

CLIP: From Fixed Labels to Language-Defined Vision

Following our Vision Transformer lecture

How Images and Text Learn a Shared Comparison Space

Last time: an image became a useful ViT representation.

Today: let language say what we want to find.



Cat

"a photo of a cat"



Dog

"a photo of a dog"

| An image goes in. We choose the descriptions to compare it with.

SECTION 01

From fixed labels to language

Why must ViT's answer vocabulary be fixed?

Last lecture: one image, 1,000 stored answers

SCHEMATIC · our earlier ViT-Tiny classifier



Newfoundland



| The image representation is useful. The classifier chooses from its training labels.

Can we choose the labels ourselves?

APPLICATION 01 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Newfoundland

Oxford-IIIT Pet · CC BY-SA 4.0

Descriptions we supply

a photo of a dog ?

a photo of a cat ?

a photo of a bird ?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Which supplied description will match the dog photograph best?

What happened? · New labels

APPLICATION 01 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Newfoundland

Oxford-IIIT Pet · CC BY-SA 4.0

Descriptions we supply · cosine

a photo of a dog	+0.2723
a photo of a cat	+0.2152
a photo of a bird	+0.2184

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

| Dog has the largest cosine. Every score comes from the same image vector.

Coffee, tea, juice or soup?

APPLICATION 02 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Coffee

Rachel Michetti · CC0

Descriptions we supply

a cup of coffee	?
a cup of tea	?
a glass of orange juice	?
a bowl of soup	?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Can we ask what is inside the cup, rather than just name the cup?

What happened? · Attributes

APPLICATION 02 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Coffee

Rachel Michetti · CC0

Descriptions we supply · cosine

a cup of coffee	+0.2820
a cup of tea	+0.2522
a glass of orange juice	+0.2398
a bowl of soup	+0.2258

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

| Coffee scores above tea. The output is a ranking of the descriptions we supplied.

Find “something to drink”

APPLICATION 03 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing

Our text query

“something to drink”

Compare it with gallery images.



?

Red mug · generated



?

Blue mug · generated



?

Coffee



?

Rocket launch

Four pictured candidates from the 22-image gallery; the saved file contains every score.

Text vector → compare with image vectors → rank the gallery.

| What will a search for ‘something to drink’ return from this gallery?

What happened? · Text → image

APPLICATION 03 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities

Our text query

“something to drink”

Compare it with gallery images.



0.2559

Red mug · generated



0.2512

Blue mug · generated



0.2413

Coffee



0.1614

Rocket launch

Four pictured candidates from the 22-image gallery; the saved file contains every score.

Text vector → compare with image vectors → rank the gallery.

| The generated mugs score above the coffee photograph for this query.

Which supplied sentence describes this scene?

APPLICATION 04 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Rocket launch

SpaceX · CC0

Descriptions we supply

a rocket launching into the night sky ?

a tall building beside a river ?

fireworks above a city ?

an airplane on a runway ?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Which of these four complete sentences best matches the rocket photograph?

What happened? · Image → text

APPLICATION 04 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Rocket launch

SpaceX · CC0

Descriptions we supply · cosine

a rocket launching into the night sky	+0.2635
a tall building beside a river	+0.2041
fireworks above a city	+0.2085
an airplane on a runway	+0.1966

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

The rocket sentence wins. We supplied all four sentences; CLIP generated none of them.

Find another image like this one

APPLICATION 05 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Persian cat

Query image · exclude the image itself



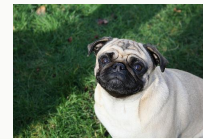
?

Chelsea the cat



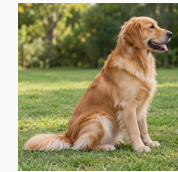
?

Newfoundland



?

Pug



?

Dog photograph style · generated

These are the top four retrieved images, shown in result order. Inspect their similarities.

Image vector → compare with other image vectors. The text encoder is not used.

| Which other gallery image will be nearest to the Persian cat photograph?

What happened? · Image → image

APPLICATION 05 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



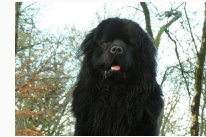
Persian cat

Query image · exclude the image itself



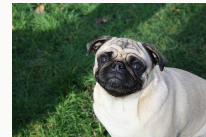
0.8100

Chelsea the cat



0.7299

Newfoundland



0.7113

Pug



0.6693

Dog photograph style ·
generated

These are the top four retrieved images, shown in result order. Inspect their similarities.

Image vector → compare with other image vectors. The text encoder is not used.

Chelsea is the nearest other image. This comparison uses only the image encoder.

What if we add a hat?

APPLICATION 06 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · predict before revealing



Before

After

A straw hat is added to the portrait.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors → $\text{normalize}(\text{after} - \text{before})$ → compare with unit text vectors.

Which candidate word best describes the change between the two pictures?

Descriptions we supply

hat	?
cup	?
cat	?
boat	?

What happened? · Hat change

APPLICATION 06 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · raw cosines, not probabilities



Before

After

A straw hat is added to the portrait.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors → $\text{normalize}(\text{after} - \text{before})$ → compare with unit text vectors.

Hat has the largest change cosine for this pair and these four words.

Descriptions we supply · cosine

hat	+0.0808
cup	+0.0132
cat	+0.0026
boat	+0.0254

What if the mug changes colour?

APPLICATION 07 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · predict before revealing



Before

After

The mug’s surface color changes from red to blue.

Two generated inputs. CLIP does not edit either picture.

Descriptions we supply

blue	?
red	?
cup	?
hat	?

Unit image vectors → normalize(after – before) → compare with unit text vectors.

Which word matches the change from the red mug to the blue mug?

What happened? · Color change

APPLICATION 07 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · raw cosines, not probabilities



Before

After

The mug’s surface color changes from red to blue.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors → $\text{normalize}(\text{after} - \text{before})$ → compare with unit text vectors.

Blue is positive, red is negative, and cup is near zero: the cup is shared by both images.

Descriptions we supply · cosine

blue	+0.1121
red	-0.1778
cup	-0.0120
hat	-0.0117

Can words help us find a brick kiln?

APPLICATION 08 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Brick kiln · generated

AI-generated teaching image · OpenAI image tool · 29 Sep 2026

Descriptions we supply

an aerial image of a brick kiln ?

an aerial image of a solar farm ?

an aerial image of agricultural fields ?

an aerial image of industrial warehouses ?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Which supplied scene description matches the generated kiln tile?

What happened? · Sustainability

APPLICATION 08 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Brick kiln · generated

AI-generated teaching image · OpenAI image tool · 29 Sep 2026

Descriptions we supply · cosine

an aerial image of a brick kiln **+0.3135**

an aerial image of a solar farm **+0.2882**

an aerial image of agricultural fields **+0.2903**

an aerial image of industrial warehouses **+0.2902**

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Kiln wins this generated scene. This is a demonstration, not a satellite benchmark.

What if we add a chimney?

APPLICATION 09 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · predict before revealing



Before

After

A tall brick chimney is added to a kiln scene.
Two generated inputs. CLIP does not edit either picture.

Unit image vectors → normalize(after – before) → compare with unit text vectors.

Which word matches the change between the two kiln scenes?

Descriptions we supply	
chimney	?
tree	?
truck	?
solar panel	?

What happened? · Chimney change

APPLICATION 09 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · raw cosines, not probabilities



Before



After

A tall brick chimney is added to a kiln scene.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors $\rightarrow$ $\text{normalize}(\text{after} - \text{before}) \rightarrow$ compare with unit text vectors.

Descriptions we supply · cosine

chimney	+0.0336
tree	+0.0160
truck	+0.0057
solar panel	+0.0048

Chimney wins this synthetic pair. Shadows and scene details can also affect the difference.

Does removing the stripes change the match?

MEASURED ORIGINAL CLIP · ViT-B/32 · CPU float32 · generated before/after images · raw cosines



Before: stripes



After: generated edit

Supplied description	Before	After
a photo of a zebra	0.3358	?
a photo of a horse	0.2689	?
a photo of a donkey	0.2638	?
a photo of an okapi	0.2646	?

Classify each image separately. Same model, same descriptions; no vector subtraction here.

The first image matches “zebra”. Which description will win after the edit?

Does removing the stripes change the match?

MEASURED ORIGINAL CLIP · ViT-B/32 · CPU float32 · generated before/after images · raw cosines



Before: stripes



After: generated edit

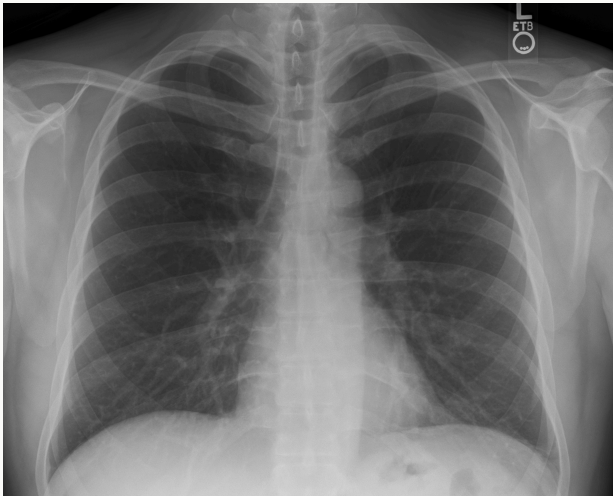
Supplied description	Before	After
a photo of a zebra	0.3358	0.2233
a photo of a horse	0.2689	0.2980
a photo of a donkey	0.2638	0.2718
a photo of an okapi	0.2646	0.1824

Classify each image separately. Same model, same descriptions; no vector subtraction here.

Zebra drops from 0.3358 to 0.2233; horse becomes the best match. The edit changes the prediction.

What kind of medical image is this?

APPLICATION 10 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Chest radiograph

Stillwaterising · Wikimedia Commons · CC0

Descriptions we supply

a chest X-ray ?

an X-ray of a hand ?

a brain MRI scan ?

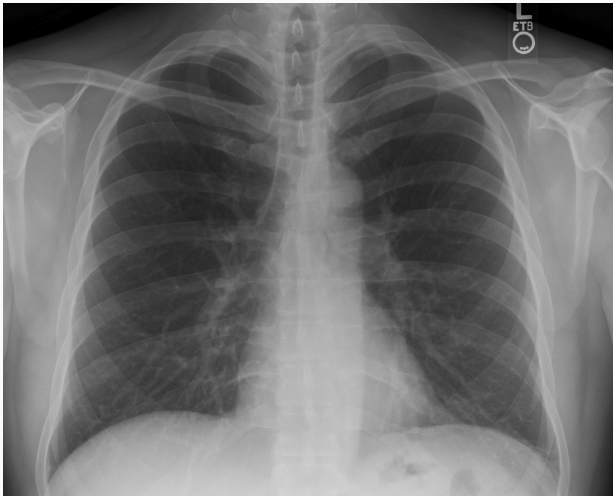
a photograph of a person ?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Can CLIP match this public radiograph to an imaging modality?

What happened? · Healthcare

APPLICATION 10 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Chest radiograph

Stillwaterising · Wikimedia Commons · CC0

Descriptions we supply · cosine

a chest X-ray **+0.2673**

an X-ray of a hand **+0.2295**

a brain MRI scan **+0.2047**

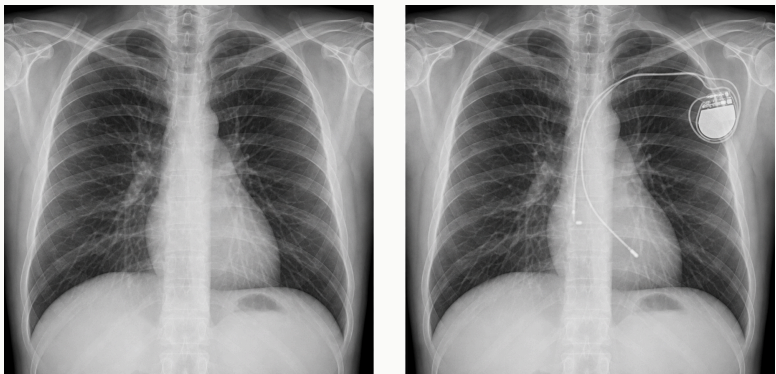
a photograph of a person **+0.2304**

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Chest X-ray wins among these four descriptions. We have not tested a diagnosis.

Will an added device give a useful change vector?

APPLICATION 11 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · predict before revealing



Before

After

A pacemaker-like device and leads are added to a synthetic radiograph-style image.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors → $\text{normalize}(\text{after} - \text{before})$ → compare with unit text vectors.

Will the added device make ‘pacemaker’ the highest-scoring word?

Descriptions we supply

pacemaker

?

rib

?

stethoscope

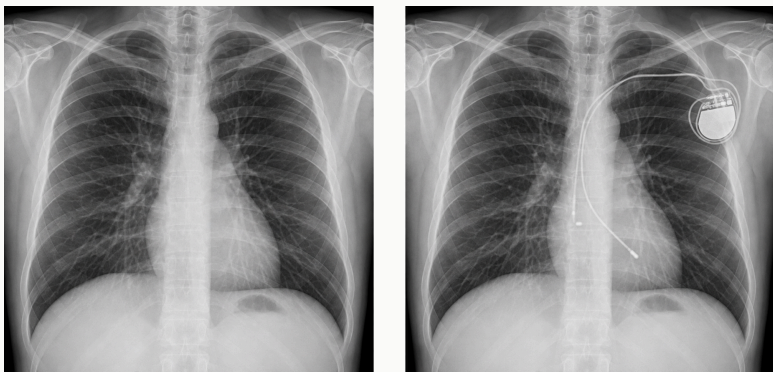
?

hat

?

What happened? · Device change

APPLICATION 11 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · raw cosines, not probabilities



Before

After

A pacemaker-like device and leads are added to a synthetic radiograph-style image.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors $\rightarrow$ $\text{normalize}(\text{after} - \text{before}) \rightarrow$ compare with unit text vectors.

Descriptions we supply · cosine

pacemaker **+0.0127**

rib **-0.0133**

stethoscope **+0.0318**

hat **+0.0372**

| Hat wins; pacemaker does not. The intended edit does not guarantee a useful direction.

Can a sketch still match “dog”?

APPLICATION 12 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Dog sketch · generated

AI-generated teaching image · OpenAI image tool · 29 Sep 2026

Descriptions we supply

a drawing of a dog	?
a drawing of a cat	?
a drawing of a horse	?
a drawing of a fox	?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Can the dog concept still match when the image is a pencil sketch?

What happened? · Photo → sketch

APPLICATION 12 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Dog sketch · generated

AI-generated teaching image · OpenAI image tool · 29 Sep 2026

Descriptions we supply · cosine

a drawing of a dog	+0.3176
--------------------	---------

a drawing of a cat	+0.2601
--------------------	---------

a drawing of a horse	+0.2757
----------------------	---------

a drawing of a fox	+0.2639
--------------------	---------

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Dog wins for the sketch too. The subject can remain recognizable across styles.

What changes when a photo becomes a sketch?

APPLICATION 13 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · predict before revealing



Before



After

A photograph-style dog scene is rendered as a pencil sketch.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors → $\text{normalize}(\text{after} - \text{before})$ → compare with unit text vectors.

| Does the change direction emphasize pencil sketch, photograph, dog, or cat?

Descriptions we supply

pencil sketch	?
photograph	?
dog	?
cat	?

What happened? · Style change

APPLICATION 13 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · generated image pair · raw cosines, not probabilities



Before



After

A photograph-style dog scene is rendered as a pencil sketch.

Two generated inputs. CLIP does not edit either picture.

Unit image vectors → $\text{normalize}(\text{after} - \text{before})$ → compare with unit text vectors.

Pencil sketch wins; dog is near zero because the subject is shared by both pictures.

Descriptions we supply · cosine

pencil sketch **+0.1468**

photograph **−0.0105**

dog **−0.0078**

cat **−0.0022**

What if the right answer is missing?

APPLICATION 14 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Newfoundland

Oxford-IIIT Pet · CC BY-SA 4.0

Descriptions we supply

a photo of a cat ?

a photo of a bird ?

a photo of a car ?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

What will win when none of the descriptions says 'dog'?

What happened? · Candidate trap

APPLICATION 14 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Newfoundland

Oxford-IIIT Pet · CC BY-SA 4.0

Descriptions we supply · cosine

a photo of a cat	+0.2152
a photo of a bird	+0.2184
a photo of a car	+0.2179

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

Bird wins a menu with no dog option. The highest score can still select a wrong description.

Does “without” change what the model sees?

APPLICATION 15 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · predict before revealing



Persian cat

Oxford-IIIT Pet · CC BY-SA 4.0

Descriptions we supply	
a photo of a cat	?
a photo <u>without</u> a cat	?
a photo of a dog	?

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

| Will ‘a photo without a cat’ score much lower than ‘a photo of a cat’?

What happened? · Language limits

APPLICATION 15 / 15 · frozen pretrained CLIP ViT-B/32 · saved Node CPU q8 measurements · raw cosines, not probabilities



Persian cat

Oxford-IIIT Pet · CC BY-SA 4.0

Descriptions we supply · cosine

a photo of a cat	+0.2737
a photo <u>without</u> a cat	+0.2699
a photo of a dog	+0.2236

Image vector + one vector per supplied description → cosine scores. CLIP writes no answer text.

“A cat” and “without a cat” differ by only 0.0038 here. Wording alone does not ensure logic.

What made all these questions possible?

APPLICATION TOUR · from observed behaviour to the model that makes it possible



Newfoundland



Rocket launch



Brick kiln · generated



Chest radiograph



Dog sketch · generated

The model stayed fixed. We changed the pictures, the words or the comparison.

| Now let's follow those inputs through CLIP, then work out how it learns to match them.

SECTION 02

Two encoders, one comparison

How can a photograph and a sentence become comparable?

The classifier has 1,000 stored class vectors

EARLIER ViT-TINY · 192 image features · actual stored checkpoint weights, rounded for display



Stored output layer W

1,000 rows × 192 weights

Fixed category	Its stored weights · first 3 of 192	Bias
Newfoundland dog class	$\mathbf{w}_{\text{dog}} = [+0.0068, +0.0195, -0.0066, \dots]$	+0.0017
Persian cat cat class	$\mathbf{w}_{\text{cat}} = [+0.0218, +0.0067, -0.0400, \dots]$	-0.0024
macaw bird class	$\mathbf{w}_{\text{bird}} = [-0.0683, -0.0459, -0.0311, \dots]$	+0.0026
⋮ 997 more class rows, each with its own 192 weights and bias		

Each row: dot with the same $\mathbf{h}$, then add its bias.

$$\mathbf{s} = \mathbf{W}\mathbf{h} + \mathbf{b} \longrightarrow 1,000 \text{ class scores}$$

A new photo changes $\mathbf{h}$. The 1,000 class vectors stay fixed when we predict.

Let the words supply the class vector

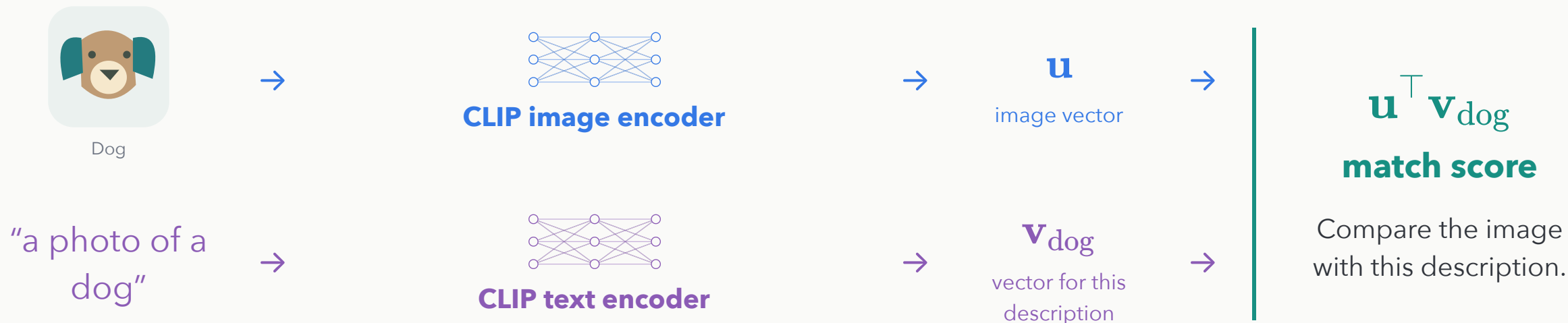
BRIDGE TO CLIP · a stored class row $\rightarrow$ a vector computed from our description

Previous slide: 1,000 fixed categories

Each category has a stored vector, such as $\mathbf{w}_{\text{dog}}$.

$$\mathbf{h}^T \mathbf{w}_{\text{dog}} + b_{\text{dog}}$$

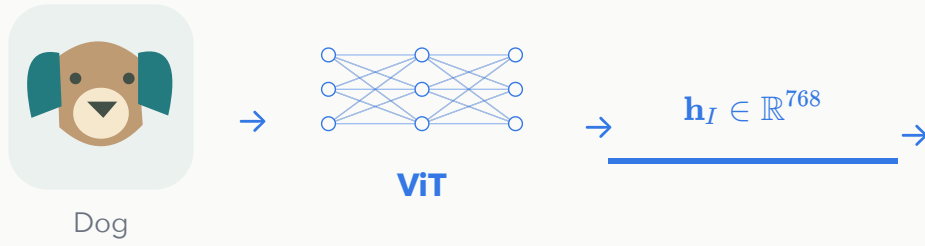
one class score



Change the description, and CLIP computes a new class vector. The model weights stay fixed.

First turn the image into one vector

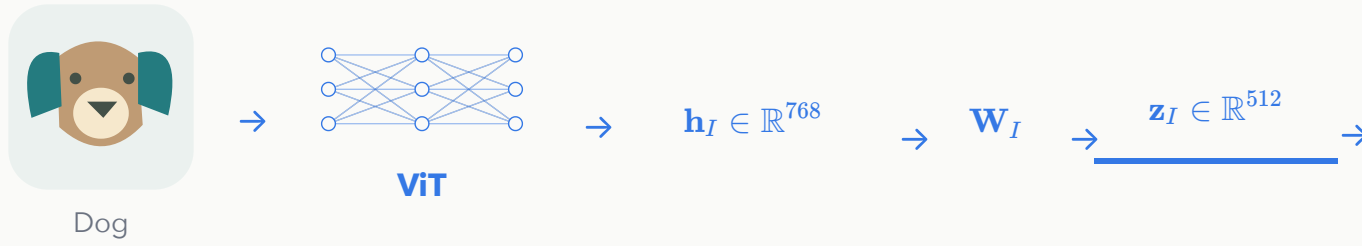
SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



Use the final CLS state after the ViT blocks and LayerNorm.

Project the image summary

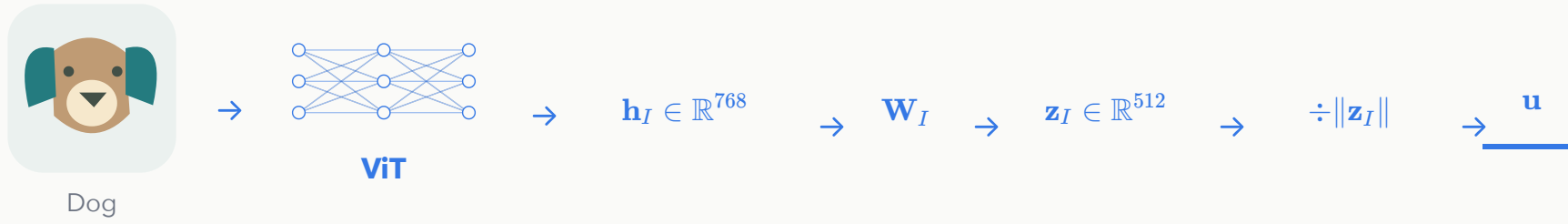
SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



| A learned 768×512 matrix maps the image summary into the comparison space.

Normalize it

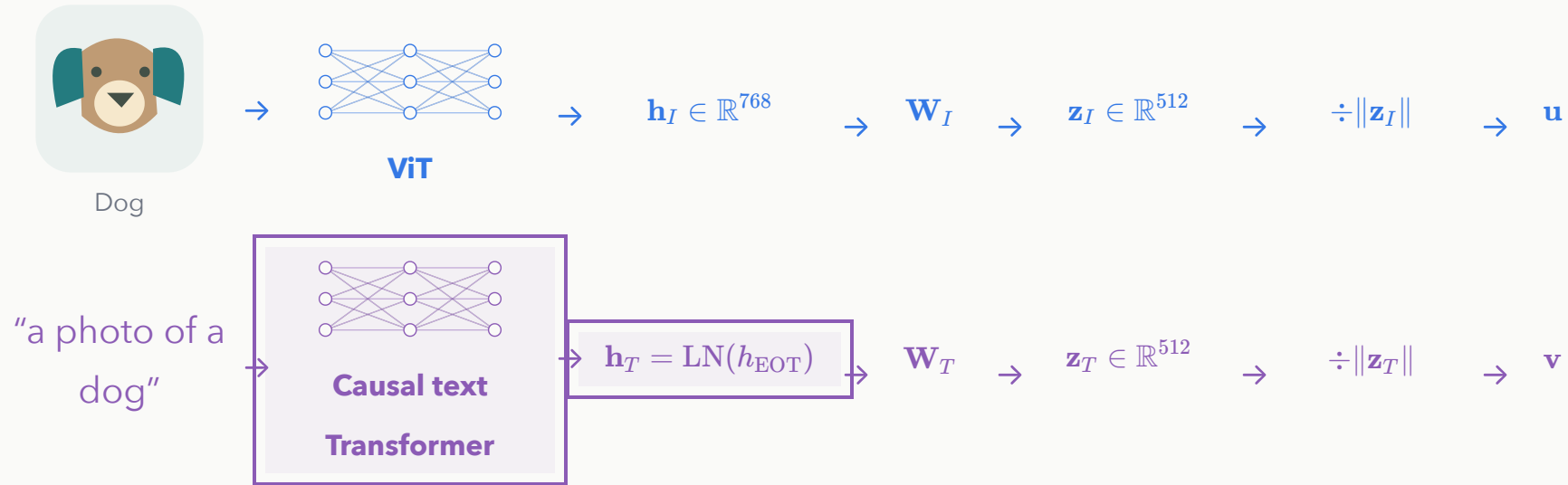
SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



Divide by the vector's length. The resulting image vector has length 1.

The complete image-text comparison

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



$$\mathbf{u}^T \mathbf{v}$$

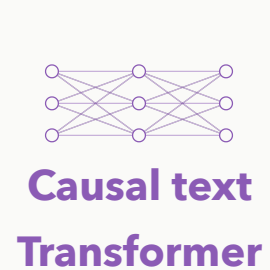
cosine
similarity

Next, zoom into the highlighted text Transformer and its EOT output.

Zoom in: EOT is our sentence readout

ORIGINAL CLIP · causal attention in the text branch; full attention in the image ViT

Recall Beyond Attention We selected the updated **CLS** state to represent a sentence.



Each token reads itself and earlier tokens.
EOT can read the whole description.

↓
Readout: select the final EOT state

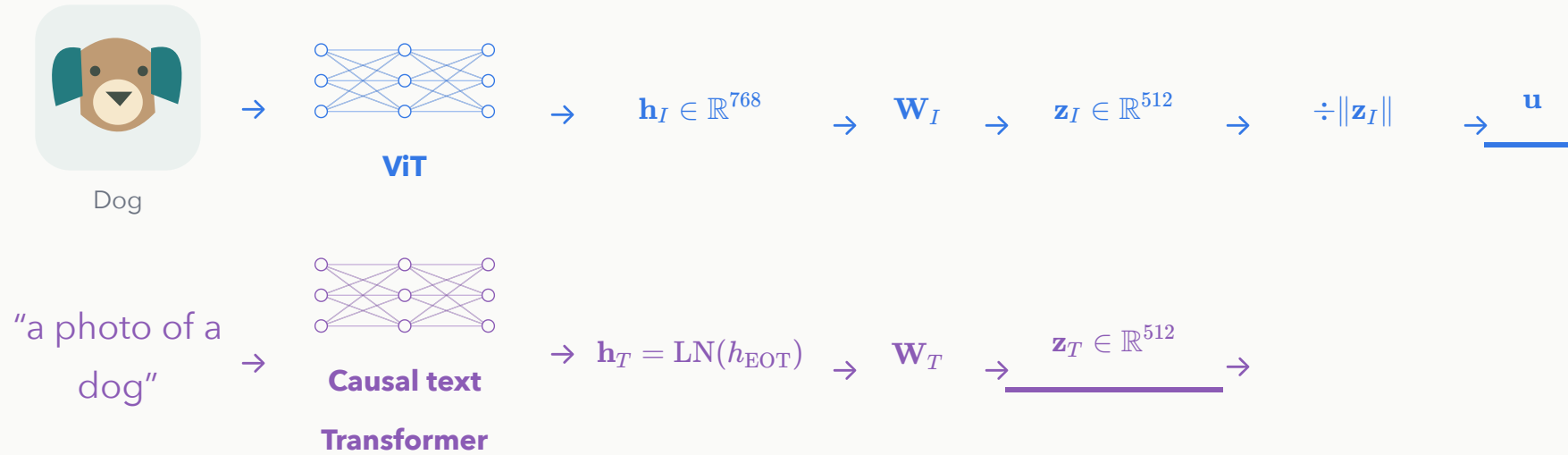
$$\mathbf{h}_T = \text{LN}(h_{\text{EOT}}) \in \mathbb{R}^{512}$$

LayerNorm (LN) → 512 numbers for this description. Projection comes next.

The same readout idea: use one updated token state as the representation of the whole description.

Project the text summary

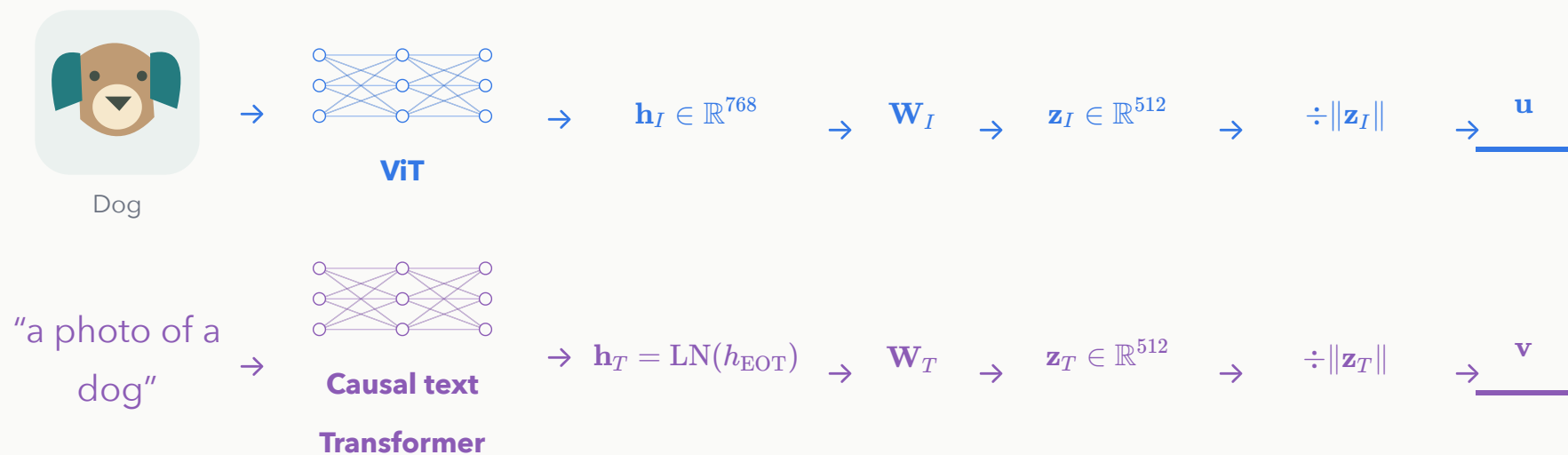
SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



| A learned 512×512 projection can change direction without changing width.

Normalize the text vector too

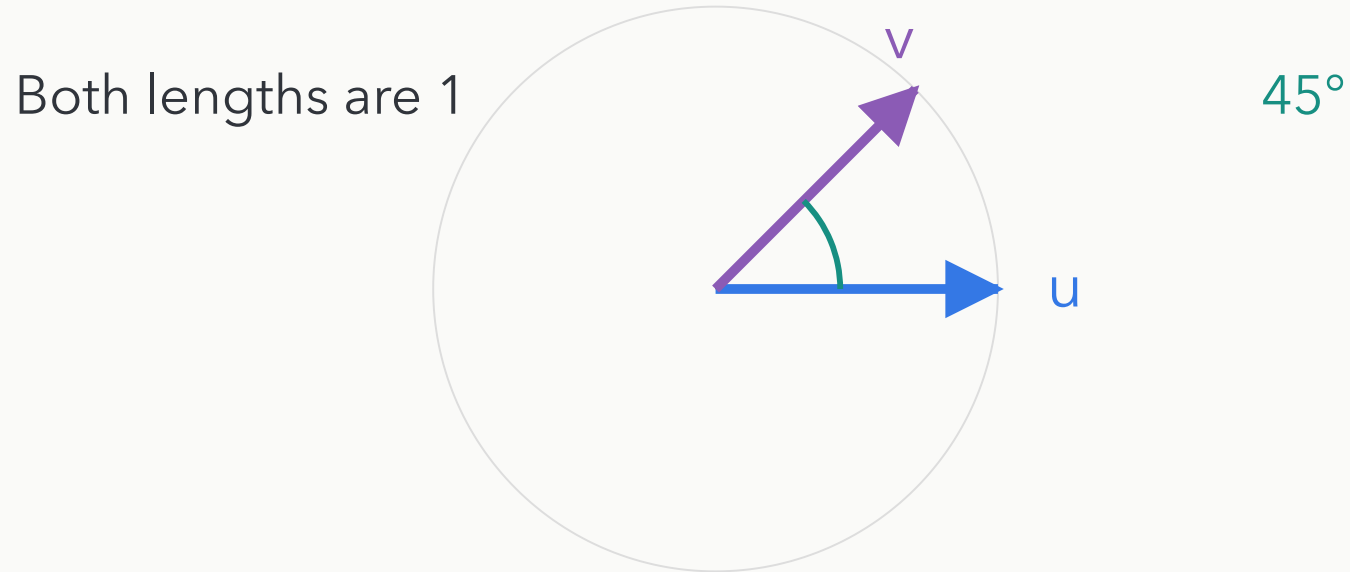
SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



| We now have one image vector and one text vector, each with 512 coordinates and unit length.

Only now compare the directions

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise

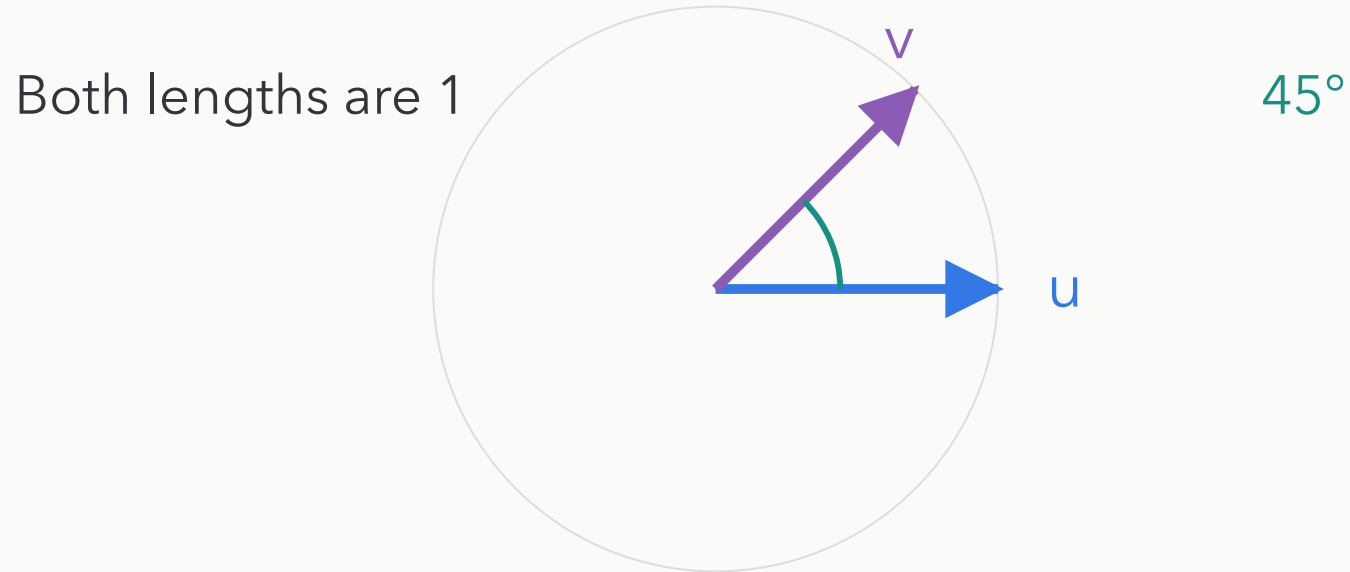


$$\mathbf{u}^\top \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta$$

| The dot product depends on both lengths and the angle between the vectors.

Unit length leaves only the angle

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise

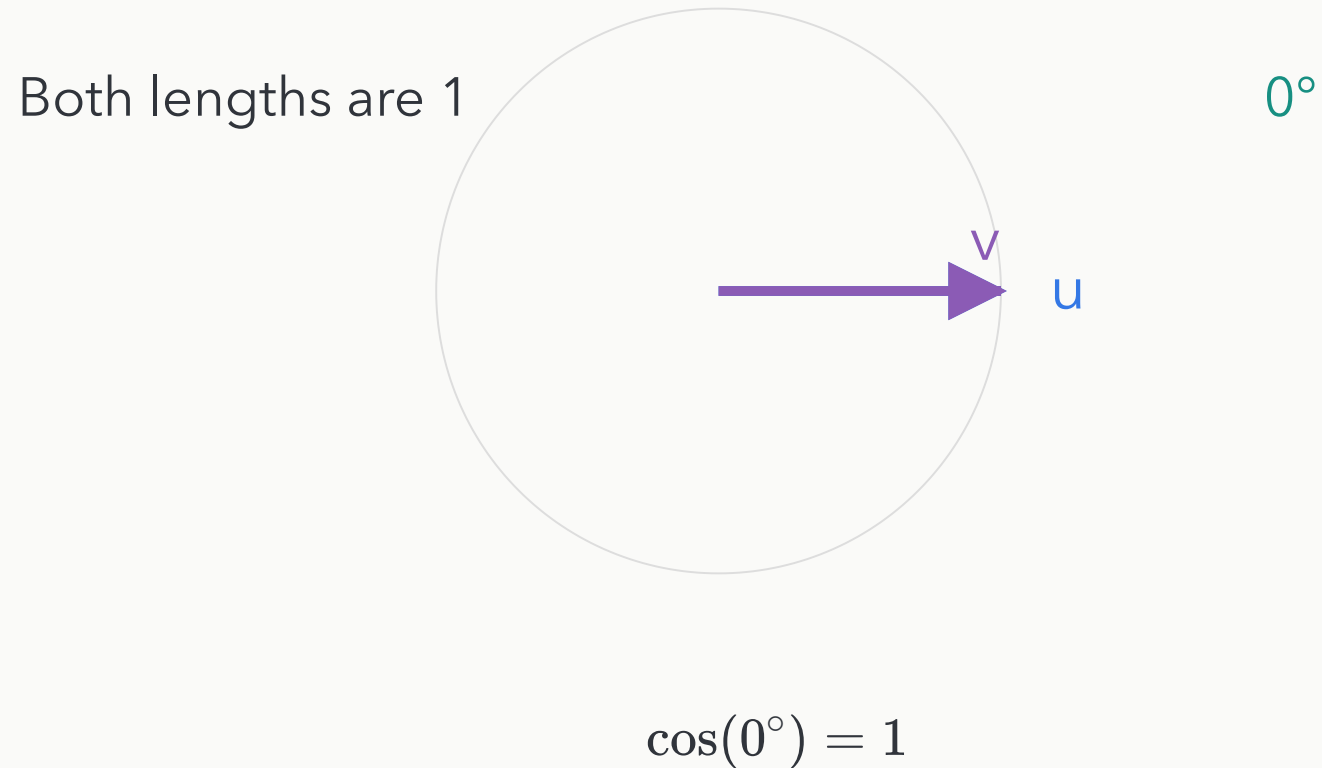


$$\mathbf{u}^\top \mathbf{v} = 1 \cdot 1 \cdot \cos \theta = \cos \theta$$

| Normalization makes the dot product equal to cosine similarity.

Same direction: cosine 1

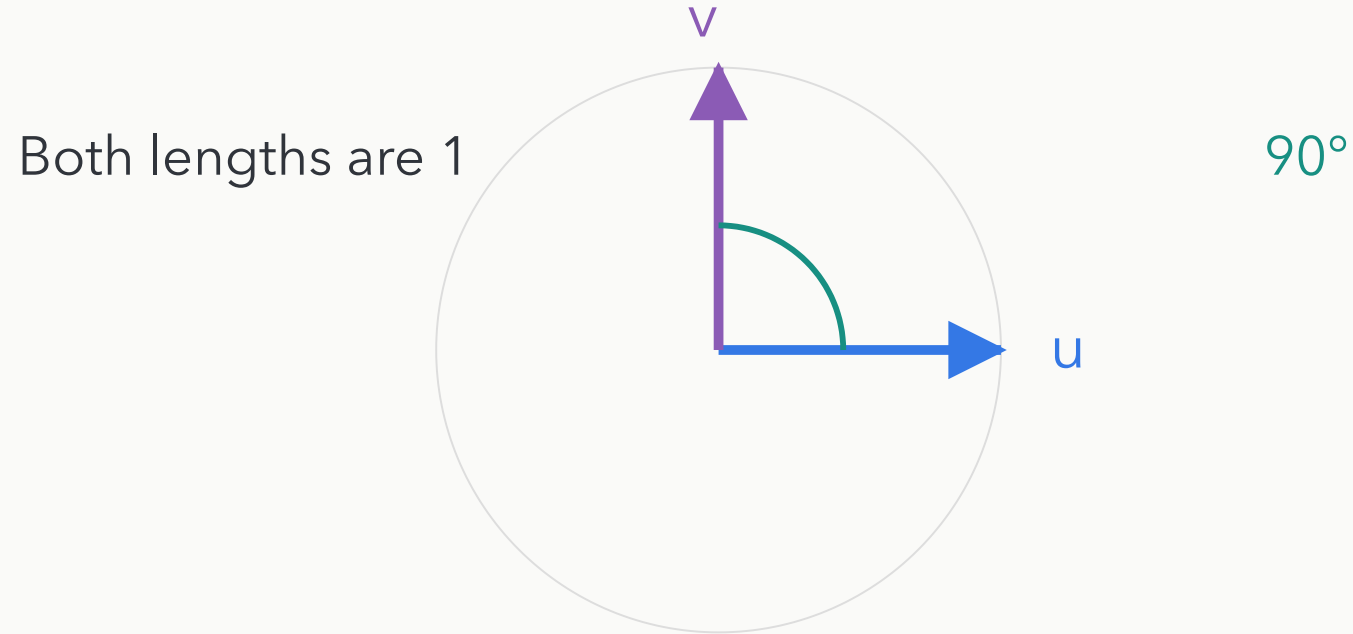
SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



This is geometry. Training must make this geometry useful.

A right angle: cosine 0

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise

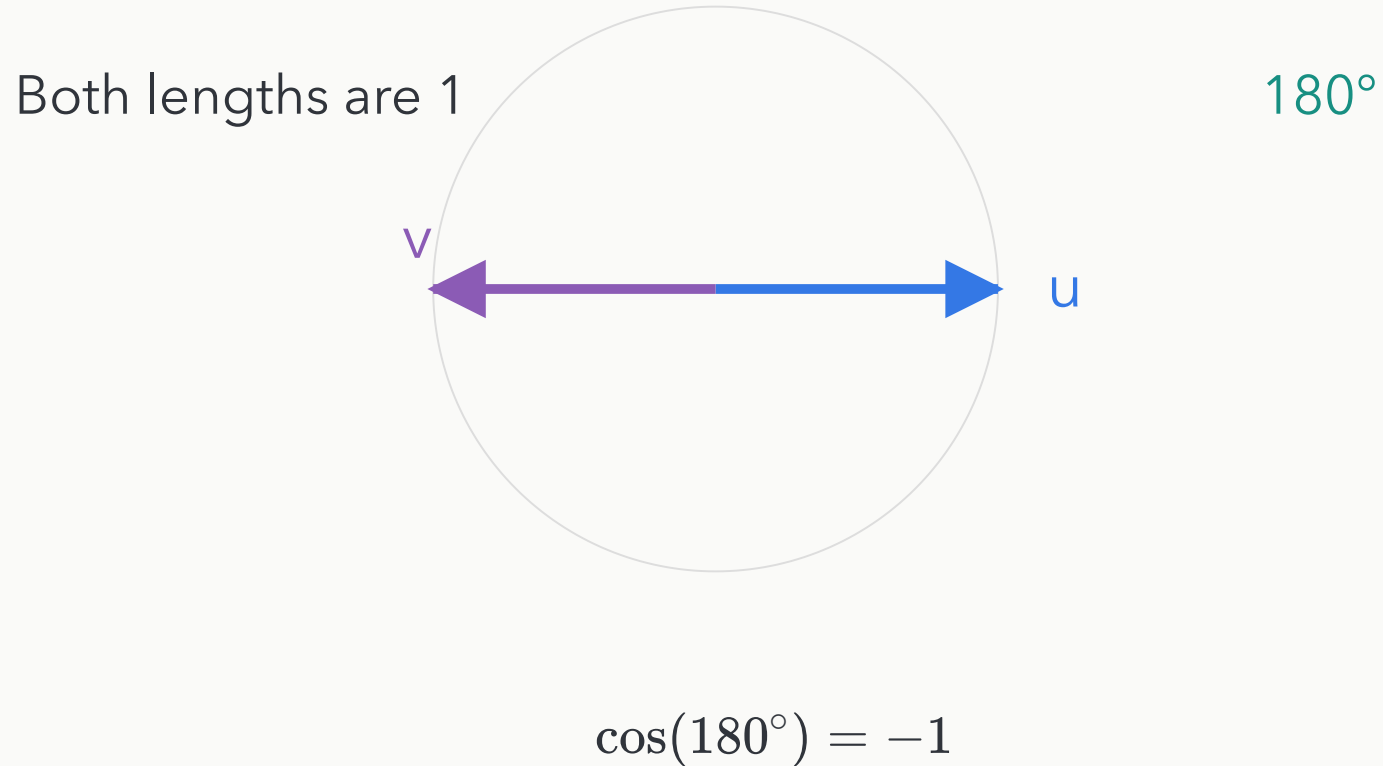


$$\cos(90^\circ) = 0$$

| This is geometry. Training must make this geometry useful.

Opposite directions: cosine -1

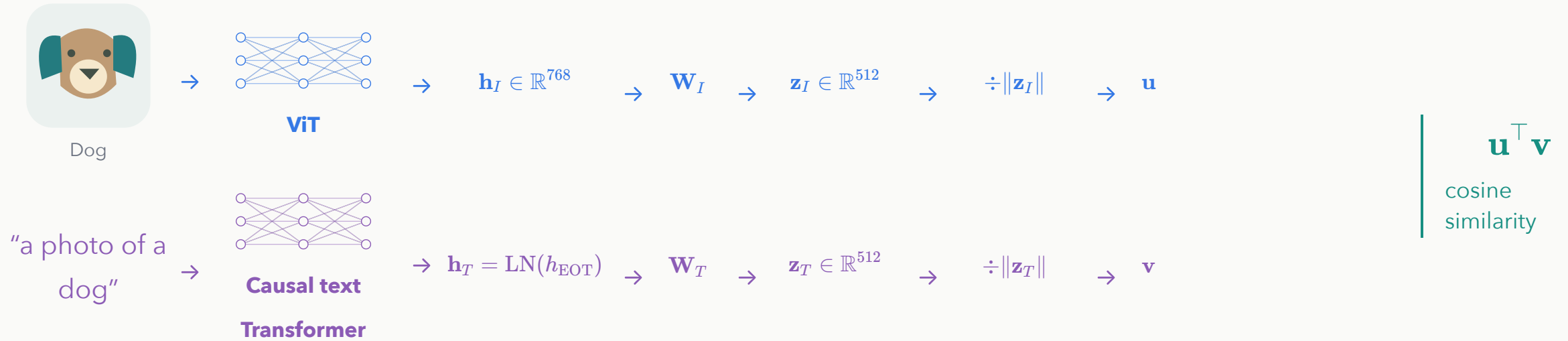
SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



This is geometry. Training must make this geometry useful.

Two independent branches meet at the score

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



The branches do not exchange tokens. They meet after encoding, at the dot product.

SECTION 03

Why should a cosine mean “match”?

Equal dimensions let us calculate a score. What makes the score meaningful?

Equal width is only the starting point

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise

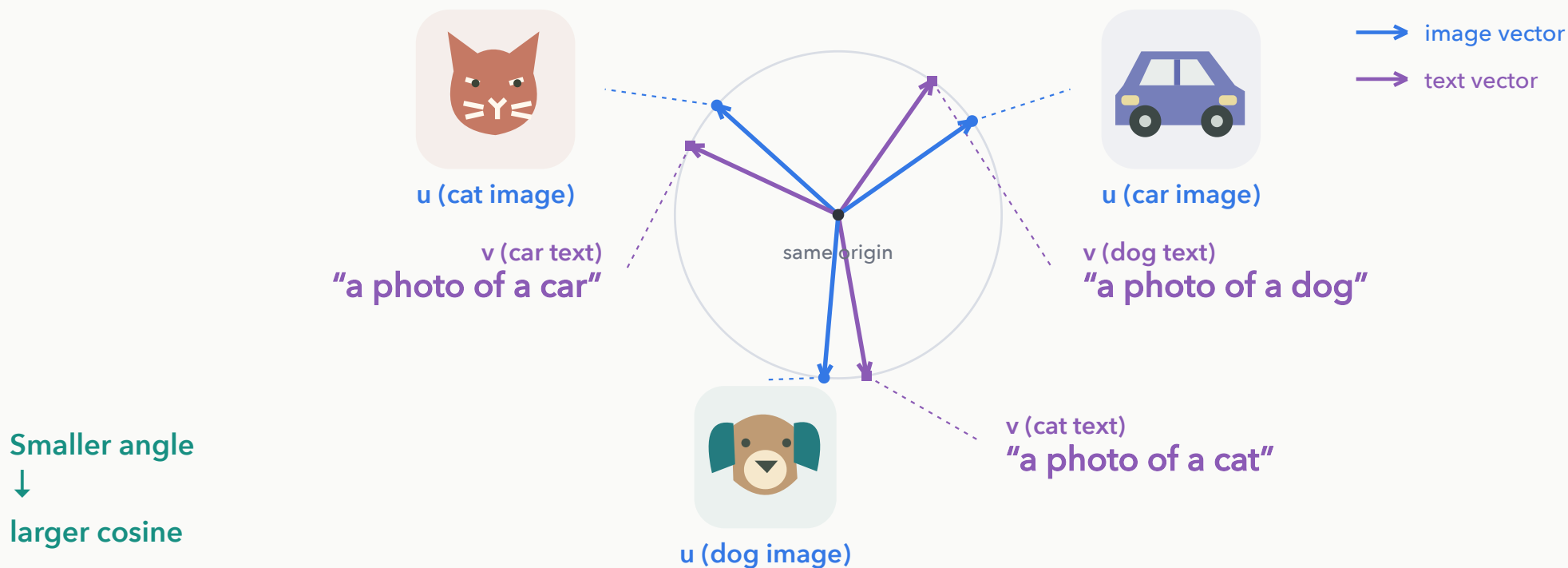
$$\mathbf{u}, \mathbf{v} \in \mathbb{R}^{512}$$

Would two random encoders understand each other?

| Same width permits comparison. Training gives the comparison meaning.

Before training: the closest caption can be wrong

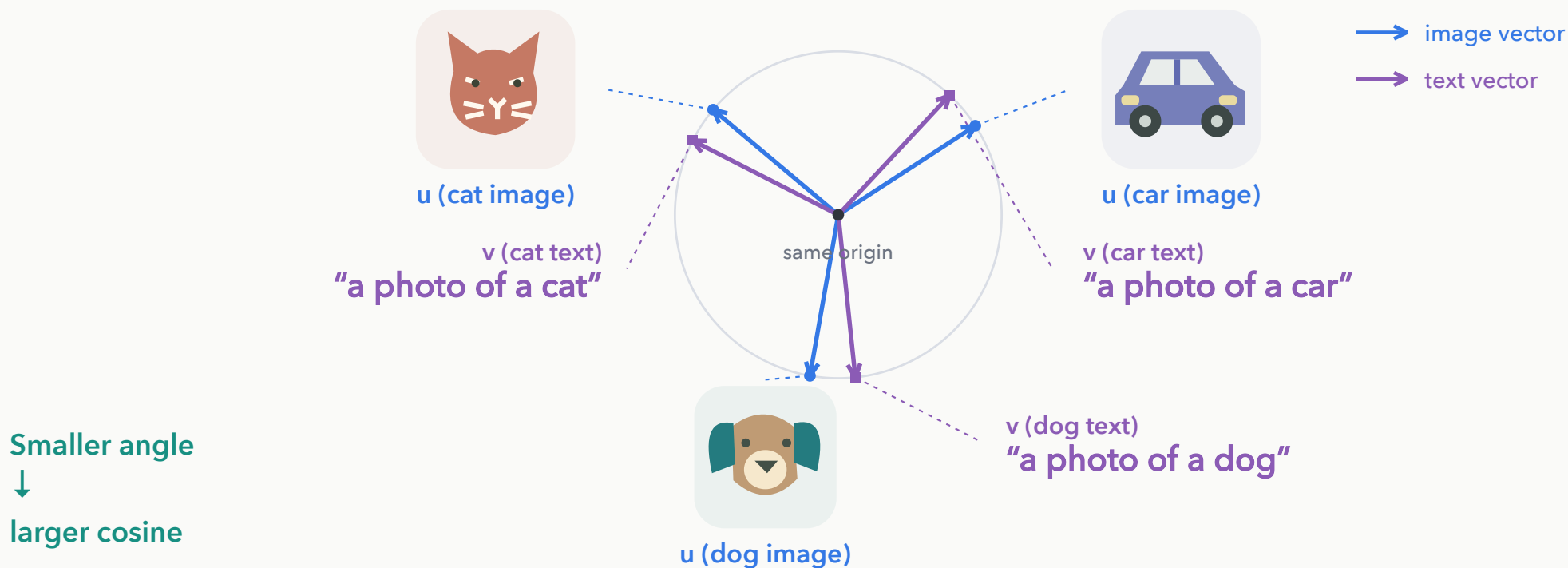
SCHEMATIC · illustrative directions, not measured checkpoints



The model has no useful pairing information merely because both outputs have width 512.

After training: paired directions move closer

SCHEMATIC · illustrative directions, not measured checkpoints



Learn a geometry in which observed pairs beat competing pairs.

An individual coordinate has no assigned concept

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise

$$\mathbf{u} = [u_1, u_2, \dots, u_{512}]$$

We did not label coordinates “dogness”, “redness” or “furriness”.

| The matching information is carried by the learned geometry.

If every caption scores 1, how do we choose?

HYPOTHETICAL FAILURE · these are illustrative scores, not measured CLIP predictions

Imagine the model maps every image and every caption to the **same unit vector**.



Dog



Compare this
image
with each caption

Supplied caption	Cosine with this image
"a photo of a dog" Paired description for this image	1.00
"a photo of a cat"	1.00
"a photo of a car"	1.00

Three equal scores. The paired description has no advantage.

| A high score helps only if the paired caption scores higher than the competing captions.

A paired description must beat its competitors

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



Dog



"a photo of a dog"

observed partner



"a photo of a cat"

competitor in this batch

| Contrastive training compares the observed pairing with alternative pairings.

SECTION 04

Follow one batch through CLIP

Keep the examples in view as we build the computation, one step at a time.

Start with images and the text found with them

SCHEMATIC · toy illustration and supplied caption; not an original CLIP web record

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › Update



Cat

"a photo of a cat"

One image-text record

Where does the text come from?

A webpage may have a caption, title or description associated with an image.

Collect those pairs. Filter the collection, then sample complete records for training.

| The pairing supplies the training target; the model does not write the caption.

Here are the 12 pairs in our small dataset

TOY DATASET · 12 authored image-caption pairs from the interactive

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › Update



Cat

"a photo of a cat"



Dog

"a photo of a dog"



Car

"a photo of a car"



Bicycle

"a bicycle"



Airplane

"an airplane"



Tree

"a tree"



Shoe

"a shoe"



Mug

"a mug"



Banana

"a banana"



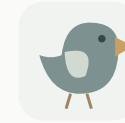
Chair

"a chair"



Building

"a building"



Bird

"a bird"

For this walkthrough, take cat, dog and car as the first batch.

Sample three complete pairs

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › Update



Cat

"a photo of a cat"



Dog

"a photo of a dog"



Car

"a photo of a car"

Batch 1: **cat · dog · car**

9 other records wait for later batches.

| Keep image 1 with text 1, image 2 with text 2, and image 3 with text 3.

The batch gives us partners and competitors

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > Columns > Loss > Update				
Images ↓ Text →		Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
 Image I_1 Cat		$I_1 \leftrightarrow T_1$ Observed pair	$I_1 \leftrightarrow T_2$ Competitor	$I_1 \leftrightarrow T_3$ Competitor
 Image I_2 Dog		$I_2 \leftrightarrow T_1$ Competitor	$I_2 \leftrightarrow T_2$ Observed pair	$I_2 \leftrightarrow T_3$ Competitor
 Image I_3 Car		$I_3 \leftrightarrow T_1$ Competitor	$I_3 \leftrightarrow T_2$ Competitor	$I_3 \leftrightarrow T_3$ Observed pair

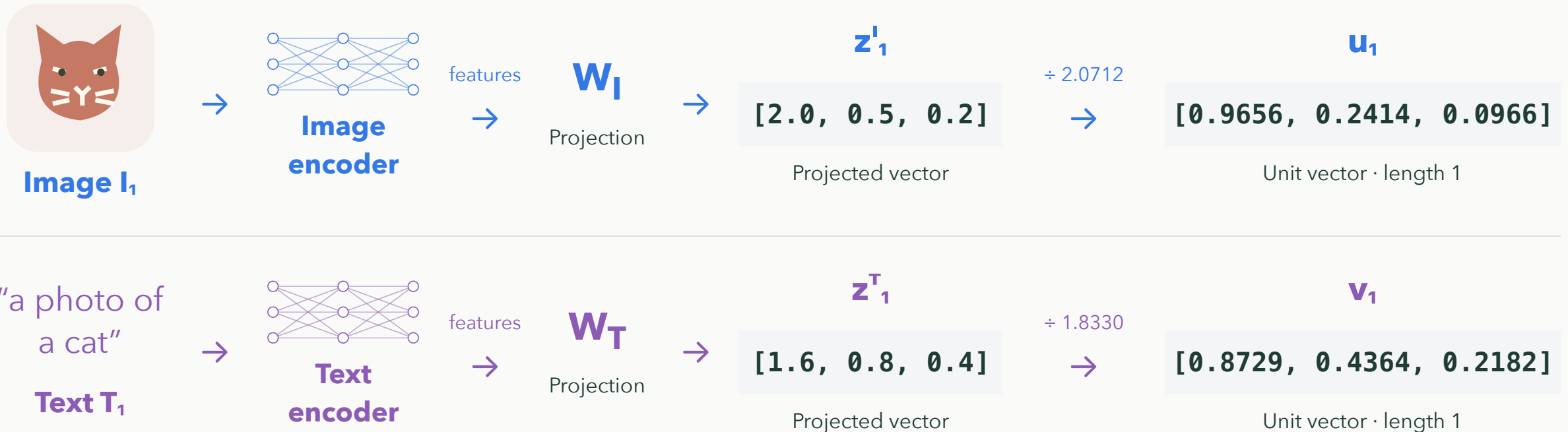
3 observed pairs · 6 competitors · 9 comparisons

| I_1 pairs with T_1 , I_2 with T_2 , and I_3 with T_3 . The other cells are competitors in this batch.

One image, one caption, two paths

TEACHING EXAMPLE · toy illustration, supplied caption, chosen vectors · 3 coordinates to show every calculation

Pairs > **Encode** > Normalize > Dot products > Temperature > Rows > Columns > Loss > Update

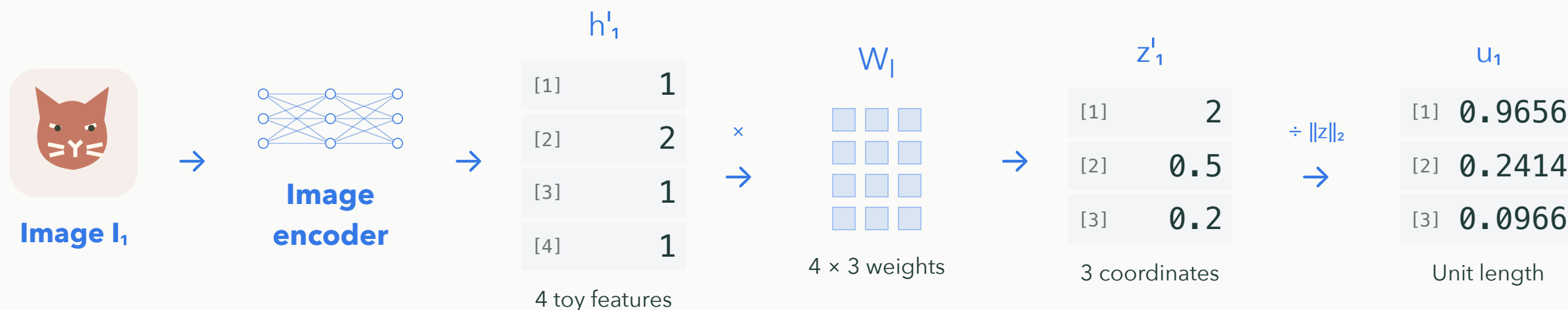


| We will compare the two final unit vectors with $u_1 \cdot v_1$. Next, work through each step.

Follow the cat image all the way to its unit vector

CHOSEN ARITHMETIC · 4 image features → 3 coordinates · same cat vector as the interactive

Pairs > **Encode** > Normalize > Dot products > Temperature > Rows > Columns > Loss > Update



The encoder makes features. The projection changes coordinates. Normalization sets the length to 1.

First, the encoder turns the image into features

CHOSEN ARITHMETIC · 4 image features → 3 coordinates · same cat vector as the interactive

Pairs > **Encode** > Normalize > Dot products > Temperature > Rows > Columns > Loss > Update



Image I_1 · cat

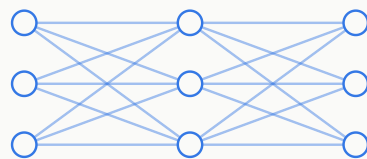


Image encoder



h'_1

[1]	1
[2]	2
[3]	1
[4]	1

1 row × 4 features

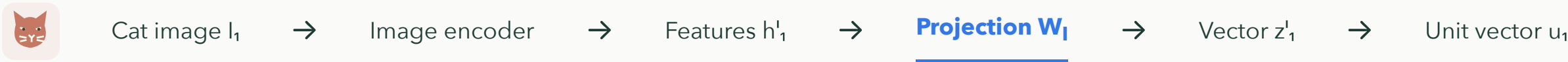
In the real ViT, we read the final CLS state after LayerNorm: **768 image features**.

For this arithmetic example, take $h'_1 = [1, 2, 1, 1]$. These four values are chosen.

Project the image features into three coordinates

CHOSEN ARITHMETIC · 4 image features → 3 coordinates · same cat vector as the interactive

Pairs > **Encode** > Normalize > Dot products > Temperature > Rows > Columns > Loss > Update



W_I · 4 input features × 3 output coordinates

	$z[1]$	$z[2]$	$z[3]$
$h[1] = 1$	1	0	0.1
$h[2] = 2$	0.5	0.2	0
$h[3] = 1$	0	0.1	0
$h[4] = 1$	0	0	0.1

Multiply h by W

$$\underbrace{h^I_1}_{1 \times 4} \underbrace{W_I}_{4 \times 3} = \underbrace{z^I_1}_{1 \times 3}$$

Each column of W supplies the weights for one output coordinate.

Use all four features for each column.

The projection learns how to combine features. Its output will be compared with text.

Calculate one output coordinate at a time

CHOSEN ARITHMETIC · 4 image features → 3 coordinates · same cat vector as the interactive

Pairs › **Encode** › Normalize › Dot products › Temperature › Rows › Columns › Loss › Update



Cat image I_1



Image encoder



Features h'_1



Projection W_I



Vector z'_1



Unit vector u_1

$$z[1] \quad (1 \times 1) + (2 \times 0.5) + (1 \times 0) + (1 \times 0) \quad = 2$$

$$z[2] \quad (1 \times 0) + (2 \times 0.2) + (1 \times 0.1) + (1 \times 0) \quad = 0.5$$

$$z[3] \quad (1 \times 0.1) + (2 \times 0) + (1 \times 0) + (1 \times 0.1) \quad = 0.2$$

$$\mathbf{z}_1^I = [2, 0.5, 0.2] \in \mathbb{R}^3$$

This is the interactive's raw cat vector. Next, divide it by its length.

Normalize the cat image vector

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › **Normalize** › Dot products › Temperature › Rows › Columns › Loss › Update



Cat image I_1



Image encoder



Features h'_1



Projection W_1



Vector z'_1



Unit vector u_1



Image I_1

Projected vector

[2.0, 0.5, 0.2]

$$\mathbf{u}_1 = \mathbf{z} / \|\mathbf{z}\|_2$$

Divide each coordinate by the same length

$$\|\mathbf{z}\|_2 = \sqrt{2^2 + 0.5^2 + 0.2^2} = 2.071232$$

$$2.0 \div 2.071232 = \mathbf{0.965609}$$

$$0.5 \div 2.071232 = \mathbf{0.241402}$$

$$0.2 \div 2.071232 = \mathbf{0.096561}$$

u_1

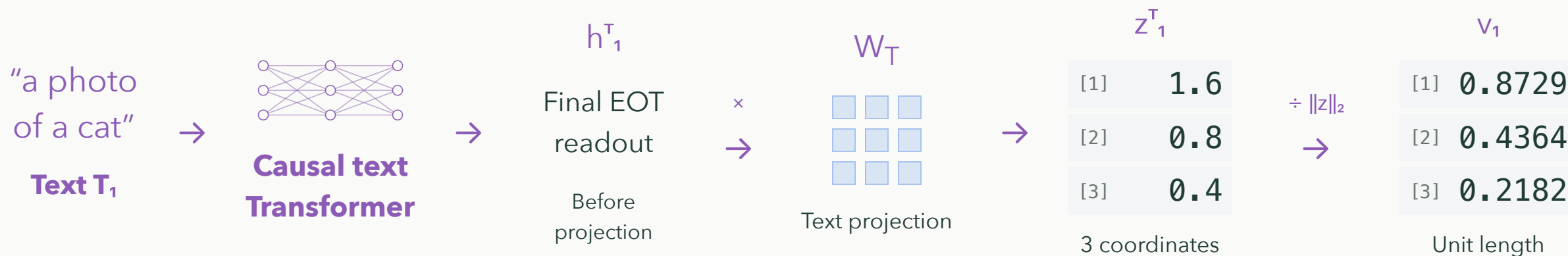
[0.9656, 0.2414, 0.0966]

The vector still has three coordinates. Normalization changes its length, not its direction.

The caption follows its own encoder and projection

CHOSEN 3D EXAMPLE · text vector from the interactive · real ViT-B/32 text projection: 512 → 512

Pairs > **Encode** > Normalize > Dot products > Temperature > Rows > Columns > Loss > Update



The two branches use different weights. Their projected vectors have the same number of coordinates.

Normalize the cat text vector

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › **Normalize** › Dot products › Temperature › Rows › Columns › Loss › Update

“a photo of a cat”

Text T_1

Projected vector

[1.6, 0.8, 0.4]

$$\mathbf{v}_1 = \mathbf{z} / \|\mathbf{z}\|_2$$

Divide each coordinate by the same length

$$\|\mathbf{z}\|_2 = \sqrt{1.6^2 + 0.8^2 + 0.4^2} = 1.833030$$

$$1.6 \div 1.833030 = \mathbf{0.872872}$$

$$0.8 \div 1.833030 = \mathbf{0.436436}$$

$$0.4 \div 1.833030 = \mathbf{0.218218}$$

$\mathbf{v}_1$

$$\mathbf{[0.8729, 0.4364, 0.2182]}$$

The vector still has three coordinates. Normalization changes its length, not its direction.

Now compare the image with its caption

CHOSEN 3D VECTORS · unit-vector dot product · shown coordinates are rounded; calculation uses full precision

Pairs › Encode › Normalize › **Dot products** › Temperature › Rows › Columns › Loss › Update



Image $I_1 \rightarrow u_1$

[0.9656, 0.2414, 0.0966]

“a photo of a cat”

Text $T_1 \rightarrow v_1$

[0.8729, 0.4364, 0.2182]

$$(0.9656 \times 0.8729) + (0.2414 \times 0.4364) + (0.0966 \times 0.2182)$$

Cosine for I_1 and T_1

0.969281

One image and one caption give one score. Next, repeat this calculation for the other pairs.

Repeat both branches for every pair

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › **Normalize** › Dot products › Temperature › Rows › Columns › Loss › Update

Record	Raw image output z'	Raw text output z^T
 "a photo of a cat"	[2.0, 0.5, 0.2]	[1.6, 0.8, 0.4]
 "a photo of a dog"	[0.4, 2.0, 0.1]	[0.7, 1.4, 0.3]
 "a photo of a car"	[0.2, 0.3, 2.0]	[0.4, 0.5, 1.5]

Use the same image weights for all images and the same text weights for all captions. Normalize every output.

Six unit vectors, ready to compare

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › **Normalize** › Dot products › Temperature › Rows › Columns › Loss › Update

U · image vectors

u_1 [0.9656, 0.2414, 0.0966]

u_2 [0.1959, 0.9794, 0.0490]

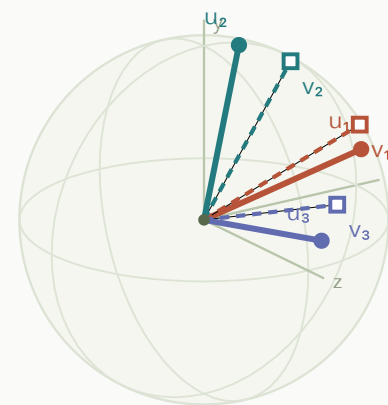
u_3 [0.0984, 0.1476, 0.9841]

V · text vectors

v_1 [0.8729, 0.4364, 0.2182]

v_2 [0.4392, 0.8784, 0.1882]

v_3 [0.2453, 0.3066, 0.9197]



● solid = image □ dashed = text · colour = pair

Colour identifies a pair. Each vector has length 1; the directions can still differ.

Stack the image vectors as rows

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › **Normalize** › Dot products › Temperature › Rows › Columns › Loss › Update

U: image rows

Rows ↓ / columns →	Coordinate 1	Coordinate 2	Coordinate 3
Image 1 · cat	0.9656	0.2414	0.0966
Image 2 · dog	0.1959	0.9794	0.0490
Image 3 · car	0.0984	0.1476	0.9841

3 images × 3 coordinates

Row 1 is the cat image. Its three columns are coordinates, not classes.

Stack the text vectors in the same pair order

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › **Normalize** › Dot products › Temperature › Rows › Columns › Loss › Update

V: text rows

Rows ↓ / columns →	Coordinate 1	Coordinate 2	Coordinate 3
Text 1 · cat	0.8729	0.4364	0.2182
Text 2 · dog	0.4392	0.8784	0.1882
Text 3 · car	0.2453	0.3066	0.9197

3 texts × 3 coordinates

Row 1 is “a photo of a cat”. Keep it aligned with image row 1.

Transpose V: each caption now has a column

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › **Dot products** › Temperature › Rows › Columns › Loss › Update

V^T : text columns

Rows ↓ / columns →	Text 1 cat	Text 2 dog	Text 3 car
Coordinate 1	0.8729	0.4392	0.2453
Coordinate 2	0.4364	0.8784	0.3066
Coordinate 3	0.2182	0.1882	0.9197

3 coordinates × 3 texts

Transpose moves the numbers. Text 1’s coordinates now run down column 1.

An image row meets a text column

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › **Dot products** › Temperature › Rows › Columns › Loss › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

Observed pairs

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	$I_1 \leftrightarrow T_1$ positive	$I_1 \leftrightarrow T_2$ negative	$I_1 \leftrightarrow T_3$ negative
Image I_2 dog	$I_2 \leftrightarrow T_1$ negative	$I_2 \leftrightarrow T_2$ positive	$I_2 \leftrightarrow T_3$ negative
Image I_3 car	$I_3 \leftrightarrow T_1$ negative	$I_3 \leftrightarrow T_2$ negative	$I_3 \leftrightarrow T_3$ positive

DIAGONAL = PAIRED PARTNERS

One dot product per cell

$$\underbrace{\mathbf{U}}_{3 \times 3} \underbrace{\mathbf{V}^T}_{3 \times 3} = \underbrace{\mathbf{C}}_{3 \times 3}$$

U: image rows × coordinates

V^T: coordinates × text columns

C: image rows × text columns

In real ViT-B/32: (3×512) times (512×3) still gives nine scores.

Follow the cat image and the cat caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › **Dot products** › Temperature › Rows › Columns › Loss › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

$C \cdot \text{cosine}$

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	?	?	?
Image I_2 dog	?	?	?
Image I_3 car	?	?	?

DIAGONAL = PAIRED PARTNERS

Build C_{11}

$$C_{11} = \mathbf{u}_1^\top \mathbf{v}_1$$

$$\mathbf{u}_1 = [0.9656, 0.2414, 0.0966]$$

$$\mathbf{v}_1 = [0.8729, 0.4364, 0.2182]$$

Multiply matching coordinates, then add the three products.

The same coordinate positions meet: first with first, second with second, third with third.

Cat image × cat caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.969 correct pair	?	?
Image I ₂ dog	?	?	?
Image I ₃ car	?	?	?

DIAGONAL = PAIRED PARTNERS

Cell C₁₁

$$\begin{aligned} & (0.9656 \times 0.8729) \\ & + (0.2414 \times 0.4364) \\ & + (0.0966 \times 0.2182) \\ & 0.842853 + 0.105357 + 0.021071 \end{aligned}$$

Cosine **0.969281**

| This is the observed partner.

Cat image × dog caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.969 correct pair	0.654 I ₁ · T ₂	?
Image I ₂ dog	?	?	?
Image I ₃ car	?	?	?

DIAGONAL = PAIRED PARTNERS

Cell C₁₂

$$(0.9656 \times 0.4392)$$
$$+ (0.2414 \times 0.8784)$$
$$+ (0.0966 \times 0.1882)$$

$$0.424114 + 0.212057 + 0.018176$$

Cosine

0.654347

This is an in-batch competitor. Use the same dot-product calculation.

Cat image × car caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.969 correct pair	0.654 I ₁ · T ₂	0.400 I ₁ · T ₃
Image I ₂ dog	?	?	?
Image I ₃ car	?	?	?

DIAGONAL = PAIRED PARTNERS

Cell C₁₃

(0.9656 × 0.2453)

+ (0.2414 × 0.3066)

+ (0.0966 × 0.9197)

0.236821 + 0.074007 + 0.088808

Cosine

0.399636

This is an in-batch competitor. Use the same dot-product calculation.

Dog image × cat caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.969 correct pair	0.654 I ₁ · T ₂	0.400 I ₁ · T ₃
Image I ₂ dog	0.609 I ₂ · T ₁	?	?
Image I ₃ car	?	?	?

DIAGONAL = PAIRED PARTNERS

Cell C₂₁

$$(0.1959 \times 0.8729)$$
$$+ (0.9794 \times 0.4364)$$
$$+ (0.0490 \times 0.2182)$$
$$0.170979 + 0.427447 + 0.010686$$

Cosine

0.609112

This is an in-batch competitor. Use the same dot-product calculation.

Dog image × dog caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T₁ cat	Text T₂ dog	Text T₃ car
Image I₁ cat	0.969 correct pair	0.654 $I_1 \cdot T_2$	0.400 $I_1 \cdot T_3$
Image I₂ dog	0.609 $I_2 \cdot T_1$	0.956 correct pair	?
Image I₃ car	?	?	?

DIAGONAL = PAIRED PARTNERS

Cell C₂₂

(0.1959×0.4392)
 $+ (0.9794 \times 0.8784)$
 $+ (0.0490 \times 0.1882)$
 $0.086035 + 0.860346 + 0.009218$

Cosine

0.955599

| This is the observed partner.

Dog image × car caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T₁ cat	Text T₂ dog	Text T₃ car
Image I₁ cat	0.969 correct pair	0.654 $I_1 \cdot T_2$	0.400 $I_1 \cdot T_3$
Image I₂ dog	0.609 $I_2 \cdot T_1$	0.956 correct pair	0.393 $I_2 \cdot T_3$
Image I₃ car	?	?	?

DIAGONAL = PAIRED PARTNERS

Cell C₂₃

$$(0.1959 \times 0.2453)$$
$$+ (0.9794 \times 0.3066)$$
$$+ (0.0490 \times 0.9197)$$

$$0.048041 + 0.300256 + 0.045038$$

Cosine

0.393335

This is an in-batch competitor. Use the same dot-product calculation.

Car image × cat caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T₁ cat	Text T₂ dog	Text T₃ car
Image I₁ cat	0.969 correct pair I₁ · T₁	0.654 I₁ · T₂	0.400 I₁ · T₃
Image I₂ dog	0.609 I₂ · T₁	0.956 correct pair I₂ · T₂	0.393 I₂ · T₃
Image I₃ car	0.365 I₃ · T₁	?	?

DIAGONAL = PAIRED PARTNERS

Cell C₃₁

$$(0.0984 \times 0.8729)$$
$$+ (0.1476 \times 0.4364)$$
$$+ (0.9841 \times 0.2182)$$
$$0.085902 + 0.064427 + 0.214756$$

Cosine

0.365085

| This is an in-batch competitor. Use the same dot-product calculation.

Car image × dog caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.969 correct pair I₁ · T₁	0.654 I₁ · T₂	0.400 I₁ · T₃
Image I ₂ dog	0.609 I₂ · T₁	0.956 correct pair I₂ · T₂	0.393 I₂ · T₃
Image I ₃ car	0.365 I₃ · T₁	0.358 I₃ · T₂	? I₃ · T₃

DIAGONAL = PAIRED PARTNERS

Cell C₃₂

$$(0.0984 \times 0.4392)$$
$$+ (0.1476 \times 0.8784)$$
$$+ (0.9841 \times 0.1882)$$
$$0.043225 + 0.129675 + 0.185250$$

Cosine

0.358151

This is an in-batch competitor. Use the same dot-product calculation.

Car image × car caption

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

C · cosine

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.969 correct pair I₁ · T₁	0.654 I₁ · T₂	0.400 I₁ · T₃
Image I ₂ dog	0.609 I₂ · T₁	0.956 correct pair I₂ · T₂	0.393 I₂ · T₃
Image I ₃ car	0.365 I₃ · T₁	0.358 I₃ · T₂	0.975 correct pair I₃ · T₃

DIAGONAL = PAIRED PARTNERS

Cell C₃₃

(0.0984 × 0.2453)
+ (0.1476 × 0.3066)
+ (0.9841 × 0.9197)

0.024136 + 0.045256 + 0.905118

Cosine

0.974511

| This is the observed partner.

Now every image has three caption scores

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > **Dot products** > Temperature > Rows > Columns > Loss > Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

$C \cdot \text{cosine}$

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.969 correct pair $I_1 \cdot T_1$	0.654 $I_1 \cdot T_2$	0.400 $I_1 \cdot T_3$
Image I_2 dog	0.609 $I_2 \cdot T_1$	0.956 correct pair $I_2 \cdot T_2$	0.393 $I_2 \cdot T_3$
Image I_3 car	0.365 $I_3 \cdot T_1$	0.358 $I_3 \cdot T_2$	0.975 correct pair $I_3 \cdot T_3$

DIAGONAL = PAIRED PARTNERS

Read the cat row

Cat caption: **0.9693**

Dog caption: 0.6543

Car caption: 0.3996

$$C = UV^T$$

Cosines describe directions. We have not calculated probabilities yet.

SECTION 05

From competition to a training loss

Which caption belongs to each image—and which image belongs to each caption?

Scale the cosines before making a choice

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › **Temperature** › Rows › Columns › Loss › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

$S \cdot \text{logits}$

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.939 correct pair	1.309 $I_1 \cdot T_2$	0.799 $I_1 \cdot T_3$
Image I_2 dog	1.218 $I_2 \cdot T_1$	1.911 correct pair	0.787 $I_2 \cdot T_3$
Image I_3 car	0.730 $I_3 \cdot T_1$	0.716 $I_3 \cdot T_2$	1.949 correct pair

DIAGONAL = PAIRED PARTNERS

Temperature $\tau = 0.5$

$$S_{ij} = C_{ij} / \tau$$

$$S_{11} = 0.969281 / 0.5$$

Cat-cat logit

1.938561

Every entry is multiplied by 2.

Scaling preserves the ranking. It changes how sharply softmax will separate the choices.

Temperature changes the strength of competition

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › **Temperature** › Rows › Columns › Loss › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

Keep the cat-image cosines: [0.9693, 0.6543, 0.3996].

Score scale $\alpha = 1 / \tau$ 2.0



T1 **53.98%**

T2 **28.75%**

T3 **17.27%**

$\alpha = 2.0$; $\tau = 1/\alpha = 0.500$. The three shares sum to 100%.

Smaller τ gives sharper shares; larger τ spreads them more evenly. The top caption stays the same.

Take the cat image's row of logits

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update



Image I_1 · cat

Keep this image fixed.

S · logits

	TEXT COLUMNS →		
	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.9386 $I_1 \cdot T_1$	1.3087 $I_1 \cdot T_2$	0.7993 $I_1 \cdot T_3$
Image I_2 dog	1.2182 $I_2 \cdot T_1$	1.9112 $I_2 \cdot T_2$	0.7867 $I_2 \cdot T_3$
Image I_3 car	0.7302 $I_3 \cdot T_1$	0.7163 $I_3 \cdot T_2$	1.9490 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1	Text T_2	Text T_3
	"a photo of a cat"	"a photo of a dog"	"a photo of a car"
Logits S_{1j}	1.9386	1.3087	0.7993

These are the three scores for Image I_1 . Softmax turns them into shares of one total.

Values shown are rounded; calculations use full precision.

Read across: T_1 is the cat caption, T_2 the dog caption, and T_3 the car caption.

Exponentiate each score in this row

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update



Image I_1 · cat

Keep this image fixed.

S · logits

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.9386 $I_1 \cdot T_1$	1.3087 $I_1 \cdot T_2$	0.7993 $I_1 \cdot T_3$
Image I_2 dog	1.2182 $I_2 \cdot T_1$	1.9112 $I_2 \cdot T_2$	0.7867 $I_2 \cdot T_3$
Image I_3 car	0.7302 $I_3 \cdot T_1$	0.7163 $I_3 \cdot T_2$	1.9490 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{1j}	1.9386	1.3087	0.7993
Exponentiate $e^{S_{1j}}$	$e^{1.9386}$ ≈ 6.9487	$e^{1.3087}$ ≈ 3.7013	$e^{0.7993}$ ≈ 2.2239

Each score becomes a positive weight. A larger score gives a larger weight.

Values shown are rounded; calculations use full precision.

The three captions still occupy the same three columns.

Add the three weights in this row

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update

**Image I_1 · cat**
Keep this image fixed.

$S \cdot \text{logits}$

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.9386 $I_1 \cdot T_1$	1.3087 $I_1 \cdot T_2$	0.7993 $I_1 \cdot T_3$
Image I_2 dog	1.2182 $I_2 \cdot T_1$	1.9112 $I_2 \cdot T_2$	0.7867 $I_2 \cdot T_3$
Image I_3 car	0.7302 $I_3 \cdot T_1$	0.7163 $I_3 \cdot T_2$	1.9490 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{1j}	1.9386	1.3087	0.7993
Exponentiate $e^{S_{1j}}$	$e^{1.9386}$ ≈ 6.9487	$e^{1.3087}$ ≈ 3.7013	$e^{0.7993}$ ≈ 2.2239

Add across this row:

$$D_1 = e^{1.9386} + e^{1.3087} + e^{0.7993}$$
$$\approx 6.9487 + 3.7013 + 2.2239 \approx 12.8740$$

Values shown are rounded; calculations use full precision.

This one sum is the denominator for all three captions competing for Image I_1 .

Divide the cat caption's weight by the row sum

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update

**Image I_1 · cat**
Keep this image fixed.

S · logits

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.9386 $I_1 \cdot T_1$	1.3087 $I_1 \cdot T_2$	0.7993 $I_1 \cdot T_3$
Image I_2 dog	1.2182 $I_2 \cdot T_1$	1.9112 $I_2 \cdot T_2$	0.7867 $I_2 \cdot T_3$
Image I_3 car	0.7302 $I_3 \cdot T_1$	0.7163 $I_3 \cdot T_2$	1.9490 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{1j}	1.9386	1.3087	0.7993
Exponentiate $e^{S_{1j}}$	$e^{1.9386}$ ≈ 6.9487	$e^{1.3087}$ ≈ 3.7013	$e^{0.7993}$ ≈ 2.2239
Divide by the row sum P_{1j}	$\frac{6.9487}{12.8740}$ ≈ 0.5398	Next caption	Next caption

Every caption uses the same row sum: $D_1 \approx 12.8740$.

Values shown are rounded; calculations use full precision.

Softmax share = this caption's exponential weight ÷ the sum of all three weights in the row.

Divide the dog caption's weight by the row sum

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update

**Image I_1 · cat**
Keep this image fixed.

S · logits

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.9386 $I_1 \cdot T_1$	1.3087 $I_1 \cdot T_2$	0.7993 $I_1 \cdot T_3$
Image I_2 dog	1.2182 $I_2 \cdot T_1$	1.9112 $I_2 \cdot T_2$	0.7867 $I_2 \cdot T_3$
Image I_3 car	0.7302 $I_3 \cdot T_1$	0.7163 $I_3 \cdot T_2$	1.9490 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{1j}	1.9386	1.3087	0.7993
Exponentiate $e^{S_{1j}}$	$e^{1.9386}$ ≈ 6.9487	$e^{1.3087}$ ≈ 3.7013	$e^{0.7993}$ ≈ 2.2239
Divide by the row sum P_{1j}	$\frac{6.9487}{12.8740}$ ≈ 0.5398	$\frac{3.7013}{12.8740}$ ≈ 0.2875	Next caption

Every caption uses the same row sum: $D_1 \approx 12.8740$.

Values shown are rounded; calculations use full precision.

Softmax share = this caption's exponential weight ÷ the sum of all three weights in the row.

Divide the car caption's weight by the row sum

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update

**Image I_1 · cat**
Keep this image fixed.

$S \cdot \text{logits}$

	TEXT COLUMNS →		
	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.9386 $I_1 \cdot T_1$	1.3087 $I_1 \cdot T_2$	0.7993 $I_1 \cdot T_3$
Image I_2 dog	1.2182 $I_2 \cdot T_1$	1.9112 $I_2 \cdot T_2$	0.7867 $I_2 \cdot T_3$
Image I_3 car	0.7302 $I_3 \cdot T_1$	0.7163 $I_3 \cdot T_2$	1.9490 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{1j}	1.9386	1.3087	0.7993
Exponentiate $e^{S_{1j}}$	$e^{1.9386}$ ≈ 6.9487	$e^{1.3087}$ ≈ 3.7013	$e^{0.7993}$ ≈ 2.2239
Divide by the row sum P_{1j}	$\frac{6.9487}{12.8740}$ ≈ 0.5398	$\frac{3.7013}{12.8740}$ ≈ 0.2875	$\frac{2.2239}{12.8740}$ ≈ 0.1727

Every caption uses the same row sum: $D_1 \approx 12.8740$.

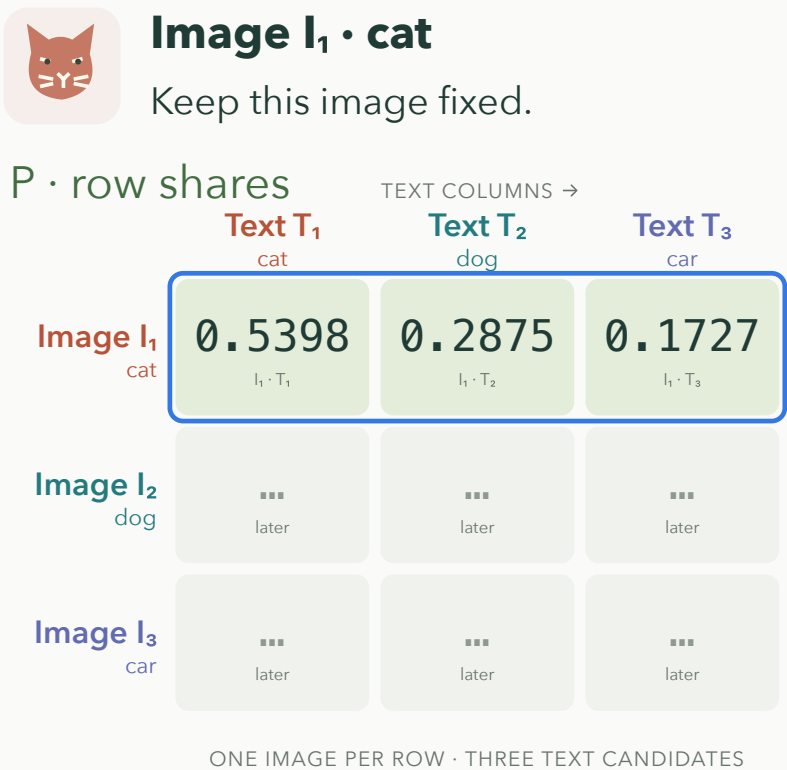
Values shown are rounded; calculations use full precision.

Softmax share = this caption's exponential weight ÷ the sum of all three weights in the row.

Now the cat image's three shares sum to 1

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update



Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{1j}	1.9386	1.3087	0.7993
Exponentiate $e^{S_{1j}}$	$e^{1.9386}$ ≈ 6.9487	$e^{1.3087}$ ≈ 3.7013	$e^{0.7993}$ ≈ 2.2239
Divide by the row sum P_{1j}	$\frac{6.9487}{12.8740}$ ≈ 0.5398	$\frac{3.7013}{12.8740}$ ≈ 0.2875	$\frac{2.2239}{12.8740}$ ≈ 0.1727

$$0.5398 + 0.2875 + 0.1727 = 1.0000$$

Values shown are rounded; calculations use full precision.

P stores the softmax shares. We have completed only the first image row so far.

Repeat the same calculation for the dog image

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update

 **Image I_2 · dog**
Keep this image fixed.

P · row shares

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.5398 $I_1 \cdot T_1$	0.2875 $I_1 \cdot T_2$	0.1727 $I_1 \cdot T_3$
Image I_2 dog	0.2740 $I_2 \cdot T_1$	0.5480 $I_2 \cdot T_2$	0.1780 $I_2 \cdot T_3$
Image I_3 car	... later	... later	... later

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{2j}	1.2182	1.9112	0.7867
Exponentiate $e^{S_{2j}}$	$e^{1.2182}$ ≈ 3.3812	$e^{1.9112}$ ≈ 6.7612	$e^{0.7867}$ ≈ 2.1961
Divide by the row sum P_{2j}	$\frac{3.3812}{12.3384}$ ≈ 0.2740	$\frac{6.7612}{12.3384}$ ≈ 0.5480	$\frac{2.1961}{12.3384}$ ≈ 0.1780

Every caption uses the same row sum: **$D_2 \approx 12.3384$** .

Values shown are rounded; calculations use full precision.

This image gets its own denominator: add the exponential weights across its row.

Repeat the same calculation for the car image

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > **Rows** > Columns > Loss > Update



Image I_3 · car

Keep this image fixed.

P · row shares

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.5398 $I_1 \cdot T_1$	0.2875 $I_1 \cdot T_2$	0.1727 $I_1 \cdot T_3$
Image I_2 dog	0.2740 $I_2 \cdot T_1$	0.5480 $I_2 \cdot T_2$	0.1780 $I_2 \cdot T_3$
Image I_3 car	0.1862 $I_3 \cdot T_1$	0.1837 $I_3 \cdot T_2$	0.6301 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

Compare the three captions across this row →

Same image in every cell	Text T_1 "a photo of a cat"	Text T_2 "a photo of a dog"	Text T_3 "a photo of a car"
Logits S_{3j}	0.7302	0.7163	1.9490
Exponentiate $e^{S_{3j}}$	$e^{0.7302}$ ≈ 2.0754	$e^{0.7163}$ ≈ 2.0468	$e^{1.9490}$ ≈ 7.0218
Divide by the row sum P_{3j}	$\frac{2.0754}{11.1441}$ ≈ 0.1862	$\frac{2.0468}{11.1441}$ ≈ 0.1837	$\frac{7.0218}{11.1441}$ ≈ 0.6301

Every caption uses the same row sum: $D_3 \approx 11.1441$.

Values shown are rounded; calculations use full precision.

This image gets its own denominator: add the exponential weights across its row.

Every image row now sums to 1

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › **Temperature** › **Rows** › Columns › Loss › Update

P · row shares

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.5398 $I_1 \cdot T_1$	0.2875 $I_1 \cdot T_2$	0.1727 $I_1 \cdot T_3$
Image I_2 dog	0.2740 $I_2 \cdot T_1$	0.5480 $I_2 \cdot T_2$	0.1780 $I_2 \cdot T_3$
Image I_3 car	0.1862 $I_3 \cdot T_1$	0.1837 $I_3 \cdot T_2$	0.6301 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

One softmax per image row



Image I_1 $0.5398 + 0.2875 + 0.1727 \approx 1$



Image I_2 $0.2740 + 0.5480 + 0.1780 \approx 1$



Image I_3 $0.1862 + 0.1837 + 0.6301 \approx 1$

$$P_{ij} = \frac{e^{S_{ij}}}{\sum_{k=1}^3 e^{S_{ik}}}$$

Keep this completed matrix P. We will use its paired entries for the image → text loss.

Keep P. Return to the original scores S.

CHOSEN 3D EXAMPLE · C = cosine similarities · $S = C / \tau$ · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > **Columns** > Loss > Update

Keep P for the image → text loss

P · row shares

	TEXT COLUMNS →		
	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.5398 $I_1 \cdot T_1$	0.2875 $I_1 \cdot T_2$	0.1727 $I_1 \cdot T_3$
Image I_2 dog	0.2740 $I_2 \cdot T_1$	0.5480 $I_2 \cdot T_2$	0.1780 $I_2 \cdot T_3$
Image I_3 car	0.1862 $I_3 \cdot T_1$	0.1837 $I_3 \cdot T_2$	0.6301 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

$S \rightarrow$ row softmax $\rightarrow P$ ✓ complete

Return to the same scores: $S = C / \tau$

S · logits

	TEXT COLUMNS →		
	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	1.9386 $I_1 \cdot T_1$	1.3087 $I_1 \cdot T_2$	0.7993 $I_1 \cdot T_3$
Image I_2 dog	1.2182 $I_2 \cdot T_1$	1.9112 $I_2 \cdot T_2$	0.7867 $I_2 \cdot T_3$
Image I_3 car	0.7302 $I_3 \cdot T_1$	0.7163 $I_3 \cdot T_2$	1.9490 $I_3 \cdot T_3$

ONE IMAGE PER ROW · THREE TEXT CANDIDATES

$S \rightarrow$ column softmax $\rightarrow Q$ next

Both softmax calculations start from $S = C / \tau$. Q will supply the text → image loss.

Back at S: which image belongs to the cat caption?

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > **Columns** > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

S · logits

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	1.939 correct pair	1.309 I ₁ · T ₂	0.799 I ₁ · T ₃
Image I ₂ dog	1.218 I ₂ · T ₁	1.911 correct pair	0.787 I ₂ · T ₃
Image I ₃ car	0.730 I ₃ · T ₁	0.716 I ₃ · T ₂	1.949 correct pair

DIAGONAL = PAIRED PARTNERS

Read column 1 of S ↓

	I ₁ cat	I ₂ dog	I ₃ car
Cosine	0.9693	0.6091	0.3651
÷ 0.5 → logit	1.9386	1.2182	0.7302

$$Q_{i1} = \frac{e^{S_{i1}}}{\sum_{k=1}^3 e^{S_{k1}}}$$

P stays unchanged. Start from S again, fix caption T₁, and compare the three images down its column.

Add the weights down this column

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > **Columns** > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

S · logits

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	1.939 correct pair	1.309 $I_1 \cdot T_2$	0.799 $I_1 \cdot T_3$
Image I ₂ dog	1.218 $I_2 \cdot T_1$	1.911 correct pair	0.787 $I_2 \cdot T_3$
Image I ₃ car	0.730 $I_3 \cdot T_1$	0.716 $I_3 \cdot T_2$	1.949 correct pair

DIAGONAL = PAIRED PARTNERS

A different set of competitors

	I ₁ cat	I ₂ dog	I ₃ car
Cosine	0.9693	0.6091	0.3651
÷ 0.5 → logit	1.9386	1.2182	0.7302
exp(logit)	6.9487	3.3812	2.0754

Sum this column ↓
 $6.9487 + 3.3812 + 2.0754$
= 12.405358

The cat-cat numerator is unchanged. The column denominator includes different cells.

The cat caption assigns the cat image a share

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > **Columns** > Loss > Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

$Q \cdot$ column shares TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.560 correct pair	0.296 $I_1 \cdot T_2$	0.194 $I_1 \cdot T_3$
Image I_2 dog	0.273 $I_2 \cdot T_1$	0.540 correct pair	0.192 $I_2 \cdot T_3$
Image I_3 car	0.167 $I_3 \cdot T_1$	0.164 $I_3 \cdot T_2$	0.614 correct pair

DIAGONAL = PAIRED PARTNERS

Follow $I_1 \times T_1$

$$Q_{11} = \frac{e^{S_{11}}}{\sum_k e^{S_{k1}}}$$

$$= \frac{6.948746}{12.405358}$$

Share

0.560141

Divide by the column sum because the candidate images compete for one caption.

The cat caption assigns the dog image a share

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > **Columns** > Loss > Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

$Q \cdot$ column shares

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.560 correct pair	0.296 $I_1 \cdot T_2$	0.194 $I_1 \cdot T_3$
Image I_2 dog	0.273 $I_2 \cdot T_1$	0.540 correct pair	0.192 $I_2 \cdot T_3$
Image I_3 car	0.167 $I_3 \cdot T_1$	0.164 $I_3 \cdot T_2$	0.614 correct pair

DIAGONAL = PAIRED PARTNERS

Follow $I_2 \times T_1$

$$Q_{21} = \frac{e^{S_{21}}}{\sum_k e^{S_{k1}}}$$

$$= \frac{3.381178}{12.405358}$$

Share

0.272558

Divide by the column sum because the candidate images compete for one caption.

The cat caption assigns the car image a share

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › **Columns** › Loss › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

$Q \cdot$ column shares TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.560 correct pair	0.296 $I_1 \cdot T_2$	0.194 $I_1 \cdot T_3$
Image I_2 dog	0.273 $I_2 \cdot T_1$	0.540 correct pair	0.192 $I_2 \cdot T_3$
Image I_3 car	0.167 $I_3 \cdot T_1$	0.164 $I_3 \cdot T_2$	0.614 correct pair

DIAGONAL = PAIRED PARTNERS

Follow $I_3 \times T_1$

$$Q_{31} = \frac{e^{S_{31}}}{\sum_k e^{S_{k1}}}$$

$$= \frac{2.075434}{12.405358}$$

Share

0.167301

Divide by the column sum because the candidate images compete for one caption.

Now let the dog caption choose its image

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > **Columns** > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂

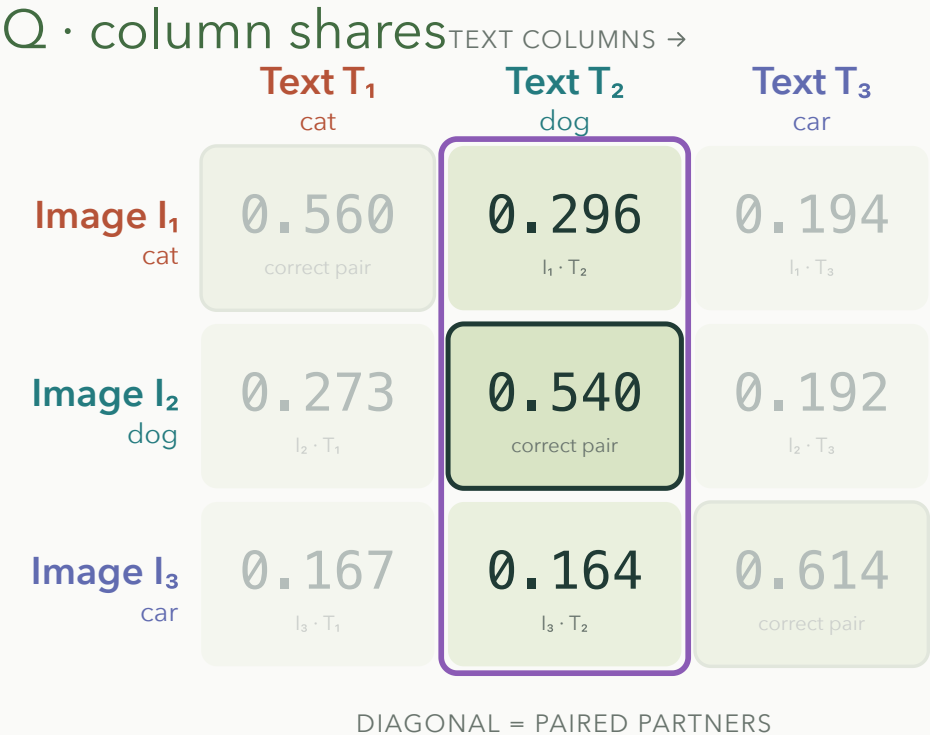


Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"



Column 2 ↓

	I₁ cat	I₂ dog	I₃ car
÷ 0.5 → logit	1.3087	1.9112	0.7163
exp(logit)	3.7013	6.7612	2.0468
÷ sum → share	0.2959	0.5405	0.1636

Sum this column ↓

$3.7013 + 6.7612 + 2.0468$

= 12.509366

Every caption gets its own denominator, computed down its column.

Now let the car caption choose its image

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > **Columns** > Loss > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

Q · column shares

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.560 correct pair	0.296 I₁ · T₂	0.194 I₁ · T₃
Image I ₂ dog	0.273 I₂ · T₁	0.540 correct pair	0.192 I₂ · T₃
Image I ₃ car	0.167 I₃ · T₁	0.164 I₃ · T₂	0.614 correct pair

DIAGONAL = PAIRED PARTNERS

Column 3 ↓

	I ₁ cat	I ₂ dog	I ₃ car
÷ 0.5 → logit	0.7993	0.7867	1.9490
exp(logit)	2.2239	2.1961	7.0218
÷ sum → share	0.1944	0.1919	0.6137

Sum this column ↓

2.2239 + 2.1961 + 7.0218

= **11.441806**

Every caption gets its own denominator, computed down its column.

Each column now sums to 1

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › **Columns** › Loss › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

Q · column shares TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.560 correct pair $I_1 \cdot T_1$	0.296 $I_1 \cdot T_2$	0.194 $I_1 \cdot T_3$
Image I_2 dog	0.273 $I_2 \cdot T_1$	0.540 correct pair $I_2 \cdot T_2$	0.192 $I_2 \cdot T_3$
Image I_3 car	0.167 $I_3 \cdot T_1$	0.164 $I_3 \cdot T_2$	0.614 correct pair $I_3 \cdot T_3$

DIAGONAL = PAIRED PARTNERS

Text → image

$$Q_{ij} = \frac{e^{S_{ij}}}{\sum_k e^{S_{kj}}}$$

P is kept for the image → text loss.

Q is ready for the text → image loss.

We now have two probability matrices, both computed from the same scores S . We will average their two losses.

Same pair and numerator. Different competition.

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › **Columns** › Loss › Update

Cat image → caption

$$P_{11} = \frac{6.948746}{12.874005} = 0.539750$$

Competes with the dog and car captions.

Cat caption → image

$$Q_{11} = \frac{6.948746}{12.405358} = 0.560141$$

Competes with the dog and car images.

We train both retrieval directions.

The recorded pair tells us which answer is correct

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

P · row shares

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.540 correct pair	0.288 $I_1 \cdot T_2$	0.173 $I_1 \cdot T_3$
Image I_2 dog	0.274 $I_2 \cdot T_1$	0.548 correct pair	0.178 $I_2 \cdot T_3$
Image I_3 car	0.186 $I_3 \cdot T_1$	0.184 $I_3 \cdot T_2$	0.630 correct pair

DIAGONAL = PAIRED PARTNERS

Target for the cat image

$[1, 0, 0]$

Caption 1 is the observed partner.

Its share

0.539750

A loss should be small when this share is close to 1.

The target comes from the data record, not from whichever cell currently has the largest score.

A small share for the partner should cost more

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

$$p = 0.9$$

$-\ln(p)$

0.105

$$p = 0.333333$$

$-\ln(p)$

1.099

$$p = 0.01$$

$-\ln(p)$

4.605

$$\ell = -\ln p$$

As the paired share approaches 1, its penalty approaches 0. All logs here are natural.

Cross-entropy reads the cat image’s row of P

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update



Image I_1 · cat

Bring back row 1 of P: this image’s three caption probabilities.

Same caption in each column ↓	Text T_1 “a photo of a cat”	Text T_2 “a photo of a dog”	Text T_3 “a photo of a car”
Row 1 of P $p_j = P_{1j}$	0.539750	0.287505	0.172745
Target y	1 Recorded partner	0 Other caption	0 Other caption
Each loss term $-y_j \ln p_j$	$-1 \ln(0.539750)$	$-0 \ln(0.287505)$	$-0 \ln(0.172745)$

$$CE(\mathbf{y}, \mathbf{p}) = - \sum_{j=1}^3 y_j \ln p_j$$

Add the three terms:
 $- \ln(0.539750) + 0 + 0 \approx 0.616649$

The target $[1, 0, 0]$ selects $-\ln(P_{11})$. All three captions influenced P_{11} through softmax.

Cat image: read its paired cell

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > Columns > **Loss** > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

P · row shares

TEXT COLUMNS →

	Text T₁ cat	Text T₂ dog	Text T₃ car
Image I₁ cat	0.540 correct pair	0.288 $I_1 \cdot T_2$	0.173 $I_1 \cdot T_3$
Image I₂ dog	0.274 $I_2 \cdot T_1$	0.548 correct pair	0.178 $I_2 \cdot T_3$
Image I₃ car	0.186 $I_3 \cdot T_1$	0.184 $I_3 \cdot T_2$	0.630 correct pair

DIAGONAL = PAIRED PARTNERS

One correct-match penalty

$$\begin{aligned}\ell_1 &= -\ln P_{11} \\ &= -\ln(0.539750)\end{aligned}$$

Penalty **0.616649**

A smaller penalty means more share went to the observed partner.

Dog image: read its paired cell

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

$P \cdot$ row shares

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.540 correct pair $I_1 \cdot T_1$	0.288 $I_1 \cdot T_2$	0.173 $I_1 \cdot T_3$
Image I_2 dog	0.274 $I_2 \cdot T_1$	0.548 correct pair $I_2 \cdot T_2$	0.178 $I_2 \cdot T_3$
Image I_3 car	0.186 $I_3 \cdot T_1$	0.184 $I_3 \cdot T_2$	0.630 correct pair $I_3 \cdot T_3$

DIAGONAL = PAIRED PARTNERS

One correct-match penalty

$$\begin{aligned} \ell_2 &= -\ln P_{22} \\ &= -\ln(0.547977) \end{aligned}$$

Penalty

0.601521

A smaller penalty means more share went to the observed partner.

Car image: read its paired cell

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > Columns > **Loss** > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

P · row shares

TEXT COLUMNS →

	Text T₁ cat	Text T₂ dog	Text T₃ car
Image I₁ cat	0.540 correct pair	0.288 $I_1 \cdot T_2$	0.173 $I_1 \cdot T_3$
Image I₂ dog	0.274 $I_2 \cdot T_1$	0.548 correct pair	0.178 $I_2 \cdot T_3$
Image I₃ car	0.186 $I_3 \cdot T_1$	0.184 $I_3 \cdot T_2$	0.630 correct pair

DIAGONAL = PAIRED PARTNERS

One correct-match penalty

$$\begin{aligned} \ell_3 &= -\ln P_{33} \\ &= -\ln(0.630093) \end{aligned}$$

Penalty **0.461888**

A smaller penalty means more share went to the observed partner.

Average the three image → text penalties

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update



cat

0.616649



dog

0.601521



car

0.461888

$$\mathcal{L}_{I \rightarrow T} = \frac{0.616649 + 0.601521 + 0.461888}{3} = 0.560020$$

Three prediction tasks contribute to this mean. We do not average nine cells.

Cat caption: read its paired cell

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

Q · column shares TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.560 correct pair	0.296 $I_1 \cdot T_2$	0.194 $I_1 \cdot T_3$
Image I_2 dog	0.273 $I_2 \cdot T_1$	0.540 correct pair	0.192 $I_2 \cdot T_3$
Image I_3 car	0.167 $I_3 \cdot T_1$	0.164 $I_3 \cdot T_2$	0.614 correct pair

DIAGONAL = PAIRED PARTNERS

One correct-match penalty

$$\begin{aligned}\ell_1 &= -\ln Q_{11} \\ &= -\ln(0.560141)\end{aligned}$$

Penalty

0.579567

A smaller penalty means more share went to the observed partner.

Dog caption: read its paired cell

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

Image I_1



Text T_1

"a photo of a cat"

Image I_2



Text T_2

"a photo of a dog"

Image I_3



Text T_3

"a photo of a car"

Q · column shares

TEXT COLUMNS →

	Text T_1 cat	Text T_2 dog	Text T_3 car
Image I_1 cat	0.560 correct pair	0.296 $I_1 \cdot T_2$	0.194 $I_1 \cdot T_3$
Image I_2 dog	0.273 $I_2 \cdot T_1$	0.540 correct pair	0.192 $I_2 \cdot T_3$
Image I_3 car	0.167 $I_3 \cdot T_1$	0.164 $I_3 \cdot T_2$	0.614 correct pair

DIAGONAL = PAIRED PARTNERS

One correct-match penalty

$$\begin{aligned}\ell_2 &= -\ln Q_{22} \\ &= -\ln(0.540489)\end{aligned}$$

Penalty

0.615280

A smaller penalty means more share went to the observed partner.

Car caption: read its paired cell

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > Columns > **Loss** > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

Q · column shares

TEXT COLUMNS →

	Text T ₁ cat	Text T ₂ dog	Text T ₃ car
Image I ₁ cat	0.560 correct pair I₁ · T₁	0.296 I₁ · T₂	0.194 I₁ · T₃
Image I ₂ dog	0.273 I₂ · T₁	0.540 correct pair I₂ · T₂	0.192 I₂ · T₃
Image I ₃ car	0.167 I₃ · T₁	0.164 I₃ · T₂	0.614 correct pair I₃ · T₃

DIAGONAL = PAIRED PARTNERS

One correct-match penalty

$$\ell_3 = -\ln Q_{33}$$
$$= -\ln(0.613698)$$

Penalty **0.488252**

A smaller penalty means more share went to the observed partner.

Average the three text → image penalties

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update



cat

0.579567



dog

0.615280



car

0.488252

$$\mathcal{L}_{T \rightarrow I} = \frac{0.579567 + 0.615280 + 0.488252}{3} = 0.561033$$

Three prediction tasks contribute to this mean. We do not average nine cells.

Give both retrieval directions equal weight

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

Image → text

0.560020

Text → image

0.561033

$$\mathcal{L} = \frac{\mathcal{L}_{I \rightarrow T} + \mathcal{L}_{T \rightarrow I}}{2}$$

This batch's CLIP loss

0.560526

The final value averages all six correct-match penalties.

What does “no preference” look like for one image?

HYPOTHETICAL BASELINE · three candidates · equal logits chosen as 0 · natural logs

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update



Image I₁ · cat

Suppose the model gives all three captions the same score.

	Text T ₁ “a photo of a cat” Recorded partner	Text T ₂ “a photo of a dog” Other caption	Text T ₃ “a photo of a car” Other caption
Same image in every cell			
Equal logits	0	0	0
Equal weights	$e^0 = 1$	$e^0 = 1$	$e^0 = 1$
Divide by 1 + 1 + 1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Loss for the recorded partner T₁

$$\ell_1 = -\ln P_{11} = -\ln(1/3) \approx 1.098612$$

The correct caption gets only one third of the probability, just like either competitor.

If every choice is a tie, averaging keeps the same loss

HYPOTHETICAL BASELINE · three paired records · both retrieval directions tie

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

Now suppose **every row and every column** gives equal shares.

Correct-match penalties		 Cat	 Dog	 Car
Image → text Each paired share = ⅓	$I_1 \rightarrow T_1$	1.098612	$I_2 \rightarrow T_2$	$I_3 \rightarrow T_3$
			1.098612	1.098612
Text → image Each paired share = ⅓	$T_1 \rightarrow I_1$	1.098612	$T_2 \rightarrow I_2$	$T_3 \rightarrow I_3$
			1.098612	1.098612

3 image penalties + 3 text penalties; average all six.

$$\mathcal{L}_{\text{tie}} = \frac{3 \ln 3 + 3 \ln 3}{6} = \ln 3 \approx 1.098612$$

Six equal penalties, averaged: the batch baseline is 1.098612.

Now 0.5605 has something to compare with

READING THE LOSS · one recorded partner per image and caption · natural logs

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

Ideal limit

→ 0

Every paired answer gets a share approaching 1.

Our cat / dog / car batch

0.560526

The paired answers do better than an equal split.

Equal-choice baseline

1.098612

Every candidate gets $\frac{1}{3}$. The model has no preference.

Lower is better. A confidently wrong prediction can cost more than this baseline.

$$N \text{ pairs: } p_{\text{paired}} = 1/N \Rightarrow \mathcal{L}_{\text{tie}} = \ln N$$

For the same three-pair batch: $0.560526 < 1.098612$. The loss rewards putting more share on the paired answers.

This is the whole calculation we just built

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › **Loss** › Update

$$\mathbf{C} = \mathbf{U}\mathbf{V}^\top, \quad \mathbf{S} = \mathbf{C}/\tau$$

$$P_{ij} = \frac{e^{S_{ij}}}{\sum_k e^{S_{ik}}}, \quad Q_{ij} = \frac{e^{S_{ij}}}{\sum_k e^{S_{kj}}}$$

$$\mathcal{L} = -\frac{1}{2N} \left(\sum_i \ln P_{ii} + \sum_j \ln Q_{jj} \right)$$

Normalize vectors, compare all pairs, normalize each direction, then penalize the recorded partners.

Trace another cell in the interactive

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs > Encode > Normalize > Dot products > Temperature > Rows > Columns > **Loss** > Update

Image I₁



Text T₁
"a photo of a cat"

Image I₂



Text T₂
"a photo of a dog"

Image I₃



Text T₃
"a photo of a car"

P · row shares

TEXT COLUMNS →

	Text T₁ cat	Text T₂ dog	Text T₃ car
Image I₁ cat	0.540 correct pair	0.288 $I_1 \cdot T_2$	0.173 $I_1 \cdot T_3$
Image I₂ dog	0.274 $I_2 \cdot T_1$	0.548 correct pair	0.178 $I_2 \cdot T_3$
Image I₃ car	0.186 $I_3 \cdot T_1$	0.184 $I_3 \cdot T_2$	0.630 correct pair

DIAGONAL = PAIRED PARTNERS

The dog caption competes for the cat image

	T₁ cat	T₂ dog	T₃ car
÷ 0.5 → logit	1.9386	1.3087	0.7993
exp(logit)	6.9487	3.7013	2.2239
÷ sum → share	0.5398	0.2875	0.1727

Select a cell, switch row/column, or change temperature. The examples and numbers are the same.

The loss tells us how to change the two branches

REAL CLIP · gradients pass through both branches; the logit scale is also learned

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**



← forward computation · backpropagation follows the arrows above →

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}$$

| The optimizer updates the shared parameters that produced the vectors.

Inspect one gradient update in the toy

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**

Update the cat's raw image vector

Before: `[2.000000, 0.500000, 0.200000]`

Gradient: `[-0.017257, 0.050730, 0.045746]`

$$\mathbf{z}_{\text{new}} = \mathbf{z} - 0.25 \nabla_{\mathbf{z}} \mathcal{L}$$

After: `[2.004314, 0.487317, 0.188564]`

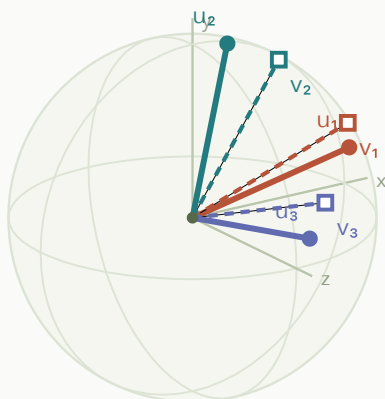
The toy updates all six raw vectors through normalization. Real CLIP instead updates encoder and projection weights.

One step: compare the same batch before and after

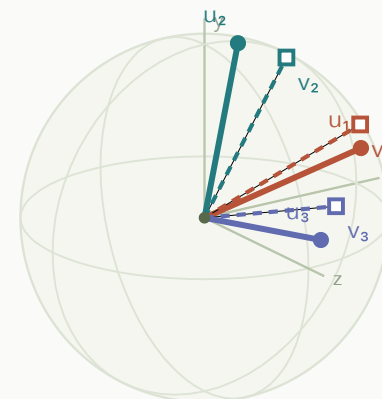
COMPUTED TOY UPDATE · six trainable raw vectors; no neural encoder is trained

Pairs > Encode > Normalize > Dot products > Temperature > Rows > Columns > Loss > **Update**

Before



After one step



● solid = image □ dashed = text · colour = pair

Same batch · $\tau = 0.5$ · $\eta = 0.25$

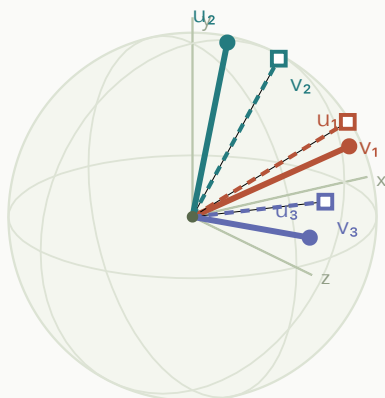
0.560526 → 0.539743

The first step makes the paired choices easier. The movement is small; more updates accumulate.

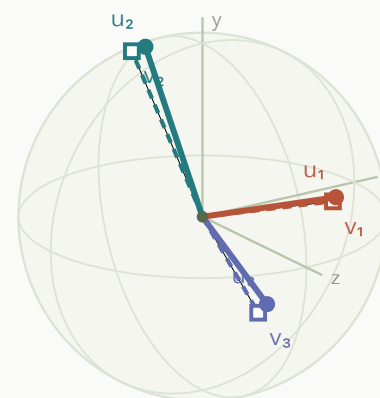
Repeat 100 updates on this batch

COMPUTED TOY EXPERIMENT · fixed cat / dog / car batch · $\eta = 0.25$ Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**

Starting directions



After 100 updates



● solid = image □ dashed = text · colour = pair

Same three pairs · fixed temperature

0.560526 → 0.124937

Paired directions come closer and become easier to distinguish from competing pairs.

Now sample the next three records

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**



Bicycle
"a bicycle"



Airplane
"an airplane"



Tree
"a tree"

Batch 2: **bicycle · airplane · tree**

| The row numbers now refer to these records. Build a fresh 3×3 matrix and repeat the calculation.

New pairs, new scores, the same loss calculation

CHOSEN 3D EXAMPLE · batch 2: bicycle / airplane / tree · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**

Image I_1



Text T_1
"a bicycle"

Image I_2



Text T_2
"an airplane"

Image I_3



Text T_3
"a tree"

C · cosine

	TEXT CANDIDATES		
	T_1	T_2	T_3
I_1	0.953 correct pair	-0.065 $I_1 \cdot T_2$	-0.406 $I_1 \cdot T_3$
I_2	0.006 $I_2 \cdot T_1$	0.942 correct pair	0.928 $I_2 \cdot T_3$
I_3	-0.384 $I_3 \cdot T_1$	0.501 $I_3 \cdot T_2$	0.921 correct pair

DIAGONAL = PAIRED PARTNERS

Batch 2: before an update

Image → text

0.447912

Text → image

0.456861

$$\mathcal{L} = \frac{0.447912 + 0.456861}{2}$$

Batch loss

0.452386

Image 1 is now bicycle, and text 1 is "a bicycle". The target is still the diagonal.

Compare batch 2 with itself after one step

COMPUTED TOY UPDATE · batch 2 · $\tau = 0.5$ · $\eta = 0.25$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**



Bicycle
"a bicycle"



Airplane
"an airplane"



Tree
"a tree"

Batch 2 · same pairs before and after

0.452386 → 0.442814

Do not compare a new batch's loss with the old batch's loss and call that progress.

One pass through the dataset is an epoch

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**

Batch 1



cat · dog · car

Batch 2



bicycle · airplane · tree

Batch 3



shoe · mug · banana

Batch 4



chair · building · bird

After all 12 records have appeared, shuffle the pairs and begin the next epoch.

Changing the batch changes the competitors. Keep the learned parameters; do not restart the model.

Watch the next batch learn

CHOSEN 3D EXAMPLE · same cat / dog / car batch as the interactive · $\tau = 0.5$

Pairs › Encode › Normalize › Dot products › Temperature › Rows › Columns › Loss › **Update**



Bicycle

"a bicycle"



Airplane

"an airplane"



Tree

"a tree"

One training step

1. Sample complete pairs.
2. Encode and normalize.
3. Compare every image and caption.
4. Average both cross-entropies.
5. Backpropagate, update, repeat.

Compare before and after on the same batch. A different batch can have a different loss.

SECTION 06

How web-scale training works

How does the same recipe scale from 3 pairs to hundreds of millions?

The dataset stores pairs, not a giant score matrix

SCHEMATIC · hypothetical teaching dataset: 4 million complete pair records

record 1: image_001 ↔ associated text_001

record 2: image_002 ↔ associated text_002

...

record 4,000,000: image_M ↔ associated text_M

A caption may be a title, description or other associated web text. It need not be a clean class label.

Sample four complete records for this step

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



record 1

"a photo of a Newfoundland"



record 2

"a photo of a Persian cat"



record 3

"a photo of a pug"



record 4

"a cup of coffee"

Apply image preprocessing and tokenize each paired text. Keep their pair order together.

Run the same recipe on the four sampled records

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise



| Then sample the next mini-batch. Each step creates new in-batch competition.

Dataset size and batch size are different quantities

SCHEMATIC · CLIP ViT-B/32 unless stated otherwise

400M pair records



Sample B pairs



$B \times B$ scores

$$4^2 = 16, \quad 1024^2 = 1,048,576$$

| 400 million records does not mean a 400M × 400M score matrix.

Original CLIP: this objective at web scale

HISTORICAL · original CLIP, 2021



Radford et al. · ICML 2021

The published recipe learns both branches from naturally associated image-text data.

~400 million image-text pairs

32,768 candidate texts per image in the reported
contrastive batch

ResNet or ViT image encoders; a text Transformer

Two captions can describe the same image

TEACHING EXAMPLE · supplied captions for the same toy dog



Image I_1 · the same dog

T1 "a dog" **Fits this image**

T2 "a dog with floppy ears" **Fits this image**

T3 "a cat" **Does not fit**

| A positive is an image-text pair we know should match. This dog has two here.

The one-pair target can miss a valid caption

TEACHING EXAMPLE · supplied captions for the same toy dog



Image I_1 · the same dog

For image I_1 , compare these three candidate texts

Supplied text	Fits I_1 ?	Target weight
T1 "a dog"	Yes	1
T2 "a dog with floppy ears"	Yes	0
T3 "a cat"	No	0

| T_2 describes this dog too, but the target $[1, 0, 0]$ makes it a competitor.

Option 1: choose one valid caption each time

TRAINING OPTION · extends the one-pair-per-image setup we just used

We have saved both dog captions with this image.

This visit to the image



↔ "a dog"

Same image I_1

One image + one selected caption

A later visit to the image



↔ "a dog with floppy ears"

Same image I_1

One image + one selected caption

In this option, the other caption for I_1 stays out of that batch.

Keep the same one-pair loss. Across visits, the model can learn from both captions.

Option 2: include both and give both credit

ONE MULTIPLE-POSITIVE LOSS · image-to-text direction shown



Image I_1 · the same dog

For image I_1 , compare these three candidate texts

Supplied text	Fits I_1 ?	Target weight
T1 "a dog"	Yes	$\frac{1}{2}$
T2 "a dog with floppy ears"	Yes	$\frac{1}{2}$
T3 "a cat"	No	0

Share the target equally: $\mathbf{y} = [\frac{1}{2}, \frac{1}{2}, \mathbf{0}]$. Average the two caption penalties.

$$\ell_{I_1} = -\frac{1}{2} \ln p(T_1 | I_1) - \frac{1}{2} \ln p(T_2 | I_1)$$

The $\frac{1}{2}$ values are target weights we set, not probabilities predicted by the model.

SECTION 07

Return to the opening question

We can now explain the hat experiment, one operation at a time.

How did we calculate what changed?

SAVED CLIP MEASUREMENTS · original hat pair · ViT-B/32 · 512D · rounded previews · Node CPU q8



Before: no hat



After: hat added

"hat"

"boat"

"cup"

"cat"

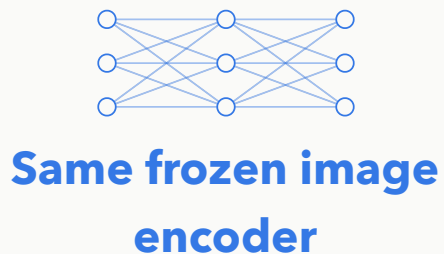
| Which vectors would you compute, and what would you compare?

Encode both images with the same model

SAVED CLIP MEASUREMENTS · original hat pair · ViT-B/32 · 512D · rounded previews · Node CPU q8



Before



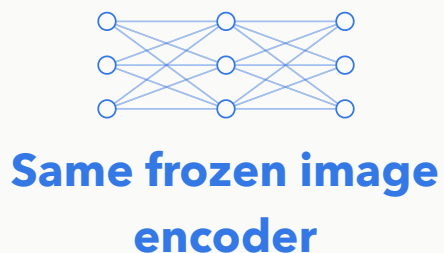
b $[+0.051608, -0.018763, -0.043882, \dots]$

First 3 of 512 coordinates

Projection included · divide by length



After



a $[+0.066042, -0.016296, -0.049582, \dots]$

First 3 of 512 coordinates

Projection included · divide by length

a and b are unit image vectors. Their 512 coordinates use the same learned space.

Subtract before from after, coordinate by coordinate

SAVED CLIP MEASUREMENTS · original hat pair · ViT-B/32 · 512D · rounded previews · Node CPU q8

$$\delta = a - b$$

Coordinate	After: a	Before: b	$\delta = a - b$
1	+0.066042	+0.051608	+0.014434
2	-0.016296	-0.018763	+0.002468
3	-0.049582	-0.043882	-0.005700

Continue the same subtraction through coordinate 512.

| The subtraction happens between image vectors. The pixels and model weights stay as they are.

The difference needs its own normalization

SAVED CLIP MEASUREMENTS · original hat pair · ViT-B/32 · 512D · rounded previews · Node CPU q8

$$\|\delta\|_2 = \sqrt{\delta_1^2 + \delta_2^2 + \dots + \delta_{512}^2} \approx 0.483430$$

$$\mathbf{d} = \frac{\delta}{\|\delta\|_2}$$

+0.014434 ÷ 0.483430

+0.029858

+0.002468 ÷ 0.483430

+0.005104

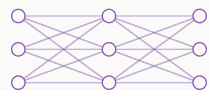
-0.005700 ÷ 0.483430

-0.011791

| a and b each have length 1. Their difference usually does not; d does.

Encode the exact words we offered at the start

SAVED CLIP MEASUREMENTS · original hat pair · ViT-B/32 · 512D · rounded previews · Node CPU q8



**Same frozen text
encoder**

Tokenize each word → encode + project → divide by length

"hat" → [+0.013264, +0.001263, -0.016369, ...] $\mathbf{V}_{\text{hat}}$

"boat" → [+0.012867, -0.004047, +0.009648, ...] $\mathbf{V}_{\text{boat}}$

"cup" → [+0.005676, -0.001876, -0.014071, ...] $\mathbf{V}_{\text{cup}}$

"cat" → [+0.020644, -0.024952, -0.021981, ...] $\mathbf{V}_{\text{cat}}$

Each candidate becomes a unit text vector with 512 coordinates.

Work out the score for “hat”

SAVED CLIP MEASUREMENTS · original hat pair · ViT-B/32 · 512D · rounded previews · Node CPU q8

$$s_{\text{hat}} = \mathbf{d}^\top \mathbf{v}_{\text{hat}} = \sum_{k=1}^{512} d_k v_{\text{hat},k}$$

$$(+0.029858) \times (+0.013264) \qquad +0.00039603$$

$$(+0.005104) \times (+0.001263) \qquad +0.00000645$$

$$(-0.011791) \times (-0.016369) \qquad +0.00019300$$

$$\text{Sum of the remaining 509 products} \qquad +0.08016226$$

$$s_{\text{hat}} \approx 0.080758$$

Both vectors have length 1, so this dot product is a cosine.

Now repeat the same calculation for all four words

SAVED CLIP MEASUREMENTS · original hat pair · ViT-B/32 · 512D · rounded previews · Node CPU q8

Candidate word	Before · word	After · word	Change cosine
"hat"	0.211620	0.250661	+0.080758
"boat"	0.202544	0.214823	+0.025401
"cup"	0.203043	0.209409	+0.013169
"cat"	0.192427	0.193662	+0.002556

$$\mathbf{d}^\top \mathbf{v} = \frac{\mathbf{a}^\top \mathbf{v} - \mathbf{b}^\top \mathbf{v}}{\|\mathbf{a} - \mathbf{b}\|_2}$$

For "hat": (0.250661 – 0.211620) ÷ 0.483430 ≈ 0.080758

“Hat” wins this candidate list. The score measures alignment with the change direction.

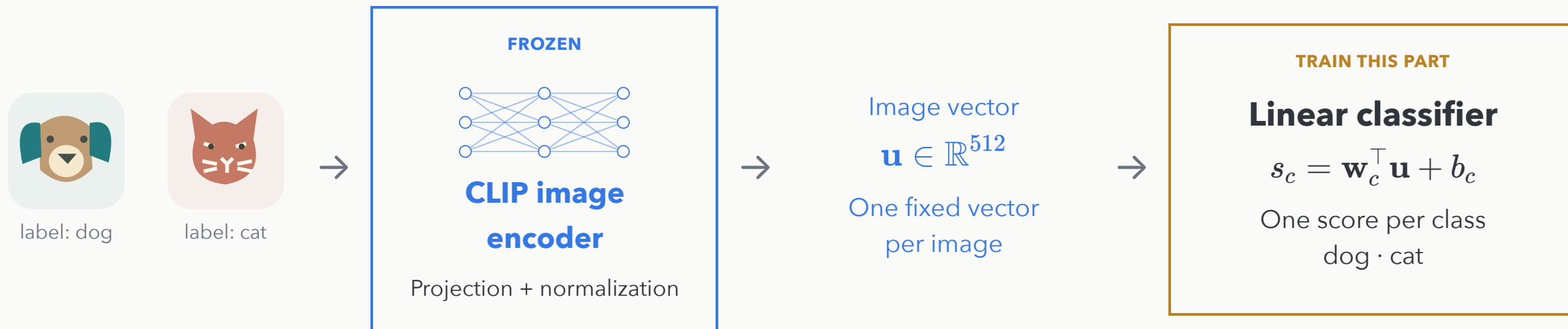
SECTION 08

Linear probing

Can we reuse the learned image features for a labelled task?

Linear probing: train only the new classifier

SCHEMATIC · after CLIP pretraining · labelled task examples · no measured probe result



Known label $\rightarrow$ cross-entropy $\rightarrow$ update only $\mathbf{W}$ and $\mathbf{b}$

“Linear”: each score is a weighted sum of $\mathbf{u}$ ’s coordinates, plus a bias.

The classifier learns from labels. CLIP’s image encoder and projection stay fixed.

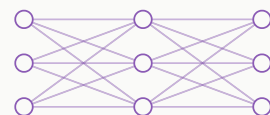
Two ways to classify with frozen CLIP

SCHEMATIC · same frozen image representation · different sources of class vectors

Zero-shot

"a photo of a dog"

"a photo of a cat"



**Frozen text
encoder**



$\mathbf{v}_{\text{dog}}, \mathbf{v}_{\text{cat}}$

Class vectors from words

No labelled task examples

Linear probe



label: dog



label: cat



**Frozen image vectors
+ known labels**

Fit a linear classifier



$\mathbf{w}_{\text{dog}}, \mathbf{w}_{\text{cat}}, \mathbf{b}$

Class weights learned from examples

Test on held-out images. Their labels must not be used to fit the classifier.

A linear probe asks: how useful are the image features for this labelled task?

The hat comparison in a few lines of PyTorch

OPENAI CLIP API · same method; saved lecture numbers use the Node q8 checkpoint

before and **after** are the two supplied PIL RGB images. Load the model once, then compare.

```
import torch, clip
import torch.nn.functional as F

model, preprocess = clip.load("ViT-B/32", device="cpu")
model.eval()
images = torch.stack([preprocess(before), preprocess(after)])
words = ["hat", "boat", "cup", "cat"]
with torch.no_grad():
    u = F.normalize(model.encode_image(images).float(), dim=-1)
    v = F.normalize(model.encode_text(clip.tokenize(words)).float(), dim=-1)
    delta = u[1] - u[0] # after minus before
    if delta.norm() < 1e-6:
        raise ValueError("No stable change direction")
    d = F.normalize(delta, dim=-1) # normalize again
    scores = v @ d # one cosine per word
print(sorted(zip(words, scores.tolist()), key=lambda x: -x[1]))
```

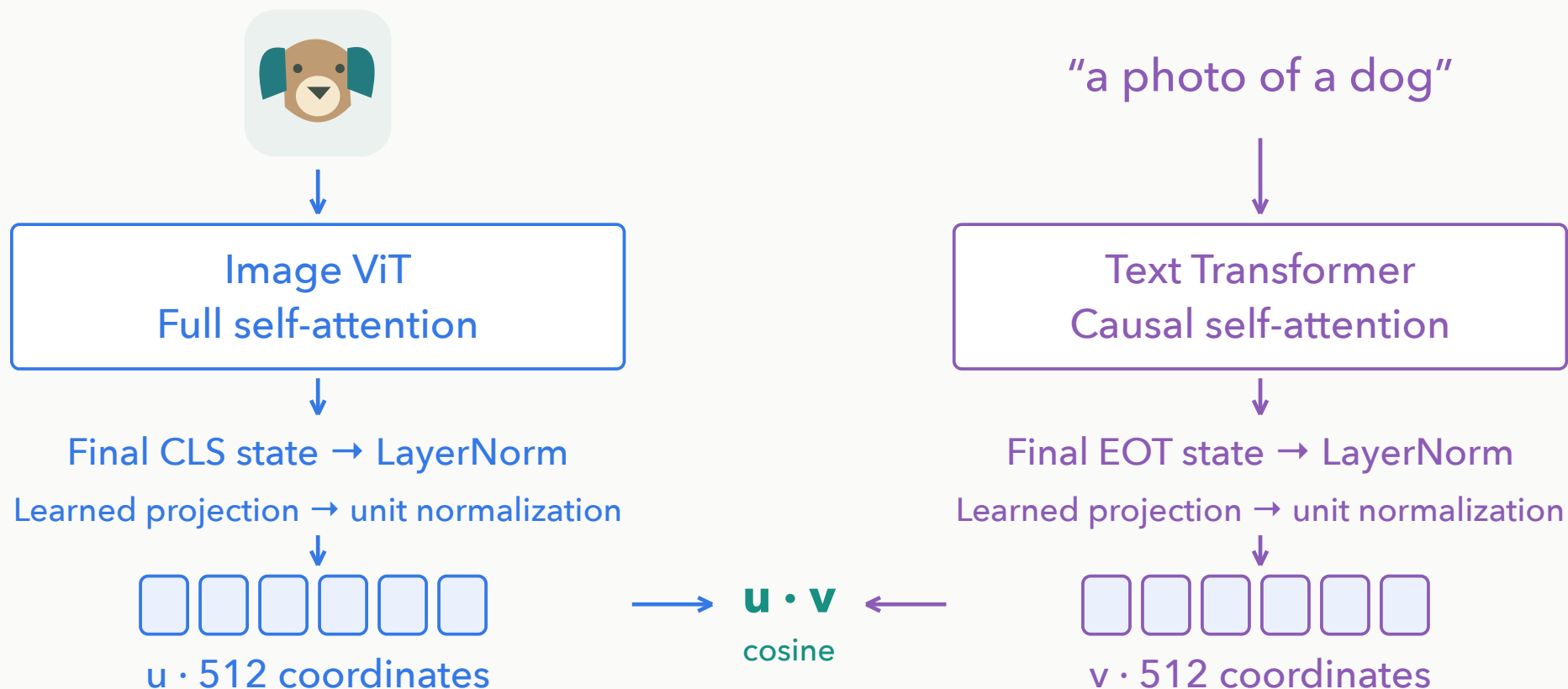
[Optional: tiny training experiment](#)

[Training notebook](#)

[Live CLIP playground](#)

Two encoders build a shared comparison space

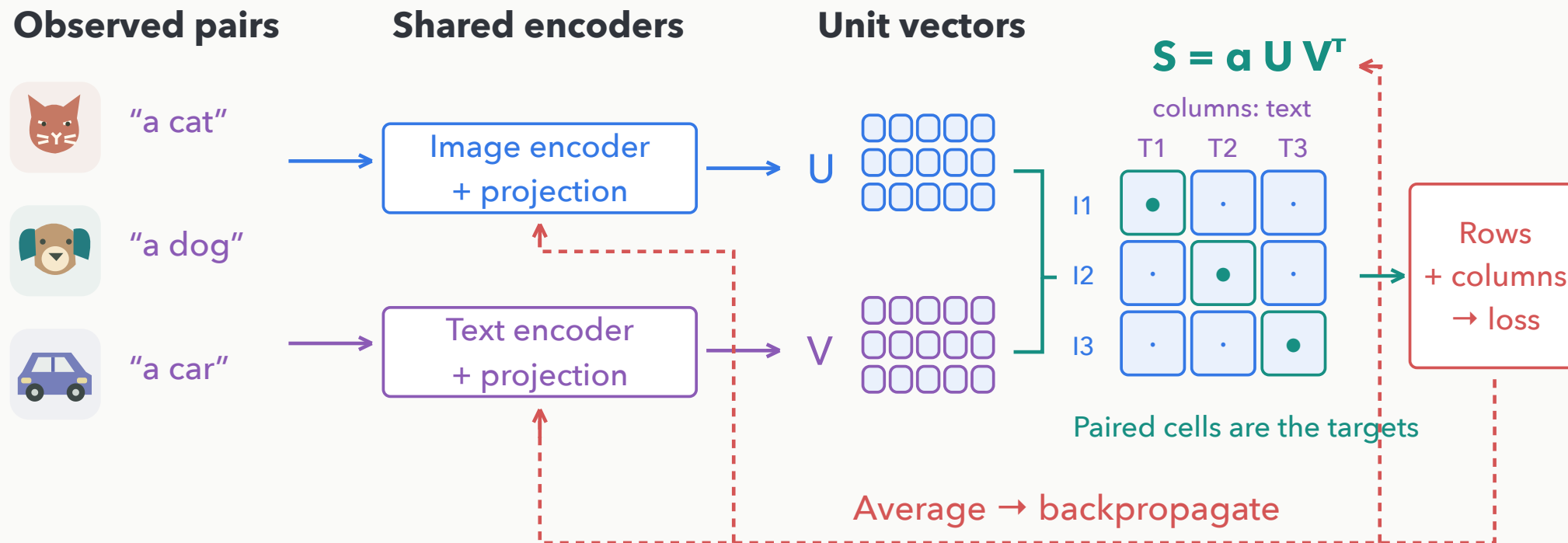
SUMMARY · architecture and geometry are schematic



Read one vector from each branch, project, normalize, then compare.

The loss teaches the two branches what should match

SUMMARY · architecture and geometry are schematic



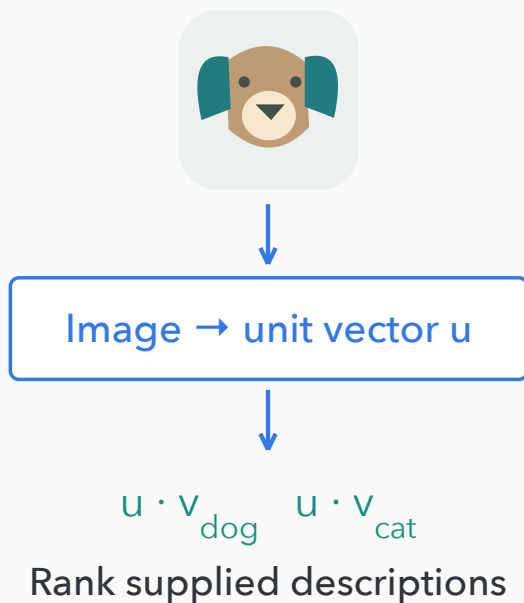
Update both encoders, both projections and the score scale.

Pairing supplies the targets. Gradients change the shared model parameters.

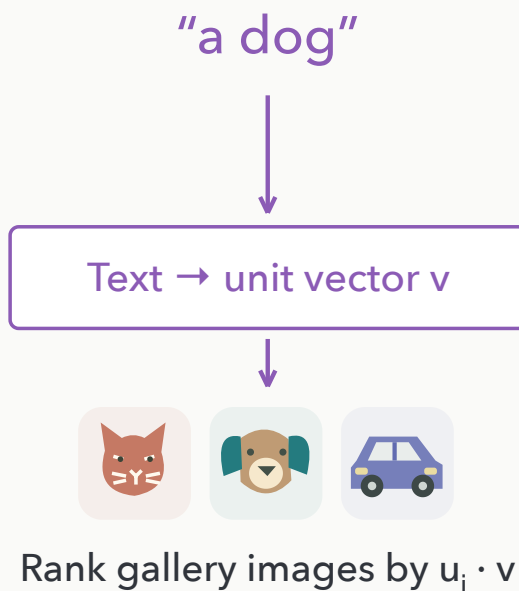
Once trained, the same vectors answer different questions

SUMMARY · architecture and geometry are schematic

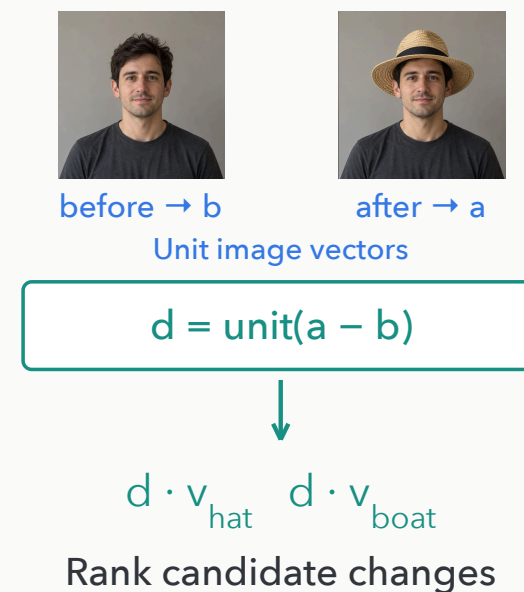
Name an image



Search with words



Describe a change



Reuse the frozen encoders. Choose which vectors to compare.

Choose the comparison that matches the question you want to ask.