Contrastive learning

Week 7 of MATH70134

Mathematical Foundations of Machine Learning

Dr. Max Weissenbacher

Lecture overview

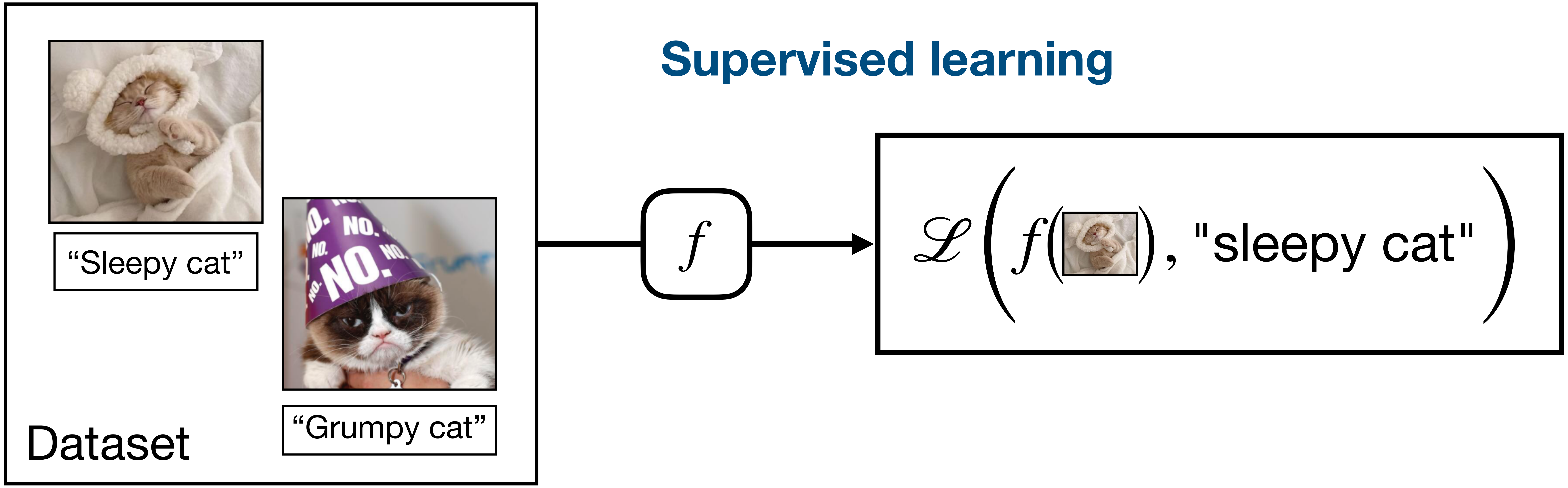
- What are self supervised and contrastive learning?
- Examples and applications
- A closer look: normalisation, batch size, number of negative samples
- What makes contrastive loss work so well?

Self-supervised learning

- … is a paradigm in machine learning where a model is trained using only the data itself without access to external labels

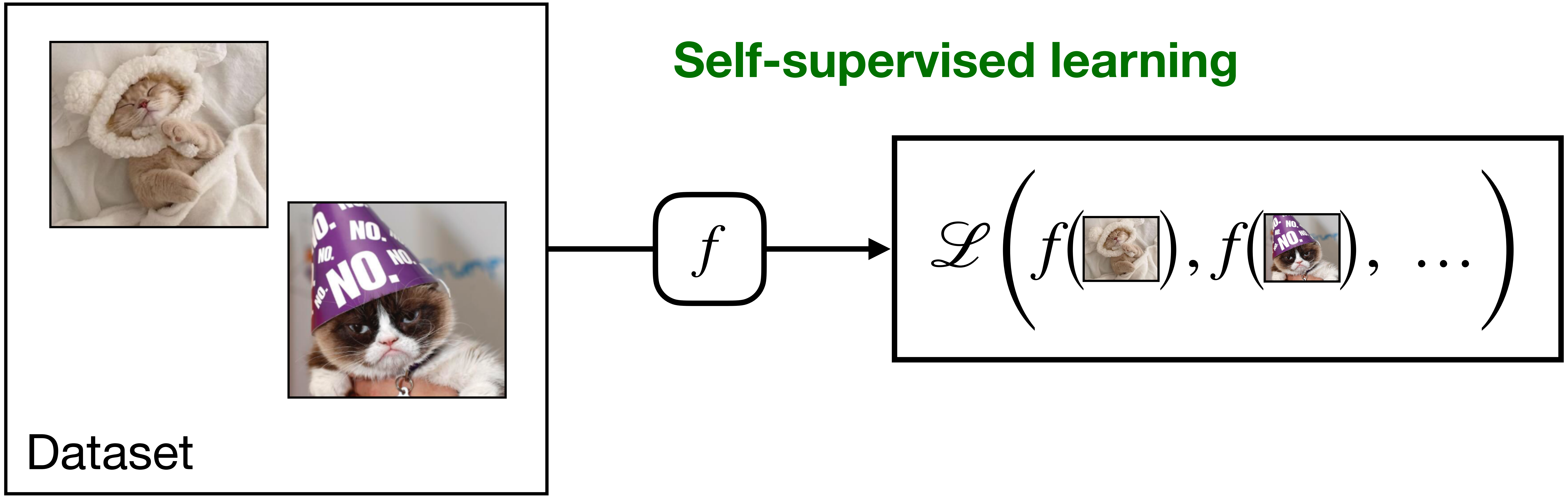
Self-supervised learning

- … is a paradigm in machine learning where a model is trained using only the data itself without access to external labels



Self-supervised learning

- … is a paradigm in machine learning where a model is trained using only the data itself without access to external labels



Types of self-supervised learning

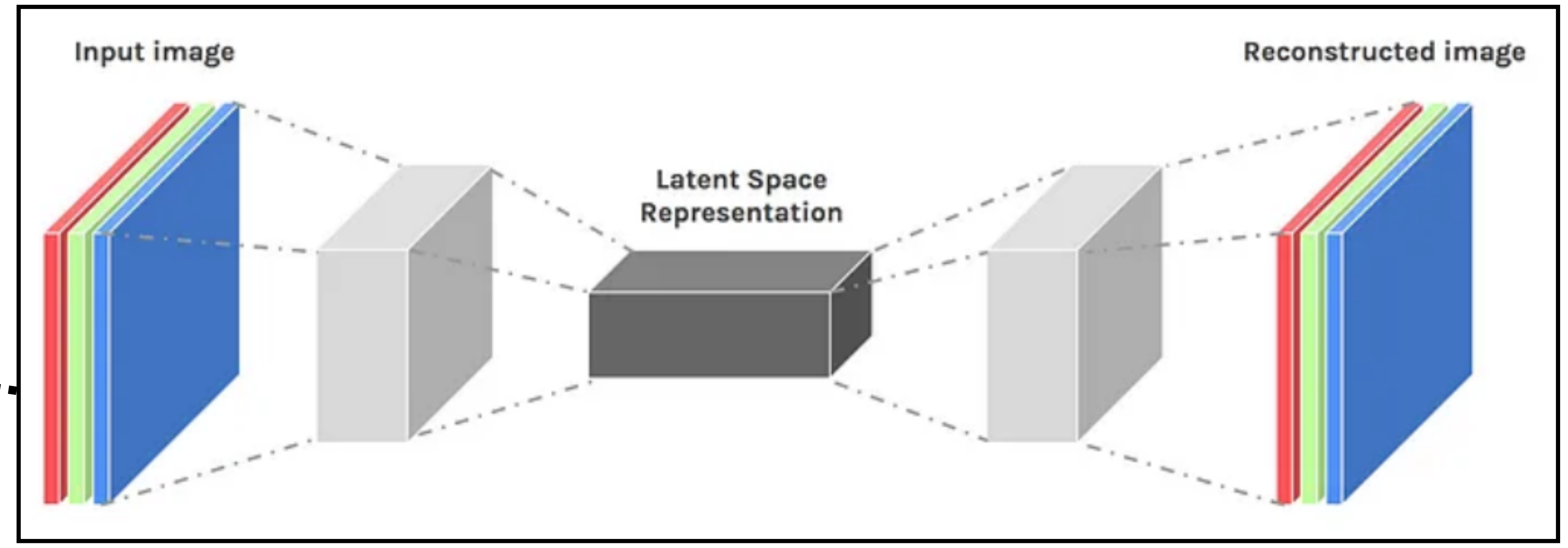
Autoencoders

Contrastive learning

Non-contrastive learning

Types of self-supervised learning

Autoencoders

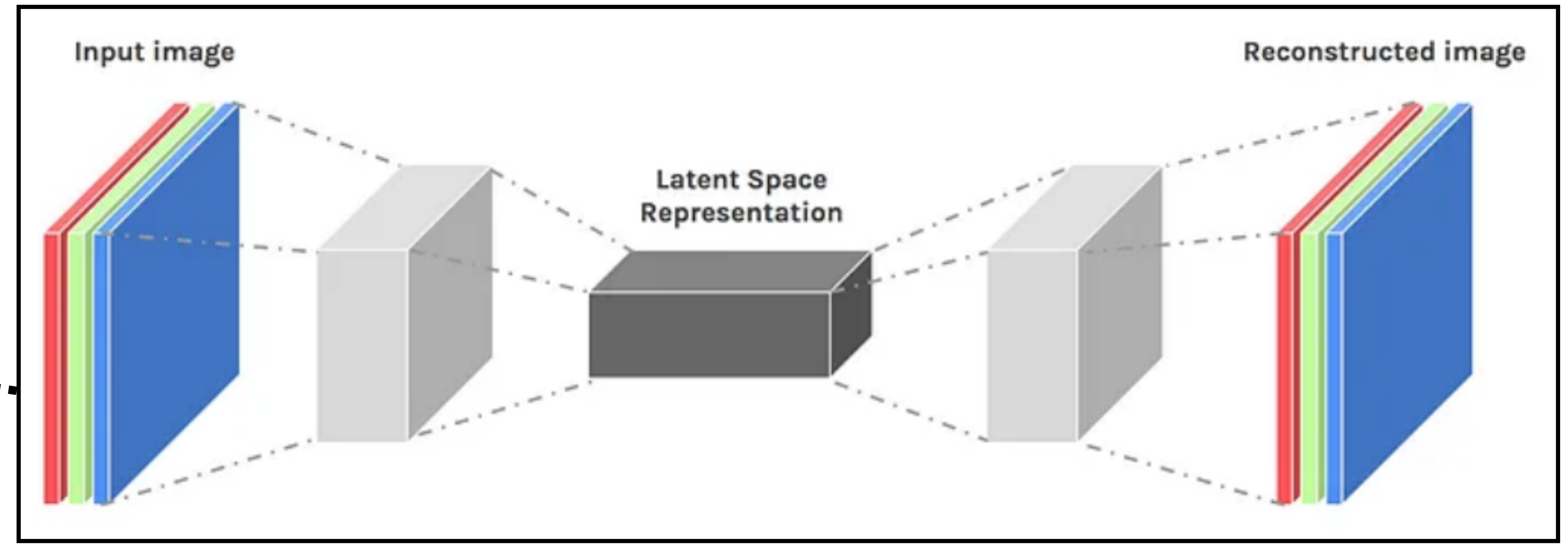


Contrastive learning

Non-contrastive learning

Types of self-supervised learning

Autoencoders



Contrastive learning

Non-contrastive learning

Why self-supervised learning?

- Data labelling is expensive and high-quality labeled data is limited
- Learning good representations facilitates downstream tasks with fewer labeled data (*few-shot learning*) or transfer to new tasks
- Learning good representations enables better generalisation
- More closely imitates the way humans learn to classify objects

Contrastive learning in the news 📰

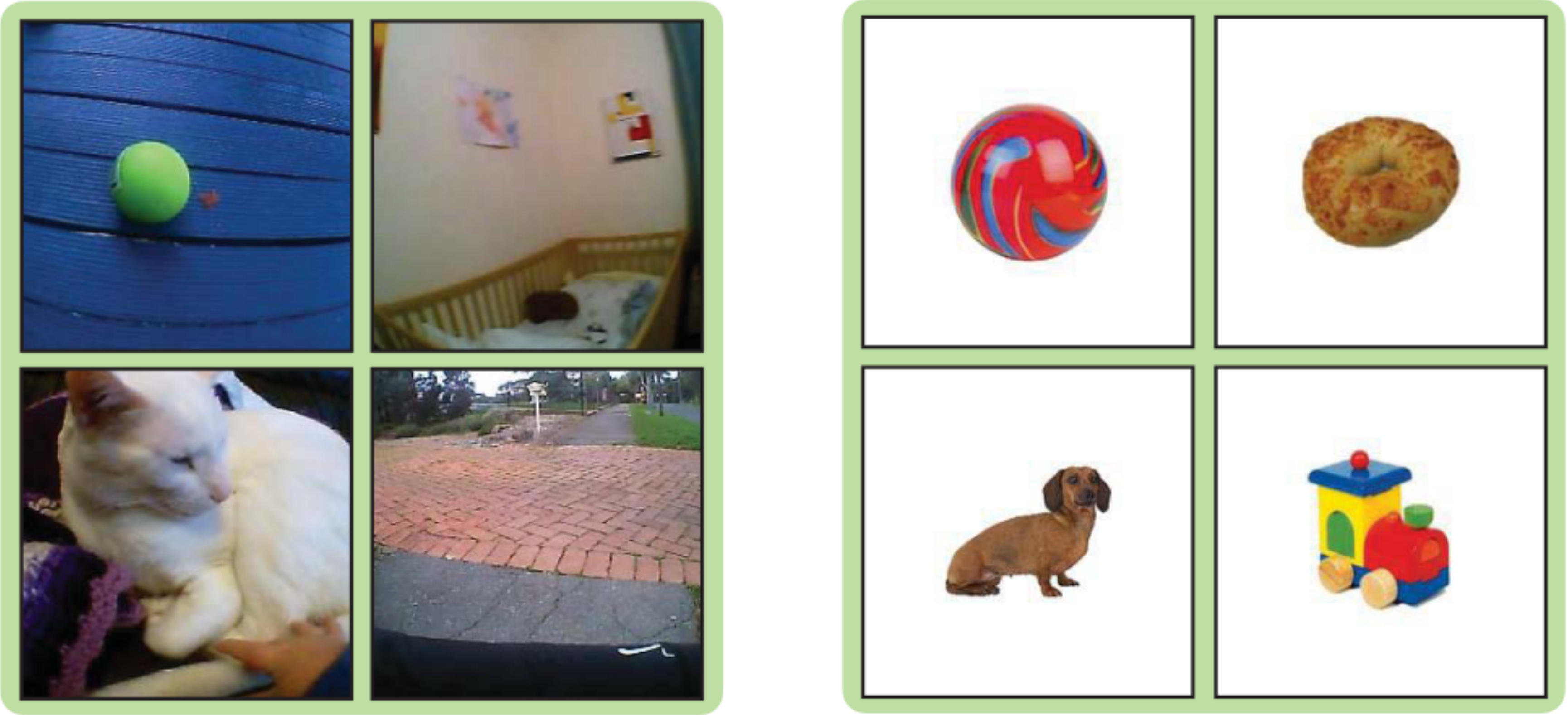
AI MODEL LEARNT LANGUAGE BY SEEING THE WORLD LIKE A BABY

A neural network taught itself to recognize objects using the filmed experiences of a single infant.

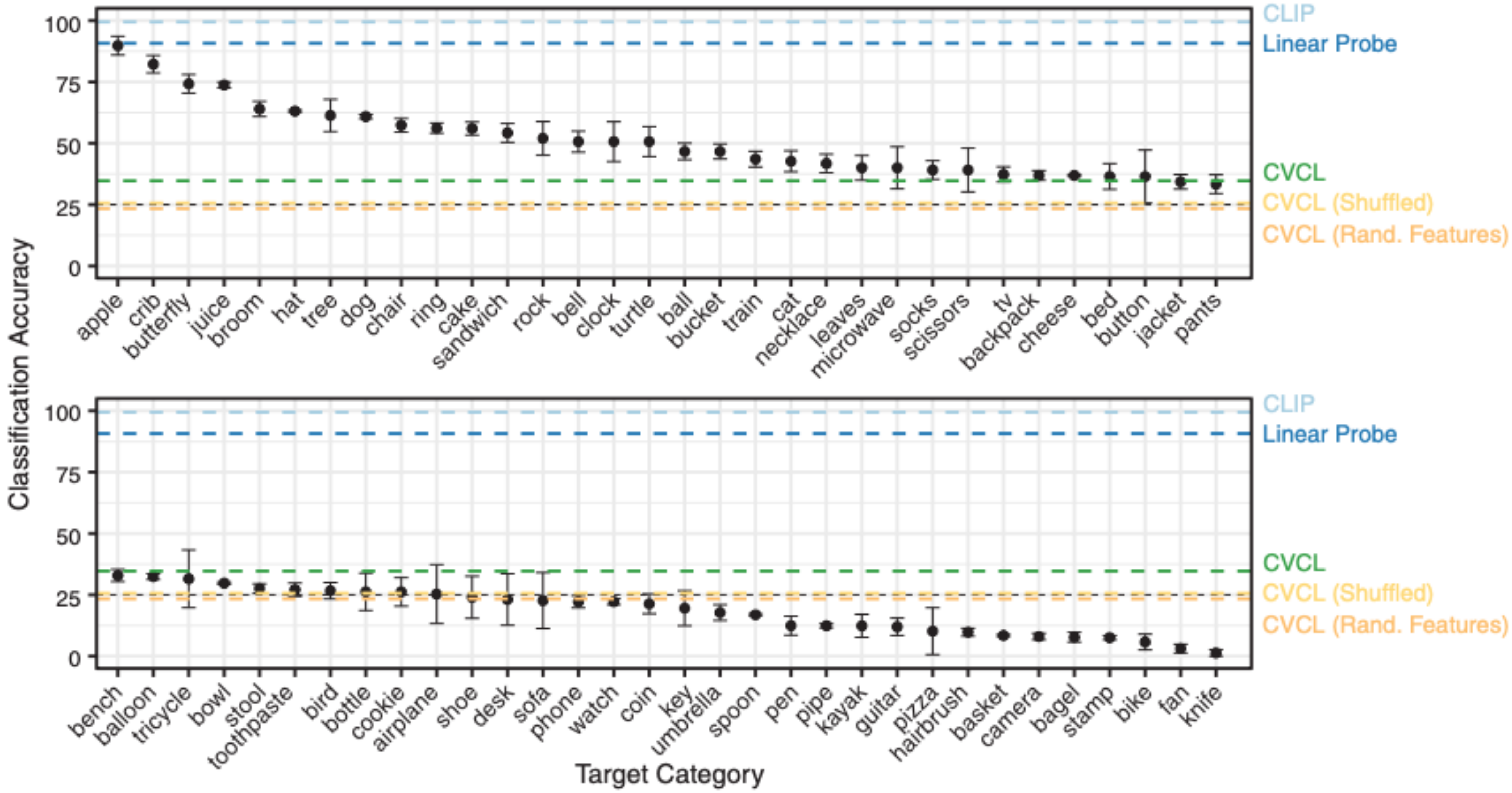
Sam — here aged 18 months — wore a camera whose recordings trained an AI model.

References: [Vong et al. | Science ’24], [Article in Nature, ‘24]

Contrastive learning in the news 📰

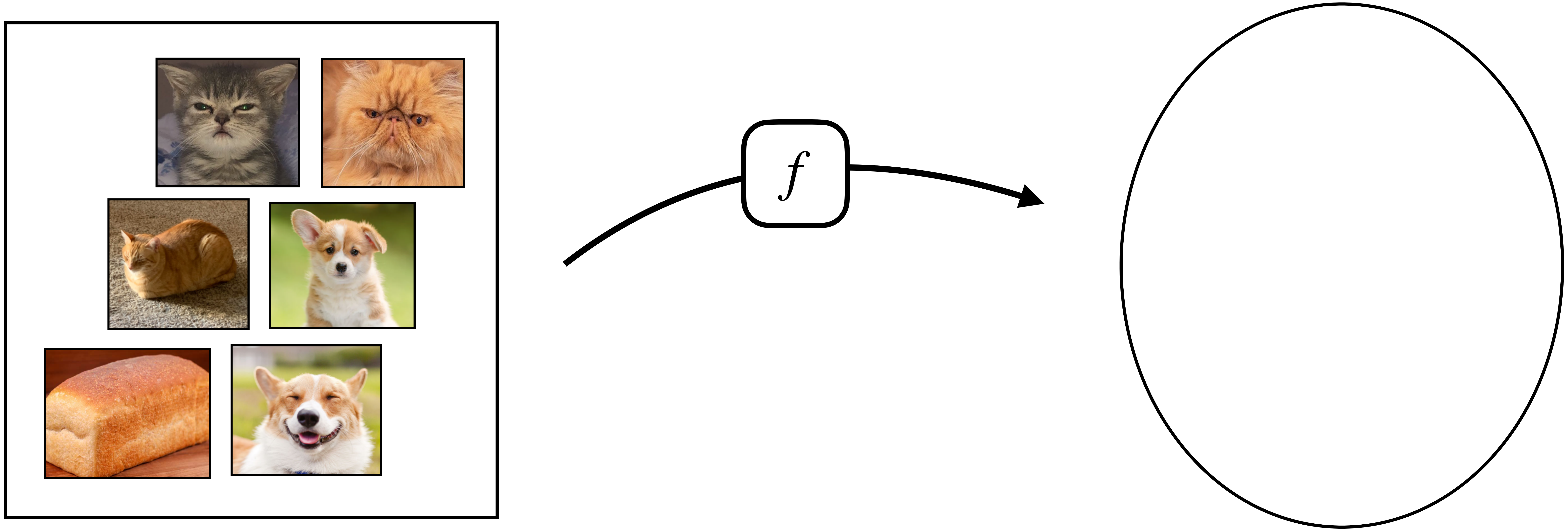


Task: Which one is the **ball**?



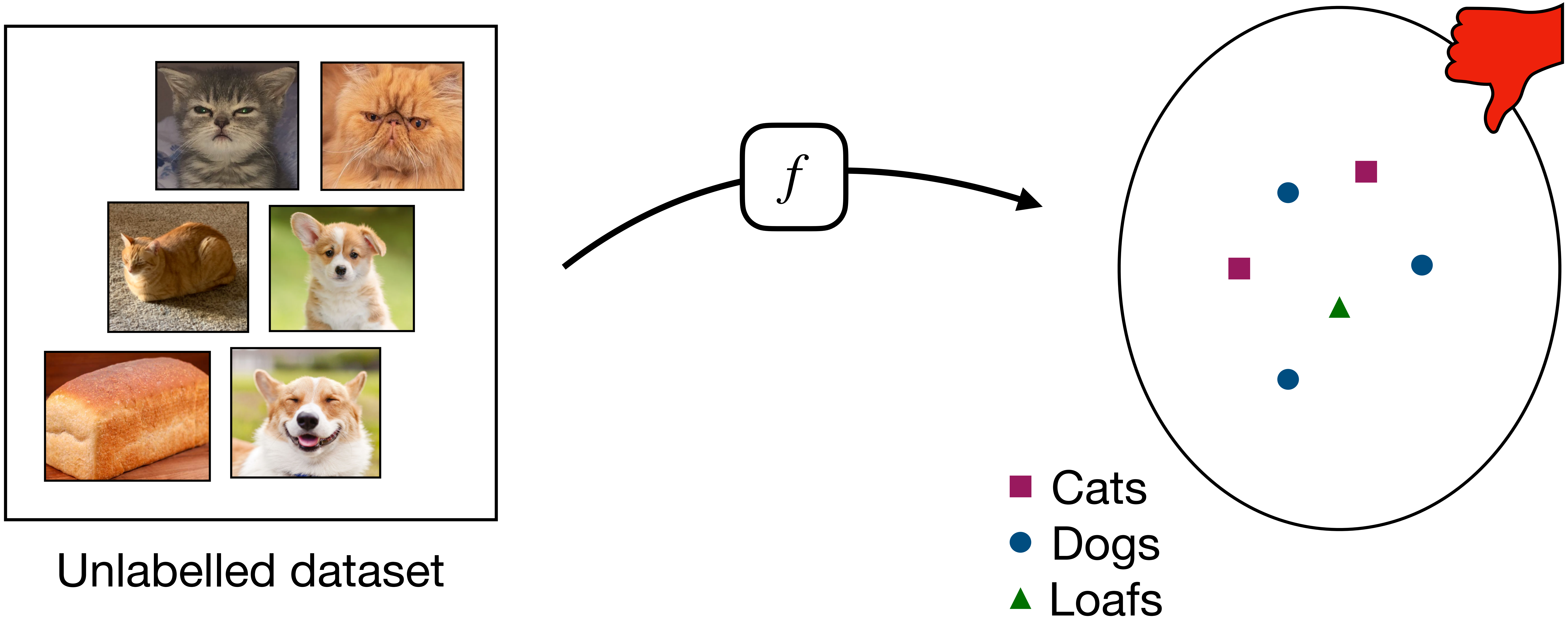
References: [Vong et al. | Science '24], [Article in Nature, '24]

Introduction: contrastive loss in pictures

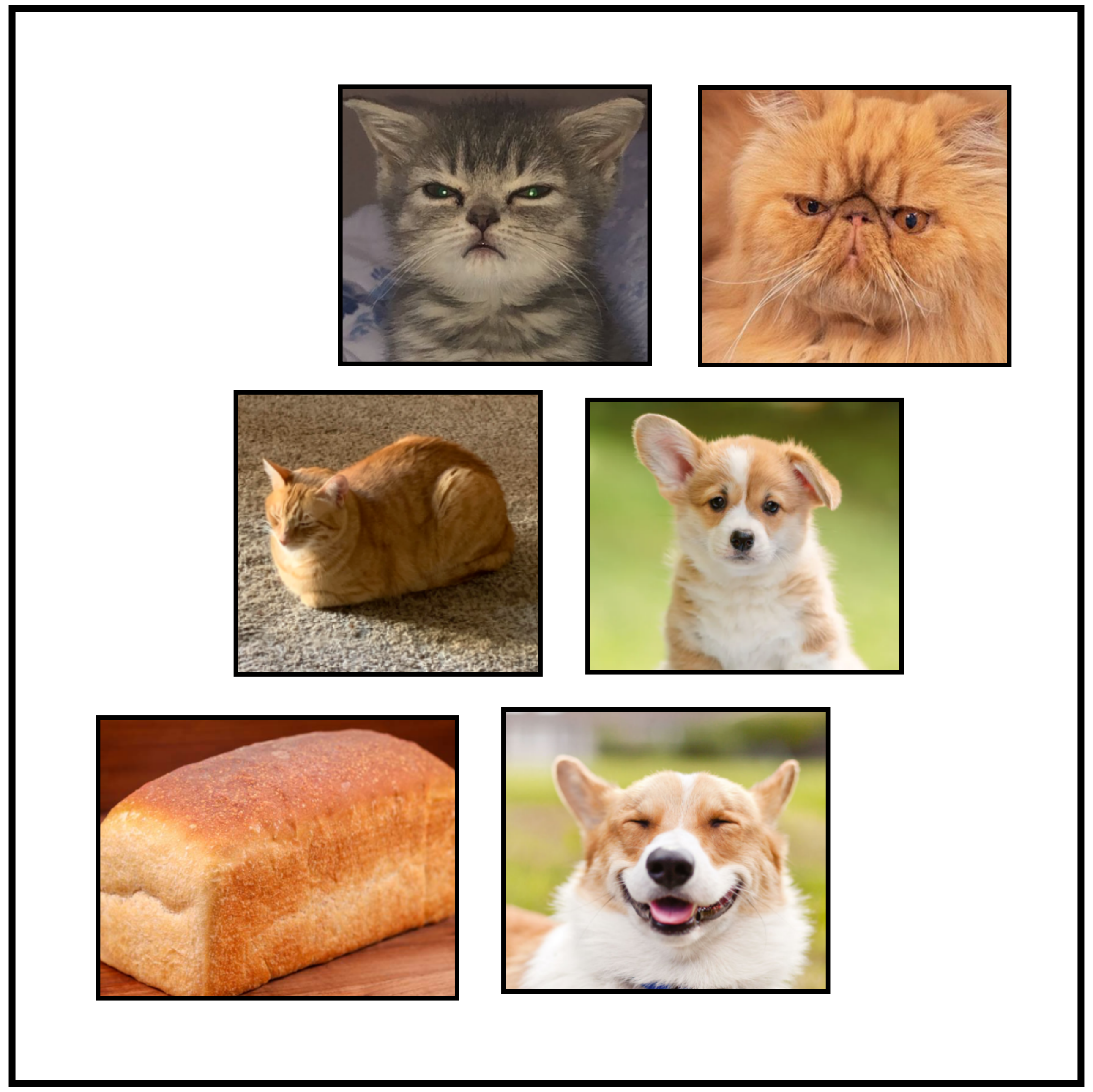


Unlabelled dataset

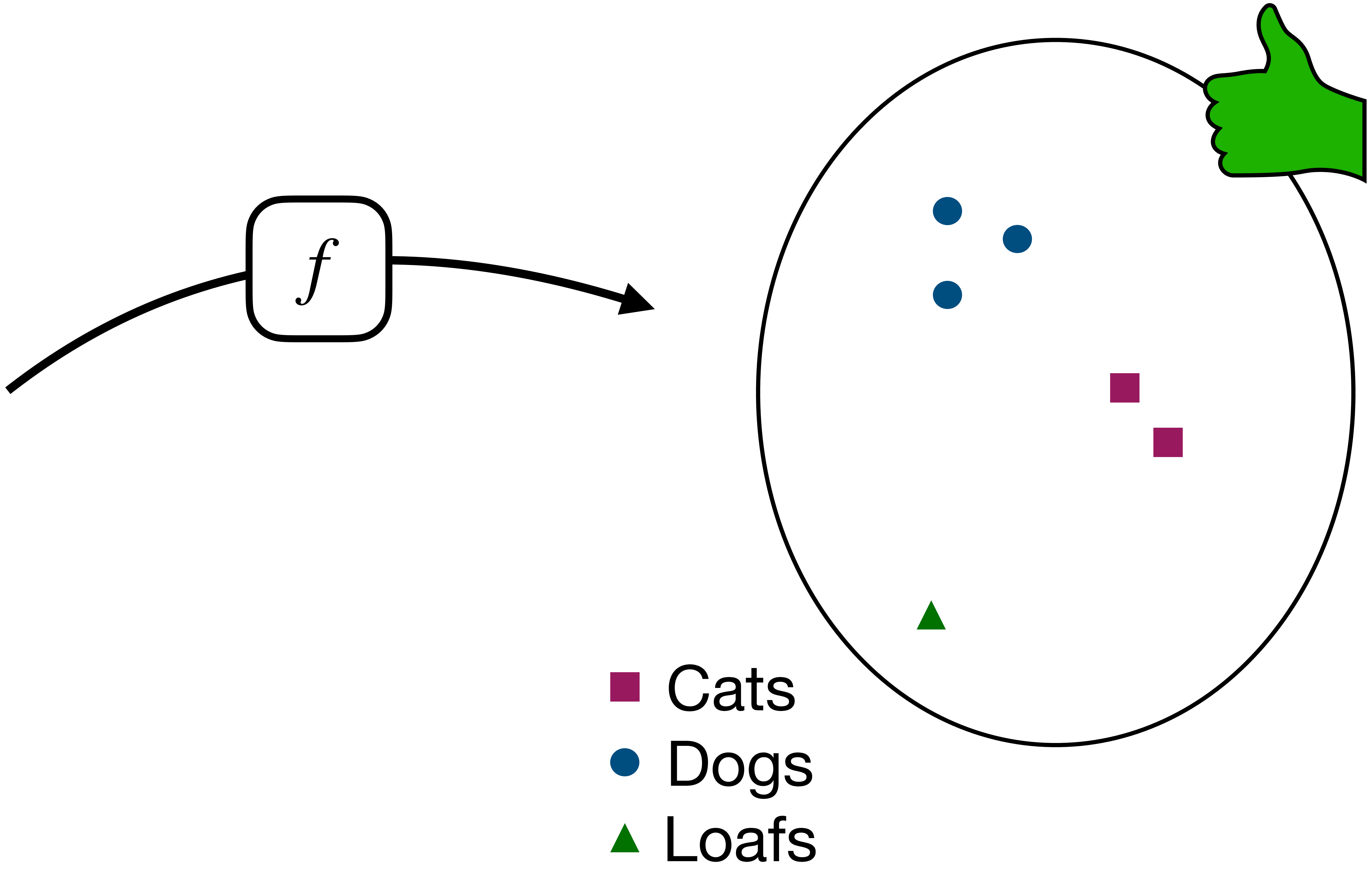
Introduction: contrastive loss in pictures



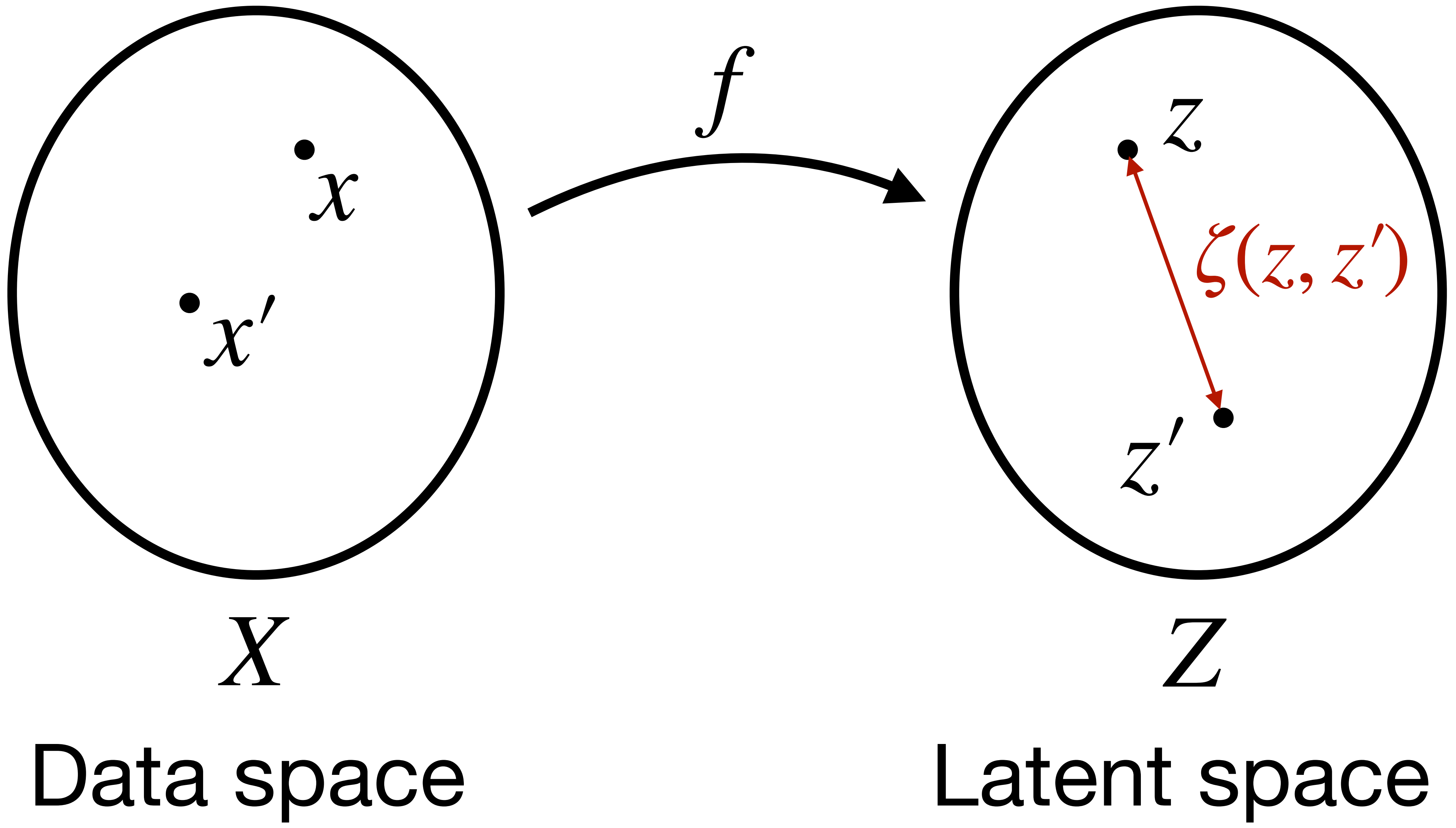
Introduction: contrastive loss in pictures



Unlabelled dataset

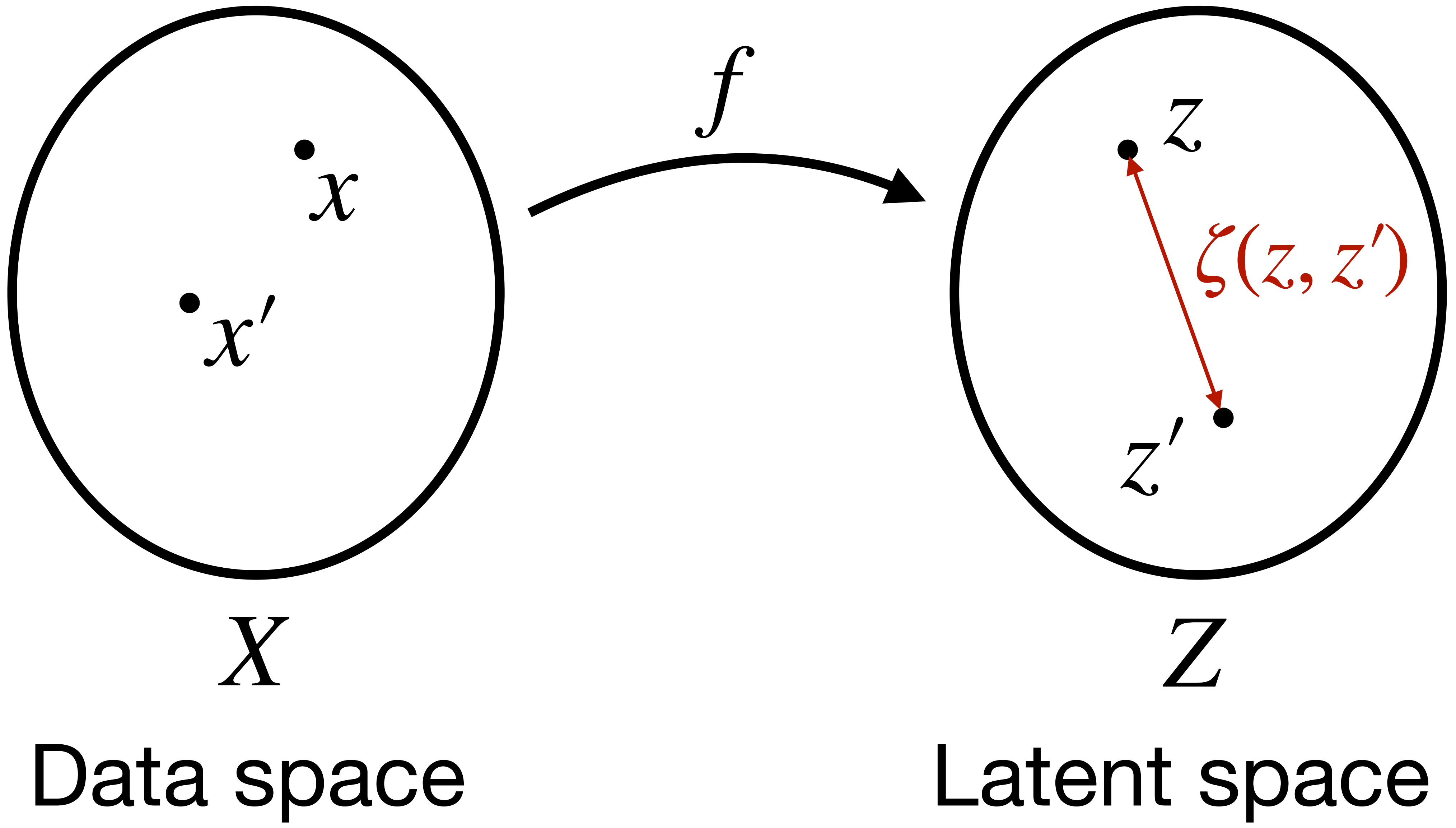


Defining contrastive loss in generality



- Model $f : X \to Z$ (i.e. neural net)

Defining contrastive loss in generality

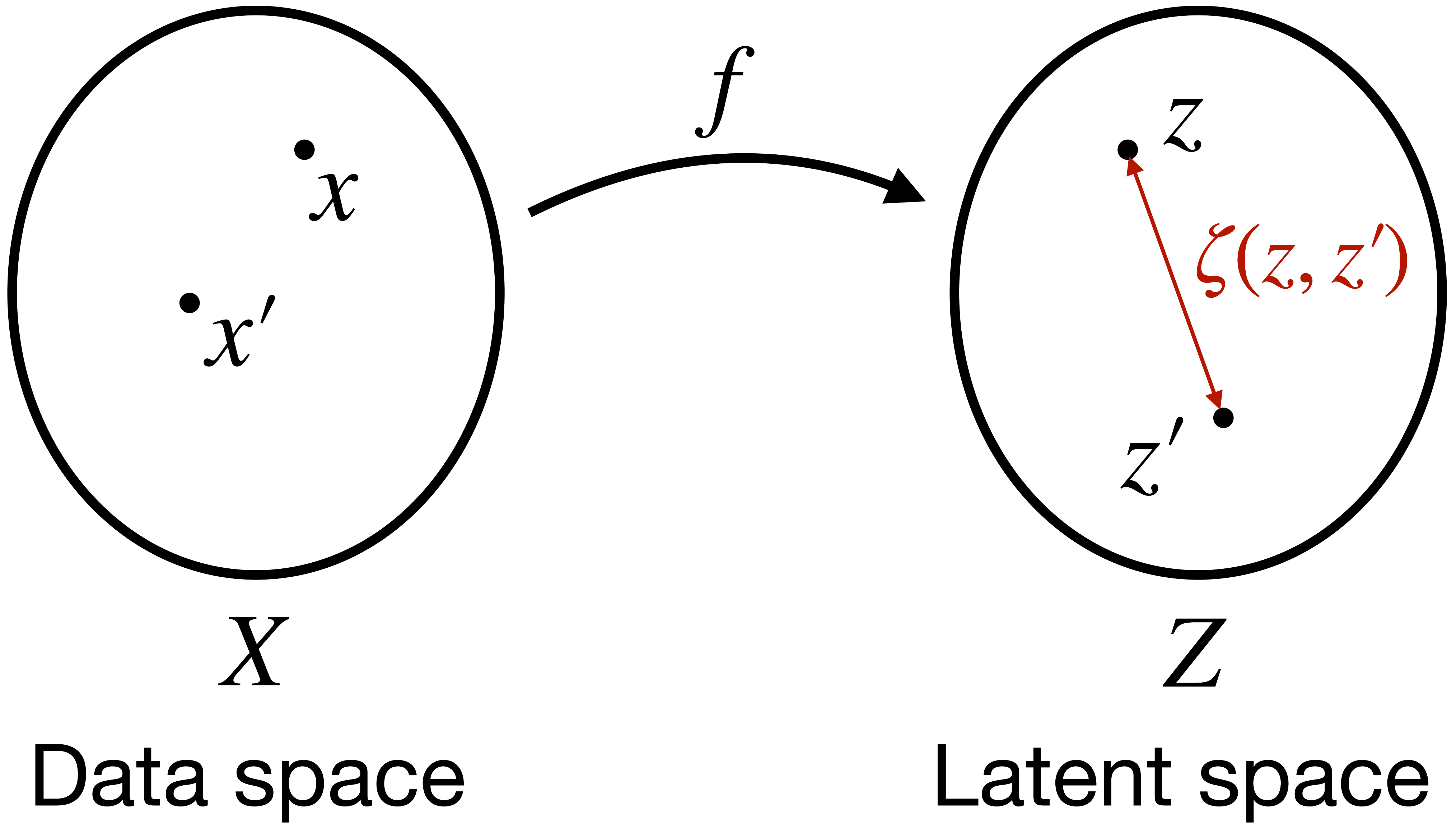


- Model $f: X \to Z$ (i.e. neural net)
- Z equipped with similarity metric ζ
- Common choices for Z and ζ:

$$Z = \mathbb{R}^d, \quad \zeta(z, z') = ||z - z'||^2$$

$$Z = \mathbb{S}^d, \quad \zeta(z, z') = \frac{z^T z'}{||z||\,||z'||}$$

Defining contrastive loss in generality



Shorthand notation: $d_{x,y} = \zeta(f(x), f(y))$

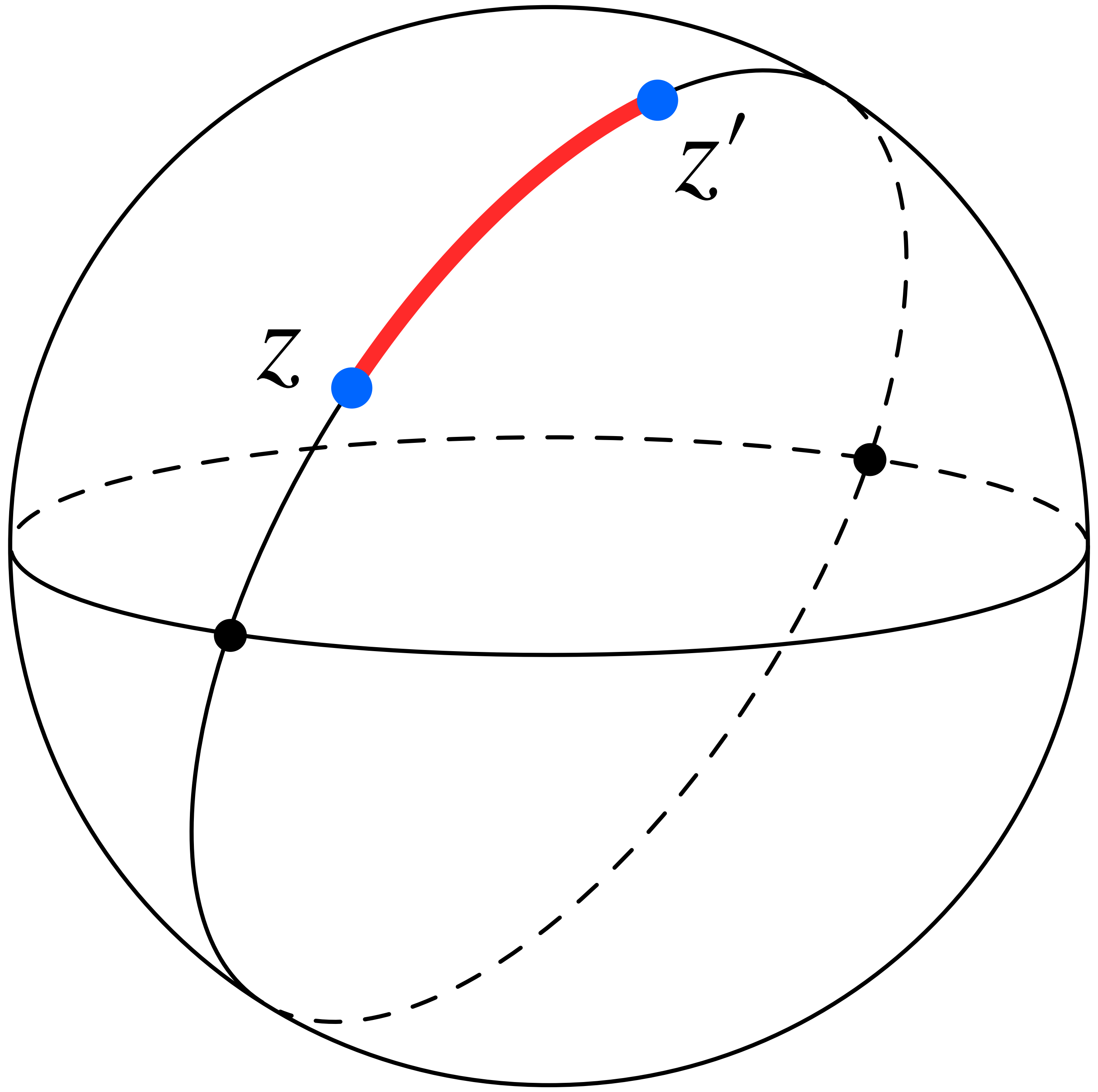
- Model $f : X \to Z$ (i.e. neural net)
- Z equipped with similarity metric ζ
- Common choices for Z and ζ:

$$Z = \mathbb{R}^d, \quad \zeta(z, z') = \|z - z'\|^2$$

$$Z = \mathbb{S}^d, \quad \zeta(z, z') = \frac{z^T z'}{\|z\| \|z'\|}$$

Cosine similarity

- Recall $\mathbb{S}^{d-1} = \{z \in \mathbb{R}^d \mid \|z\| = 1\} \subset \mathbb{R}^d$
- Cosine similarity $\zeta(z, z') = z^T z'$

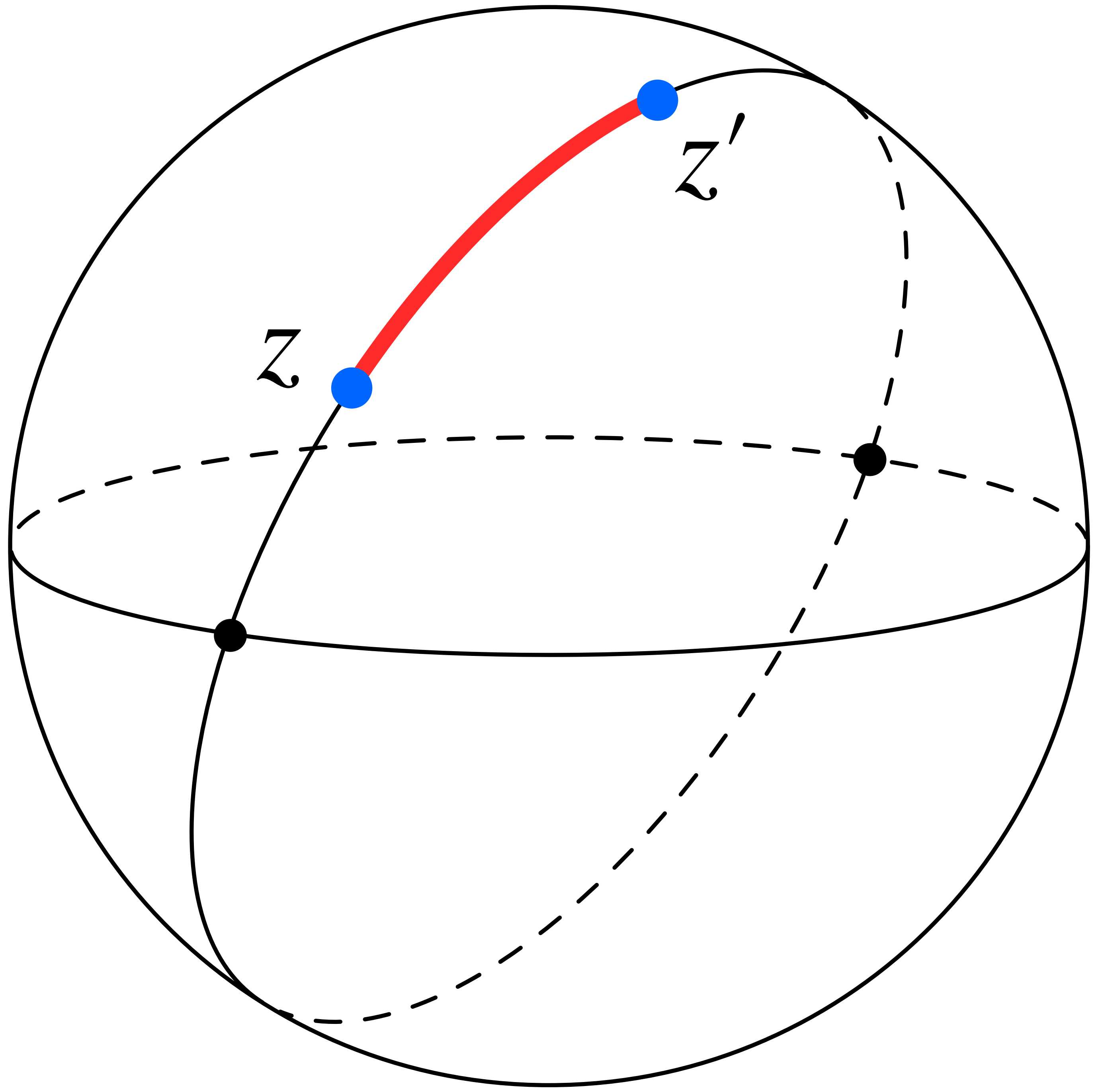


Depiction of the 2-sphere $\mathbb{S}^2$

References: [Wiki article on cosine similarity]

Cosine similarity

- Recall $\mathbb{S}^{d-1} = \{z \in \mathbb{R}^d \mid \|z\| = 1\} \subset \mathbb{R}^d$
- Cosine similarity $\zeta(z, z') = z^T z'$
- If $z = z'$, then $\zeta(z, z') = 1$
- If $z = -z'$, then $\zeta(z, z') = -1$

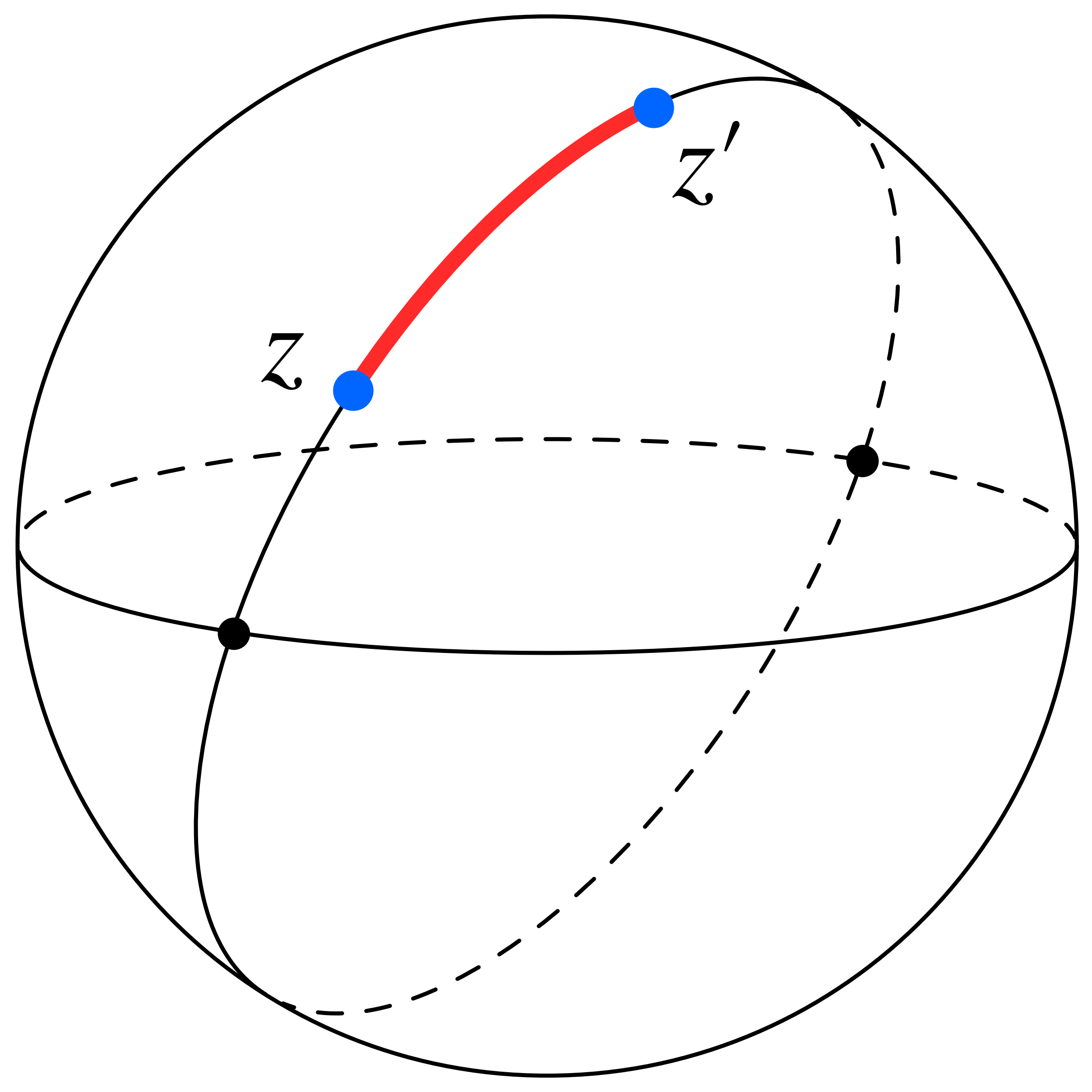


Depiction of the 2-sphere $\mathbb{S}^2$

References: [Wiki article on cosine similarity]

Cosine similarity

- Most often, models output $z \in \mathbb{R}^d$

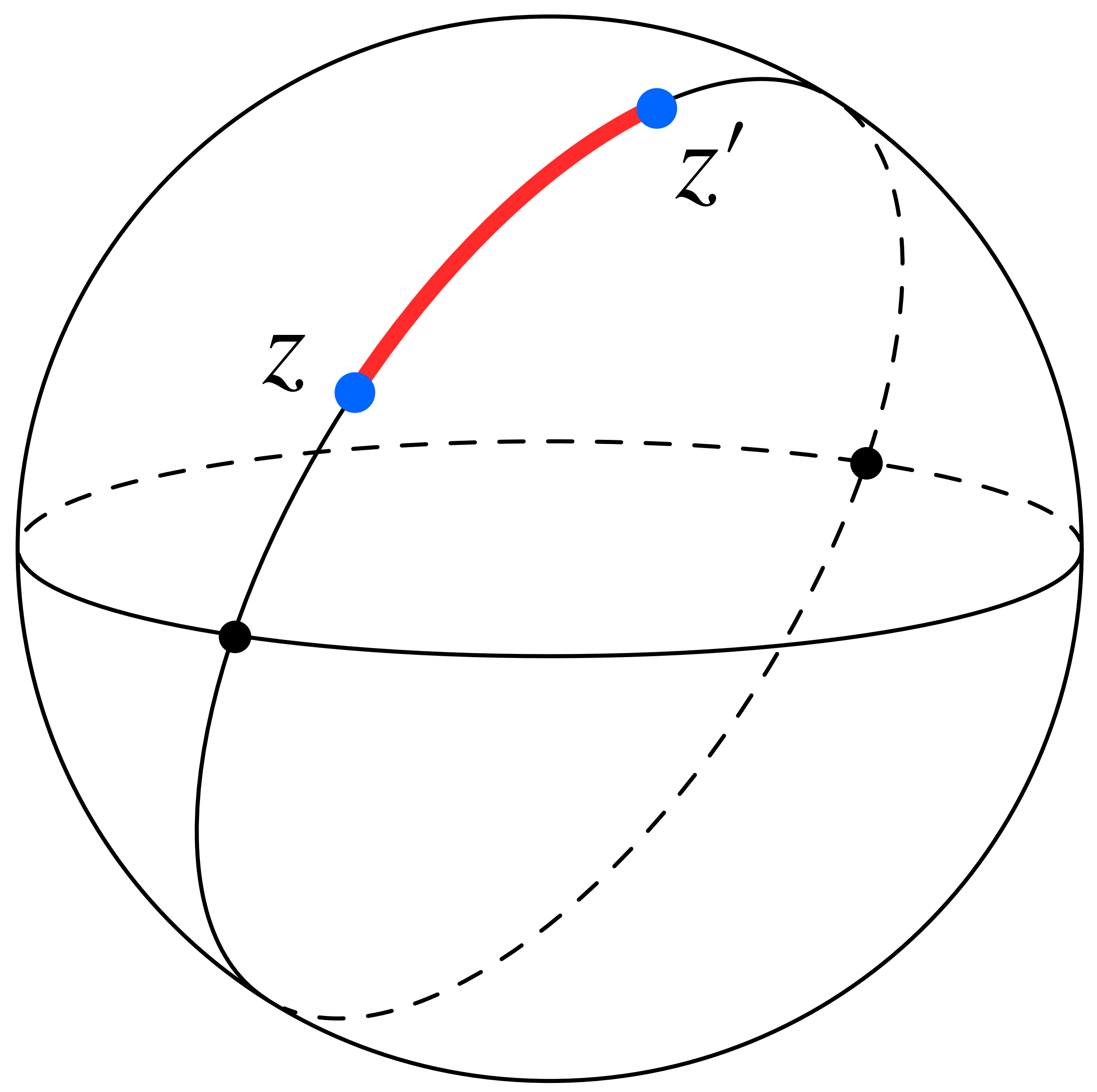


References: [Wiki article on cosine similarity]

Cosine similarity

- Most often, models output $z \in \mathbb{R}^d$
- We first project to the sphere by mapping

$$z \to \frac{z}{||z||} \qquad \Longrightarrow \qquad \zeta(z, z') = \frac{z^T z'}{||z|| \, ||z'||}$$



References: [Wiki article on cosine similarity]

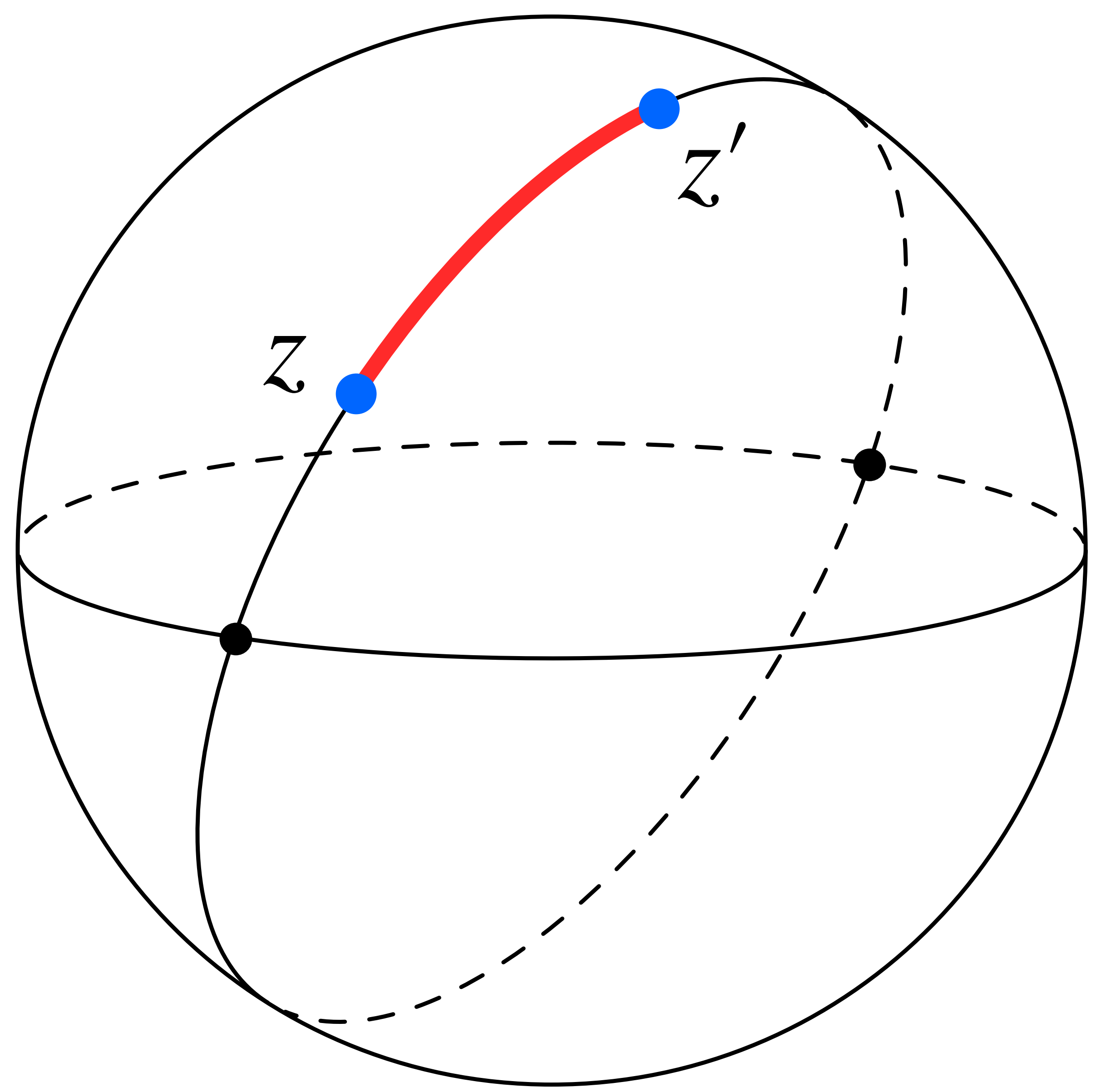
Cosine similarity

- Most often, models output $z \in \mathbb{R}^d$
- We first project to the sphere by mapping

$$z \to \frac{z}{\|z\|} \quad \Longrightarrow \quad \zeta(z, z') = \frac{z^T z'}{\|z\| \, \|z'\|}$$

- $\exists\, \phi : [-1,1] \to [0,\infty)$ strictly increasing such that

$$\phi(\zeta(z, z')) = \inf \left\{ \int_0^1 |\dot{\gamma}(s)| \, ds \mid \gamma : [0,1] \to \mathbb{S}^{d-1}, \gamma(0) = z, \gamma(1) = z' \right\}$$



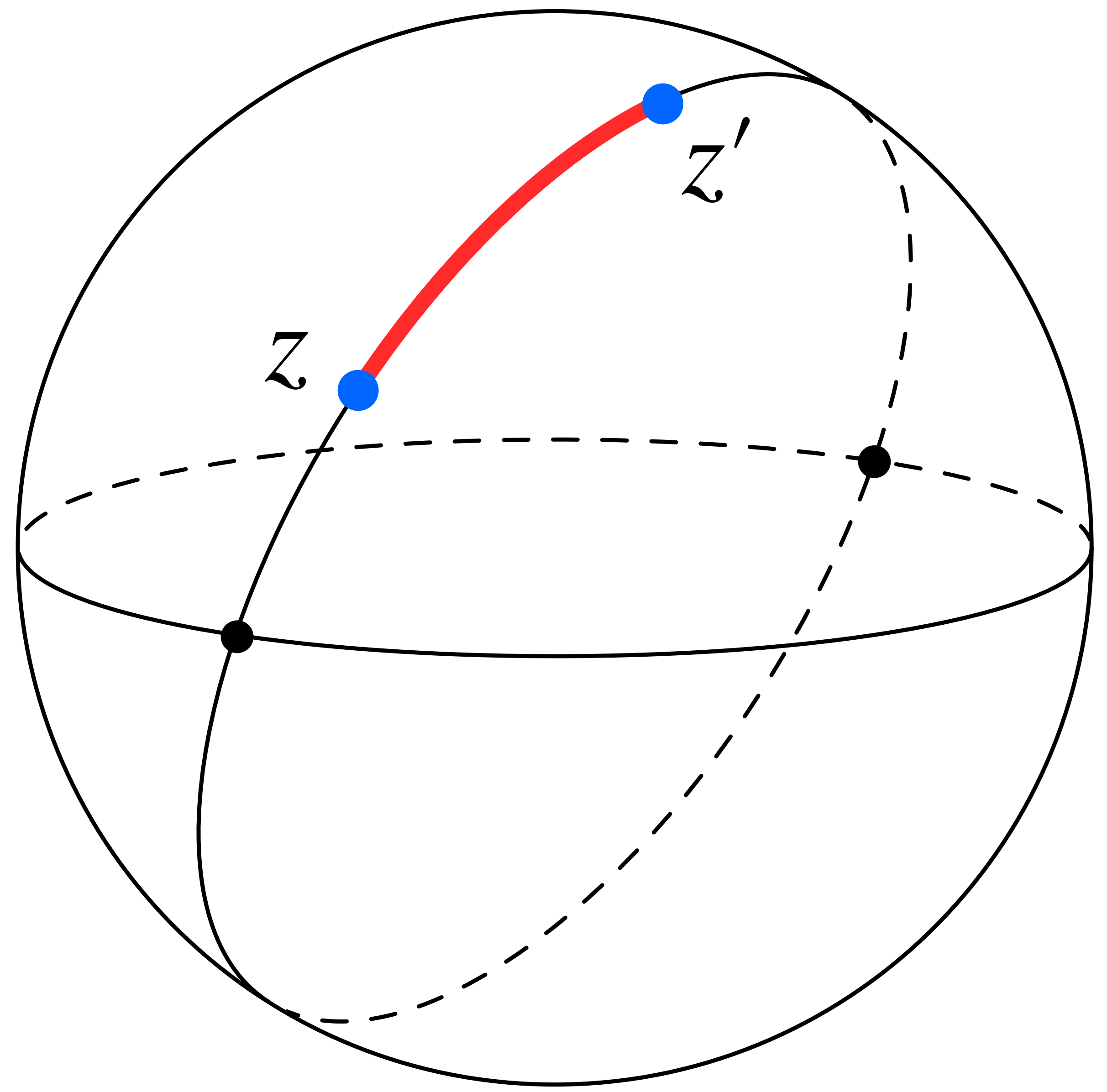
References: [Wiki article on cosine similarity]

Cosine similarity

- Most often, models output $z \in \mathbb{R}^d$
- We first project to the sphere by mapping

$$z \to \frac{z}{||z||} \qquad \Longrightarrow \qquad \zeta(z, z') = \frac{z^T z'}{||z||\,||z'||}$$

- $\exists\, \phi : [-1,1] \to [0,\infty)$ strictly increasing such that



Riemannian (or geodesic) distance

$$\phi(\zeta(z, z')) = \inf \left\{ \int_0^1 |\dot{\gamma}(s)|\, ds \mid \gamma : [0,1] \to \mathbb{S}^{d-1}, \gamma(0) = z, \gamma(1) = z' \right\}$$

References: [Wiki article on cosine similarity]

Defining contrastive loss in generality

- Let $p^+(\cdot \mid \cdot)$ and $p^-(\cdot \mid \cdot)$ be two conditional distributions

References: [Tian | Neurips ’22], [Schneider, Lee, Mathis | Nature ‘23]

Defining contrastive loss in generality

Intuitively, for a given 'anchor' x:

Sampling from $p^+(\cdot \mid x)$ allows us to generate samples similar to x ('**positive examples**')

Sampling from $p^-(\cdot \mid x)$ allows us to generate samples different from x ('**negative examples**')

- Let $p^+(\cdot \mid \cdot)$ and $p^-(\cdot \mid \cdot)$ be two conditional distributions

References: [Tian | Neurips '22], [Schneider, Lee, Mathis | Nature '23]

Defining contrastive loss in generality

Intuitively, for a given 'anchor' x:

Sampling from $p^+(\cdot \mid x)$ allows us to generate samples similar to x ('**positive examples**')

Sampling from $p^-(\cdot \mid x)$ allows us to generate samples different from x ('**negative examples**')

- Let $p^+(\cdot \mid \cdot)$ and $p^-(\cdot \mid \cdot)$ be two conditional distributions
- Let $\phi, \psi \in C^1(\mathbb{R}; \mathbb{R})$ be monotonically increasing
- Let $d_{x,y} = \zeta(f(x), f(y))$

References: [Tian | Neurips '22], [Schneider, Lee, Mathis | Nature '23]

Intuitively, for a given 'anchor' x:

Sampling from $p^+(\cdot \mid x)$ allows us to generate samples similar to x ('**positive examples**')

Sampling from $p^-(\cdot \mid x)$ allows us to generate samples different from x ('**negative examples**')

Defining contrastive loss in generality

- Let $p^+(\cdot \mid \cdot)$ and $p^-(\cdot \mid \cdot)$ be two conditional distributions
- Let $\phi, \psi \in C^1(\mathbb{R}; \mathbb{R})$ be monotonically increasing
- Let $d_{x,y} = \zeta(f(x), f(y))$

$$\mathscr{L}[f] = \mathbb{E}_{\substack{x \sim p(x),\ y^+ \sim p^+(y \mid x), \\ y_1^-, \ldots, y_n^- \sim p^-(y \mid x)}} \left[\phi\left(\sum_{i=1}^{n} \psi\left(d_{x,y^+} - d_{x,y_i^-} \right) \right) \right]$$

References: [Tian | Neurips '22], [Schneider, Lee, Mathis | Nature '23]

Specifying a contrastive loss in practice

- We need to make two choices:

Specifying a contrastive loss in practice

- We need to make two choices:

  1. An explicit choice for ϕ and ψ (and the latent space (Z, ζ))

  2. A way to generate positive and negative examples: $p^+(\cdot \mid x)$ and $p^-(\cdot \mid x)$

Specifying a contrastive loss in practice

- We need to make two choices:

  1. An explicit choice for ϕ and ψ (and the latent space (Z, ζ))

  2. A way to generate positive and negative examples: $p^+(\cdot \mid x)$ and $p^-(\cdot \mid x)$

- Let's look at some common choices for ϕ, ψ and $p^+(\cdot \mid x), p^-(\cdot \mid x)$!

Triplet loss: Definition

- Let $\epsilon > 0$ (*margin*), $n = 1$ and $\phi(x) = x$, $\psi(x) = \max(0, x + \epsilon)$

References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Definition

- Let $\epsilon > 0$ (*margin*), $n = 1$ and $\phi(x) = x$, $\psi(x) = \max(0, x + \epsilon)$

$$\mathscr{L}_{\text{triplet}}[f] = \mathbb{E}_{x,y^+,y^-}\left[\max(0, \epsilon + d_{x,y^+} - d_{x,y^-})\right]$$

References: [Weinberger, Saul ‘09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Definition

- Let $\epsilon > 0$ (*margin*), $n = 1$ and $\phi(x) = x$, $\psi(x) = \max(0, x + \epsilon)$

$$\mathscr{L}_{\text{triplet}}[f] = \mathbb{E}_{x,y^+,y^-}\left[\max(0, \epsilon + d_{x,y^+} - d_{x,y^-})\right]$$

- When using $n > 1$ negative examples, this generalises to the **N-pair loss**

$$\mathscr{L}_{\text{n-pair}}[f] = \mathbb{E}_{x,y^+,y_1^-,\ldots,y_n^-}\left[\sum_{i=1}^{n} \max(0, \epsilon + d_{x,y^+} - d_{x,y_i^-})\right]$$

References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Definition

- Let $\epsilon > 0$ (*margin*), $n = 1$ and $\phi(x) = x$, $\psi(x) = \max(0, x + \epsilon)$

$$\mathcal{L}_{\text{triplet}}[f] = \mathbb{E}_{x,y^+,y^-}\left[\max(0, \epsilon + d_{x,y^+} - d_{x,y^-})\right]$$

- When using $n > 1$ negative examples, this generalises to the **N-pair loss**

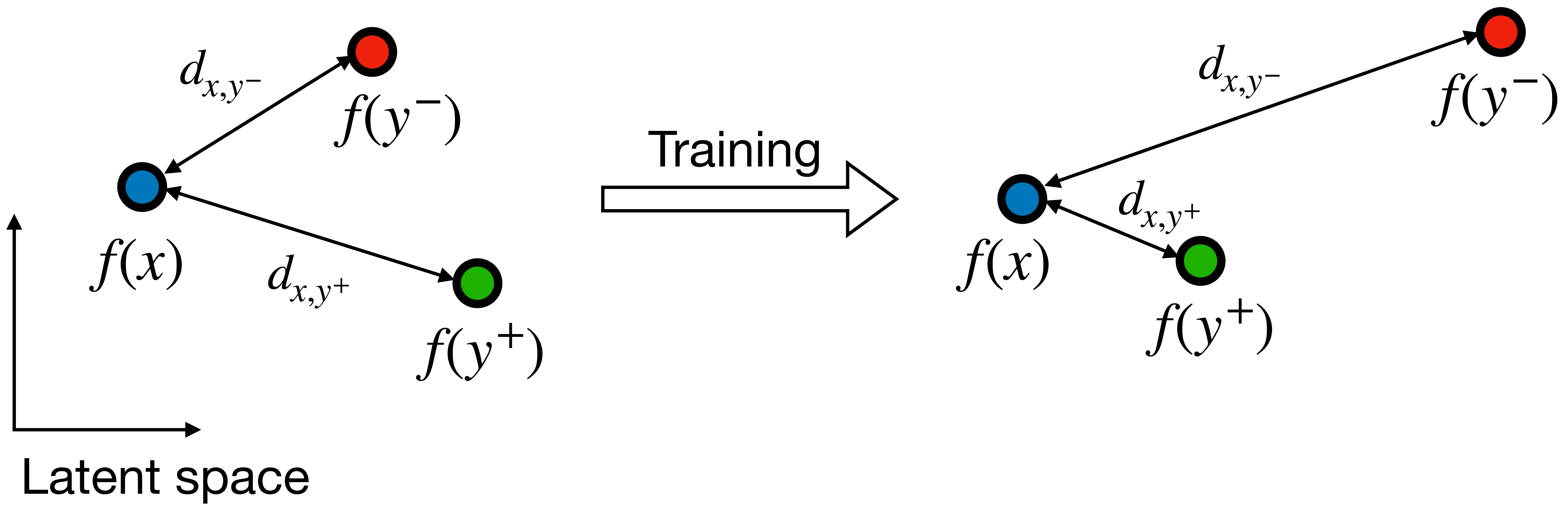
$$\mathcal{L}_{\text{n-pair}}[f] = \mathbb{E}_{x,y^+,y_1^-,\ldots,y_n^-}\left[\sum_{i=1}^{n}\max(0, \epsilon + d_{x,y^+} - d_{x,y_i^-})\right]$$

- Let us develop some intuition for this loss…

References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Intuition

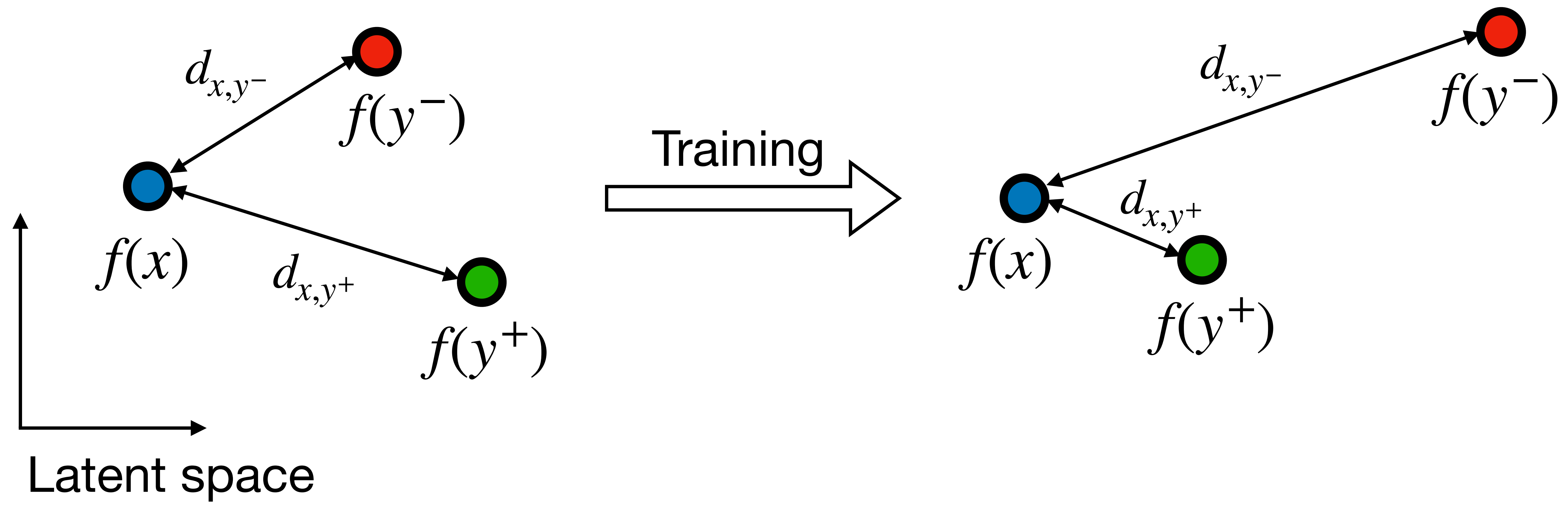
- Why not use the simpler loss function $\mathscr{L}[f] = \mathbb{E}_{x,y^+,y^-}[d_{x,y^+} - d_{x,y^-}]$? Intuitively:



References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Intuition

- Why not use the simpler loss function $\mathscr{L}[f] = \mathbb{E}_{x,y^+,y^-}[d_{x,y^+} - d_{x,y^-}]$? Intuitively:



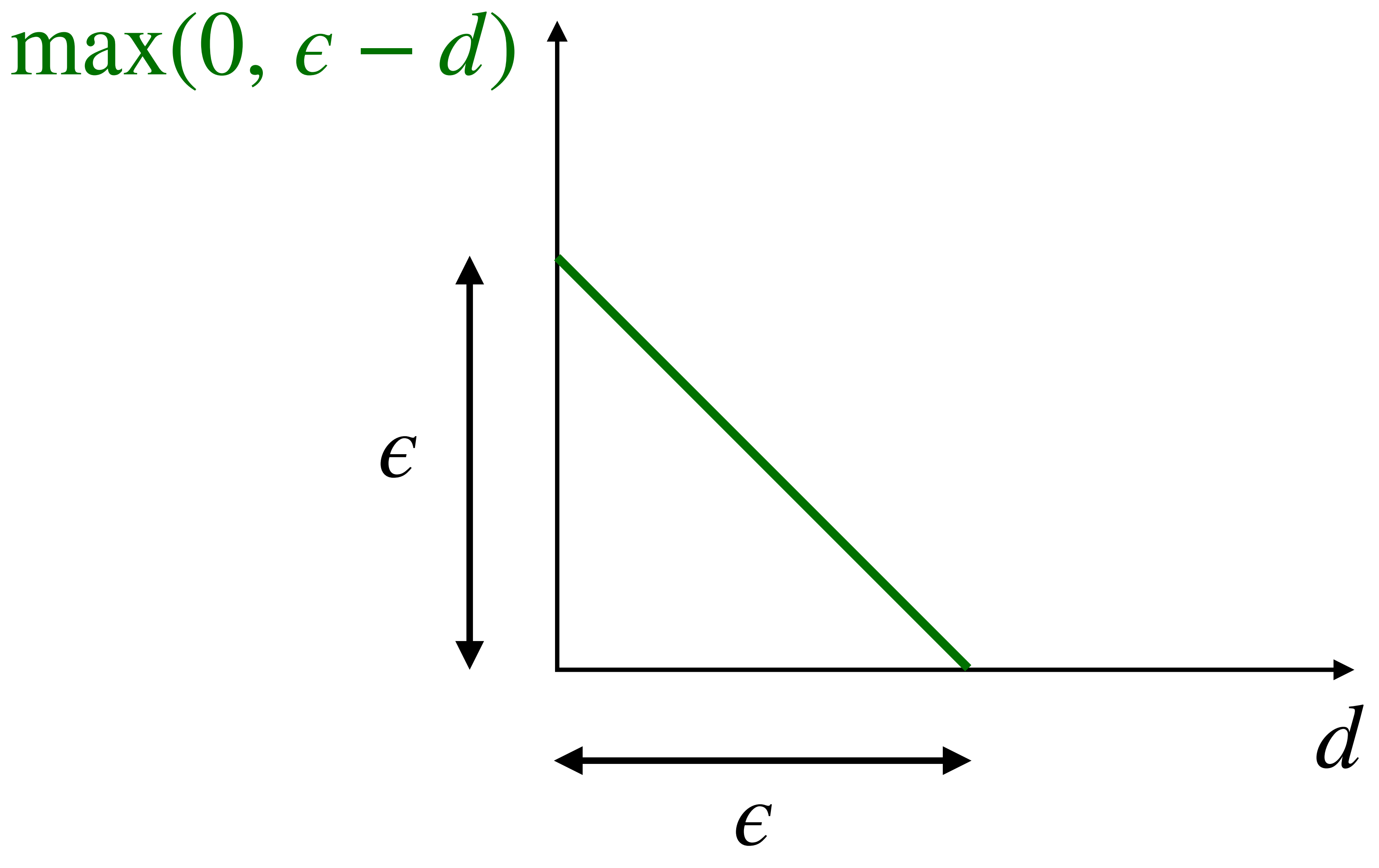
- Caveat: this loss is *not lower-bounded* (unless f is bounded)

➡Divergence during train time

References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Intuition

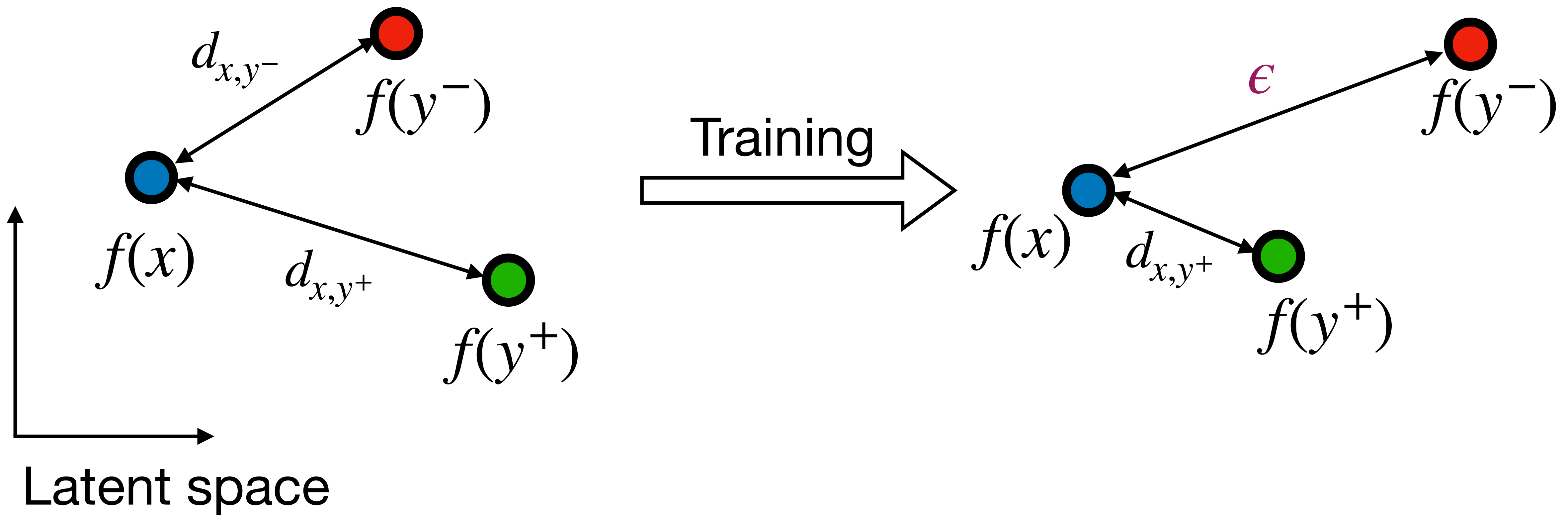
- Instead, use a **hinge loss** $\mathscr{L} = d_{x,y^+} + \max(0, \epsilon - d_{x,y^-})$, where $\epsilon > 0$



References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Intuition

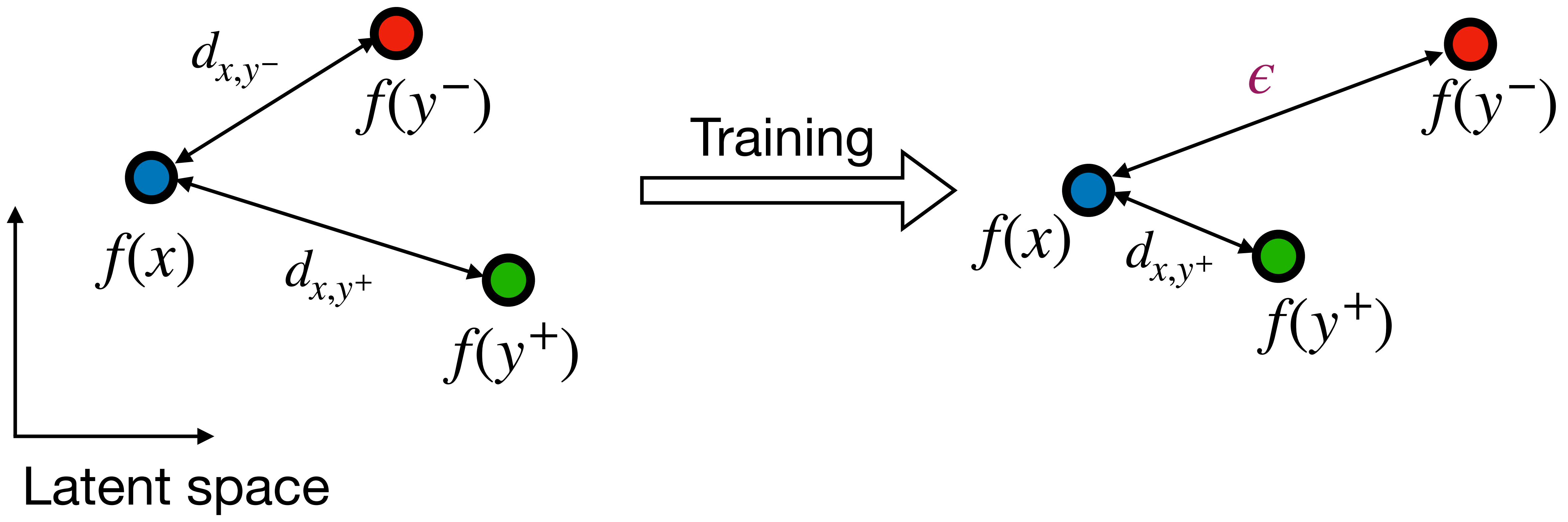
- Now $\mathscr{L} \geq 0$ and once $d_{x,y^-} \geq \epsilon$, the pair (x, y^-) does not contribute to the loss



References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

Triplet loss: Intuition

- Now $\mathscr{L} \geq 0$ and once $d_{x,y^-} \geq \epsilon$, the pair (x, y^-) does not contribute to the loss



- Commonly, model outputs are normalised and cosine similarity is used

References: [Weinberger, Saul '09] [Schroff, Kalenichenko, Philbin | CVPR '15]

InfoNCE loss: Definition

- For $\epsilon \geq 0$ and $\tau > 0$ (*temperature*): $\phi(x) = \tau \log(\epsilon + x), \psi(x) = e^{x/\tau}$:

$$\mathscr{L}_{\text{NCE}}[f] = \mathbb{E}_{x,y^+,y_1^-,\ldots,y_n^-}\left[-\log\left(\frac{\exp(-d_{x,y^+}/\tau)}{\epsilon \exp(-d_{x,y^+}/\tau) + \sum_{i=1}^{n} \exp(-d_{x,y_i^-}/\tau)}\right)\right]$$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR ‘20] [Blog post]

Jump forward

InfoNCE loss: Interpretation

- Problem: Given a reference point $x \sim p(x)$ and $n + 1$ samples $\{x_1, x_2, \ldots, x_{n+1}\}$ where $x_{\mathcal{T}} \sim p^+(\cdot \mid x)$ is one positive sample and $x_i \sim p^-(\cdot), i \neq \mathcal{T}$ are 'noise' samples. **Identify the positive sample.**

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR '20] [Blog post]

InfoNCE loss: Interpretation

- Problem: Given a reference point $x \sim p(x)$ and $n+1$ samples $\{x_1, x_2, \ldots, x_{n+1}\}$ where $x_{\mathcal{T}} \sim p^+(\cdot \mid x)$ is one positive sample and $x_i \sim p^-(\cdot), i \neq \mathcal{T}$ are 'noise' samples. **Identify the positive sample.**
- The probability that the i-th sample is the positive one is

$$\mathbb{P}(i = + \,|\, x) = \frac{p^+(x_i \,|\, x)\Pi_{j \neq i} p^-(x_j)}{\sum_j p^+(x_j \,|\, x)\Pi_{k \neq j} p^-(x_k)} = \frac{\frac{p^+(x_i \,|\, x)}{p^-(x_i)}}{\sum_j \frac{p^+(x_j \,|\, x)}{p^-(x_j)}}$$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR '20] [Blog post]

InfoNCE loss: Interpretation

- Let us introduce the abbreviation $g(x_i;\ x) = \dfrac{p^+(x_i\,|\,x)}{p^-(x_i)}$
- The cross entropy of identifying the positive sample correctly is then

$$\mathbb{E}_x\left[-\log \mathbb{P}(\mathcal{T} = +\,|\,x)\right] = \mathbb{E}_x\left[-\log \frac{g(x_{\mathcal{T}};\ x)}{\sum_j g(x_j;\ x)}\right]$$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR ‘20] [Blog post]

InfoNCE loss: Interpretation

- If we identify $\exp(-d_{x,y}/\tau) = g(y;\ x)$ we see **cross entropy = InfoNCE loss**

with $\epsilon = 1$

and $p^{-}(\,\cdot\mid x) = p^{-}(\,\cdot\,)$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR '20] [Blog post]

Jump back

InfoNCE loss: Interpretation

- If we identify $\exp(-d_{x,y}/\tau) = g(y;\ x)$ we see **cross entropy = InfoNCE loss**

  with $\epsilon = 1$
  and $p^-(\,\cdot \mid x) = p^-(\,\cdot\,)$

- Minimising InfoNCE loss $\iff$ maximising the probability of correctly identifying a positive sample among a set of n negative and one positive samples.

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR '20] [Blog post]

Jump back

InfoNCE loss: Interpretation

- If we identify $\exp(-d_{x,y}/\tau) = g(y;\ x)$ we see **cross entropy = InfoNCE loss**

  with $\epsilon = 1$
  and $p^{-}(\cdot \mid x) = p^{-}(\cdot)$

- Minimising InfoNCE loss $\iff$ maximising the probability of correctly identifying a positive sample among a set of n negative and one positive samples.

- We can think of our model as learning the density ratio $\exp(-d_{x,y}/\tau) = \dfrac{p^{+}(y|x)}{p^{-}(y)}$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR '20] [Blog post]

Jump back

InfoNCE loss: Interpretation #2

- Mutual information $I(X \mid Y) = \sum_{x,y} p(x,y) \, \log \frac{p(x \mid y)}{p(x)}$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR ‘20] [Blog post]

InfoNCE loss: Interpretation #2

- Mutual information $I(X \mid Y) = \sum_{x,y} p(x,y) \ \log \frac{p(x \mid y)}{p(x)}$
- If we assume further $p^-(y) = p(y)$ then $\frac{p^+(y|x)}{p^-(y)} = \frac{p^+(y|x)}{p(y)}$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR ‘20] [Blog post]

InfoNCE loss: Interpretation #2

- Mutual information $I(X \mid Y) = \sum_{x,y} p(x,y) \, \log \dfrac{p(x \mid y)}{p(x)}$
- If we assume further $p^{-}(y) = p(y)$ then $\dfrac{p^{+}(y|x)}{p^{-}(y)} = \dfrac{p^{+}(y|x)}{p(y)}$
- Minimising InfoNCE loss $\iff$ maximising mutual information $I(f(x^{+}) \mid f(x))$

References: [van den Oord, Li, Vinyals] [Chen et al. | PMLR '20] [Blog post]

More flavours of loss functions…

Contrastive Loss	$\phi(x)$	$\psi(x)$
InfoNCE (Oord et al., 2018)	$\tau \log(\epsilon + x)$	$e^{x/\tau}$
MINE (Belghazi et al., 2018)	$\log(x)$	e^{x}
Triplet (Schroff et al., 2015)	x	$[x + \epsilon]_+$
Soft Triplet (Tian et al., 2020c)	$\tau \log(1 + x)$	$e^{x/\tau+\epsilon}$
N+1 Tuplet (Sohn, 2016)	$\log(1 + x)$	e^{x}
Lifted Structured (Oh Song et al., 2016)	$[\log(x)]_+^2$	$e^{x+\epsilon}$
Modified Triplet Eqn. 10 (Coria et al., 2020)	x	$\mathrm{sigmoid}(cx)$
Triplet Contrastive Eqn. 2 (Ji et al., 2021)	linear	linear

Overview of loss functions (from [Tian | Neurips '22])

The distribution p^+: Choosing positive examples

Image-based data

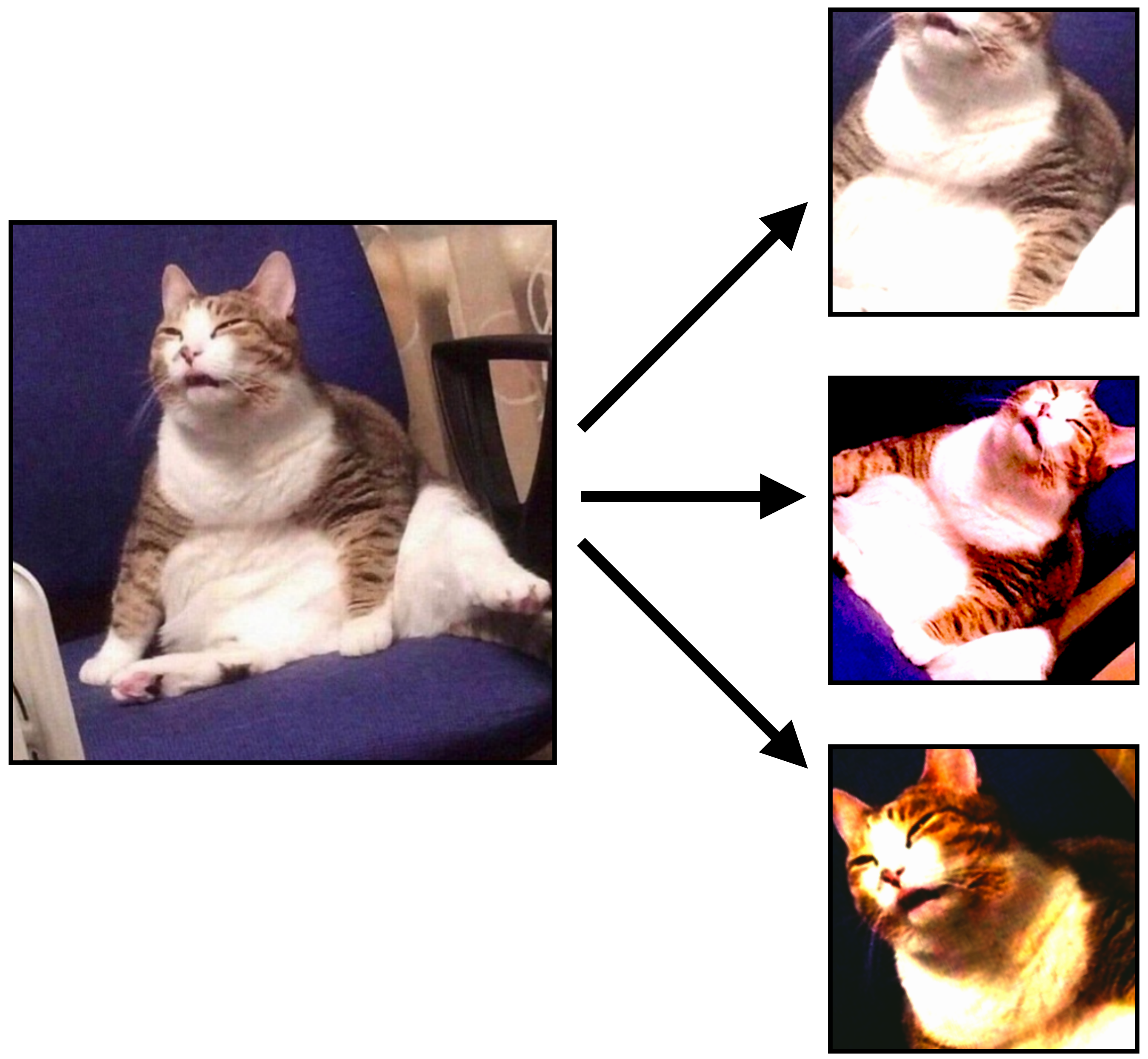
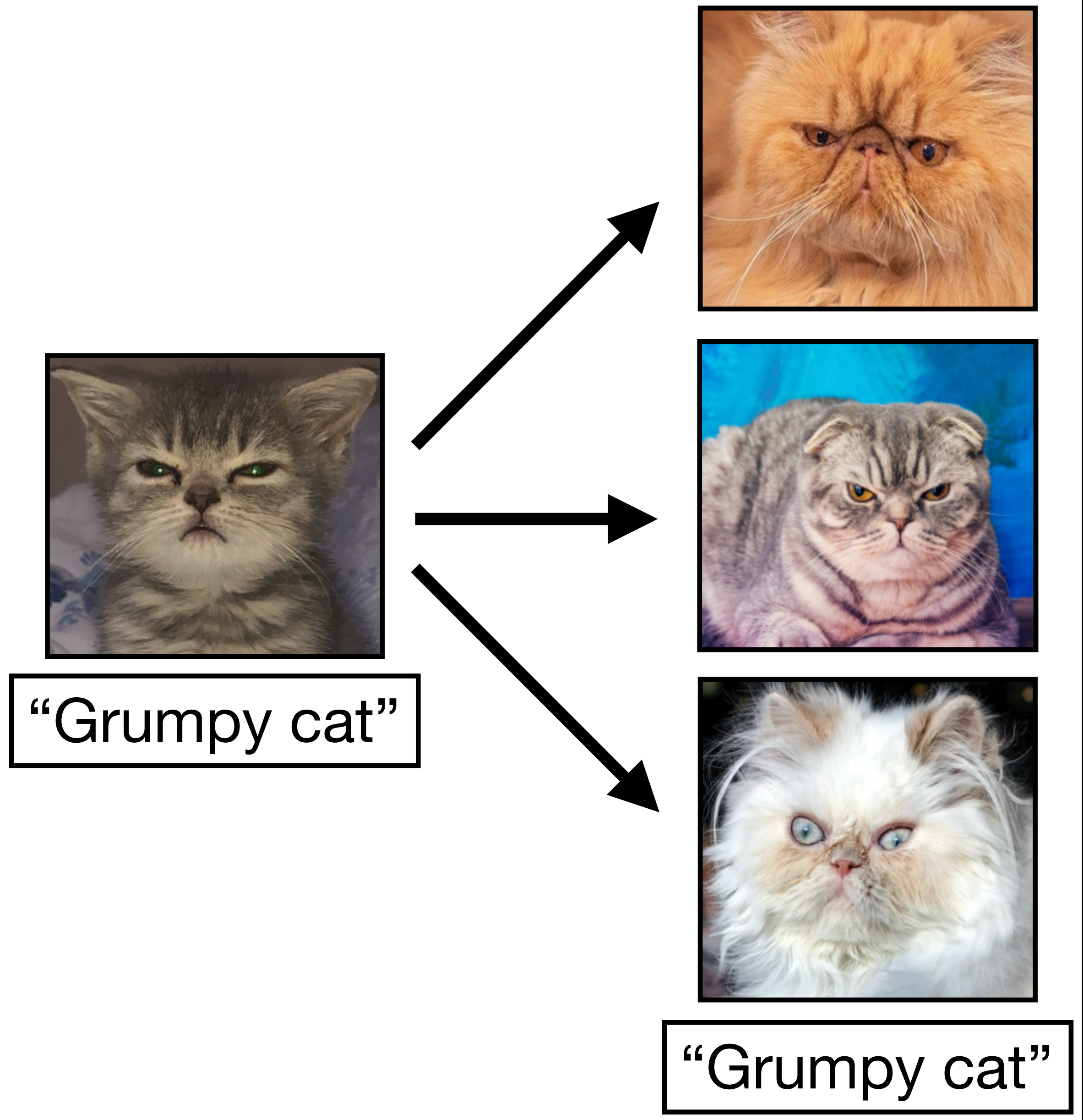


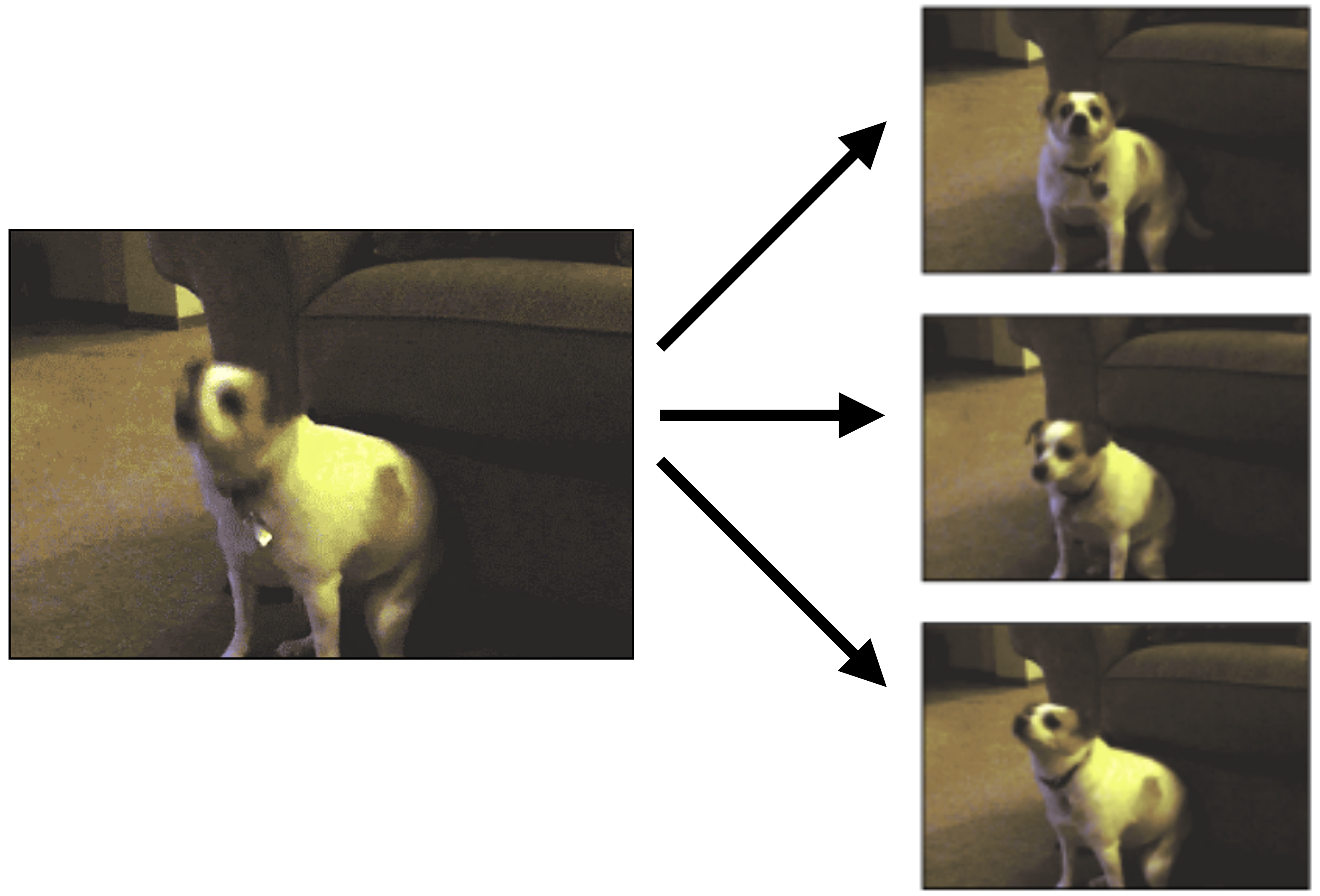
Image augmentations

Labeled data



Sampling within a class

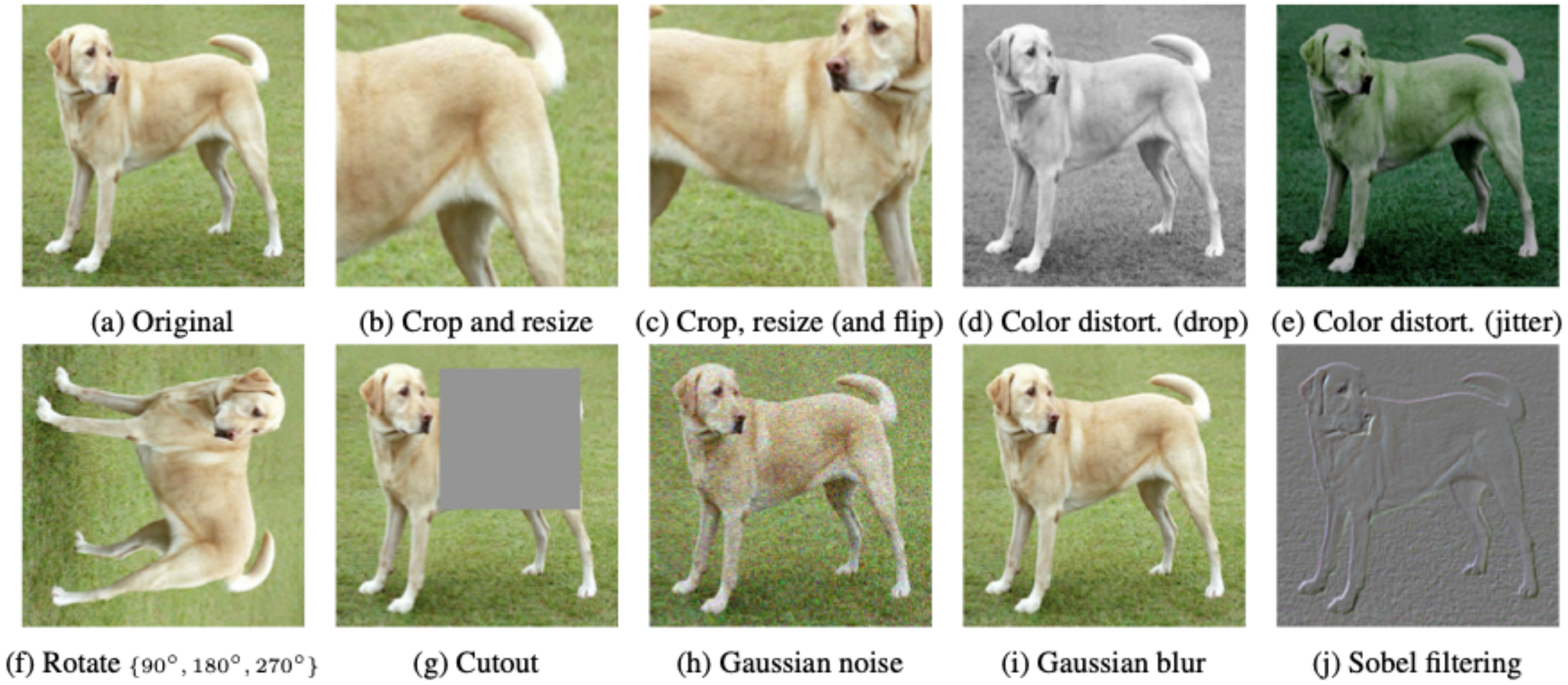
Time series data



Nearby (in time) samples

The distribution p^+: Choosing positive examples

What are image augmentations?



From [Chen et al. | PMLR ‘20]

References: [Cubuk et al. | CVPR ‘19] [Cubuk et al. | CVPR ‘20]

The distribution p^{-}: Choosing negative examples

- Most common: random data sample $p^{-}(y \mid x) = p(y)$

The distribution p^-: Choosing negative examples

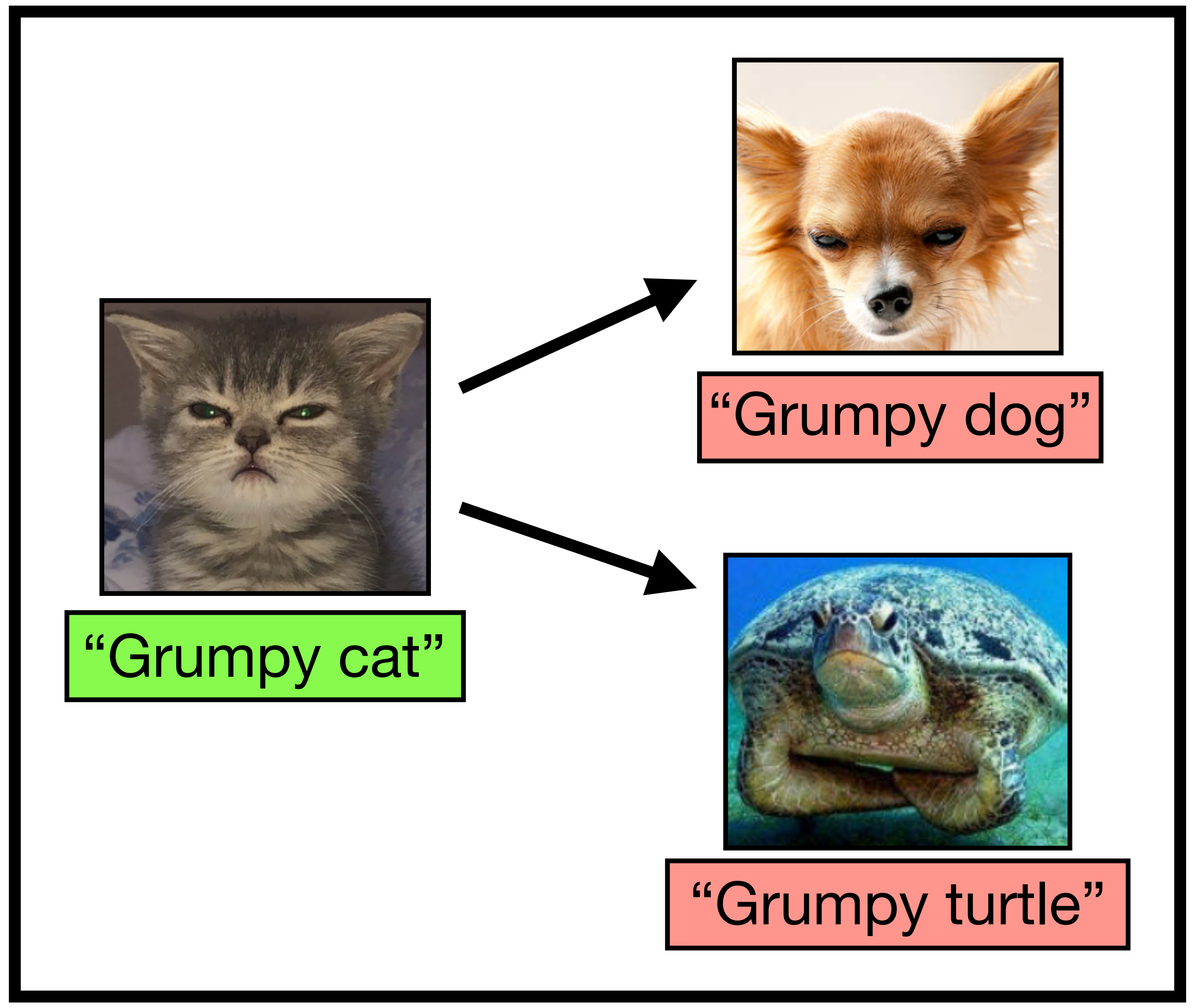
- Most common: random data sample $p^-(y \mid x) = p(y)$

Works best if
$n_{\text{classes}} > n_{\text{samples-per-class}}$

The distribution p^-: Choosing negative examples

- Most common: random data sample $p^-(y \mid x) = p(y)$ ← Works best if $n_{\text{classes}} > n_{\text{samples-per-class}}$

- For labeled data: choose with uniform probability from a distinct class



Application: Supervised contrastive learning for image labelling

Supervised Contrastive Learning

Prannay Khosla * Google Research
Piotr Teterwak *† Boston University
Chen Wang † Snap Inc.
Aaron Sarna ‡ Google Research
Yonglong Tian † MIT
Phillip Isola † MIT
Aaron Maschinot Google Research
Ce Liu Google Research
Dilip Krishnan Google Research

Abstract

Contrastive learning applied to self-supervised representation learning has seen a resurgence in recent years, leading to state of the art performance in the unsupervised training of deep image models. Modern batch contrastive approaches subsume or significantly outperform traditional contrastive losses such as triplet, max-margin and the N-pairs loss. In this work, we extend the self-supervised batch contrastive approach to the *fully-supervised* setting, allowing us to effec-

References: [Khosla et al. | Neurips '20]

Application: Supervised contrastive learning for image labelling

- Introduces “SupCon” loss = variant of InfoNCE loss with multiple positives
- How are positive and negative samples generated?

Application: Supervised contrastive learning for image labelling

- Introduces "SupCon" loss = variant of InfoNCE loss with multiple positives
- How are positive and negative samples generated?
  - **Negative** samples: choose randomly from another class
  - **Positive** samples:
    - First generate two image augmentations of each sample
    - All augmentations of images from the same class are positive

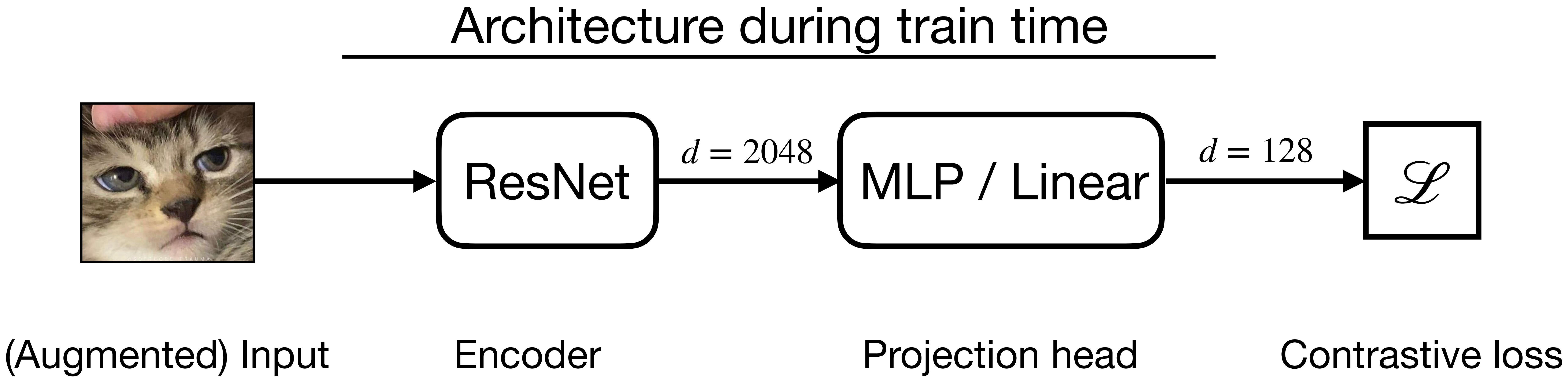
Application: Supervised contrastive learning for image labelling

- Model architecture features a projection head which is discarded for inference

References: [Khosla et al. | Neurips ‘20]

Application: Supervised contrastive learning for image labelling

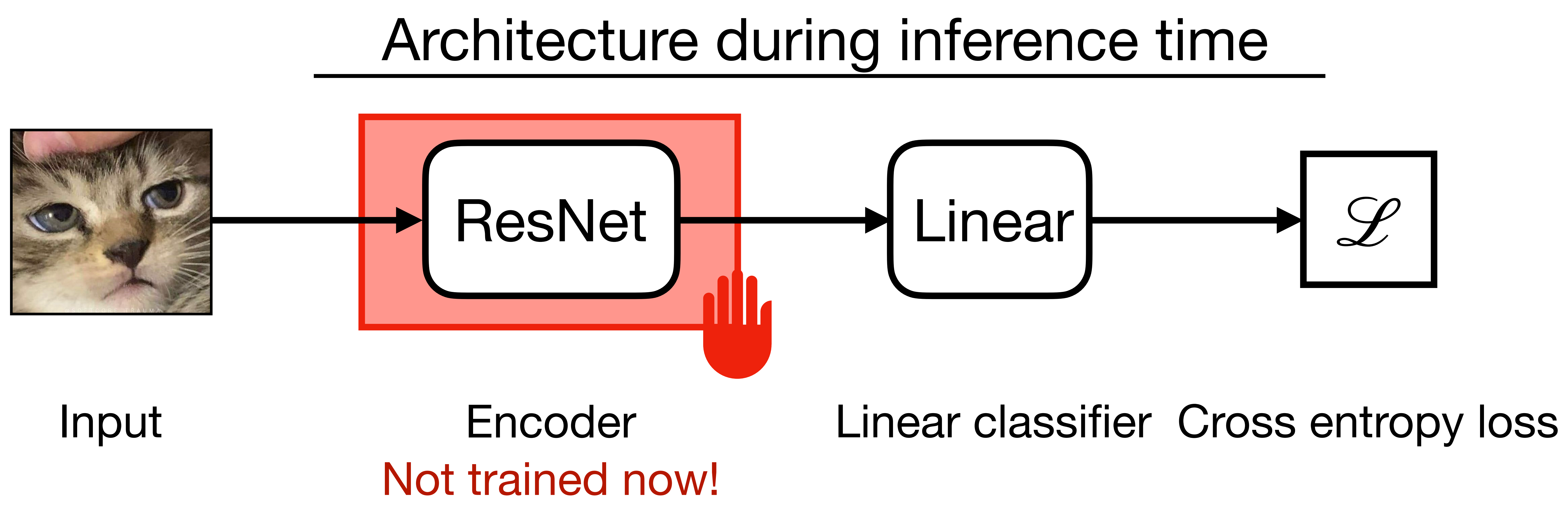
- Model architecture features a projection head which is discarded for inference



References: [Khosla et al. | Neurips '20]

Application: Supervised contrastive learning for image labelling

- Model architecture features a projection head which is discarded for inference



References: [Khosla et al. | Neurips '20]

Application: Supervised contrastive learning for image labelling

- State of the art accuracy on various image datasets

Dataset	SimCLR[3]	Cross-Entropy	Max-Margin [32]	SupCon
CIFAR10	93.6	95.0	92.4	**96.0**
CIFAR100	70.7	75.3	70.5	**76.5**
ImageNet	70.2	78.2	78.0	**78.7**

Top-1 accuracy on ResNet-50.

- Recall: **Top-n accuracy** counts the number of times in which the correct class appears within the first n most probable classes predicted by the classifier

- Performance is significantly better when normalising outputs (cosine similarity)

References: [Khosla et al. | Neurips '20]

Application: Joint behavioural and neural analysis

Article

Learnable latent embeddings for joint behavioural and neural analysis

https://doi.org/10.1038/s41586-023-06031-6

Received: 30 March 2022

Accepted: 28 March 2023

Published online: 3 May 2023

Open access

Check for updates

Steffen Schneider[1,2], Jin Hwa Lee[1,2] & Mackenzie Weygandt Mathis[1]✉

Mapping behavioural actions to neural activity is a fundamental goal of neuroscience. As our ability to record large neural and behavioural data increases, there is growing interest in modelling neural dynamics during adaptive behaviours to probe neural representations[1–3]. In particular, although neural latent embeddings can reveal underlying correlates of behaviour, we lack nonlinear techniques that can explicitly and flexibly leverage joint behaviour and neural data to uncover neural dynamics[3–5]. Here, we fill this gap with a new encoding method, CEBRA, that jointly uses behavioural and neural data in a (supervised) hypothesis- or (self-supervised) discovery-driven manner to produce both consistent and high-performance latent spaces. We show that consistency can be used as a metric for uncovering meaningful differences, and the inferred latents can be used for decoding. We validate its accuracy and demonstrate our tool's utility for both calcium and electrophysiology datasets, across sensory and motor tasks and in simple or complex behaviours across species. It allows leverage of single- and multi-session datasets for hypothesis testing or can be used label free. Lastly, we show that CEBRA can be used for the mapping of space, uncovering complex kinematic features, for the production of consistent latent spaces across two-photon and Neuropixels data, and can provide rapid, high-accuracy decoding of natural videos from visual cortex.

References: [Schneider, Lee, Mathis | Nature '23]

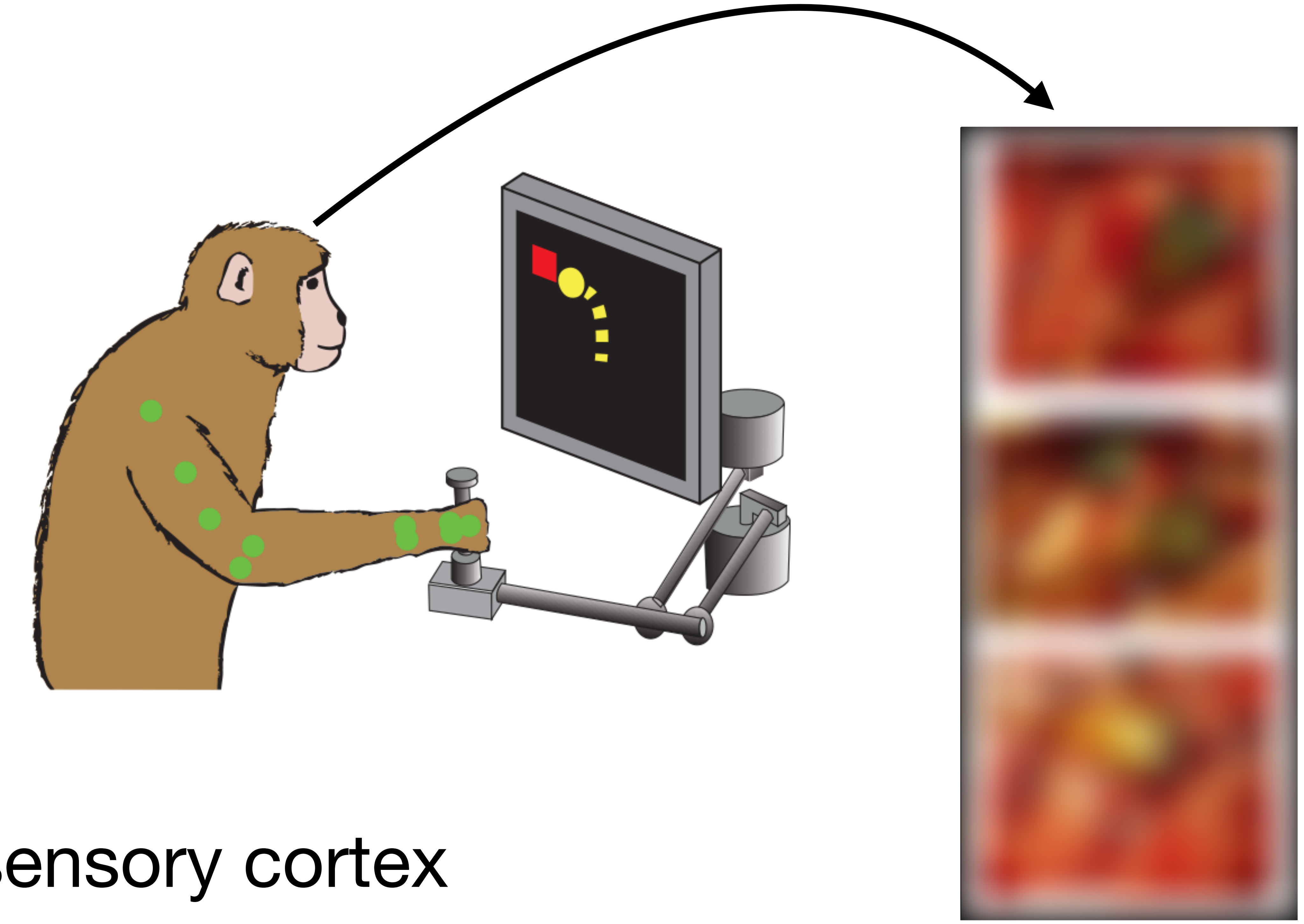
Application: Joint behavioural and neural analysis

- Data are time series $t \mapsto (s_t, c_t)$, where
  - s_t represents a neural state
  - c_t represents a context vector

References: [Schneider, Lee, Mathis | Nature '23]

Application: Joint behavioural and neural analysis

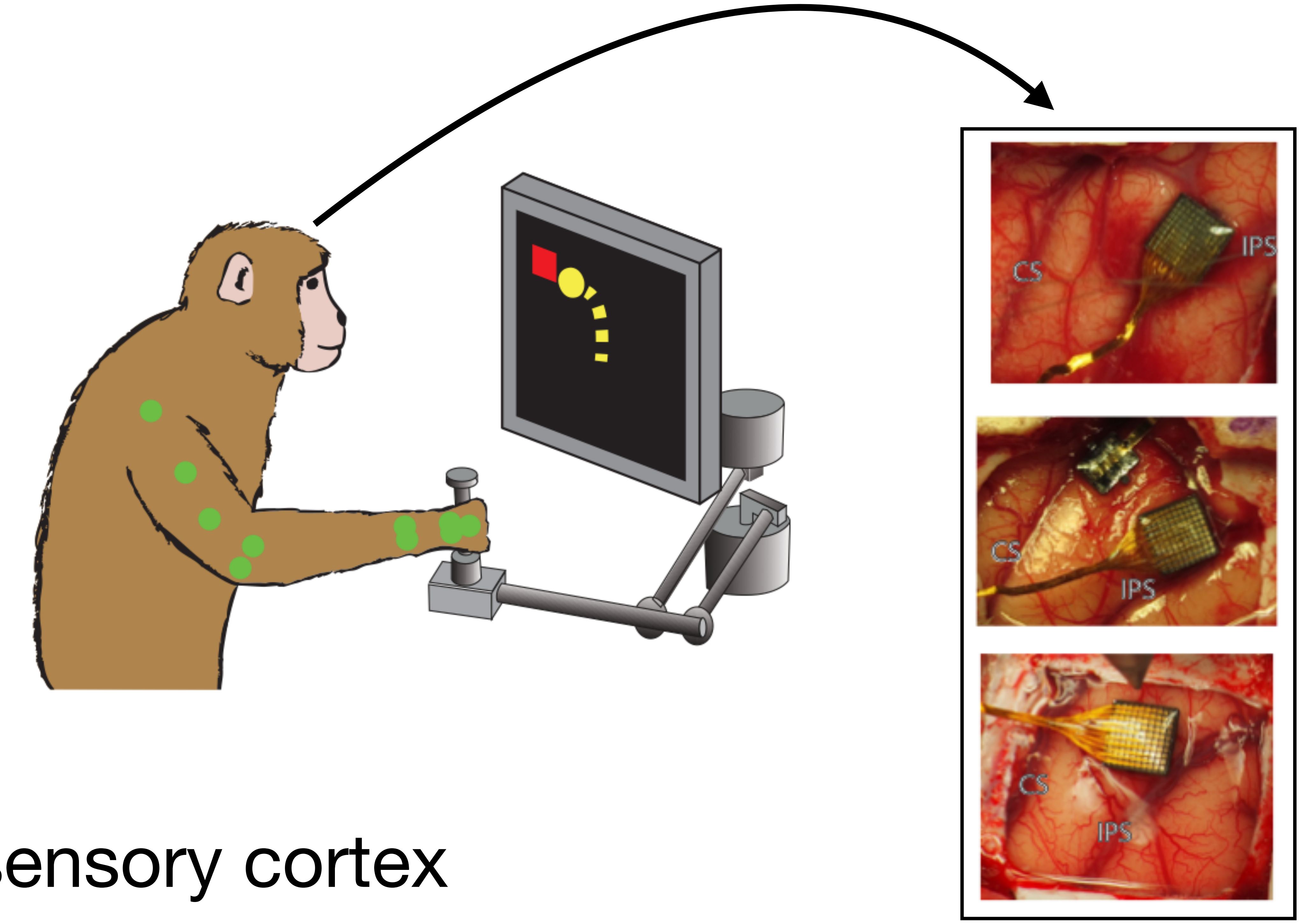
- Data are time series $t \mapsto (s_t, c_t)$, where
  - s_t represents a neural state
  - c_t represents a context vector
- Example: **Monkey reaching task**
  - s_t = electrophysiology recordings of somatosensory cortex
  - c_t = position of the monkey's hand



References: [Schneider, Lee, Mathis | Nature '23] [Chowdhury et al. | eLife '20]

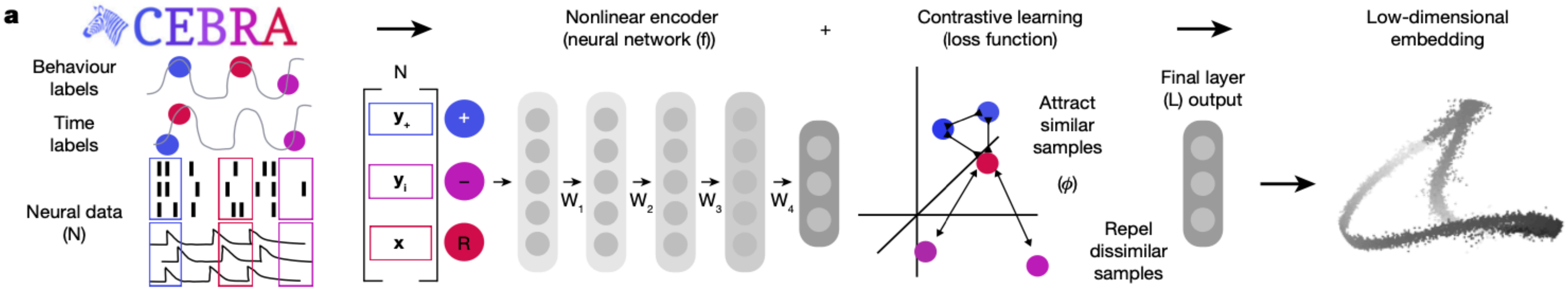
Application: Joint behavioural and neural analysis

- Data are time series $t \mapsto (s_t, c_t)$, where
  - s_t represents a neural state
  - c_t represents a context vector
- Example: **Monkey reaching task**
  - s_t = electrophysiology recordings of somatosensory cortex
  - c_t = position of the monkey's hand



References: [Schneider, Lee, Mathis | Nature '23] [Chowdhury et al. | eLife '20]

Application: Joint behavioural and neural analysis



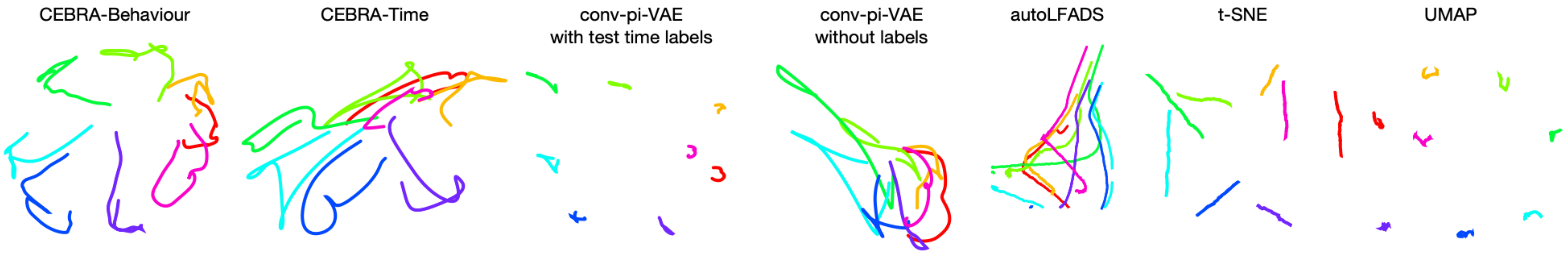
- Uses **InfoNCE** loss and *two* ways of choosing positive examples (negative examples are chosen randomly)
  - Based on closeness in time: for anchor (s_t, c_t) pick $(s_{t+\Delta t}, c_{t+\Delta t})$ for some small Δt
  - Based on similar context variable: for anchor (s_t, c_t) pick $(s_{t'}, c_{t'})$ such that $c_t \approx c_{t'}$

References: [Schneider, Lee, Mathis | Nature ‘23]

Application: Joint behavioural and neural analysis

- When trained with behavioural information (*CEBRA-Behaviour*), computes embeddings which can be used to reconstruct or visualise behaviour

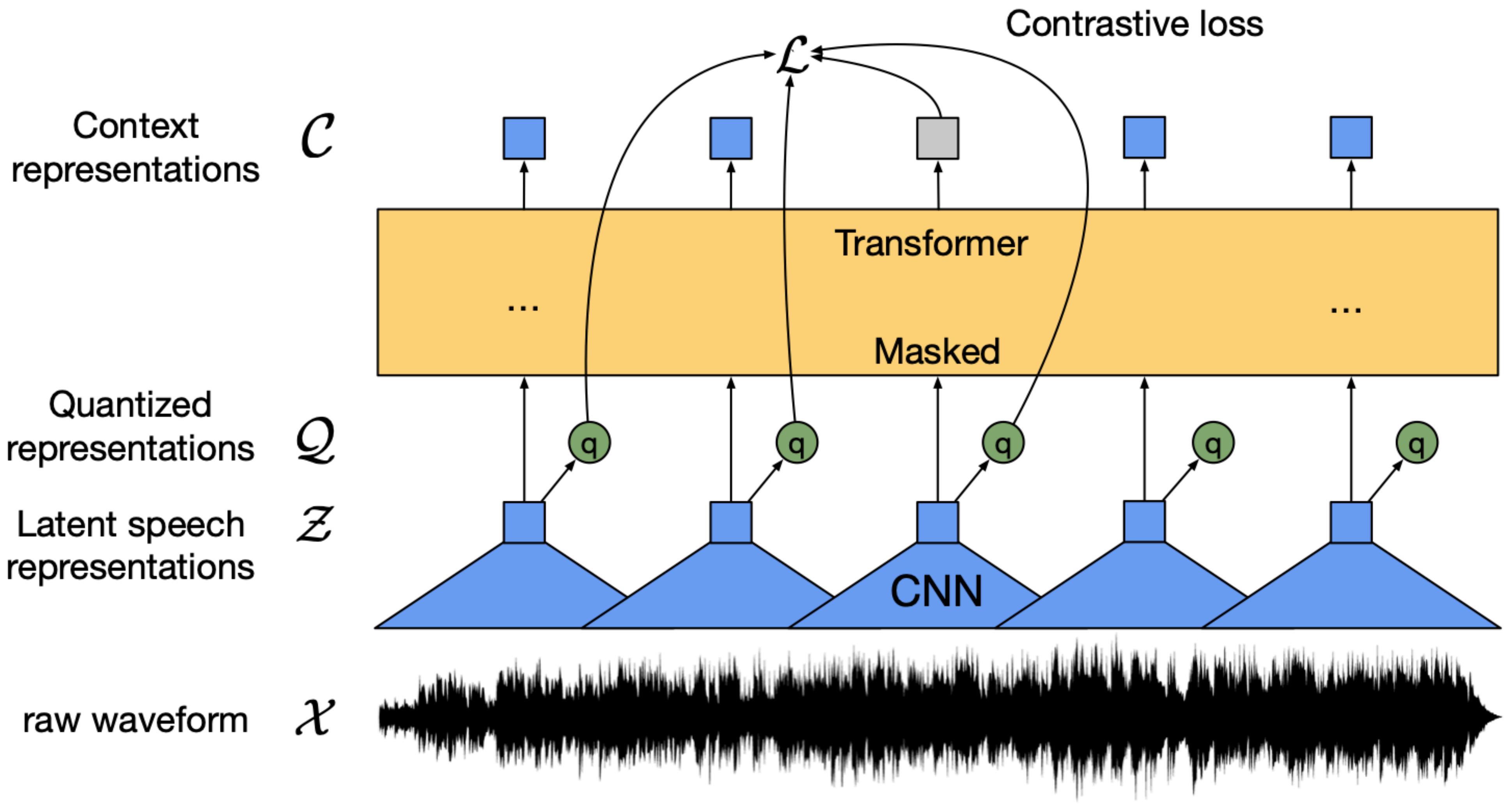
- When trained using only closeness in time (*CEBRA-Time*), still allows to reconstruct some degree of behavioural information!

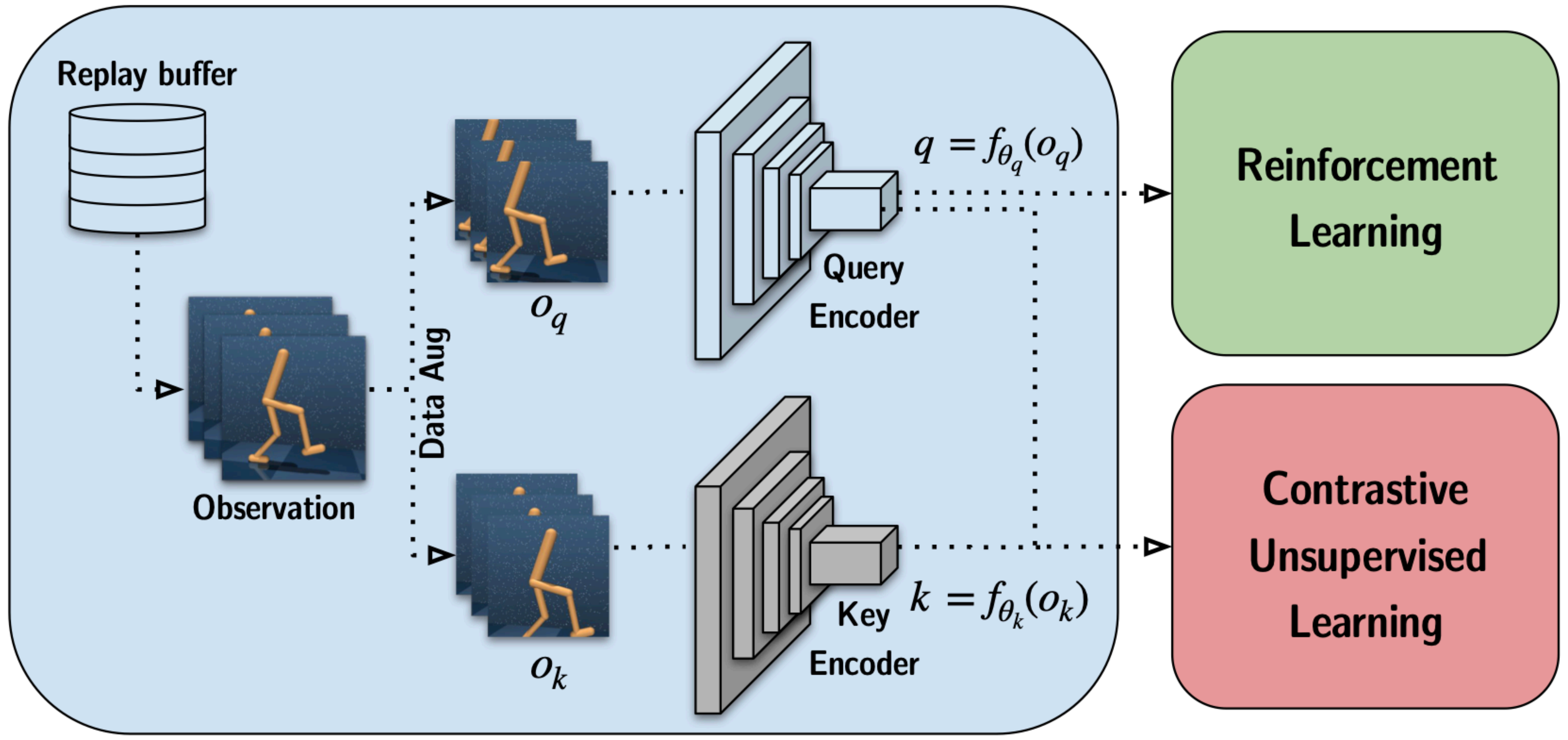


References: [Schneider, Lee, Mathis | Nature '23]

Further applications

- Speech recognition (**wav2vec**) *[Schneider et al. | INTERSPEECH '19] [Baevski et al. | Neurips '20]*
- Improving sample efficiency of **reinforcement learning** *[Srinivas, Laskin, Abbeel | MLR '20]*





The effect of batch size

- Recall: due to memory constraints data is split into batches during train time
- Assume we have a dataset D and partition it into m batches of equal size

$$D = \coprod_{i=1}^{m} B_i, \quad |B_i| = |B_j| \quad \forall i, j < m$$

- During train time, after each iteration of the full dataset, batches are reshuffled

The effect of batch size

- Then we can rewrite our loss as

$$\mathscr{L}[\theta] = \sum_{x \in D} l(\theta; x) = \sum_{i=1}^{m} \sum_{x \in B_i} l(\theta; x)$$

The effect of batch size

- Then we can rewrite our loss as

$$\mathscr{L}[\theta] = \sum_{x \in D} l(\theta; x) = \sum_{i=1}^{m} \sum_{x \in B_i} l(\theta; x)$$

- Gradient updates computed on **entire dataset** (**batch gradient descent**)

$$\theta_{i+1} = \theta_i - \alpha \sum_{i=1}^{m} \sum_{x \in B_i} \nabla_\theta l(\theta; x)$$

The effect of batch size

- Then we can rewrite our loss as

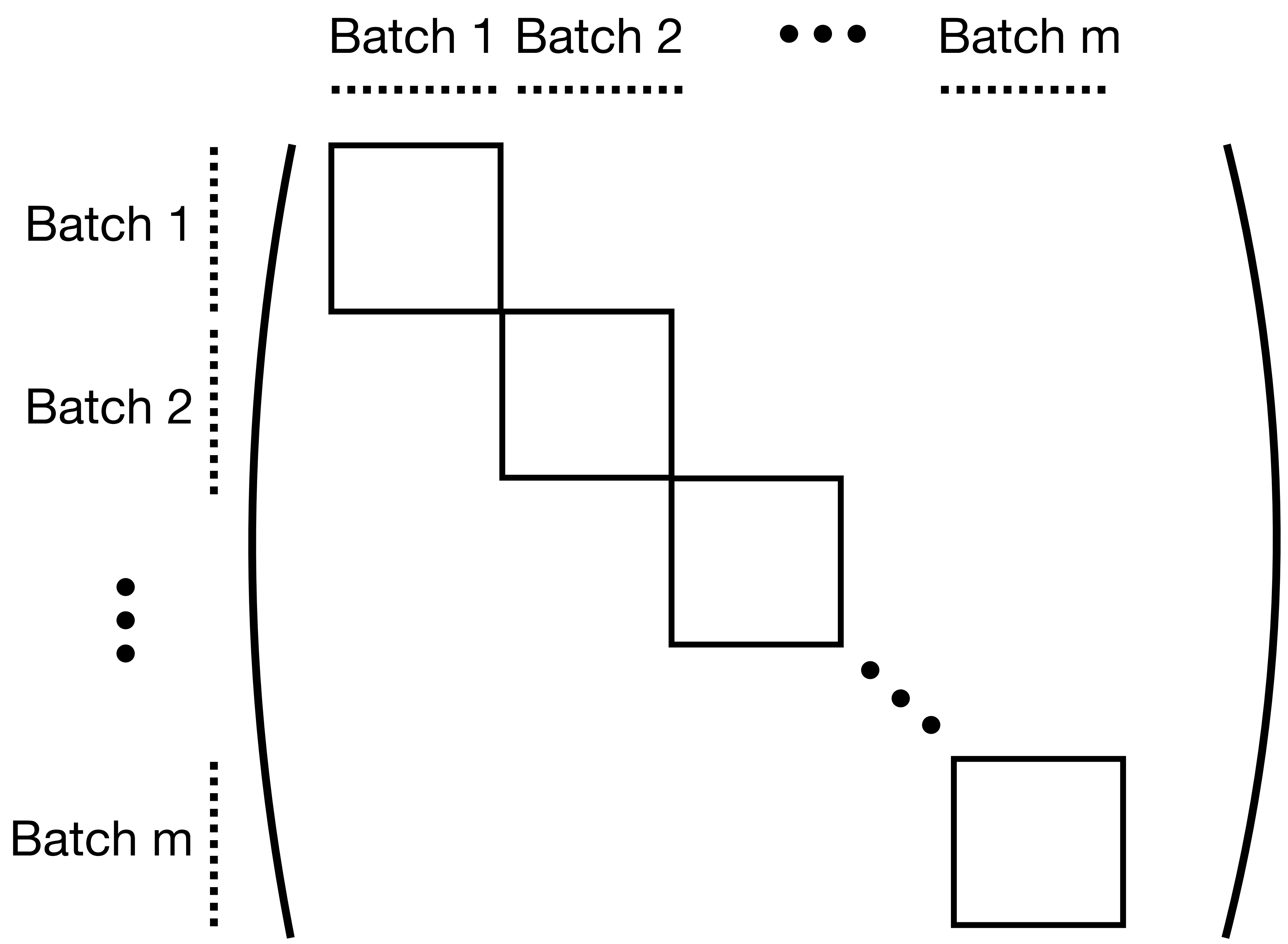
$$\mathscr{L}[\theta] = \sum_{x \in D} l(\theta; x) = \sum_{i=1}^{m} \sum_{x \in B_i} l(\theta; x)$$

- Gradient updates computed on **each batch** (**mini-batch gradient descent**)

$$\theta_{i+1} = \theta_i - \alpha \sum_{x \in B_i} \nabla_\theta l(\theta; x)$$

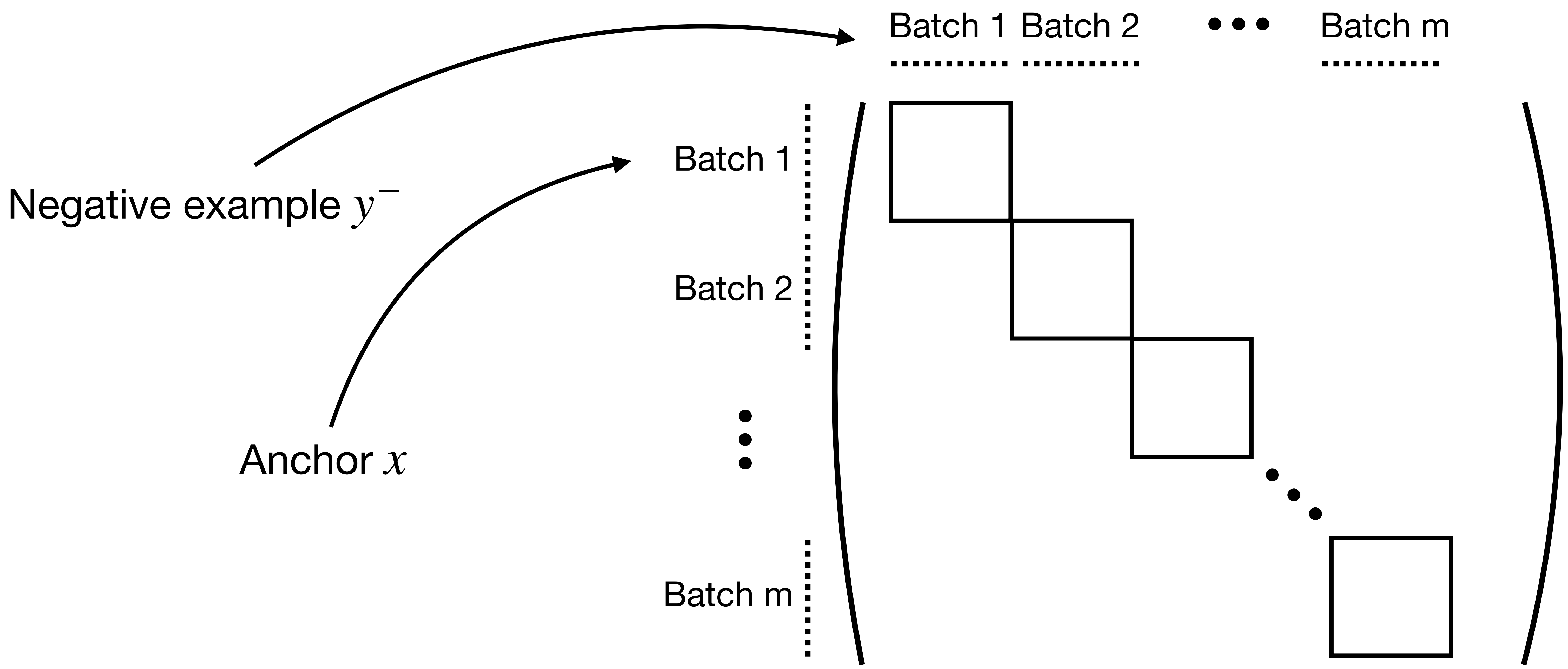
The effect of batch size

- For contrastive loss, positive/negative samples only found within one batch



The effect of batch size

- For contrastive loss, positive/negative samples only found within one batch



The effect of batch size

- A more formal way of expressing the same picture:

[Wiki entry on Jensen's inequality]

The effect of batch size

- A more formal way of expressing the same picture:
- Recall Jensen's inequality $\frac{1}{n}\sum_{i=1}^{n}\log(x_i) \leq \log\left(\frac{1}{n}\sum_{i=1}^{n}x_i\right)$

[Wiki entry on Jensen's inequality]

The effect of batch size

- A more formal way of expressing the same picture:
- Recall Jensen's inequality $\frac{1}{n}\sum_{i=1}^{n}\log(x_i) \leq \log\left(\frac{1}{n}\sum_{i=1}^{n}x_i\right)$
- For the InfoNCE loss (with $\epsilon = 0, \tau = 1$) we have

$$\sum_{batches}\log\sum_{i}\exp(-d_{x,y_i^-}) \leq \log\sum_{batches}\sum_{i}\exp(-d_{x,y_i^-})$$

[Wiki entry on Jensen's inequality]

The effect of batch size

- A more formal way of expressing the same picture:
- Recall Jensen's inequality $\frac{1}{n}\sum_{i=1}^{n}\log(x_i) \leq \log\left(\frac{1}{n}\sum_{i=1}^{n}x_i\right)$
- For the InfoNCE loss (with $\epsilon = 0, \tau = 1$) we have

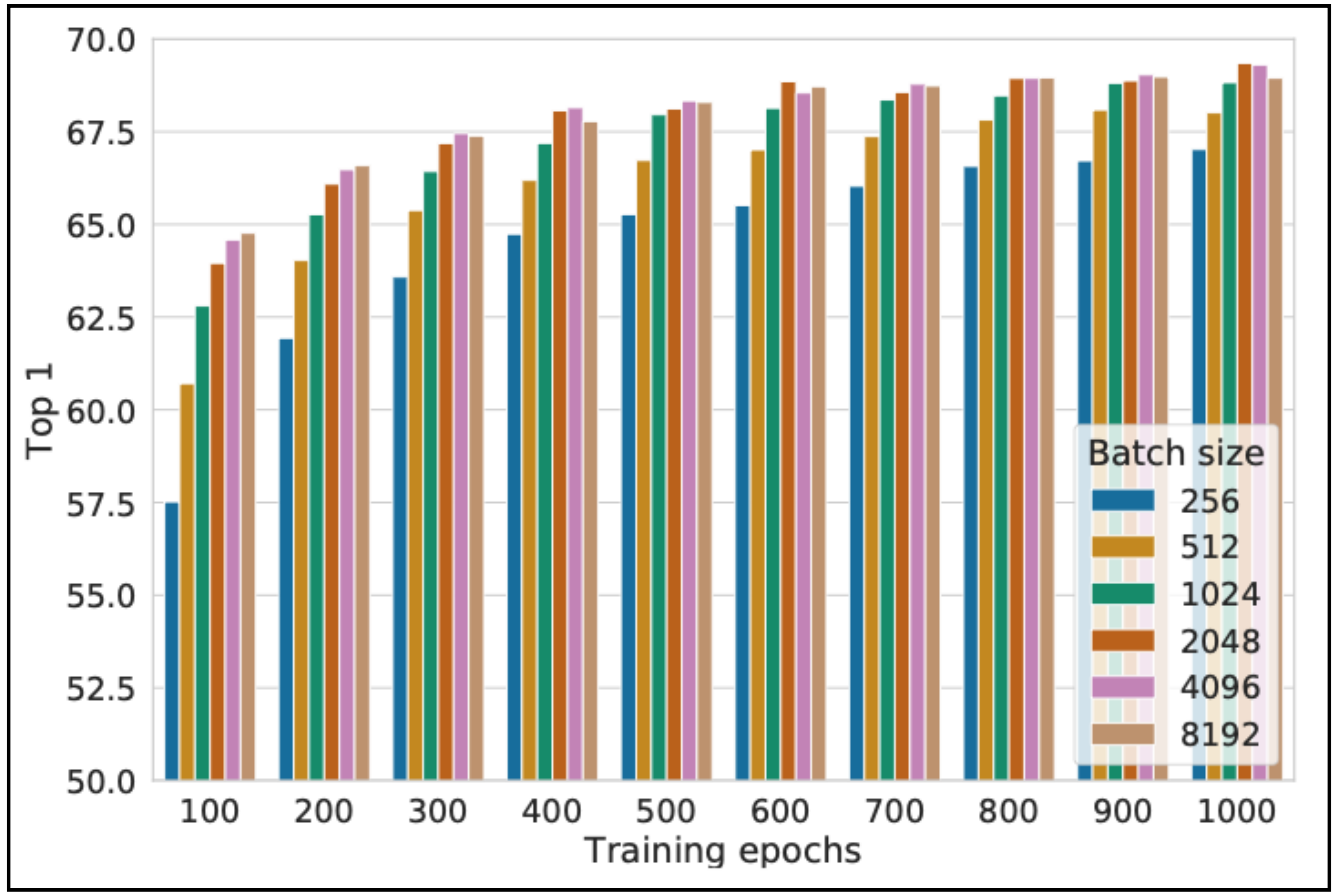
$$\sum_{batches} \log \sum_{i} \exp(-d_{x,y_i^-}) \leq \log \sum_{batches} \sum_{i} \exp(-d_{x,y_i^-})$$

- We are only optimising a lower bound of the actual objective!

[Wiki entry on Jensen's inequality]

The effect of batch size

- **Typically large batch sizes are required**

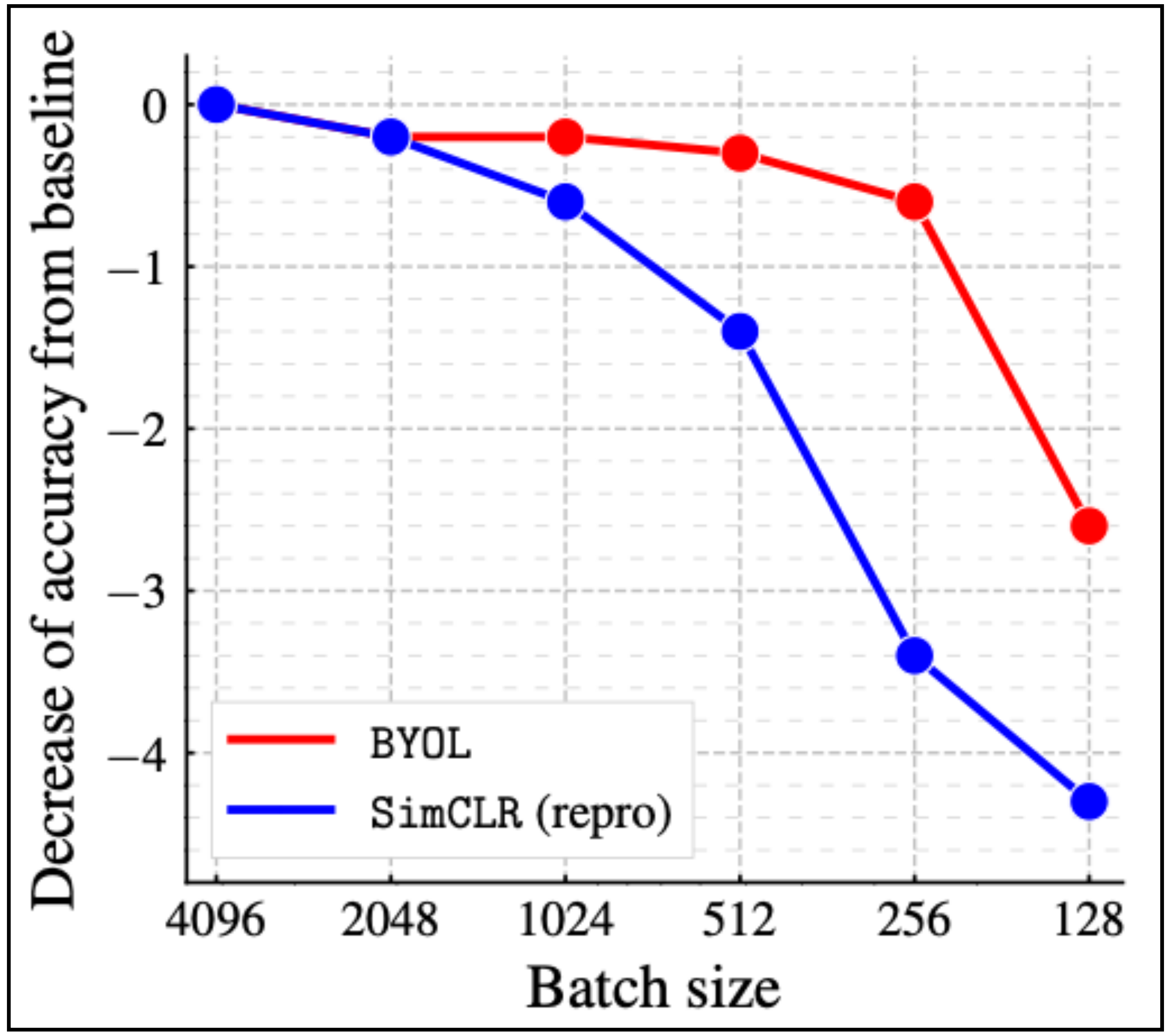


Performance of SimCLR as a function of batch size and epochs. From [Chen et al. | PMLR '20]

The effect of batch size

- **Typically large batch sizes are required**
- One possible way around this:

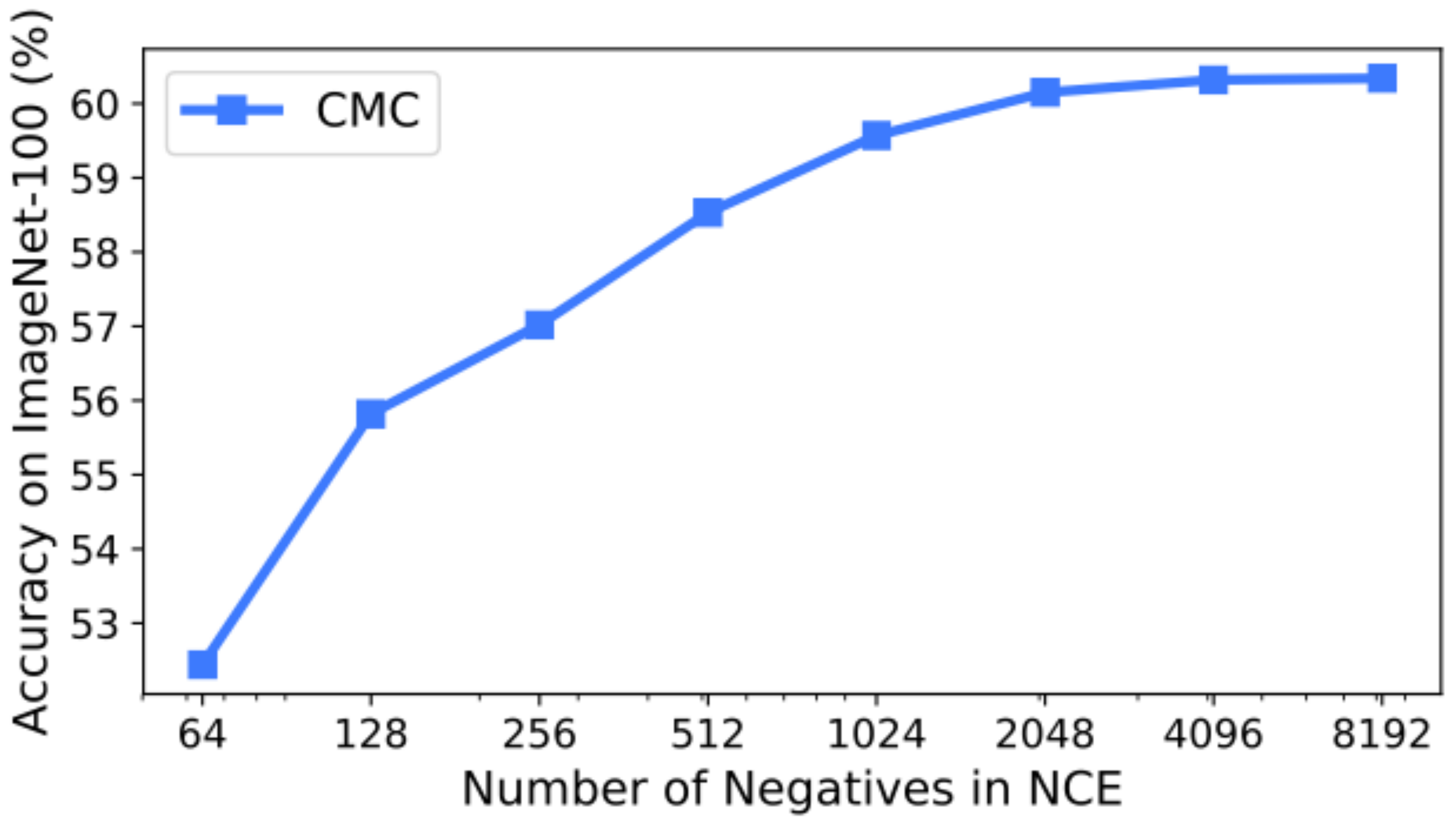
  Non-contrastive learning, i.e. BYOL (later!)



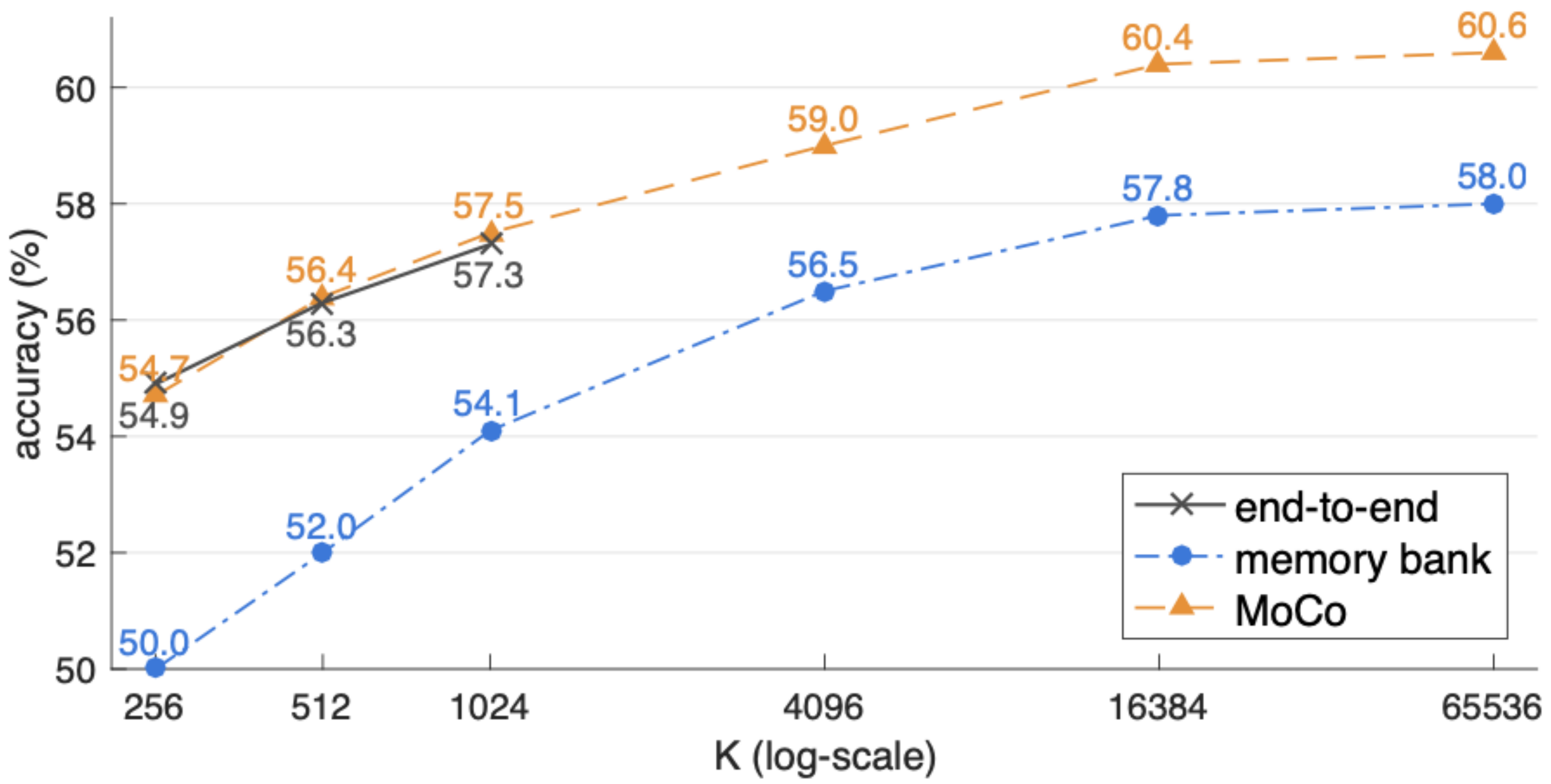
Comparing performance of BYOL vs. SimCLR for small batch sizes. From [Grill et al. | Neurips '20]

The effect of the number of negative samples

- Increasing the number of negative samples tends to increase performance



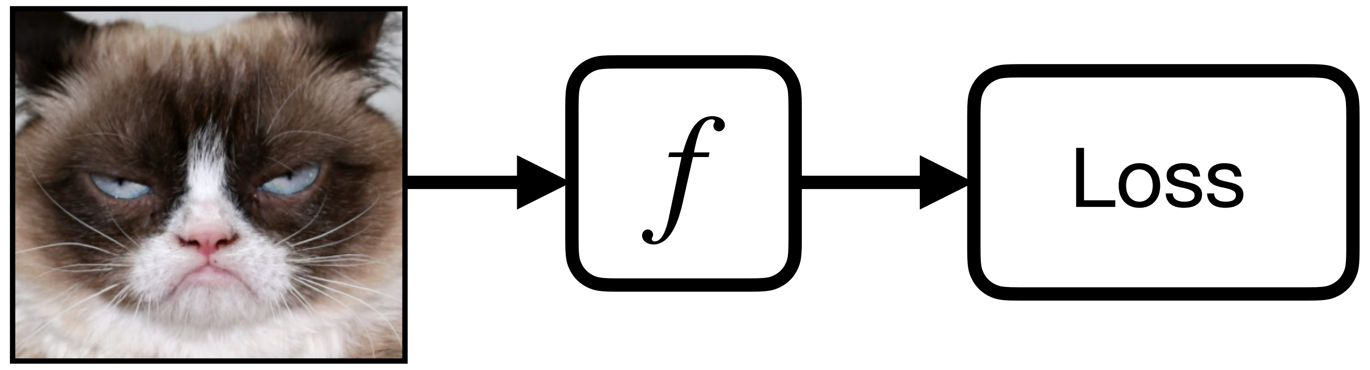
From [Tian, Krishnan, Isola | ECCV '20]

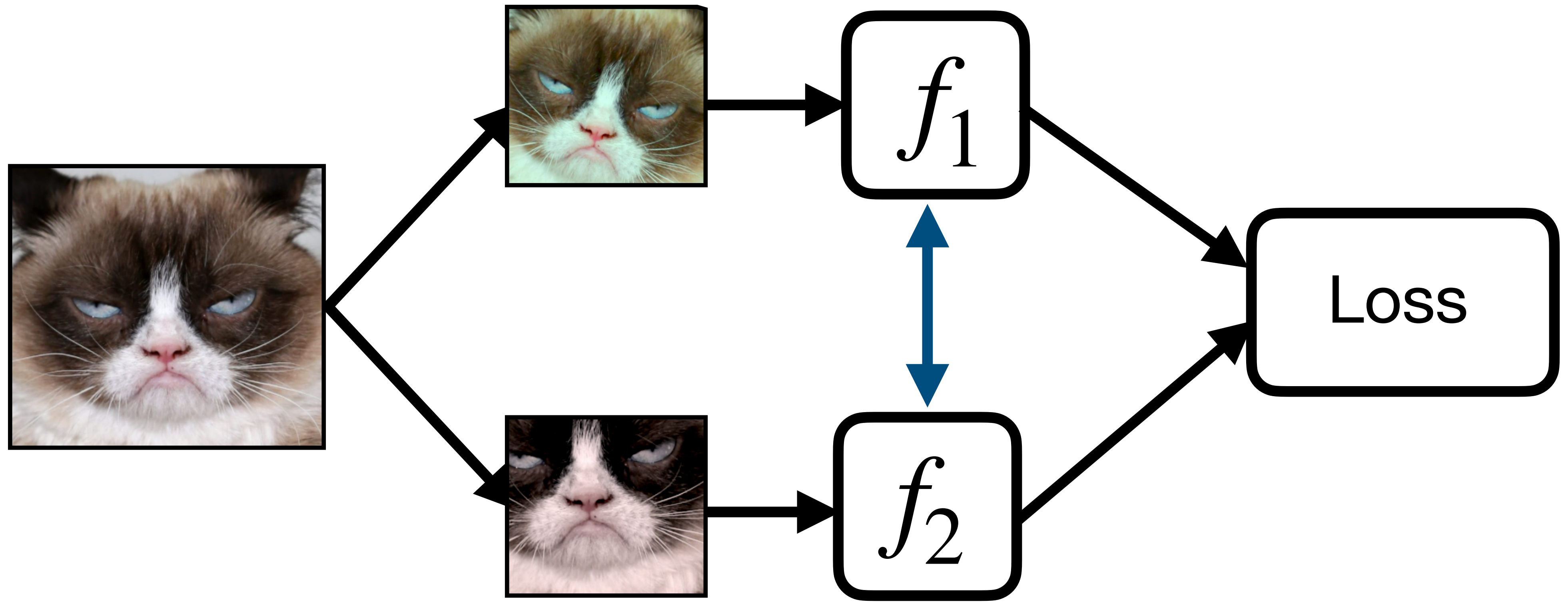


K = number of negative samples.
From [He et al. | CVPR '20]

Non-contrastive learning

- Siamese networks: twin networks joined by a loss function at the top

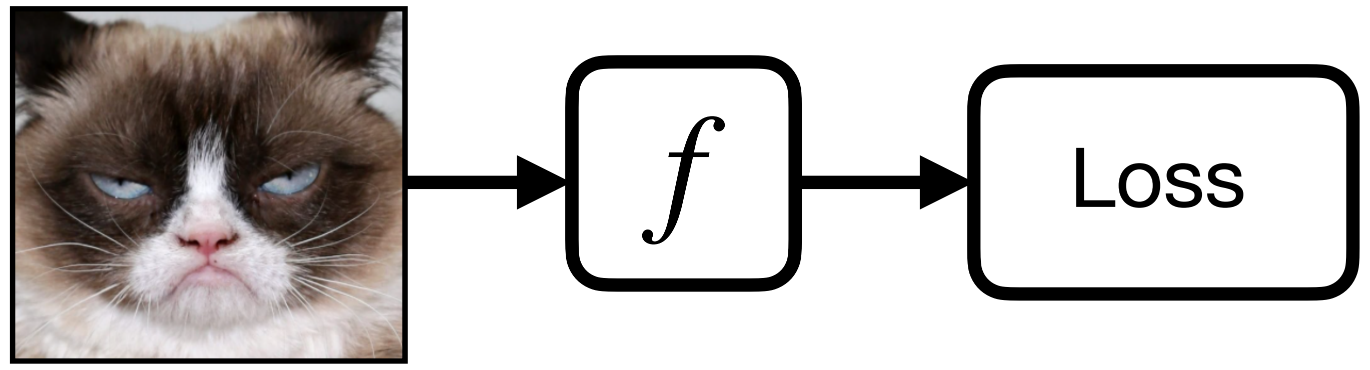


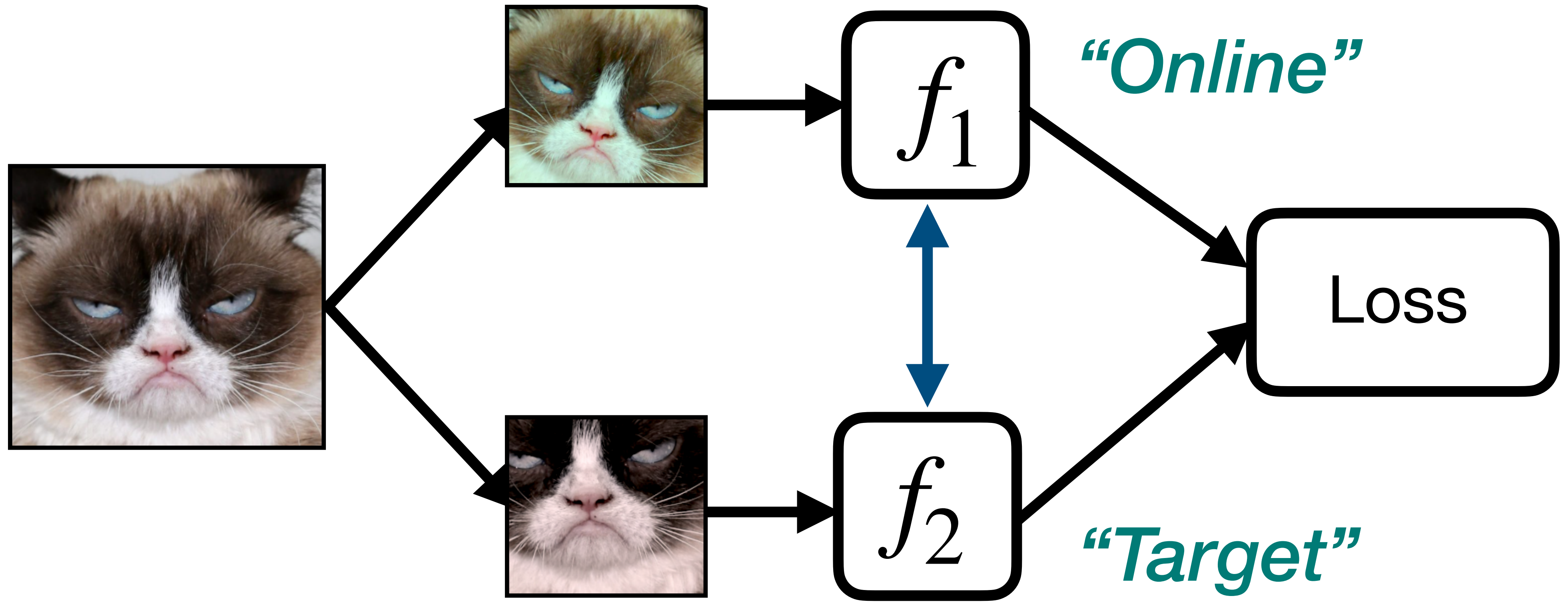


References: [Wiki entry on Siamese networks], [PyTorch documentation on stop gradient]

Non-contrastive learning

- Siamese networks: twin networks joined by a loss function at the top

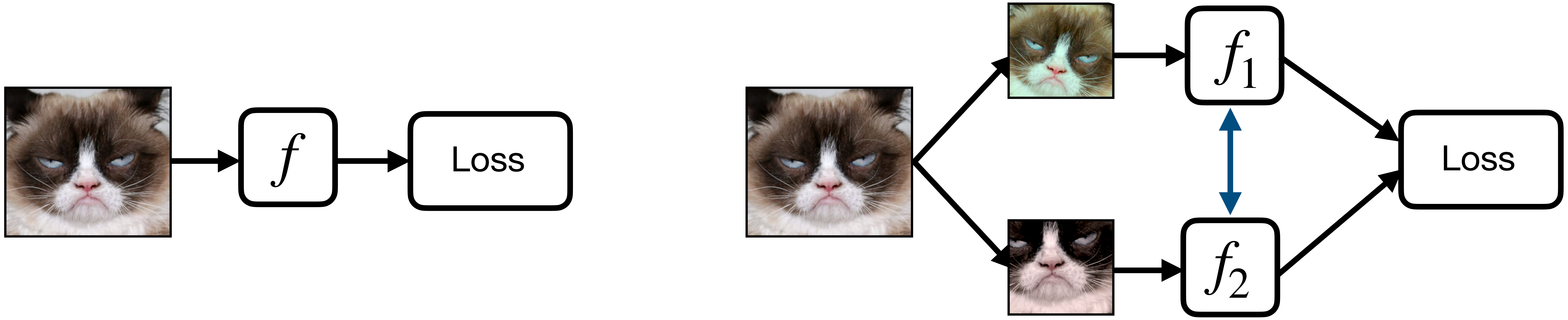




References: [Wiki entry on Siamese networks], [PyTorch documentation on stop gradient]

Non-contrastive learning

- Siamese networks: twin networks joined by a loss function at the top



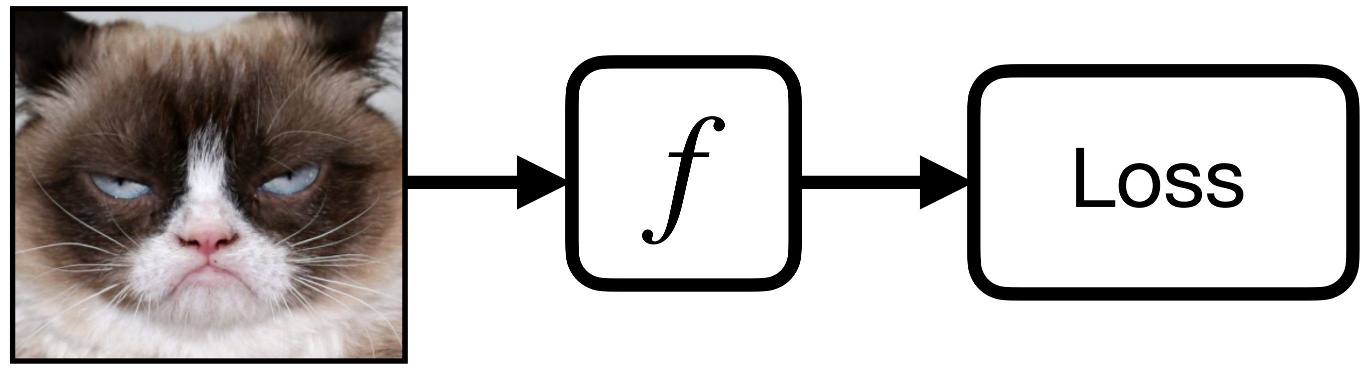
- Ways to **link** the dual networks: let f_i be parametrised by vector θ_i $(i = 1,2)$

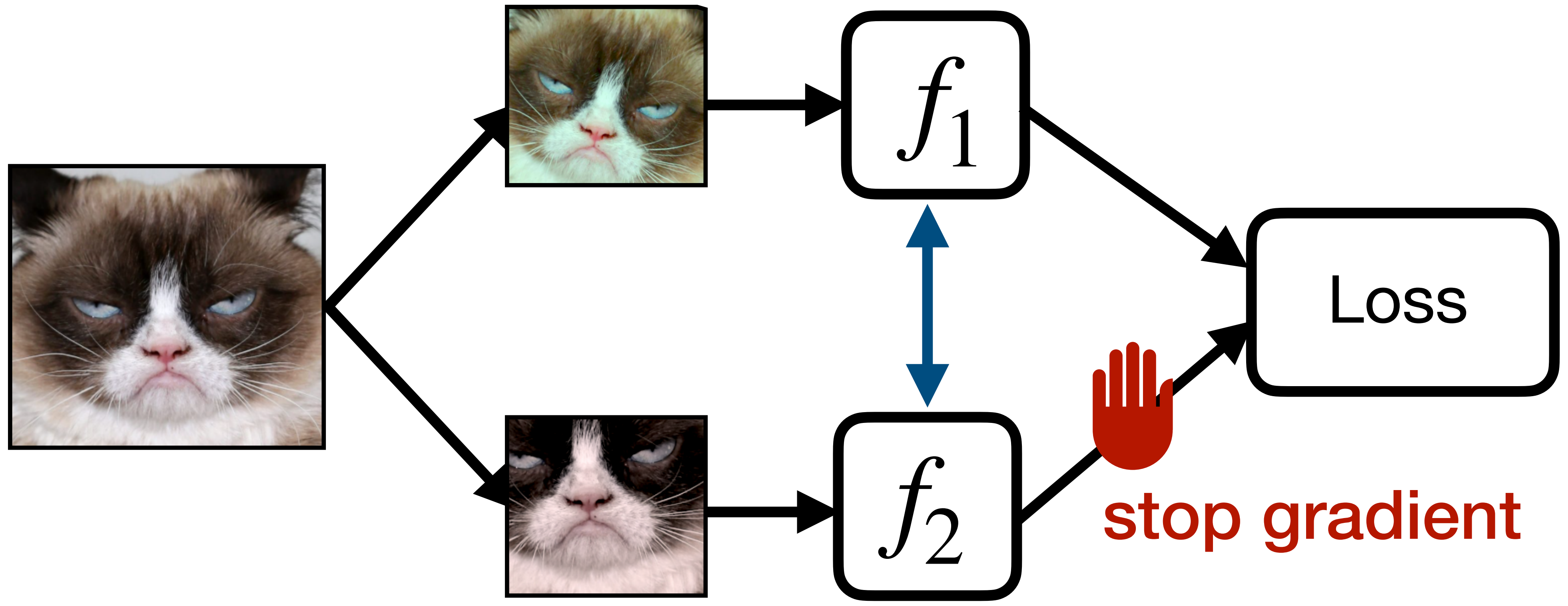
Direct copy	Exponential moving average
$\theta_2 = \theta_1$	$\theta_2 = \alpha\theta_1 + (1 - \alpha)\theta_2$, where $0 \leq \alpha \leq 1$

References: [Wiki entry on Siamese networks], [PyTorch documentation on stop gradient]

Non-contrastive learning

- Siamese networks: twin networks joined by a loss function at the top





- Ways to **link** the dual networks: let f_i be parametrised by vector θ_i $(i = 1,2)$

Direct copy	Exponential moving average
$\theta_2 = \theta_1$	$\theta_2 = \alpha\theta_1 + (1-\alpha)\theta_2$, where $0 \leq \alpha \leq 1$

References: [Wiki entry on Siamese networks], [PyTorch documentation on stop gradient]

Non-contrastive learning: BYOL and SimSiam

Bootstrap Your Own Latent
A New Approach to Self-Supervised Learning

Jean-Bastien Grill*[1] Florian Strub*[1] Florent Altché*[1] Corentin Tallec*[1] Pierre H. Richemond*[1,2]
Elena Buchatskaya[1] Carl Doersch[1] Bernardo Avila Pires[1] Zhaohan Daniel Guo[1]
Mohammad Gheshlaghi Azar[1] Bilal Piot[1] Koray Kavukcuoglu[1] Rémi Munos[1] Michal Valko[1]
[1]DeepMind [2]Imperial College
[jbgrill,fstrub,altche,corentint,richemond]@google.com

Abstract

We introduce **B**ootstrap **Y**our **O**wn **L**atent (BYOL), a new approach to self-supervised image representation learning. BYOL relies on two neural networks, referred to as *online* and *target* networks, that interact and learn from each other. From an augmented view of an image, we train the online network to predict the target network representation of the same image under a different augmented view. At the same time, we update the target network with a slow-moving average of the online network. While state-of-the art methods rely on negative pairs, BYOL achieves a new state of the art *without them.* BYOL reaches 74.3% top-1 classification accuracy on ImageNet using a linear evaluation with a ResNet-50 architecture and 79.6% with a larger ResNet. We show that BYOL performs on par or better than the current state of the art on both transfer and semi-supervised benchmarks. Our implementation and pretrained models are given on GitHub.[3]

Exploring Simple Siamese Representation Learning

Xinlei Chen Kaiming He
Facebook AI Research (FAIR)

Abstract

*Siamese networks have become a common structure in various recent models for unsupervised visual representation learning. These models maximize the similarity between two augmentations of one image, subject to certain conditions for avoiding collapsing solutions. In this paper, we report surprising empirical results that **simple Siamese** networks can learn meaningful representations even using **none** of the following: (i) negative sample pairs, (ii) large batches, (iii) momentum encoders. Our experiments show that collapsing solutions do exist for the loss and structure, but a stop-gradient operation plays an essential role in preventing collapsing. We provide a hypothesis on the implication of stop-gradient, and further show proof-of-concept*

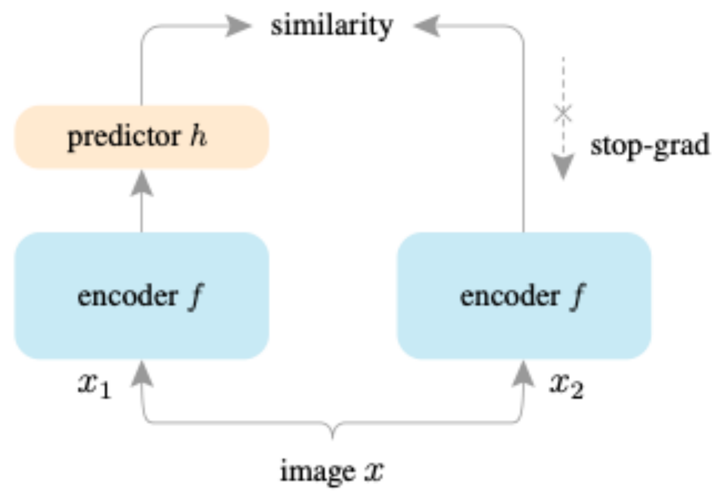
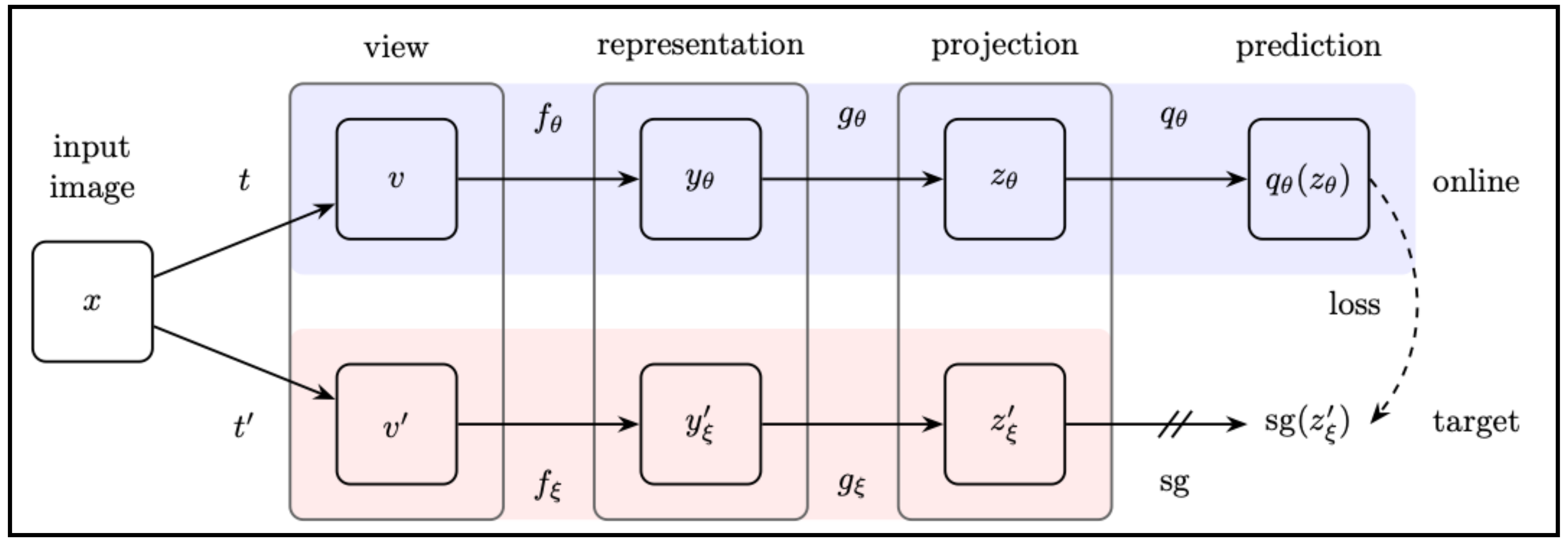


Figure 1. **SimSiam architecture**. Two augmented views of one image are processed by the same encoder network f (a backbone plus a projection MLP). Then a prediction MLP h is applied on one side, and a stop-gradient operation is applied on the other side. The model maximizes the similarity between both sides. It uses neither negative pairs nor a momentum encoder.

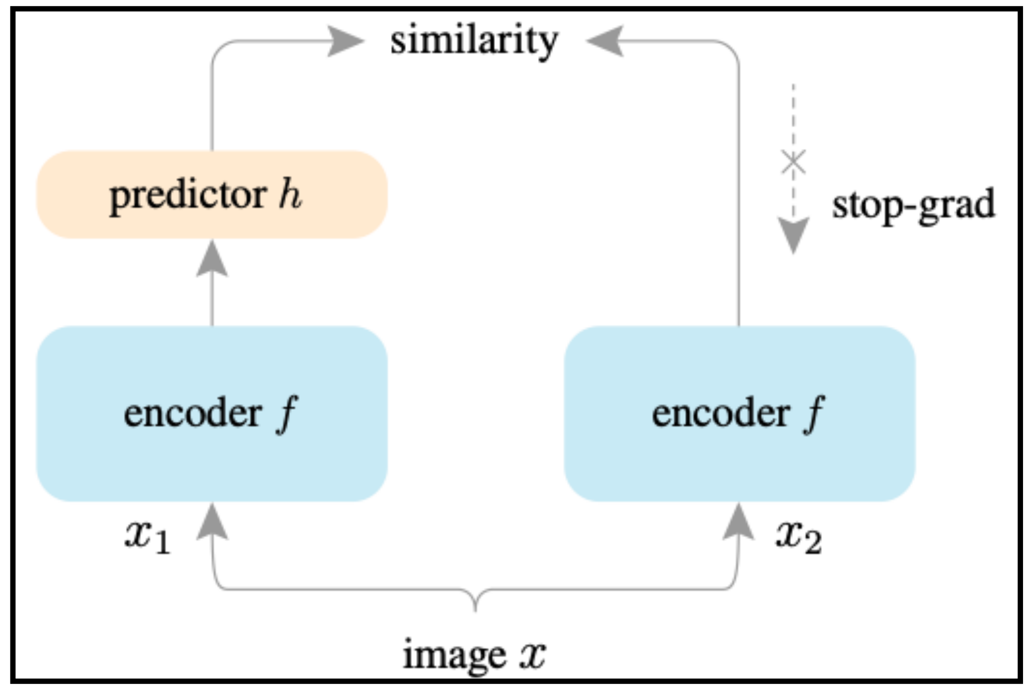
References: [Grill et al. | Neurips ‘20] [Chen, He | IEEE '21]

Non-contrastive learning: BYOL and SimSiam

- Representations produced by two Siamese networks are trained to match
- Target network parameters are updated as:
  - exponential moving average of online parameters (BYOL)
  - Direct copy of online parameters (SimSiam)



Architecture of BYOL

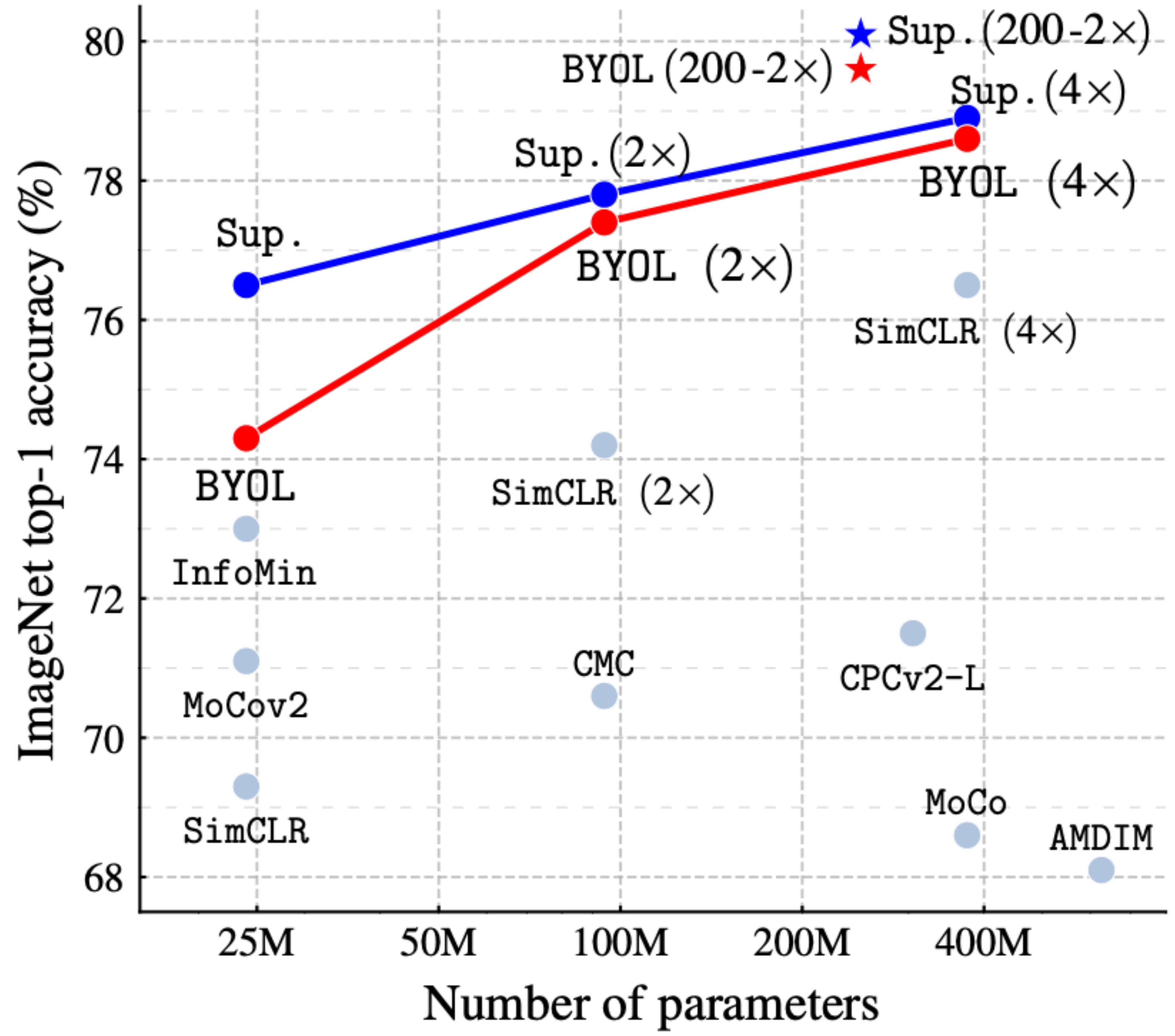


SimSiam architecture

References: [Grill et al. | Neurips '20] [Chen, He | IEEE '21]

Non-contrastive learning: BYOL and SimSiam

- In downstream tasks: representations learned by online network are used
- State-of-the-art performance on ImageNet

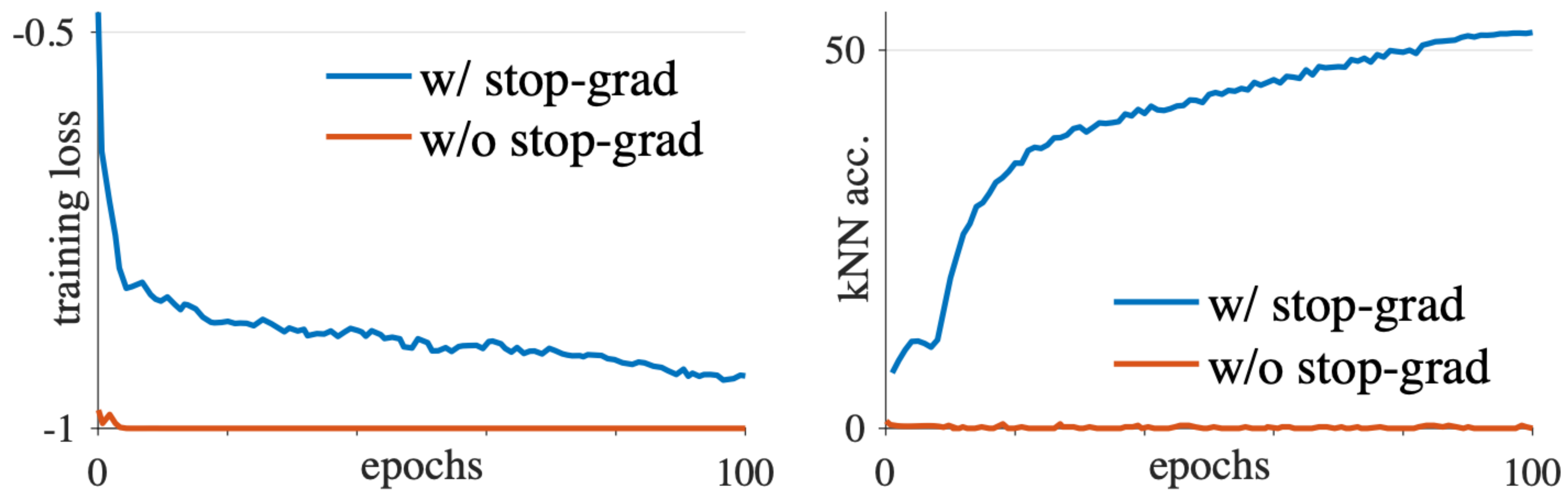


Performance of BYOL and other algorithms as a function of number of parameters

References: [Grill et al. | Neurips ‘20] [Chen, He | IEEE '21]

Non-contrastive learning

- Why does the model not collapse into a trivial (constant) representation?
- *Still a largely unanswered research question!*
- The stop-gradient is crucial to prevent representational collapse

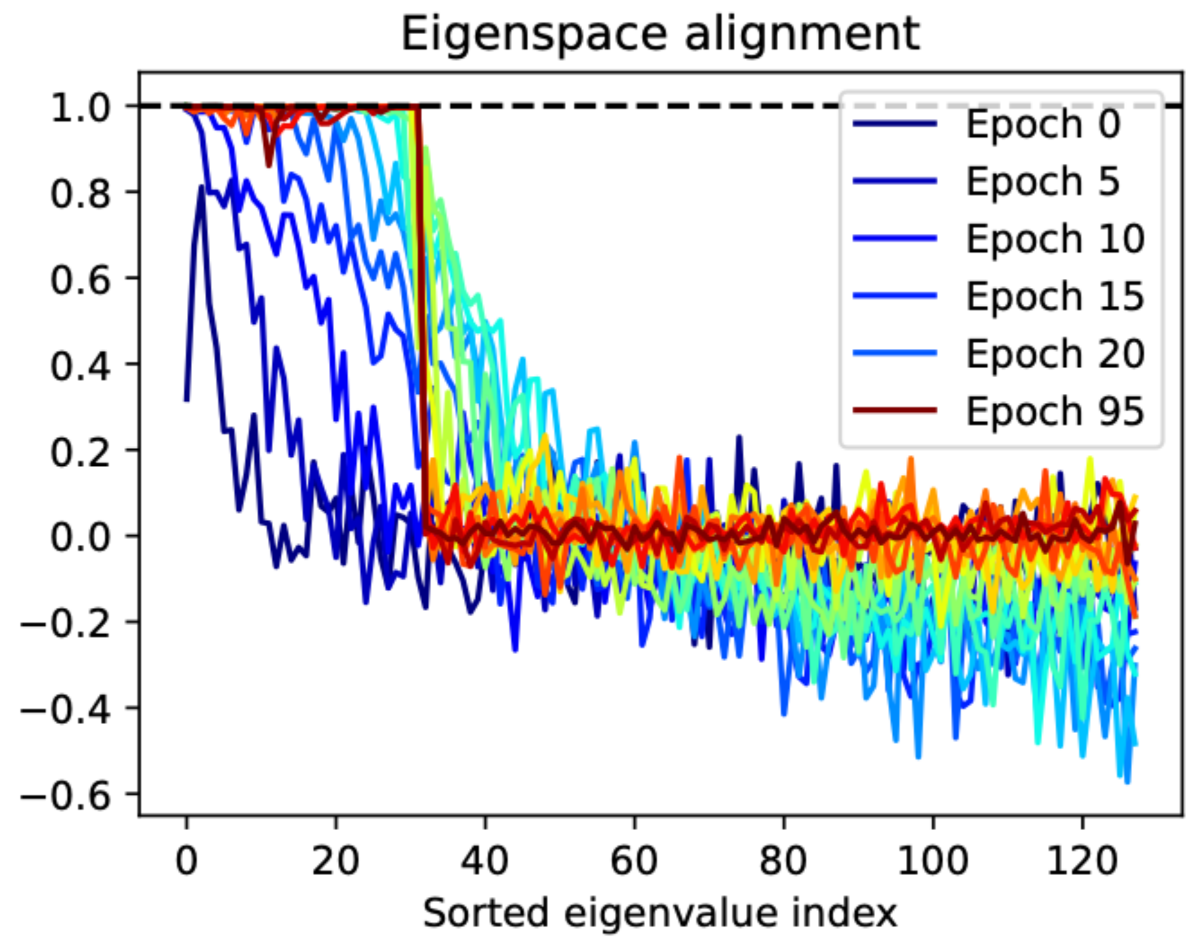


Training loss and kNN accuracy for SimSiam when trained with or w/o stop-gradient; this is reflected in theoretical results

References: [Tian, Chen, Ganguli | ICLR '21] [Chen, He | IEEE '21]

Non-contrastive learning

- Presence of predictor network is crucial to prevent representational collapse
- ‘Eigenspace alignment’ between predictor and the correlation matrix of the outputs of the online network



References: [Tian, Chen, Ganguli | ICLR '21] [Chen, He | IEEE '21]