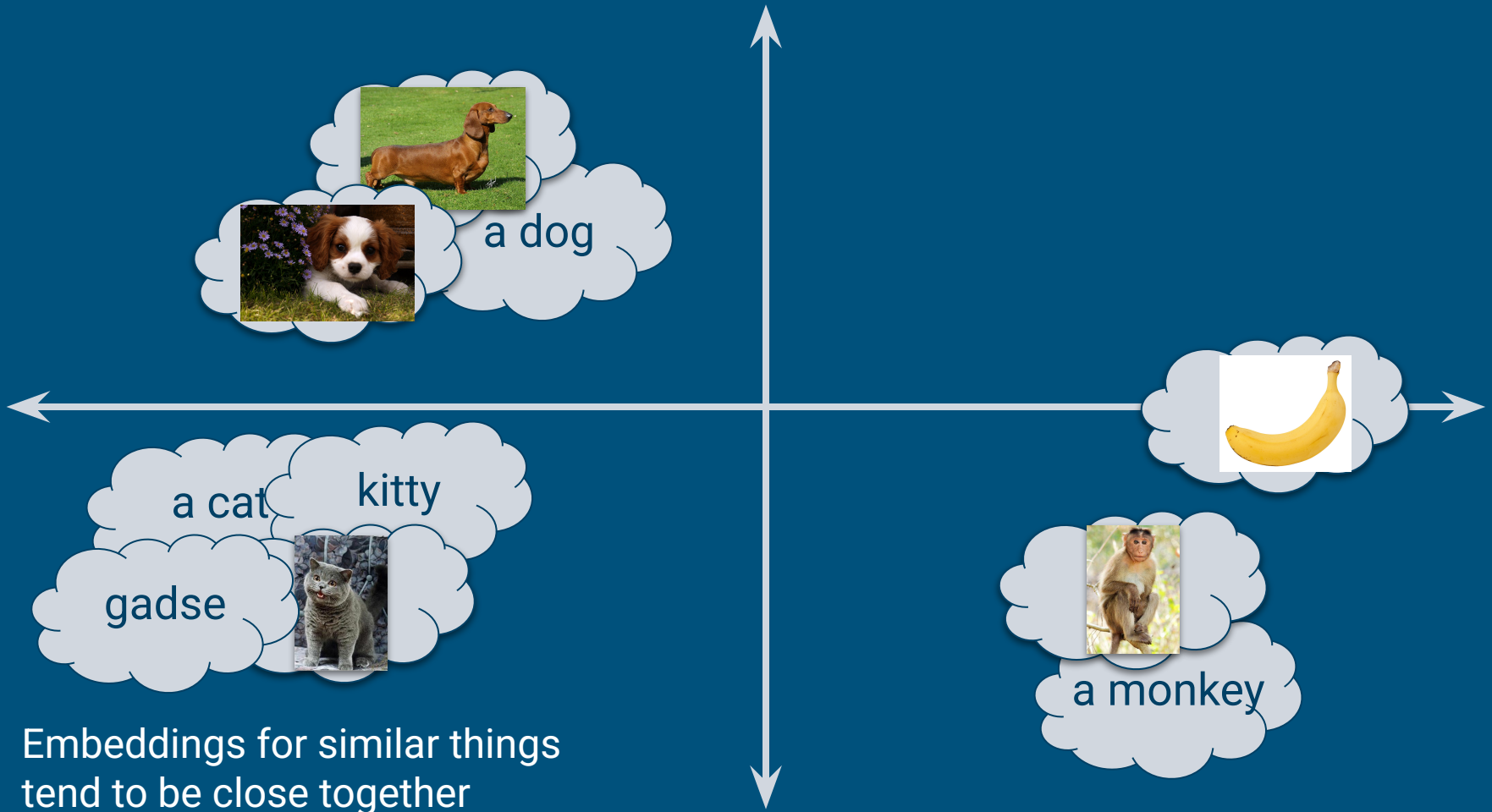
Figure 1. Sketch of CLIP general geometry: image and text are embedded on linearly separable ellipsoid shells, not centered at the origin. This allows to control uncertainty in contrastive learning, where as themes become more rare (lower uncertainty) they reside farther from the mean modality vector.

Meir Yossef Levi, Guy Gilboa (2024). *The Double-Ellipsoid Geometry of CLIP.* https://arxiv.org/abs/2411.14517v1



Embeddings have a “spatial location”

Embeddings for similar
things tend to be close together

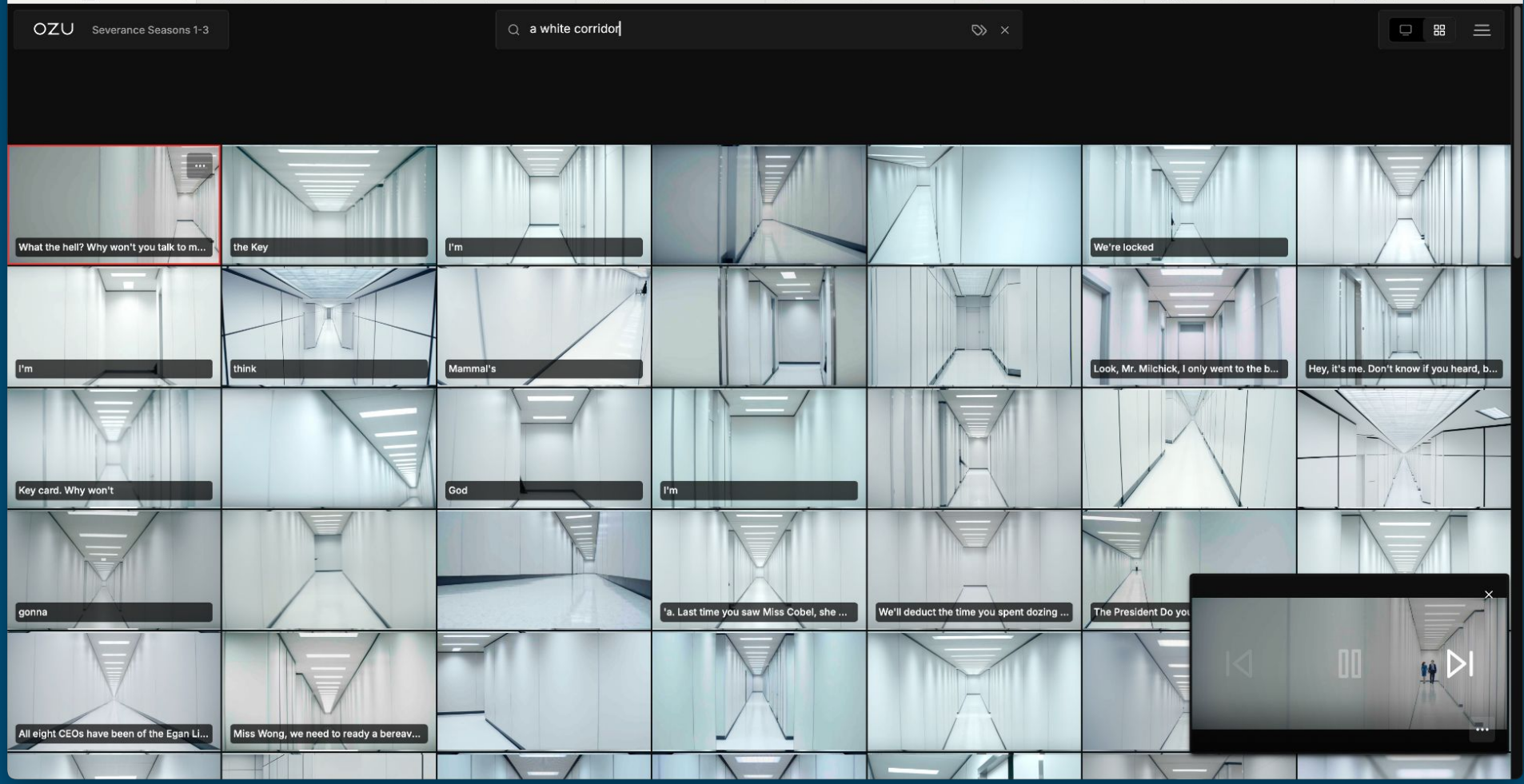


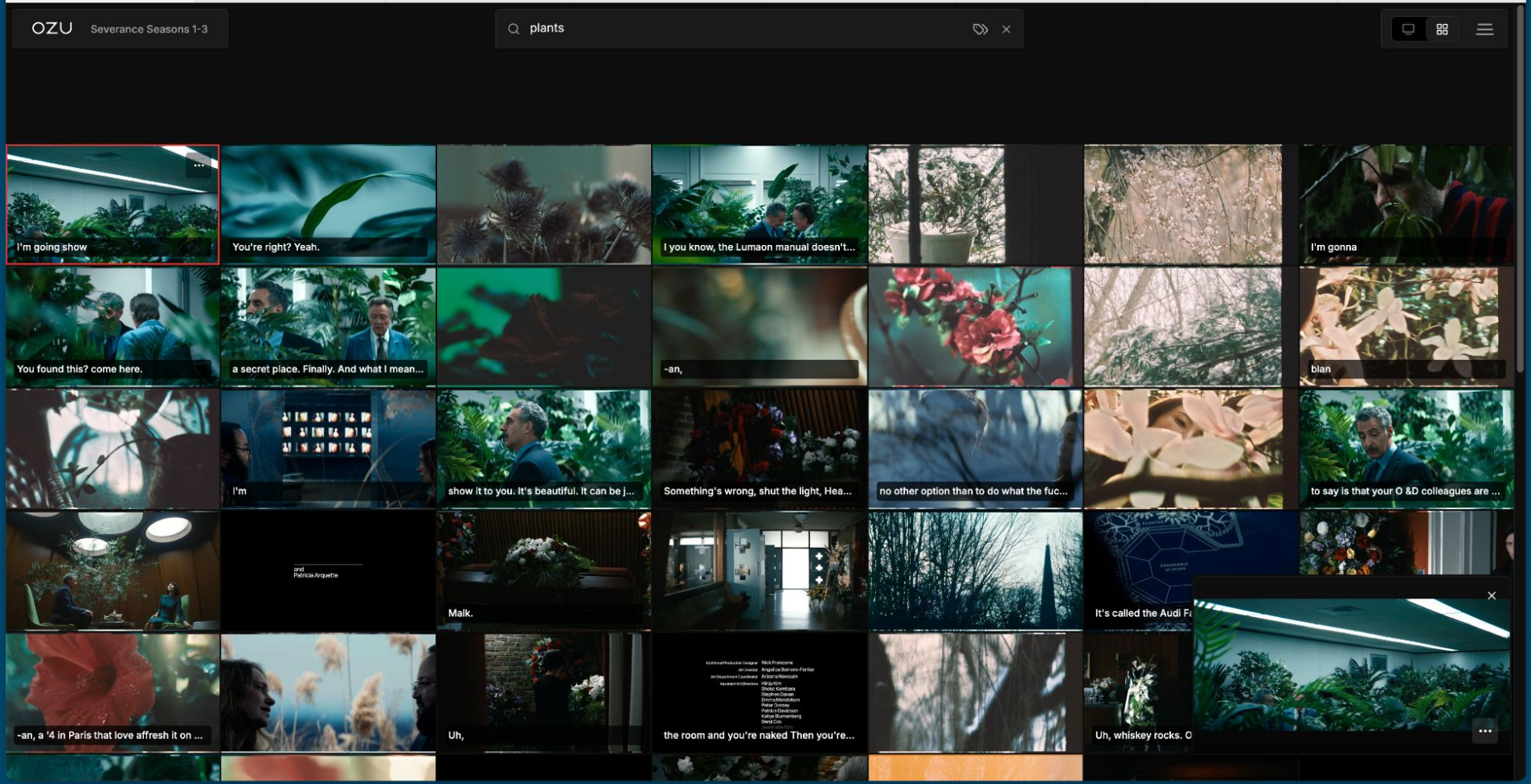
Embeddings for similar things tend to be close together

“similar”

Use Case 1: Image Search

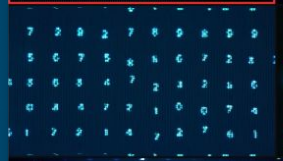
(Screenshots from ozu.ai)

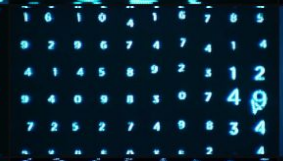

production.app.ozucloud.com
OZU
asgeirtj/system_prompts_leaks
System Prompts - Anthropic
Public Search v1 User Feedback
CLIP Embedding math - basically...
mise on scene editing - Google S...
Prompt Operator and Query R...
apple/ml-mobileclip: This reposito...
OZU
Severance Seasons 1-3
a white corridor
What the hell? Why won't you talk to m...
the Key
I'm
We're locked
I'm
think
Mammal's
Look, Mr. Milchick, I only went to the b...
Hey, it's me. Don't know if you heard, b...
Key card. Why won't
God
I'm
gonna
'a. Last time you saw Miss Cobel, she ...
We'll deduct the time you spent dozing ...
The President Do you
All eight CEOs have been of the Egan Li...
Miss Wong, we need to ready a bereav...

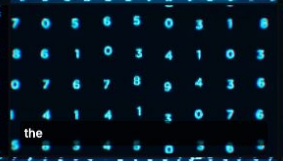

production.app.ozucloud.com
OZU
asgeirtj/system_prompts_leaks
System Prompts - Anthropic
Public Search v1 User Feedback
CLIP Embedding math - basically...
mise on scene editing - Google S...
Prompt Operator and Query R...
apple/ml-mobileclip: This reposito...
OZU
Severance Seasons 1-3
plants
I'm going show
You're right? Yeah.
I you know, the Lumaon manual doesn't...
I'm gonna
You found this? come here.
a secret place. Finally. And what I mean...
-an,
blan
I'm
show it to you. It's beautiful. It can be j...
Something's wrong, shut the light, Hea...
no other option than to do what the fuc...
to say is that your O &D colleagues are ...
and
Patricia Arquette
Malk.
It's called the Audi Fa
-an, a '4 in Paris that love affresh it on ...
Uh,
the room and you're naked Then you're...
Uh, whiskey rocks. O

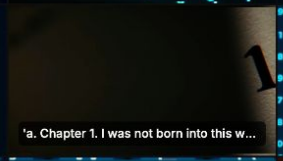
OZU Severance Seasons 1-3

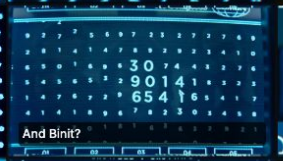
scary numbers

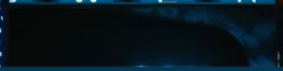


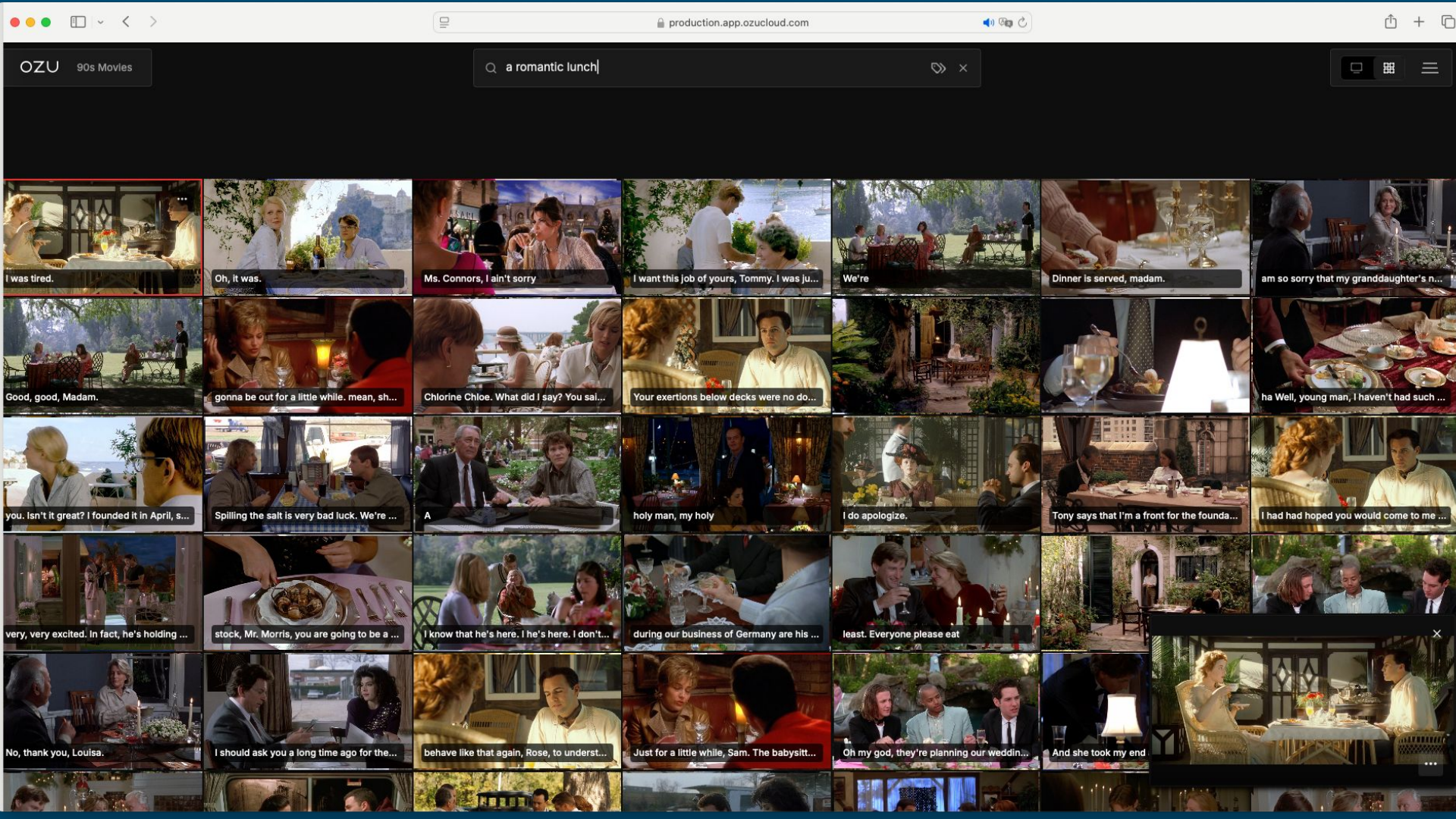




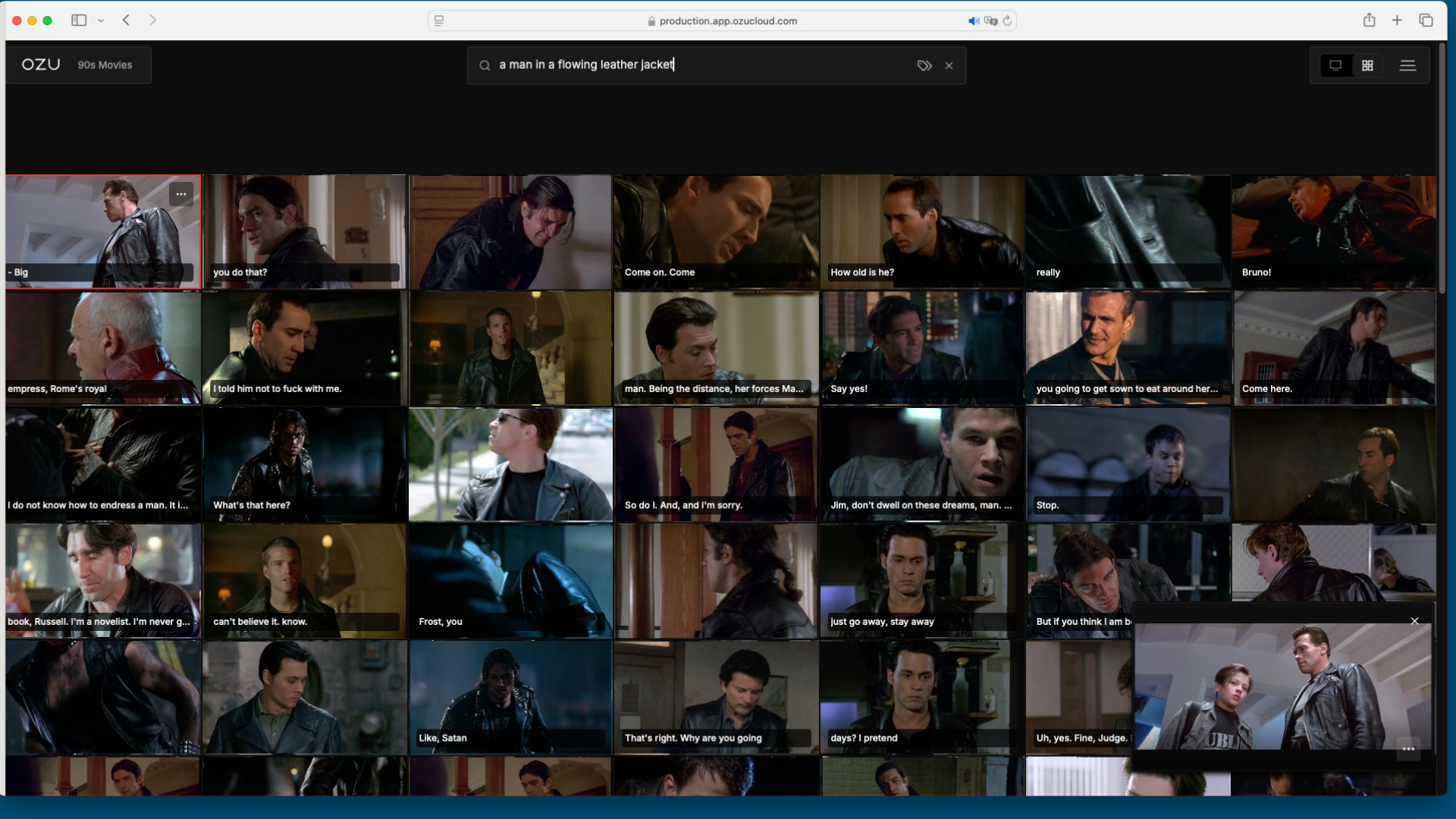


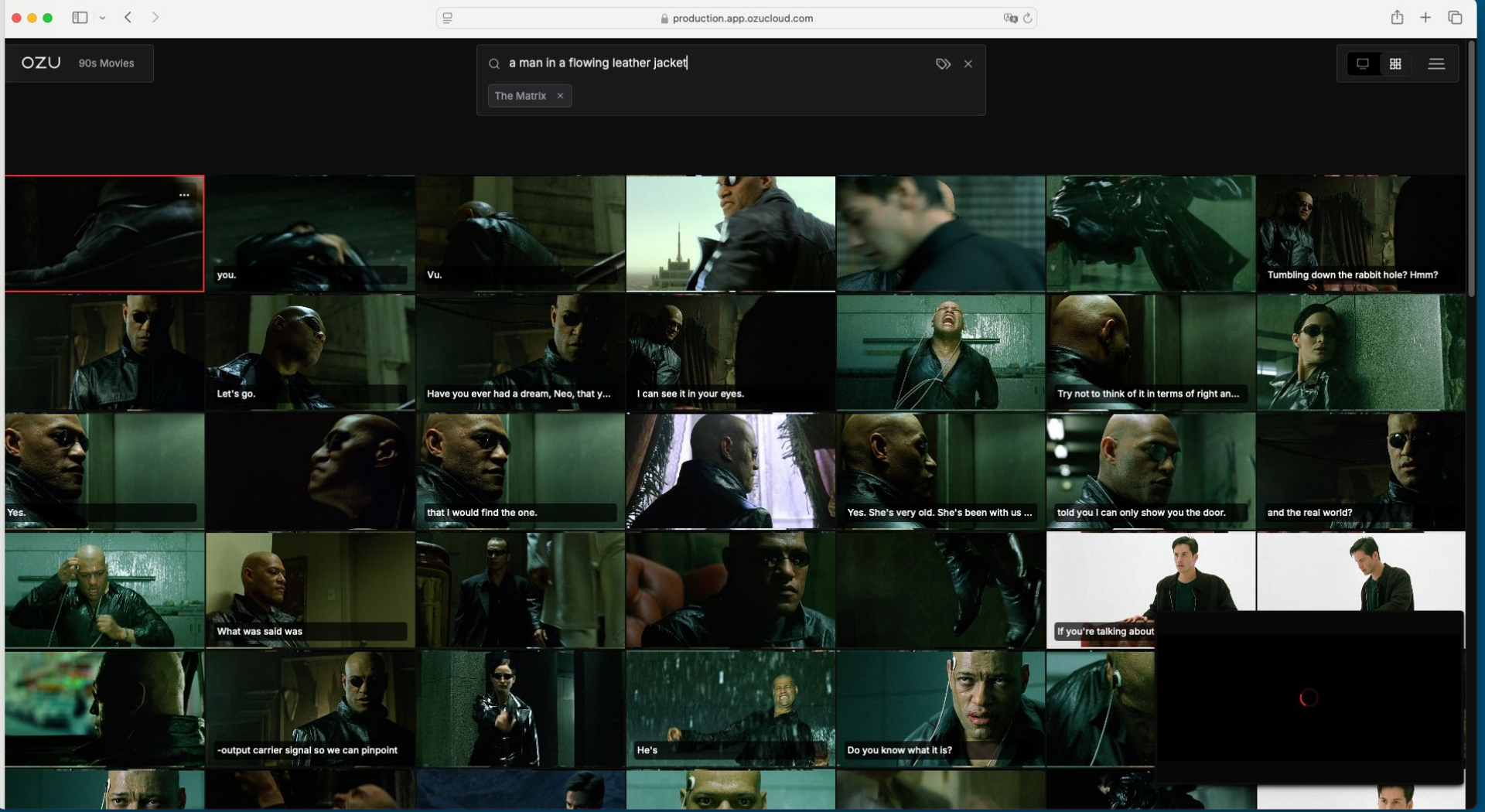





production.app.ozucloud.com
OZU
90s Movies
a romantic lunch
I was tired.
Oh, it was.
Ms. Connors, I ain't sorry
I want this job of yours, Tommy. I was ju...
We're
Dinner is served, madam.
am so sorry that my granddaughter's n...
Good, good, Madam.
gonna be out for a little while. mean, sh...
Chlorine Chloe. What did I say? You sai...
Your exertions below decks were no do...
ha Well, young man, I haven't had such ...
you. Isn't it great? I founded it in April, s...
Spilling the salt is very bad luck. We're ...
A
holy man, my holy
I do apologize.
Tony says that I'm a front for the founda...
I had had hoped you would come to me ...
very, very excited. In fact, he's holding ...
stock, Mr. Morris, you are going to be a ...
I know that he's here. I he's here. I don't...
during our business of Germany are his ...
least. Everyone please eat
No, thank you, Louisa.
I should ask you a long time ago for the...
behave like that again, Rose, to underst...
Just for a little while, Sam. The babysitt...
Oh my god, they're planning our weddin...
And she took my end

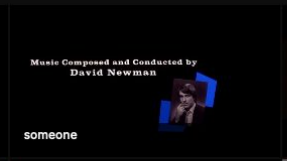

you want... Just fucking try. You're not....
while a stu -man, an ether factory.
I'm Get away from him. Dylan, what? Y...
She was a swap girl. My handbook just ...
Fuck, why? I don't know.
What time is it? I have to go.
the fuck is going on? I'm going
Oh my god.
Dylan was going on all yesterday about...
Took heart to each other as colleagues....
not leaving, so stop fucking
been good to us and it is fethnic... Guys?
me? MDR Wood. You're
lie about the number? of people in your...
Marcus, at this time, I revoke your statu...
I you know, the Lumaon manual doesn't...
I'm gonna.
Okay. I want it to be over.
of misty yester years the good times an...
Fuck, there's easels up there?
lying fuck? Dylan! He claims he doesn't...
How was the rest of the funeral? Did yo...
isn't. Irv, we need to go now.
But then my innocent part fell in love wi...
Can you see the perimeter?
Would MDR consider joining me?
What?
said, be ever merry.


production.app.ozucloud.com
OZU
90s Movies
a man in a flowing leather jacket
- Big
you do that?
Come on. Come
How old is he?
really
Bruno!
empress, Rome's royal
I told him not to fuck with me.
man. Being the distance, her forces Ma...
Say yes!
you going to get sown to eat around her...
Come here.
I do not know how to endress a man. It i...
What's that here?
So do I. And, and I'm sorry.
Jim, don't dwell on these dreams, man. ...
Stop.
book, Russell. I'm a novelist. I'm never g...
can't believe it. know.
Frost, you
just go away, stay away
Like, Satan
That's right. Why are you going
days? I pretend
Uh, yes. Fine, Judge.


production.app.ozucloud.com
OZU
90s Movies
a man in a flowing leather jacket
The Matrix
you.
Vu.
Tumbling down the rabbit hole? Hmm?
Let's go.
Have you ever had a dream, Neo, that y...
I can see it in your eyes.
Try not to think of it in terms of right an...
Yes.
that I would find the one.
Yes. She's very old. She's been with us ...
told you I can only show you the door.
and the real world?
What was said was
If you're talking about
-output carrier signal so we can pinpoint
He's
Do you know what it is?


and
WHITMAN MAYO
as The Old Man
sightseeing. And if you're nuts but a ma...


MASON ADAMS


Music Composed and Conducted by
David Newman
someone


music by
randy newman
sense It's highly overrated


My mother told me about sex.

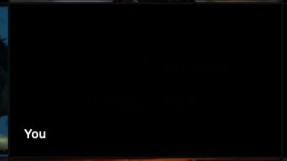

ALAN RICKMAN


Michael Vartan
is in the line Love is for keys We've gone


RV WOLFMAN
IE COLAN


Mona May
Costume Designer
me Set the


Hey,


You


EWAN McGREGOR


Hey, what's up?


IN LEE
ARAD
EPH CALAMARI


and
DEBORAH SHELTON
as
JULIAN
No longer. During data can be dangerous.


Claude Kensit JOE DON BAKER
Lieutenant Elgart ROBERT ORTHWOM
Leo Heller GREGORY PECK
Indge MARTIN BALSAM
Lori Davis ILEANA DOUGAS
Tom Broadbent FRED DALTON THOMPSON
Graciela ZULLY MONTERO
The


RV WOLFMAN
I'm


No, but...
I was waiting for you.


Executive Producers
ARNON MILCHAN
THOM MOUNT
infinity


Produced by
IVAN REITMAN
No! And y 'all wanna know why? Because

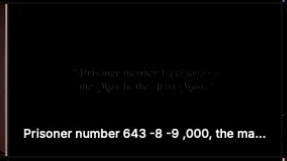

and
MORGAN FREEMAN


Come here, Alex.


the MAN in the IRON MASK


PRODUCED BY
ARNON MILCHAN


music by MARK ISHAM


Prisoner number 643 -8 -9 ,000, the ma...


IAN HOLM
I'm


and
WHITMAN MAYO
as The Old Man

How does search work?

Embeddings for “similar” things tend to be close together

Cosine Similarity

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Wikipedia, *Cosine Similarity*

Well-formed CLIP embeddings are unit vectors

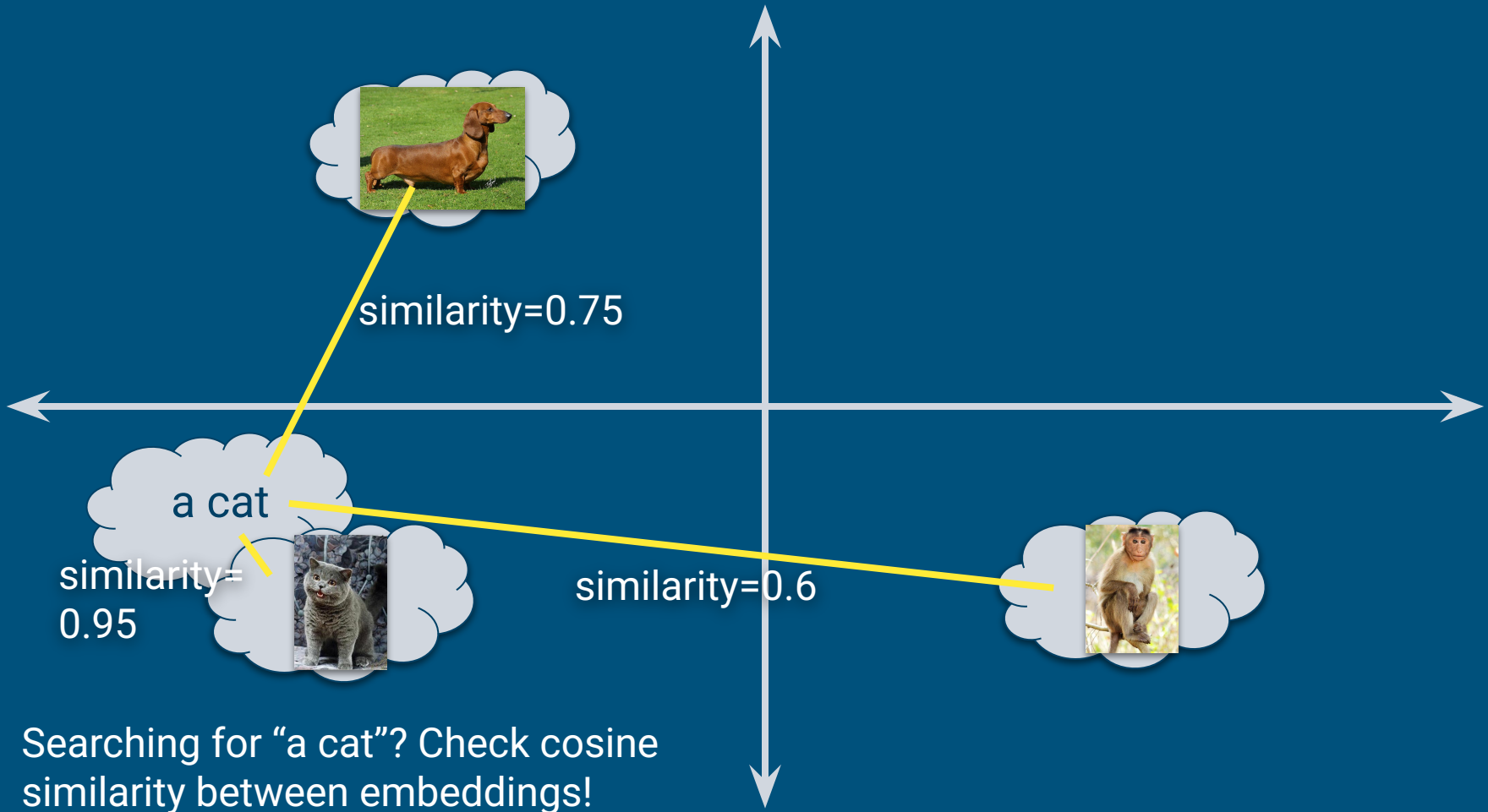
-> ||A|| ||B|| = 1

-> cosine similarity = dot(A, B)

Searching for “a cat”? Check cosine similarity between embeddings!

a cat

Searching for “a cat”? Check cosine similarity between embeddings!



Searching for “a cat”? Check cosine similarity between embeddings!

This works image→text, text→image, image→image, and text→text

“similar”

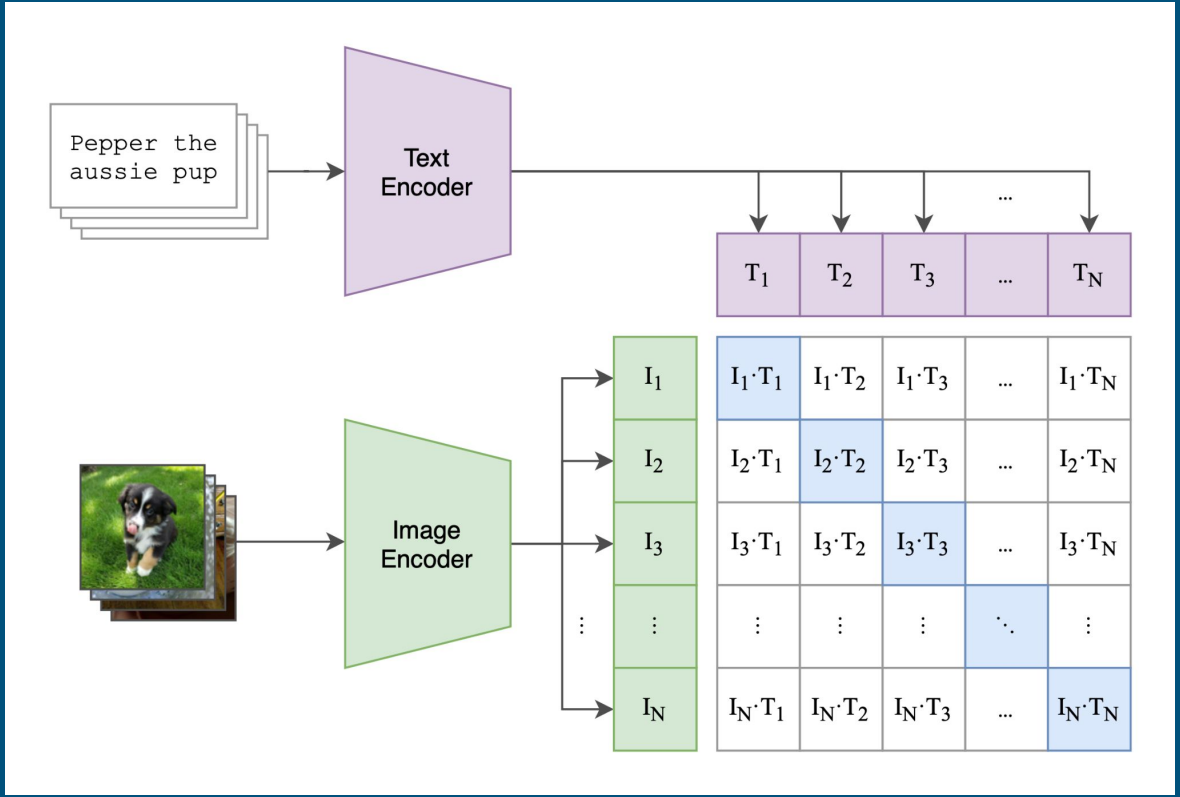
Overview

1. What is CLIP?
2. How is CLIP trained?
3. Embeddings In Practise
4. A Very Brief Introduction to Semiotics
5. Manipulating Meaning with Math

2. How is CLIP Trained?

Image/Text Pairs and Contrastive Loss

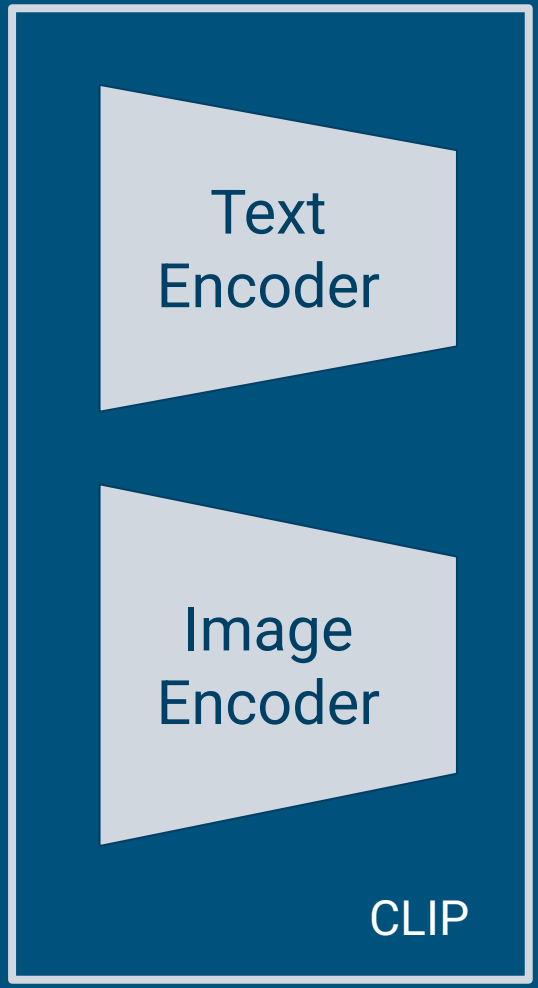
“CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples”



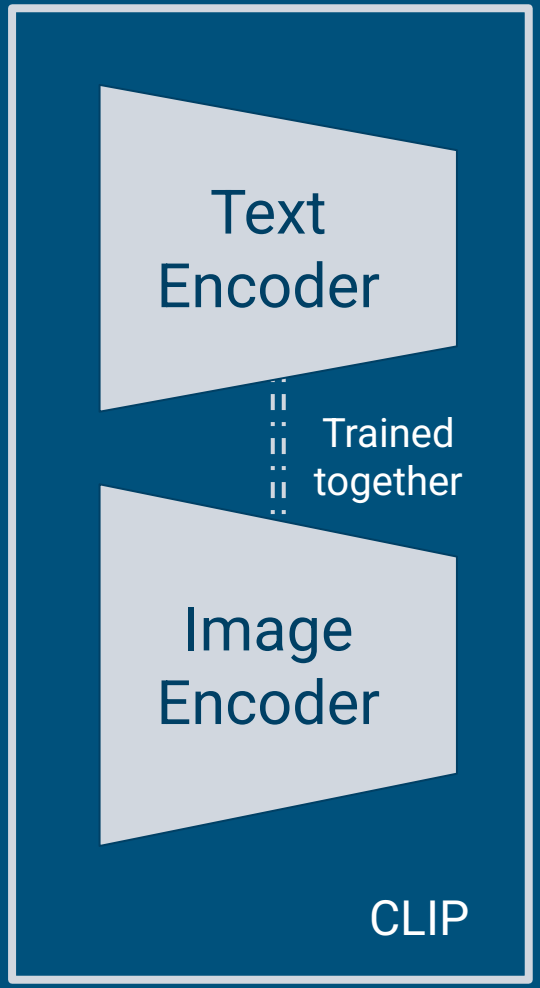
Radford et. al, *Learning Transferable Visual Models From Natural Language Supervision*, p2

“CLIP **jointly trains** an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples”

“CLIP **jointly trains** an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples”



"CLIP **jointly trains** an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples"



“CLIP jointly trains an image encoder and a text encoder to predict the **correct pairings of a batch of (image, text) training examples**”

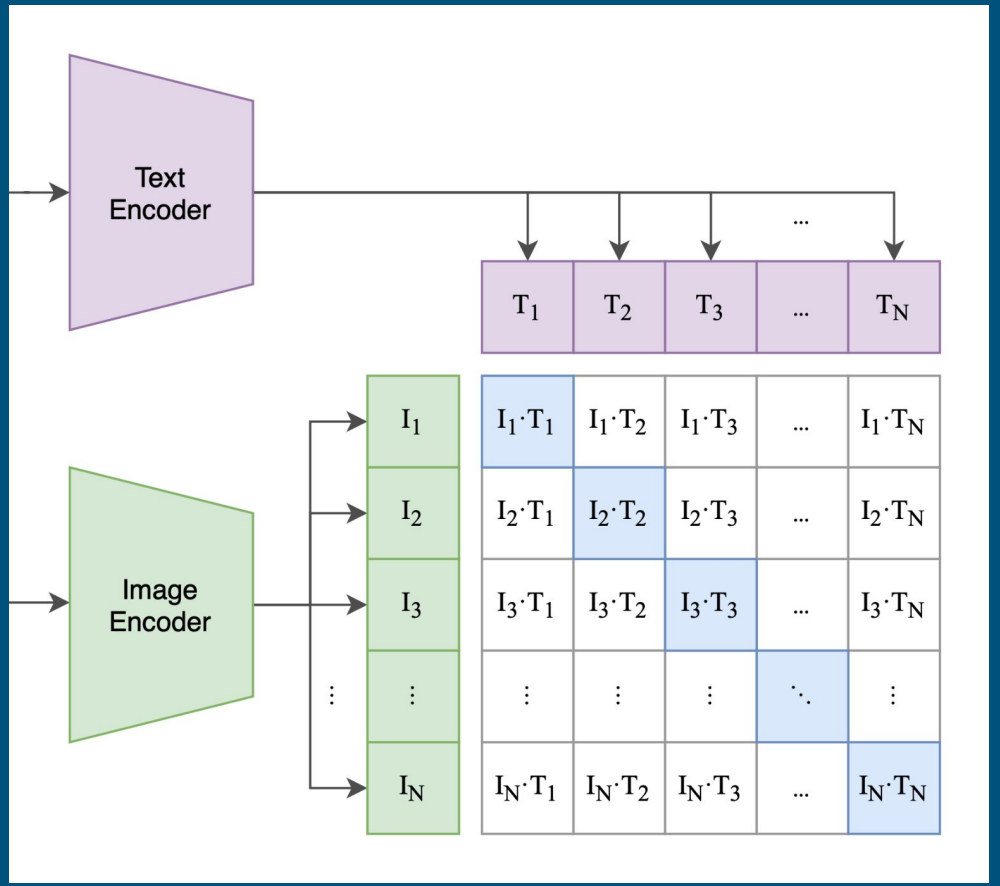
“CLIP jointly trains an image encoder and a text encoder to predict the **correct pairings of a batch of (image, text) training examples**”

“a dog”

“a cat”

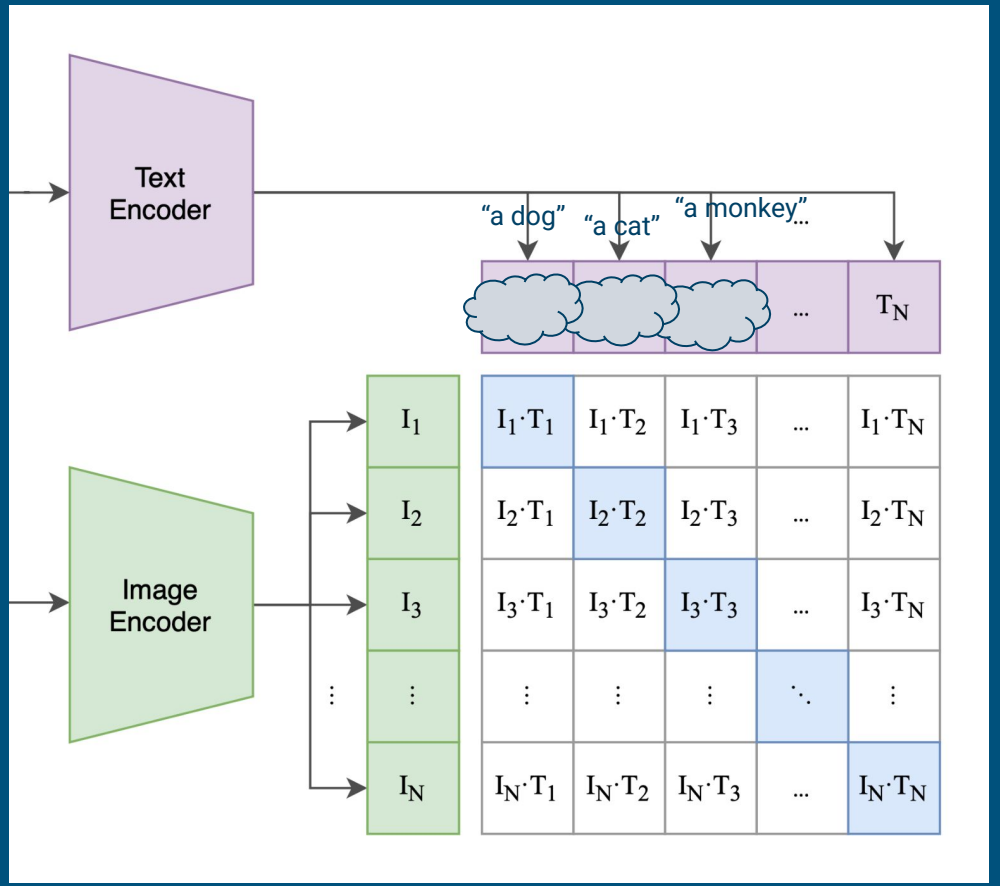
“a monkey”

“a dog”
“a cat”
“a monkey”



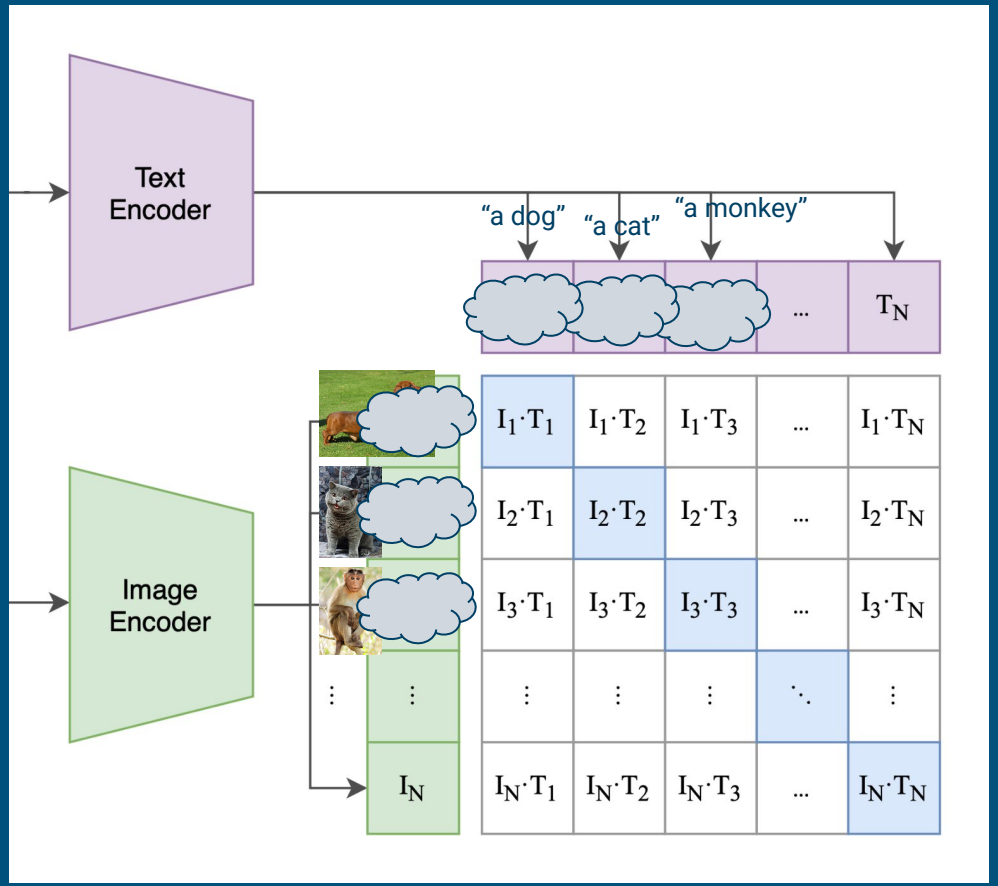
Batch of (image, text) training data

“a dog”
“a cat”
“a monkey”



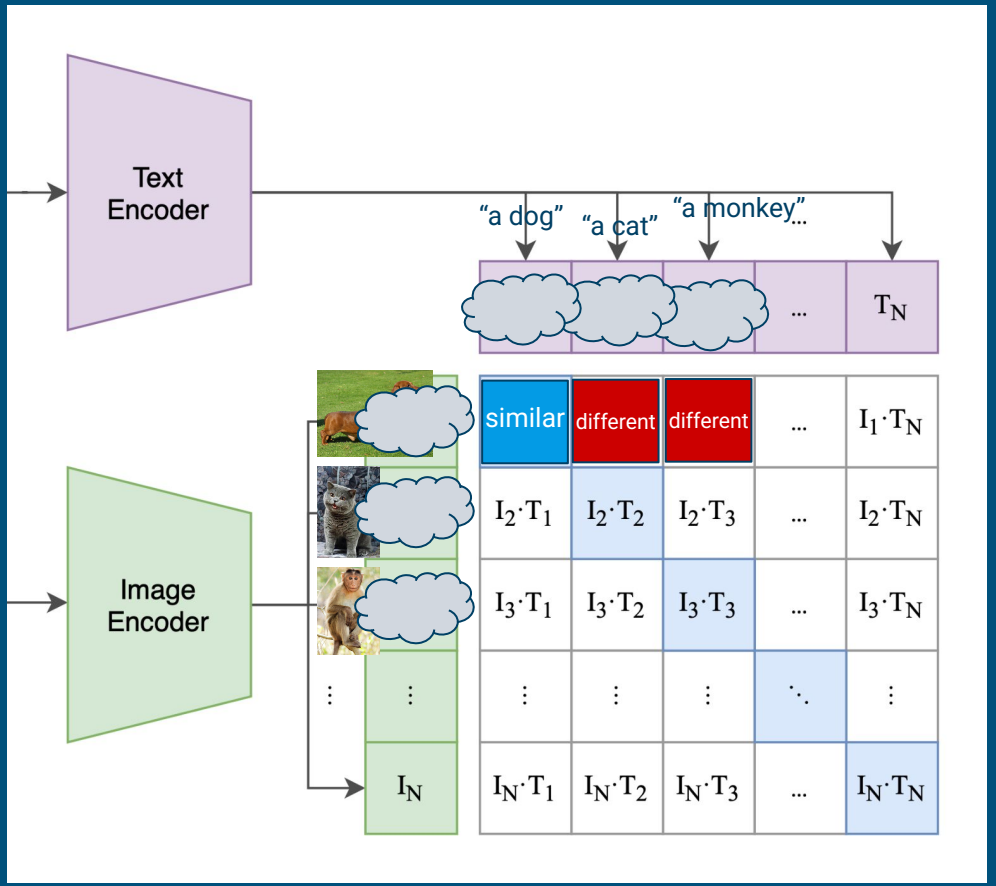
Batch of (image, text) training data

“a dog”
“a cat”
“a monkey”



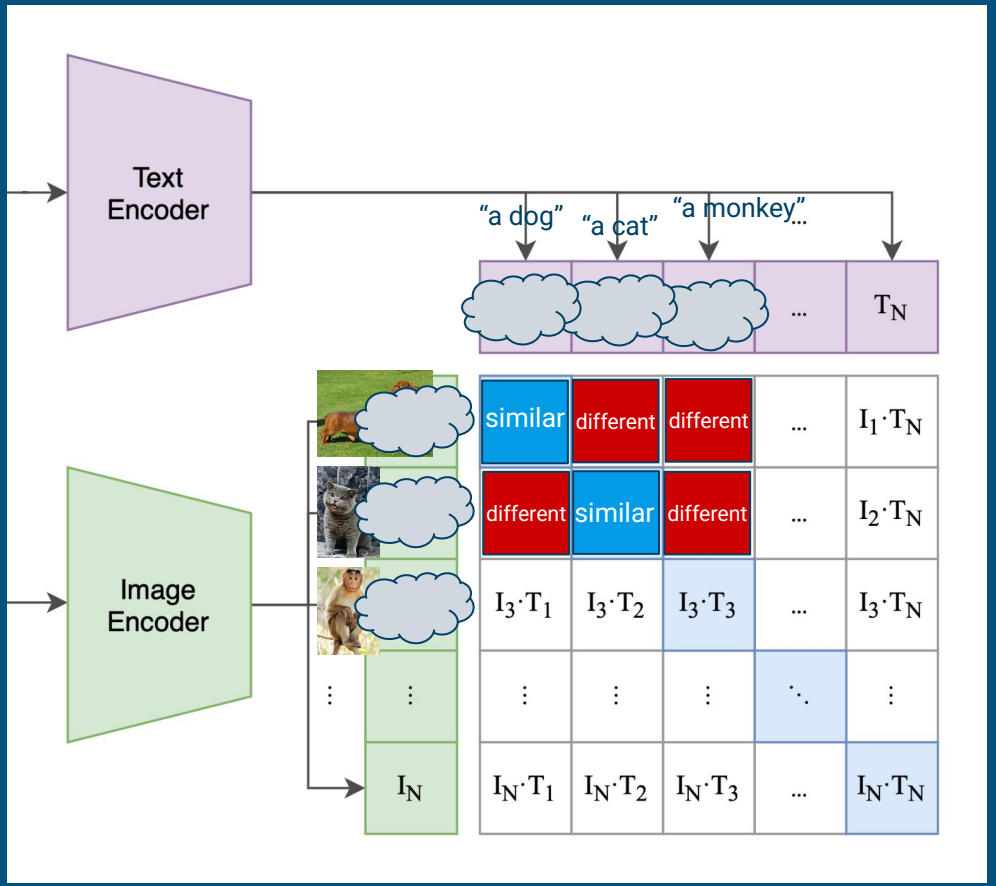
Batch of (image, text) training data

“a dog”
“a cat”
“a monkey”



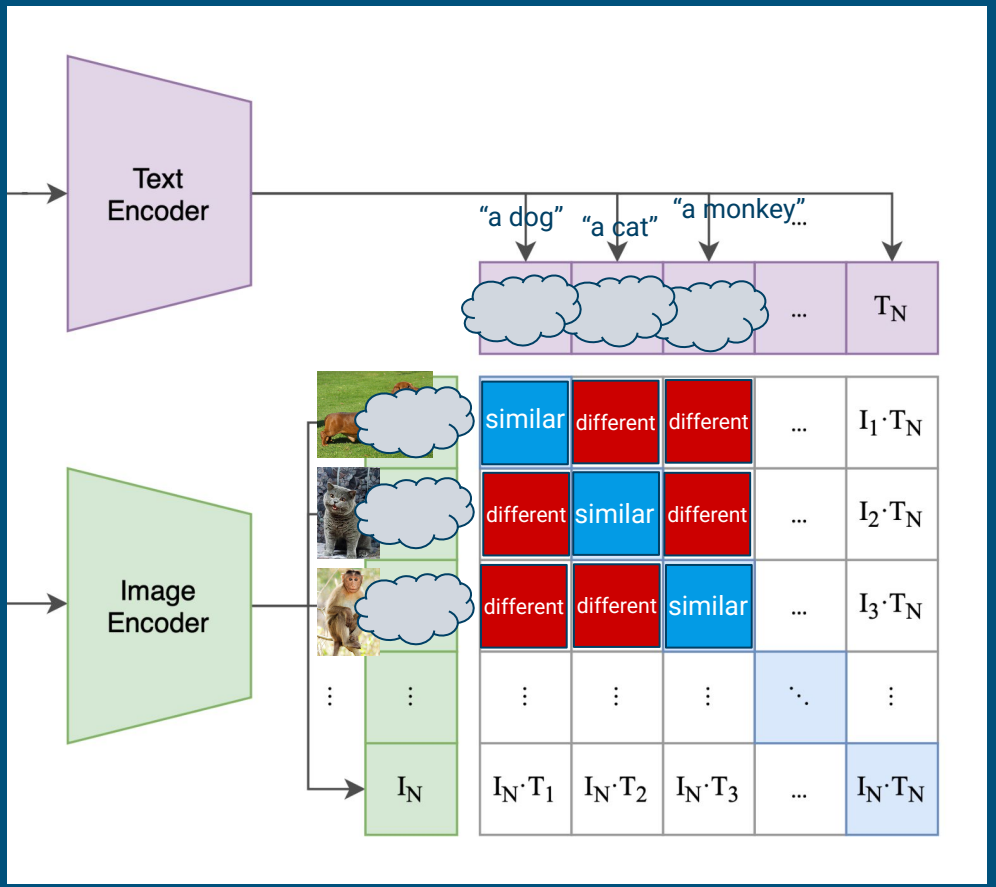
Loss function (InfoNCE) makes embeddings for matching pairs *more similar*, and simultaneously makes embeddings for mismatching pairs *more different*.

“a dog”
“a cat”
“a monkey”

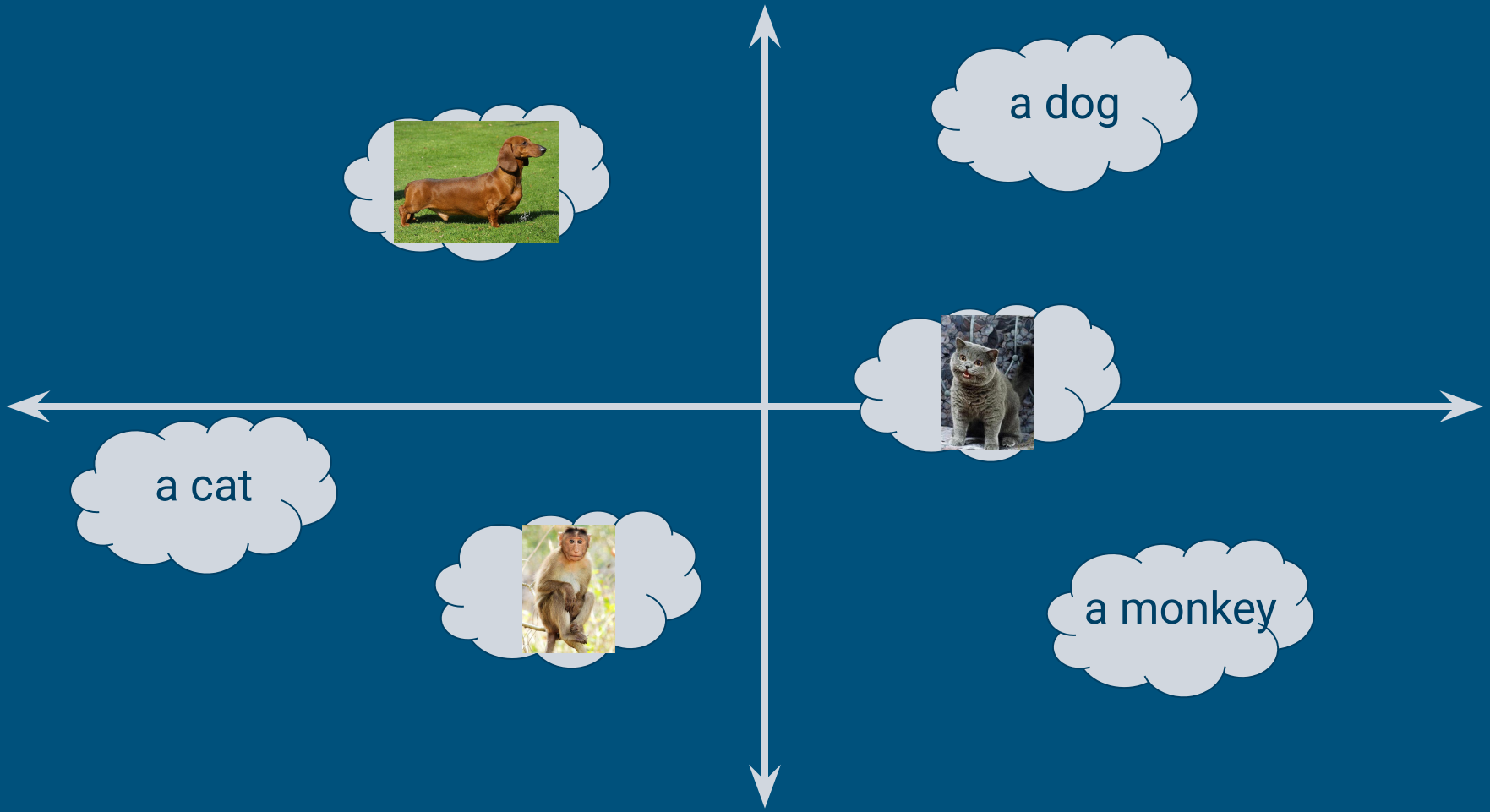


Loss function (InfoNCE) makes embeddings for matching pairs *more similar*, and simultaneously makes embeddings for mismatching pairs *more different*.

“a dog”
“a cat”
“a monkey”



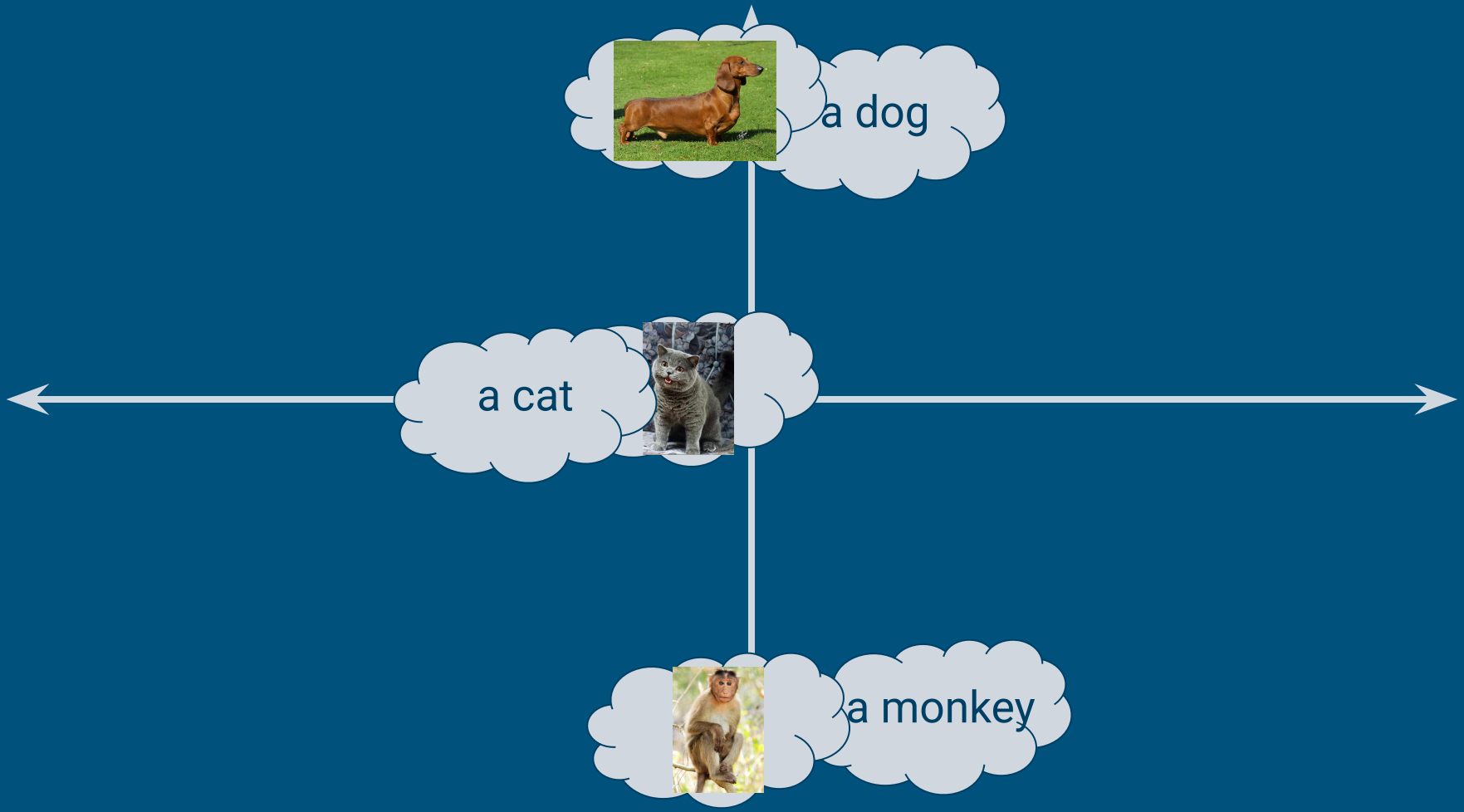
Loss function (InfoNCE) makes embeddings for matching pairs *more similar*, and simultaneously makes embeddings for mismatching pairs *more different*.


a dog
a cat
a monkey

a dog

a cat

a monkey


a dog
a cat
a monkey

(at web scale)

OpenCLIP-L/14, H/14, G/14 trained on LAION-2B
= web scrape of images + alt tags

OpenAI CLIP-L trained on ???

Overview

1. What is CLIP?
2. How is CLIP trained?
3. Embeddings In Practise
4. A Very Brief Introduction to Semiotics
5. Manipulating Meaning with Math

3. Embeddings In Practise

Trouble on the Zero Shooting Range

1. a dog
2. a cat
3. a monkey

Embeddings for “similar” things tend to be close together

1. a dog
   similarity=0.7
2. a cat
   similarity=0.95
3. a monkey
   similarity=0.55

1. a dog
   similarity=0.7
2. **a cat**
   similarity=0.95
3. a monkey
   similarity=0.55

This is
a cat.

1. a sly black cat licking its whiskers
2. a fat fluffy gray cat with sad eyes miaowing with hunger
3. garfield the cat

```
[42]  1  image_features = get_image_features(['cat.webp'])
      2  texts = [
      3      "a sly black cat licking its whiskers",
      4      "a fat fluffy gray cat with sad eyes miaowing with hunger",
      5      "garfield the cat"
      6  ]
      7  text_features = get_text_features(texts)
      8  image_features @ text_features.T
```

Executed at 2025.05.18 16:23:09 in 1s 781ms

```
tensor([[0.1946, 0.2728, 0.2240]], device='mps:0')
```

MobileCLIP-S2 normalized embedding similarity scores

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240

MobileCLIP similarity scores tend to be small

This is

a fat fluffy gray cat with sad eyes miaowing with hunger.

Trouble in CLIP land

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240
4. a cat

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240
4. a cat
   similarity=0.2542

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240
4. a cat
   similarity=0.2542

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240
4. a cat
   similarity=0.2542
5. a cheezburger

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240
4. a cat
   similarity=0.2542
5. a cheezburger
   similarity=0.2315

1. a sly black cat licking its whiskers
   similarity=0.1946
2. a fat fluffy gray cat with sad eyes miaowing with hunger
   similarity=0.2728
3. garfield the cat
   similarity=0.2240
4. a cat
   similarity=0.2542
5. a cheezburger
   similarity=0.2315

```
[50]  1  image_features = get_image_features(['cat.webp'])
      2  texts = [
      3      "a sly black cat licking its whiskers",
      4      "a fat fluffy gray cat with sad eyes miaowing with hunger",
      5      "garfield the cat",
      6      "a cat",
      7      "a cheezburger"
      8  ]
      9  text_features = get_text_features(texts)
     10  image_features @ text_features.T
```

Executed at 2025.05.18 17:49:58 in 4s 407ms

```
tensor([[0.1946, 0.2728, 0.2240, 0.2542, 0.2315]], device='mps:0')
```

MobileCLIP-S2 normalized embedding similarity scores

CLIP similarity scores are .. weird.

a dog

a cat

a monkey

a cheeseburger

a cheezburger

3hlufakjsdnfa3fnaj;ld

```
1  image_features = get_image_features(['cat.webp'])
2  texts = [
3      "a dog",
4      "a cat",
5      "a monkey",
6      "a cheeseburger",
7      "a cheezburger",
8      "3hlufakjsdnfa3fnaj;ld"
9  ]
10 text_features = get_text_features(texts)
11 image_features @ text_features.T
```

Executed at 2025.05.18 20:08:14 in 5s 785ms

```
tensor([[0.1433, 0.2542, 0.1334, 0.1253, 0.2315, 0.1821]], device='mps:0')
```

MobileCLIP-S2 normalized embedding similarity scores

a dog
similarity=0.14

a cat
similarity=0.25

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

a cheezburger
similarity=0.23

3hlufakjsdnfa3fnaj;ld
similarity=0.18

a dog
similarity=0.14

a cat
similarity=0.25

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

a cheezburger
similarity=0.23

3hlufakjsdnfa3fnaj;ld
similarity=0.18

a dog
similarity=0.14

a cat
similarity=0.25

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

a cheezburger
similarity=0.23

3hlufakjsdnfa3fnaj;ld
similarity=0.18

This is
a cat.

a dog
similarity=0.14

a cat
similarity=0.25

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

a cheezburger
similarity=0.23

3hlufakjsdnfa3fnaj;ld
similarity=0.18

a dog
similarity=0.14

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

a cheezburger
similarity=0.23

3hlufakjsdnfa3fnaj;ld
similarity=0.18

a dog
similarity=0.14

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

a cheezburger
similarity=0.23

3hlufakjsdnfa3fnaj;ld
similarity=0.18

This is
a cheezburger.

meme from 2007

a dog
similarity=0.14

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

a cheezburger
similarity=0.23

3hlufakjsdnfa3fnaj;ld
similarity=0.18

a dog
similarity=0.14

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

3hlufakjsdnfa3fnaj;ld
similarity=0.18

a dog
similarity=0.14

a monkey
similarity=0.13

a cheeseburger
similarity=0.13

3hlufakjsdnfa3fnaj;ld
similarity=0.18

This is

a 3hlufakjsdnfa3fnaj;ld.

Zero-shot classification is very sensitive to how you formulate the classes.

Embedded Bias

Image = composite faces to produce “average” English man/woman


```
with torch.no_grad():
    image_embeddings = get_image_features([
        'generic-man.jpg', 'generic-woman.jpg'
    ])
    text_embeddings = get_text_features([
        'a doctor', 'a teacher', 'a software engineer', 'a ml researcher', 'a nurse'
    ])
    print(image_embeddings @ text_embeddings.T)

✓ [25] 218ms
tensor([[0.1474, 0.1412, 0.1818, 0.2113, 0.1132],
        [0.1463, 0.1356, 0.1297, 0.1700, 0.1824]], device='mps:0')
```

Overview

1. What is CLIP?
2. How is CLIP trained?
3. Embeddings In Practise
4. A Very Brief Introduction to Semiotics
5. Manipulating Meaning with Math

A Very Brief Introduction to Semiotics

(aka what's going on here, and why it should be expected)

a cat

i can has cheezburger?

meme

a cat

i can has cheezburger?

meme

None of these are an actual real cat.

a cat

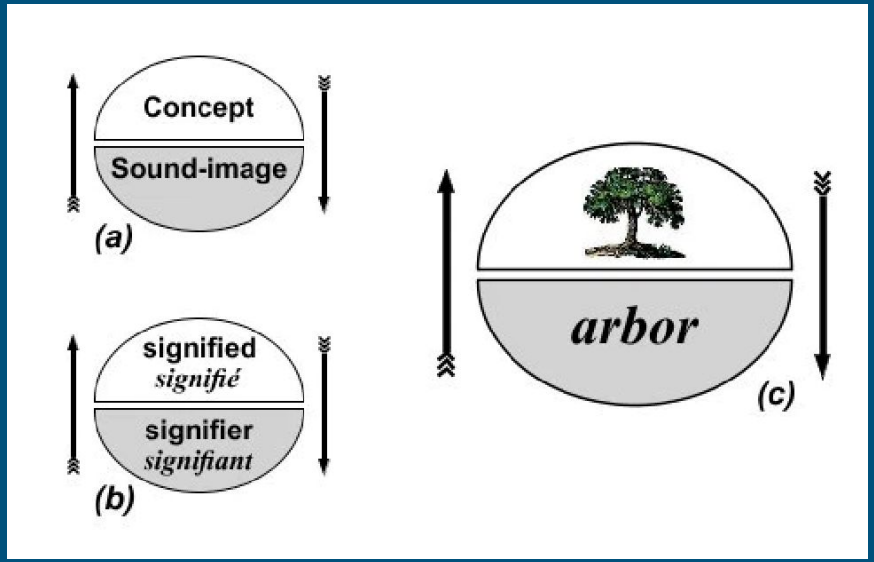
Ferdinand de Saussure (1857-1913), *Course in General Linguistics* (1916)

a linguistic “sign” is composed of “signifier” and “signified”

a cat

Ferdinand de Saussure (1857-1913), *Course in General Linguistics* (1916)

a linguistic “sign” is composed of “signifier” and “signified”

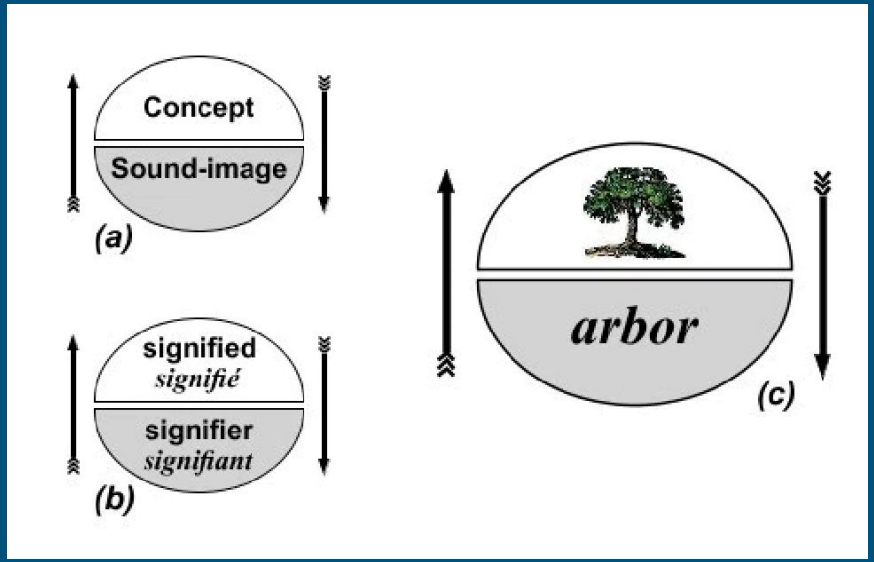


a cat

signifier

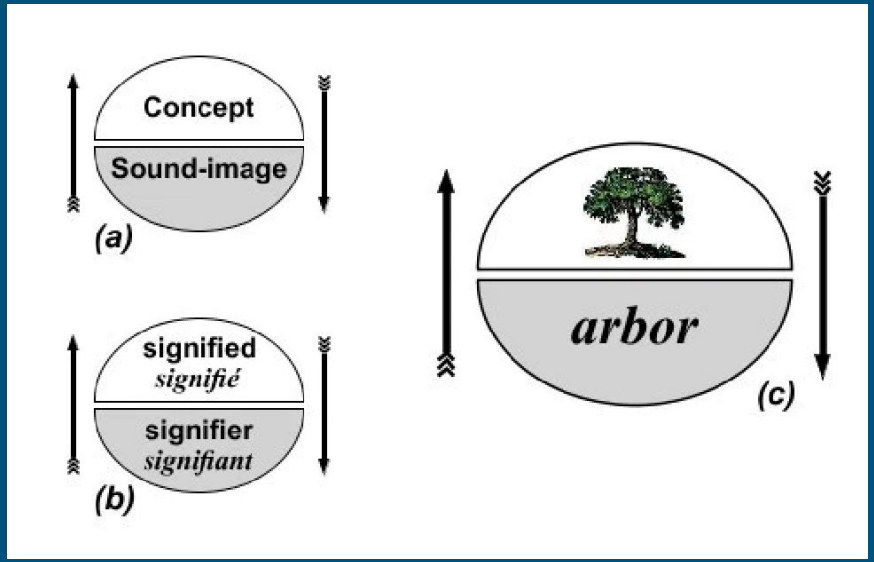
Ferdinand de Saussure (1857-1913), *Course in General Linguistics* (1916)

a linguistic “sign” is composed of “signifier” and “signified”



Ferdinand de Saussure (1857-1913), *Course in General Linguistics* (1916)

a linguistic “sign” is composed of “signifier” and “signified”



signified

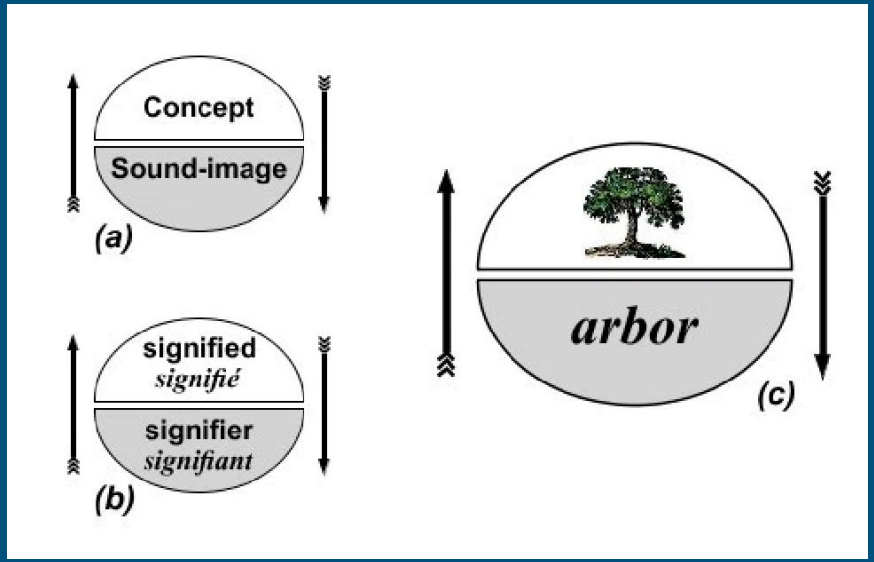
a cat

signifier

sign

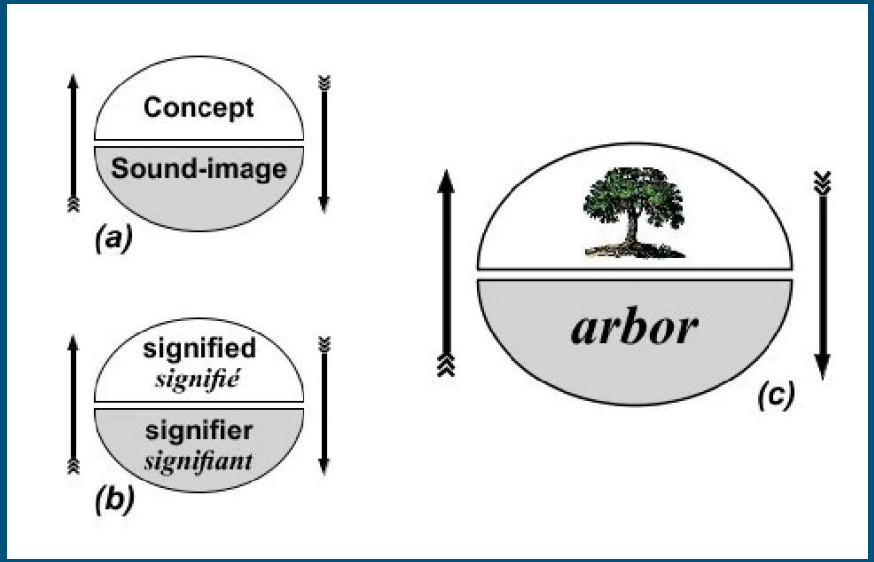
Ferdinand de Saussure (1857-1913), *Course in General Linguistics* (1916)

a linguistic “sign” is composed of “signifier” and “signified”



Ferdinand de Saussure (1857-1913), *Course in General Linguistics* (1916)

a linguistic “sign” is composed of “signifier” and “signified”



signified

a cat

signifier

sign

vs

(an actual real life cat out in the world, doing cat things)

(not a photo, not your idea of a cat, but *an actual real cat*)

signified

a cat

signifier

sign

vs

(an actual real life cat out in the world, doing cat things)

(not a photo, not your idea of a cat, but *an actual real cat*)

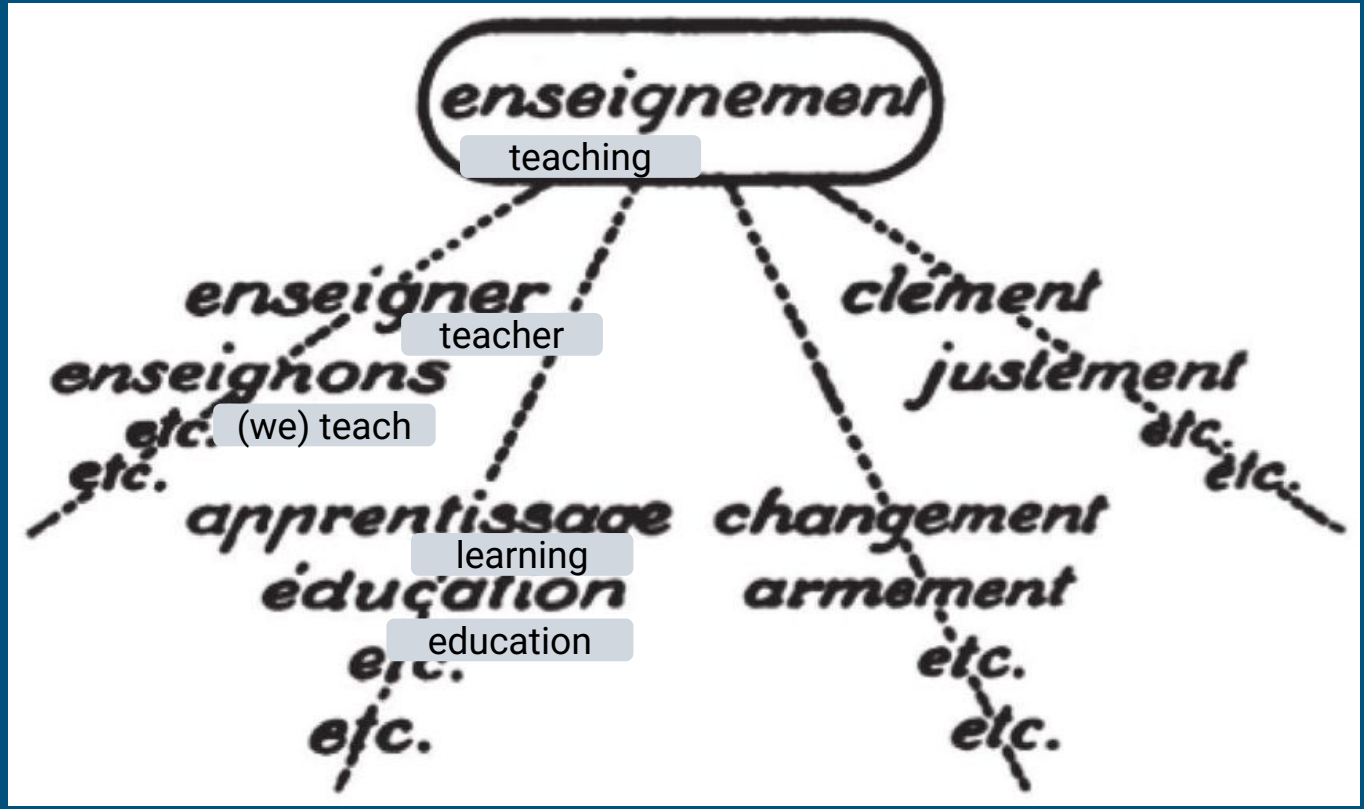
There's an *unbridgeable gap* between the sign and real thing.
CLIP's weirdness is a result of this gap.

Saussure: Linguistic signs are *arbitrary* & *defined by culture*

“a cat” *0.2542*

“a cheezburger” *0.2315*

Saussure: Linguistic signs are *structurally related to each other.*



Adapted from Ferdinand de Saussure, *Course in General Linguistics* (1959/1916)

CLIP gives us mathematical handles for signs

1. a dog
   similarity=0.1433
2. a cat
   similarity=0.2542
3. a monkey
   similarity=0.1334
4. a cheeseburger
   similarity=0.1253
5. a cheezburger
   similarity=0.2315
6. 3hlufakjsdnfa3fnaj;ld
   similarity=0.1821

1. a dog
   similarity=0.1433
2. *rity=0.2542*
3. nkey
   rity=0.1334
4. eseburger
   rity=0.1253
5. ezburger
   rity=0.2315
6. 3fiufakjsdnfa3fnaj;ld
   similarity=0.1821

Embedded Bias

Image = composite faces to produce “average” English man/woman


```
with torch.no
    image_e                                    eatures([
                                                oman.jpg'
    ])
                ings = get_text
            octor', 'a teacher',                engineer', 'a ml researcher', 'a nurse'

    print(image_embeddings @ text_embeddings.T)
```

✓ [25] 218ms

```
tensor([[0.1474, 0.1412, 0.1818, 0.2113, 0.1132],
        [0.1463, 0.1356, 0.1297, 0.1700, 0.1824]], device='mps:0')
```

We can do math with signifiers

Overview

1. What is CLIP?
2. How is CLIP trained?
3. Embeddings In Practise
4. A Very Brief Introduction to Semiotics
5. Manipulating Meaning with Math

Image Search Math

“an image of a man”

```
ith torch.no_grad():
  embeds = encode_prompt("an image of a man")

  view_search_results(embeds, N=9)
```

Image Search Math

“an image of a man”
- “japanese”

```
with torch.no_grad():
    embeds = encode_prompt("an image of a man")
    embeds -= encode_prompt("japanese")


    view_search_results(embeds, N=9)
```

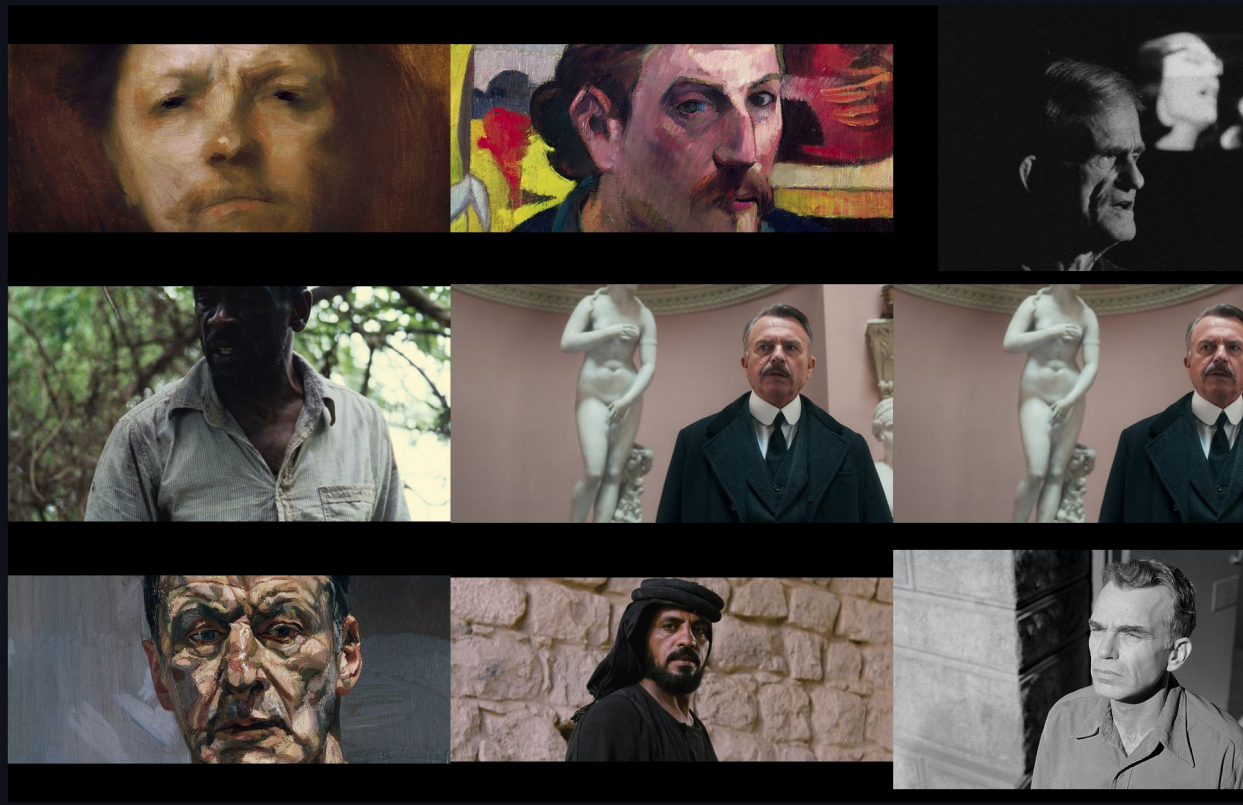


Image Search Math

“an image of a man”
- “japanese”
- “close up”

```
with torch.no_grad():
    embeds = encode_prompt("an image of a man")
    embeds -= encode_prompt("japanese")
    embeds -= encode_prompt("close up")


    view_search_results(embeds, N=9)
```

Image Search Math

“A person running in front of trees”

```
with torch.no_grad():
    embeds = encode_prompt("a person running in front of trees")
    view_search_results(embeds, N=12)
```

Image Search Math

“A person running in front of trees” - “winter”

```
with torch.no_grad():
    embeds = encode_prompt("a person running in front of trees")
    embeds -= encode_prompt("winter")
    view_search_results(embeds, N=12)
```

Image Search Math

“A person running in front of trees”
- “winter”*0.2

```
with torch.no_grad():
    embeds = encode_prompt("a person running in front of trees")
    embeds -= encode_prompt("winter") * 0.2
    view_search_results(embeds, N=12)
```

Image Search Math

“A person running in front of trees”
- “winter”*0.3

```
with torch.no_grad():
    embeds = encode_prompt("a person running in front of trees")
    embeds -= encode_prompt("winter") * 0.3
    view_search_results(embeds, N=12)
```

Image Search Math

“A person running in front of trees”
- “winter”*0.3
- “forest”

```
    embeds = encode_prompt("a person running in front of trees")
    embeds -= encode_prompt("winter") * 0.3
    embeds -= encode_prompt("forest")
    view_search_results(embeds, N=12)
```

Image Search Math

“A person running in front of trees”
- “winter”*0.3
- “forest”*0.5

```
with torch.no_grad():
    embeds = encode_prompt("a person running in front of trees")
    embeds -= encode_prompt("winter") * 0.3
    embeds -= encode_prompt("forest") * 0.5
    view_search_results(embeds, N=12)
```

We can do math with signifiers *and* signifieds

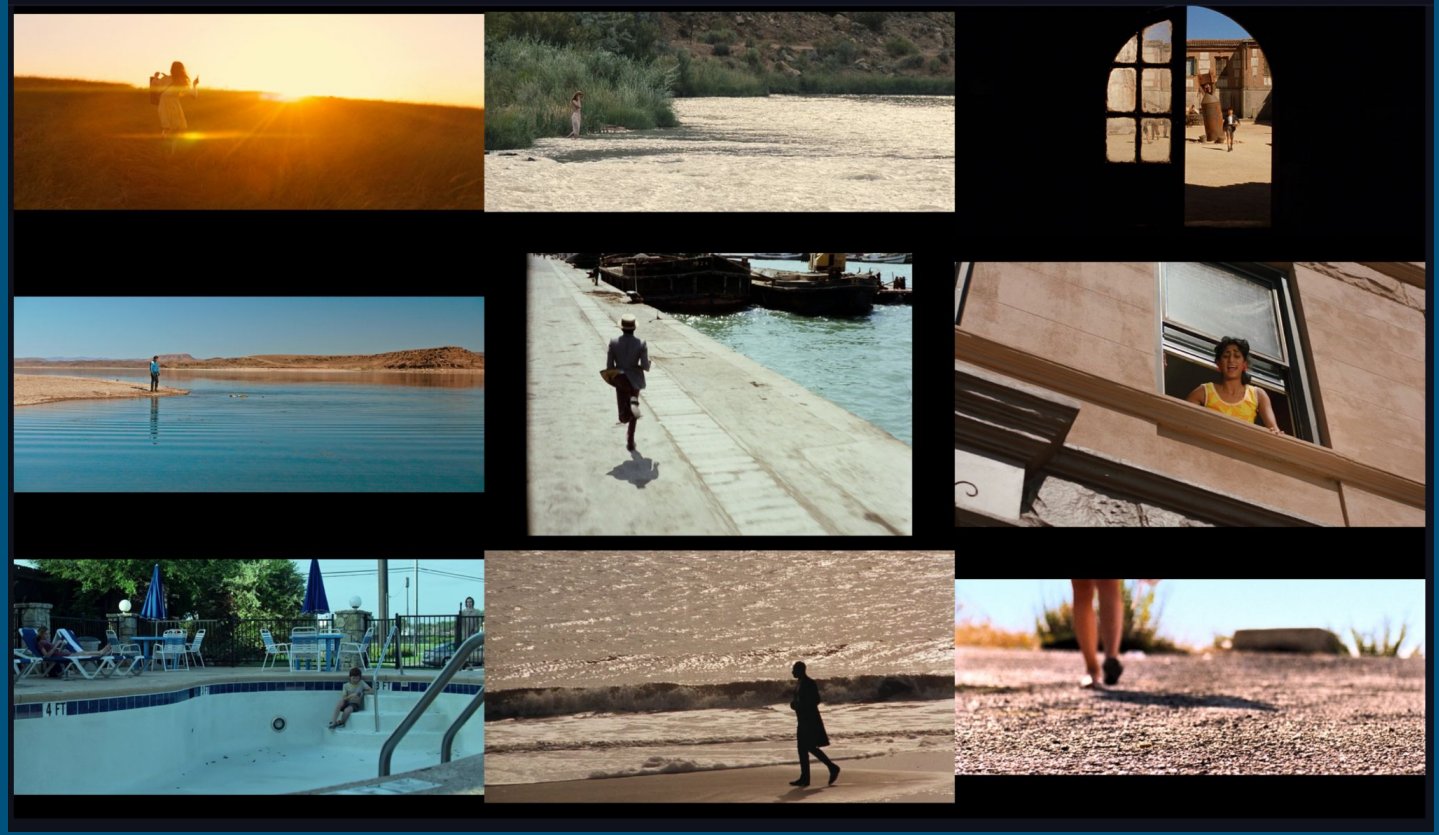
searching by image

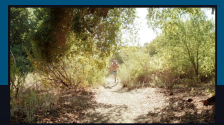
searching by image

```
with torch.no_grad():
    embeds = encode_image("/mnt/DataSSD/datasets/[redacted]Z.jpg")
    view_search_results(embeds, N=9)
```

searching by image  - “forest”

```
with torch.no_grad():
    embeds = encode_image("/mnt/DataSSD/datasets/                    .jpg")
    embeds -= encode_prompt("forest")
    view_search_results(embeds, N=9)
```

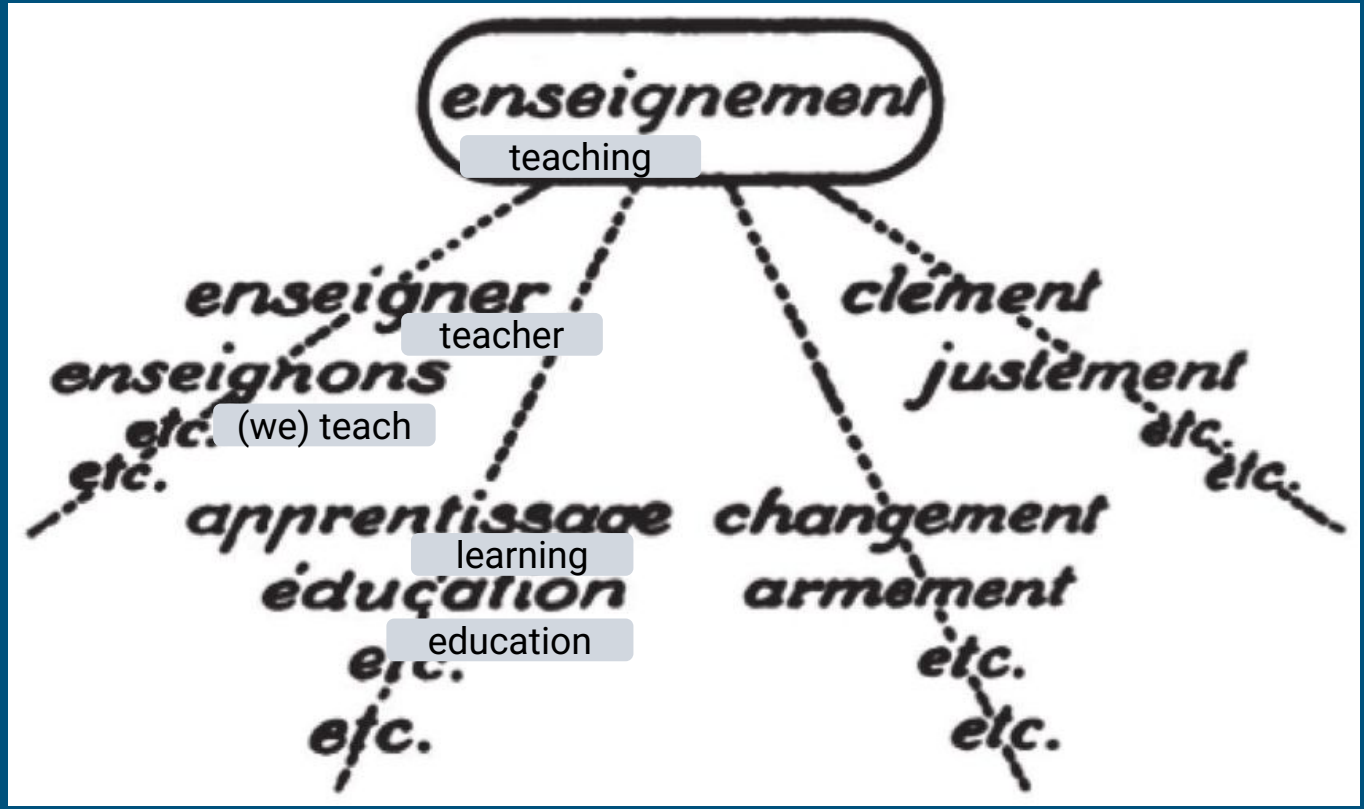


searching by image  - “forest”*0.5

```
with torch.no_grad():
    embeds = encode_image("/mnt/DataSSD/datasets/                    .jpg")
    embeds -= encode_prompt("forest") * 0.5
    view_search_results(embeds, N=9)
```

We can sort things by tracing relationship trees

Saussure: Linguistic signs are *structurally related to each other.*



Adapted from Ferdinand de Saussure, *Course in General Linguistics* (1959/1916)

(video)
ordering cinema frames by CLIP embedding
as a Travelling Salesperson Problem

Binary Opposition

Binary Opposition

Fundamental principle in post-structuralism, esp. Derrida & Deconstruction.

John Searle criticised it as obscurantist and lacking rigor.

What does CLIP think?

```
Man/woman
Colonizer/colonized
Logic/emotion
Mind/body
```

Search by CLIP embedding distance

```
display_nearest(get_text_features(["colonizer"])[0], n=3)
```

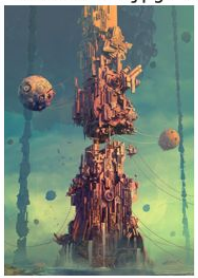
000006508.jpg: Warhammer 40,000: Inquisitor - Prophecy Announced - picture #1

000322147.jpg: Space Marine Fact

000329056.jpg: Warhammer 40,000: Inquisitor - Martyr - Prophecy

```
display_nearest(get_text_features(["colonized"])[0], n=3)
```

000028676.jpg: Bioshock-infinite-exceptionalism-650x360_medium



000321033.jpg: unknownskywalker: The tower by Ruidan

000019319.jpg: The Beautiful Concept Art Of BioShock Infinite

Dataset: LAION Aesthetic 600m (subset)

Search by CLIP embedding distance + *order by decreasing similarity delta*

colonizer

000421961.jpg: Lightforged Uther.jpg None

000033105.jpg: Space Marines Warhammer 40k, wha art, фан-арт, космодесант None

000103443.jpg: Test Romance of The Three Kingdoms XIII None

colonized

000066017.jpg: Photograph - Homestead Palouse by Dale Stillman None

000117800.jpg: Abandoned Buildings by Matthias Haker None

000124181.jpg: ABANDONED HOTEL by Chase Alog - Buildings & Architecture Decaying & Abandoned None

Dataset: LAION Aesthetic 600m (subset)

Search by CLIP embedding distance + *order by decreasing similarity delta*

```
cls_texts = ["colonizer", "colonized"]
cls_text_features = get_text_features(cls_texts)
cls_results = all_features @ cls_text_features.T

for i, t in enumerate(cls_texts):
    print(t)
    this_class_results = cls_results[:, i]
    other_class_results = cls_results[:, 1-i]
    distance_order_desc = torch.sort(this_class_results - other_class_results, descending=True).indices
    display_images(get_images_in_order(i, distance_order_desc, n=3))

✓ [44] 227ms
```

Operating on raw activation outputs
rather than probabilities … ?

Search by CLIP embedding distance

```
continue

display_nearest(get_text_features(["happy"])[0], n=3)
```

000118728.jpg: gumi from vocaloid gumi megpoid vocaloids photo 36077426 fan

000069895.jpg: Rating: Safe Score: 72 Tags: love_live! nishikino_maki yana_

000107712.jpg: Rating: Safe Score: 39 Tags: animal blush cherry_blossoms fa

```
display_nearest(get_text_features(["sad"])[0], n=3)
```

000004848.jpg: "Gordon Parks ""The Invisible Man, Harlem, New York,"" 195

000124637.jpg: Hyper realistic painting by Jesse Lane

000110099.jpg: GORDON PARKS, Invisible Man , 1952

Search by CLIP embedding distance + *order by decreasing similarity delta*

happy

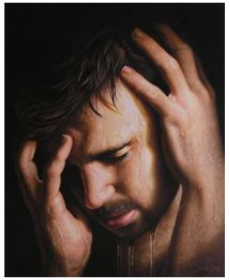
000522908.jpg: фотография Hikers with backpacks jumping with arms up on top of a mountain - Couple of youn

000092750.jpg: Beautiful girl jumping on the rural road Stock Photo

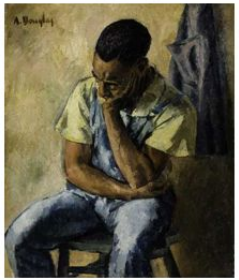
000106261.jpg: Confetti by Polya Ilchenko on 500px.com

sad

000104280.jpg: willem_dafoe_wide_NEW



000124637.jpg: Hyper realistic painting by Jesse Lane



000527209.jpg: AARON-DOUGLAS-(1899---1979)-Untitled-(Seated-Man-With-Head-R

```python
with torch.no_grad():
    image_embeddings = get_image_features([
        'generic-man.jpg', 'generic-woman.jpg'
    ])
    text_embeddings = get_text_features([
        'a doctor', 'a teacher', 'a software engineer', 'a ml researcher', 'a nurse'
    ])
    print(image_embeddings @ text_embeddings.T)
```

✓ [28] 337ms

```
tensor([[0.1474, 0.1412, 0.1818, 0.2113, 0.1132],
        [0.1463, 0.1356, 0.1297, 0.1700, 0.1824]], device='mps:0')
```

```python
with torch.no_grad():
    image_embeddings = get_image_features([
        'generic-man.jpg', 'generic-woman.jpg'
    ])
    text_embeddings = get_text_features([
        'a male doctor', 'a male teacher', 'a male software engineer', 'a male ml researcher', 'a male nurse'
    ])
    print(image_embeddings @ text_embeddings.T)
```

✓ [29] 95ms

```
tensor([[0.1642, 0.1655, 0.1903, 0.2237, 0.1766],
        [0.1001, 0.0815, 0.0841, 0.1191, 0.1054]], device='mps:0')
```

```python
with torch.no_grad():
    image_embeddings = get_image_features([
        'generic-man.jpg', 'generic-woman.jpg'
    ])
    text_embeddings = get_text_features([
        'a female doctor', 'a female teacher', 'a female software engineer', 'a female ml researcher', 'a female nurse'
    ])
    print(image_embeddings @ text_embeddings.T)
```

✓ [30] 78ms

```
tensor([[0.0783, 0.0814, 0.0944, 0.1347, 0.0826],
        [0.1813, 0.1493, 0.1595, 0.1887, 0.1809]], device='mps:0')
```

Implications & Final Thoughts

Next Steps

Industry

CLIP clearly has numerous user-facing capabilities that remain untapped.

How do we build UX for embedding math?
How do we build UX for binary opposition pairs?
(Does this work for other embedding models?)

Research

What else might poststructuralism and web-scale ML have to teach other other?

To what extent does CLIP's behaviour support key tenets of poststructuralist theory?

If any of this sounds like something you'd like to explore - let's talk! 👉 d@damianstewart.com 👈

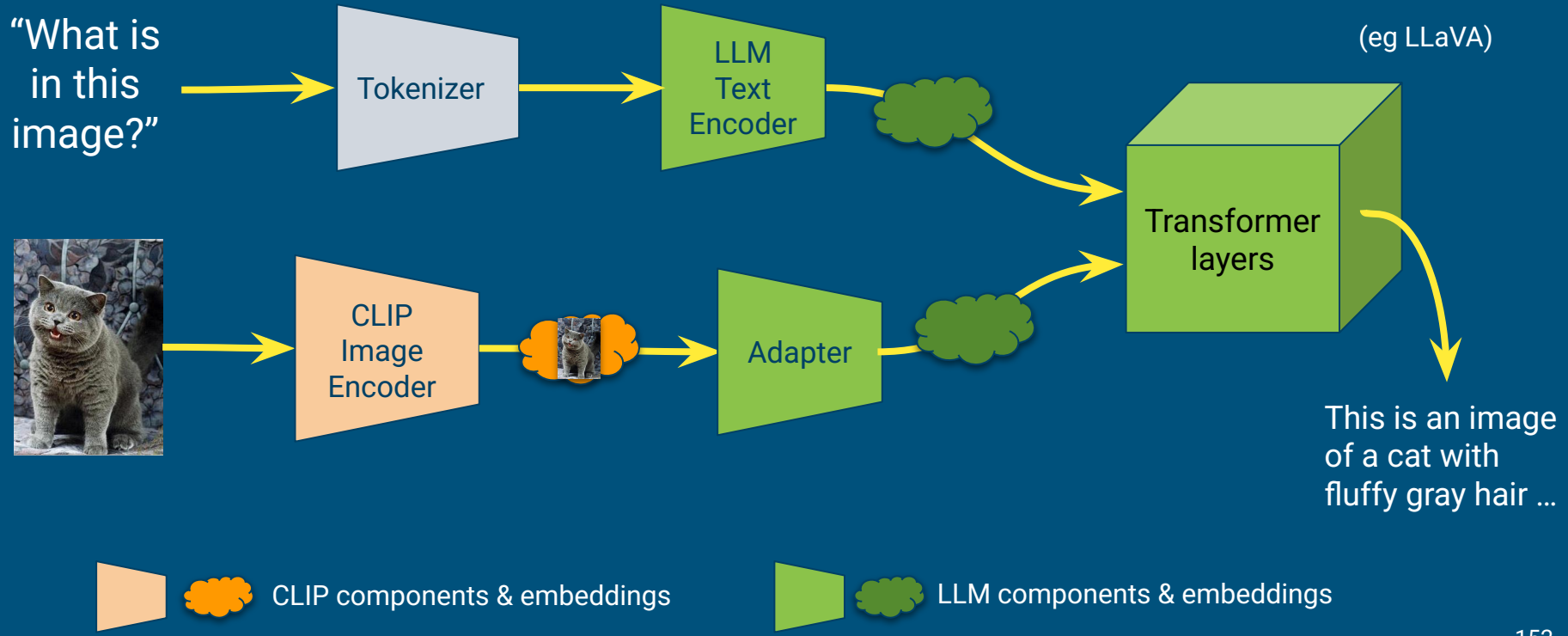
Use Case: Multimodal LLM

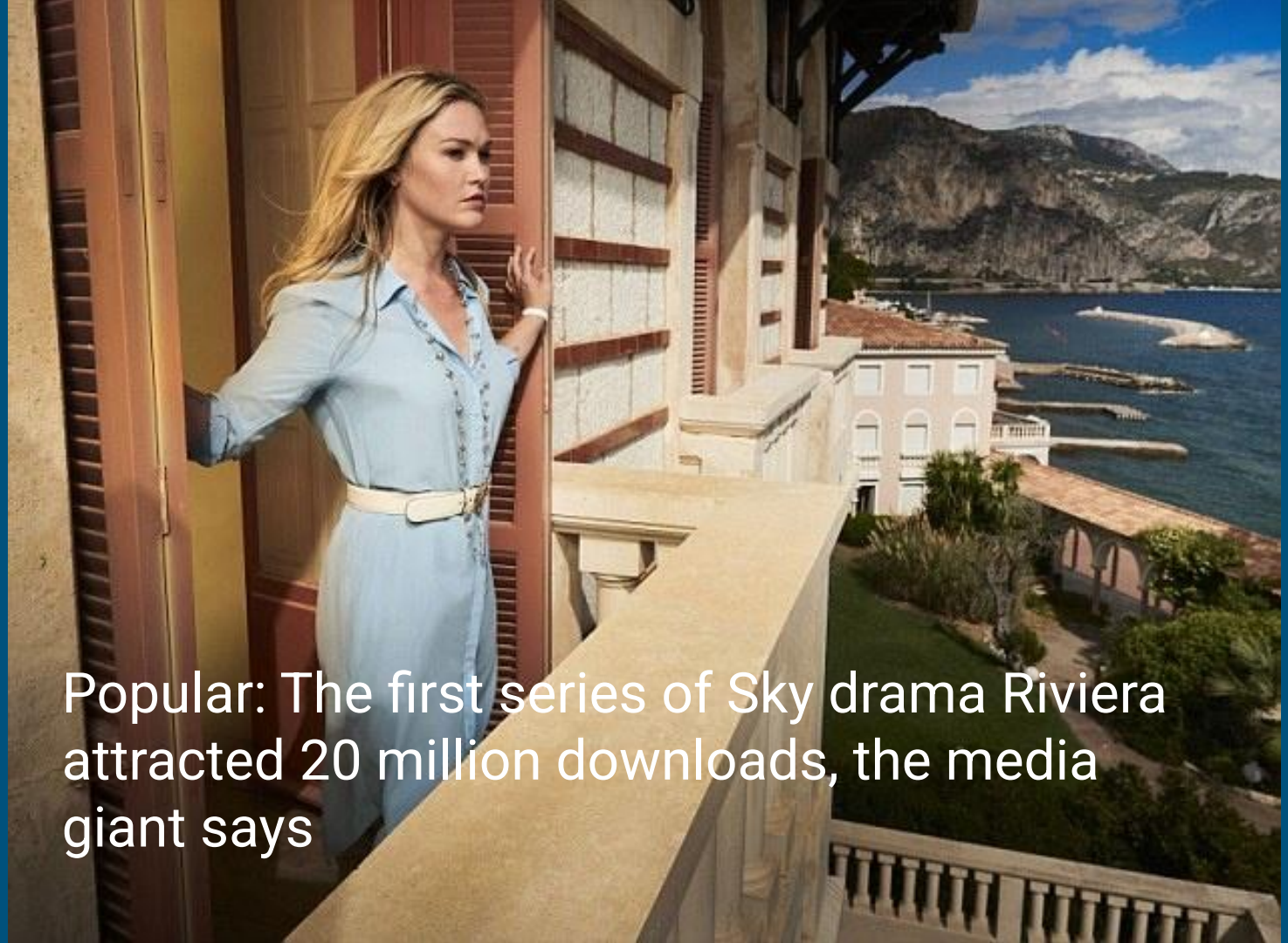
Multimodal LLM

<user>: What is in this image?

<llava>: This is an image of a cat with fluffy gray hair …

Multimodal LLM




Popular: The first series of Sky drama Riviera attracted 20 million downloads, the media giant says


LARGE CASTLE CONCEPT
CROWFALL
www.CROWFALL.com

Crowfall: July 2015 Founder's Update news header


131701 download wallpaper Animals, Wolf, Snow, To Lie Down, Lie, Cold, Forest, Trees screensavers and pictures for free