DDA4220 · Fall 2026

Lecture 14

Deep Learning and Applications

Self-supervised learning

Zhen Liu

CUHK-Shenzhen

Lecture outline

01 Pretext tasks and masked prediction

02 Contrastive learning

03 Learning without negative pairs

04 Images and language

* denotes optional slides

Training targets already in the data

The bird landed on a branch.

Input prefix		Observed target
The	→	bird
The bird	→	landed
The bird landed on a	→	branch

Images and their annotations

Annotation

bird

Raw image

Class, box or mask

Useful labels take work to obtain.

Source: Photo: Fredo Durand / MIT 6.7960

Self-supervised learning (SSL)

Construct training targets from the data itself.

Hide a token

The bird [MASK] on a branch.

Target: the removed token

Apply a rotation

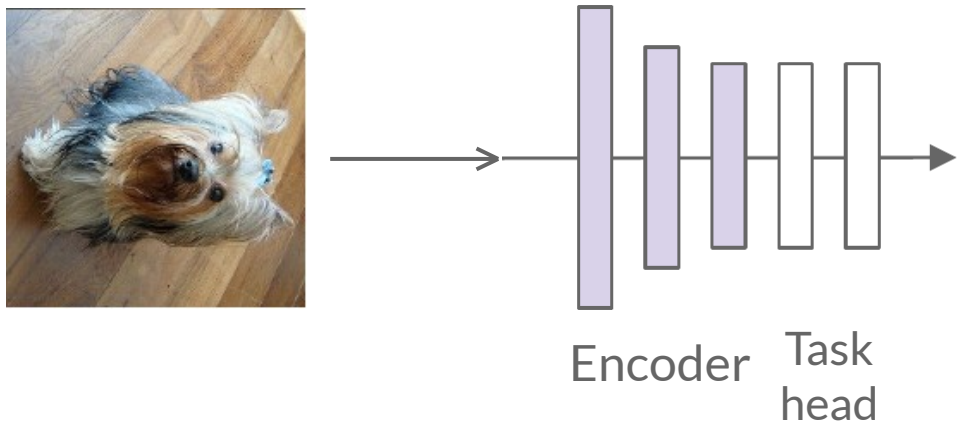
Rotate an image by 90°.

Target: the applied rotation

Pretext task: a training task chosen to learn useful features.

Pretraining and reusable features

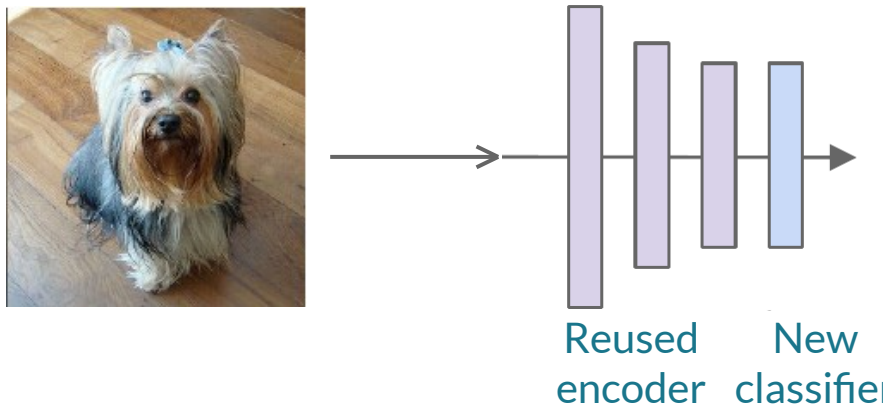
Example: rotation prediction



90°

Target known from the applied rotation

Downstream: object classification



dog

Target supplied by category annotations

Source: Adapted from CS231n · Image: Masked Autoencoders

Evaluating learned representations

Evaluate on a labeled task, using held-out examples.

Linear probe

Freeze the pretrained encoder.
Train a linear classifier on its features.

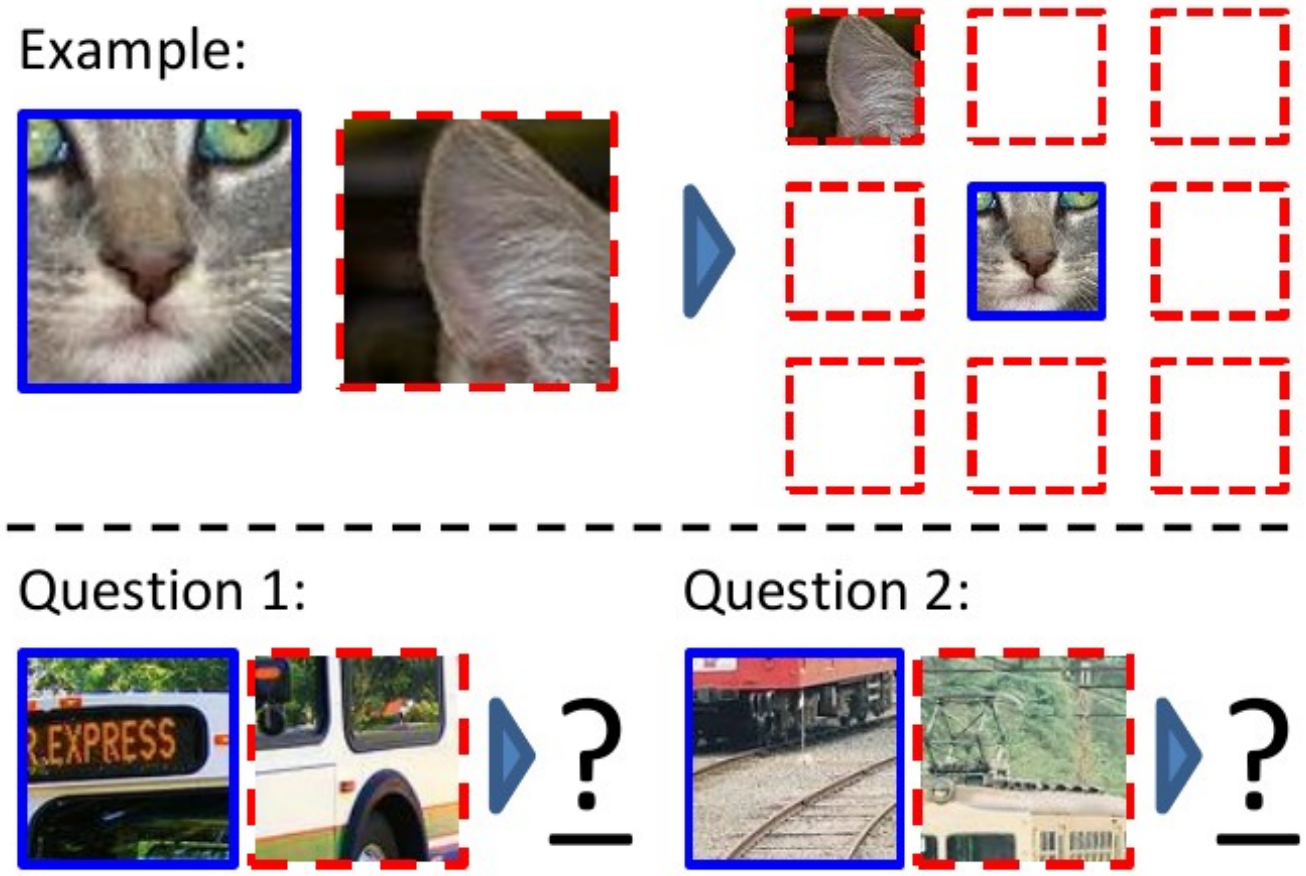
Tests what the existing features already make easy to classify.

Finetuning

Train the classifier and update the pretrained encoder.

Tests how well the pretrained model adapts to the new task.

Context prediction (2015)



Target

Position of one patch relative to the other.

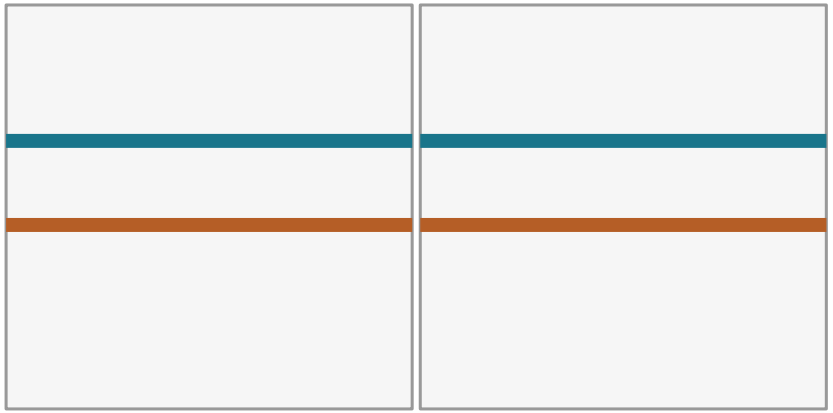
Known from the crop coordinates.

Object structure can help predict the answer.

Source: Context Prediction

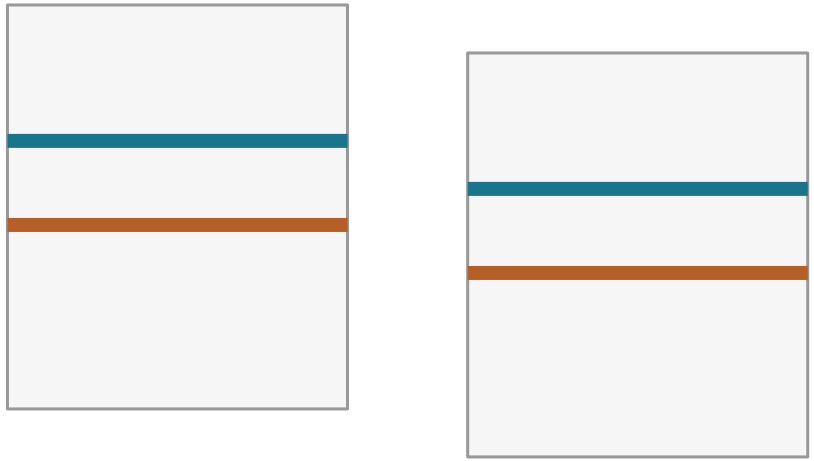
Shortcuts in a pretext task

Touching patches



Boundary continuity can reveal the answer.

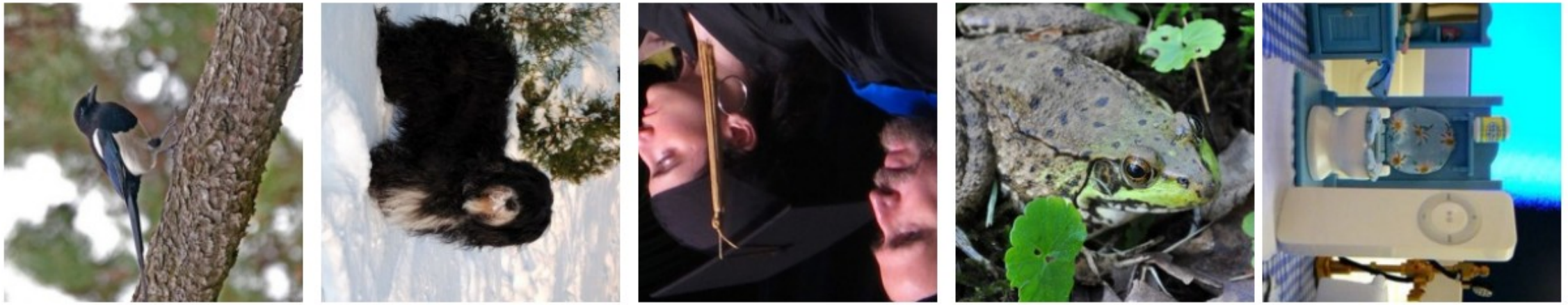
Gap and position jitter



Exact boundary matching is less useful.

Solving the pretext task does not guarantee useful features.

Rotation prediction (2018)



Input: a rotated image

Target: which of the four rotations was applied

Source: Rotation Prediction

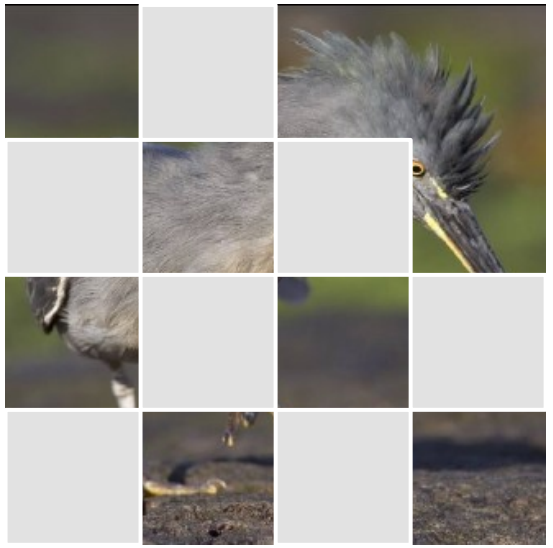
Predicting missing content

Text

The bird [MASK]
on a branch.

Predict a token from context.

Image

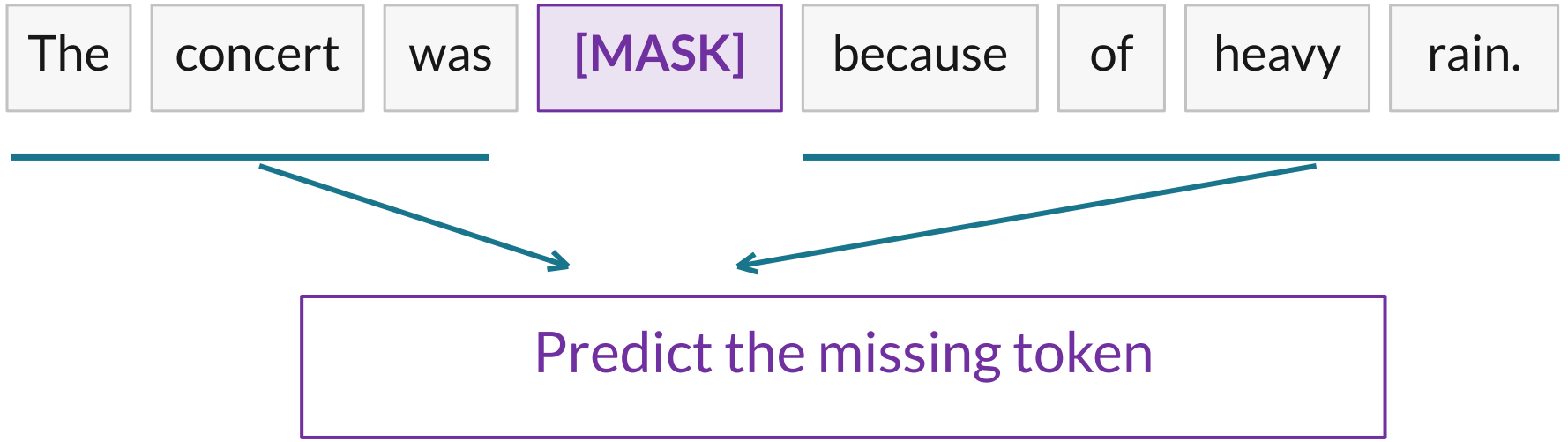


Predict missing patches.

Source: Photo: Fredo Durand / MIT 6.7960

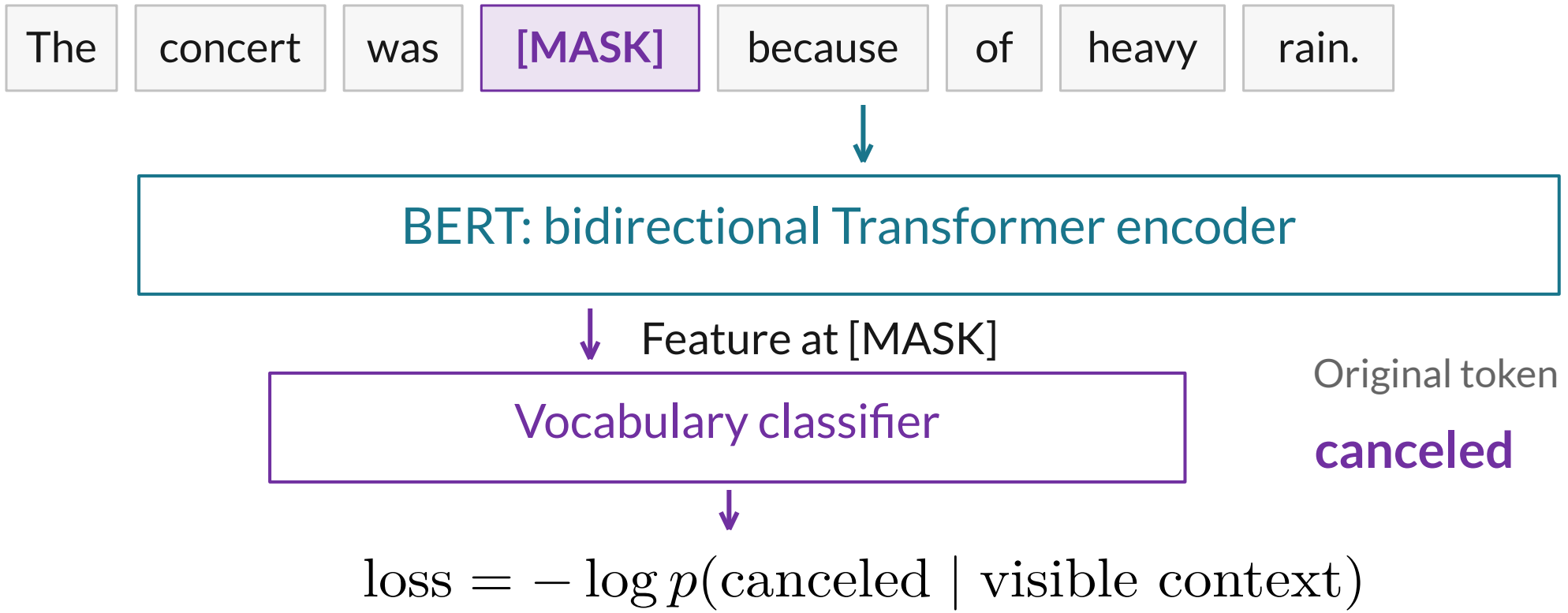
Context for a missing word

For a complete sentence, useful clues can appear on both sides.



Original target: canceled

Masked language modeling



Predict at selected positions. Average cross-entropy over those positions.

Input masks and attention masks

Input content mask

The	concert	was	[MASK]	because	of rain

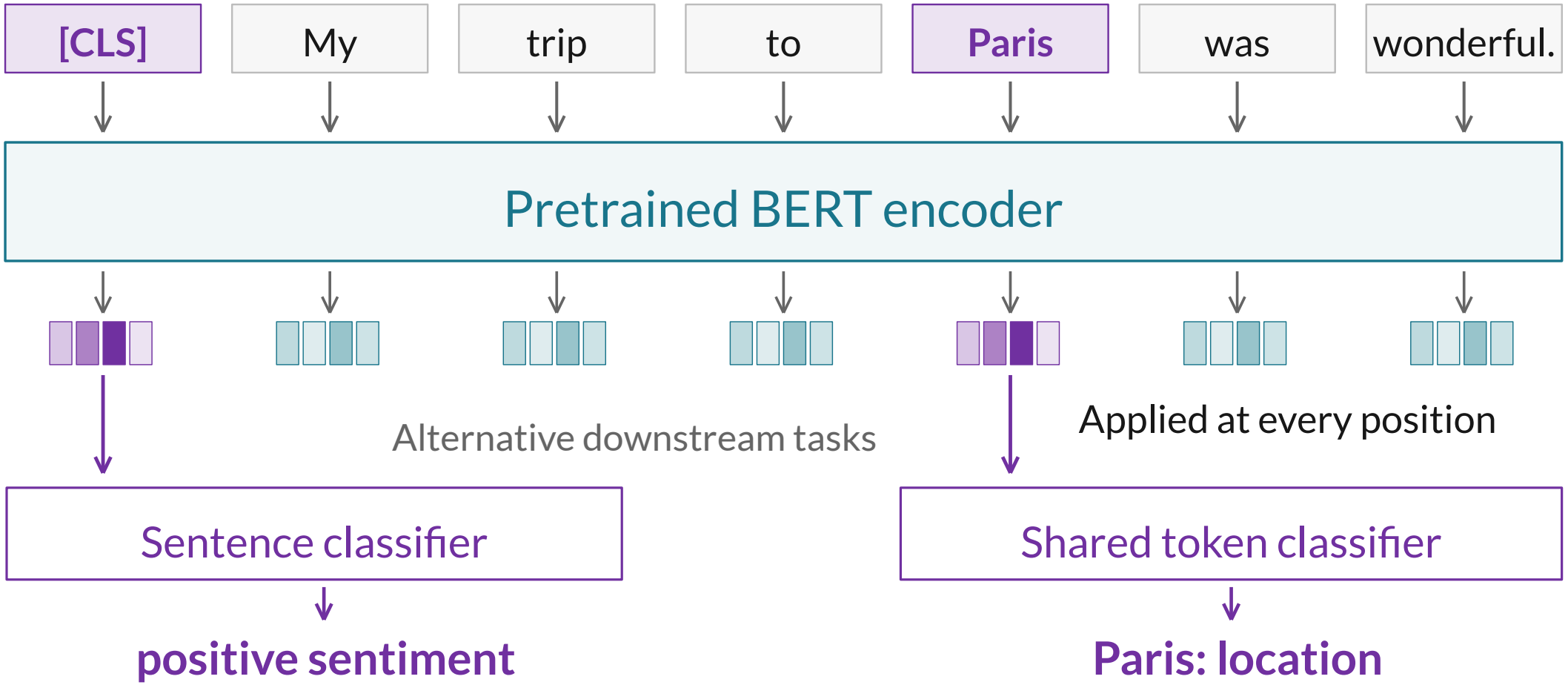
Use visible context on both sides of the missing content.

Causal attention mask

The	concert	was	canceled	because	⟶ next token

Use the prefix. Do not let the prediction read future tokens.

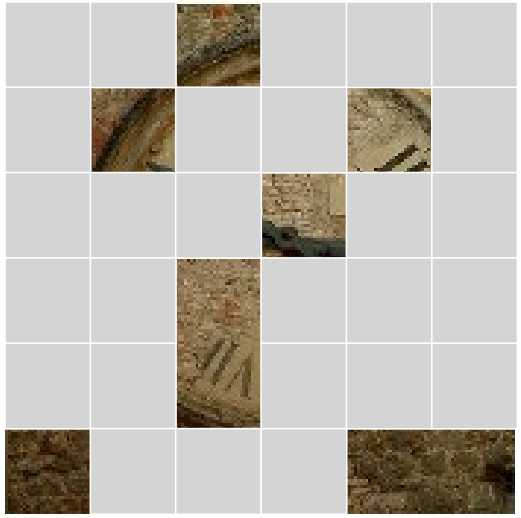
Using contextual text features



[CLS]: sequence-summary token

Masked image prediction

Visible input

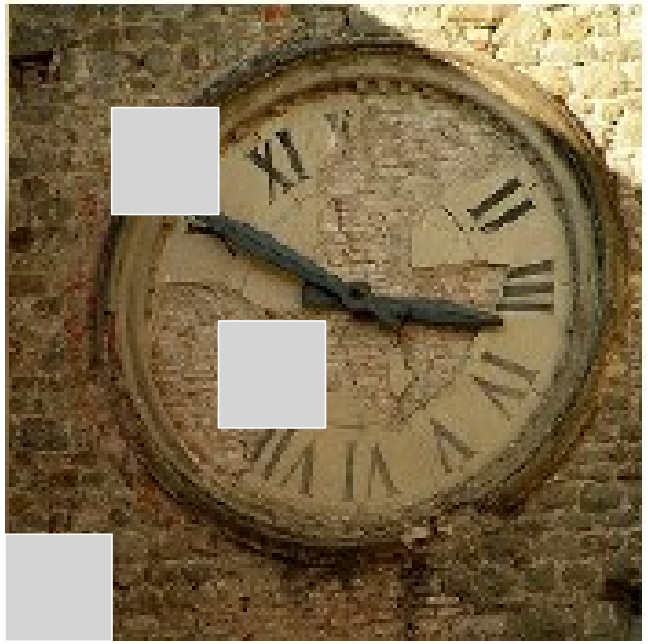


Learn to predict missing patches

Original pixels = target

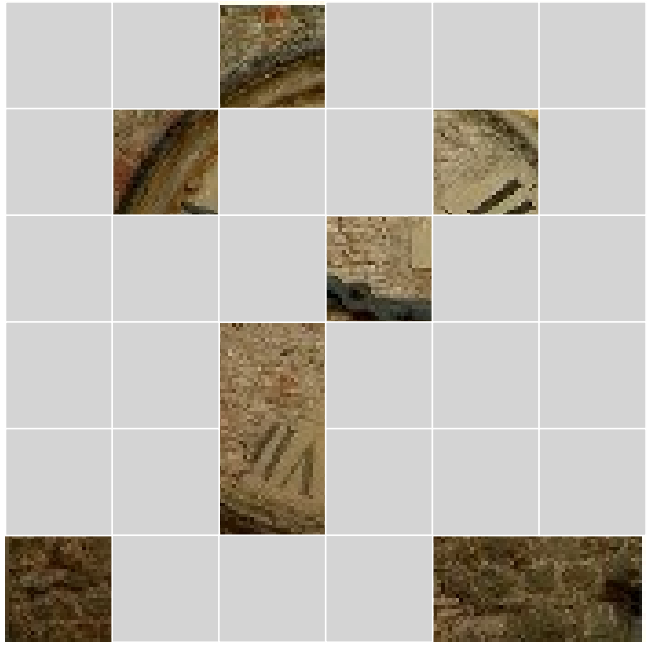
Source: Masked Autoencoders · arXiv:2111.06377

Why mask many image patches?



Hide a few patches

Nearby pixels often reveal the missing content.



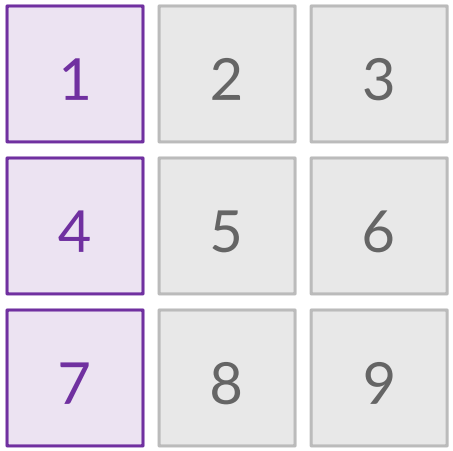
Hide many patches

Combine sparse clues from across the image.

Source: Masked Autoencoders · arXiv:2111.06377

Encoding visible patches

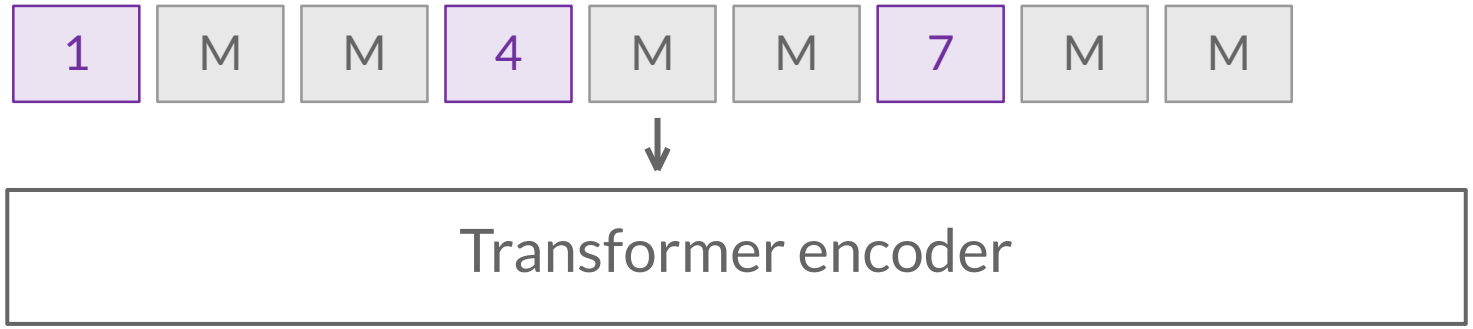
Same mask pattern



Observed: 1, 4, 7

M: mask token

All positions: 9 encoder tokens



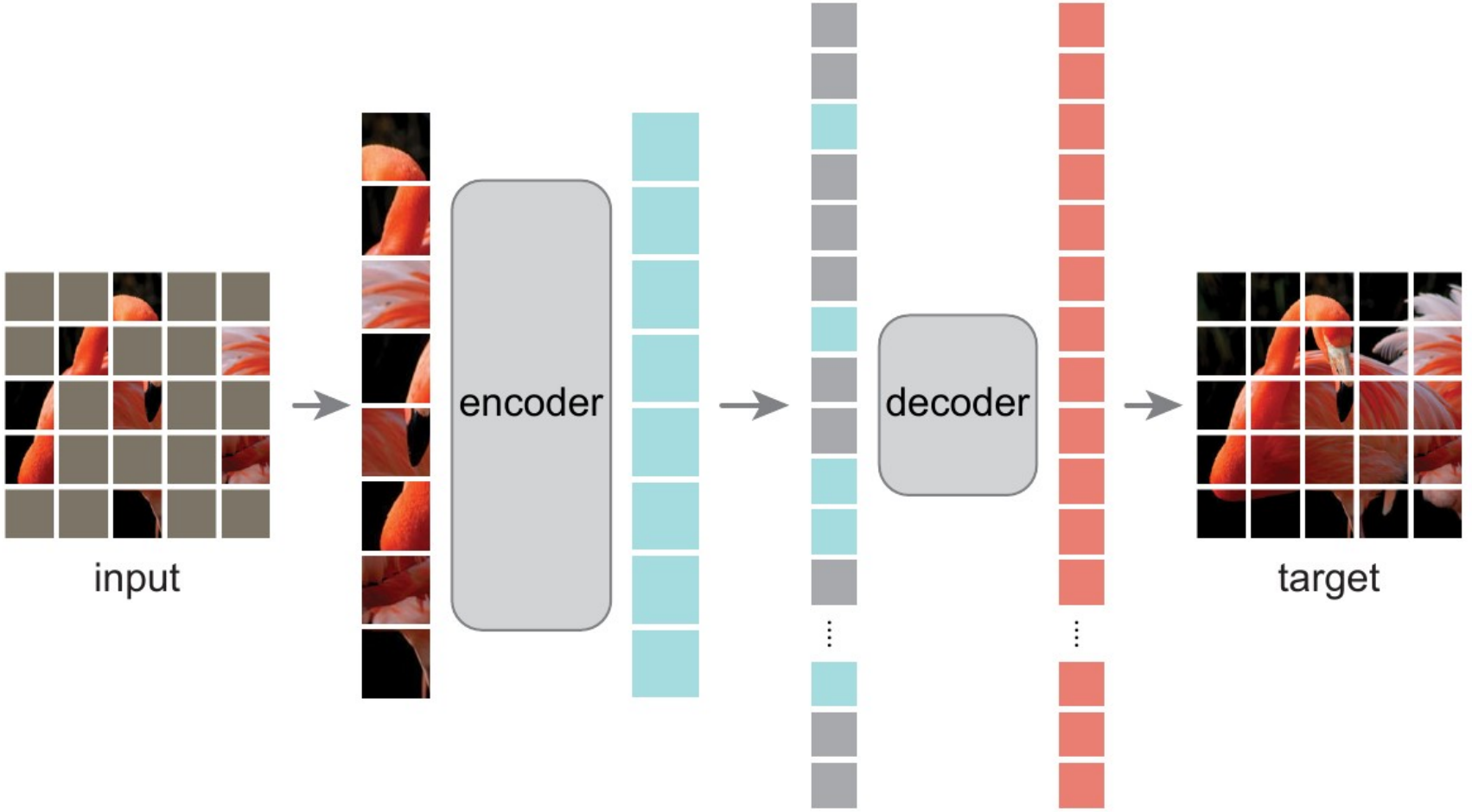
Visible tokens: less encoder computation

Keep the original position embeddings.

Six missing positions still need predictions.

Source: Masked Autoencoders · arXiv:2111.06377

Masked autoencoder (MAE)



Encode only visible patches.

Restore mask tokens and add positions.

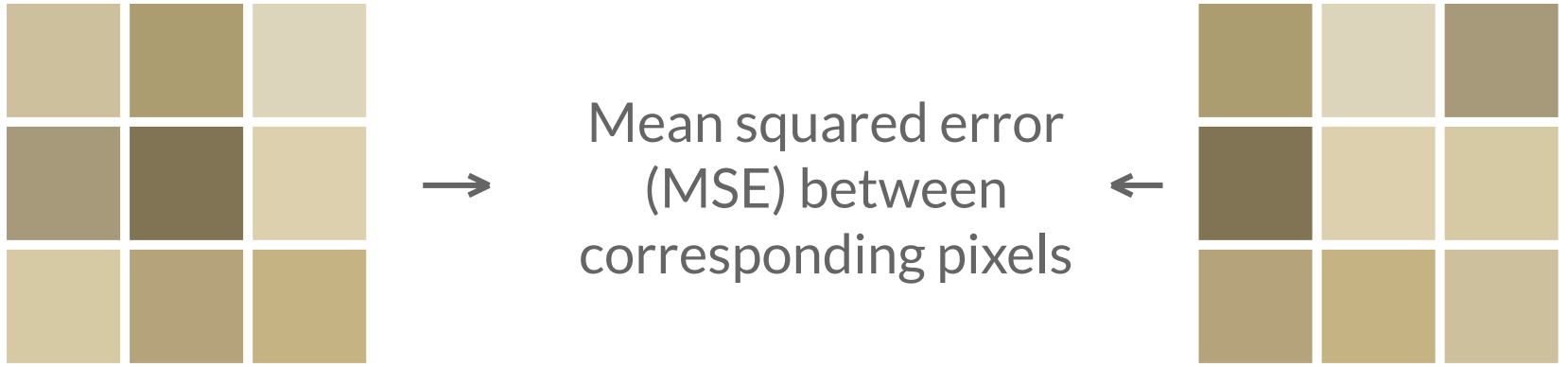
Small Transformer decoder predicts pixels in parallel.

Source: Masked Autoencoders · arXiv:2111.06377

Learning from missing patches

Predicted missing patch

Original patch target



$$\text{loss} = \text{mean}_{\text{masked pixels}} \left(\text{prediction} - \text{target}\right)^2$$

Compute loss on missing patches. Visible patches supply context.

MAE reconstructions

Masked input | Decoder output | Original

Both examples

80% of patches hidden

Source: Masked Autoencoders · arXiv:2111.06377

Using the pretrained image encoder

Pretraining

Visible patches

Encoder

Reconstruction decoder

Reconstruction

Reuse parameters

Image classification

All patches

Pretrained encoder

New classifier

clock

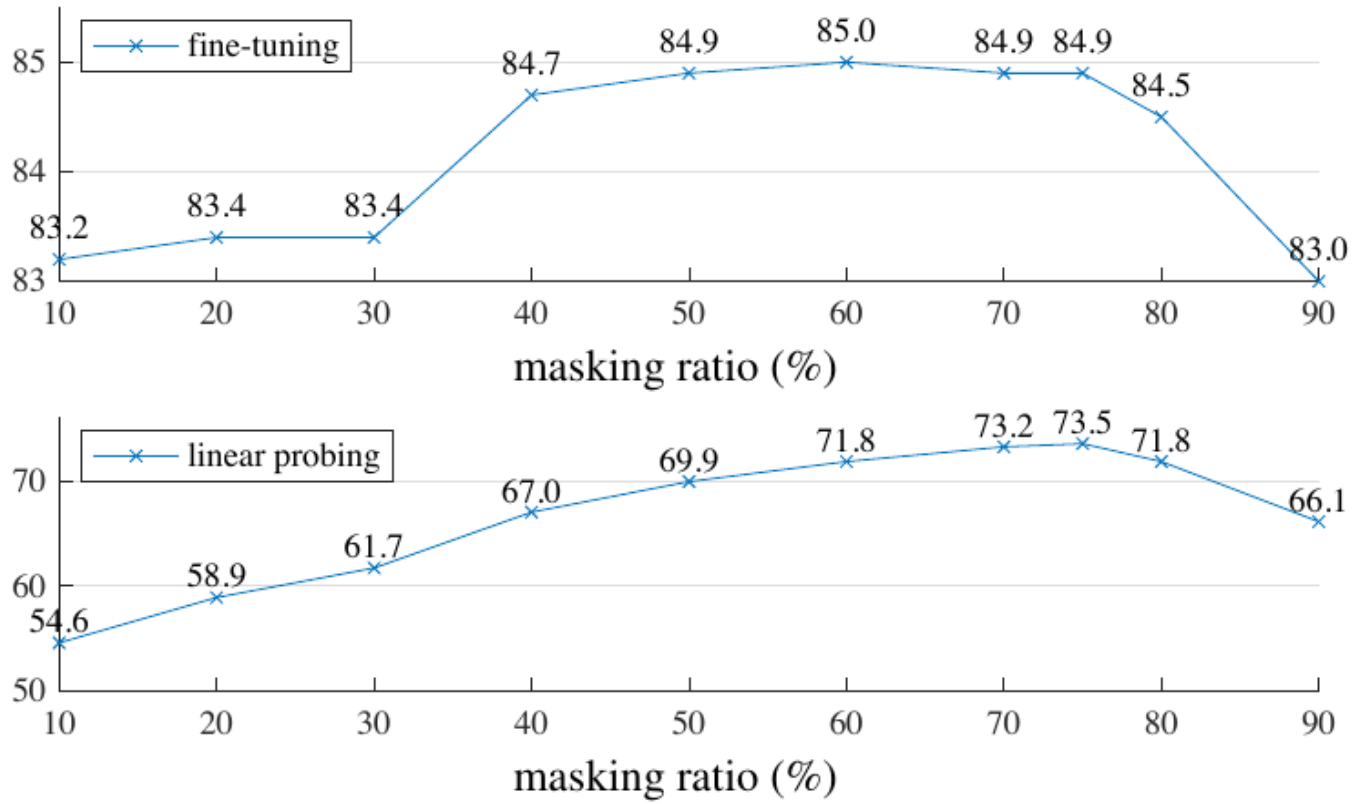
Category

Train with category labels.

Source: Masked Autoencoders · arXiv:2111.06377

Masking and feature quality

MAE-pretrained large ViT · ImageNet accuracy (%)



Finetuning

Update the encoder and classifier.

Linear probing

Freeze the encoder. Train a classifier.

Finetuning and linear probing favor different masking ratios.

Source: Masked Autoencoders · arXiv:2111.06377

Lecture outline

01 Pretext tasks and masked prediction

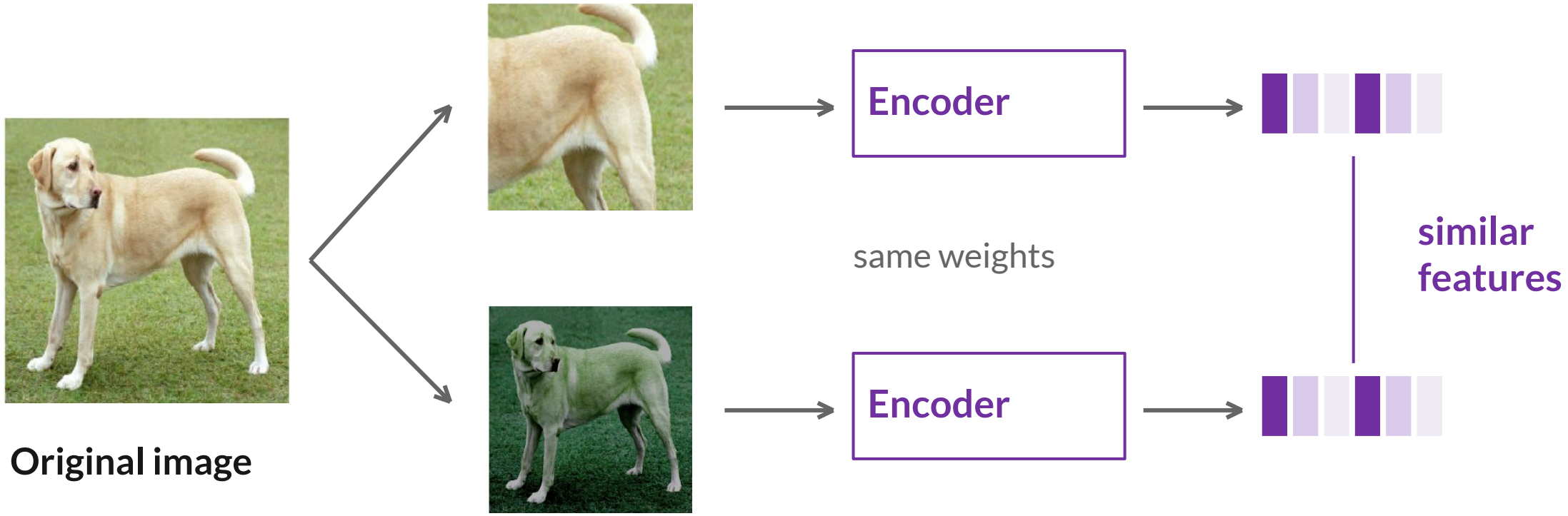
02 Contrastive learning

03 Learning without negative pairs

04 Images and language

Learning from two views

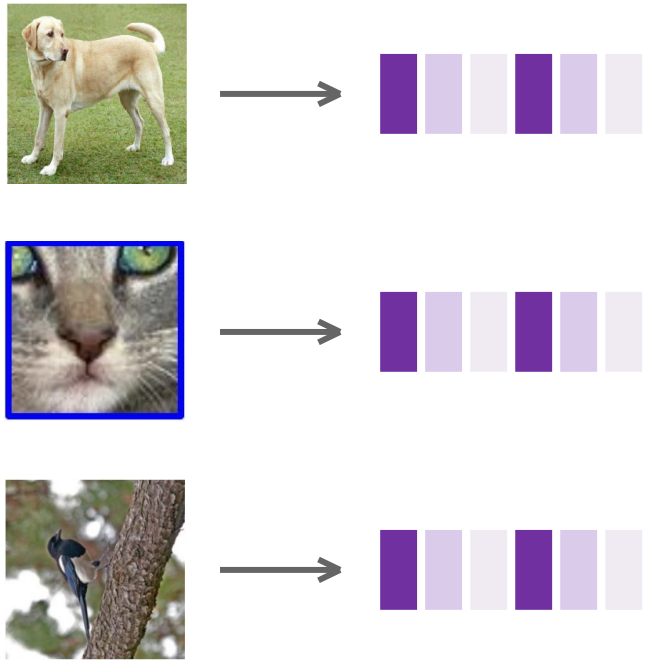
Learn features that agree across meaningful changes to an image.



Source: SimCLR

Representation collapse

Perfect agreement, no information



Every input → the same vector

Useful features retain differences

Views of one image stay close.
Different images remain distinguishable.

Source: SimCLR · Context prediction · Rotation prediction

Positive and negative pairs

Anchor

Positive

same original image

Negatives

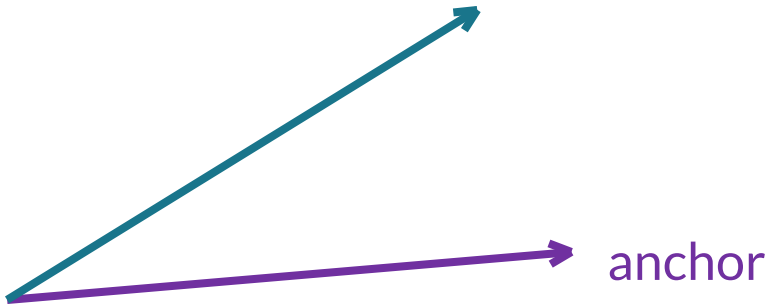
other images

Here, pair labels come from image identity, without class labels.

Source: SimCLR · Context prediction · Rotation prediction

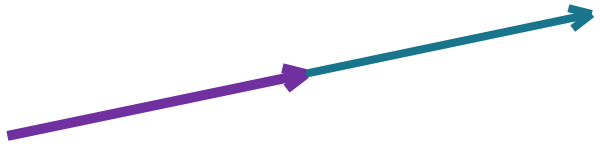
Feature similarity

Compare feature direction, rather than increasing vector length.



$$\text{sim}(u, v) = \frac{u^\top v}{\|u\| \, \|v\|}$$

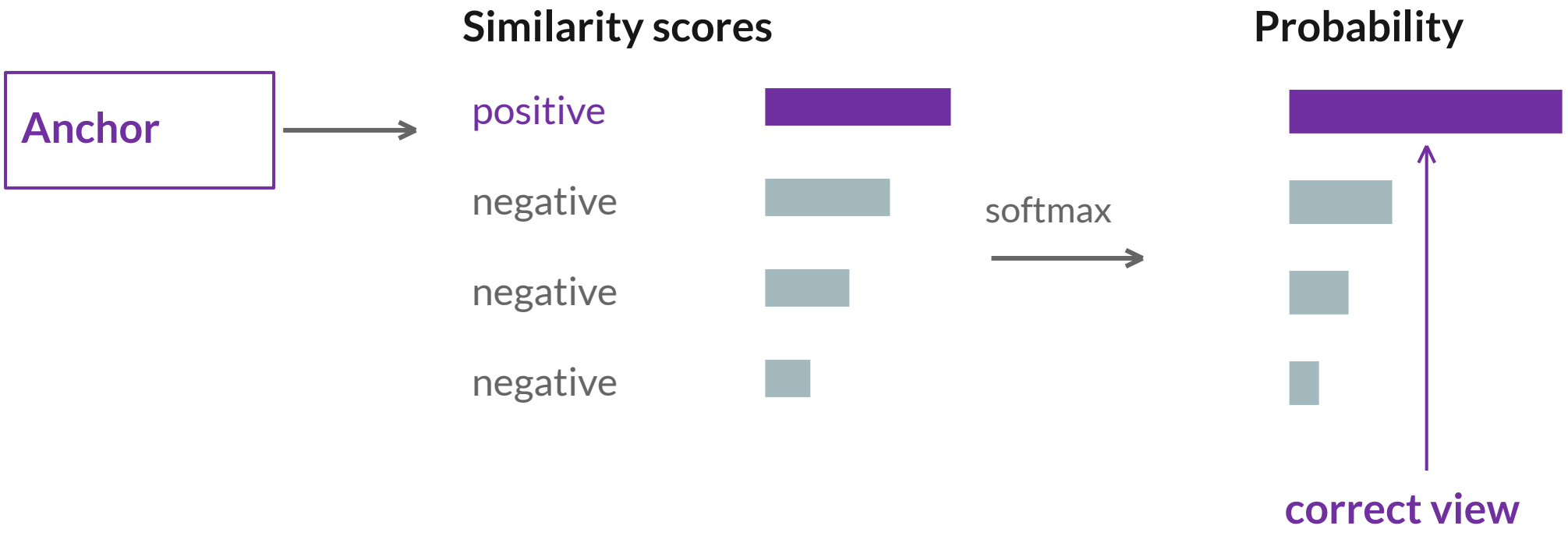
u, v: feature vectors
‖ u ‖ , ‖ v ‖ : vector lengths



Same direction → same cosine similarity

Finding the matching view

The target is the matching candidate—not an object category.



Cross-entropy rewards a high probability for the positive.

Equal features give equal probabilities: collapse cannot identify the match.

Contrastive cross-entropy

Probability of the positive

$$p_+ = \frac{\exp(s_+/\tau)}{\sum_{k \in \text{candidates}} \exp(s_k/\tau)}$$

s_+: positive similarity
s_k: candidate similarity
τ : temperature

$$\mathcal{L} = -\log p_+$$

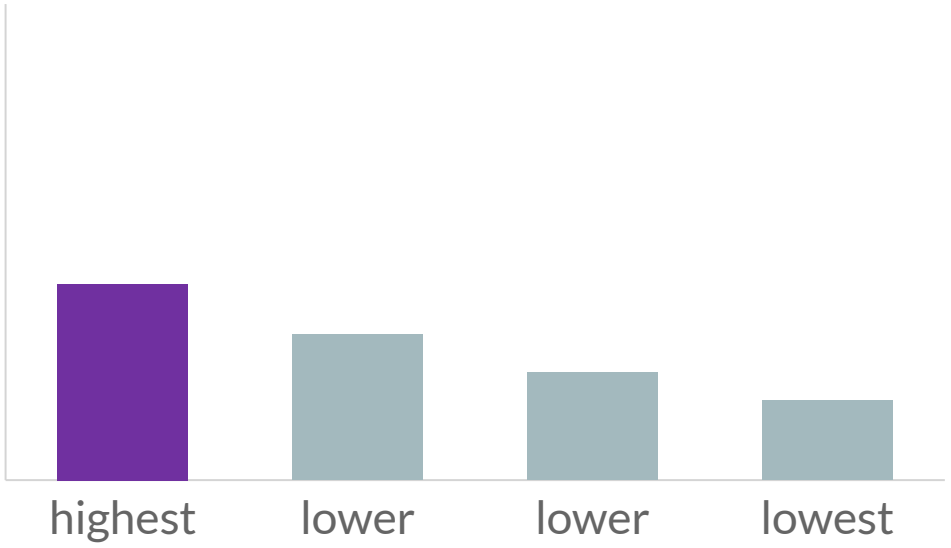
Cross-entropy over candidate views
(often called InfoNCE)

Include the positive in the denominator. Exclude the anchor itself.

Temperature

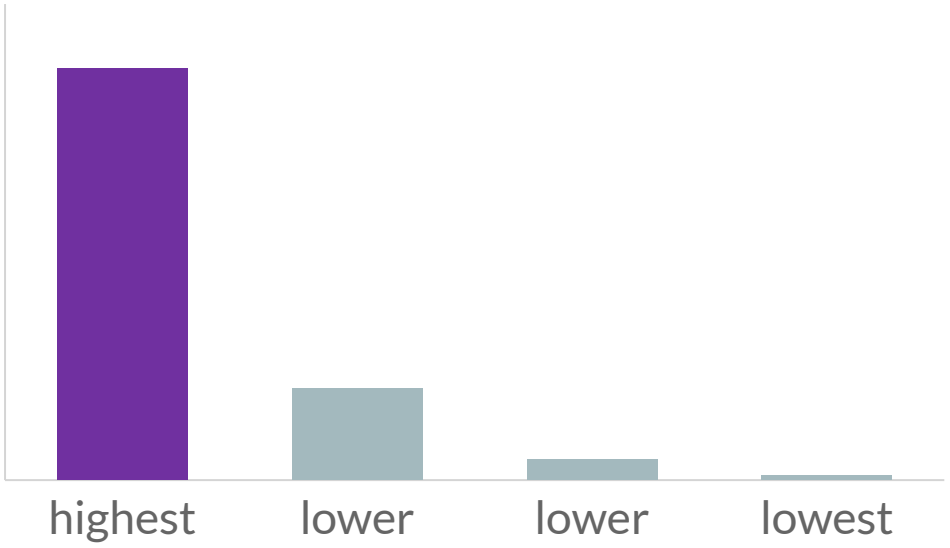
Same similarity scores. Different sharpness after softmax.

Larger temperature



More even probabilities

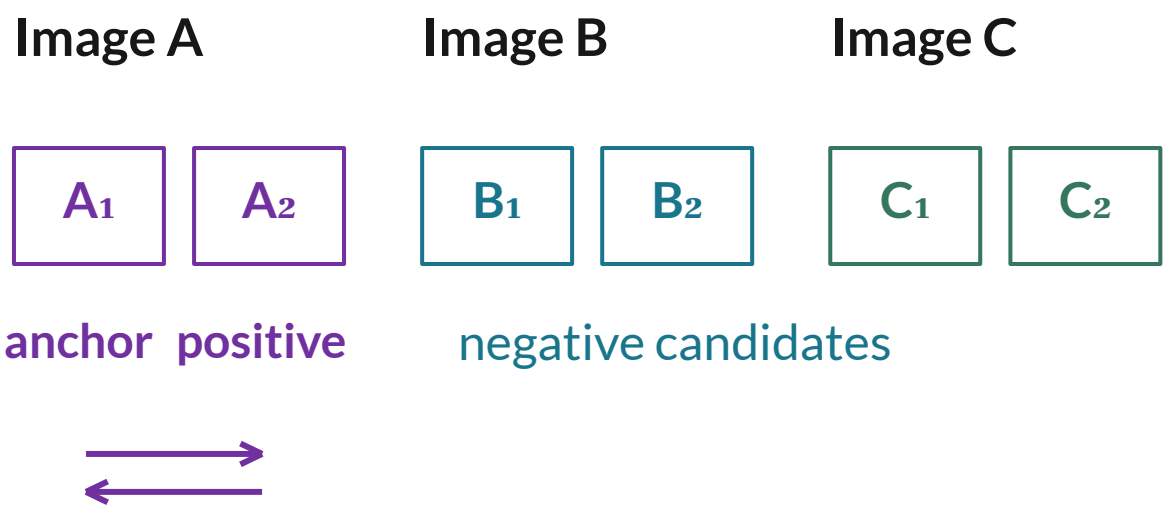
Smaller temperature



Emphasizes the highest scores

Two views of every image

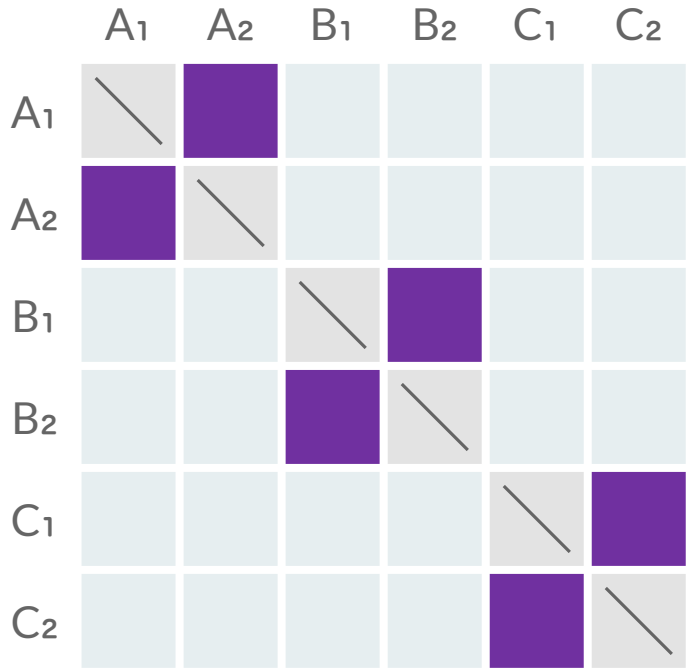
Two independently augmented views per original image.



Both views take a turn as anchor.

Average the loss over all anchors.

Rows: anchors · columns: candidates



Purple: positive · diagonal: excluded

Representation and projection

Matching color-altered views favors ignoring color.
A later task may still need color information.

Encoder representation

$$h = f(x)$$

Retain h for downstream tasks.

Projected representation

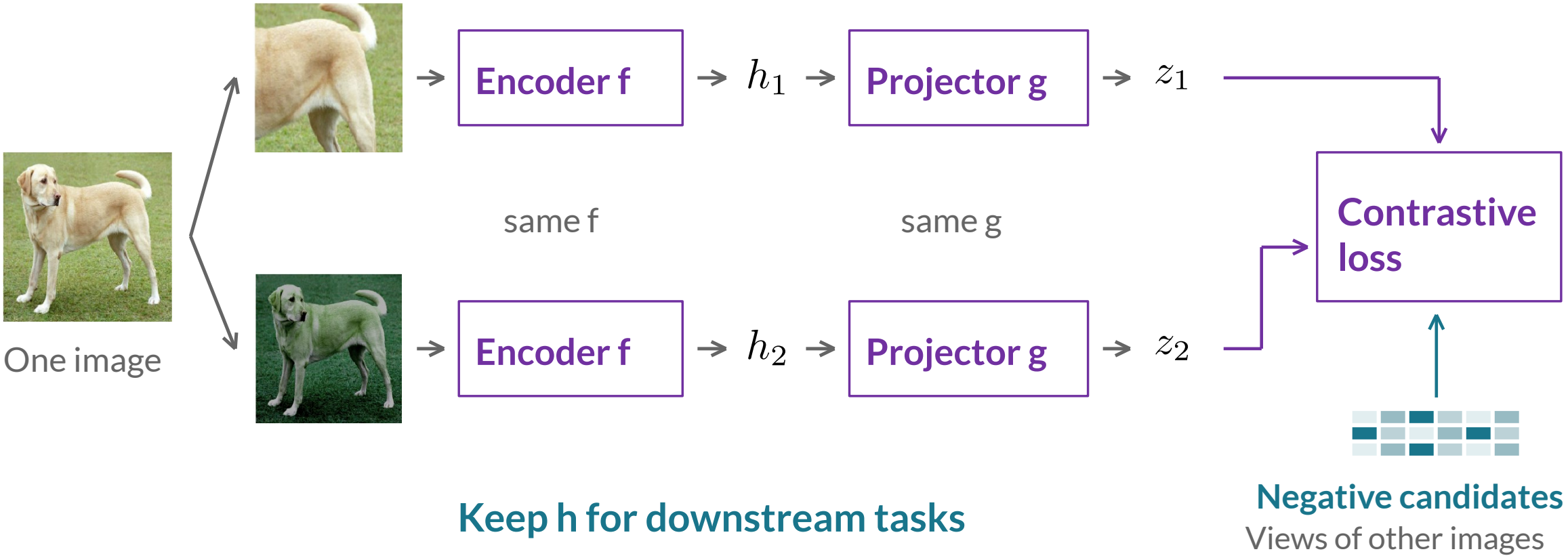
$$z = g(h)$$

Projection head g: a small MLP trained for contrastive matching.

Source: SimCLR

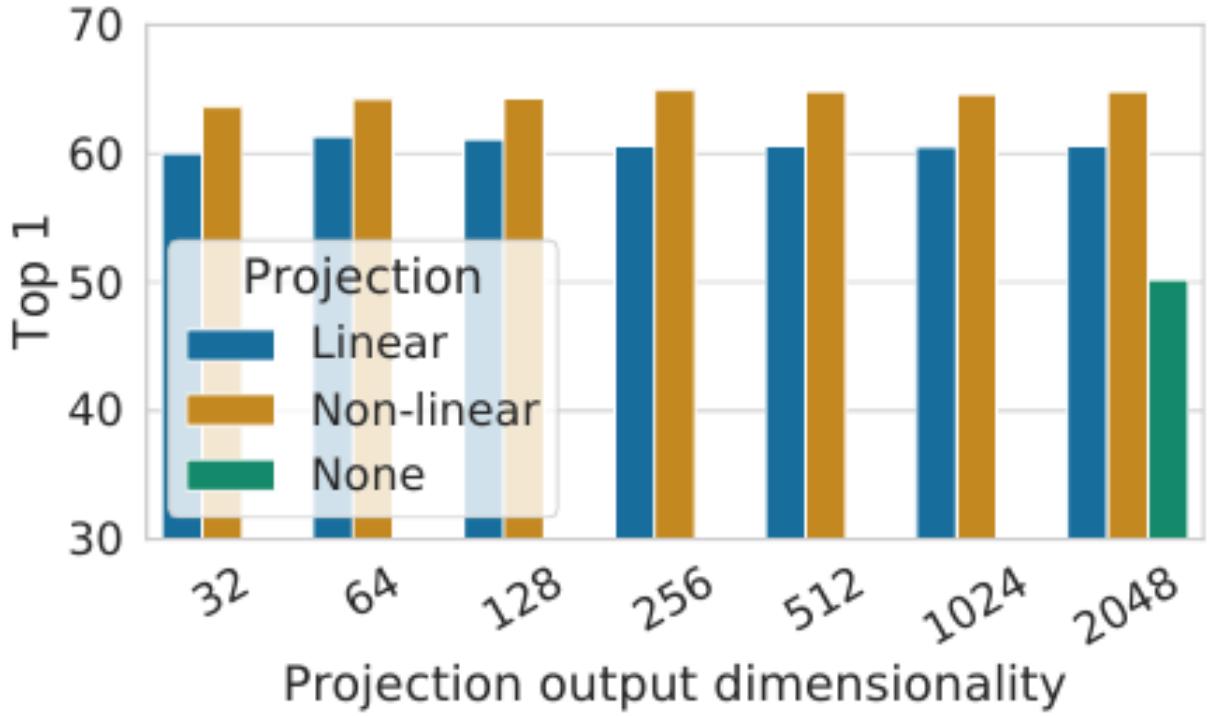
SimCLR

Shared weights. Gradients through both branches.



Source: SimCLR

Evidence for the projection head



Frozen encoder → linear classifier

Measured on h, before the head.

A nonlinear head improved the retained encoder features.

ImageNet top-1 accuracy
Original SimCLR experiment

Source: SimCLR

Augmentations define the task

Choose transformations that preserve information needed by the downstream task.

Original

Random crop

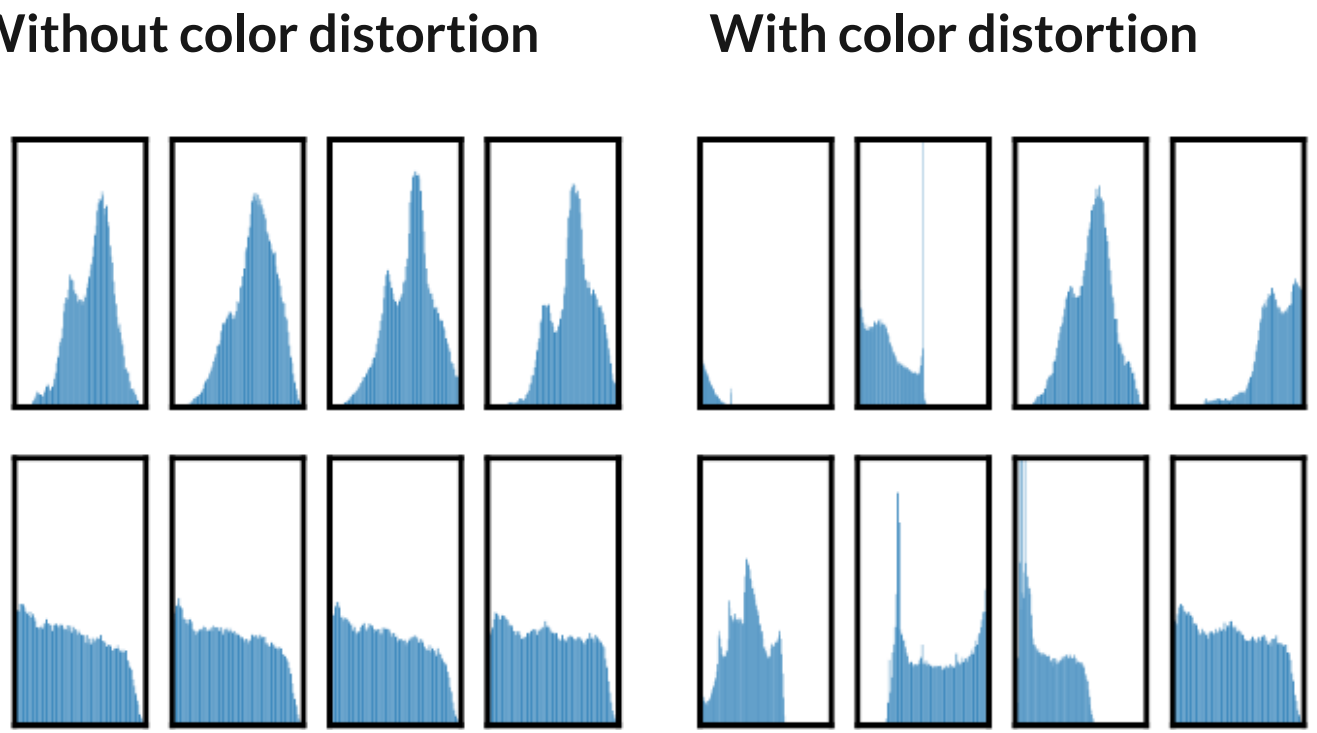
Color distortion

Gaussian blur

Source: SimCLR

Matching through color statistics

Pixel-intensity histograms of different crops



Each row: one image. Color distortion makes histogram matching less reliable.

Source: SimCLR

Combining transformations

Diagonal:
one transformation

Off-diagonal:
a combination

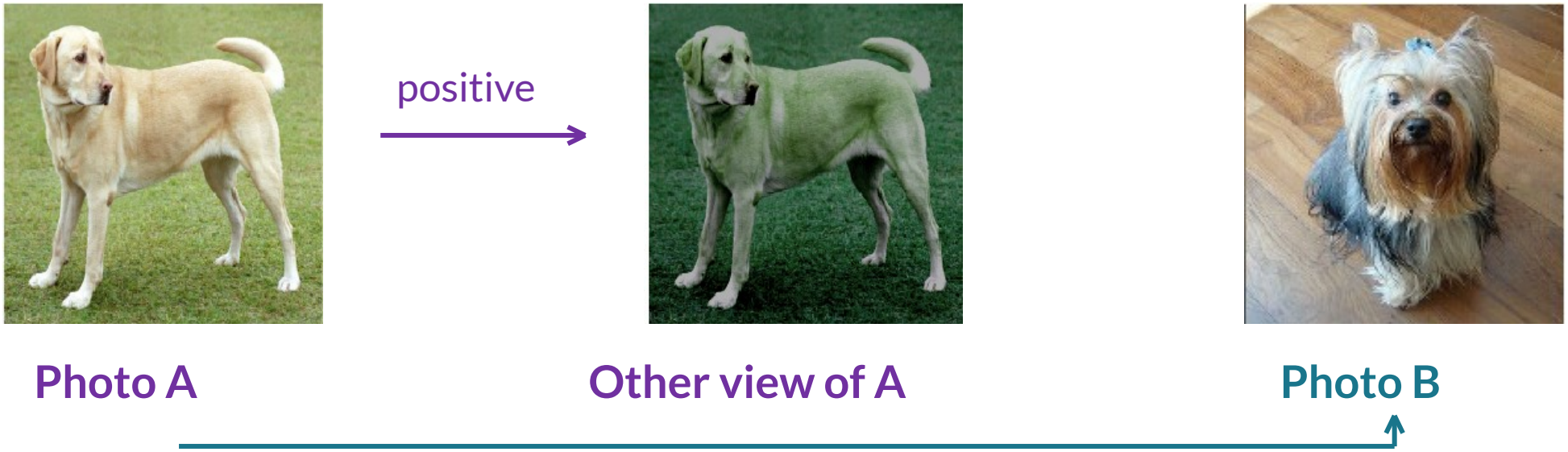
Crop + color distortion works well here.

Values: frozen-feature ImageNet top-1 accuracy

Source: SimCLR

False negatives

Different images can still belong to the same semantic category.



Source: SimCLR · Masked Autoencoders

More candidates, more memory

More negatives can improve SimCLR features, but need a larger minibatch.

Smaller batch

Fewer candidates

More views to encode and backpropagate

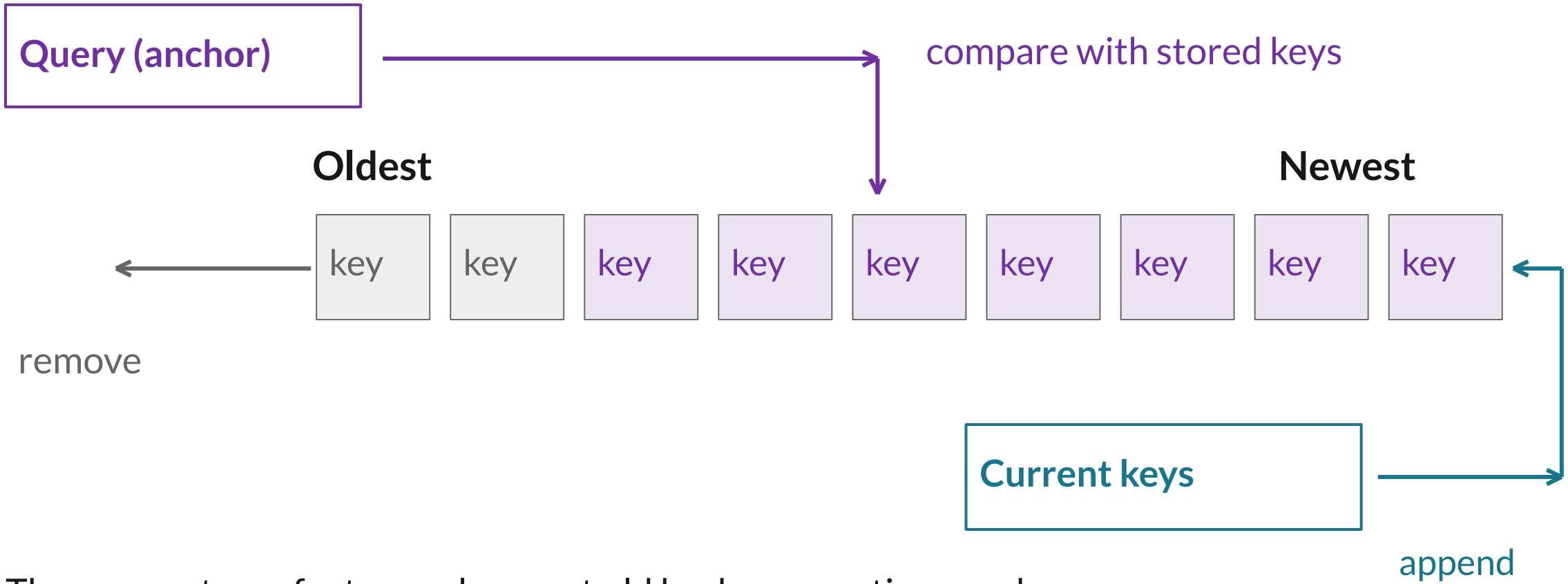
Larger batch

More candidates

Reuse features from earlier batches?

Queue of past features

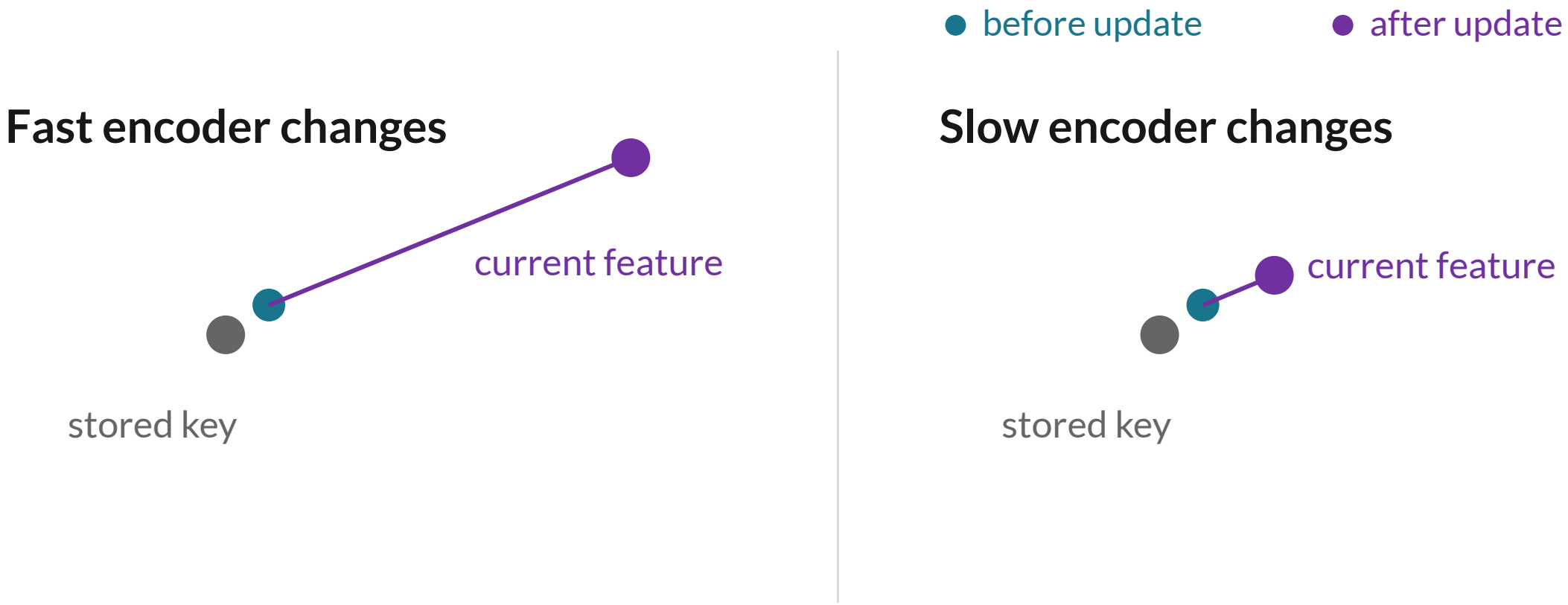
Store recent candidate features (keys). Reuse them as negatives.



The queue stores feature values, not old backpropagation graphs.

Keeping stored features comparable

Same image, before and after an encoder update. Stored keys stay fixed.



A slowly changing key encoder reduces inconsistency across recent batches.

Momentum encoder

Exponential moving average (EMA) of model parameters

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q$$

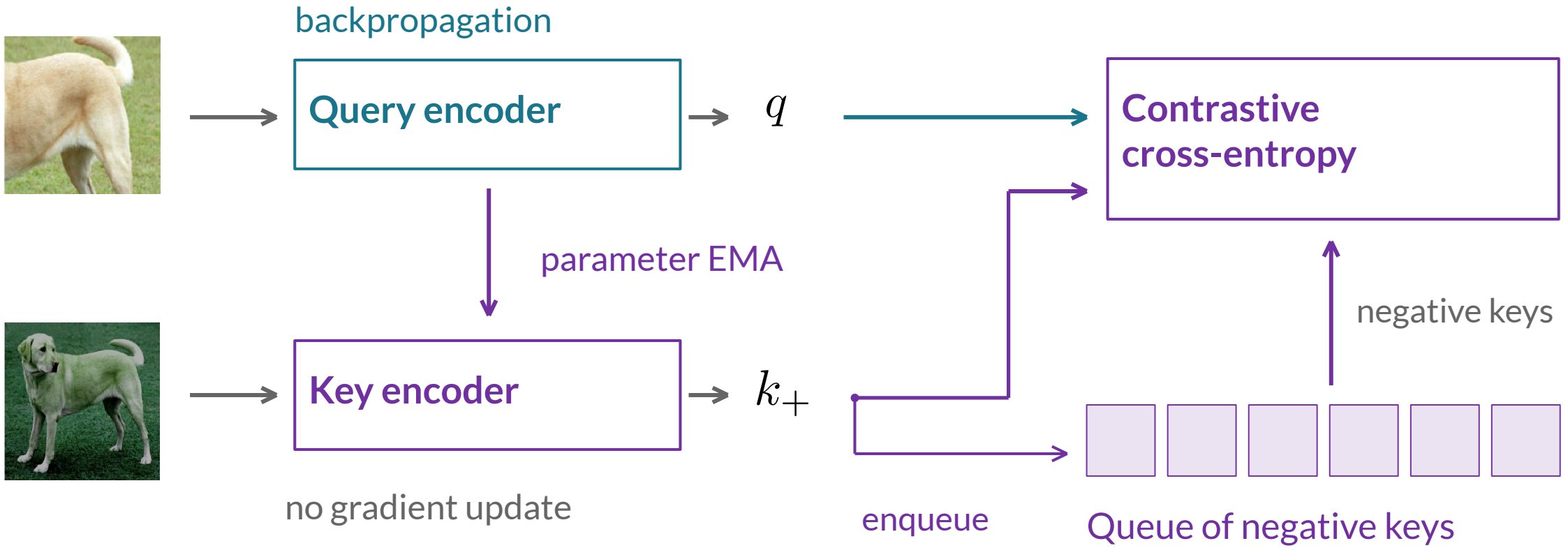
θ_k key-encoder parameters

θ_q query-encoder parameters

m averaging coefficient, close to 1

Momentum Contrast (MoCo)

Positive: paired current view. Negatives: keys stored from recent batches.



Source: Photos: SimCLR · MoCo

MoCo training step

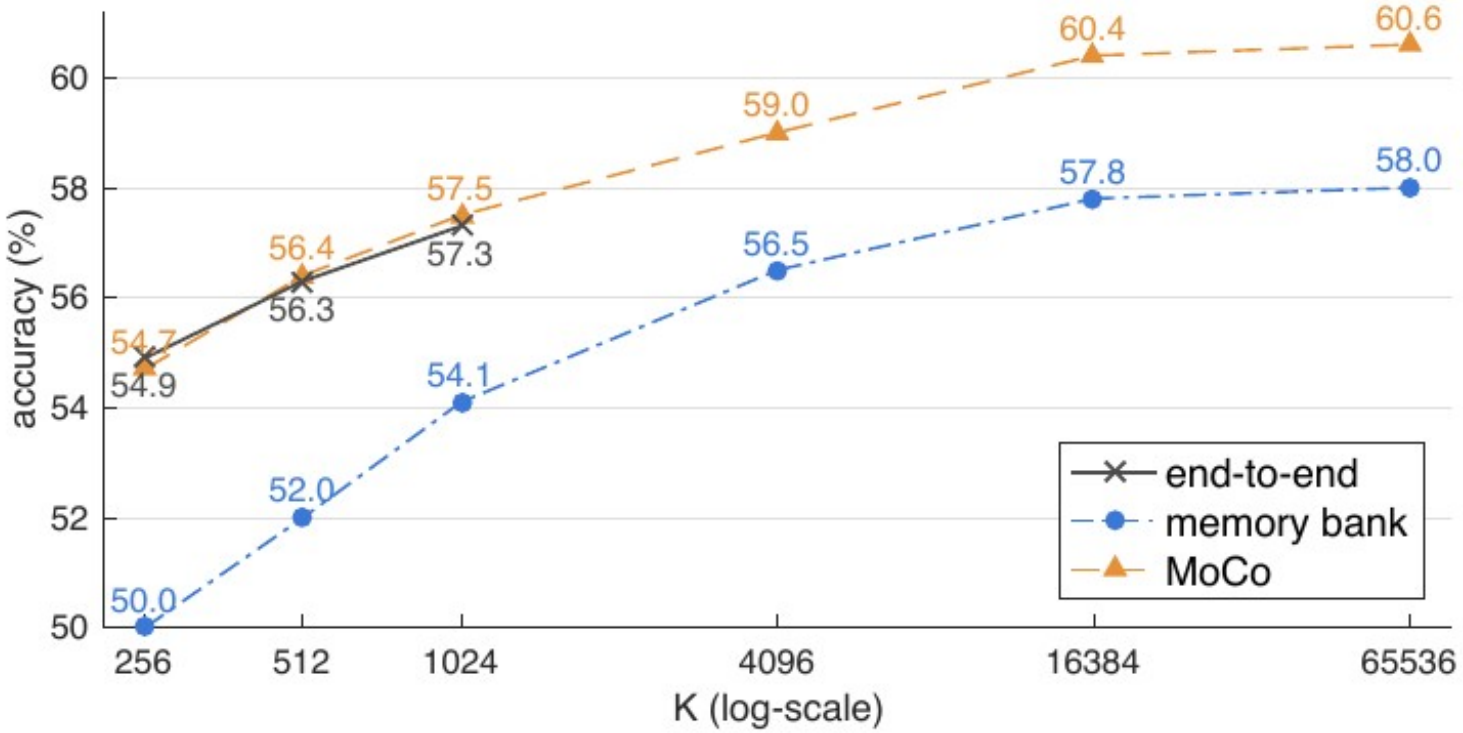
Algorithm · Momentum Contrast

Initialize the key encoder from the query encoder.

---

1. Two views give a query and a positive key. No gradient through keys.
2. Compare with the positive and the existing queue.
3. Cross-entropy, backpropagation, then update the query encoder.
4. EMA-update the key encoder from the updated query encoder.
5. Enqueue current keys. Remove the oldest keys.

Learning with reusable candidates



K: number of candidate keys

ResNet-50
ImageNet linear probe

End-to-end: current batch

Memory bank: old features

MoCo: queue + slow encoder

Source: MoCo

Lecture outline

01 Pretext tasks and masked prediction

02 Contrastive learning

03 Learning without negative pairs

04 Images and language

Learning without negative pairs

Agree

Two views of one image

Avoid treating other images as negatives.

The trivial solution remains

Any input →

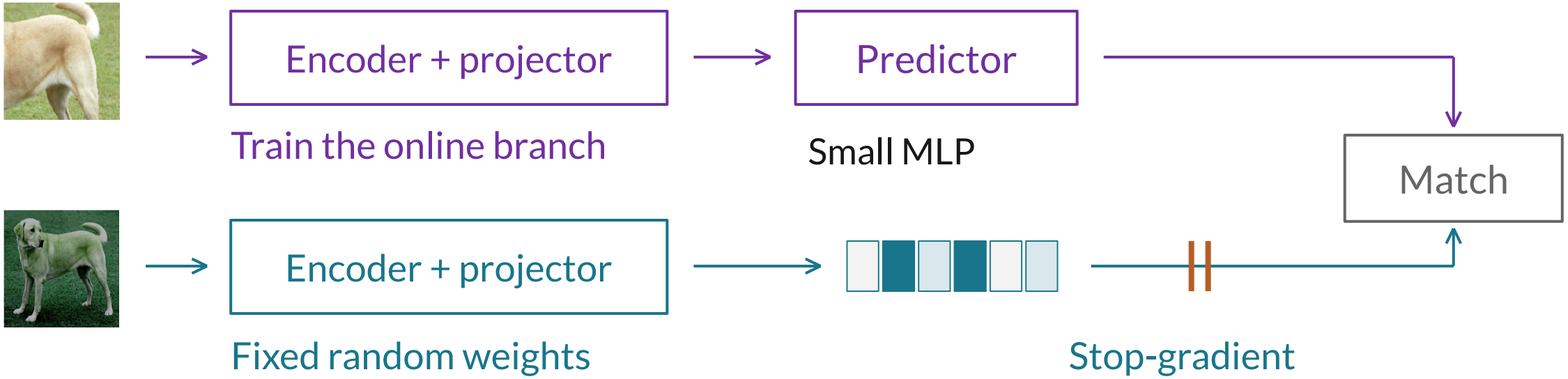
Same output vector for every image

Freeze one network to supply prediction targets.

Source: SimCLR (image)

Predicting a fixed target

Freeze a randomly initialized target network.
Its outputs can still differ across images.




```
prediction = predictor(online(view1))
target = target_net(view2).detach()  # no target gradient
loss = match(prediction, target)
```

Source: BYOL · SimCLR (image)

Updating the target network

The online network can learn better features than its fixed target.
Let the target follow the improving online network.

$$\theta_{\text{target}} \leftarrow m\,\theta_{\text{target}} + (1-m)\,\theta_{\text{online}}$$

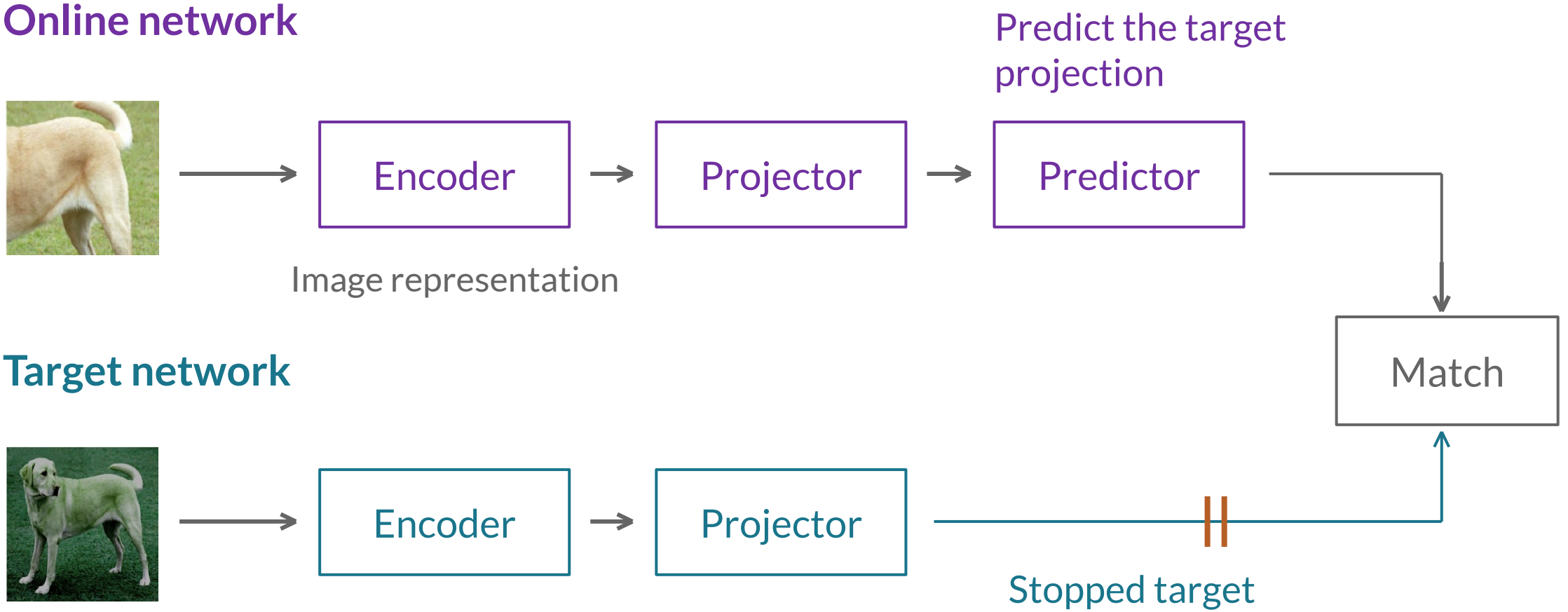
Exponential moving average (EMA)

m close to 1 makes the target change gradually.

Start both encoders and projectors with the same random weights.

Source: BYOL

BYOL: predicting another view



Source: BYOL · SimCLR (image)

BYOL's training objective

Normalize prediction and target to length 1.

$$\mathcal{L} = \left\| \widehat{\text{prediction}} - \text{stopgrad}\left(\widehat{\text{target}}\right) \right\|_2^2$$

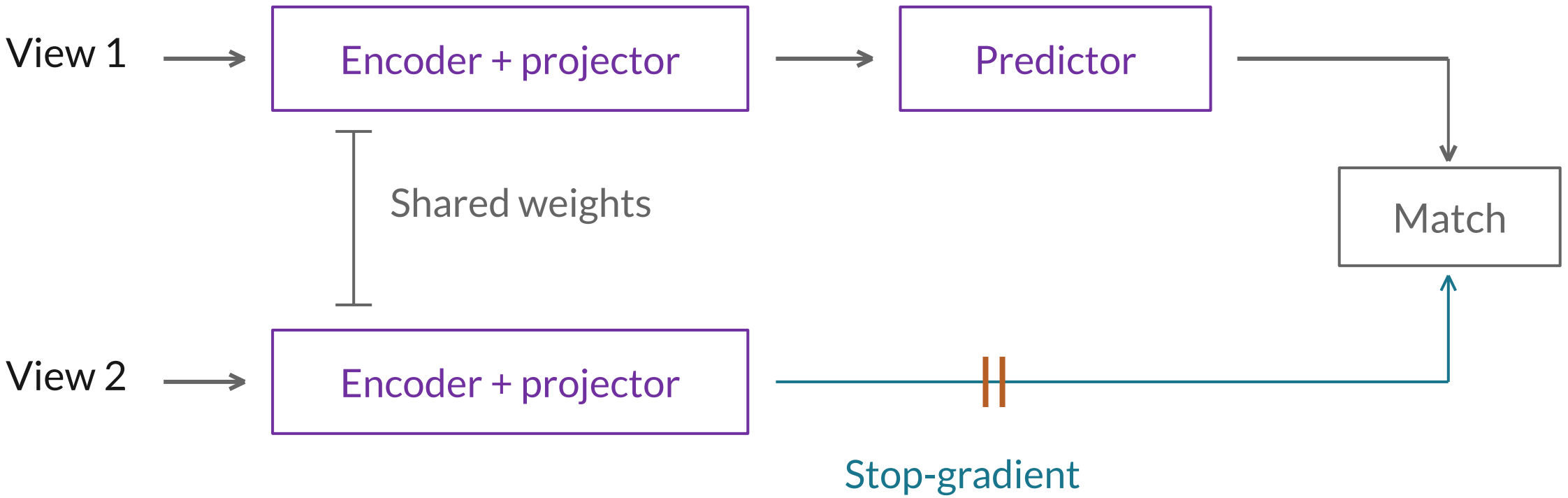
Squared distance between directions

Swap the views and add the second prediction loss.

After pretraining, reuse the online encoder.

SimSiam: removing the momentum teacher

Keep the predictor and stop-gradient.



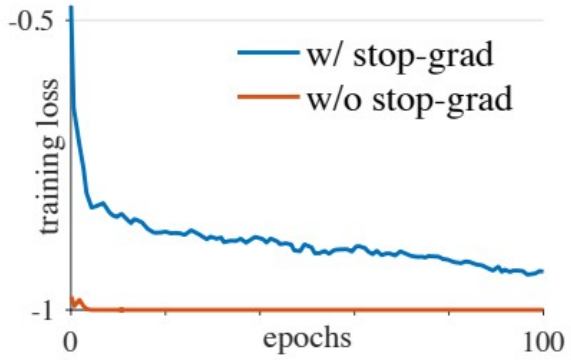
Shared weights still update through the prediction branch.

Source: SimSiam

Low loss can accompany collapse

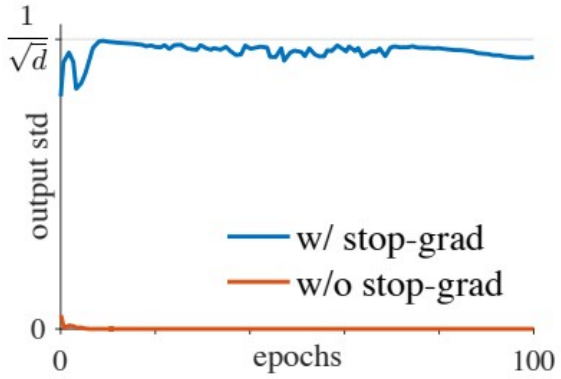
SimSiam with ResNet-50 on ImageNet

Negative cosine similarity



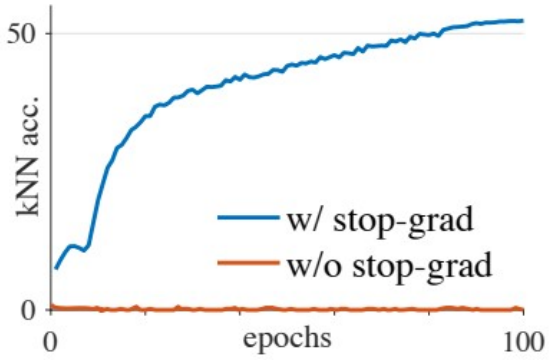
Without stop-gradient: lowest matching loss

Feature variation



Different images give the same output.

Classification accuracy



The features cannot distinguish categories.

kNN: classify using the nearest labeled image features.

Source: SimSiam, Figure 2

DINO: self-distillation

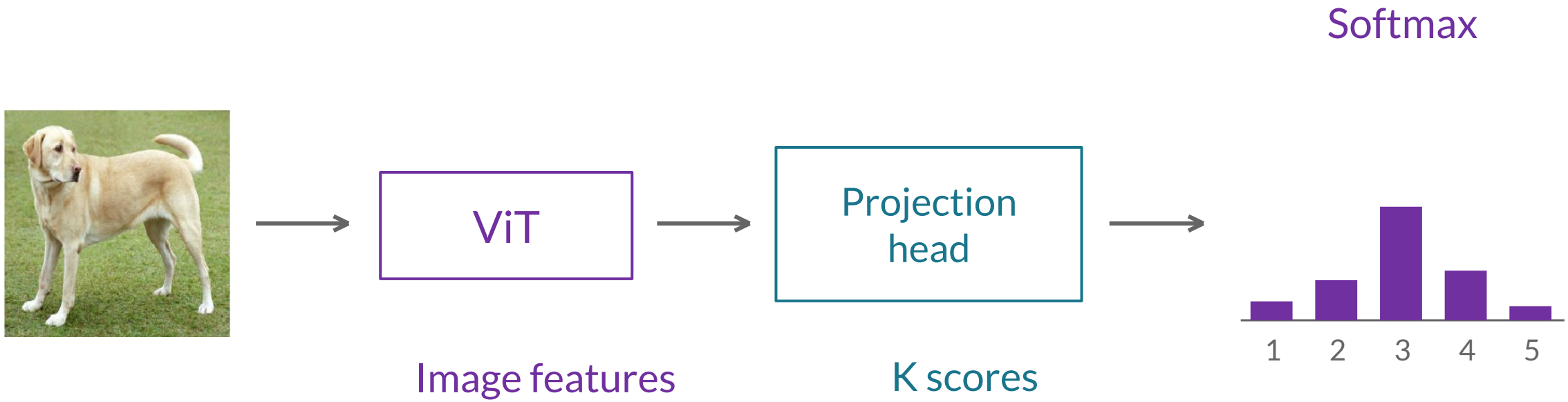
Distillation with a fixed teacher

A previously trained teacher supplies target probabilities.
The student learns while the teacher stays fixed.

DINO

The teacher develops during student training.
Its parameters follow the student by EMA.

Distributions without class labels

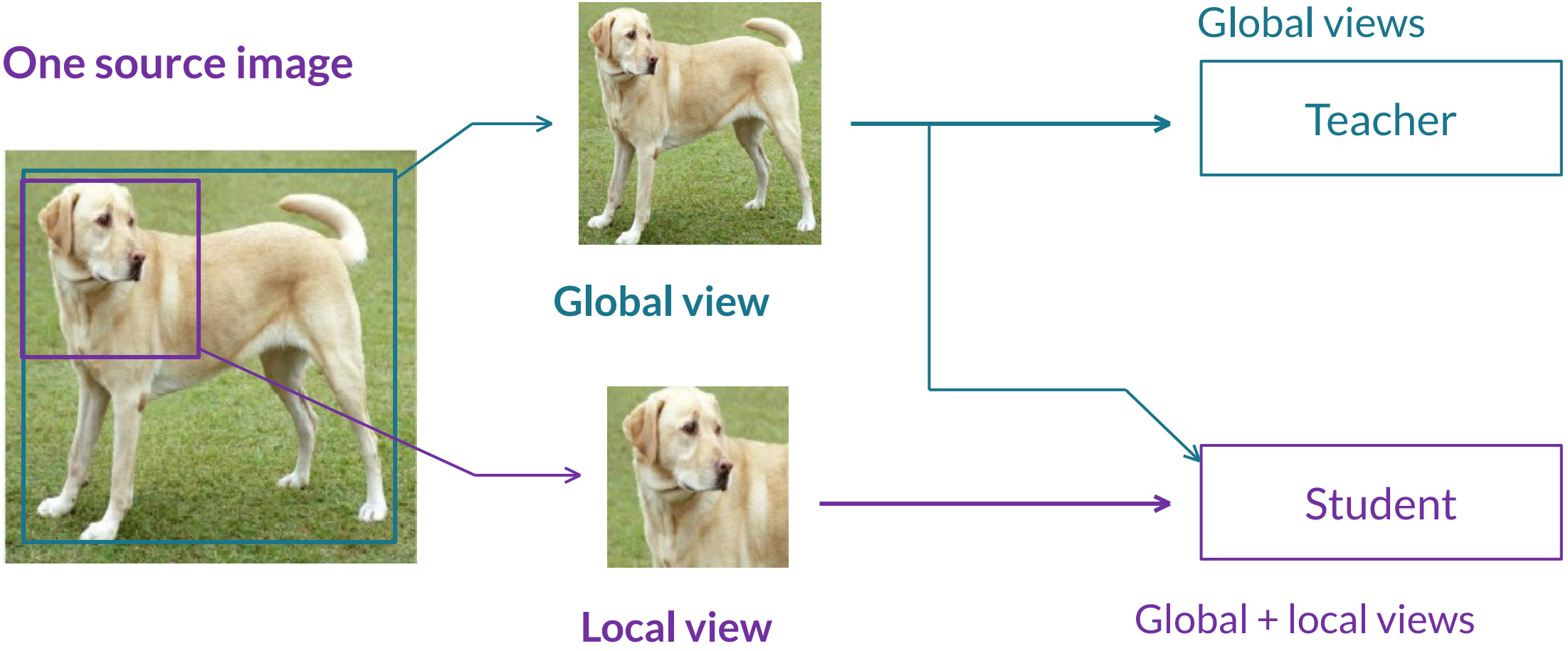


The K output coordinates have no predefined class names.

Source: SimCLR (image)

Global and local views

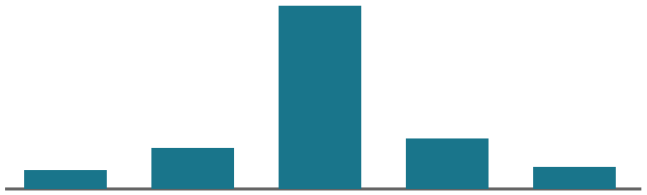
Predict a global-view target from another crop, including a small local view.



Source: DINO · SimCLR (image)

Matching teacher distributions

Teacher target



Student prediction

$$\mathcal{L} = -\sum_{k=1}^{K} p_{t,k} \log p_{s,k}$$

Cross-entropy loss

p_t: teacher probabilities, fixed for this gradient step

p_s: student probabilities, updated to match the teacher

Collapsed teacher targets

Same flat distribution

Same dominant coordinate

Different images

Neither target distinguishes the inputs.

Source: DINO (image) · SimCLR (image)

Centering and sharpening

Centering

Before softmax, subtract each coordinate's running mean across images.

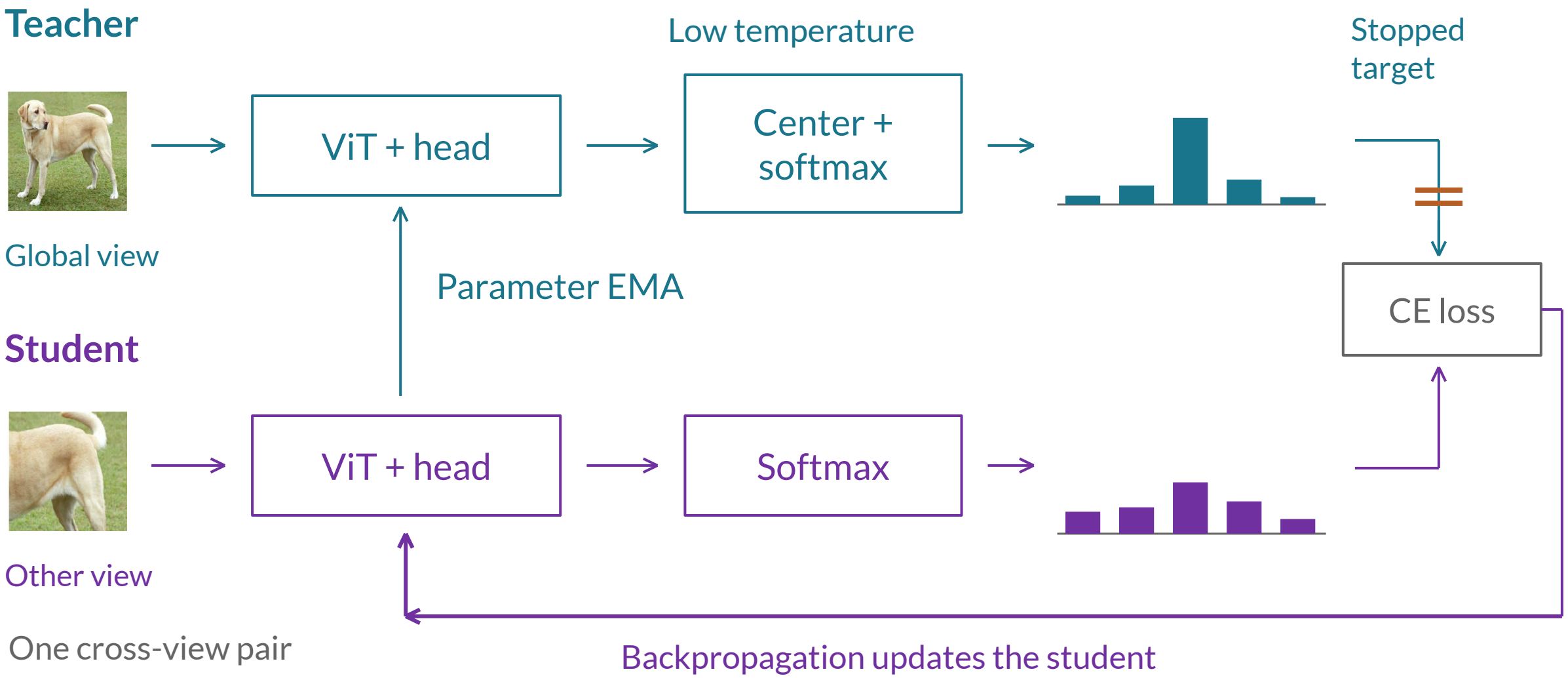
Discourage one coordinate from dominating every image.

Sharpening

Lower temperature amplifies score differences, opposing near-uniform targets.

$$p_t = \text{softmax}\left(\frac{\text{teacher scores} - \text{center}}{\text{teacher temperature}}\right)$$

DINO: one training step



Source: DINO · SimCLR (image)

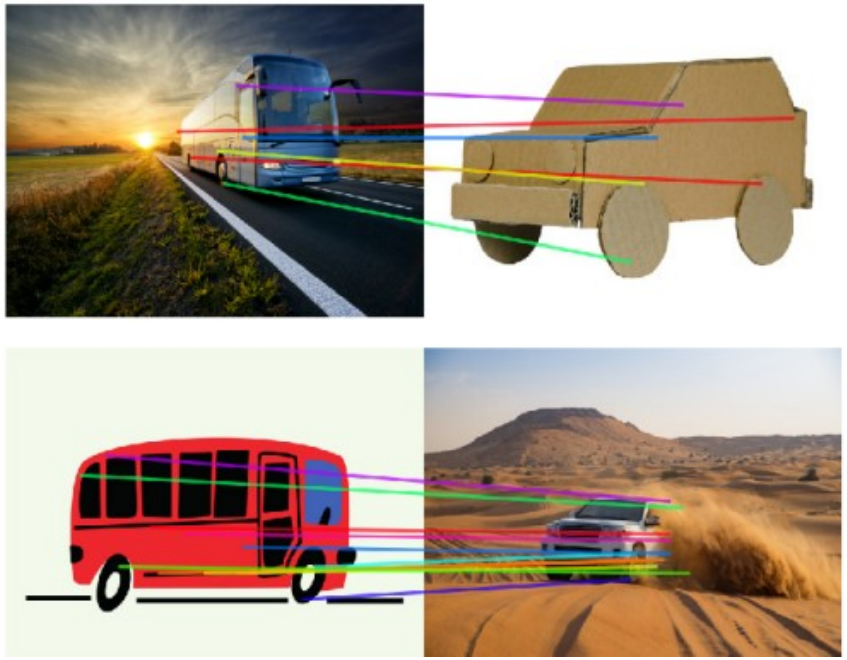
Attention learned without labels

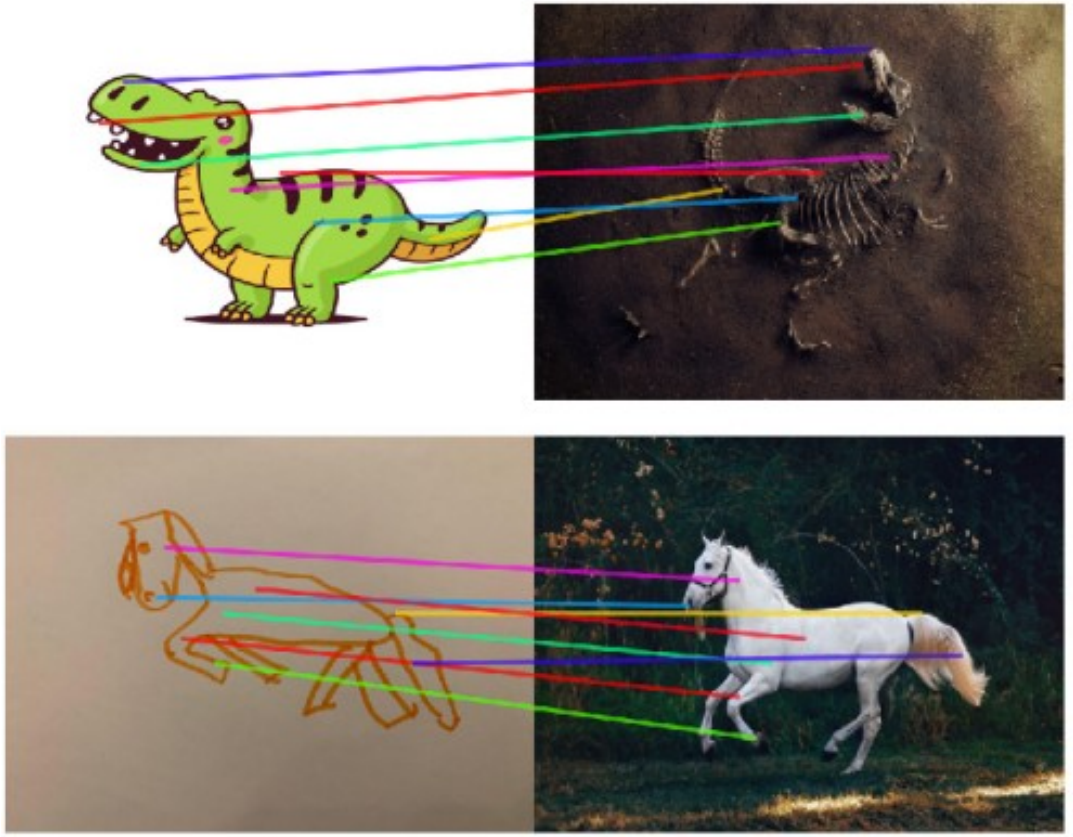
Input → last-layer [CLS] attention

Bright = larger attention weight.

Source: DINO, Figure 1

DINOv2: patch correspondences





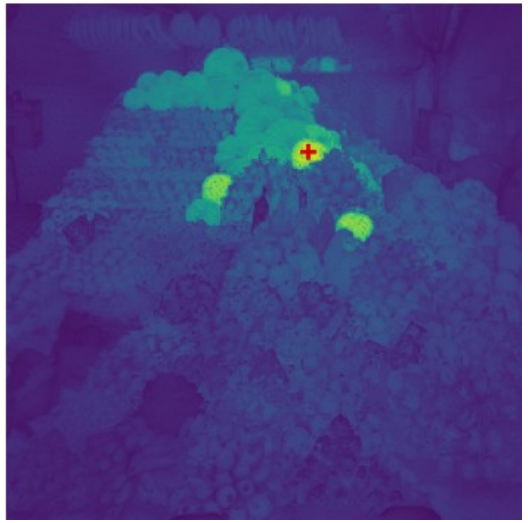
Match similar patch features
across appearance changes.

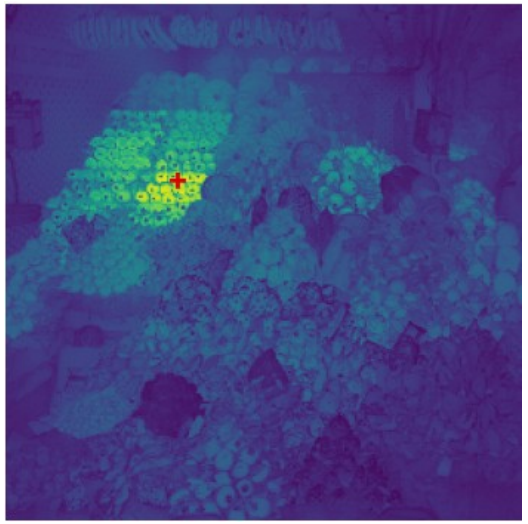
Source: DINOv2, Figure 10

DINOv3: dense image features

Input photograph

Feature similarity maps





Each cross marks a reference patch.

Bright = similar feature direction

Source: DINOv3, Figure 3

Lecture outline

01 Pretext tasks and masked prediction

02 Contrastive learning

03 Learning without negative pairs

04 Images and language

Paired images and text

Paired description

A shaggy dog
on a wooden floor.

Human-written web captions provide supervision without a fixed class list.

Source: Images: Masked Autoencoders · arXiv:2111.06377

Shared image–text representations

Image encoder

Project + normalize

Dot product

A shaggy dog on a floor.

Text encoder

Project + normalize

Compatible vectors: same dimension, learned to match across modalities.

Source: Images: Masked Autoencoders · arXiv:2111.06377

Matching an image–text batch

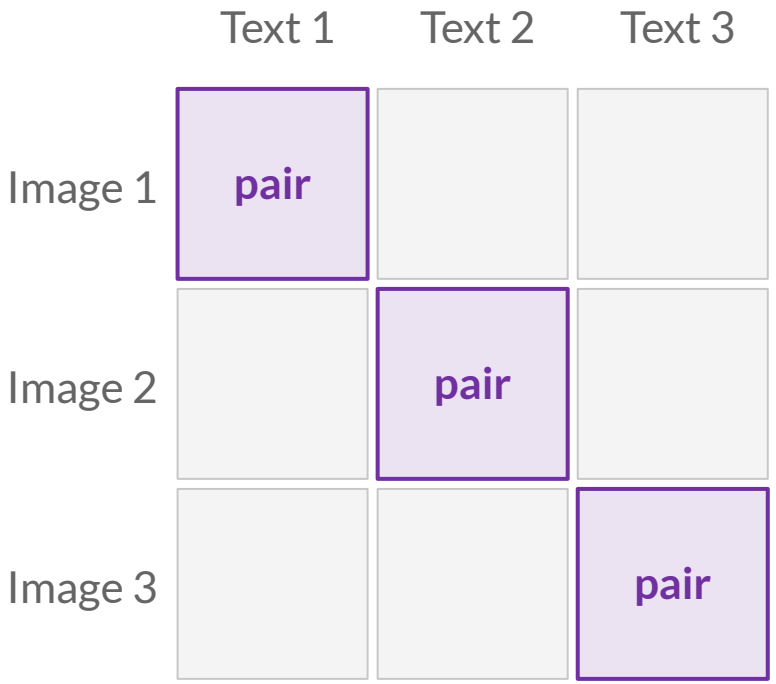
Example paired inputs

1 A shaggy dog

2 An old clock

3 A yellow tram

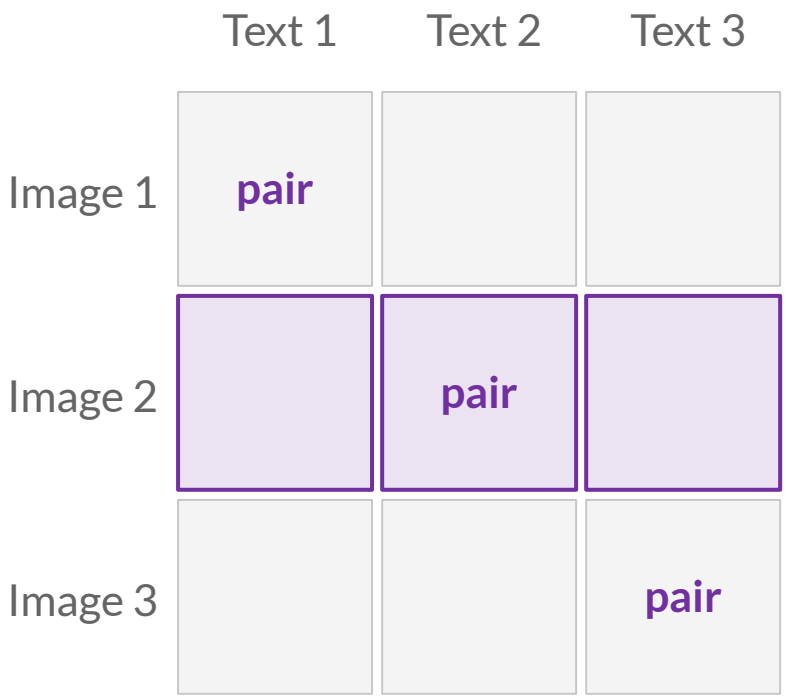
Rows: images
Columns: descriptions
Each cell: image–text similarity



Diagonal = observed pair

Source: Images: Masked Autoencoders · arXiv:2111.06377

Image-to-text matching



For Image 2, choose its paired text.

Compare the scores across one row.

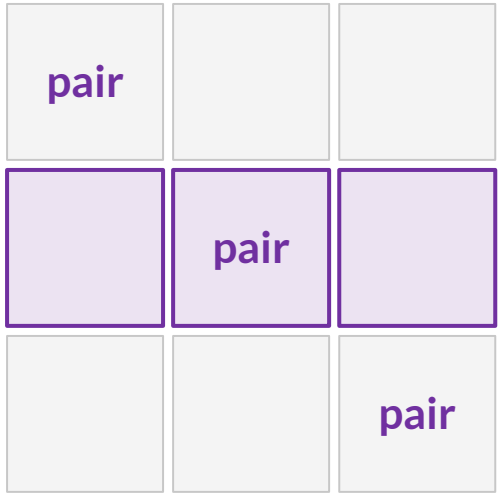
Softmax over candidate texts

$$\text{loss} = -\log p(\text{Text } 2 \mid \text{Image } 2)$$

The “classes” are descriptions in this batch.

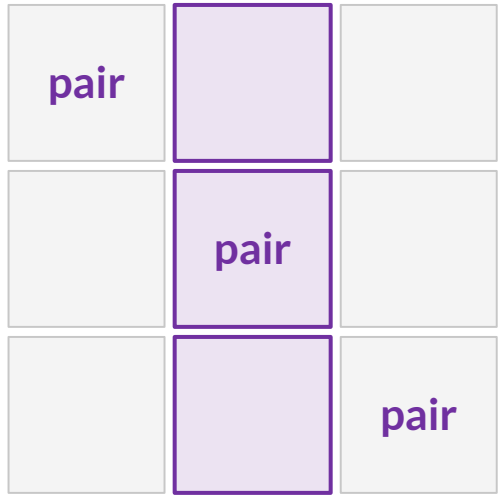
Matching in both directions

Image → text



Softmax across each row

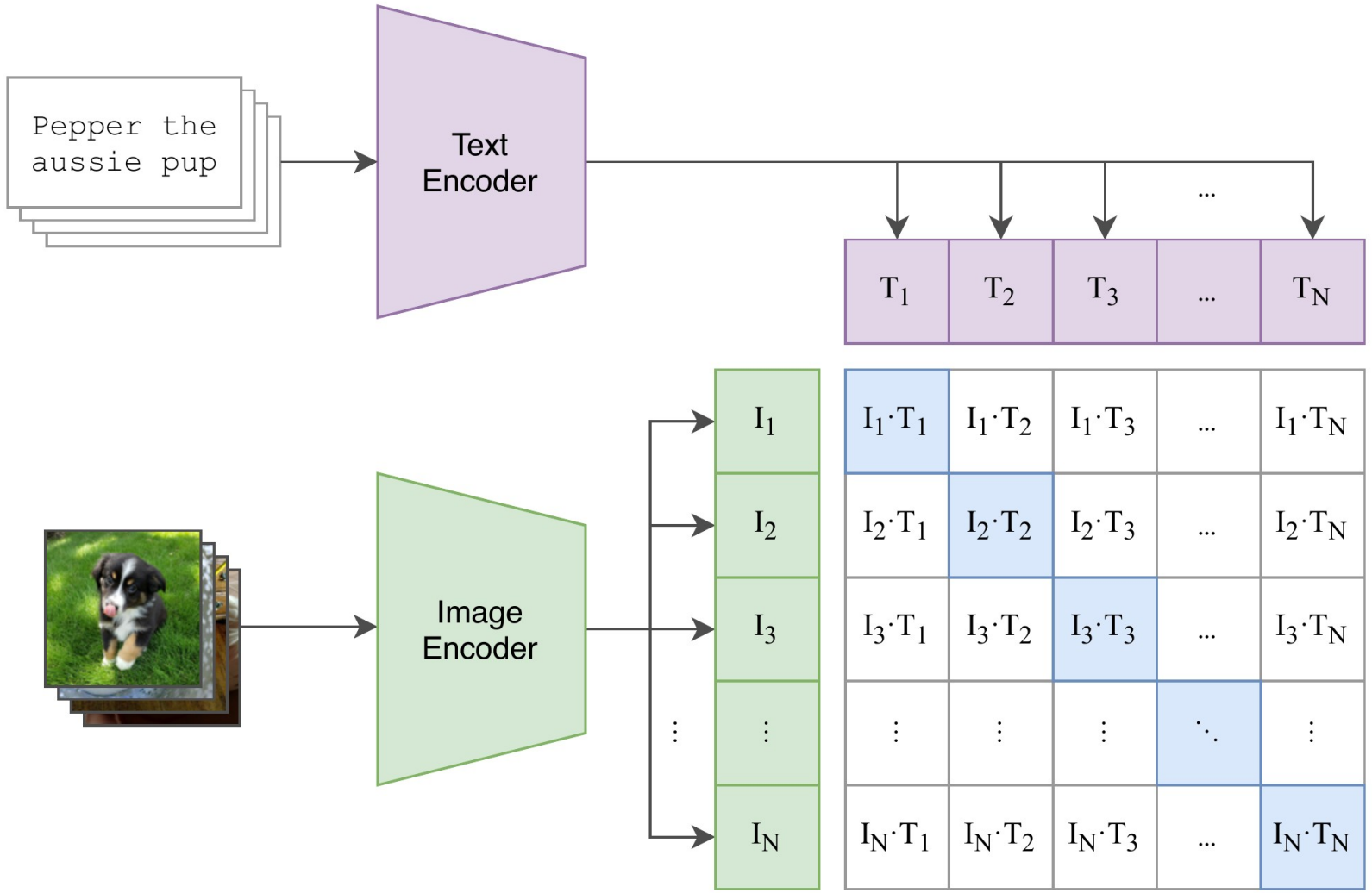
Text → image



Softmax down each column

$$\text{CLIP loss} = \tfrac{1}{2}\left(\text{image-to-text CE} + \text{text-to-image CE}\right)$$

CLIP pretraining



I: image features
T: text features

Blue diagonal: observed pairs

Cross-entropy across rows and columns trains both encoders.

Learn a scale for the normalized dot products.

Source: CLIP, Figure 1

Categories described in words

Trained linear classifier

Learn category vectors
from labeled images.

A new category set usually
requires training a new head.

CLIP zero-shot classifier

Encode a text description
for each category.

Change the descriptions
to specify a new category set.

Zero-shot classification

Image encoder

u

Normalized image vector

Candidate descriptions

a photo of a dog

a photo of a clock

a photo of a tram

Text encoder + normalize

v_{dog} v_{clock} v_{tram}

$u^\top v_{\text{dog}}$ $u^\top v_{\text{clock}}$ $u^\top v_{\text{tram}}$

Select the highest similarity

Prediction: dog

Zero-shot: no labeled examples from the target task.

Source: Adapted from CLIP · Image: Masked Autoencoders

Text descriptions matter

Category name	dog
Visual context	a photo of a dog

Domain-specific context

a satellite image of a residential area

Caption-like phrases resemble the text used during pretraining.

Image–text retrieval

Example query

a shaggy dog

Encode the gallery images once.
Compare each new text query
with the stored image features.

Image gallery

Matching image

Source: Images: Masked Autoencoders · arXiv:2111.06377

Partial supervision from captions

A shaggy dog

Observed description

An animal with long fur

Also compatible with the image

Unpaired does not always mean semantically wrong.

Source: Images: Masked Autoencoders · arXiv:2111.06377

Objects and their relationships

Plants surrounding a lightbulb

A lightbulb surrounding plants

Same objects and words. Different relationships.

CLIP models evaluated in Winoground struggled with these relationships.

Source: Winoground · arXiv:2204.03162

Learning targets and learned features

Missing content	Hidden tokens or pixels Infer what is hidden
Image views	Another view's features Preserve shared information
Image–text pairs	The matching caption / image Connect visual and language features

Self-supervision and generative modeling

Next-token language model

Targets come from text.

Sampling successive tokens generates a continuation.

Contrastive image encoder

Pairs come from image views.

A feature vector alone does not specify an image generator.

Target source and generative capability are different properties.

Next time: autoencoders and generative models

Autoencoders

Learn a representation by reconstructing the input.

Variational autoencoders

Learn a probabilistic model that can generate new examples.