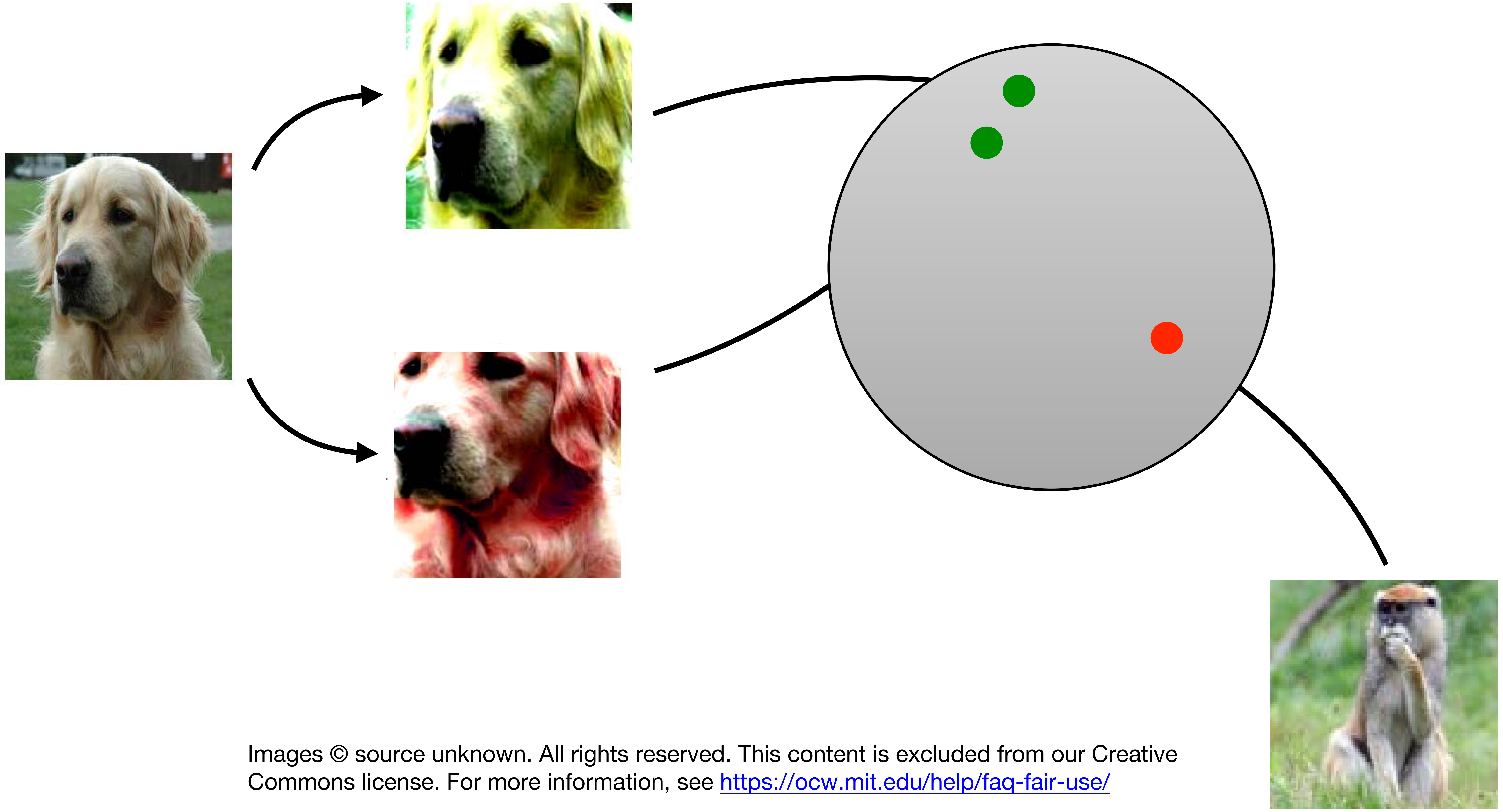
Lecture 12: Similarity-based Representation Learning

Speaker: Sara Beery



Roadmap: similarity-based representation learning

- Representation learning — why?
- What is a “good” representation?
- Metric learning
- Contrastive representation learning (self-supervised)
  - What does it do?
  - Models

Why learn representations?

- To improve generalization
- To do more learning (transfer learning)
- To exploit geometric similarity for new data or queries:
  - Have we seen the face of this person before or is it new?
  - Retrieval: which items are similar to the query?
- To improve clustering with side information (similar/dissimilar pairs)
- Dimensionality reduction (often unsupervised)

What do we expect from such representations?

What is a “good” representation?

“Generally speaking, a good representation is one that makes a subsequent learning task easier.” — *Deep Learning*, Goodfellow et al. 2016

What could this mean?

What is a "good" representation?

1. Compact (*minimal*)

2. Explanatory (*sufficient*)

What is a "good" representation?

NeurIPS 2020 Competition:
Predicting Generalization in Deep Learning (Version 1.1)

Yiding Jiang [*,†] Pierre Foret[†] Scott Yak[†] Daniel M. Roy[‡,§]
Hossein Mobahi[†,§] Gintare Karolina Dziugaite[¶,§] Samy Bengio[†,§]
Suriya Gunasekar[‖,§] Isabelle Guyon [*,§] Behnam Neyshabur[†,§]

pgdl.neurips@gmail.com

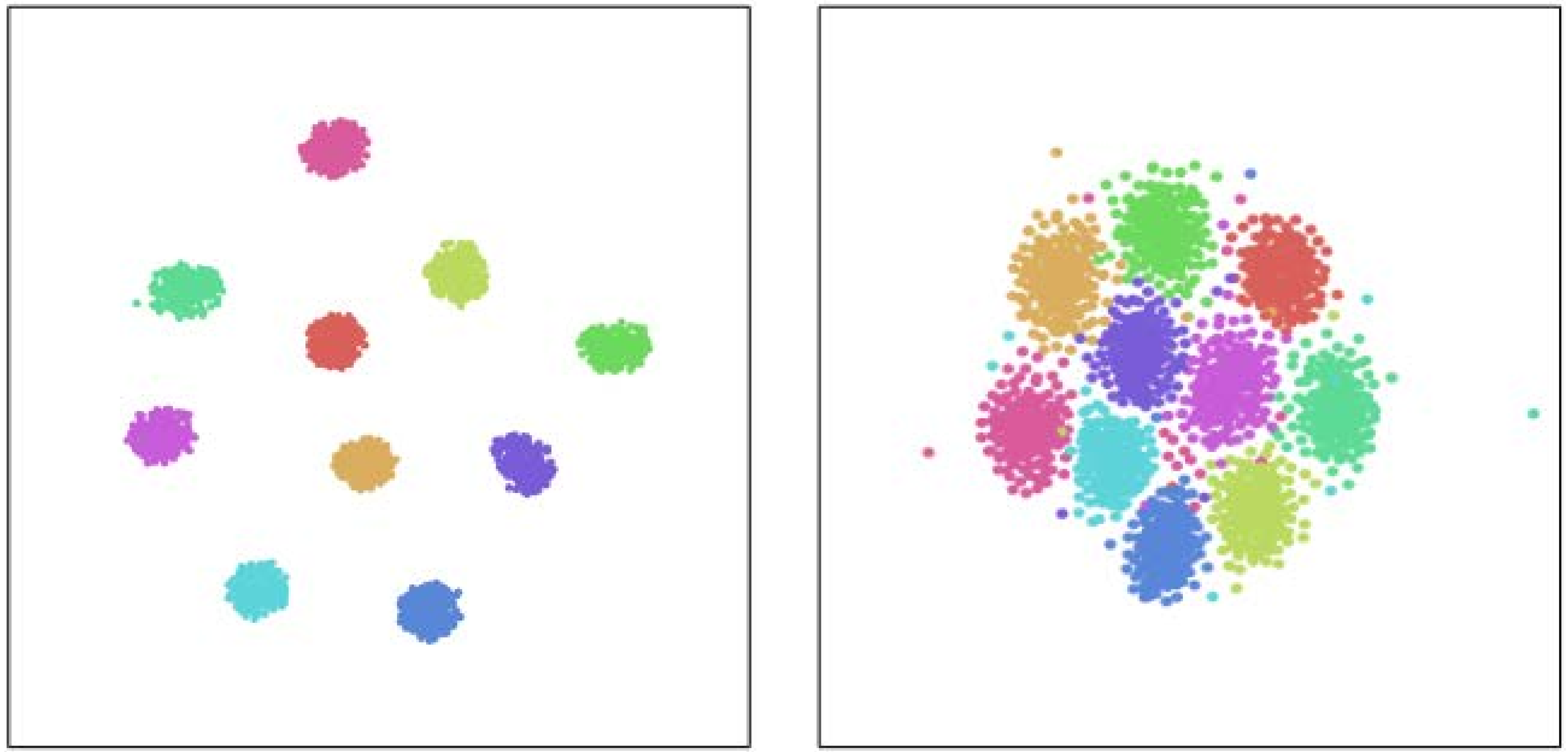
December 16, 2020

3 winning strategies look at:

- Geometry of representation: consistency, separation
- Robustness to perturbations

What helps generalization?

- Representations of CIFAR-10 data with true and random labels



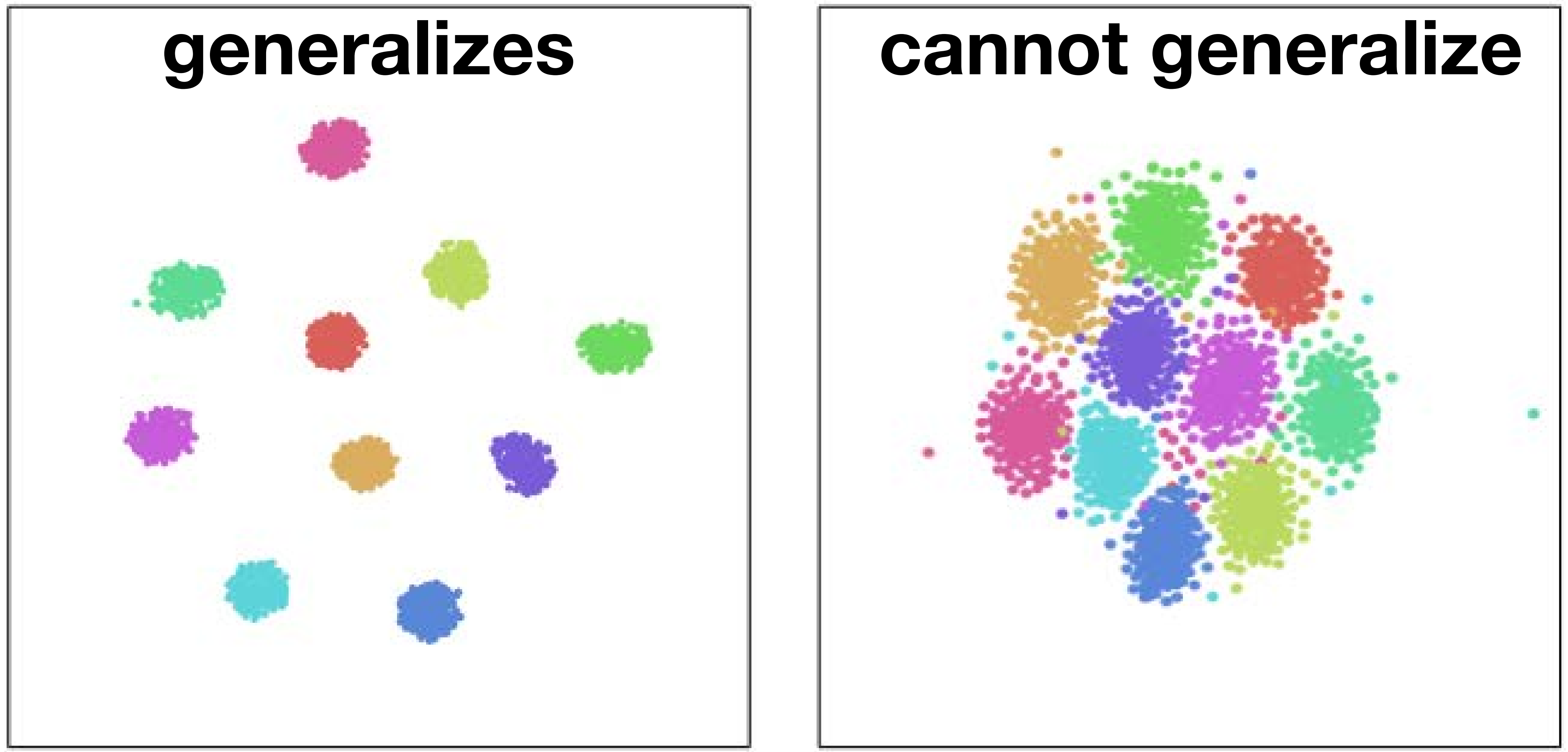
(a) Clean Labels (b) Random Labels

Figure 4: t-SNE visualization of representations. Classes are indicated by colors.

image: Chuang et al., Measuring generalization with optimal transport, 2021

What helps generalization?

- Representations of CIFAR-10 data with true and random labels



(a) Clean Labels (b) Random Labels

Figure 4: t-SNE visualization of representations. Classes are indicated by colors.

Concentration/consistency: Data from the same class is close together

Separation: classes are well separated

Robustness

image: Chuang et al., Measuring generalization with optimal transport, 2021

What is a "good" representation?

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Concentration: Data from the same class is close together
4. Separation: classes are well separated
5. Robustness to irrelevant perturbations

How could we encourage a model during training to achieve this?

Similarity-based representation learning

- Encourage good representations via feedback in terms of similarity: pairs of similar/dissimilar inputs

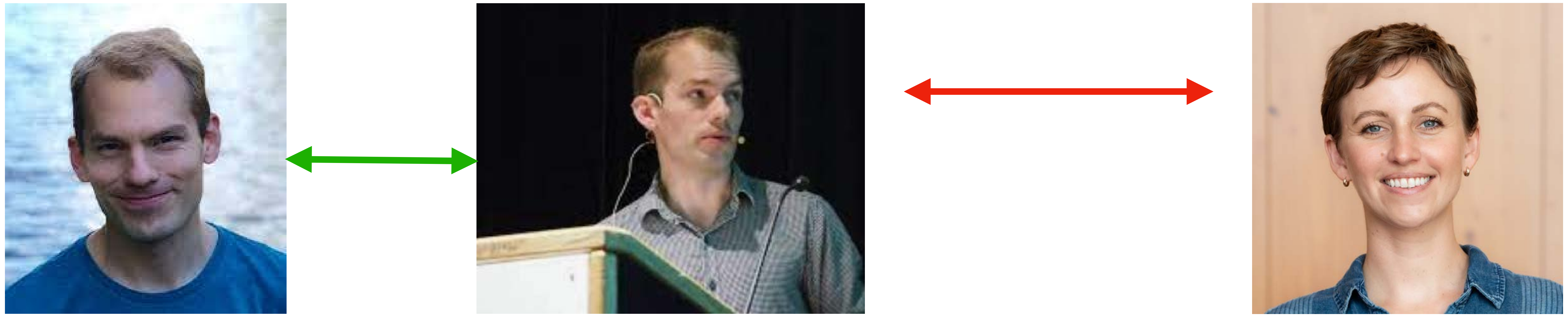
Unsupervised

Supervised

Metric Learning

- Euclidean distance in input space may be not ideal
- Instead: learn a metric that respects desired properties

- Goal: learn a metric where:
  - data points that "belong together" are *similar* (close together)
  - data points that are "different" are dissimilar (far apart)
- "Supervision": similarity information.



Metric learning (linear)

- Data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$

- Weak supervision: $\mathcal{S} := \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are in the same class}\}$ *similar*
  $\mathcal{D} := \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are in different classes}\}$ *dissimilar*

- **Goal:** learn a linear transformation $\mathbf{z} = \mathbf{W}\mathbf{x}$ that respects similarity

- Use Euclidean distance in representation space:

$$\|z_i - z_j\|^2 = (x_i - x_j)^\top W^\top W (x_i - x_j) \qquad A = W^\top W$$

Mahalanobis distance with positive semidefinite matrix $\mathbf{A}$, $d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{A}}$

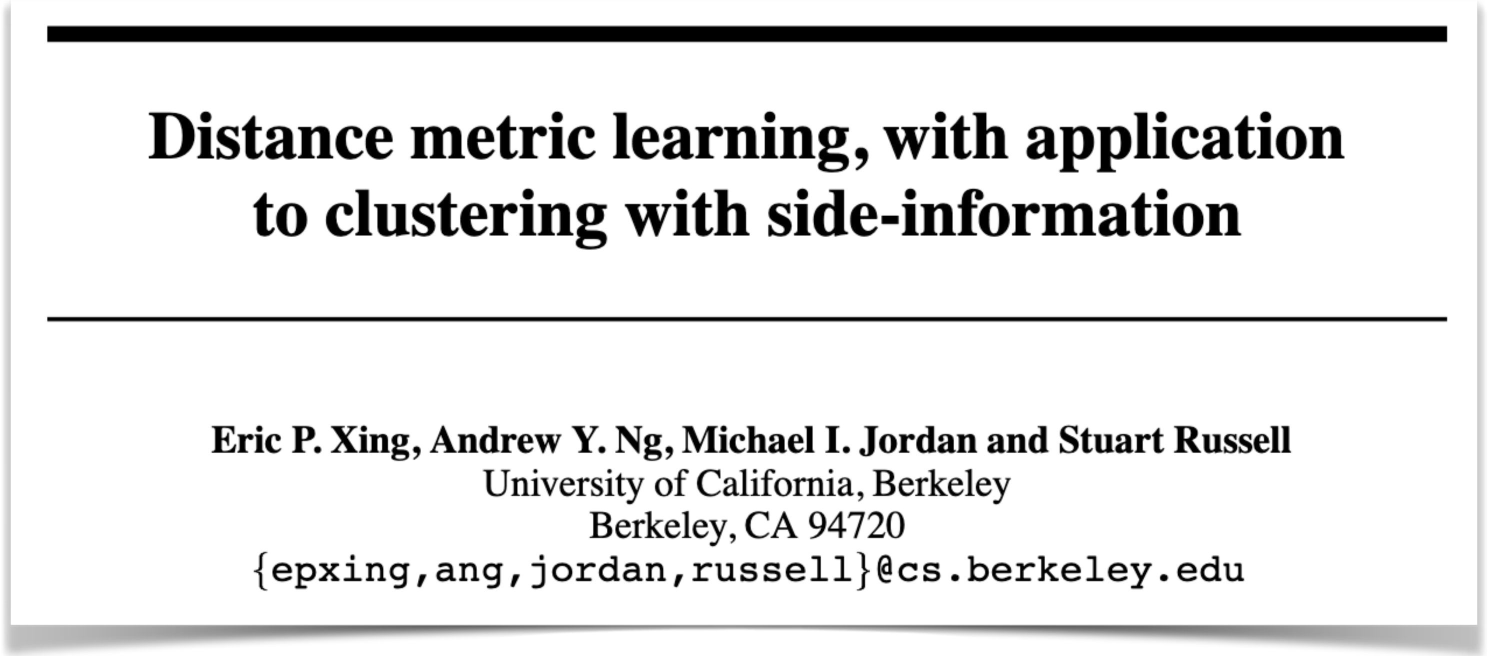
How can we phrase this as an optimization problem?

"Losses": upper/lower bound constraints

- first approach *(Xing et al 2003)*:

$$\min_{A \succeq 0} \sum_{(i,j) \sim S} d_A(x_i, x_j)^2$$ **min distance of similar points**

$$\text{s.t.} \sum_{(k,\ell) \sim D} d_A(x_k, x_\ell)^2 \geq 1$$ **keep distance of dissimilar points**



Distance metric learning, with application to clustering with side-information

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan and Stuart Russell
University of California, Berkeley
Berkeley, CA 94720
{epxing,ang,jordan,russell}@cs.berkeley.edu

introduced the term and problem in 2003

- can swap objective and constraint (upper bound for similar pairs)

- many related ideas & follow-ups, e.g.
information-theoretic metric learning (Davis et al 2007):
preserve distribution information (relative entropy between Gaussians) while observing upper/lower bounds as constraints

Simple example

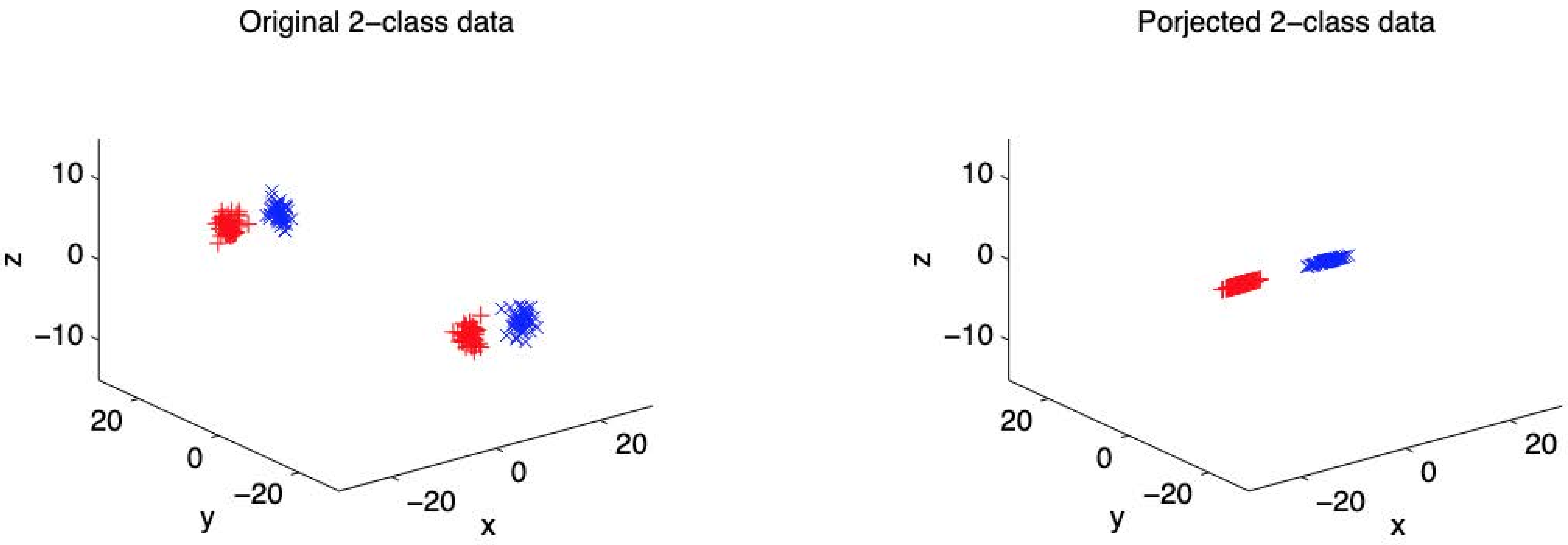


figure: Ng et al 2003

Improvements / developments

- Nonlinear transformations (kernels, deep metric learning)
- Contrastive losses
- Normalization of representations: angle instead of distance

Deep metric learning

- Linear metric learning: learn a linear transformation $\mathbf{z} = \mathbf{W}\mathbf{x}$

- Deep metric learning: learn a nonlinear transformation $\mathbf{z} = f(\mathbf{x})$

neural network

optimize not over psd matrices but weights of a neural network

Contrastive losses: intuition

Contrastive losses

distance of dissimilar pair(s) **distance of similar pair(s)**

- Triplet loss *(Schroff et al 2015)*:

$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \underbrace{\|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2}_{\text{similar}} - \underbrace{\|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2}_{\text{dissimilar}} + \epsilon\right)$$

margin (ϵ)

related: Large-margin Nearest Neighbor metric learning (LMNN) *(Weinberger et al 2009)*

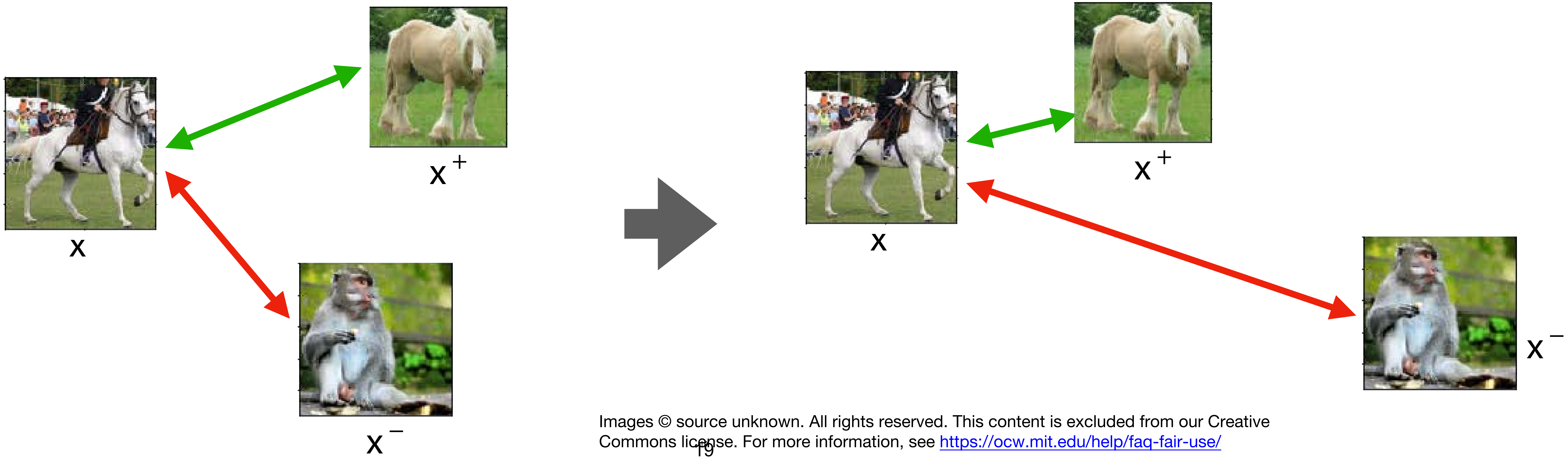
Contrastive losses

distance of dissimilar pair(s) **distance of similar pair(s)**

- Triplet loss *(Schroff et al 2015)*:

$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \underbrace{\|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2}_{} - \underbrace{\|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2}_{} + \epsilon\right)$$

margin



Triplet network

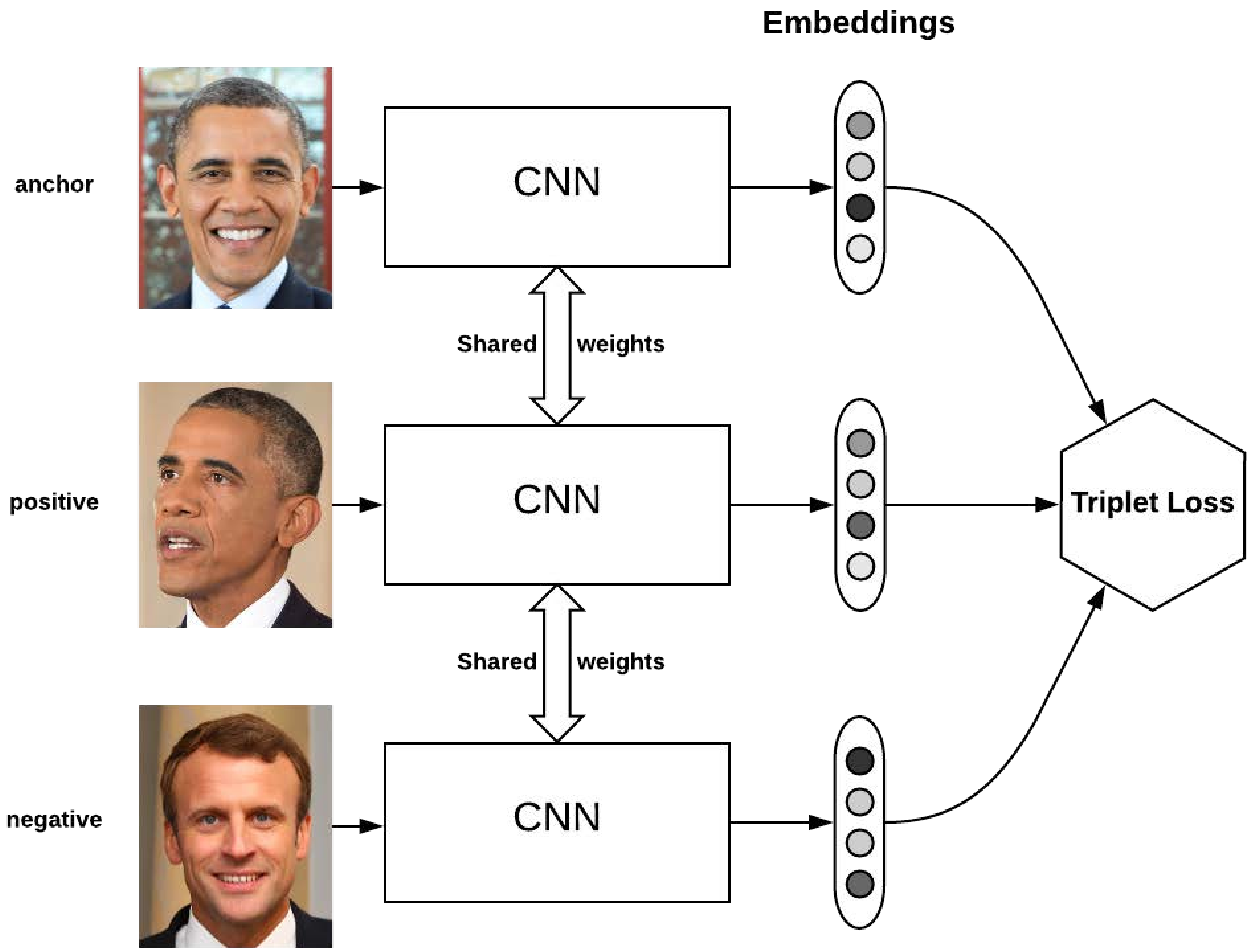


figure: https://omoindrot.github.io/triplet-loss

Contrastive losses

distance of dissimilar pair(s) **distance of similar pair(s)**

- Improvements: compare to multiple negatives per positive pair, e.g. Lifted structured loss *(Song et al 2015)*: compare to all negatives in a batch

$$\mathcal{L}_{\text{struct}} = \frac{1}{2|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max(0, \mathcal{L}^{(ij)}_{\text{struct}})^2$$

$$\text{where } \mathcal{L}^{(ij)}_{\text{struct}} = D_{ij} + \max\left(\max_{(i,k)\in\mathcal{N}} \epsilon - D_{ik}, \max_{(j,l)\in\mathcal{N}} \epsilon - D_{jl} \right)$$

$\|f(x_i) - f(x_j)\|_2$

or smooth relaxation of the max

Example embedding

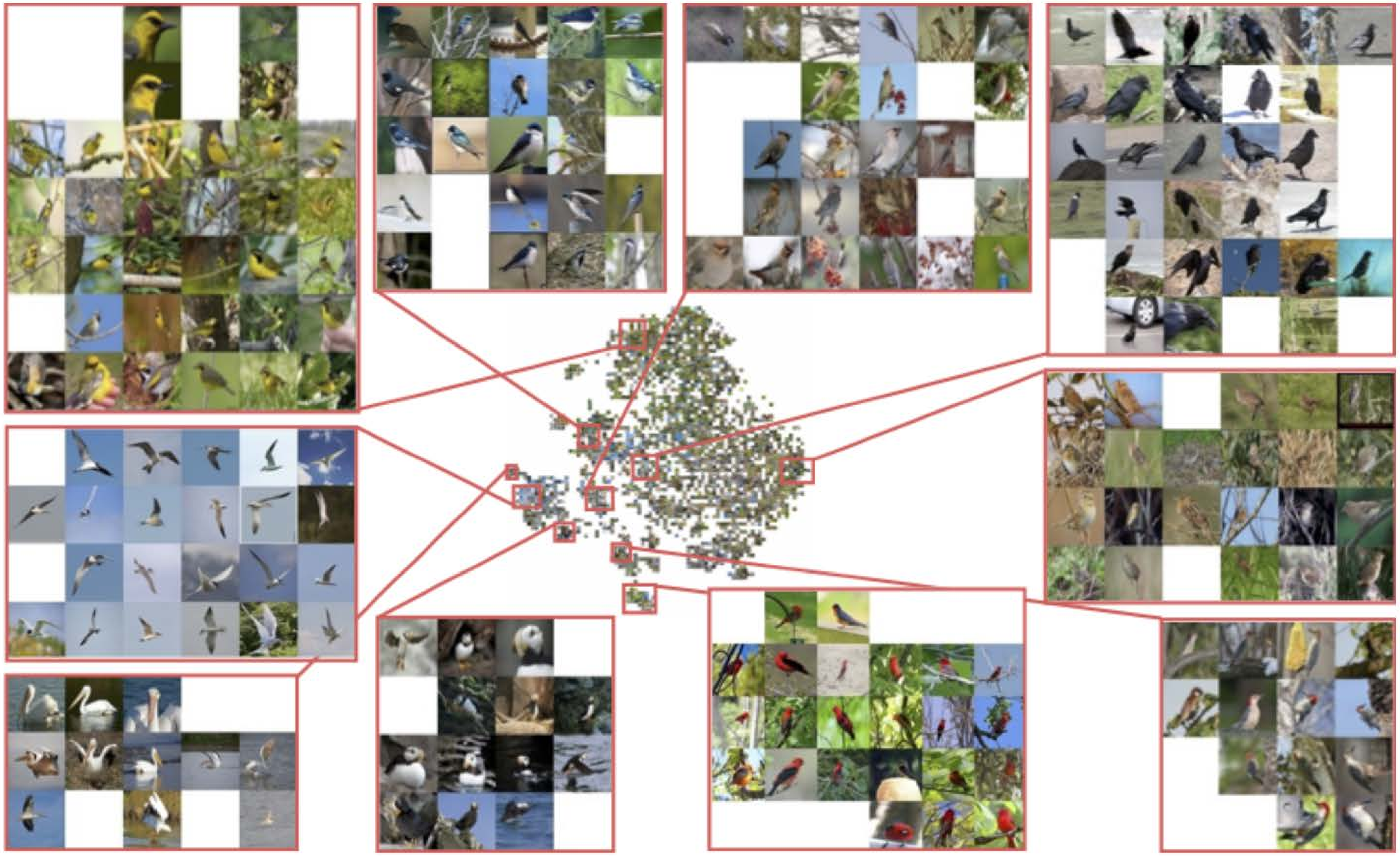


Figure 9: Barnes-Hut t-SNE visualization [36] of our embedding on the test split (class 101 to 200; 5,924 images) of CUB-200-2011. Best viewed on a monitor when zoomed in.

figure: Song et al 2015

Example query results (neighbors)

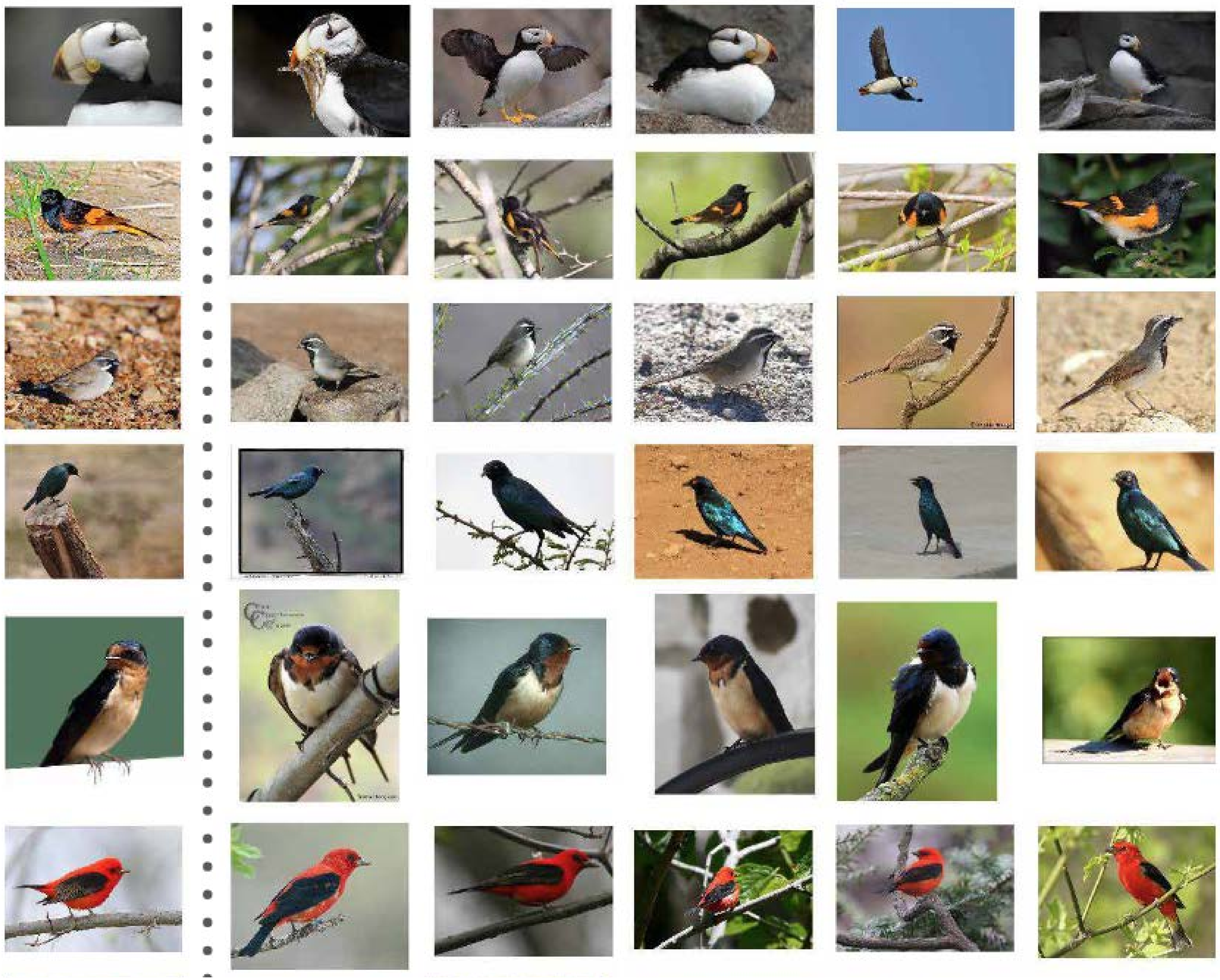
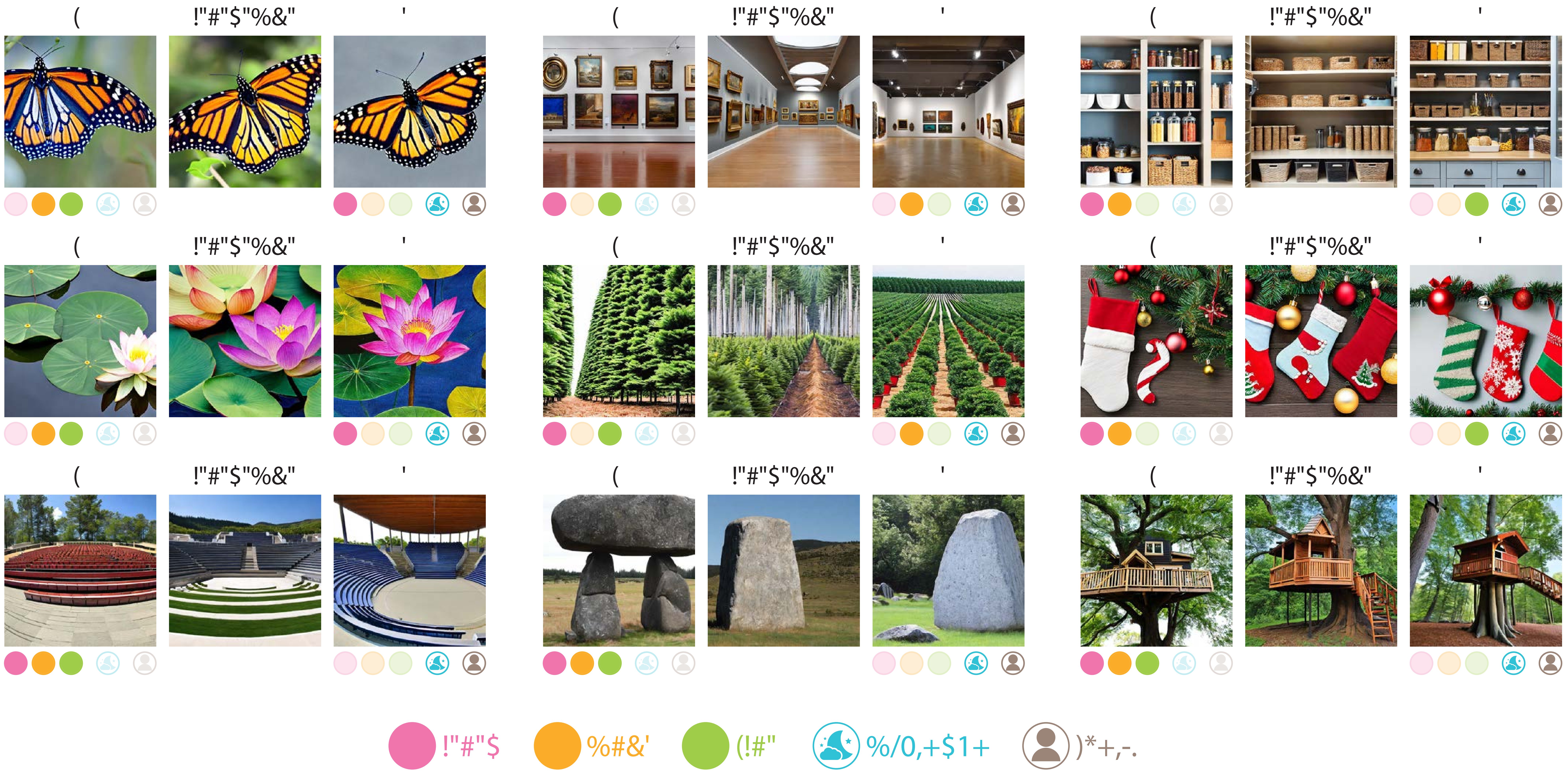


figure: Song et al 2015

What makes an image "similar"?



Similar in:

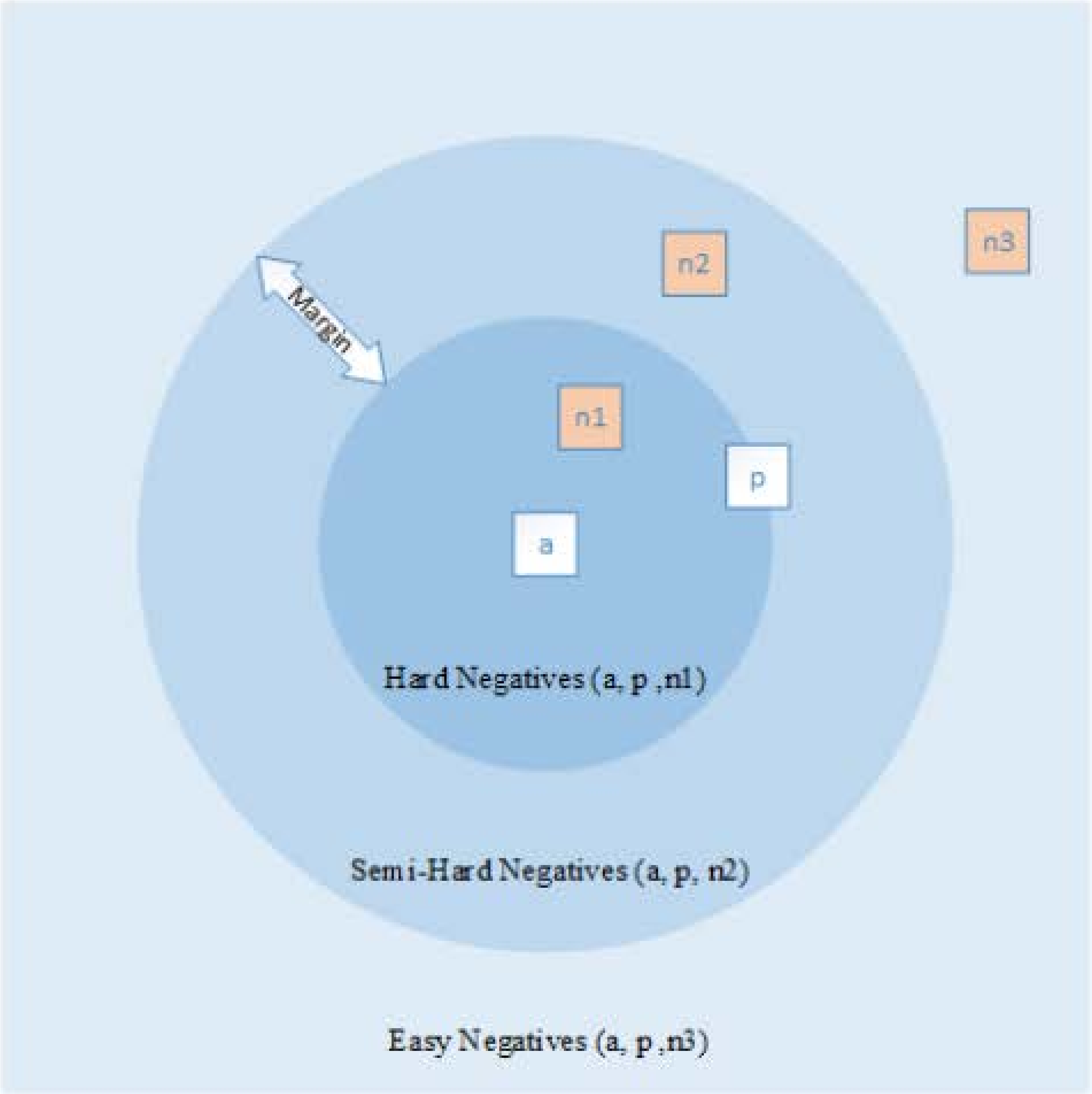
- Pose
- Perspective
- Foreground color
- Number of items
- Object shape

figure: Fu, Tamir*, Sundaram* et al 2023*

Which pairs should we present?

"hard" negatives:

- currently "misplaced", i.e., closer to anchor than a positive example
- accelerate learning, needed for triplet loss



Hard Negative Mining
$d(a,n) < d(a,p)$

Semi-Hard Negative Mining
$d(a,p) < d(a,n) < d(a,p) + margin$

Easy Negative Mining
$d(a,p) + margin < d(a,n)$

Figure 4. Negative Mining.

figure: Kaya & Bilge: Deep Metric Learning: A Survey, 2019

Roadmap: similarity-based representation learning

- Representation learning — why?
- What is a “good” representation?
- Metric learning
- Contrastive representation learning (self-supervised)
  - What does it do?
  - Models

Self-supervised contrastive representation learning

- Ideas from metric learning and self-supervision

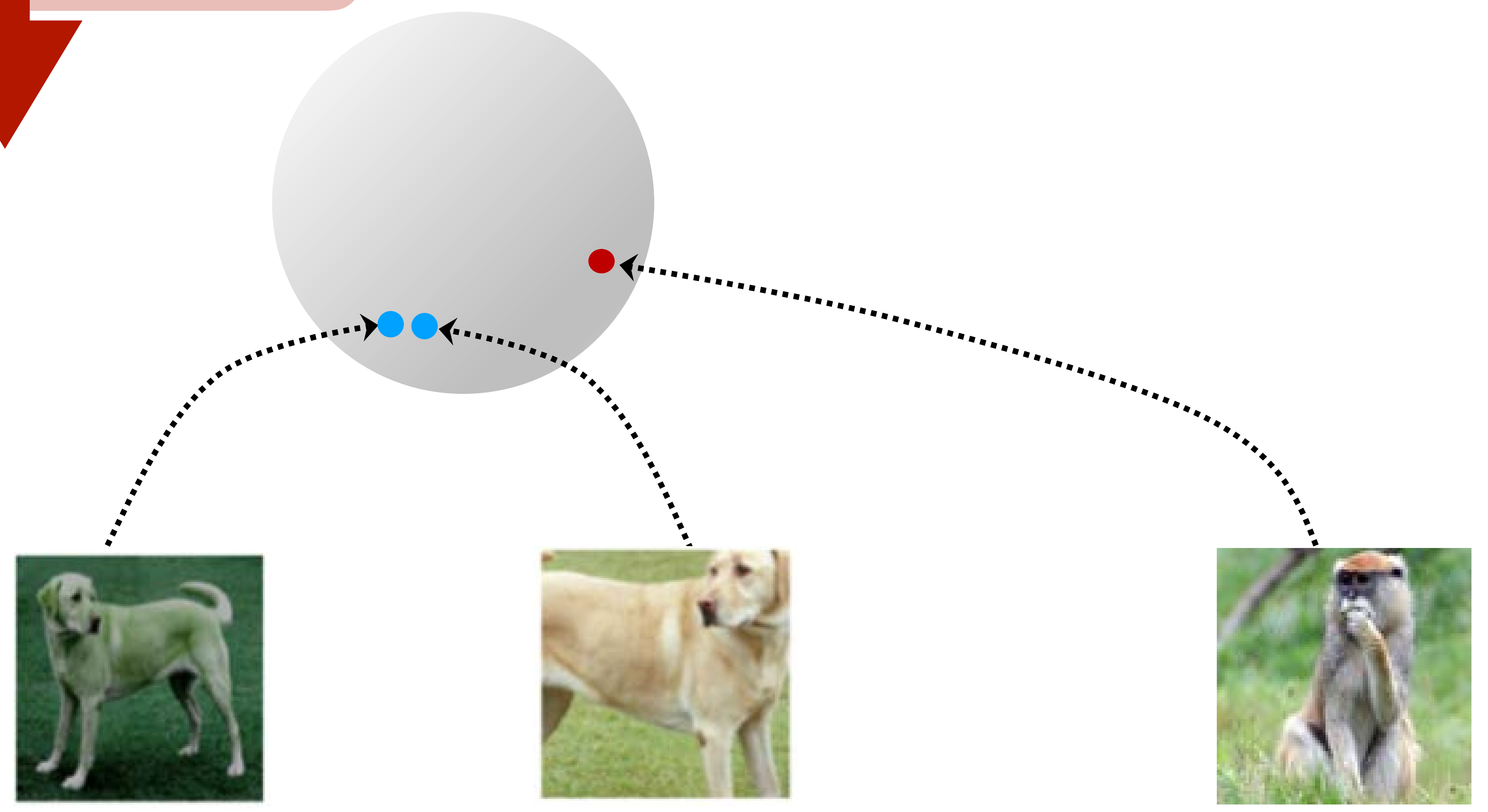
Common setup

- Encoder maps data onto a hypersphere: $f : \mathcal{X} \to \mathbb{S}^{d-1}$
- Cross-entropy for softmax "classifier" to discriminate "classes" defined by similarities

$$\min_{f} \ \mathbb{E}_{(\mathbf{x},\mathbf{x}^+) \sim p_{pos}, \{\mathbf{x}_i^-\}_{i=1}^N \sim p_{data}} \left[-\log \frac{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau}}{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau} + \sum_{i=1}^{N} e^{f(\mathbf{x})^\top f(\mathbf{x}_i^-)/\tau}} \right]$$

pull positive pair together

push negative pairs apart



Symmetry: $\forall \mathbf{x}, \mathbf{x}^+, p_{\mathtt{pos}}(\mathbf{x}, \mathbf{x}^+) = p_{\mathtt{pos}}(\mathbf{x}^+, \mathbf{x})$

Matching marginal: $\forall \mathbf{x}, \int p_{\mathtt{pos}}(\mathbf{x}, \mathbf{x}^+) d\mathbf{x}^+ = p_{\mathtt{data}}(\mathbf{x})$

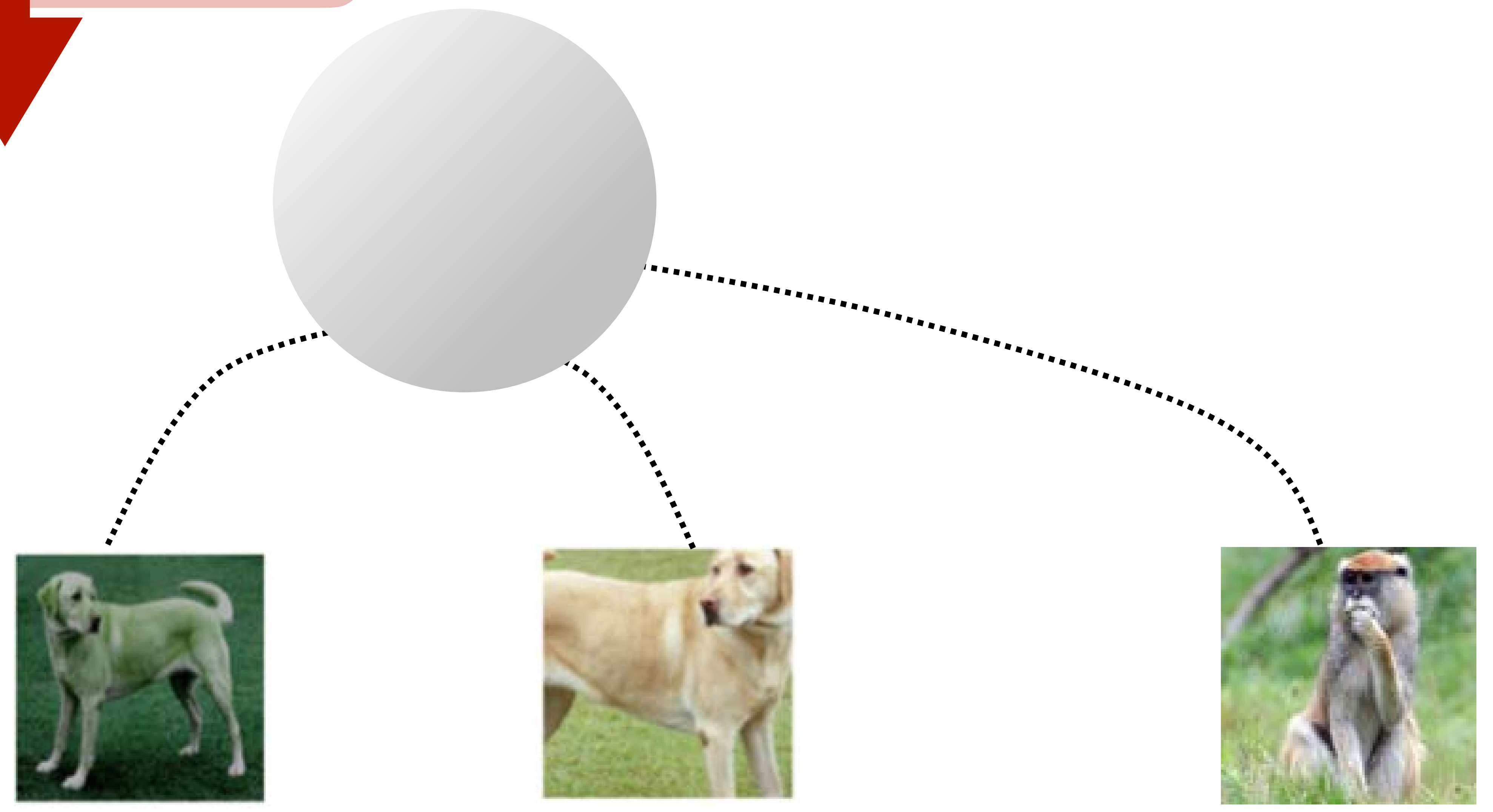
Common setup

- Encoder maps data onto a hypersphere: $f : \mathcal{X} \to \mathbb{S}^{d-1}$
- Cross-entropy for softmax "classifier"

$$\min_f \; \mathbb{E}_{(\mathbf{x},\mathbf{x}^+) \sim p_{pos}, \{\mathbf{x}_i^-\}_{i=1}^N \sim p_{data}} \left[-\log \frac{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau}}{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau} + \sum_{i=1}^{N} e^{f(\mathbf{x})^\top f(\mathbf{x}_i^-)/\tau}} \right]$$

pull positive pair together

push negative pairs apart



- *Noise-contrastive estimation (NCE) (Gutmann & Hyvärinen 2010), InfoNCE loss (van den Oord et al 2018), … similar losses also in metric learning*

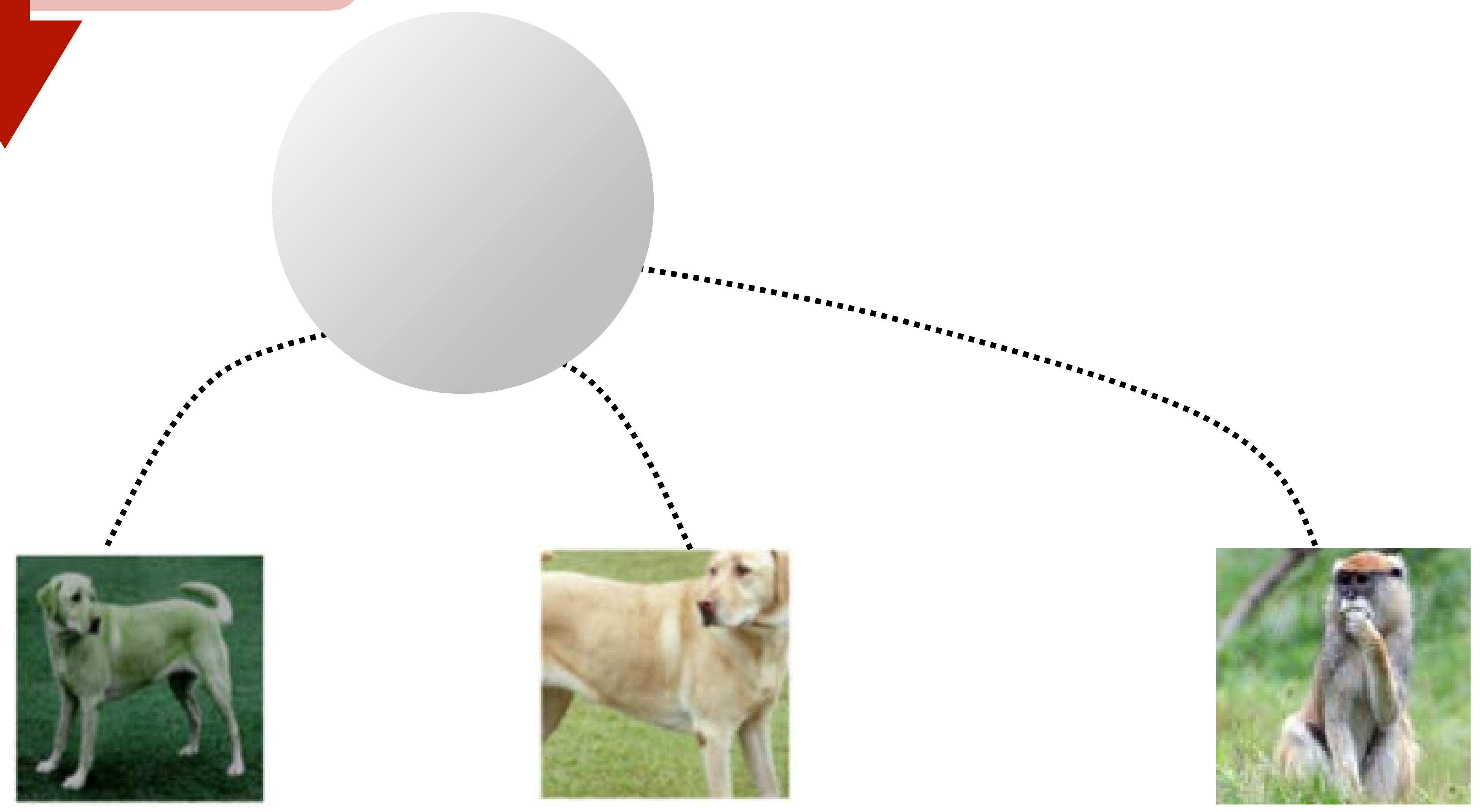
Common setup

- Encoder maps data onto a hypersphere: $f : \mathcal{X} \to \mathbb{S}^{d-1}$
- Cross-entropy for softmax "classifier"

$$\min_f \; \mathbb{E}_{(\mathbf{x},\mathbf{x}^+) \sim p_{pos}, \{\mathbf{x}_i^-\}_{i=1}^N \sim p_{data}} \left[-\log \frac{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau}}{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau} + \sum_{i=1}^N e^{f(\mathbf{x})^\top f(\mathbf{x}_i^-)/\tau}} \right]$$

pull positive pair together

push negative pairs apart



As self-supervised learning, can outperform supervised pre-training (for some tasks) *(He et al 2020, Misra & van der Maaten 2020)*

Why map to a hypersphere?

- more stable training (logistic regression needs regularization)
- well-clustered classes on hypersphere are linearly separable (cut off caps)

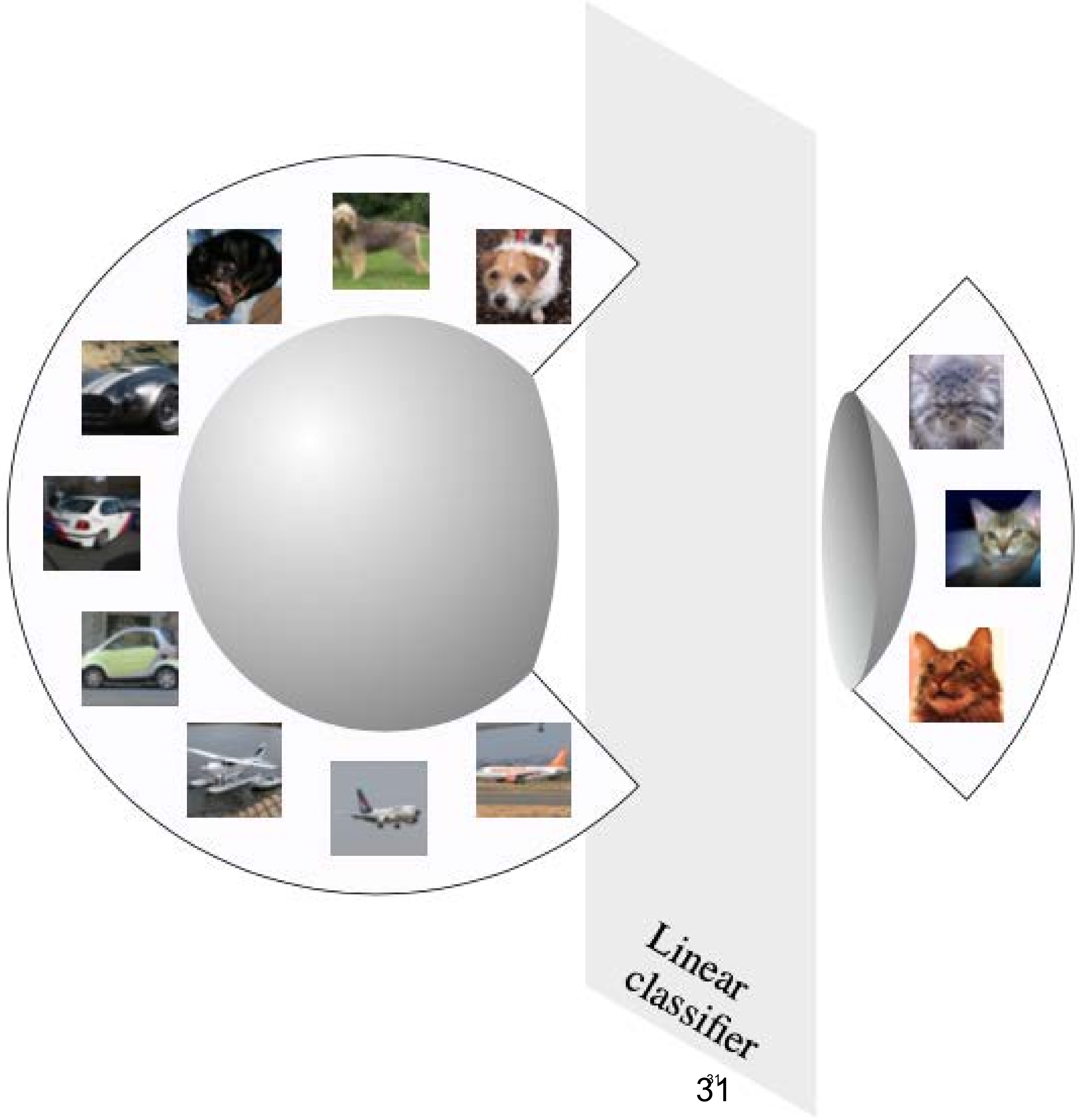


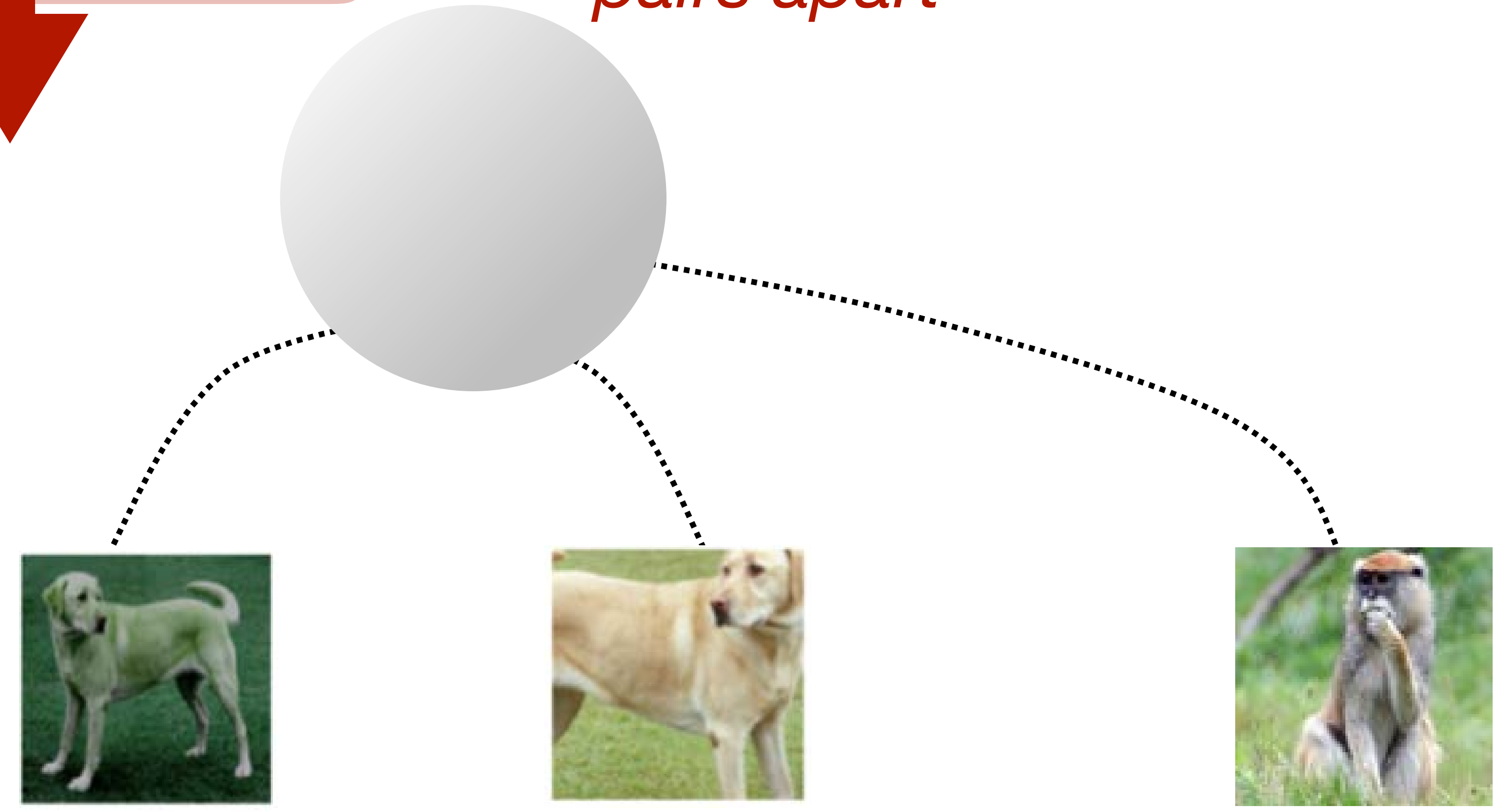
figure: Wang & Isola 2020

How can we make this "self-supervised"?

$$\min_f \; \mathbb{E}_{(\mathbf{x},\mathbf{x}^+)\sim p_{pos},\{\mathbf{x}_i^-\}_{i=1}^N \sim p_{data}} \left[-\log \frac{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau}}{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau} + \sum_{i=1}^{N} e^{f(\mathbf{x})^\top f(\mathbf{x}_i^-)/\tau}} \right]$$

pull positive pair together

push negative pairs apart

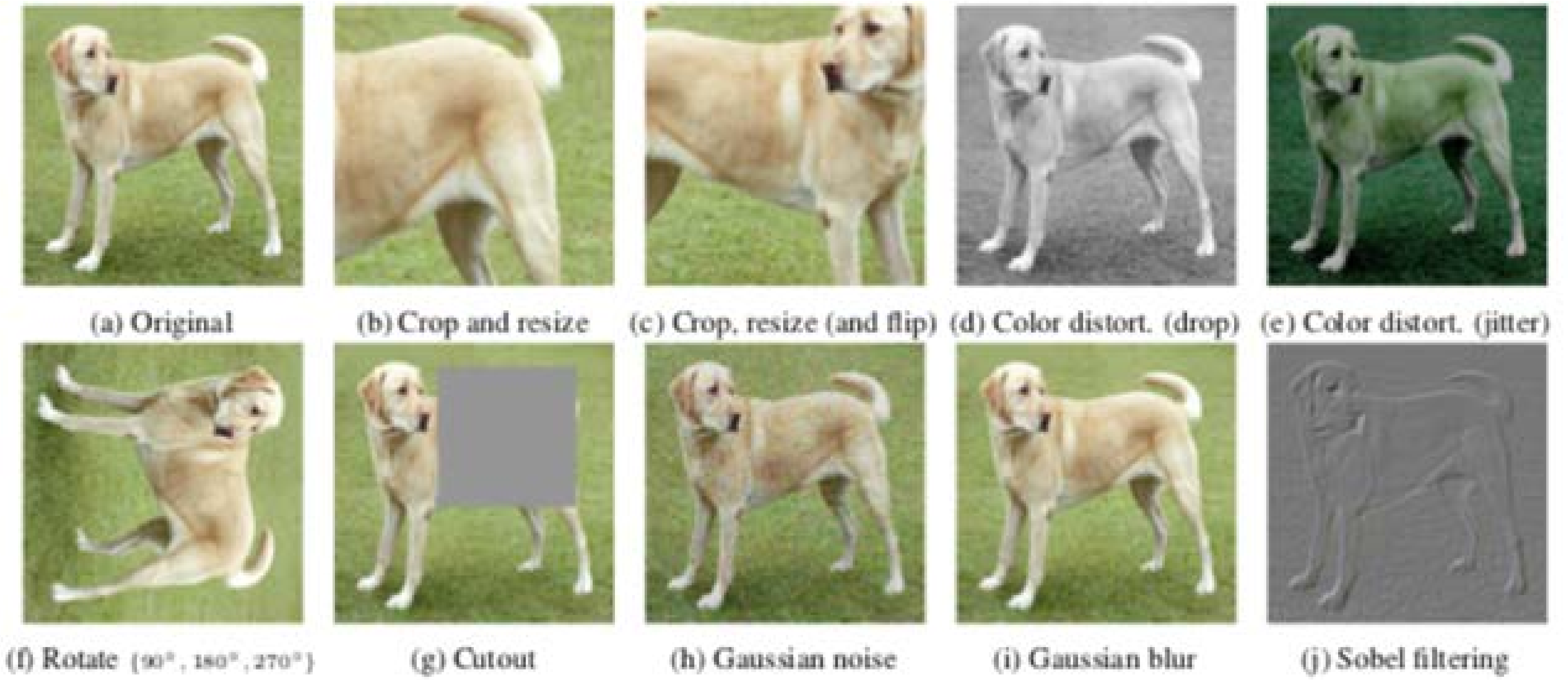


- What are the similar (positive) and dissimilar (negative) pairs?

What are positive and negative examples?

Negative examples: randomly uniformly drawn from data

Positive examples: perturbations that keep semantic meaning, data augmentation

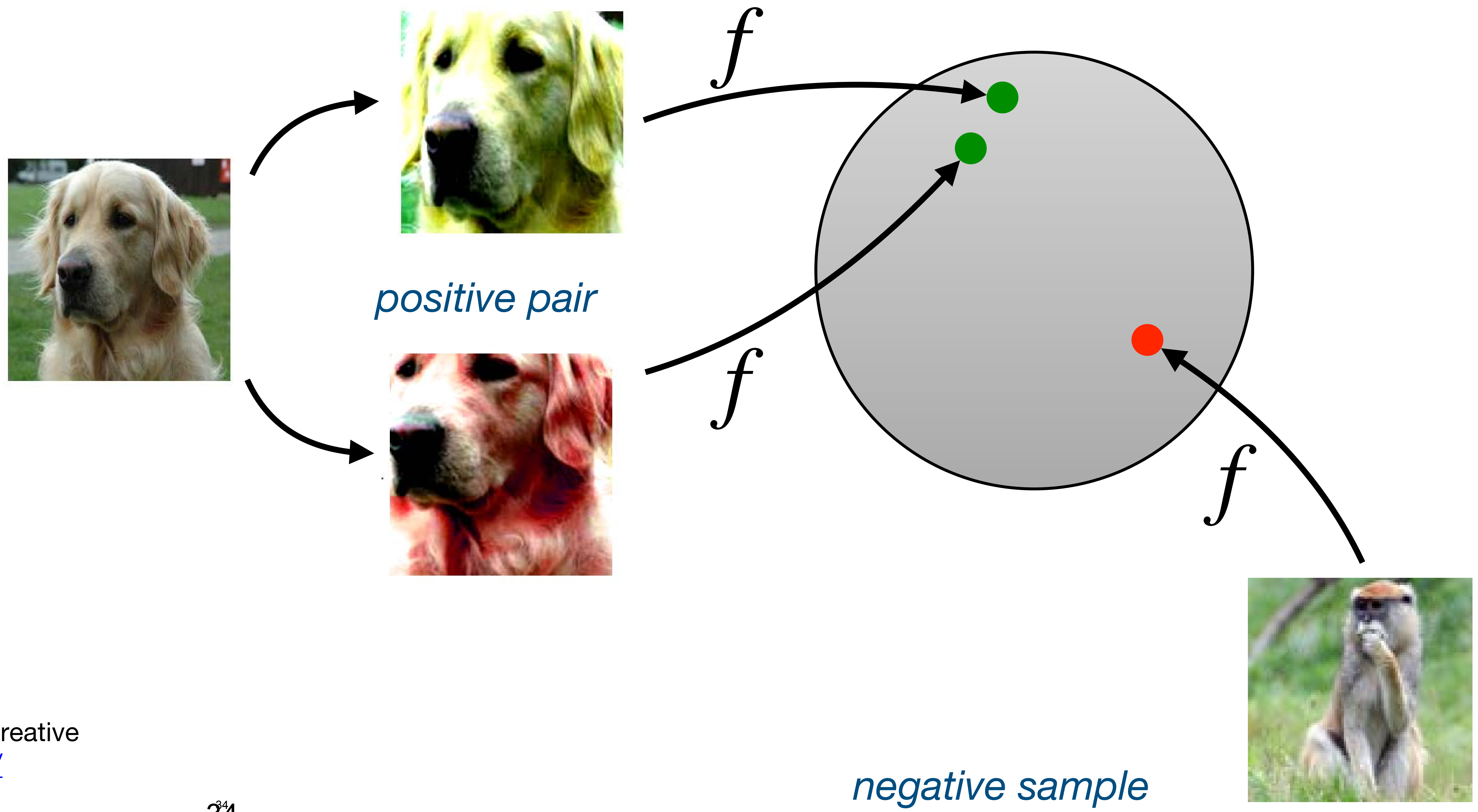


(Chen, Kornblith, Norouzi, Hinton 2020)

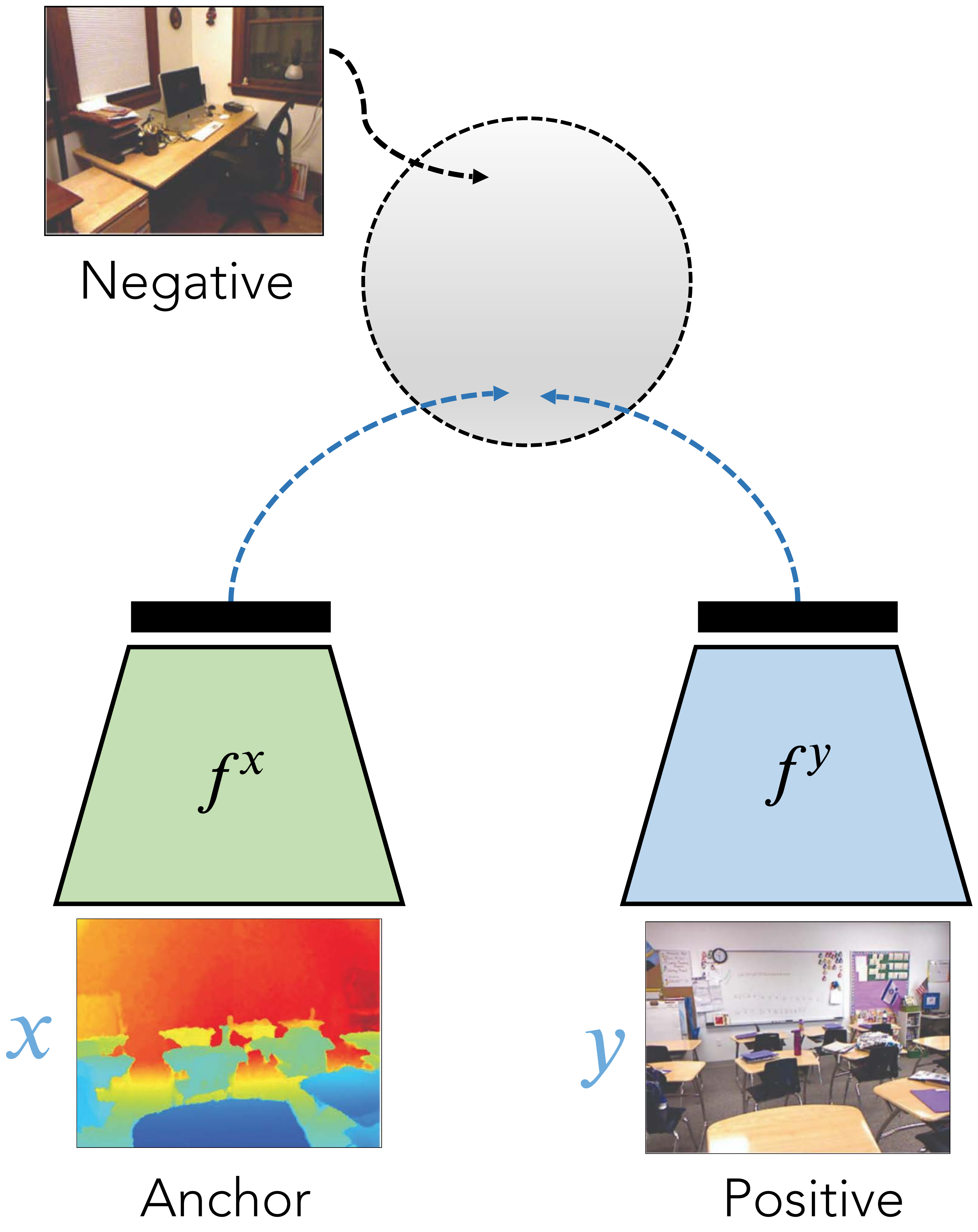
Positive and negative samples

e.g. SimCLR:

- for each data point in the batch, generate 2 random augmentations as positive pair
- all other 2(B-1) augmented samples in the batch (of size B) are used as negatives



Variations



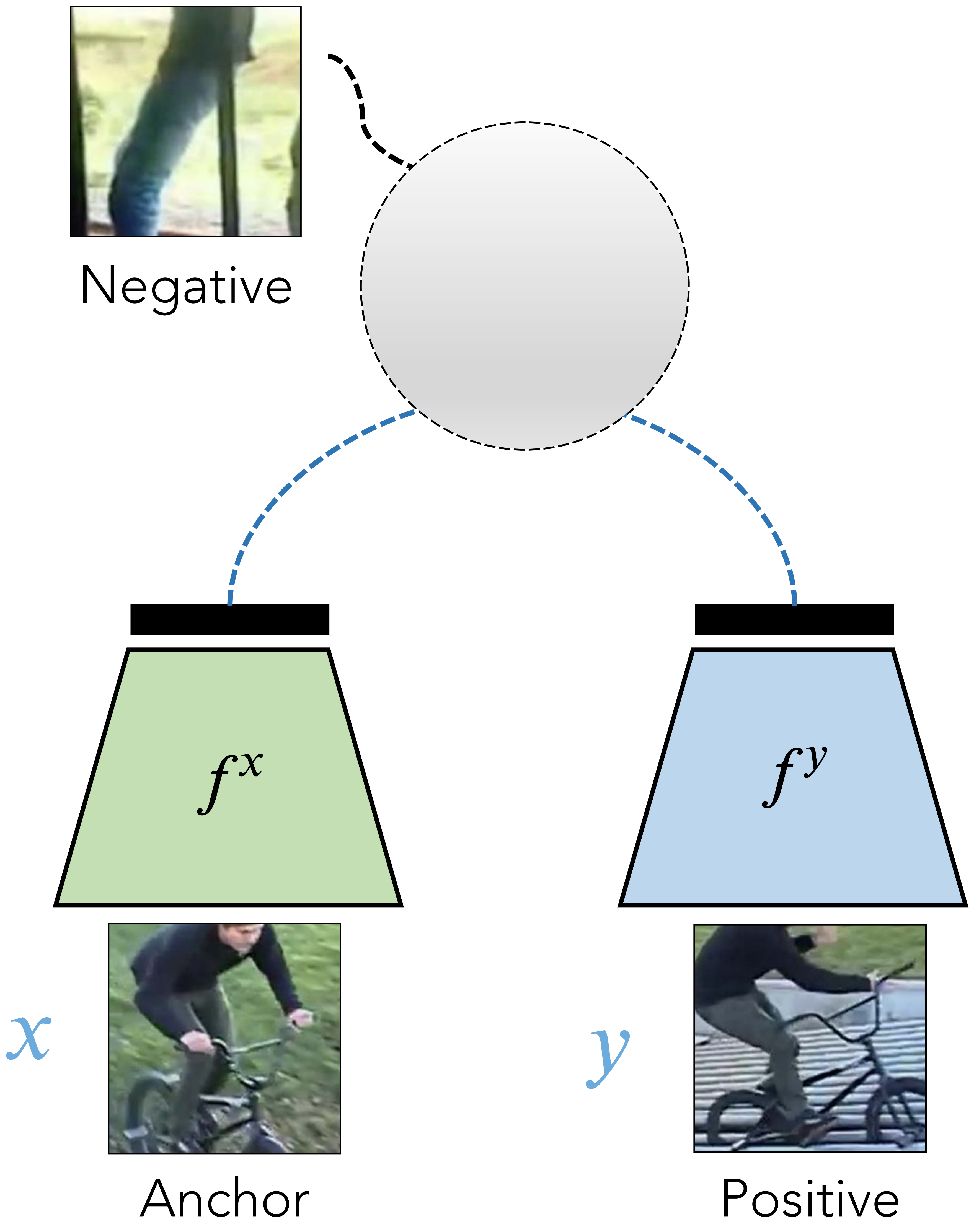
(x, y) are two "views" of the same scene

Cross-Channel Representation Learning

[CMC, Tian, Krishnan, Isola 2020]

⋮

Variations



(x, y) are two "views" of the same scene

Video Representation Learning

["Slow Feature Learning", Wiskott & Sejnowski 2002]

[Mobahi, Collobert, Weston 2009]

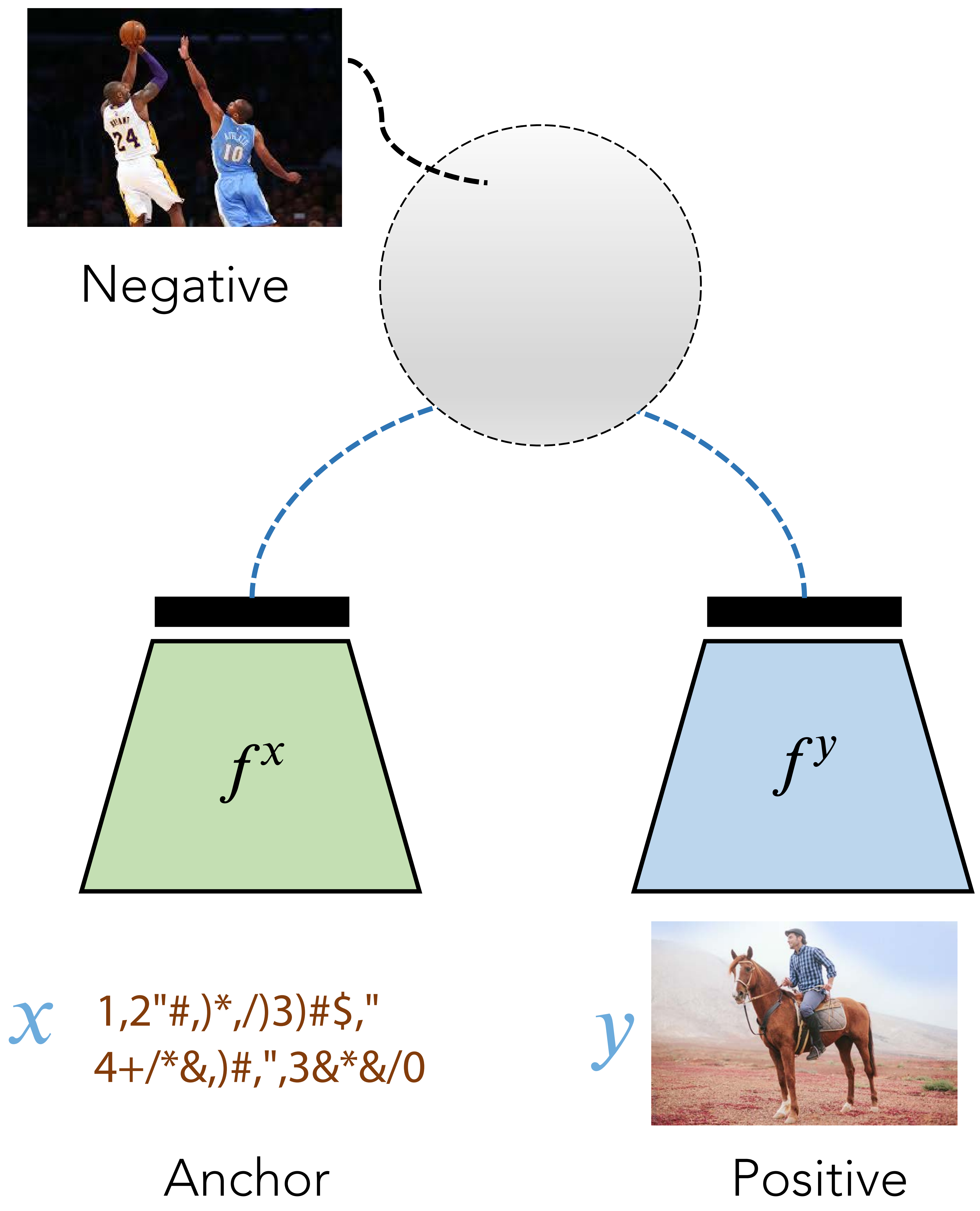
[Wang & Gupta 2015]

[Isola, Zoran, Krishnan, Adelson 2016]

[Sermanet, Lynch, Chebotar et al. 2018]

[van den Oord, Li, Vinyals 2018]

Variations



(x, y)!"#$!%&'!()*$&+,!'-!%.$!+"/$!+0$1$

!"#$%"$&'()*)+# ,-&./&*&#0"0)+#,!&"/#)#$

[Karpathy, Joulin, Fei-Fei 2014]

⋮

[CLIP, Radford, Kim et al. 2021]

What is this method doing?

2 ingredients:

- Contrastive loss (which specific form)
- Data (which positive/negative pairs)

What is the contrastive loss doing?

$$\mathcal{L}_{cont}(f) = \mathbb{E}_{(\mathbf{x},\mathbf{x}^+)\sim p_{pos}, \{\mathbf{x}_i^-\}_{i=1}^N \sim p_{data}} \left[-\log \frac{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau}}{e^{f(\mathbf{x})^\top f(\mathbf{x}^+)/\tau} + \sum_{i=1}^N e^{f(\mathbf{x})^\top f(\mathbf{x}_i^-)/\tau}} \right]$$

- cross-entropy loss to distinguish data points
- maximizes a lower bound on mutual information between "views" $f(\mathbf{x}), f(\mathbf{x}^+)$ *(Poole et al, 2019)*:

$$\mathrm{MI}(f(\mathbf{x}), f(\mathbf{x}^+)) \geq \log(N) - \mathcal{L}(f)$$

What (else) is the contrastive loss doing?

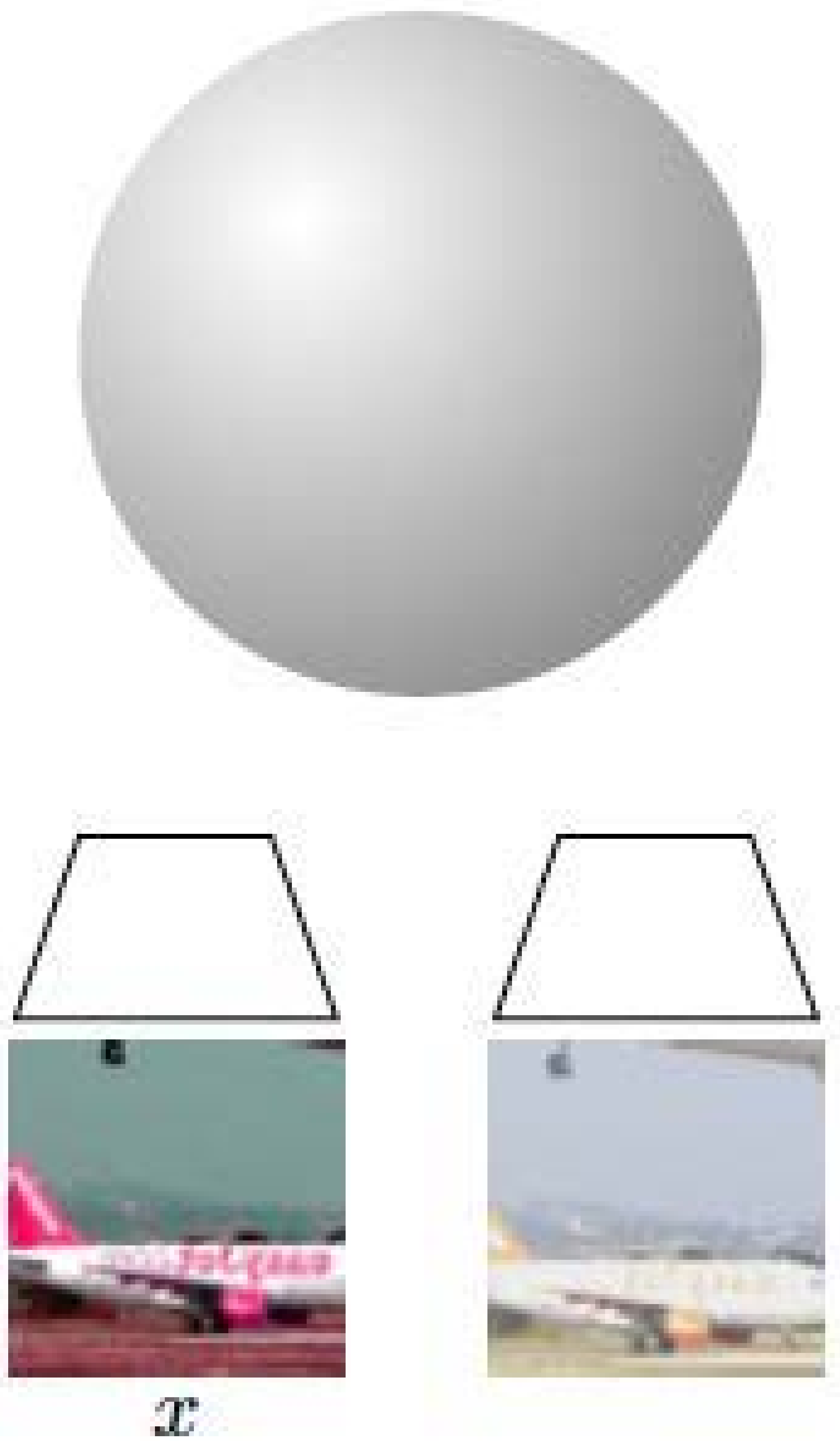
- Recall: properties of "good" representations:

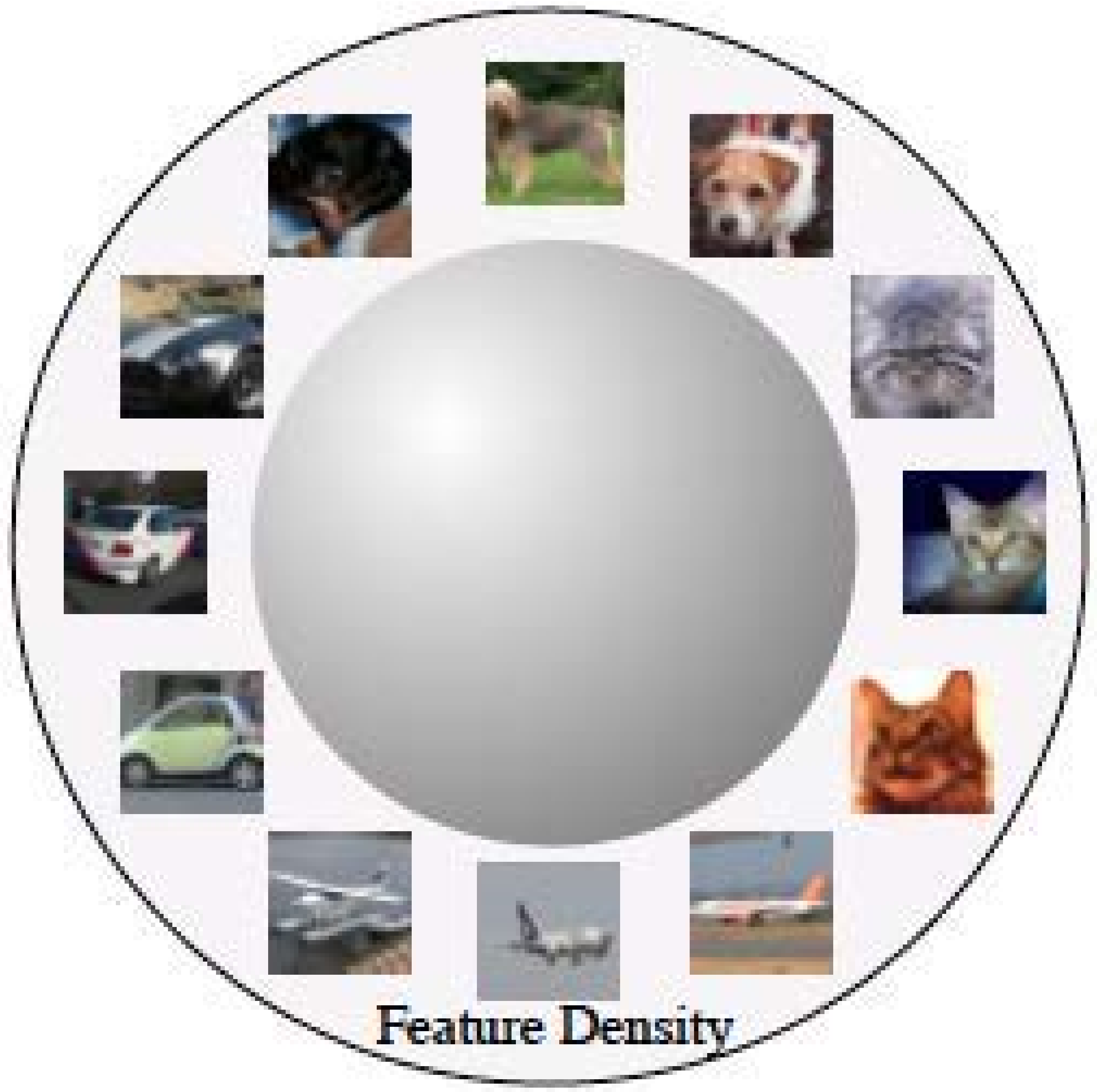
1. Concentration/Alignment: Data from the same class is close together, remove irrelevant information

2. Separation: classes are well separated, do not lose information

3. Robustness to irrelevant perturbations

Alignment and separation



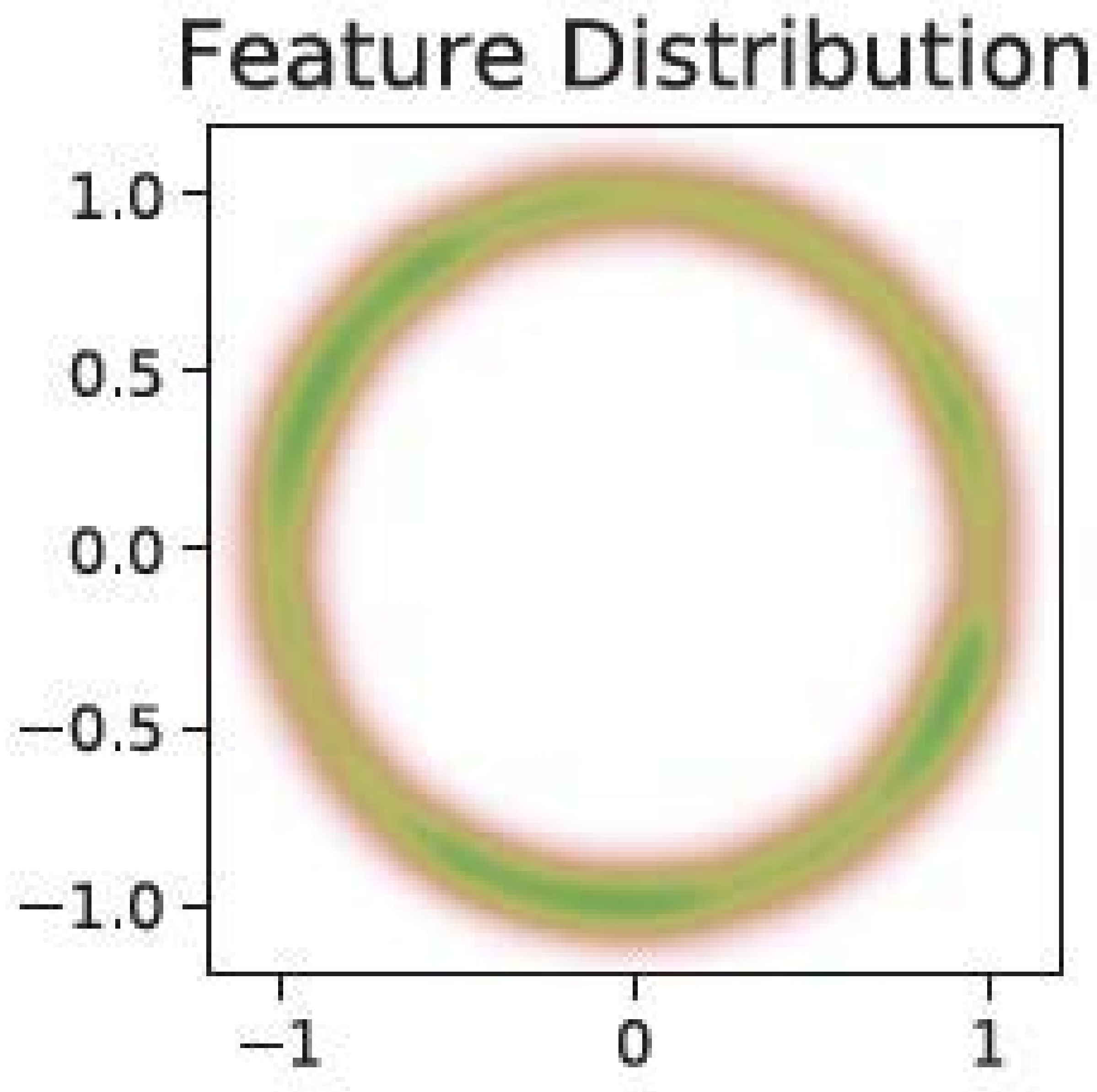


Uniformity: Preserve maximal information

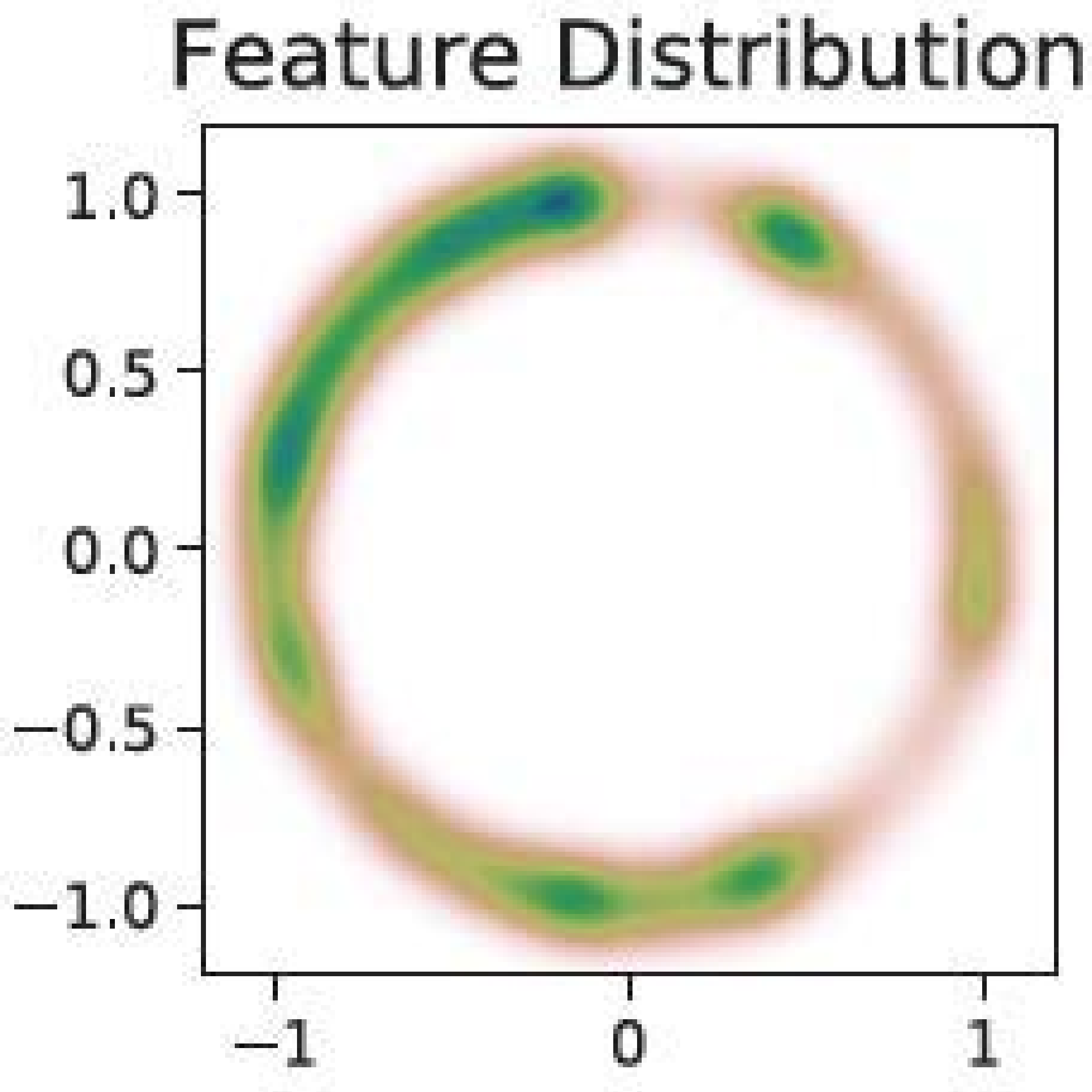
Feature distribution from Contrastive Learning

Toy example:

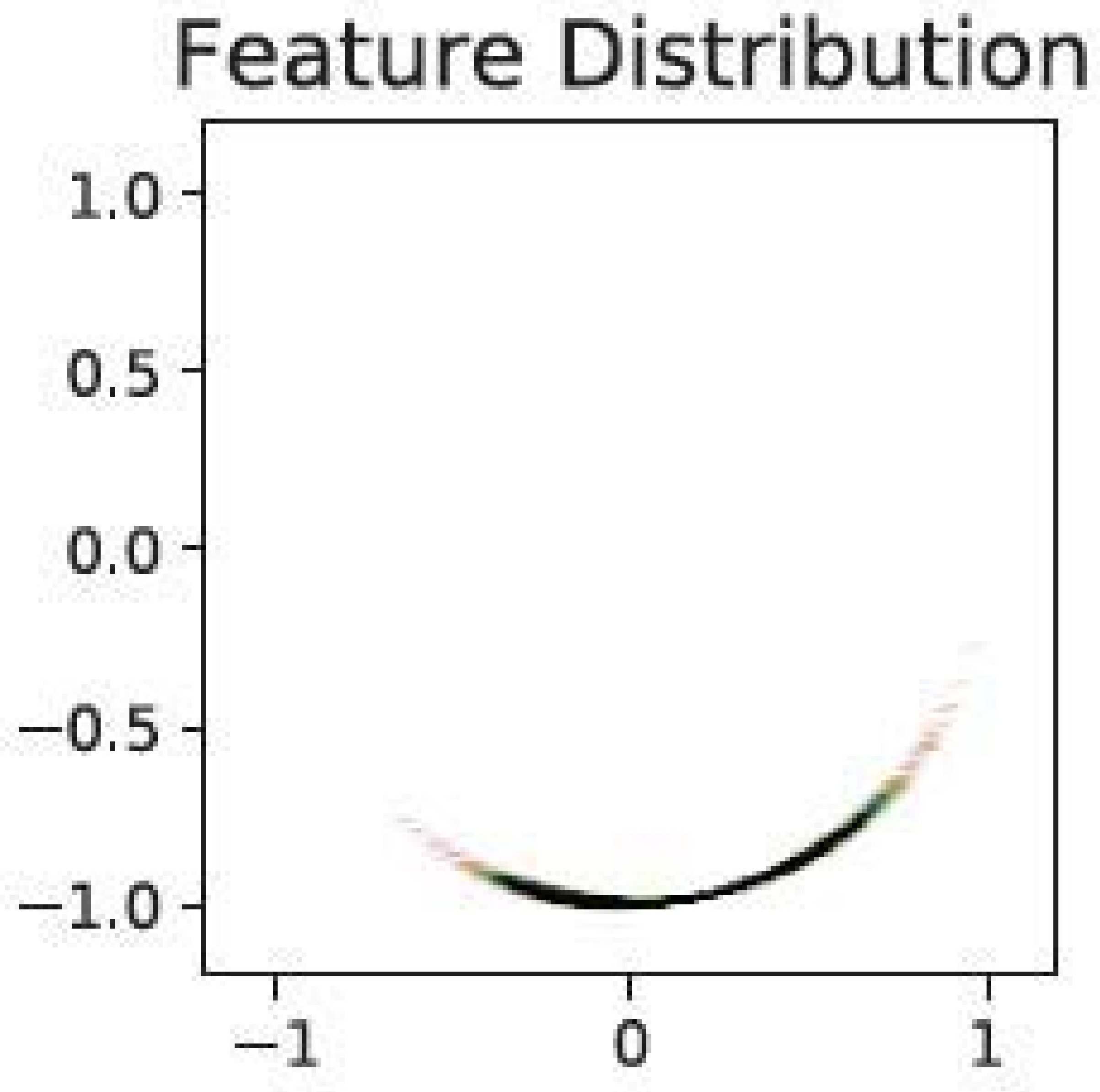
Train `CIFAR-10` encoders with $\mathcal{S}^1$ feature space (circle).
Visualize feature distributions on the validation set.



Unsupervised Contrastive Learning

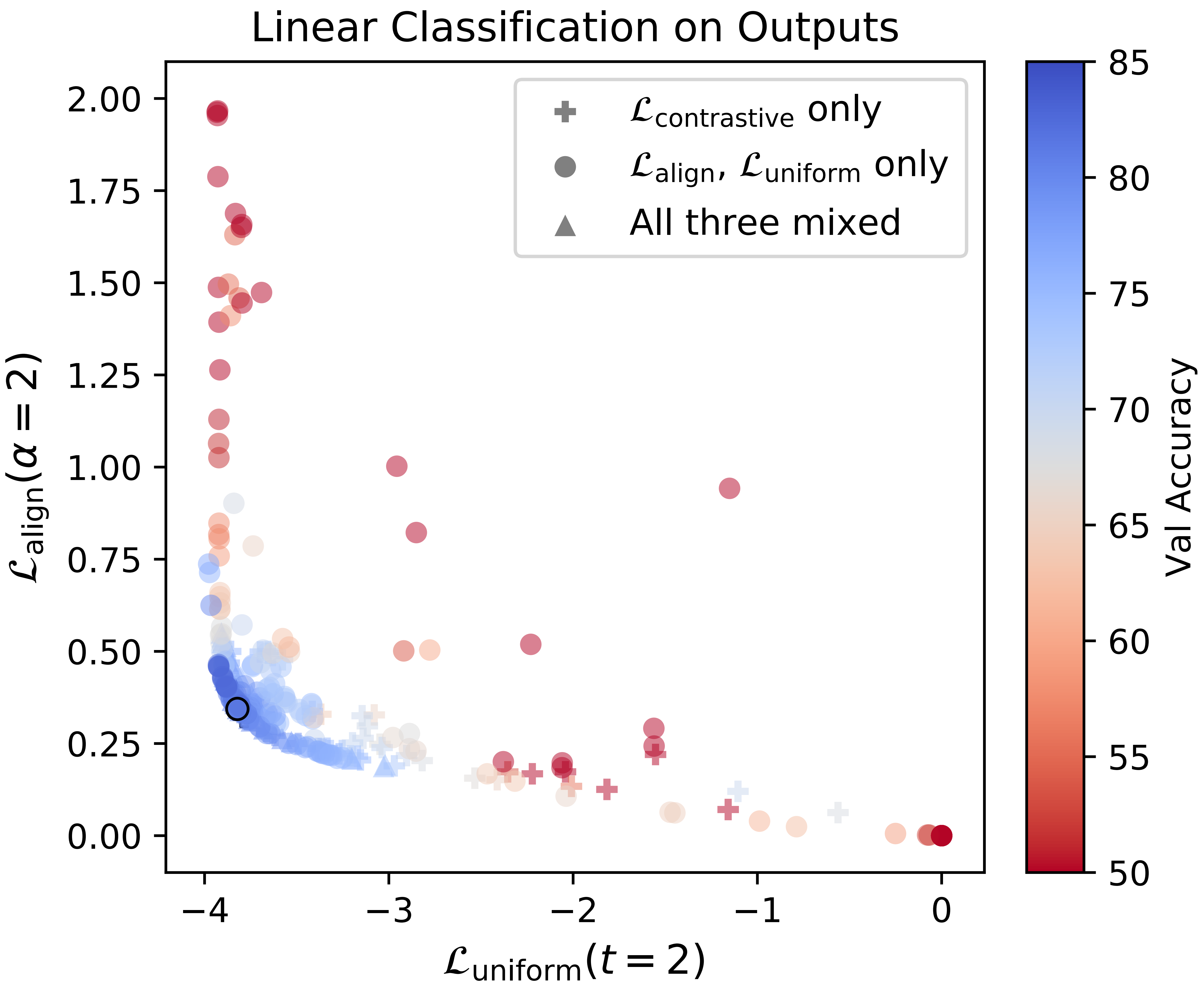


Supervised Predictive (NLL) Learning

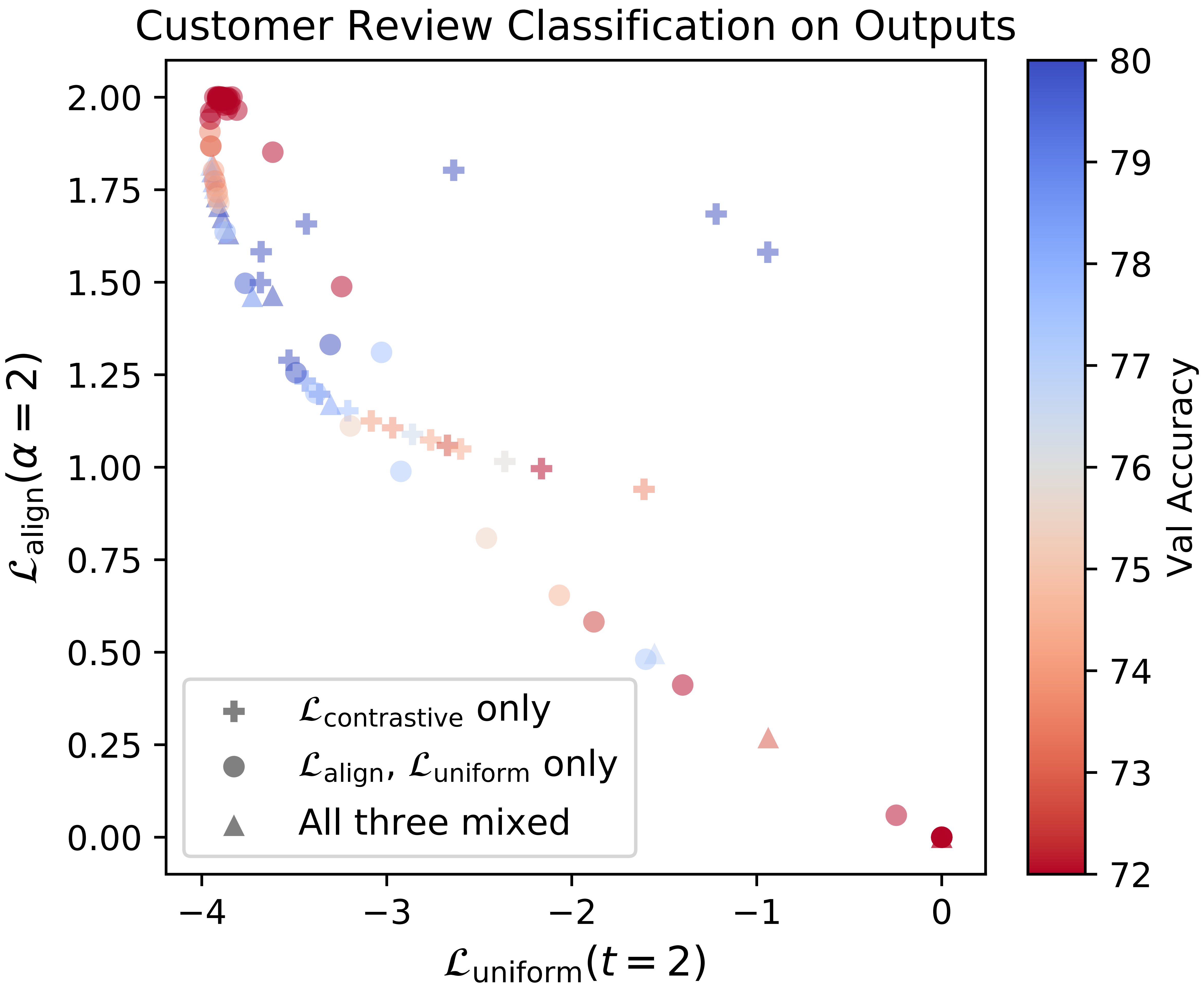


Random Network Initialization

Relation Between Representation Quality and Alignment & Uniformity



306 STL-10 Encoders



108 BookCorpus Encoders

figures: Wang & Isola, 2020

What is the contrastive loss doing?

- Loss function encourages:

1. Concentration/Alignment: Data from the same class is close together, remove irrelevant information

2. Separation: classes are well separated, do not lose information

- What do the selection of positive and negative pairs encourage?

What are we “teaching” the model via choice of pairs?

- positive pairs = augmentations of the same data point should be close
- => learned representation is invariant to perturbations induced by data augmentations: learned invariance
- Finding the “right” invariances can be challenging for different types of data
- Learned versus hard-coded invariances (geometric DL lecture): when would we use which?

What is the contrastive loss doing?

- Loss function encourages:

1. Concentration/Alignment: Data from the same class is close together, remove irrelevant information

2. Separation: classes are well separated, do not lose information

- Data encourages:

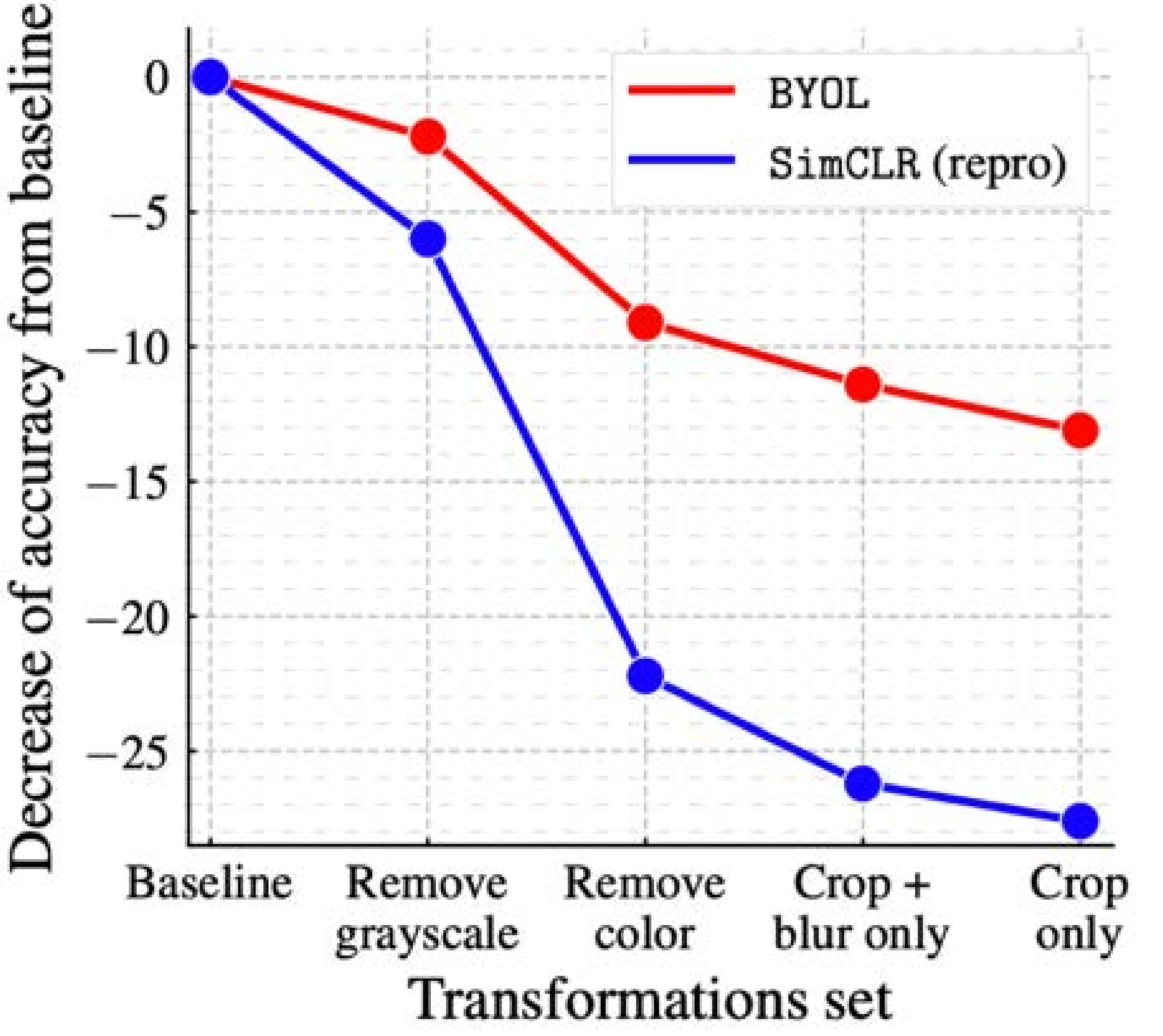
3. Robustness to **irrelevant** perturbations

Ingredients to make self-supervised CL work (better)

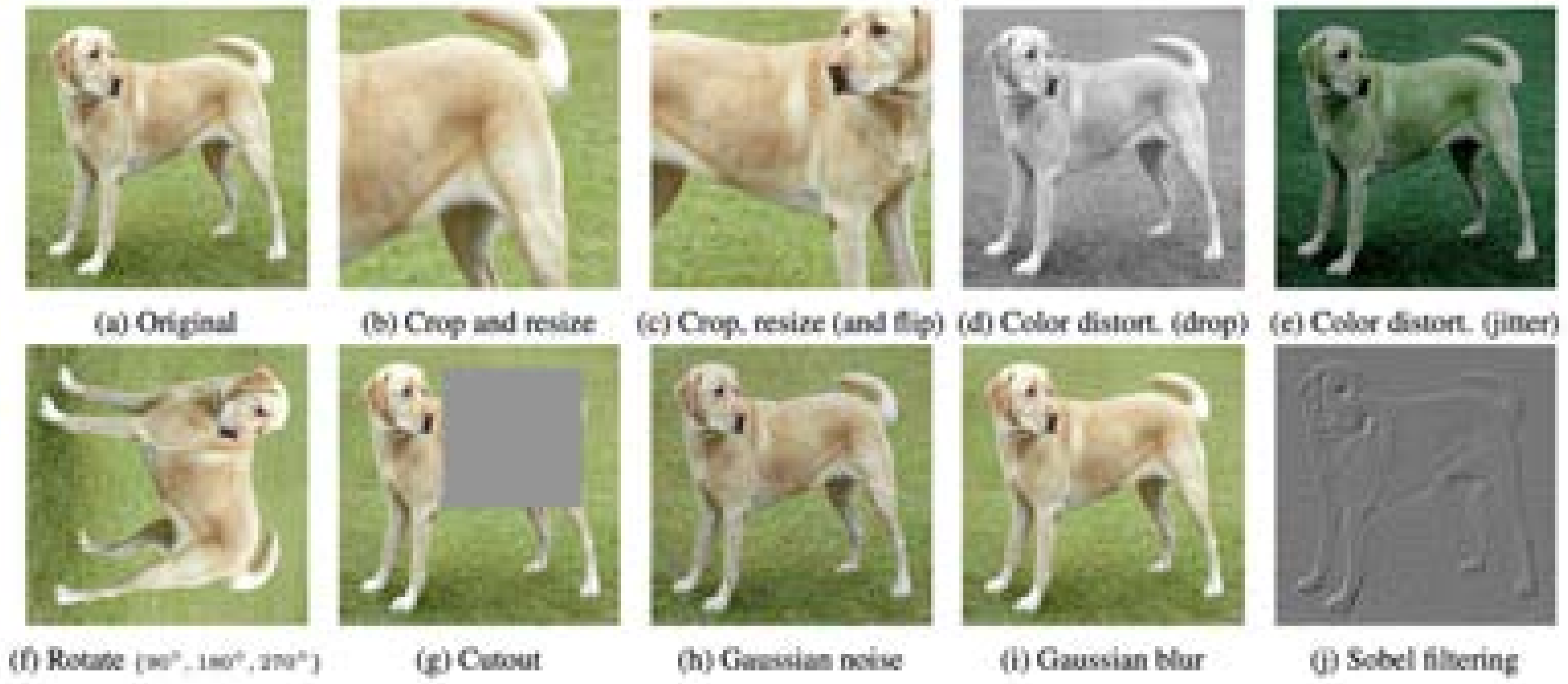
- heavy data augmentation
- projection heads
- large batch size (many negative examples)
- choice of data pairs / hard negative examples

} SimCLR model

Effect of data augmentation



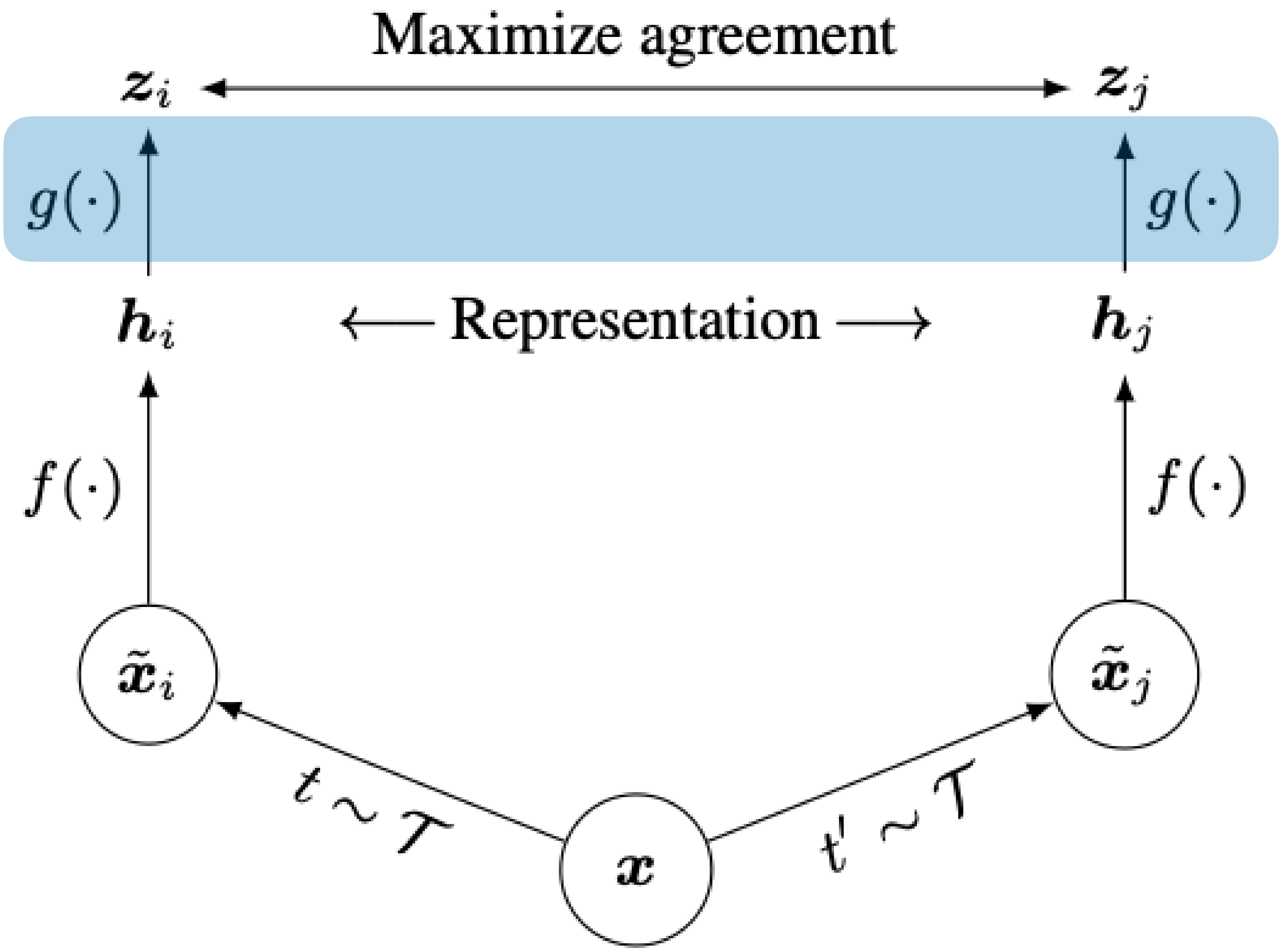
Impact of progressively removing transformations



(figure: Grill et al 2020)

Projection head

- contrastive loss is applied to a transformed version $g(\mathbf{h})$ of the representation $\mathbf{h}$
- g is linear or small MLP
- use $\mathbf{h}$ for downstream task

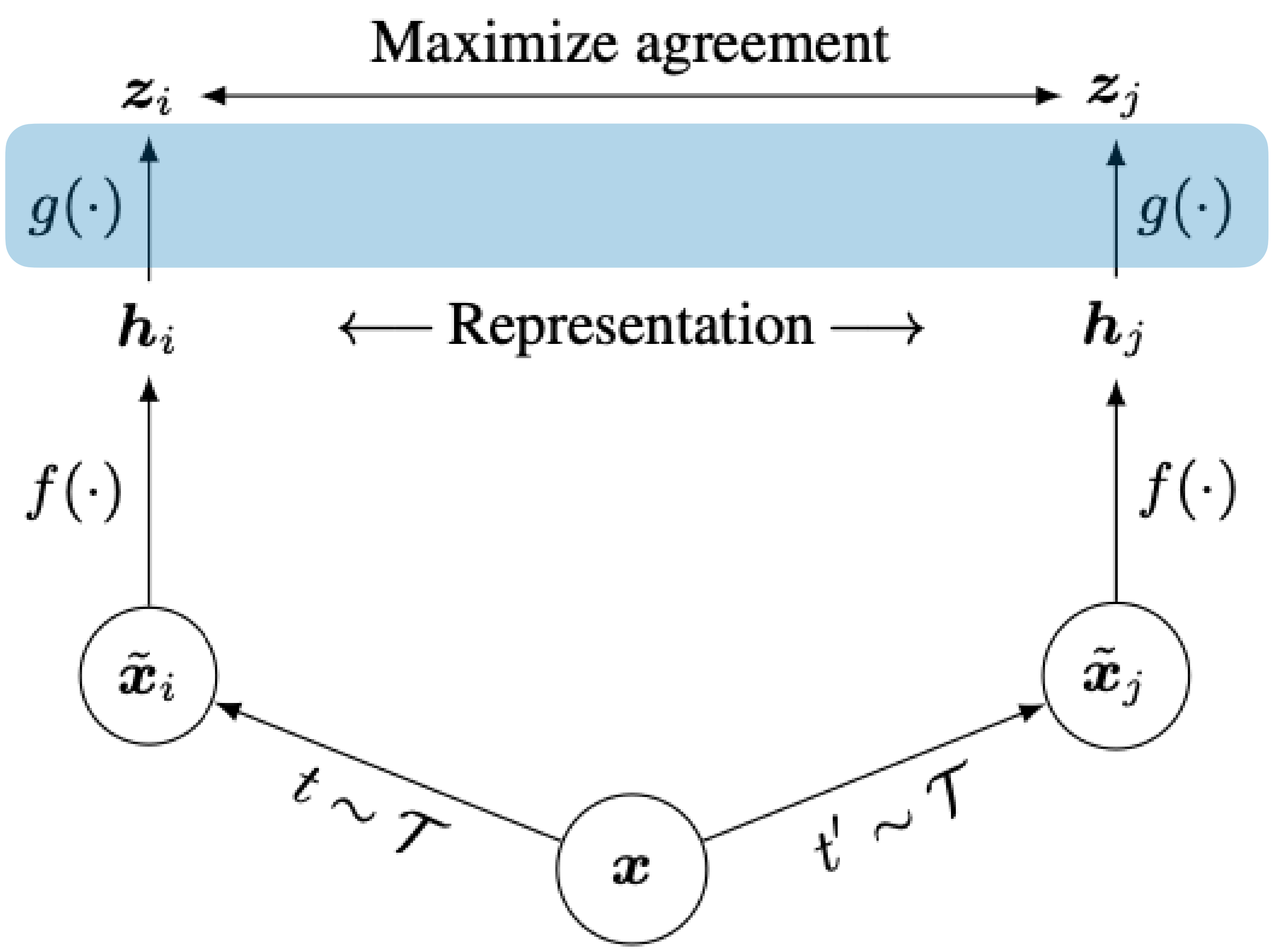


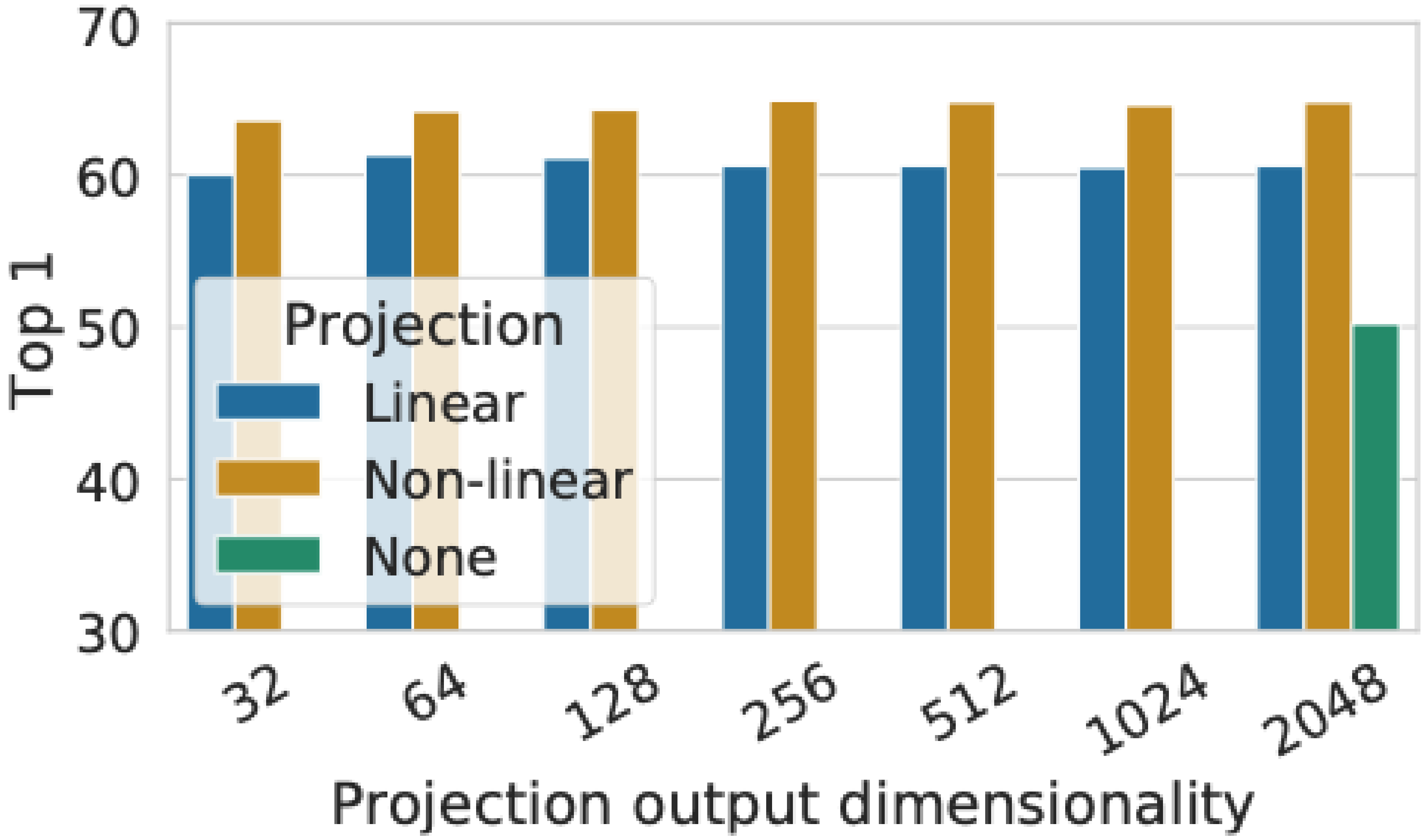
- Projection head improves performance!

Projection head

- Projection head improves performance.

- Why?
  Possibly because representation **h** then need not be completely invariant to augmentations, can retain some information





Effect of batch size

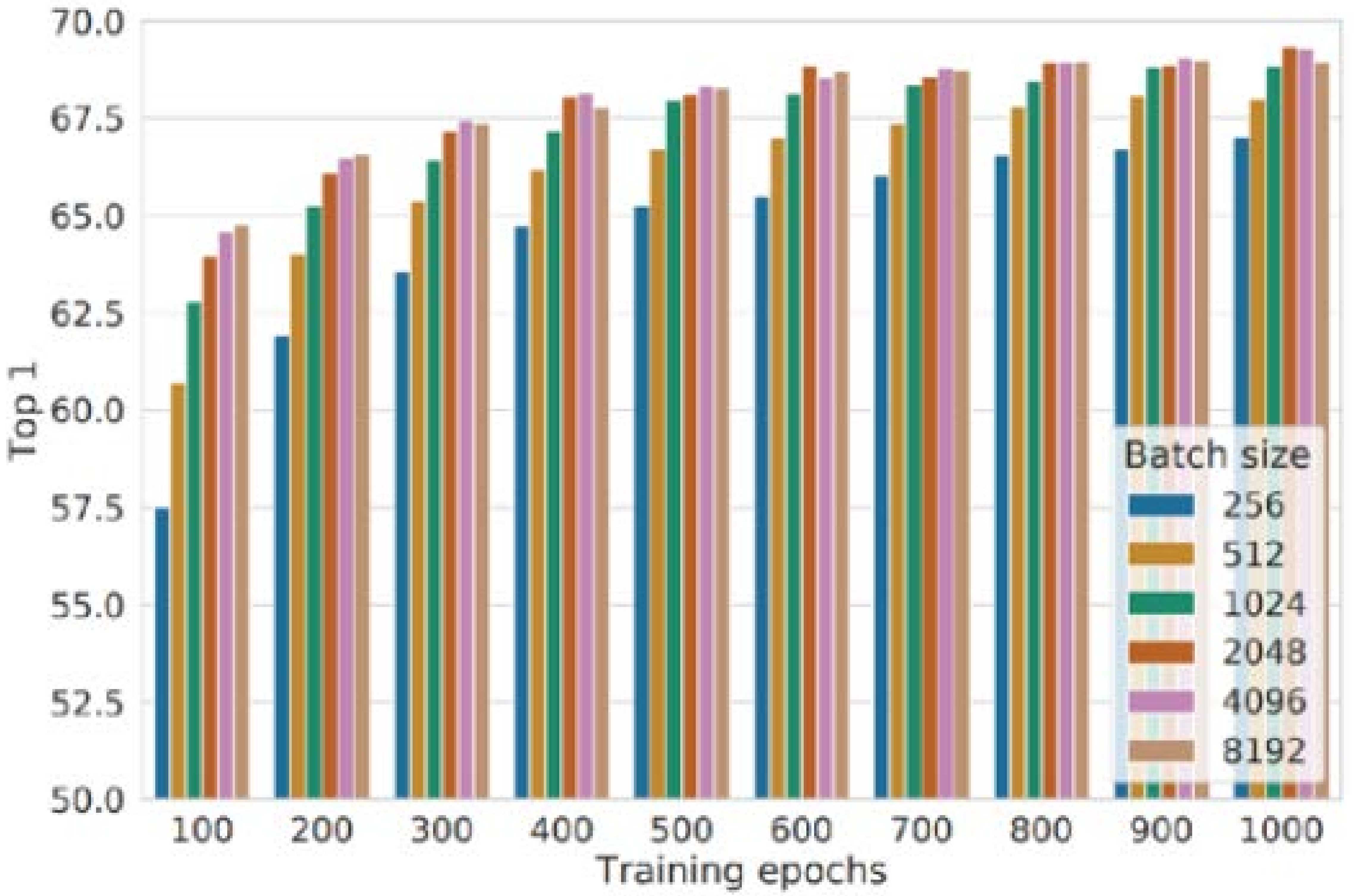


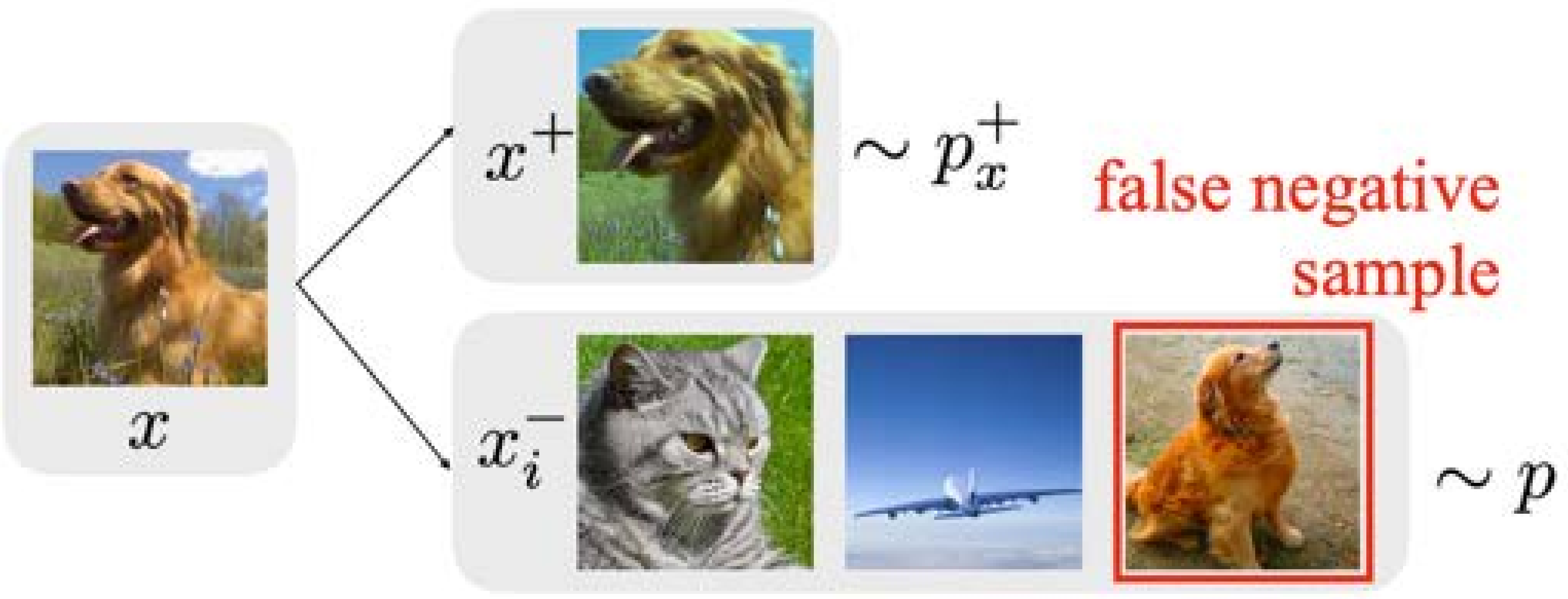
Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

(Figure from Chen et al. 2020)

- SimCLR uses all points in a batch as negative examples for a positive pair
- needs large number of negative pairs = large batch sizes
- Expensive. Newer methods make this more efficient (like MoCo, *He et al. 2020*)

Improving negative samples

- We are pushing apart negative pairs. Negative pairs are random pairs from the data.



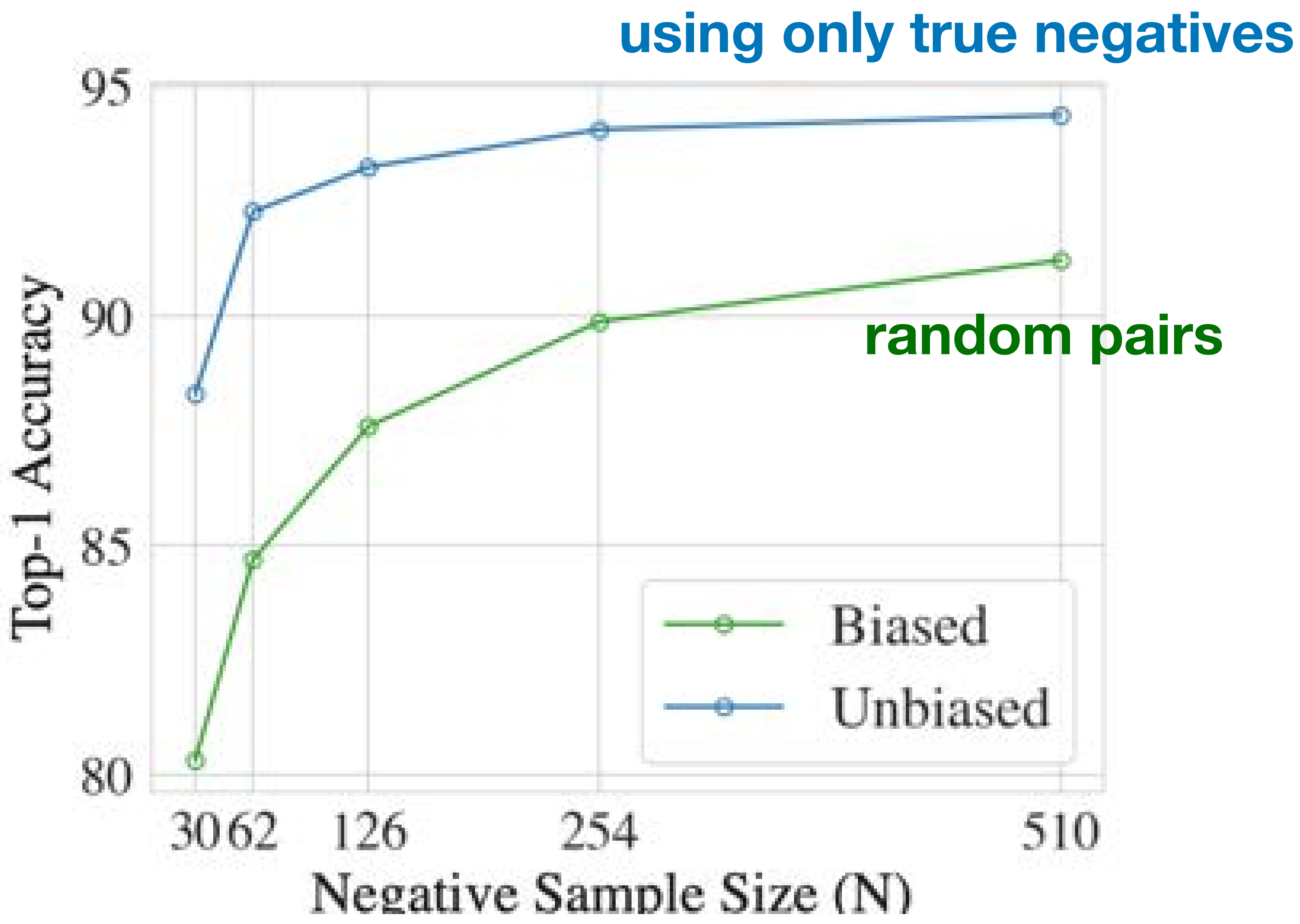
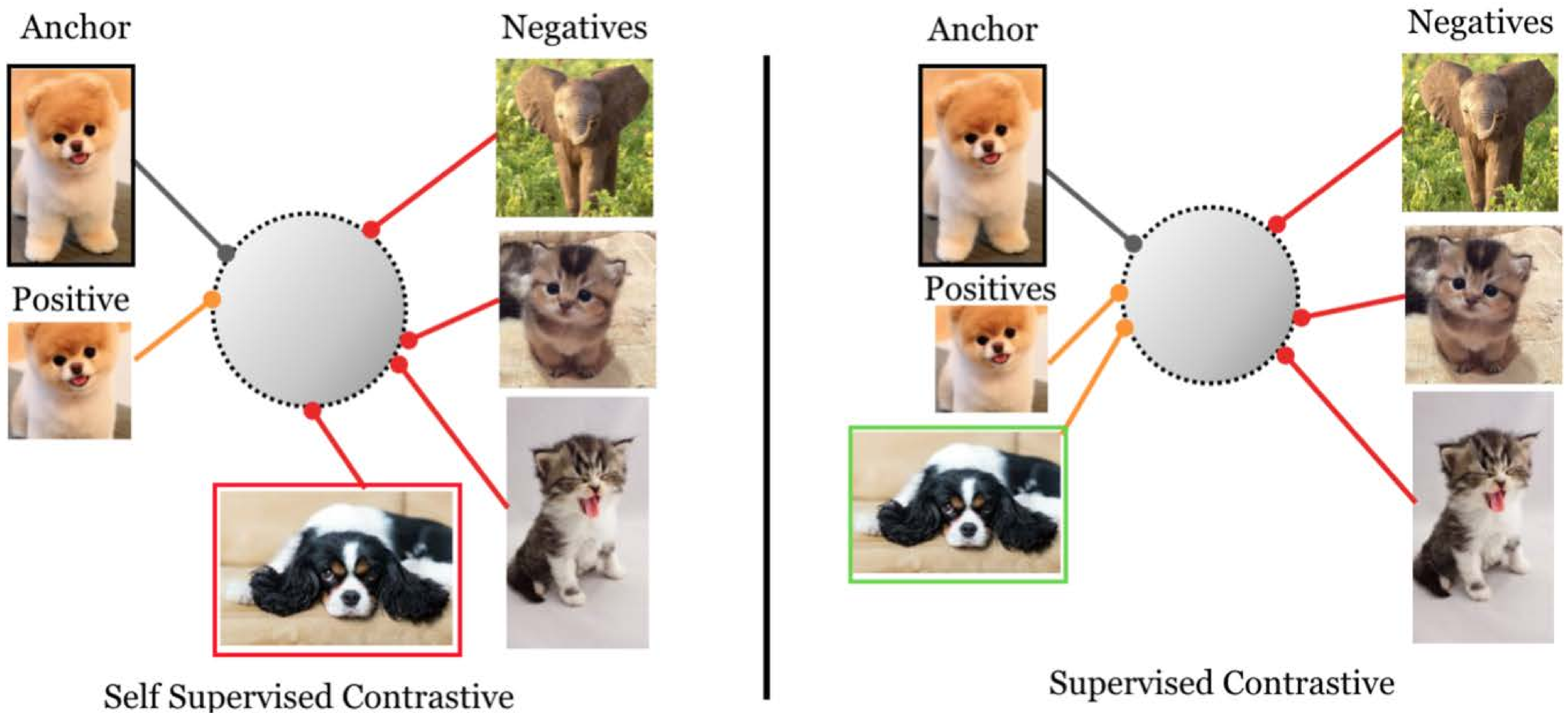


figure: Chuang et al, Debiased contrastive learning

Supervised or semi-supervised contrastive learning

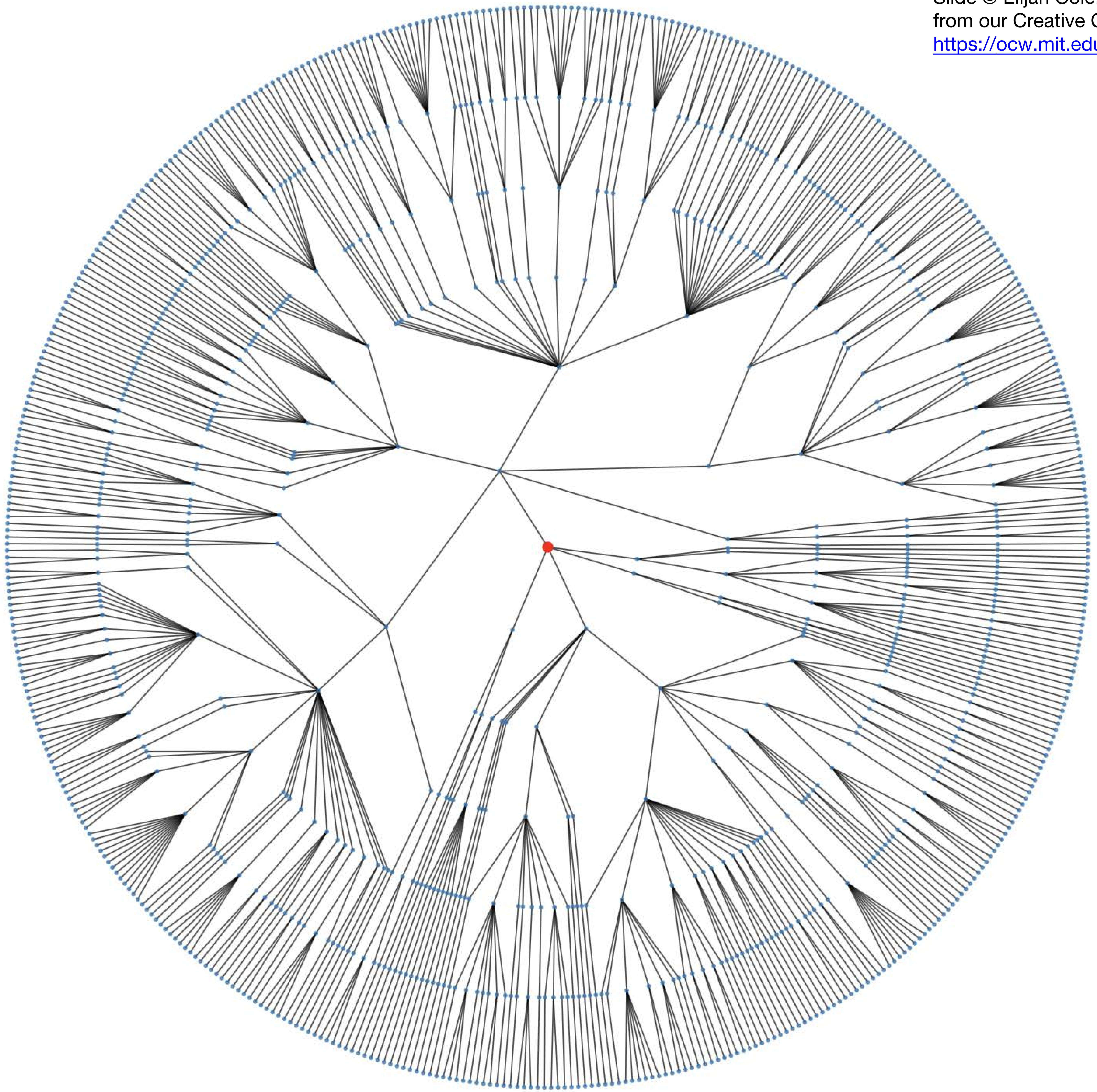
- Contrastive learning provides more geometric and robustness feedback than cross-entropy loss
- Idea: in addition to data augmentation, use images from same class as positive pairs (multiple positive pairs)



Khosla et al 2020

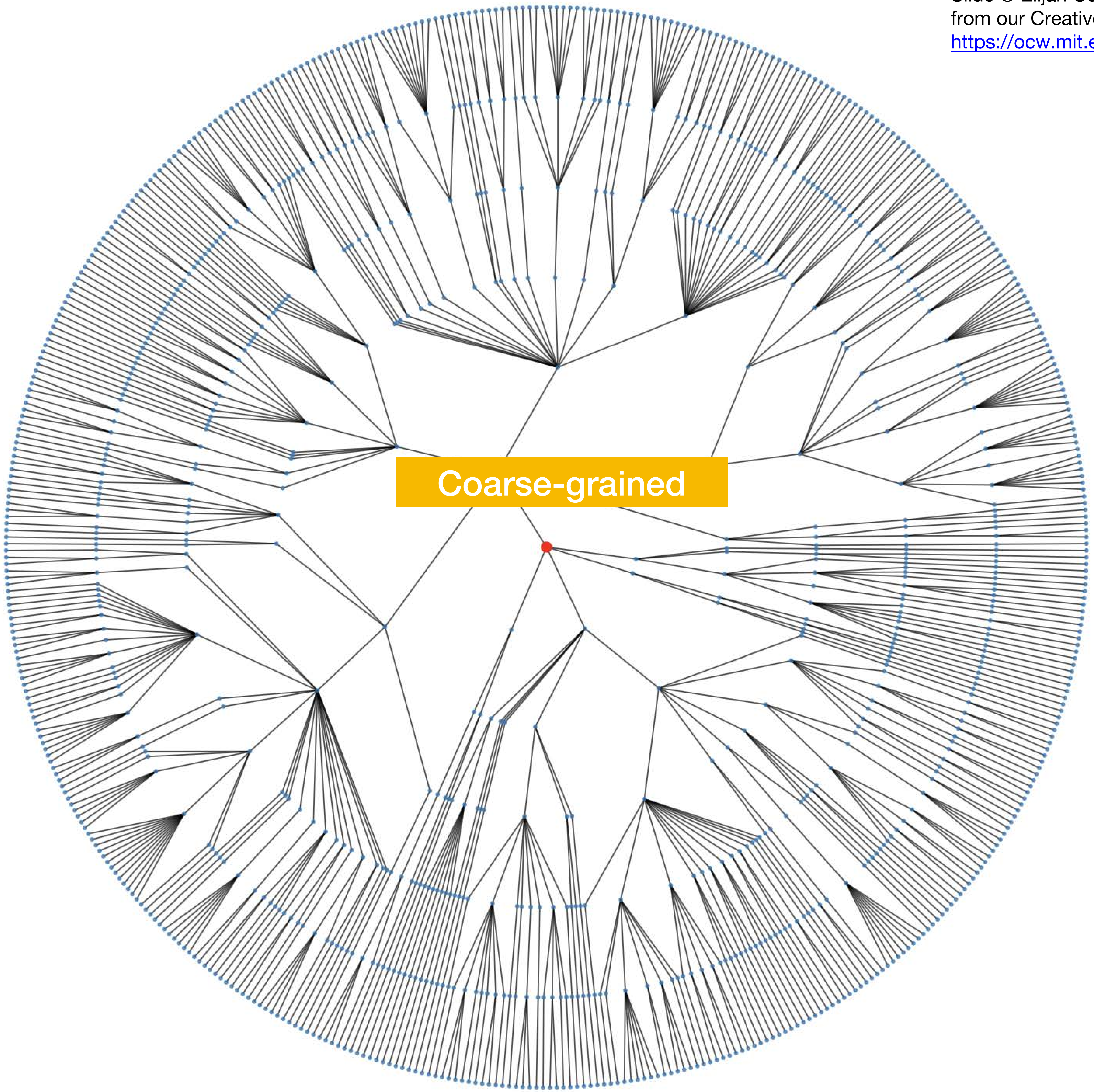
Case study: iNaturalist 2021

- 10,000 Species
- 2.7M Training Images
- 50k Validation Images
- 500k Test Images

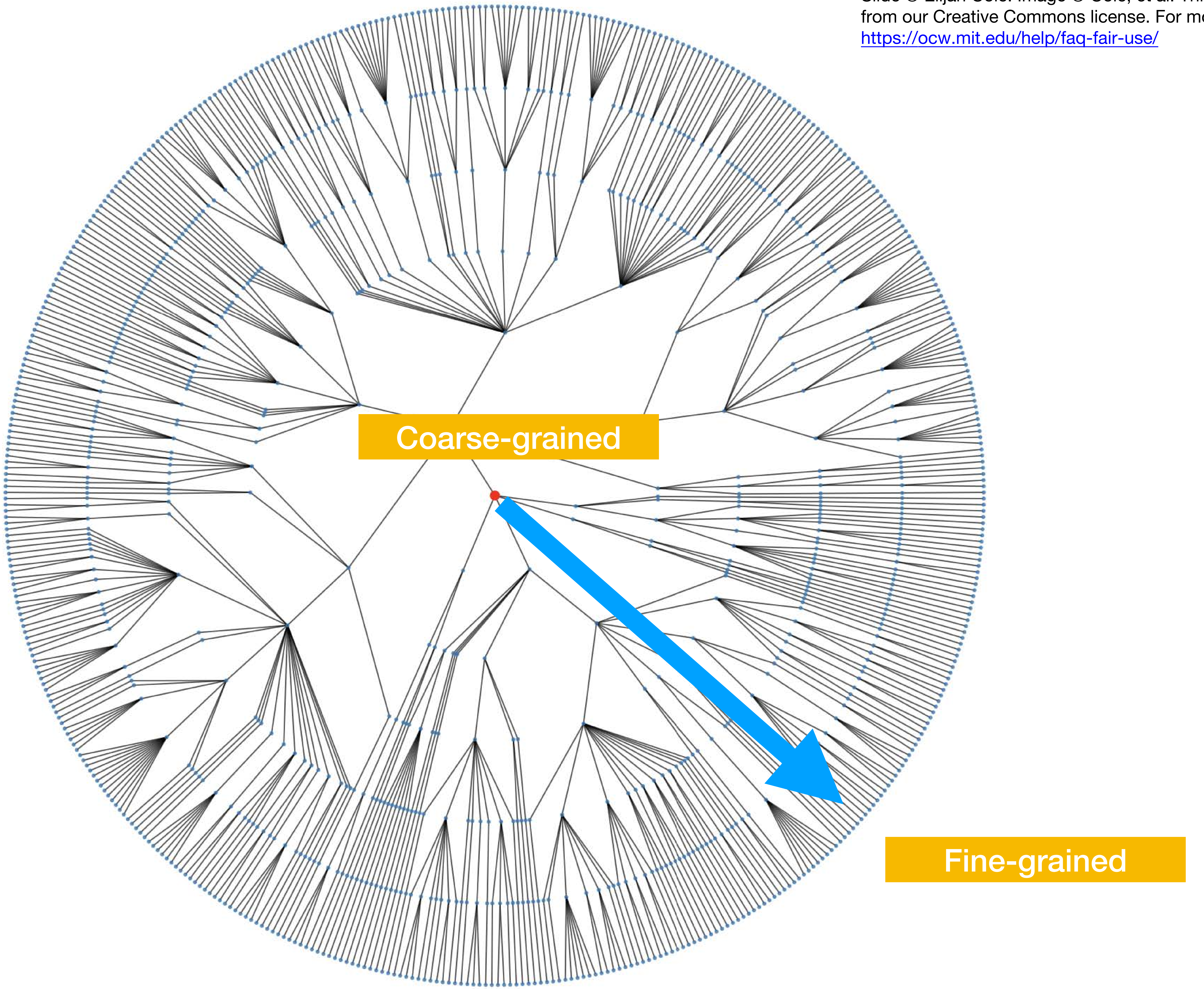


Cole et al., *When Does Contrastive Visual Representation Learning Work?*, CVPR 2022

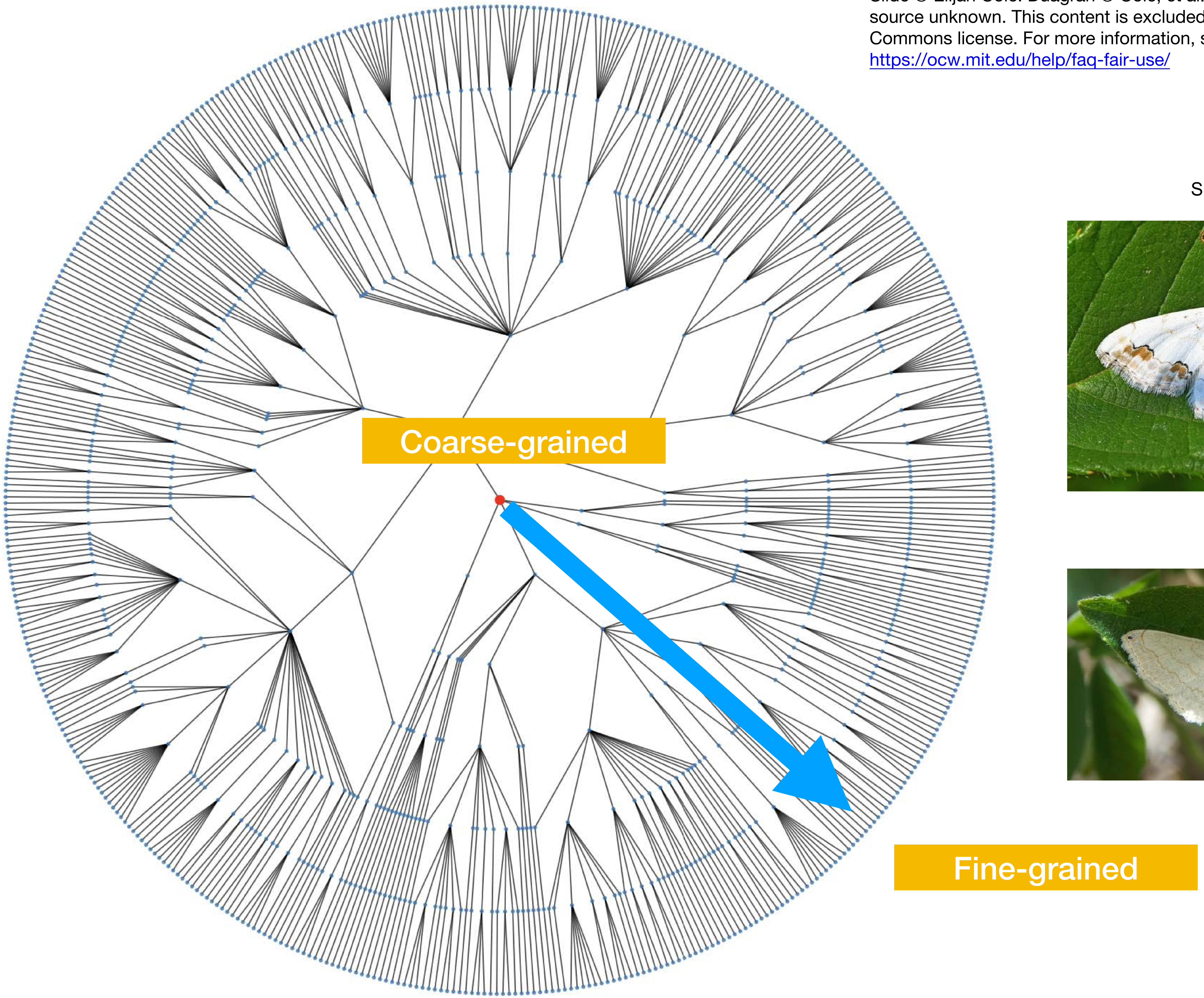
Slide: Elijah Cole



Slide: Elijah Cole



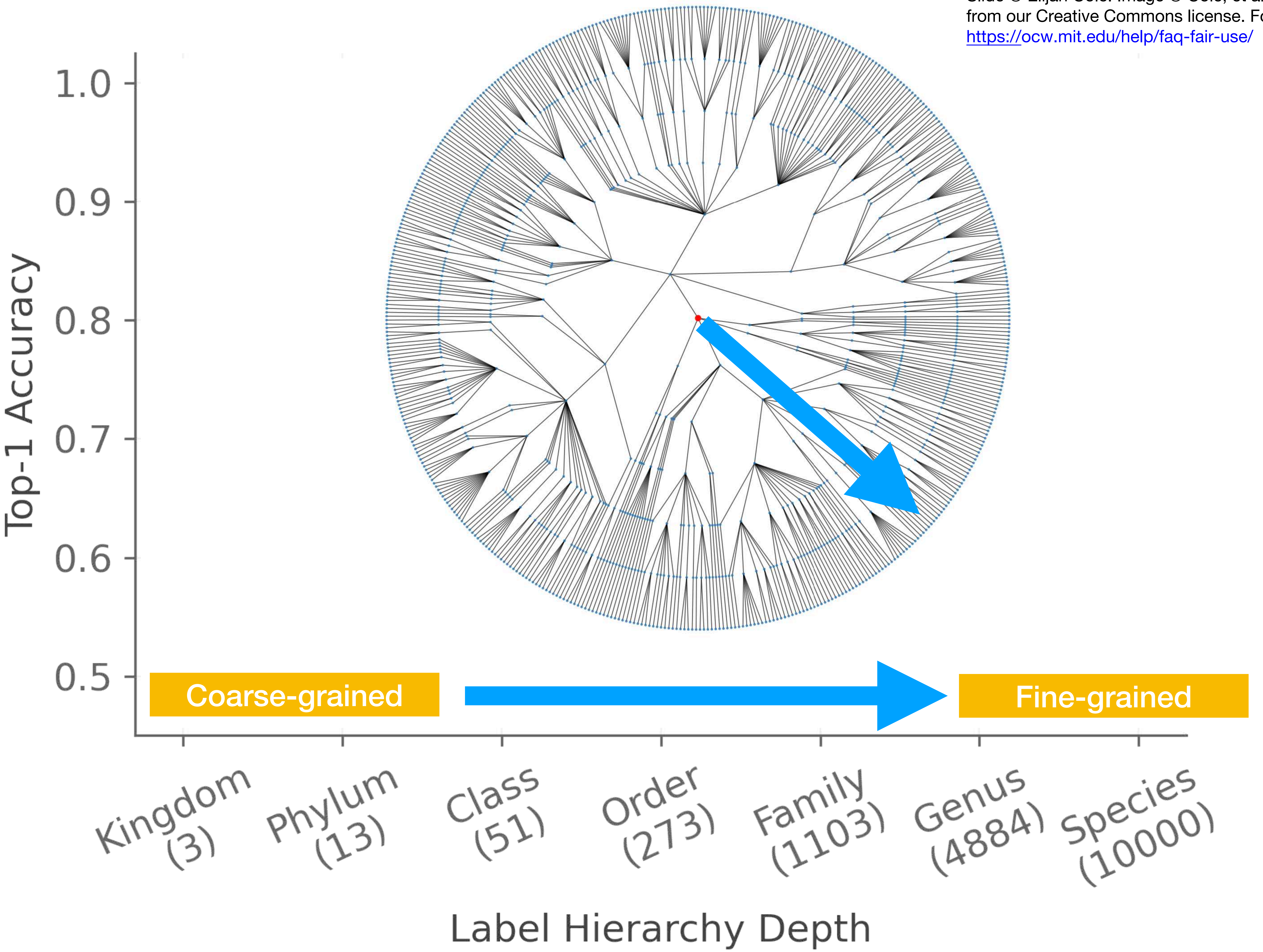
Slide: Elijah Cole



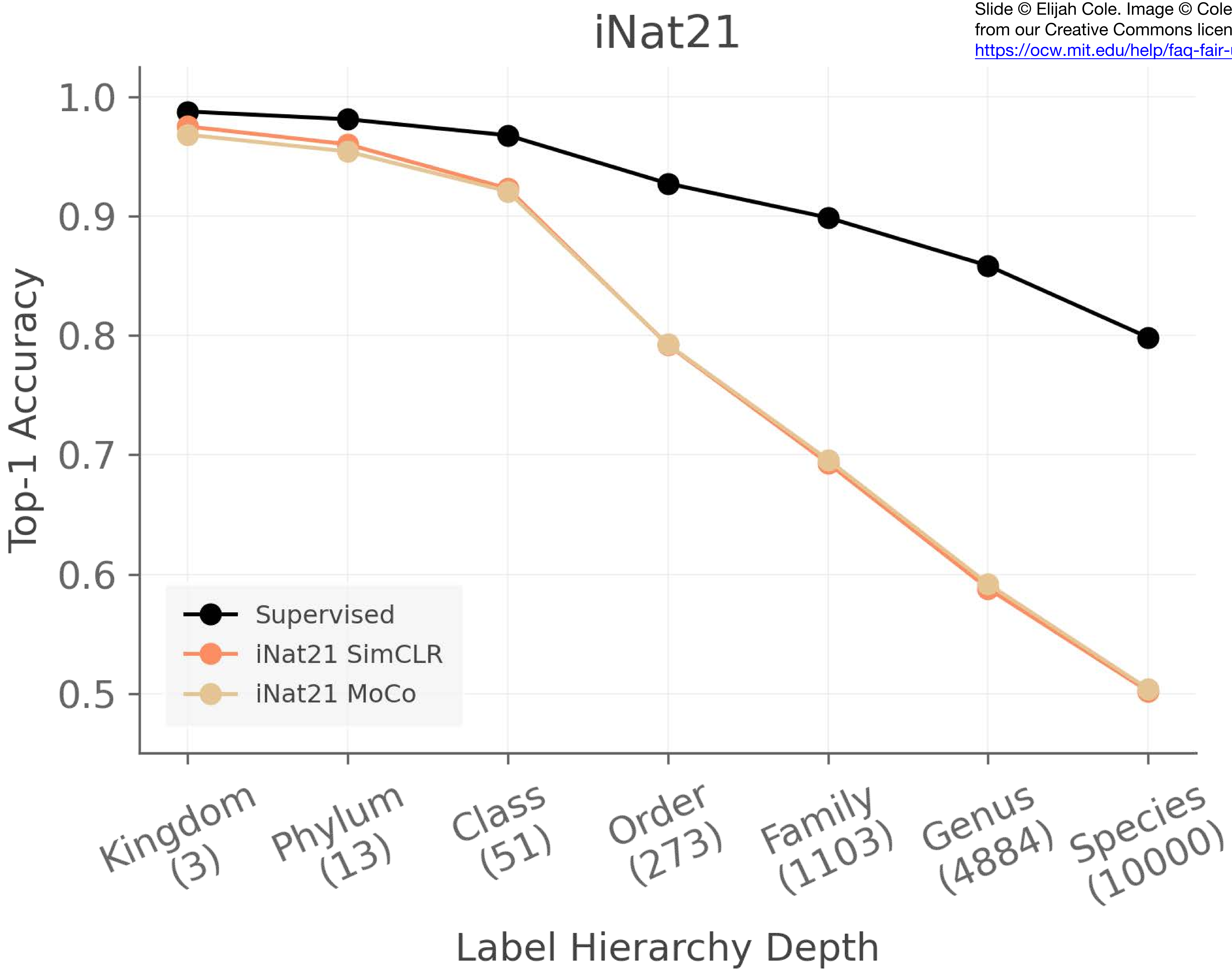
S. umbilicata

S. ornata

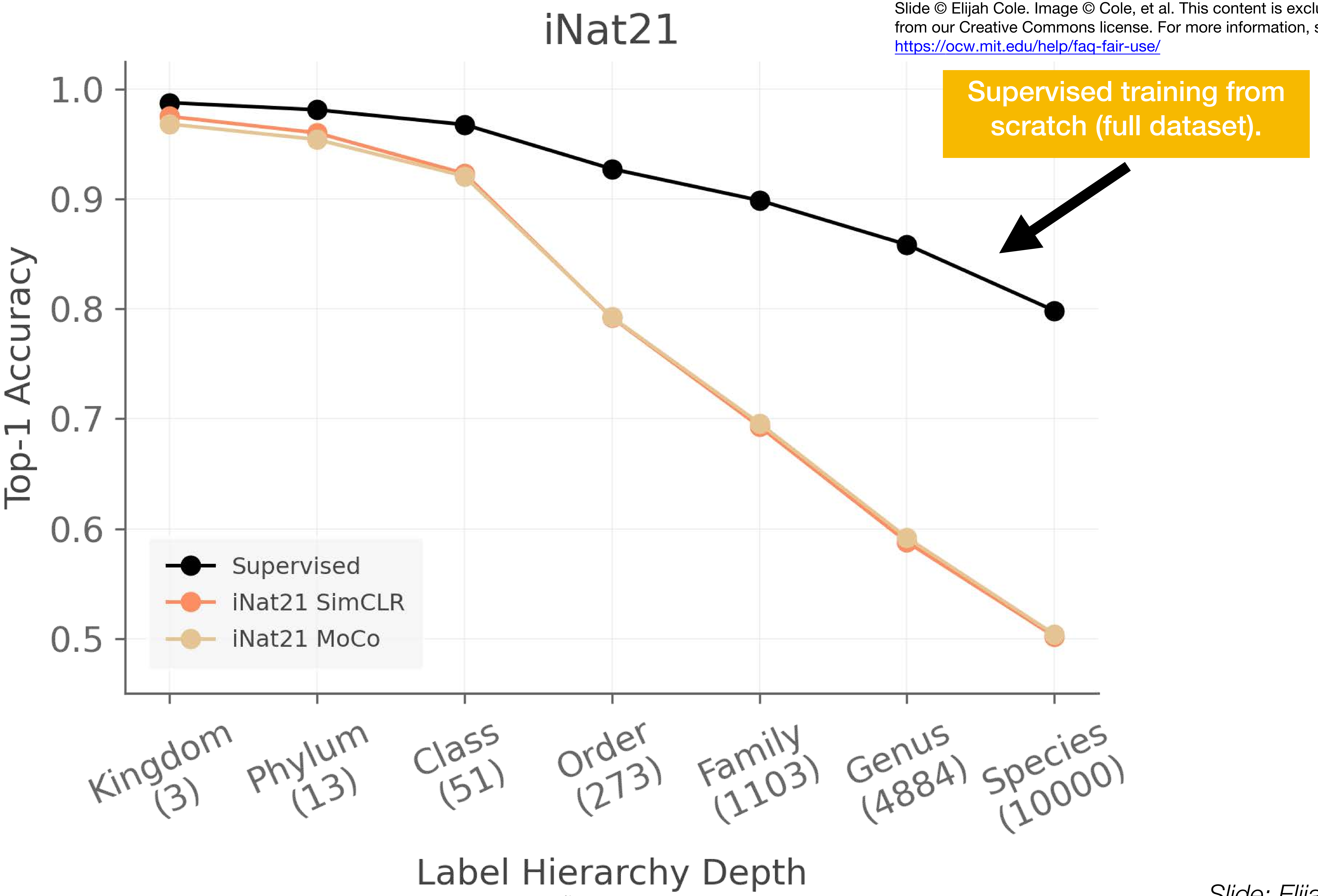
Slide: Elijah Cole



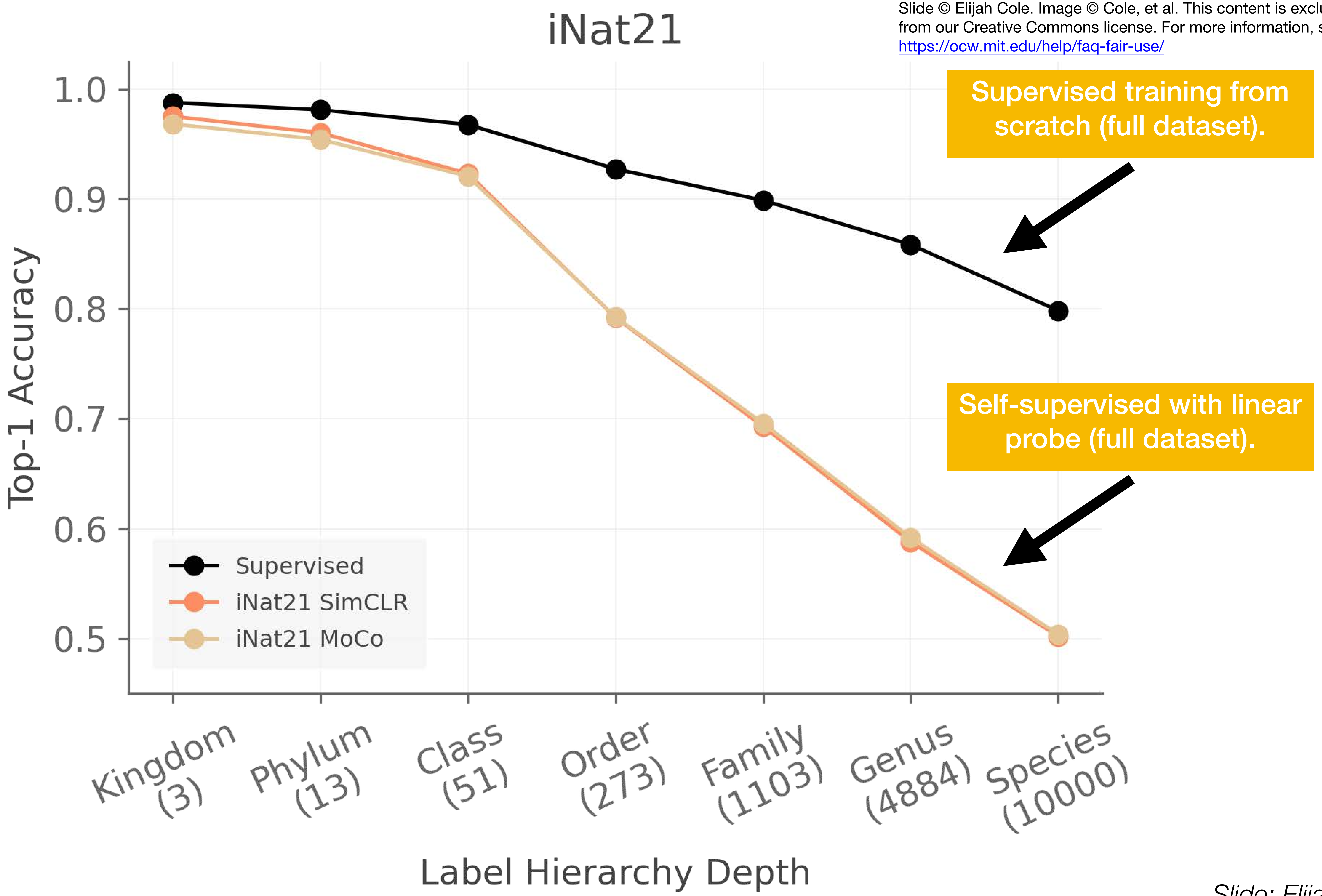
Slide: Elijah Cole



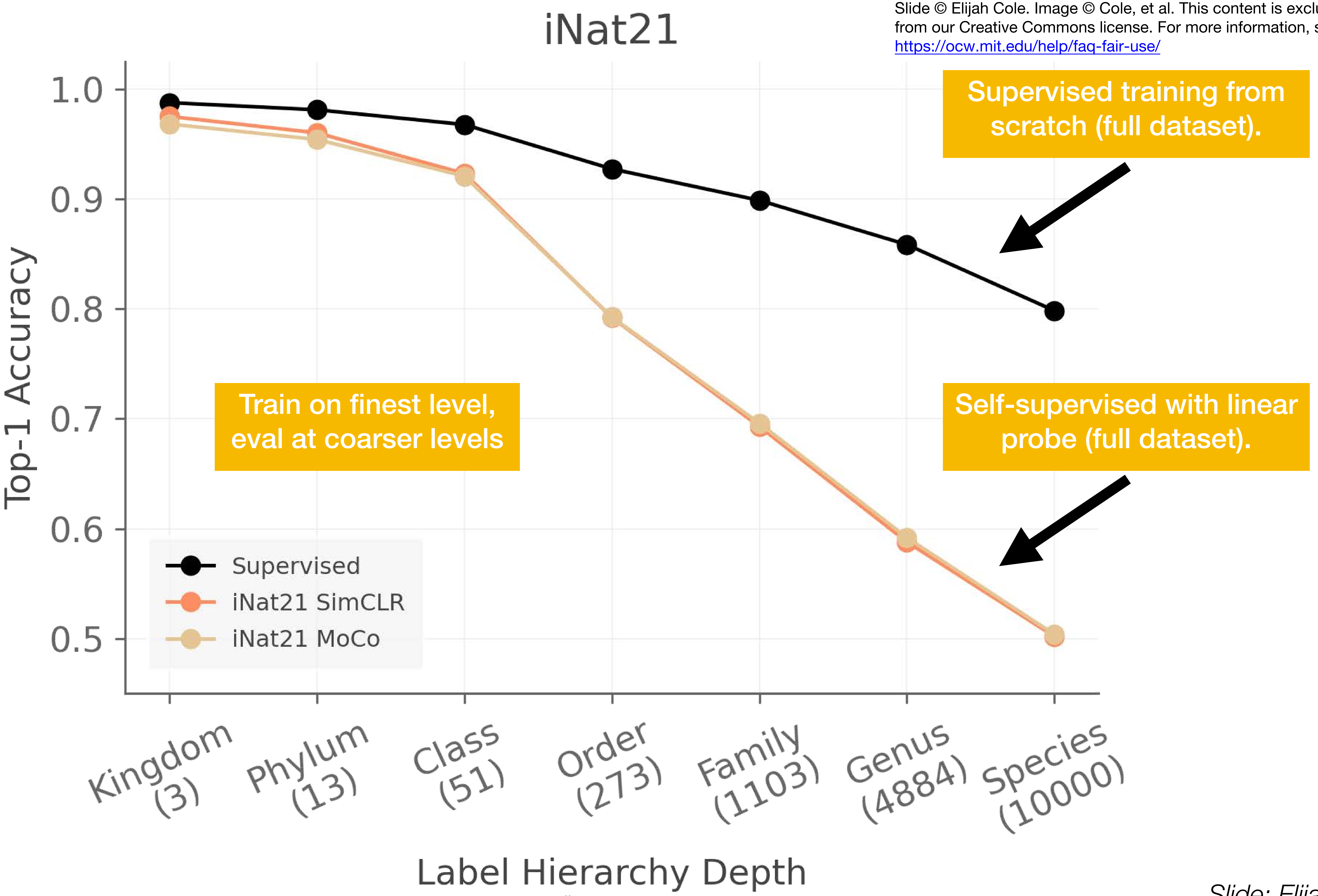
Slide: Elijah Cole



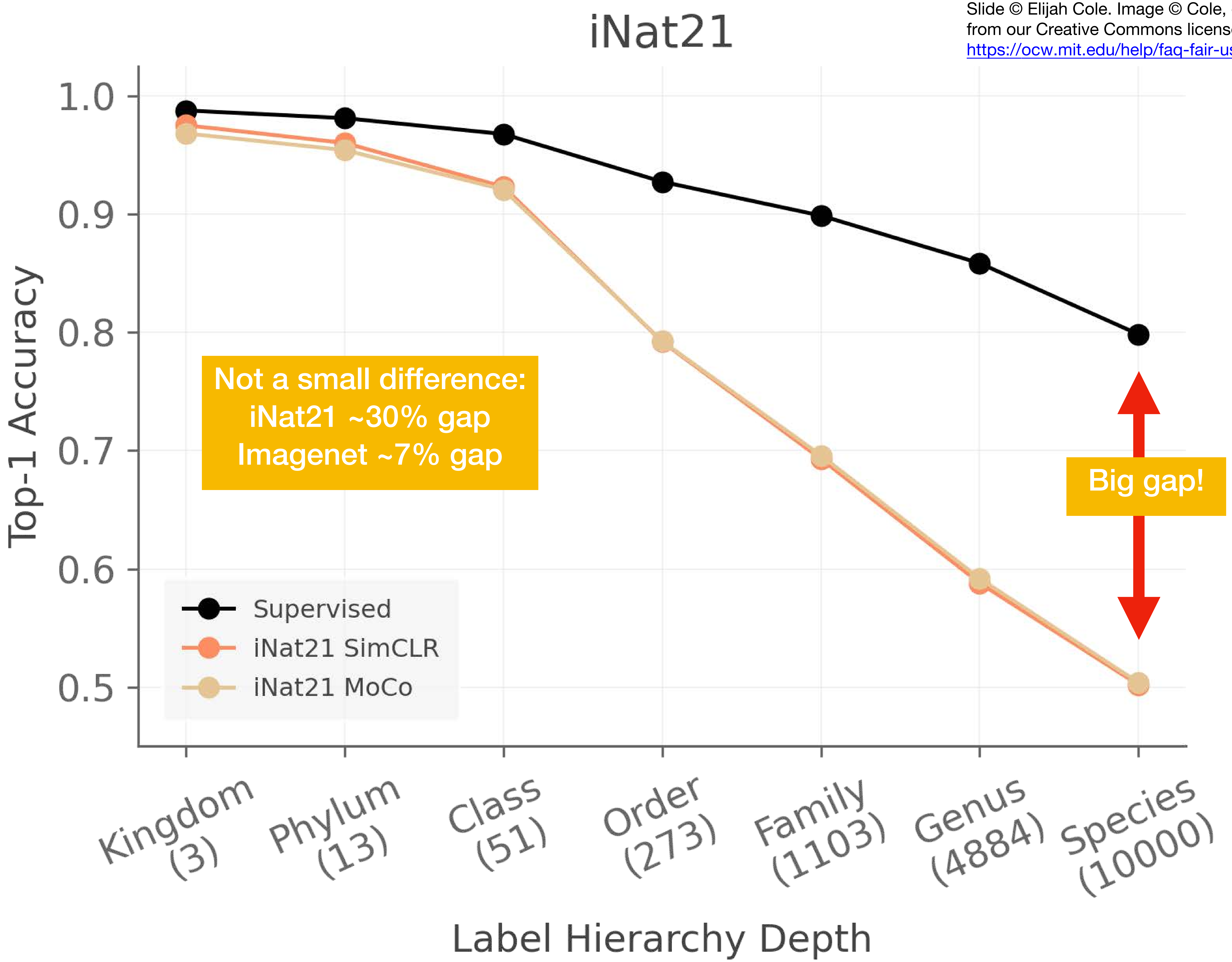
Slide: Elijah Cole



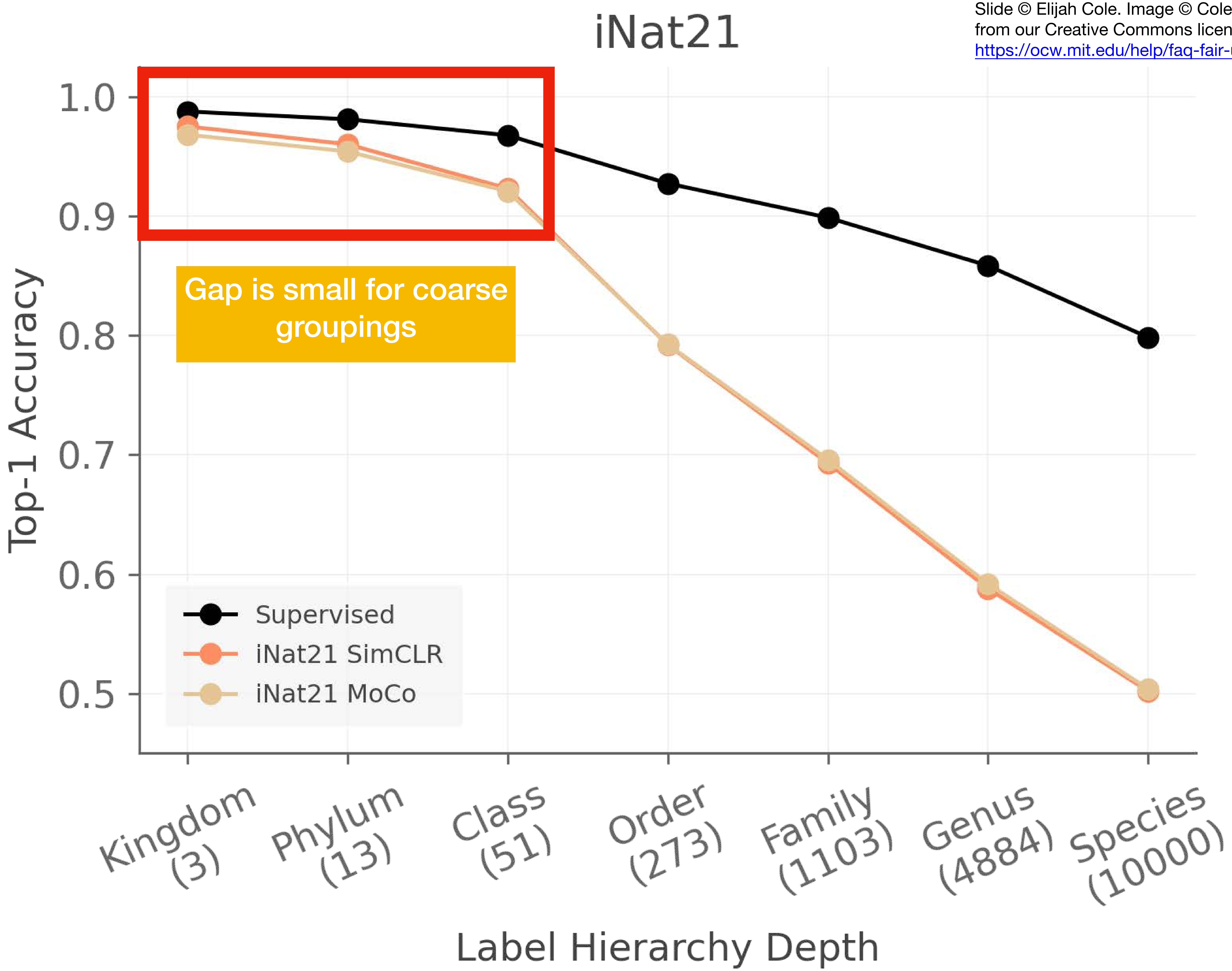
Slide: Elijah Cole



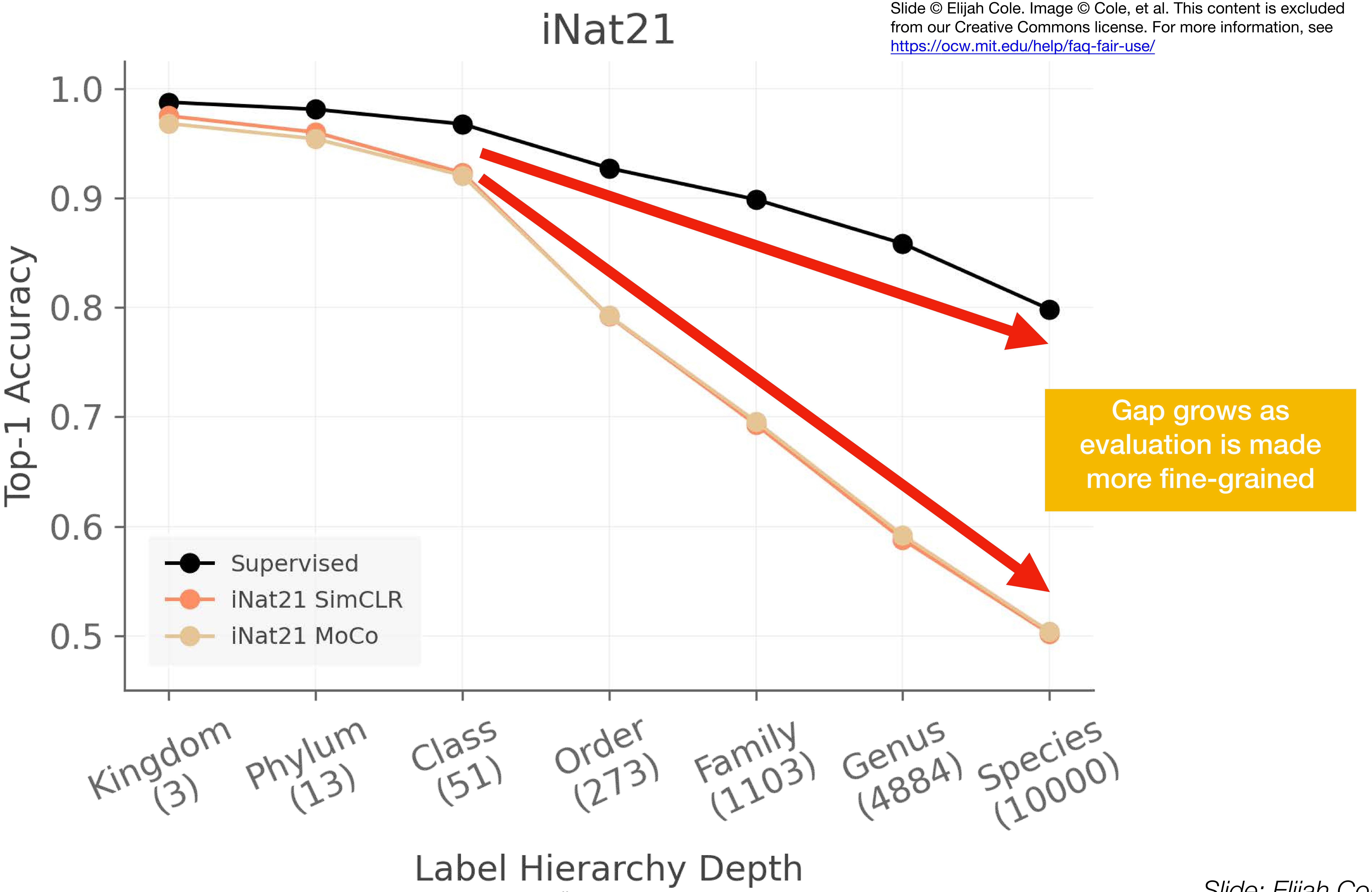
Slide: Elijah Cole



Slide: Elijah Cole

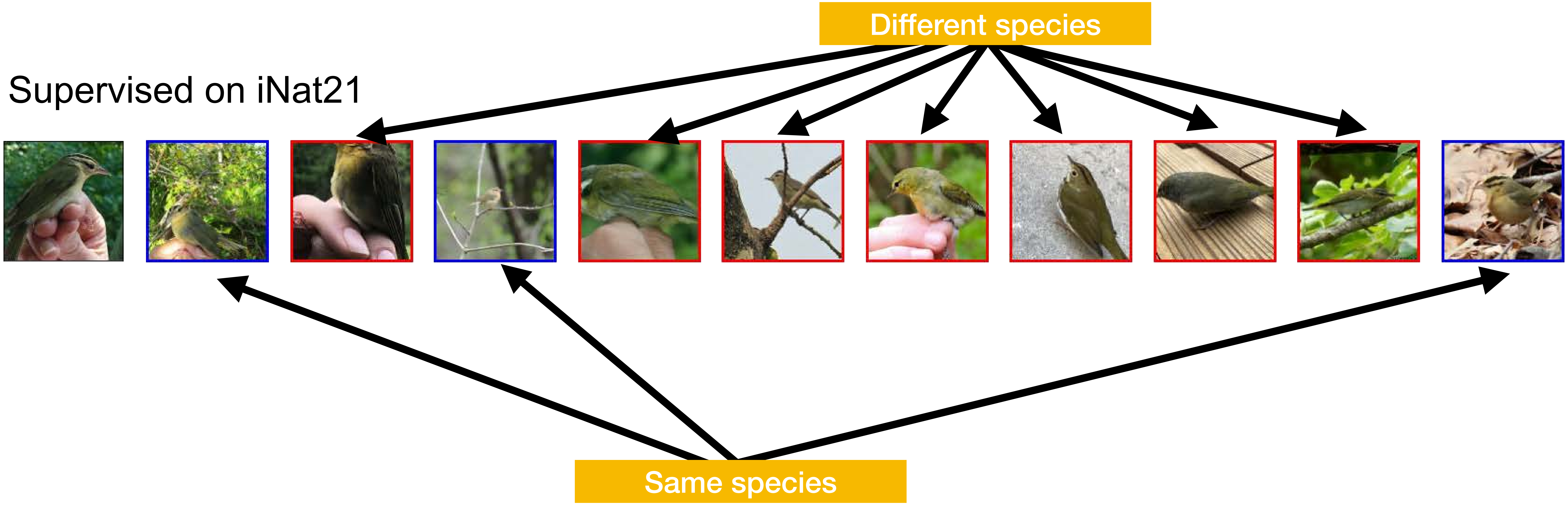


Slide: Elijah Cole



Slide: Elijah Cole

Supervised on iNat21



Slide: Elijah Cole

On ImageNet, contrastive SSL matches supervised.

On iNat21, contrastive SSL lags far behind

Supervised on iNat21

SimCLR on iNat21

Cole et al., *When Does Contrastive Visual Representation Learning Work?*, CVPR 2022

Slide: Elijah Cole

Summary

- Good representations capture relevant similarity/dissimilarity information
- well-clustered, compact and separated/spread out classes:
  - preserves relevant information
  - teaches relevant invariances (“forget” irrelevant information)
- supervised or self-supervised

6.7960 Deep Learning
Fall 2024