NEURAL INFORMATION PROCESSING SYSTEMS

NAVER LABS Europe

Hard negative mixing for contrastive learning

Yannis Kalantidis   Mert Bulent Sariyildiz   Noé Pion

Philippe Weinzaepfel   Diane Larlus

Project page
https://europe.naverlabs.com/mochi

Overview

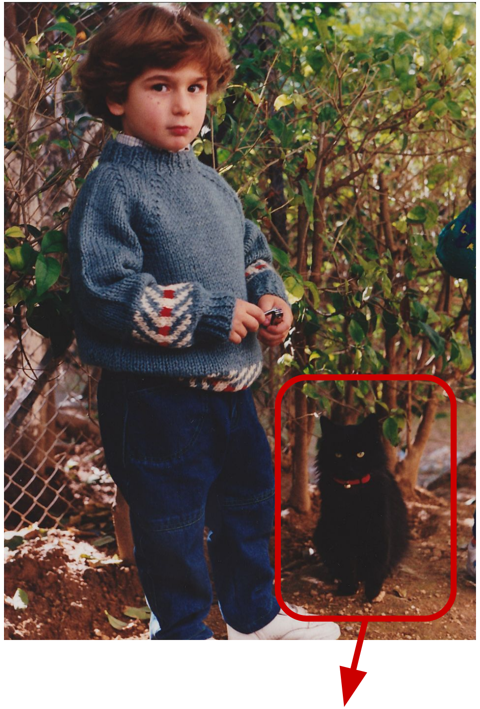
- Introduction
- Contrastive self-supervised learning
- Hard Negative Mixing (MoCHi)
- Evaluation and results
- Understanding the feature space

Overview

- Introduction
- Contrastive self-supervised learning
- Hard Negative Mixing (MoCHi)
- Evaluation and results
- Understanding the feature space

About Yannis

- Grew up in Athens, Greece
- 2009 - 2014: PhD in Athens, Greece
  - at the National Technical University of Athens
  - PhD supervised by Yannis Avrithis
  - Internships at
    - Yahoo Research Barcelona
    - Yahoo Research San Francisco (two times!)
- 2015 - 2017: Researcher at Yahoo Research (SF)
- 2017 - 2019: Researcher at Facebook AI (MPK)
- 2020- now: Researcher at NAVER LABS Europe



first cat of the lecture

Computer vision over the last decade

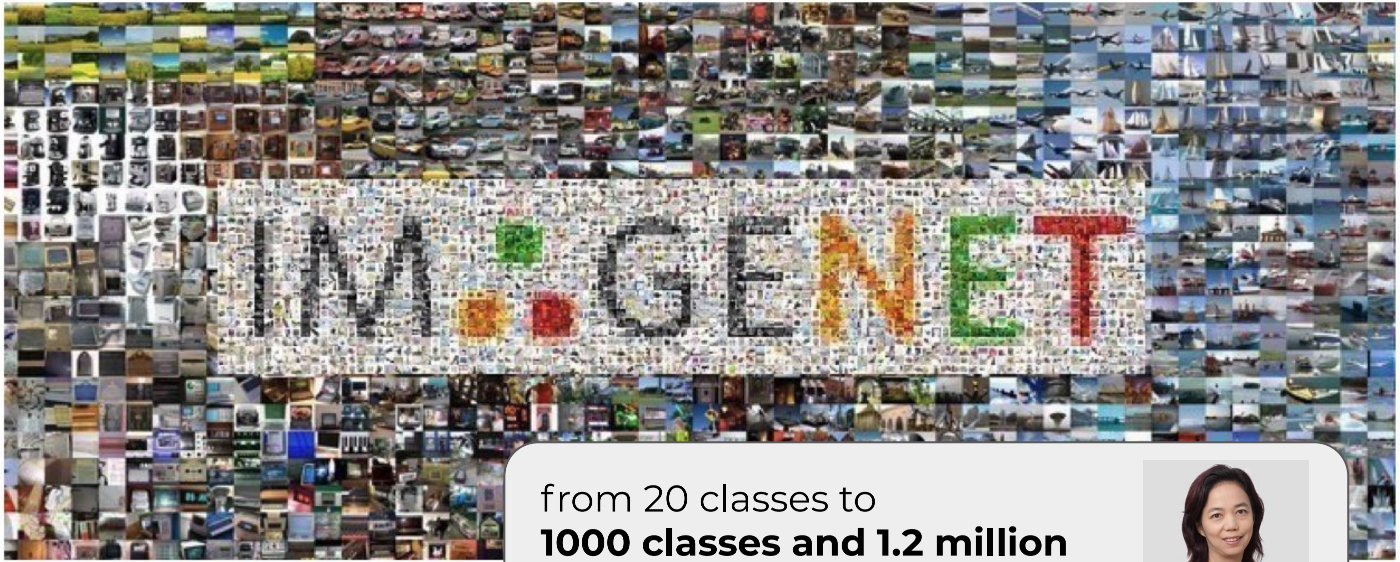
Large image collections to train deep Convolutional Neural Networks (CNN)

2005 Pascal VOC

2006 Pascal VOC

2007 Pascal VOC

2010 ImageNet



from 20 classes to
1000 classes and 1.2 million annotated images

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, **ImageNet: A Large-Scale Hierarchical Image Database.** *(CVPR), 2009.* pdf

Computer vision over the last decade

From hand-crafted to learned visual representations

Computer Vision + Machine Learning =
Visual Representation Learning

Representation Learning

- Don’t design features
- Design ***models*** that output representations and predictions
- Don’t tell the model how to solve your task; tell the model what result you want to get

Image Classification

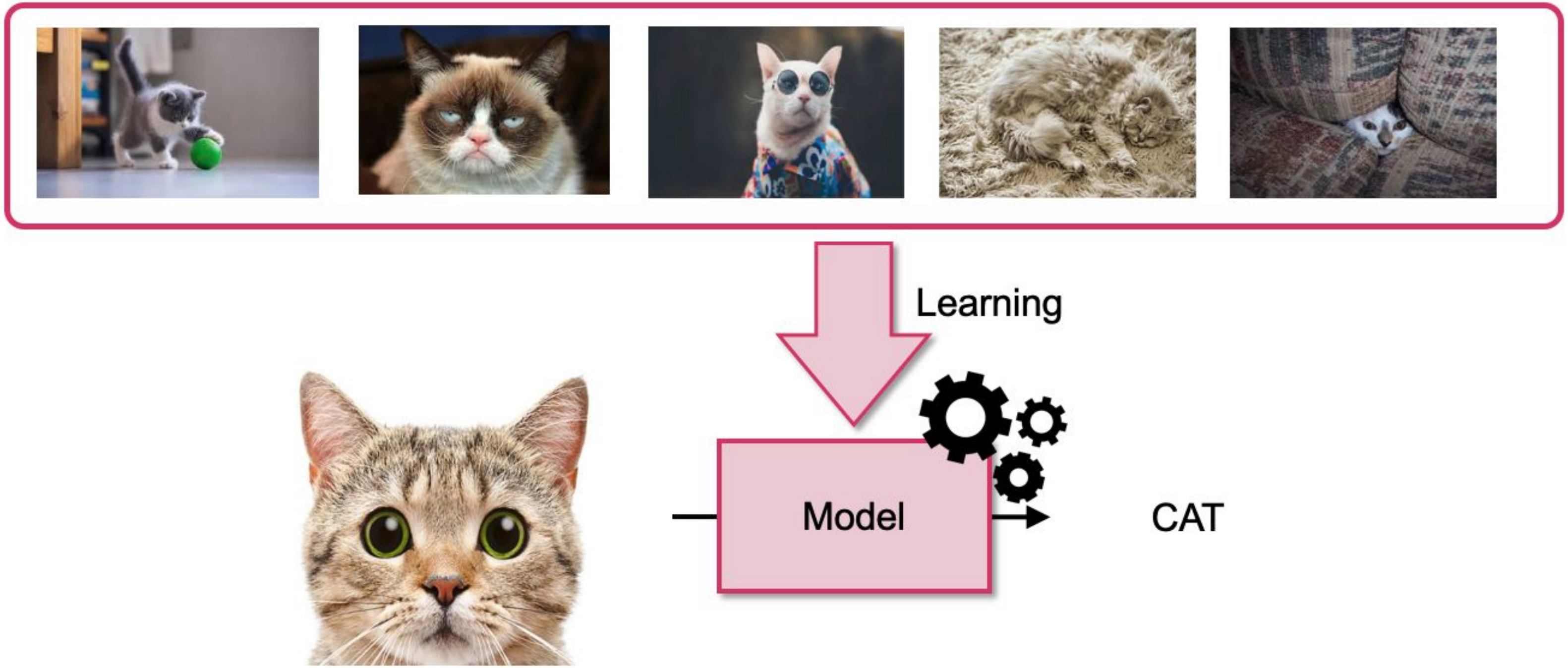


Image Classification

Given a (large) dataset of images and corresponding labels:

1. Learn visual representations
2. Learn a *classifier* on top of the representations

$$f(x_i; W) = Wx_i$$

They two can be learned *together* (end-to-end)

Image Classification

Given **a (large) dataset** of images and **corresponding labels**:

1. Learn visual representations
2. Learn a *classifier* on top of the representations

$$f(x_i; W) = W x_i$$

They two can be learned *together* (end-to-end)

The annotation bottleneck

Can we learn “reusable” / “general-purpose” visual representations…

… and use/*transfer* them for other tasks and datasets?

The annotation bottleneck

Can we learn "reusable" / "general-purpose" visual representations...

... and use/*transfer* them for other tasks and datasets?

Yes!

- Pretrained models have boosted performance on many tasks
- We can pretrain with large weakly annotated datasets
- Big gains for smaller target datasets

Razavian et al. CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRw 2014.
Mahajan, et al. "Exploring the limits of weakly supervised pretraining." ECCV 2018.
Yalniz et al. Billion-scale semi-supervised learning for image classification. Arxiv 2018.
Kolesnikov et al. "Big transfer (bit): General visual representation learning." Arxiv 2019.

The annotation bottleneck

Can we learn “reusable” / “general-purpose” visual representations...

... and use/*transfer* them for other tasks and datasets?

Yes!

- Pretrained models have boosted performance on many tasks
- We can pretrain with large weakly annotated datasets
- Big gains for smaller target datasets

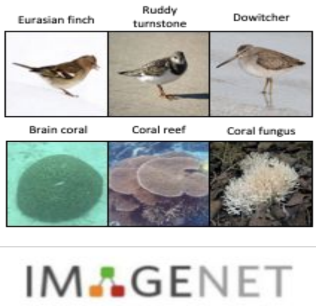
Do we really need labeled datasets for pretraining?

Razavian et al. CNN Features off-the-shelf: an Astounding Baseline for Recognition. CVPRw 2014.
Mahajan, et al. "Exploring the limits of weakly supervised pretraining." ECCV 2018.
Yalniz et al. Billion-scale semi-supervised learning for image classification. Arxiv 2018.
Kolesnikov et al. "Big transfer (bit): General visual representation learning." Arxiv 2019.

Learning transferable visual representations

Supervised learning

Train with supervision for <u>classification</u> on ImageNet
fine-grained annotations
expert knowledge



Model

Self-supervised learning:
Can we learn transferable visual representations without annotations?

Transfer Learning

Downstream tasks

Self-Supervised learning

Train on a <u>proxy task</u> (self-supervised)
annotation-free images
no annotation required

Model

Self-supervised learning (or SSL)

- Train on a proxy task (self-supervised)
  - Not (necessarily) an “important” task we care about
  - A task that is defined from the input data alone
  - Should still be a hard task
  - Should enable us to learn aspects of the visual input/world
- No annotations required
  - Scalability: use “any” image/video - no need for labels
  - Flexibility: find the data that fits your downstream task

Self-supervised learning (or SSL)

- Train on a proxy task (self-supervised)
  - Not (necessarily) an "important" task we care about
  - A task that is defined from the input data alone
  - Should still be a hard task
  - Should enable us to learn aspects of the visual input/world
- No annotations required
  - Scalability: use "any" image/video - no need for labels
  - Flexibility: find the data that fits your downstream task

"Does this mean that I don't need to care about what data I use anymore?"

Self-supervised learning (or SSL)

- Train on a proxy task (self-supervised)
  - Not (necessarily) an “important” task we care about
  - A task that is defined from the input data alone
  - Should still be a hard task
  - Should enable us to learn aspects of the visual input/world
- No annotations required
  - Scalability: use “any” image/video - no need for labels
  - Flexibility: find the data that fits your downstream task

“Does this mean that I don’t need to care about what data I use anymore?”

Of course not!

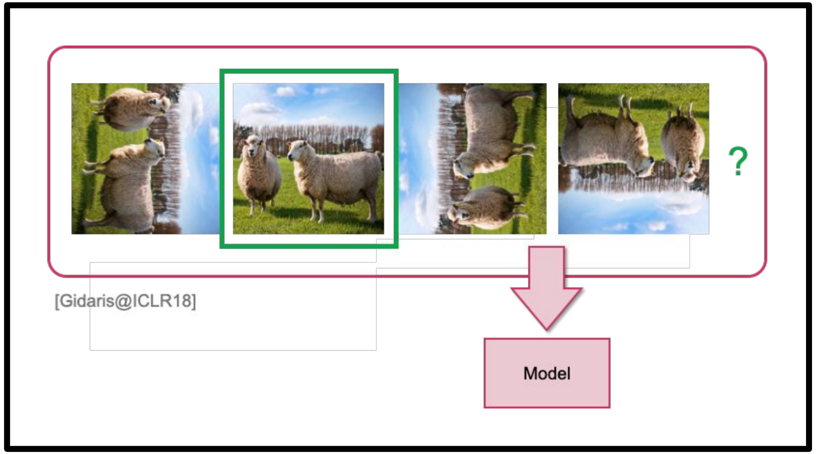
Self-supervised learning (or SSL)

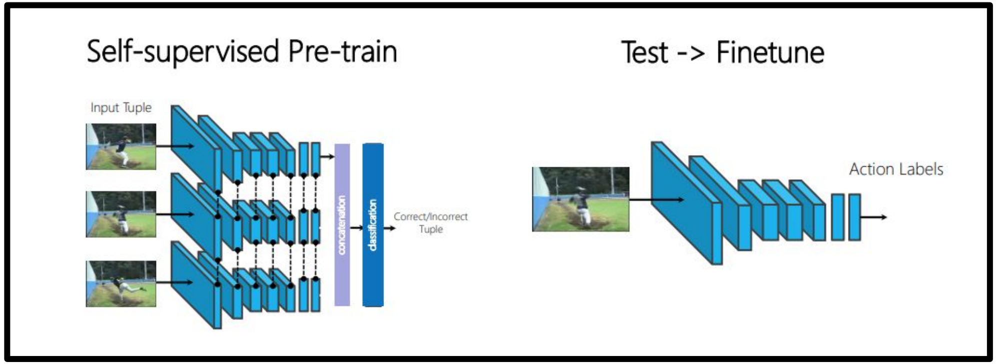
- Train on a proxy task (self-supervised)
  - A task that is defined from the input data alone
  - Should enable us to learn aspects of the visual input/world
  - **Predictive** or **Contrastive** proxy tasks

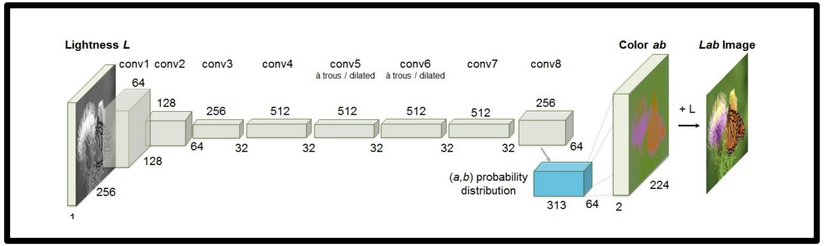
Source: *Ankesh Anand, "Contrastive Self-Supervised Learning" (2020)"*

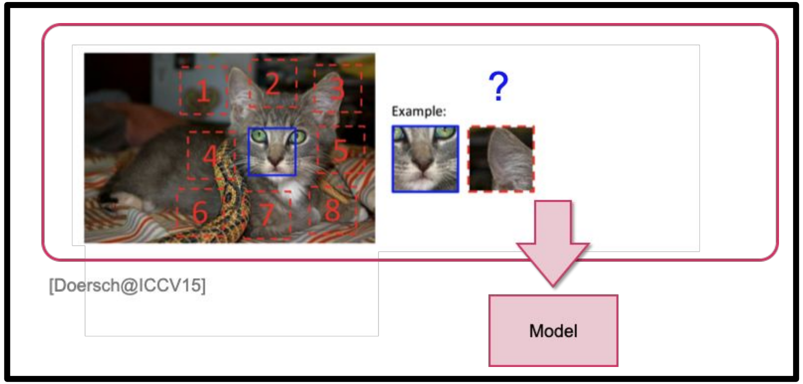
slide credit: Xavi Giró-i-Nieto

Predictive tasks for self-supervised learning



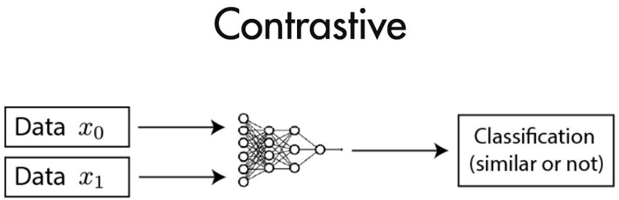






Misra, Ishan, C. Lawrence Zitnick, and Martial Hebert. **Shuffle and learn: unsupervised learning using temporal order verification.** ECCV 2016.
Gidaris, S., Singh, P., & Komodakis, N. (2018). **Unsupervised representation learning by predicting image rotations.** ICLR 2018
Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. **Unsupervised visual representation learning by context prediction.** ICCV. 2015.
Zhang, R., Isola, P., & Efros, A. A. **Colorful image colorization.** ECCV 2016.

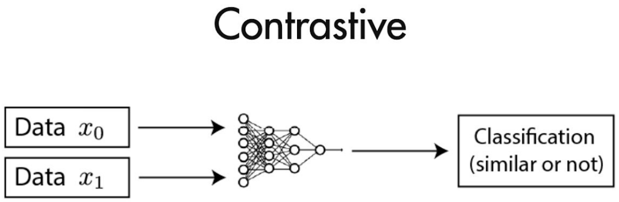
Contrastive tasks for self-supervised learning



- Contrast features from different (overlapping) patches [CPC]
- Discriminate individual instances [InstDiscr]
- Learning representations invariant to image transformations [MoCo, SimCLR, PIRL, SwAV, BYOL, many more]

[CPC] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[InstDiscr] Z Wu, Y Xiong, SX Yu, D Lin, "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[PIRL] Misra, Ishan, and Laurens van der Maaten. "Self-supervised learning of pretext-invariant representations." CVPR 2020.
[SwAV] Caron, Mathilde, et al. "Unsupervised learning of visual features by contrasting cluster assignments." NeurIPS 2020.
[InfoMin Aug.] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.
[BYOL] Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." NeurIPS 2020.

Contrastive tasks for self-supervised learning



- Contrast features from different (overlapping) patches [CPC]
- Discriminate individual instances [InstDiscr]
- Learning representations invariant to image transformations [MoCo, SimCLR, PIRL, SwAV, BYOL, many more]

[CPC] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[InstDiscr] Z Wu, Y Xiong, SX Yu, D Lin. "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[PIRL] Misra, Ishan, and Laurens van der Maaten. "Self-supervised learning of pretext-invariant representations." CVPR 2020.
[SwAV] Caron, Mathilde, et al. "Unsupervised learning of visual features by contrasting cluster assignments." NeurIPS 2020.
[InfoMin Aug.] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.
[BYOL] Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." NeurIPS 2020.

Contrastive Learning

- Given a set of "similar" and "dissimilar" pairs of inputs
- Learn the ***ranking*** of similarities, *i.e.*, learn representations such that the *similarity between "similar" inputs is higher than "dissimilar"*

Measuring similarity

$$cos(x_i, x_j) = \frac{x_i^T x_j}{||x_i|| \, ||x_j||}$$

Contrastive Learning with labels

Pairwise loss

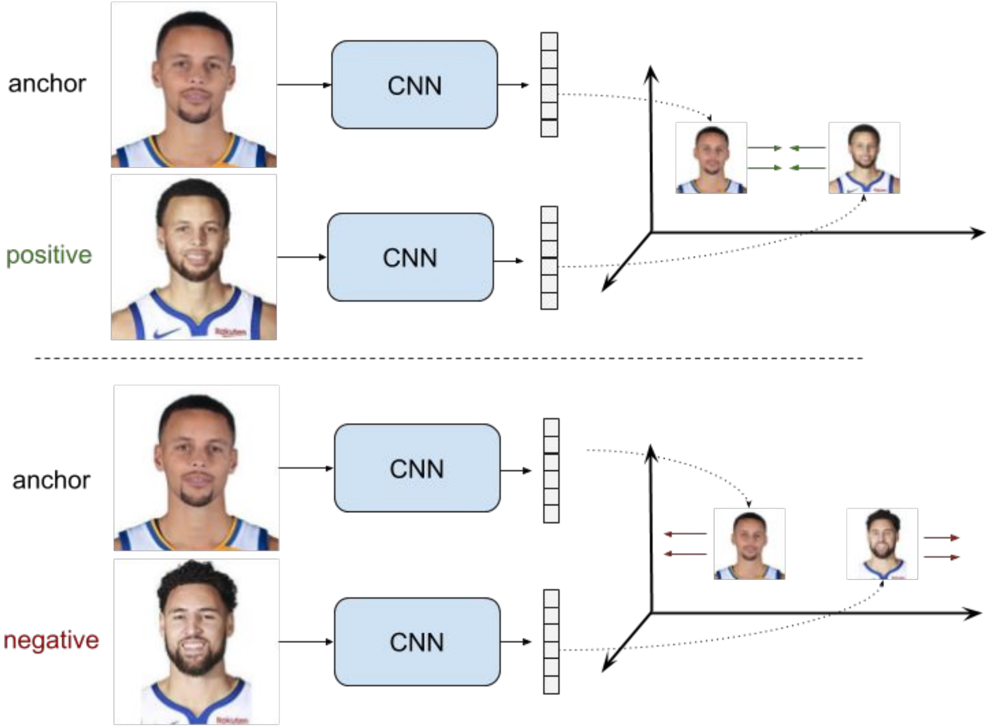


Figure from [“Understanding Ranking Loss, Contrastive Loss, Margin Loss, Triplet Loss, Hinge Loss and all those confusing names”](#) (2019)

slide credit: Xavi Giró-i-Nieto

Contrastive Learning with labels

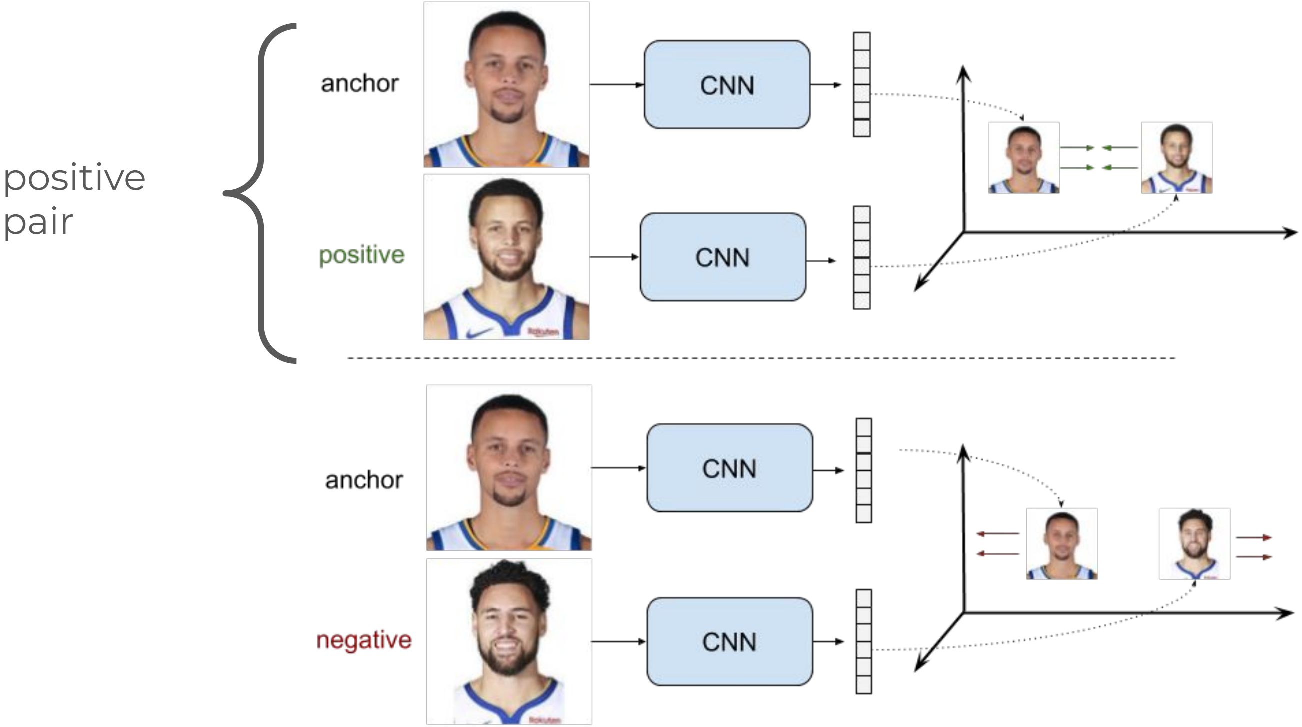


Figure from [“Understanding Ranking Loss, Contrastive Loss, Margin Loss, Triplet Loss, Hinge Loss and all those confusing names”]() (2019)

slide credit: Xavi Giró-i-Nieto

Contrastive Learning with labels

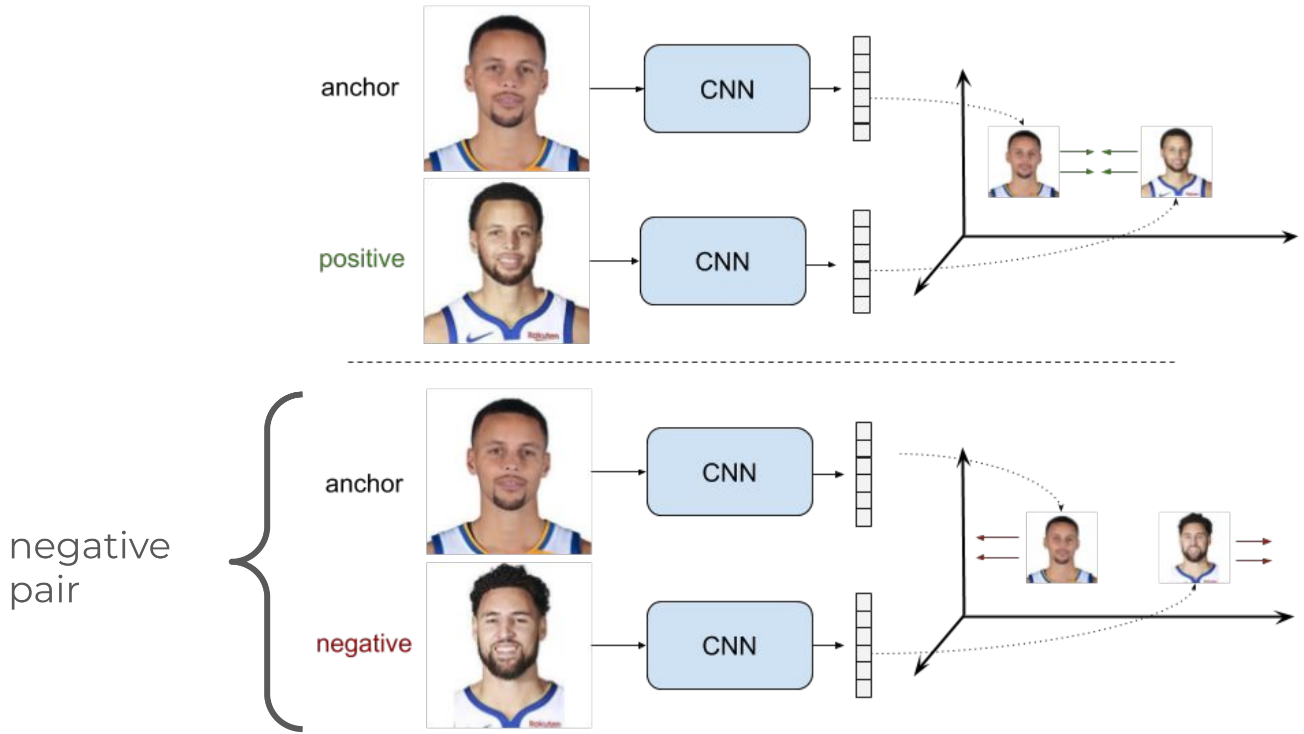


Figure from [“Understanding Ranking Loss, Contrastive Loss, Margin Loss, Triplet Loss, Hinge Loss and all those confusing names”](#) (2019)

slide credit: Xavi Giró-i-Nieto

Contrastive Learning with labels

Triplet loss

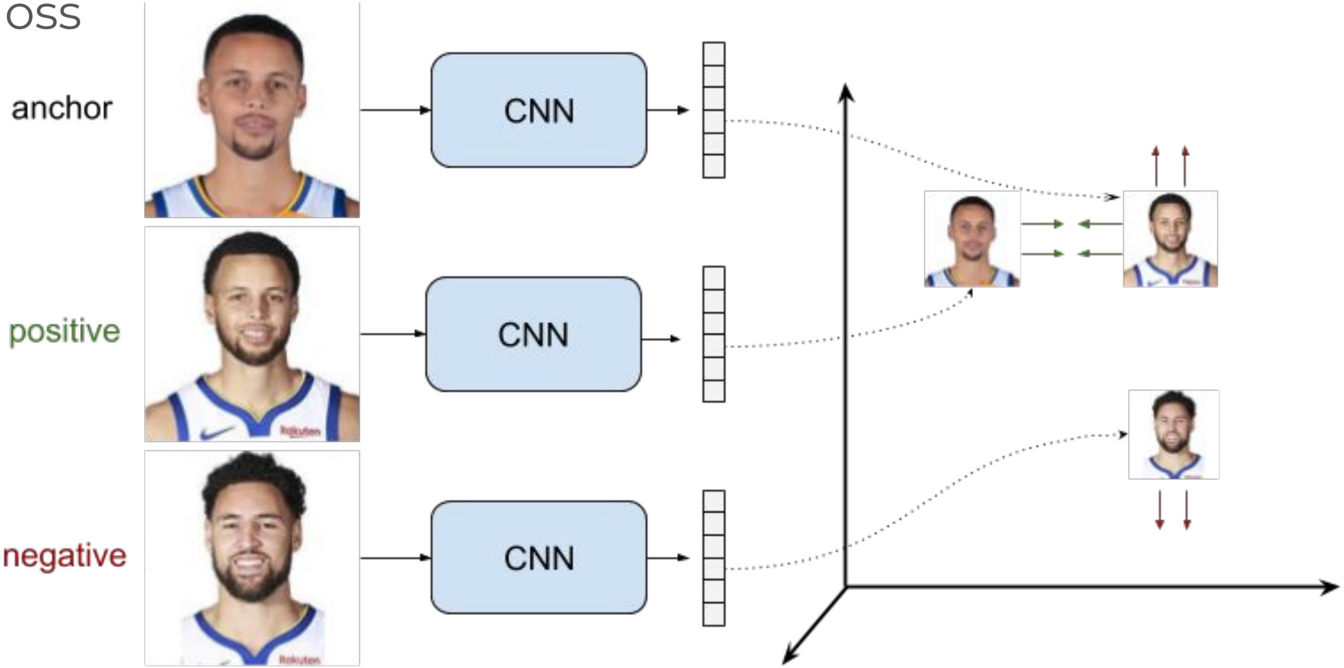


Figure from "Understanding Ranking Loss, Contrastive Loss, Margin Loss, Triplet Loss, Hinge Loss and all those confusing names" (2019)

slide credit: Xavi Giró-i-Nieto

Contrastive Learning

Why not use **multiple negatives**?

- others from the mini-batch
- or features from a memory

InfoNCE loss [CPC]:

- Learn by contrasting the similarity of the positive pair, with the similarities between the anchor and *a set of* negatives

(we will discuss this in detail soon)

[CPC] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv* 2018.

Overview

- Introduction
- Contrastive self-supervised learning
- Hard Negative Mixing (MoCHi)
- Evaluation and results
- Understanding the feature space

Contrastive self-supervised learning

- Contrastive learning, when the similar/positive and dissimilar/negative pairs are defined in a *self-supervised* way

  “a self-supervised proxy task”

- What is a good proxy task (to define positive/negative pairs)?
  - contrast features from different (overlapping) patches [CPC]
  - discriminate individual instances [InstDiscr]
  - Learning representations invariant to data augmentations

[CPC] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[InstDiscr] Z Wu, Y Xiong, SX Yu, D Lin, "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.

Contrastive self-supervised learning

- Contrastive learning, when the similar/positive and dissimilar/negative pairs are defined in a *self-supervised* way
  "a self-supervised proxy task"

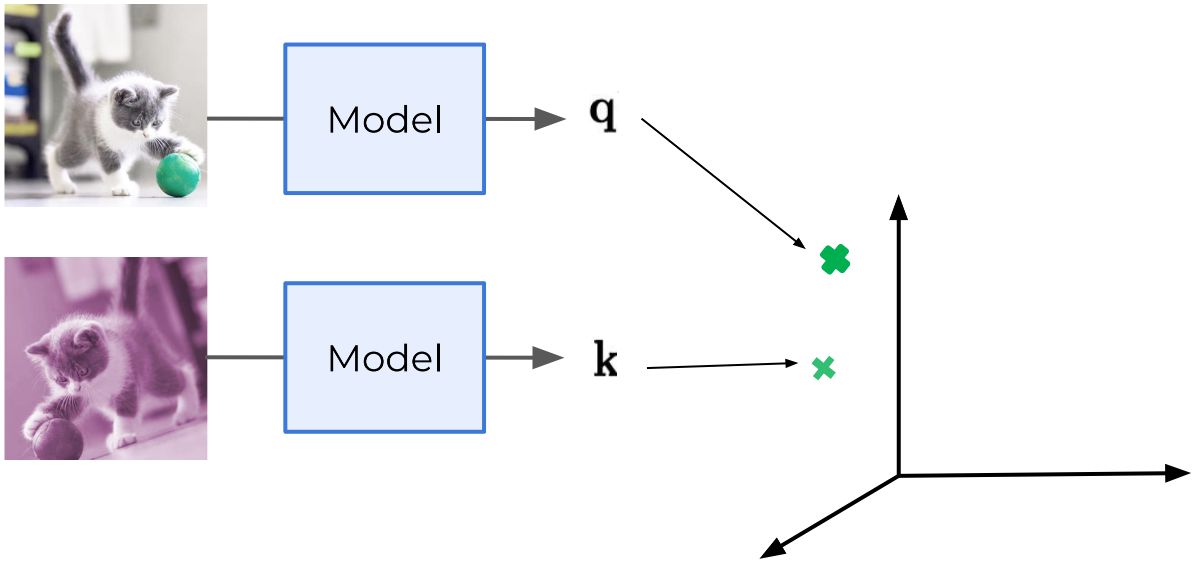
- What is a good proxy task (to define positive/negative pairs)?
  - contrast features from different (overlapping) patches [CPC]
  - discriminate individual instances [InstDiscr]
  - **Learning representations invariant to image transformations**
    [MoCo, SimCLR, PIRL, SwAV, BYOL, many more]

[CPC] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[InstDiscr] Z Wu, Y Xiong, SX Yu, D Lin, "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.

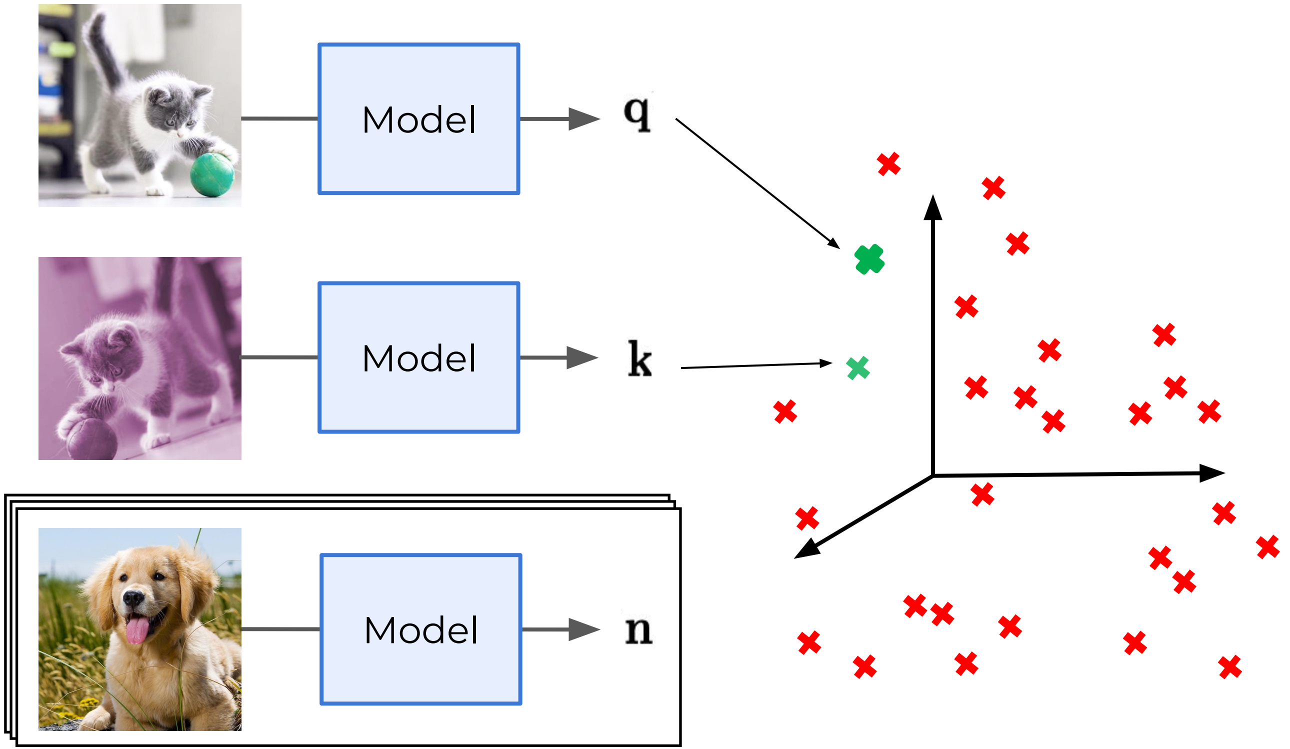
Contrastive self-supervised learning

Image Transformations

Contrastive self-supervised learning

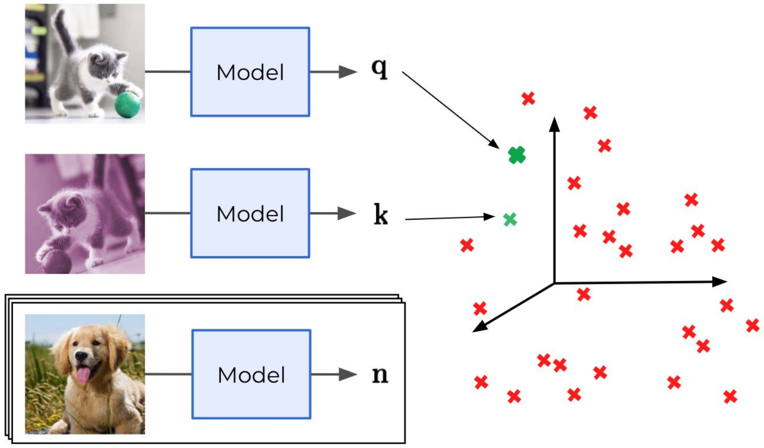


Contrastive self-supervised learning



[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning

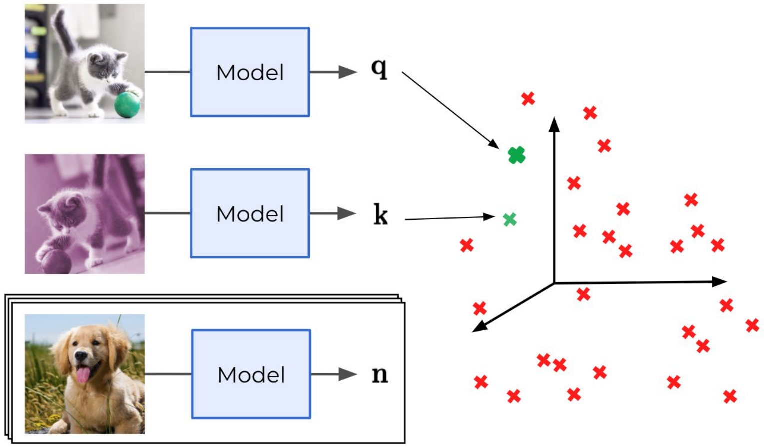


***The InfoNCE loss function* [CPC]**

$$\mathcal{L}_{\mathbf{q},\mathbf{k},Q} = -\log \frac{\exp(\mathbf{q}^T\mathbf{k}/\tau)}{\exp(\mathbf{q}^T\mathbf{k}/\tau) + \sum_{\mathbf{n}\in Q}\exp(\mathbf{q}^T\mathbf{n}/\tau)},$$

[CPC] Oord, Aaron van den, et al. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



***The InfoNCE loss function* [CPC]**

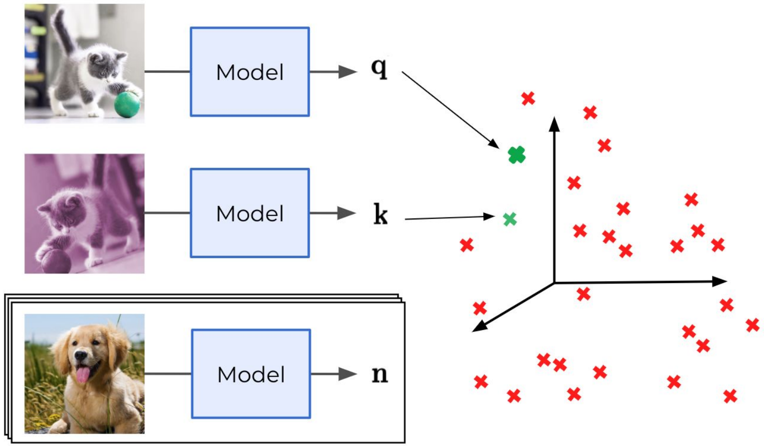
$$\mathcal{L}_{\mathbf{q},\mathbf{k},Q} = -\log \frac{\exp(\mathbf{q}^T\mathbf{k}/\tau)}{\exp(\mathbf{q}^T\mathbf{k}/\tau) + \sum_{\mathbf{n}\in Q}\exp(\mathbf{q}^T\mathbf{n}/\tau)},$$

the softmax **Cross-Entropy** loss

$$L_1 = -\log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$

[CPC] Oord, Aaron van den, et al. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



The InfoNCE loss function **[CPC]**

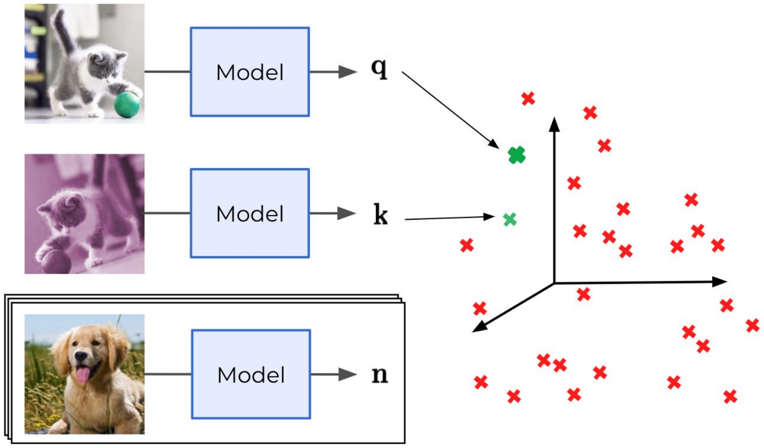
$$\mathcal{L}_{\mathbf{q},\mathbf{k},Q} = -\log \frac{\exp(\mathbf{q}^T\mathbf{k}/\tau)}{\exp(\mathbf{q}^T\mathbf{k}/\tau) + \sum_{\mathbf{n}\in Q}\exp(\mathbf{q}^T\mathbf{n}/\tau)},$$

the softmax **Cross-Entropy** loss

$$L_1 = -\log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$

[CPC] Oord, Aaron van den, et al. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



***The InfoNCE loss function* [CPC]**

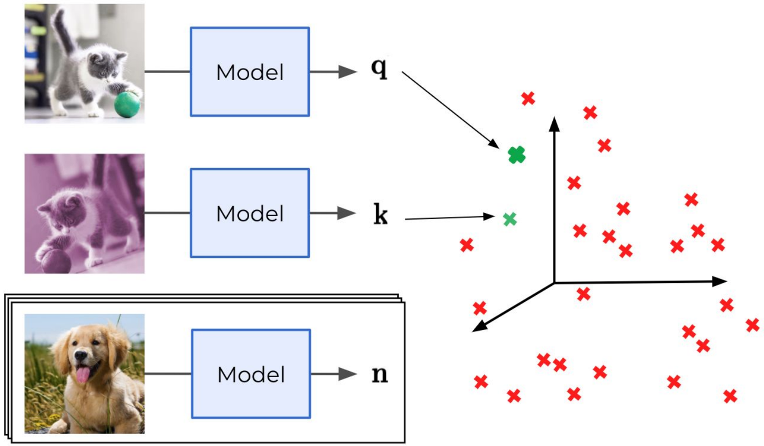
$$\mathcal{L}_{\mathbf{q},\mathbf{k},Q} = -\log \frac{\exp(\mathbf{q}^T\mathbf{k}/\tau)}{\exp(\mathbf{q}^T\mathbf{k}/\tau) + \sum_{\mathbf{n}\in Q} \exp(\mathbf{q}^T\mathbf{n}/\tau)},$$

the softmax **Cross-Entropy** loss

$$L_1 = -\log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$

[CPC] Oord, Aaron van den, et al. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



The InfoNCE loss function **[CPC]**

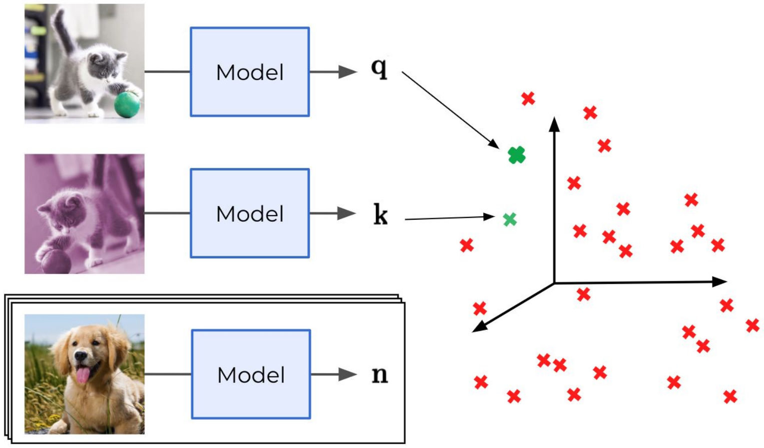
$$\mathcal{L}_{\mathbf{q},\mathbf{k},Q} = -\log \frac{\exp(\mathbf{q}^T\mathbf{k}/\tau)}{\exp(\mathbf{q}^T\mathbf{k}/\tau) + \sum_{\mathbf{n}\in Q}\exp(\mathbf{q}^T\mathbf{n}/\tau)},$$

Has softmax-like properties:

- We are applying a softmax function for each positive/query **q**

[CPC] Oord, Aaron van den, et al. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



The InfoNCE loss function **[CPC]**

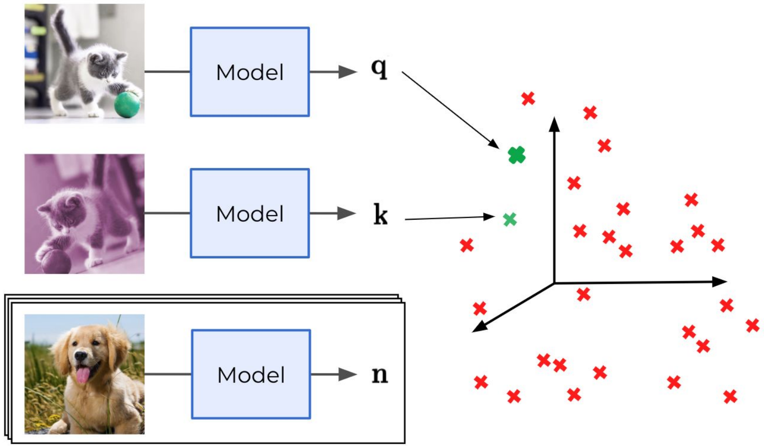
$$\mathcal{L}_{\mathbf{q},\mathbf{k},Q} = -\log \frac{\exp(\mathbf{q}^T\mathbf{k}/\tau)}{\exp(\mathbf{q}^T\mathbf{k}/\tau) + \sum_{\mathbf{n}\in Q} \exp(\mathbf{q}^T\mathbf{n}/\tau)},$$

Has softmax-like properties:

- Contributions of positive/negative logits to the loss identical to the ones for a (#neg + 1)-way cross-entropy classification loss with all gradients are scaled by 1 / τ

[CPC] Oord, Aaron van den, et al. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



Where do negatives come from?

[SimCLR]: same batch

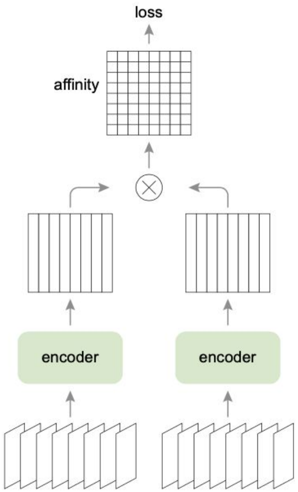
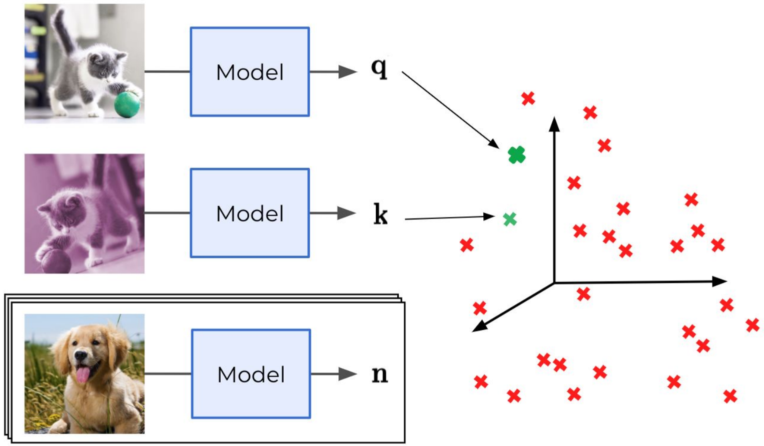


figure from [MoCo-v2]

[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)

Contrastive self-supervised learning



Where do negatives come from?

[MoCo]: queue of last batches

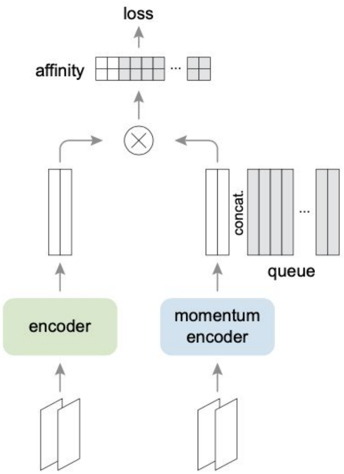
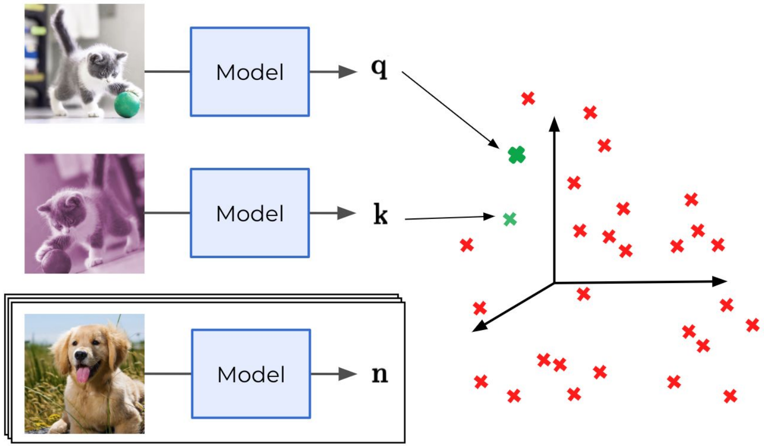


figure from [MoCo-v2]

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)

Contrastive self-supervised learning



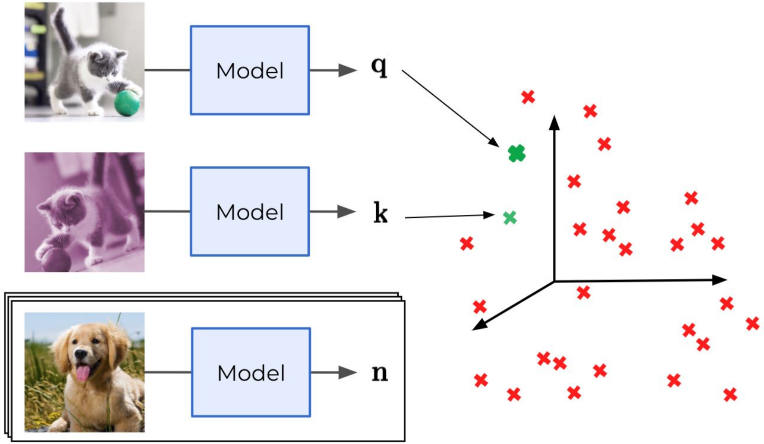
Key observation

Making the augmentation invariance proxy task more challenging leads to visual representations which generalize better

[MoCo-v2, SimCLR, InfoMin Aug, more]

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)
[InfoMin Aug.] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.

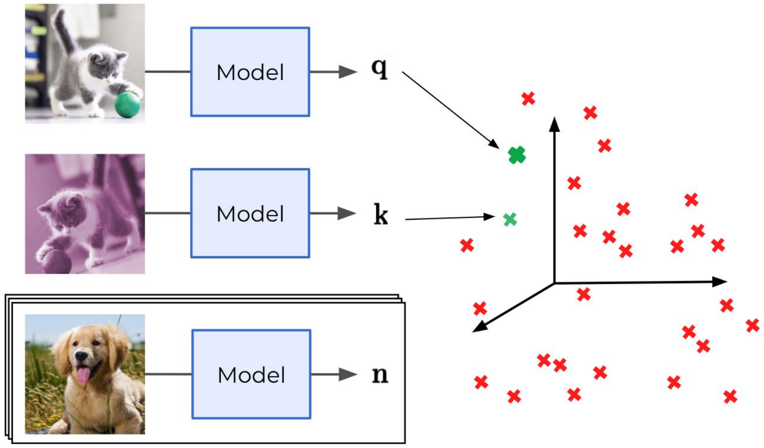
Contrastive self-supervised learning



How to make the task harder?

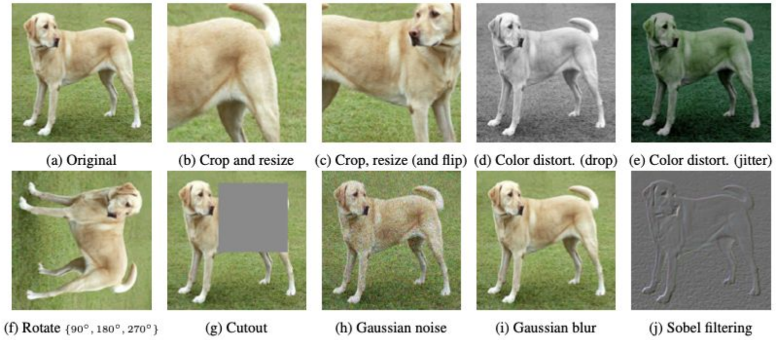
- *More challenging positive pairs*

Contrastive self-supervised learning



How to make the task harder?

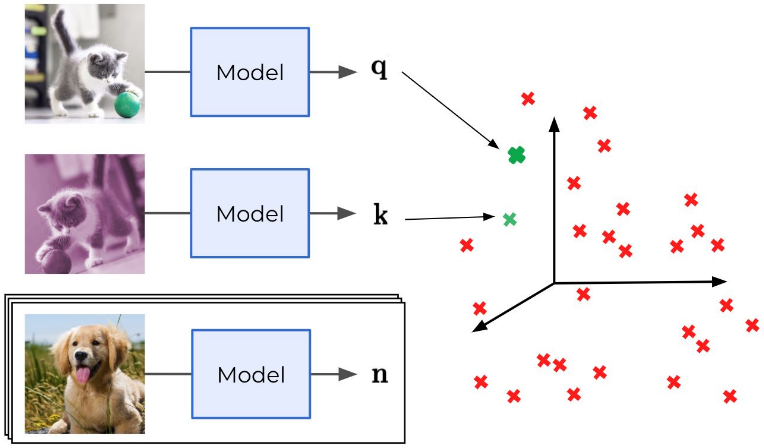
- *More challenging positive pairs*



[SimCLR]

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)
[InfoMin Aug.] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.

Contrastive self-supervised learning



How to make the task harder?

- *More challenging positive pairs*

[SimCLR]

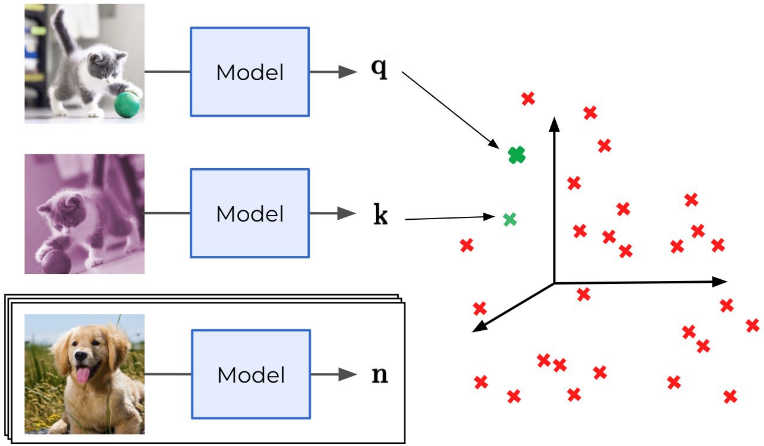
PyTorch-style data augmentation

```
RandomResizedCrop(scale=(0.2, 1.0))
RandomHorizontalFlip()
# CJ(x): random color jitter with x
cj = ColorJitter([0.8,0.8,0.8,0.4]*x)
RandomApply([cj], p=0.8)
# Blur: random blurring
blur = Blur(sigma=(0.1,2.0))
RandomApply([blur], p=0.5)
# RA: RandAugment
rnd_augment()
RandomGrayscale(p=0.2),
```

[InfoMin Aug.]

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)
[InfoMin Aug.] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.

Contrastive self-supervised learning

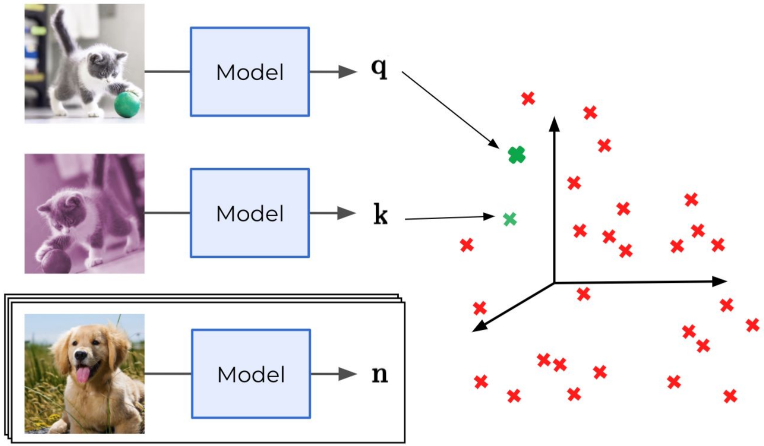


How to make the task harder?

- *More challenging positive pairs*

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning

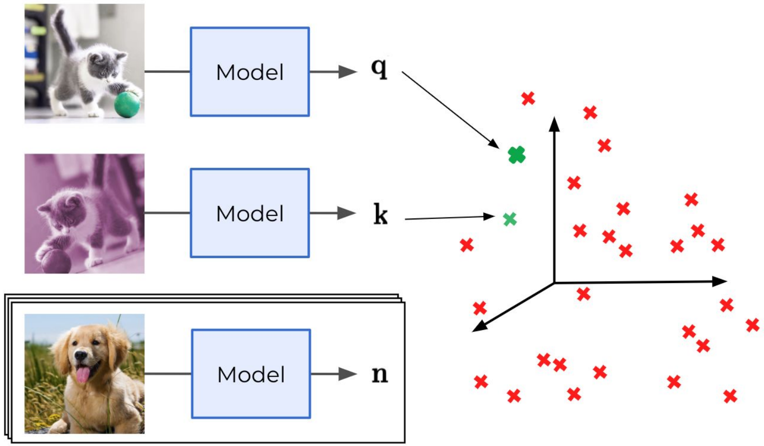


How to make the task harder?

- *More challenging positive pairs*
- *More challenging negative pairs*

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



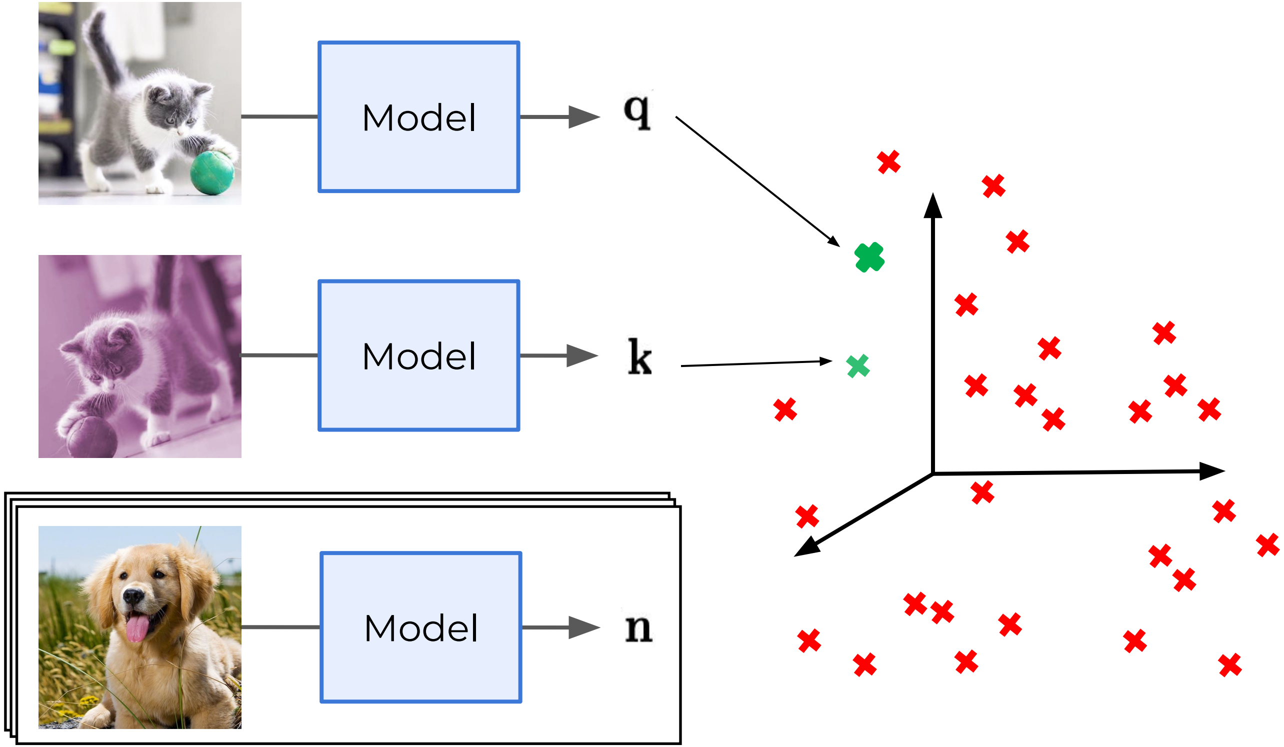
How to make the task harder?

- *More challenging positive pairs*
- *More challenging negative pairs*

How to get more challenging negatives?

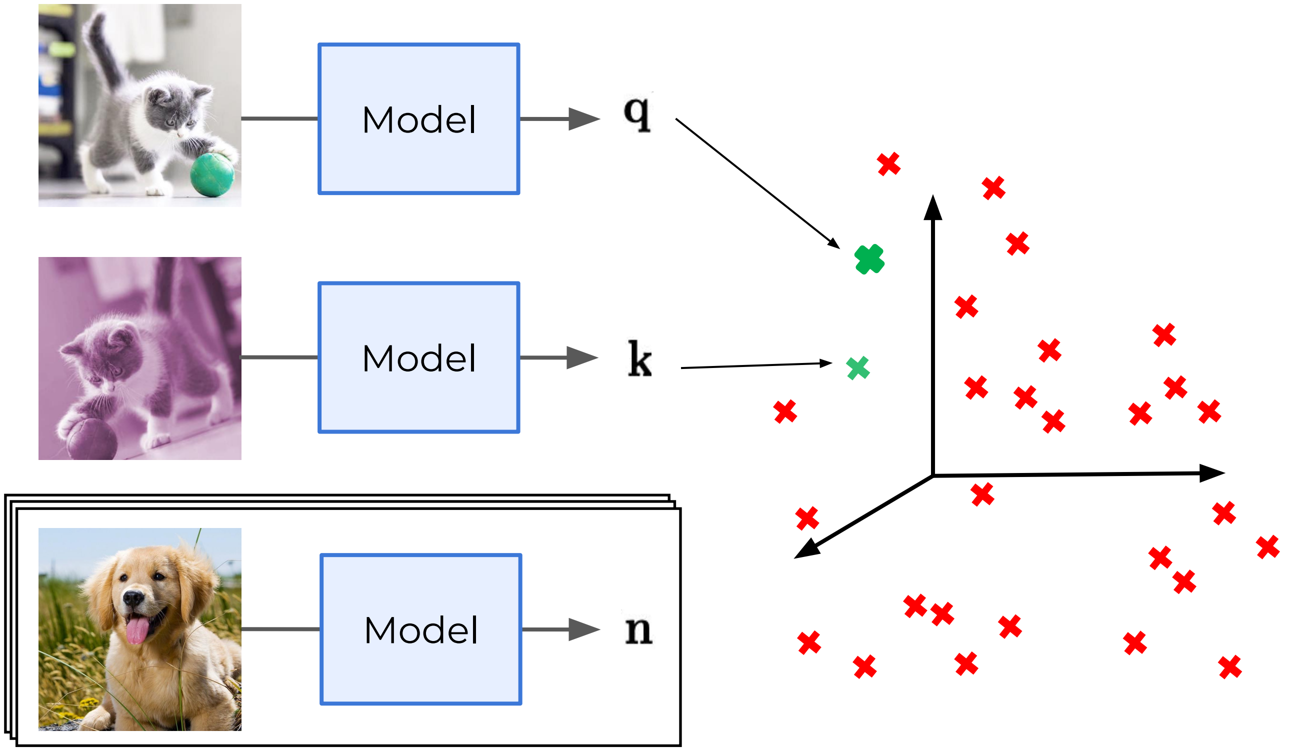
[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Contrastive self-supervised learning



[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.

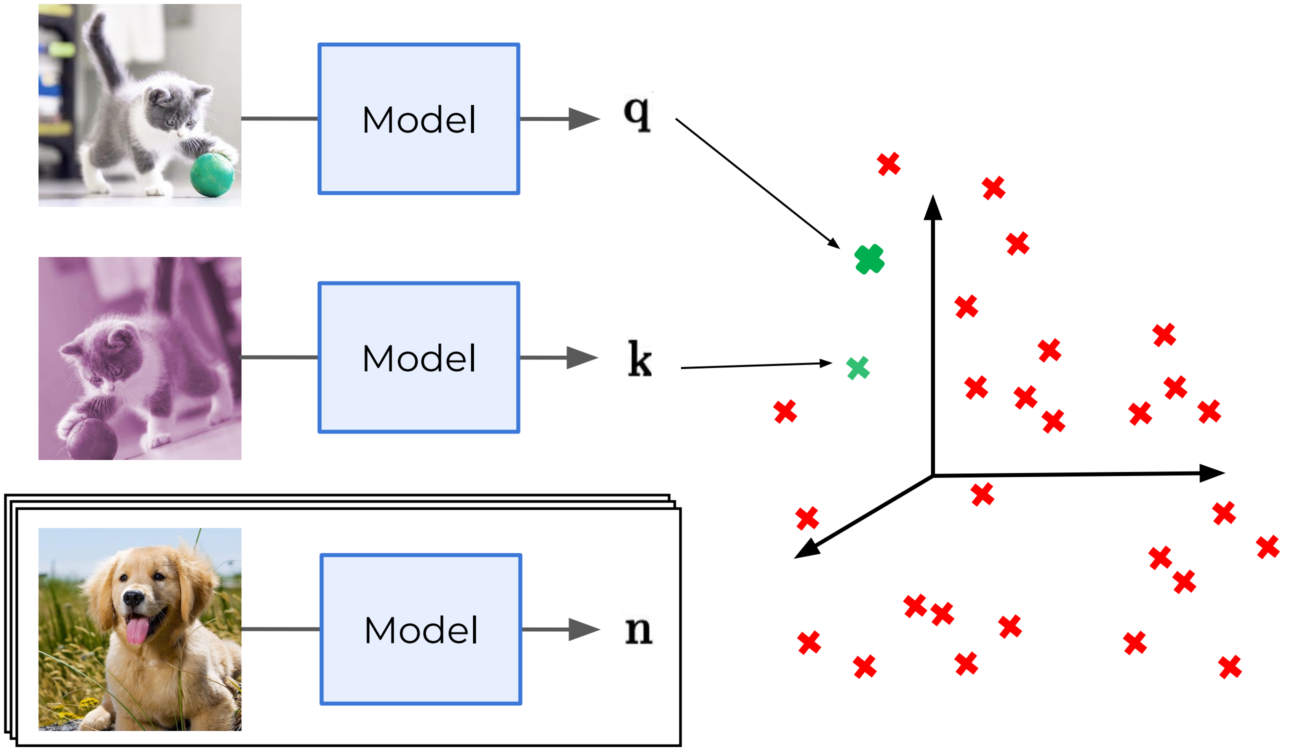
Contrastive self-supervised learning



SimCLR ***increases the batch size*** *to get more challenging negatives*

[SimCLR] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

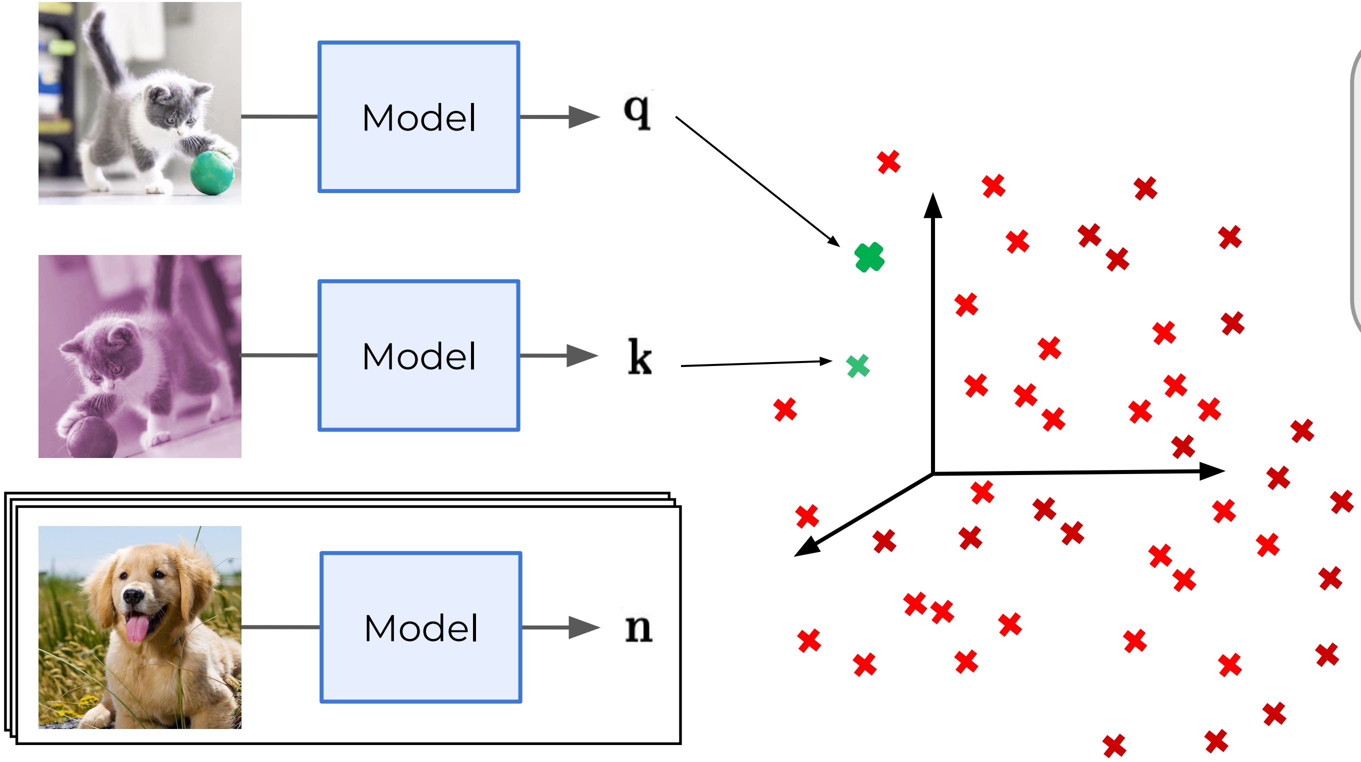
Contrastive self-supervised learning



MoCo ***increases the memory size*** *to get more challenging negatives*

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.

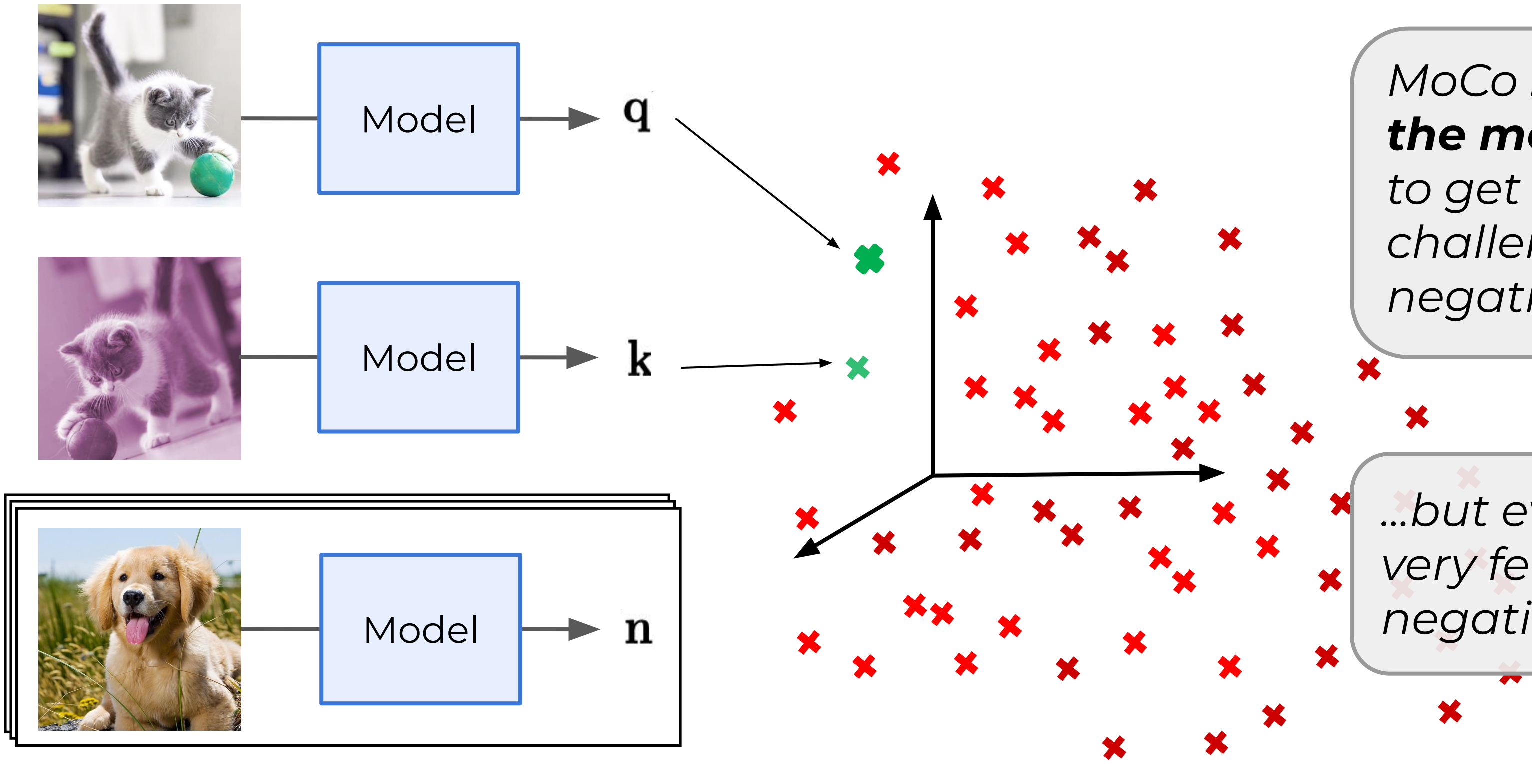
Contrastive self-supervised learning



*MoCo **increases the memory size** to get more challenging negatives*

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.

Contrastive self-supervised learning

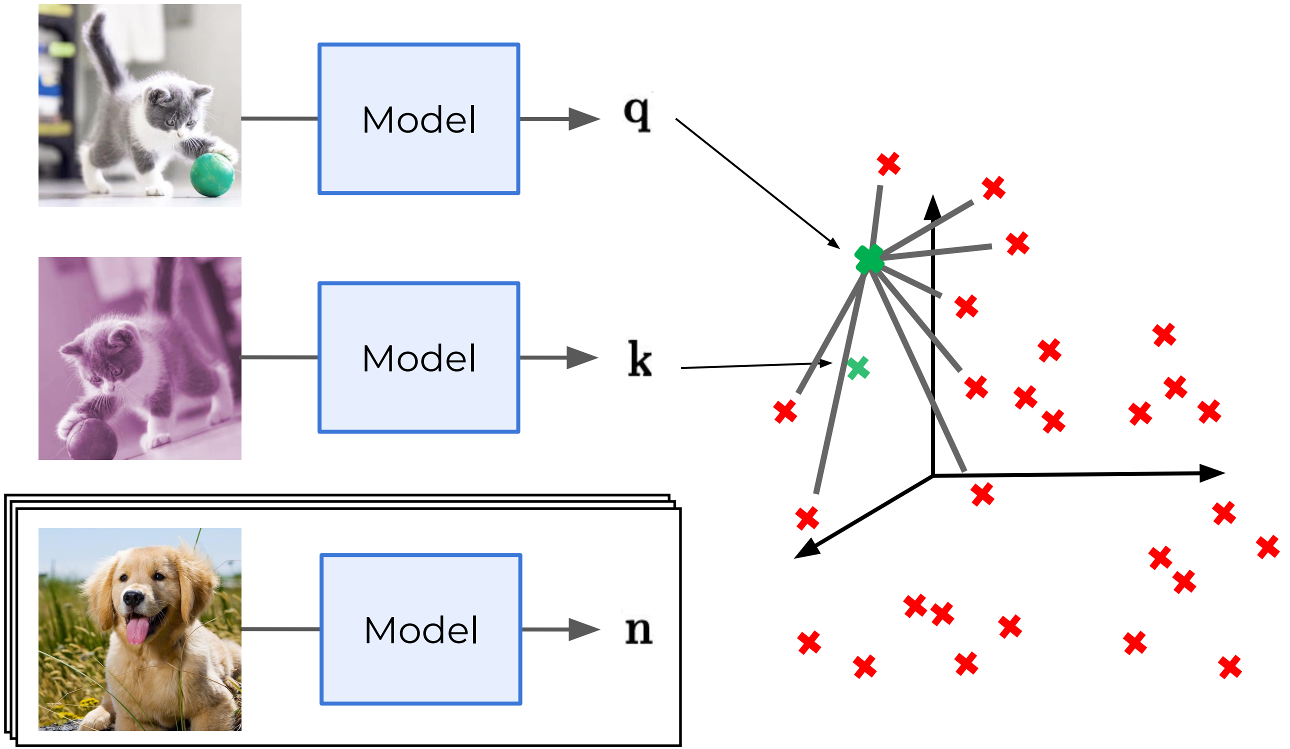


*MoCo **increases the memory size** to get more challenging negatives*

...but even then very few of the negatives are hard

[MoCo] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.

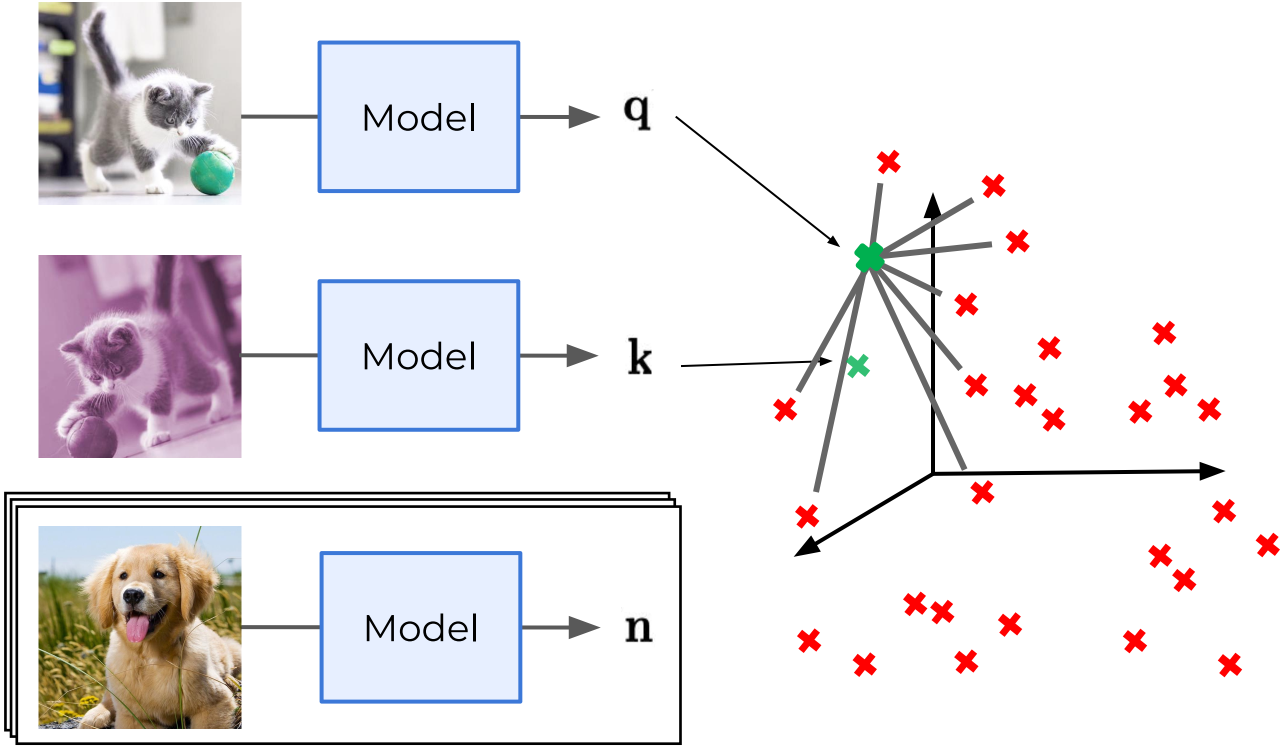
Contrastive self-supervised learning



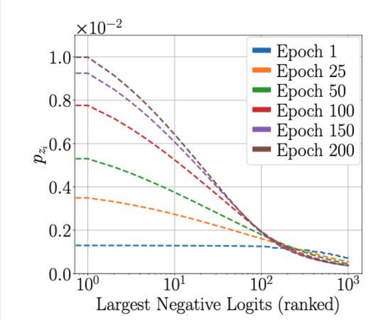
MoCo ***increases the memory size*** *to get more challenging negatives*

Yet, some hard negatives do exist in memory

Contrastive self-supervised learning



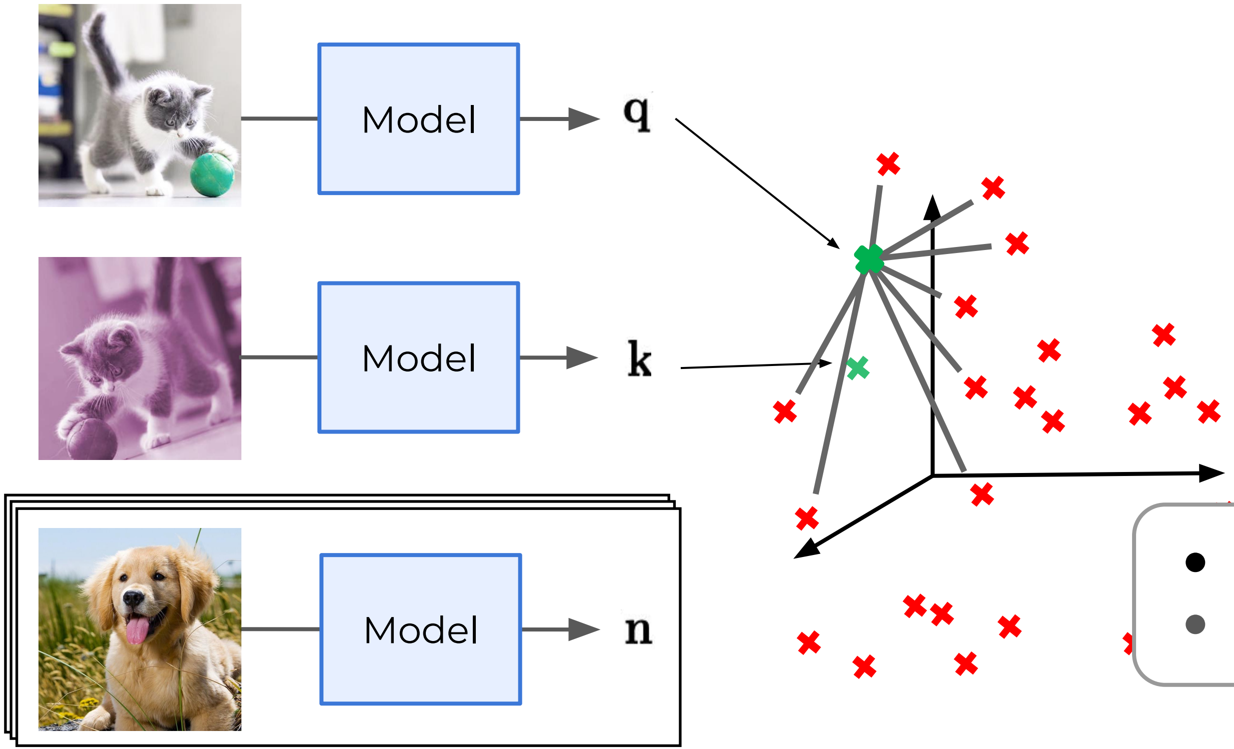
How hard are MoCo negatives?



Overview

- Introduction
- Contrastive self-supervised learning
- Hard Negative Mixing (MoCHi )
- Evaluation and results
- Understanding the feature space

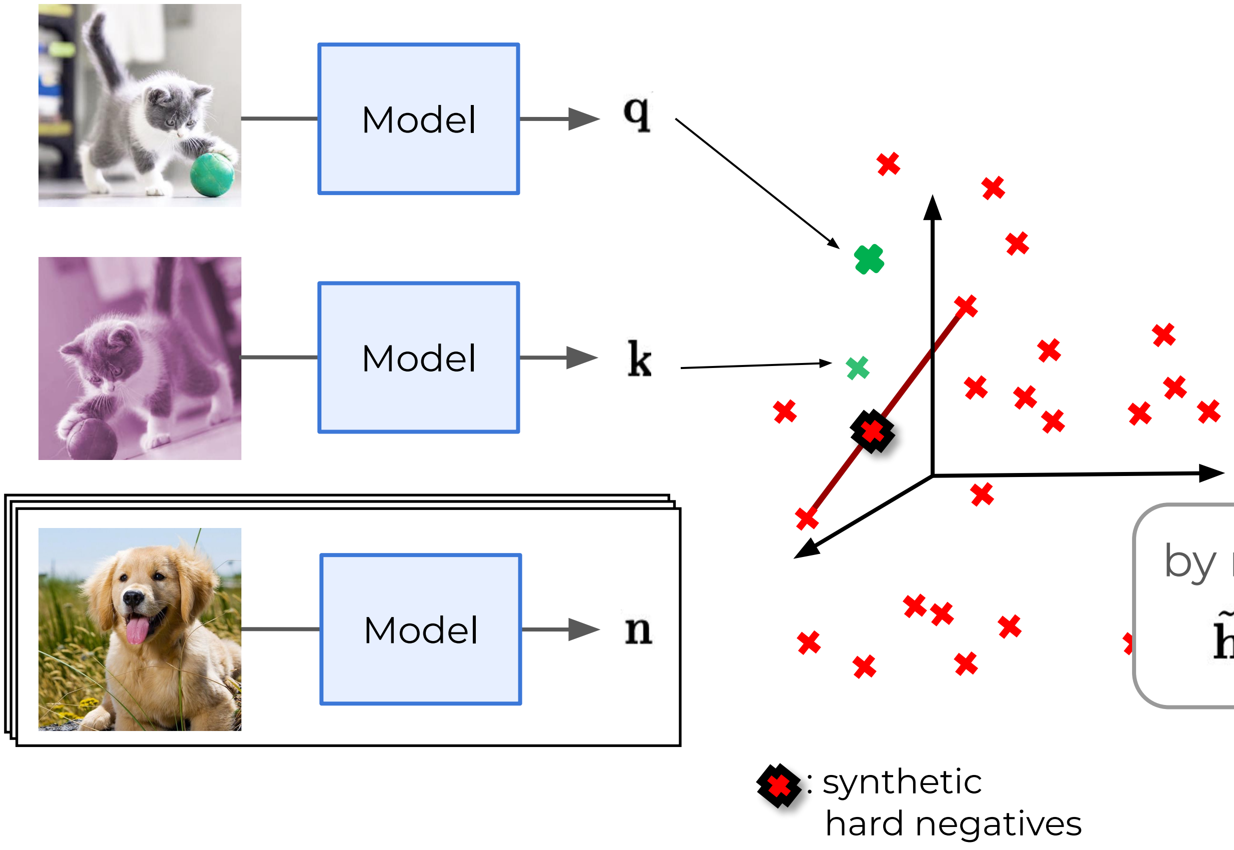
Mixing of Contrastive Hard Negatives



What if we mix the hardest negatives for each query and synthesize new hard negatives?

- on the fly for each query
- in feature space

Mixing of Contrastive Hard Negatives

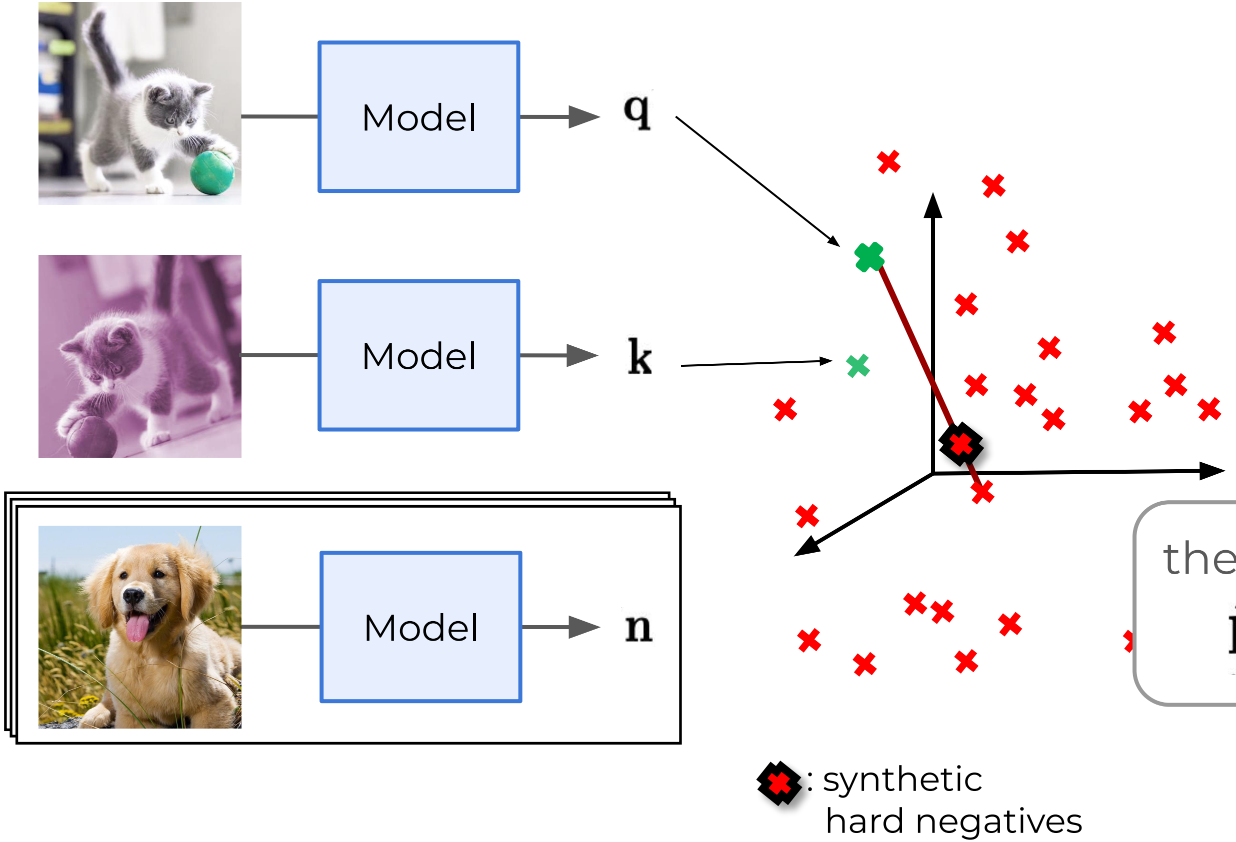


What if we mix the hardest negatives for each query and synthesize new hard negatives?

by mixing **two negatives**:

$$\tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1 - \alpha_k)\mathbf{n}_j$$

Mixing of Contrastive Hard Negatives



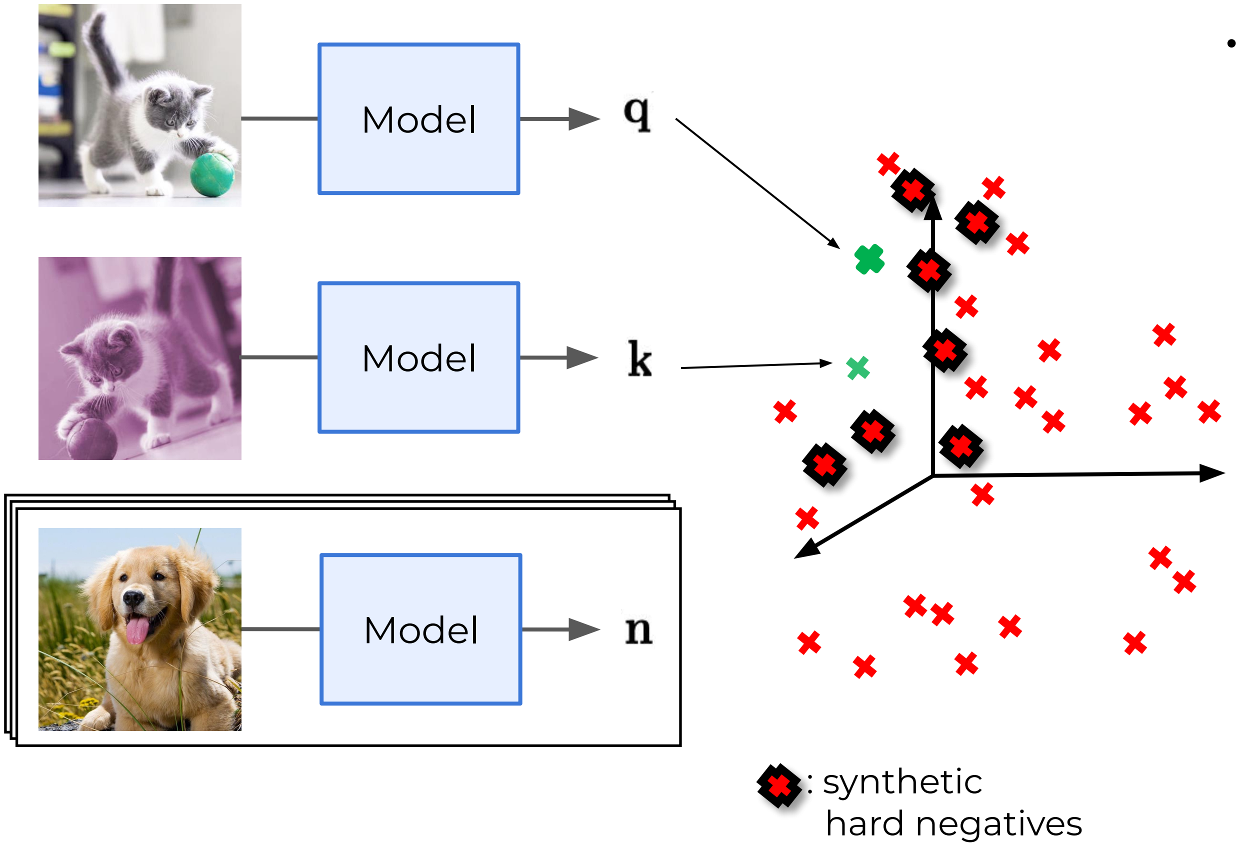
What if we mix the hardest negatives for each query and synthesize new hard negatives?

the **query** with a **negative**:

$$\tilde{\mathbf{h}}'_k = \beta_k \mathbf{q} + (1 - \beta_k) \mathbf{n}_j$$

Mixing of Contrastive Hard Negatives

...or MoCHi



What if we mix the hardest negatives for each query and synthesize new hard negatives?

Mixing of Contrastive Hard Negatives

...or MoCHi

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1 - \alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
  - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)

Mixing of Contrastive Hard Negatives

...or MoCHi

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1 - \alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
  - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

↓

How many of the hardest existing negatives to use?

[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)

Mixing of Contrastive Hard Negatives ...or MoCHi

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1 - \alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
  - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

How many points to synthesize by mixing two negatives?

[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)

Mixing of Contrastive Hard Negatives

...or MoChi

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1 - \alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
    - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

How many points to synthesize by mixing the query with a negative?

[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)

Overview

- Introduction
- Contrastive self-supervised learning
- Hard Negative Mixing (MoCHi )
- Evaluation and results
- Understanding the feature space

Experimental evaluation

Downstream tasks

Self-Supervised learning

Train on a proxy task
annotation-free:
ImageNet-1k after
discarding the labels

Model

Transfer Learning

Linear Classifiers

ImageNet-1k, ImageNet-100

Object Detection

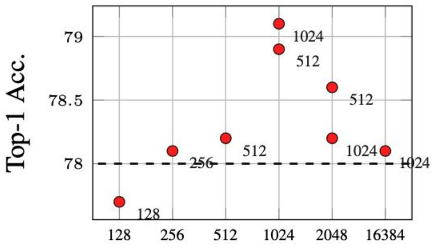
PASCAL VOC, MS COCO

Instance Segmentation

MS COCO

Results on ImageNet-100

- MoCHi increases performance for a large number of hyperparameter configurations
  - Varying number of synthetic features
  - Different ways of synthesizing
  - How many of the top negative to use



(a) Accuracy when varying N (x-axis) and s.

s \ s'	0	128	256	512
0	0.0	+0.7	+0.9	+1.0
128	+0.8	+0.4	+1.1	+0.3
256	+0.3	+0.7	+0.3	+1.0
512	+0.9	+0.8	+0.6	+0.4
1024	+0.8	+1.0	+0.7	+0.6

(b) Accuracy gains over MoCo-v2 when $N = 1024$.

Method	Top1 % ($\pm\sigma$)	diff (%)
MoCo [21]	73.4	
MoCo + iMix [36]	74.2‡	0.8
CMC [38]	75.7	
CMC + iMix [36]	75.9‡	0.2
MoCo [21]* ($t = 0.07$)	74.0	
MoCo [21]* ($t = 0.2$)	75.9	
MoCo-v2 [10]*	78.0 ($\pm$0.2)	
+ MoCHi (1024, 1024, 128)	**79.0** ($\pm$0.4)	**1.0**
+ MoCHi (1024, 256, 512)	**79.0** ($\pm$0.4)	**1.0**
+ MoCHi (1024, 128, 256)	**78.9** ($\pm$0.5)	**0.9**

Linear classification accuracy (ImageNet-100)

Results on ImageNet-1k and PASCAL VOC

Method	IN-1k Top1	VOC 2007 AP_{50}	AP	AP_{75}
		100 epoch training		
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
		200 epoch training		
SimCLR [8] (8k batch size, from [10])	66.6			
MoCo + Image Mixture [36]	60.8	76.4		
InstDis [46]†	59.5	80.9	55.2	61.2
MoCo [21]	60.6	81.5	55.9	62.6
PIRL [31]†	61.7	81.0	55.5	61.3
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	**82.8** (±0.2) (0.3)	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
+ MoCHi (256, 512, 256)	67.6	82.6 (±0.2) (0.1)	57.2 (±0.3) (0.4)	64.2 (±0.5) (0.9)
+ MoCHi (256, 2048, 2048)	*67.0*	82.5 (±0.1) (0.0)	57.1 (±0.2) (0.3)	64.4 (±0.2) (1.1)
+ MoCHi (128, 1024, 512)	*66.9*	82.7 (±0.2) (0.2)	57.5 (±0.3) (0.7)	64.4 (±0.4) (1.1)
		800 epoch training		
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	**83.3** (±0.1) (0.6)	57.3 (±0.2) (0.5)	**64.2** (±0.4) (0.3)
Supervised [21]	76.1	81.3	53.5	58.8

Results on ImageNet-1k and PASCAL VOC

Linear classification on ImageNet:

MoCHi does not show performance gains over MoCo-v2

<u>Possible explanation:</u> biases induced by training with hard negatives on the *same dataset as the downstream task*

➢ MoCHi retains state-of-the-art performance for linear classification on ImageNet

Method	IN-1k Top1	VOC 2007 AP$_{50}$	VOC 2007 AP	VOC 2007 AP$_{75}$
100 epoch training				
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
200 epoch training				
SimCLR [8] (8k batch size, from [10])	66.6			
MoCo + Image Mixture [36]	60.8	76.4		
InstDis [46]†	59.5	80.9	55.2	61.2
MoCo [21]	60.6	81.5	55.9	62.6
PIRL [31]†	61.7	81.0	55.5	61.3
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	<u>**82.8**</u> (±0.2) <u>(0.3)</u>	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
+ MoCHi (256, 512, 256)	67.6	82.6 (±0.2) (0.1)	57.2 (±0.3) (0.4)	64.2 (±0.5) (0.9)
+ MoCHi (256, 2048, 2048)	*67.0*	82.5 (±0.1) (0.0)	57.1 (±0.2) (0.3)	<u>64.4</u> (±0.2) <u>(1.1)</u>
+ MoCHi (128, 1024, 512)	*66.9*	82.7 (±0.2) (0.2)	<u>57.5</u> (±0.3) <u>(0.7)</u>	<u>64.4</u> (±0.4) <u>(1.1)</u>
800 epoch training				
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	<u>**83.3**</u> (±0.1) (0.6)	<u>57.3</u> (±0.2) <u>(0.5)</u>	<u>**64.2**</u> (±0.4) <u>(0.3)</u>
Supervised [21]	76.1	81.3	53.5	58.8

Results on ImageNet-1k and PASCAL VOC

Transfer learning performance:

MoCHi helps the model learn faster:

- ➢ Strong performance gains on PASCAL VOC when using a model with only 100 epochs of pre-training

Method	IN-1k Top1	VOC 2007 AP$_{50}$	AP	AP$_{75}$
100 epoch training				
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
200 epoch training				
SimCLR [8] (8k batch size, from [10])	66.6			
MoCo + Image Mixture [36]	60.8	76.4		
InstDis [46]†	59.5	80.9	55.2	61.2
MoCo [21]	60.6	81.5	55.9	62.6
PIRL [31]†	61.7	81.0	55.5	61.3
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	<u>**82.8**</u> (±0.2) <u>(0.3)</u>	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
+ MoCHi (256, 512, 256)	67.6	82.6 (±0.2) (0.1)	57.2 (±0.3) (0.4)	64.2 (±0.5) (0.9)
+ MoCHi (256, 2048, 2048)	*67.0*	82.5 (±0.1) (0.0)	57.1 (±0.2) (0.3)	<u>64.4</u> (±0.2) <u>(1.1)</u>
+ MoCHi (128, 1024, 512)	*66.9*	82.7 (±0.2) (0.2)	<u>57.5</u> (±0.3) <u>(0.7)</u>	<u>64.4</u> (±0.4) <u>(1.1)</u>
800 epoch training				
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	<u>**83.3**</u> (±0.1) (0.6)	<u>57.3</u> (±0.2) <u>(0.5)</u>	<u>**64.2**</u> (±0.4) <u>(0.3)</u>
Supervised [21]	76.1	81.3	53.5	58.8

Results on ImageNet-1k and PASCAL VOC

Transfer learning performance:

- ➢ MoCHi after 200 epochs performs similar to MoCo-v2 after 800 epochs
- ➢ Performance gains are consistent across multiple hyperparameter configurations

Method	IN-1k	VOC 2007		
	Top1	AP_{50}	AP	AP_{75}
100 epoch training				
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
200 epoch training				
SimCLR [8] (8k batch size, from [10])	66.6			
MoCo + Image Mixture [36]	60.8	76.4		
InstDis [46]†	59.5	80.9	55.2	61.2
MoCo [21]	60.6	81.5	55.9	62.6
PIRL [31]†	61.7	81.0	55.5	61.3
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	**82.8** (±0.2) (0.3)	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
+ MoCHi (256, 512, 256)	67.6	82.6 (±0.2) (0.1)	57.2 (±0.3) (0.4)	64.2 (±0.5) (0.9)
+ MoCHi (256, 2048, 2048)	*67.0*	82.5 (±0.1) (0.0)	57.1 (±0.2) (0.3)	64.4 (±0.2) (1.1)
+ MoCHi (128, 1024, 512)	*66.9*	82.7 (±0.2) (0.2)	57.5 (±0.3) (0.7)	64.4 (±0.4) (1.1)
800 epoch training				
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	**83.3** (±0.1) (0.6)	57.3 (±0.2) (0.5)	**64.2** (±0.4) (0.3)
Supervised [21]	76.1	81.3	53.5	58.8

Results on ImageNet-1k and PASCAL VOC

Transfer learning performance:

➢ Gains persist after longer training (800 epochs)

Method	IN-1k Top1	VOC 2007 AP_{50}	AP	AP_{75}
100 epoch training				
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
200 epoch training				
SimCLR [8] (8k batch size, from [10])	66.6			
MoCo + Image Mixture [36]	60.8	76.4		
InstDis [46]†	59.5	80.9	55.2	61.2
MoCo [21]	60.6	81.5	55.9	62.6
PIRL [31]†	61.7	81.0	55.5	61.3
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	<u>**82.8**</u> (±0.2) <u>(0.3)</u>	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
+ MoCHi (256, 512, 256)	67.6	82.6 (±0.2) (0.1)	57.2 (±0.3) (0.4)	64.2 (±0.5) (0.9)
+ MoCHi (256, 2048, 2048)	*67.0*	82.5 (±0.1) (0.0)	57.1 (±0.2) (0.3)	<u>64.4</u> (±0.2) <u>(1.1)</u>
+ MoCHi (128, 1024, 512)	*66.9*	82.7 (±0.2) (0.2)	<u>57.5</u> (±0.3) <u>(0.7)</u>	<u>64.4</u> (±0.4) <u>(1.1)</u>
800 epoch training				
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	<u>**83.3**</u> (±0.1) (0.6)	<u>57.3</u> (±0.2) <u>(0.5)</u>	<u>**64.2**</u> (±0.4) <u>(0.3)</u>
Supervised [21]	76.1	81.3	53.5	58.8

Results on ImageNet-1k and PASCAL VOC

Transfer learning performance:

- ➢ Gains persist after longer training (800 epochs)
- ➢ Large gains (<u>4% AP</u>) for self-supervised pre-training versus the "traditional" (supervised) ImageNet

Method	IN-1k Top1	VOC 2007 AP_{50}	AP	AP_{75}
100 epoch training				
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
200 epoch training				
SimCLR [8] (8k batch size, from [10])	66.6			
MoCo + Image Mixture [36]	60.8	76.4		
InstDis [46]†	59.5	80.9	55.2	61.2
MoCo [21]	60.6	81.5	55.9	62.6
PIRL [31]†	61.7	81.0	55.5	61.3
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	<u>**82.8**</u> (±0.2) <u>(0.3)</u>	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
+ MoCHi (256, 512, 256)	67.6	82.6 (±0.2) (0.1)	57.2 (±0.3) (0.4)	64.2 (±0.5) (0.9)
+ MoCHi (256, 2048, 2048)	*67.0*	82.5 (±0.1) (0.0)	57.1 (±0.2) (0.3)	<u>64.4</u> (±0.2) <u>(1.1)</u>
+ MoCHi (128, 1024, 512)	*66.9*	82.7 (±0.2) (0.2)	<u>57.5</u> (±0.3) <u>(0.7)</u>	<u>64.4</u> (±0.4) <u>(1.1)</u>
800 epoch training				
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	<u>**83.3**</u> (±0.1) (0.6)	<u>57.3</u> (±0.2) <u>(0.5)</u>	<u>**64.2**</u> (±0.4) <u>(0.3)</u>
Supervised [21]	76.1	81.3	53.5	58.8

Results on COCO

	Object Detection			Instance Segmentation		
Pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Supervised [13]	38.2	58.2	41.6	33.3	54.7	35.2
100 epoch pre-training						
MoCo-v2 [6]	37.0 ($\pm$0.1)	56.5 ($\pm$0.3)	39.8 ($\pm$0.1)	32.7 ($\pm$0.1)	53.3 ($\pm$0.2)	34.3 ($\pm$0.1)
+ MoCHi (256, 512, 0)	37.5 ($\pm$0.1) ($\uparrow$0.5)	57.0 ($\pm$0.1) ($\uparrow$0.5)	40.5 ($\pm$0.2) ($\uparrow$0.7)	33.0 ($\pm$0.1) ($\uparrow$0.3)	53.9 ($\pm$0.2) ($\uparrow$0.6)	34.9 ($\pm$0.1) ($\uparrow$0.6)
+ MoCHi (128, 1024, 512)	**37.8** ($\pm$0.1) ($\uparrow$**0.8**)	**57.2** ($\pm$0.0) ($\uparrow$**0.7**)	**40.8** ($\pm$0.2) ($\uparrow$**1.0**)	**33.2** ($\pm$0.0) ($\uparrow$**0.5**)	54.0 ($\pm$0.2) ($\uparrow$**0.7**)	35.4 ($\pm$0.1) ($\uparrow$**1.1**)
200 epoch pre-training						
MoCo [13]	38.5	58.3	41.6	33.6	54.8	35.6
MoCo (1B image train) [13]	39.1	58.7	42.2	34.1	55.4	36.4
InfoMin Aug. [28]	39.0	58.5	42.0	34.1	55.2	36.3
MoCo-v2 [6]	39.0 ($\pm$0.1)	58.6 ($\pm$0.1)	41.9($\pm$0.3)	34.2 ($\pm$0.1)	55.4 ($\pm$0.1)	36.2 ($\pm$0.2)
+ MoCHi (256, 512, 0)	39.2 ($\pm$0.1) ($\uparrow$0.2)	58.8 ($\pm$0.1) ($\uparrow$0.2)	42.4 ($\pm$0.2) ($\uparrow$0.5)	34.4 ($\pm$0.1) ($\uparrow$0.2)	55.6 ($\pm$0.1) ($\uparrow$0.2)	36.7 ($\pm$0.1) ($\uparrow$0.5)
+ MoCHi (128, 1024, 512)	39.2 ($\pm$0.1) ($\uparrow$0.2)	58.9 ($\pm$0.2) ($\uparrow$0.3)	42.4 ($\pm$0.3) ($\uparrow$0.5)	34.3 ($\pm$0.1) ($\uparrow$0.2)	55.5 ($\pm$0.1) ($\uparrow$0.1)	36.6 ($\pm$0.1) ($\uparrow$0.4)
+ MoCHi (512, 1024, 512)	**39.4** ($\pm$0.1) ($\uparrow$**0.4**)	**59.0** ($\pm$0.1) ($\uparrow$**0.4**)	**42.7** ($\pm$0.1) ($\uparrow$**0.8**)	**34.5** ($\pm$0.0) ($\uparrow$**0.3**)	**55.7** ($\pm$0.2) ($\uparrow$**0.3**)	**36.7** ($\pm$0.1) ($\uparrow$**0.5**)

Gains also consistent on COCO:

- Instance segmentation: Match supervised pre-training perf. after 100 epochs

Results on COCO

	Object Detection			Instance Segmentation		
Pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Supervised [13]	38.2	58.2	41.6	33.3	54.7	35.2
			100 epoch pre-training			
MoCo-v2 [6]	37.0 (±0.1)	56.5 (±0.3)	39.8 (±0.1)	32.7 (±0.1)	53.3 (±0.2)	34.3 (±0.1)
+ MoCHi (256, 512, 0)	37.5 (±0.1) (↑0.5)	57.0 (±0.1) (↑0.5)	40.5 (±0.2) (↑0.7)	33.0 (±0.1) (↑0.3)	53.9 (±0.2) (↑0.6)	34.9 (±0.1) (↑0.6)
+ MoCHi (128, 1024, 512)	**37.8** (±0.1) (↑**0.8**)	**57.2** (±0.0) (↑**0.7**)	**40.8** (±0.2) (↑**1.0**)	**33.2** (±0.0) (↑**0.5**)	54.0 (±0.2) (↑**0.7**)	35.4 (±0.1) (↑**1.1**)
			200 epoch pre-training			
MoCo [13]	38.5	58.3	41.6	33.6	54.8	35.6
MoCo (1B image train) [13]	39.1	58.7	42.2	34.1	55.4	36.4
InfoMin Aug. [28]	39.0	58.5	42.0	34.1	55.2	36.3
MoCo-v2 [6]	39.0 (±0.1)	58.6 (±0.1)	41.9(±0.3)	34.2 (±0.1)	55.4 (±0.1)	36.2 (±0.2)
+ MoCHi (256, 512, 0)	39.2 (±0.1) (↑0.2)	58.8 (±0.1) (↑0.2)	42.4 (±0.2) (↑0.5)	34.4 (±0.1) (↑0.2)	55.6 (±0.1) (↑0.2)	36.7 (±0.1) (↑0.5)
+ MoCHi (128, 1024, 512)	39.2 (±0.1) (↑0.2)	58.9 (±0.2) (↑0.3)	42.4 (±0.3) (↑0.5)	34.3 (±0.1) (↑0.2)	55.5 (±0.1) (↑0.1)	36.6 (±0.1) (↑0.4)
+ MoCHi (512, 1024, 512)	**39.4** (±0.1) (↑**0.4**)	**59.0** (±0.1) (↑**0.4**)	**42.7** (±0.1) (↑**0.8**)	**34.5** (±0.0) (↑**0.3**)	**55.7** (±0.2) (↑**0.3**)	**36.7** (±0.1) (↑**0.5**)

Gains also consistent on COCO:

- Instance segmentation: Match supervised pre-training perf. after 100 epochs
- Outperform the recent SoTA [InfoMin Aug] (better positives)

Results summary

- Linear classification on ImageNet
  - Retains [MoCo-v2]'s SoTA performance
  - MoCHi does not increase, maybe slightly hurts performance
- Transfer learning to other tasks (after fine-tuning)
  - Gains and SoTA performance on PASCAL VOC/COCO
- Faster learning
  - +1% AP over MoCo-v2 on PASCAL VOC when pre-training for 100 epochs
  - Match supervised pre-training performance after 100 epochs on COCO

Results summary

- Linear classification on ImageNet
  - Retains [MoCo-v2]'s SoTA performance
  - MoCHi does not increase, maybe slightly hurts performance
- Transfer learning to other tasks (after fine-tuning)
  - Gains and SoTA performance on PASCAL VOC/COCO
- Faster learning
  - +1% AP over MoCo-v2 on PASCAL VOC when pre-training for 100 epochs
  - Match supervised pre-training performance after 100 epochs on COCO

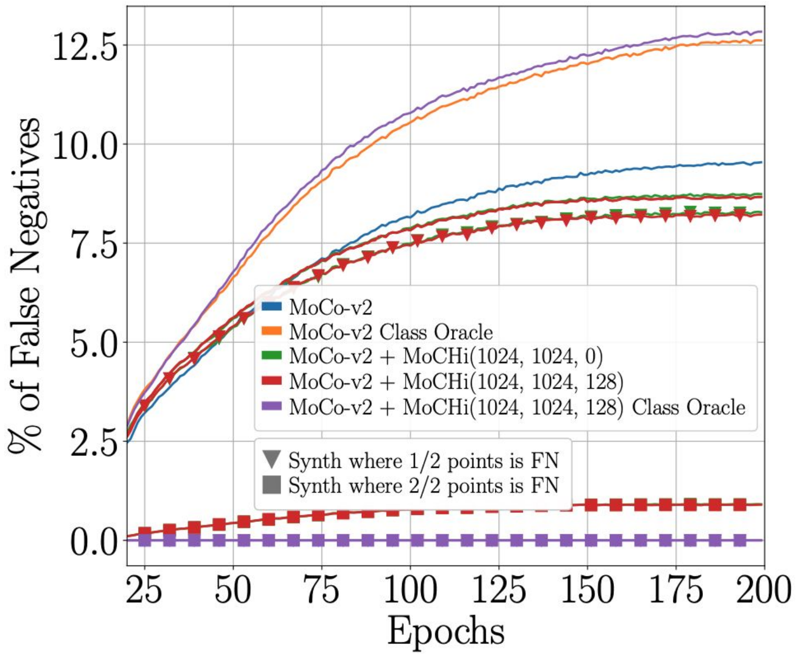
Can we better understand why MoCHi doesn't help with linear classification but performs better for downstream tasks?

Overview

- Introduction
- Contrastive self-supervised learning
- Hard Negative Mixing (MoCHi)
- Evaluation and results
- Understanding the feature space

Analysis using a class label “oracle”

We are training on ImageNet-1K...
...let’s look at the class labels!



False Negatives (FN): Use ImageNet labels to measure memory/negative items that are:

- from the same class as the **q**
- Highly rank wrt logits, *i.e.* in the top-1024 highest logits for **q**

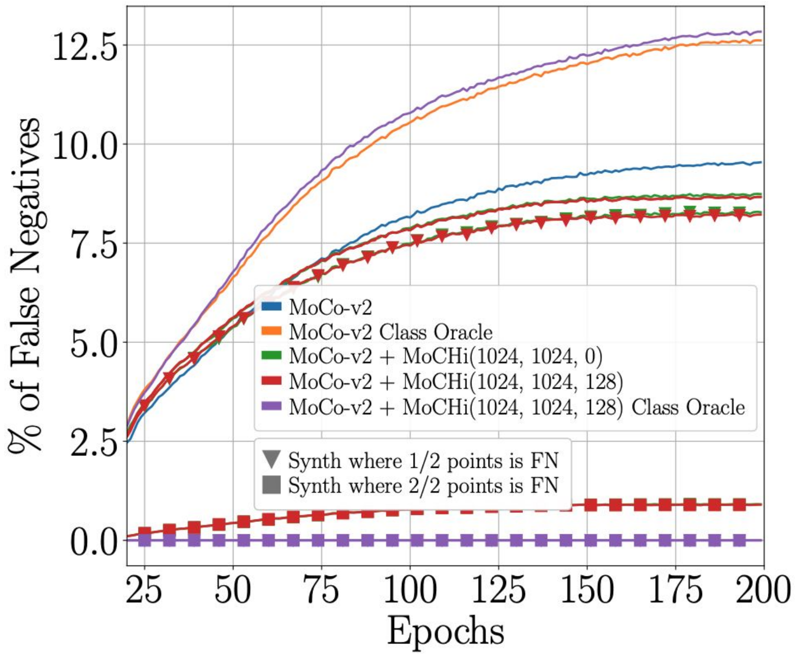
Understanding synthetic negatives

False Negatives (**FN**) are the negatives that are:

- From the same class as the query
- Highly ranked wrt their similarity to the query

Let's first look at the synthetic points:

- *How many of the synthetic points are (definitely) false negatives?*



Understanding synthetic negatives

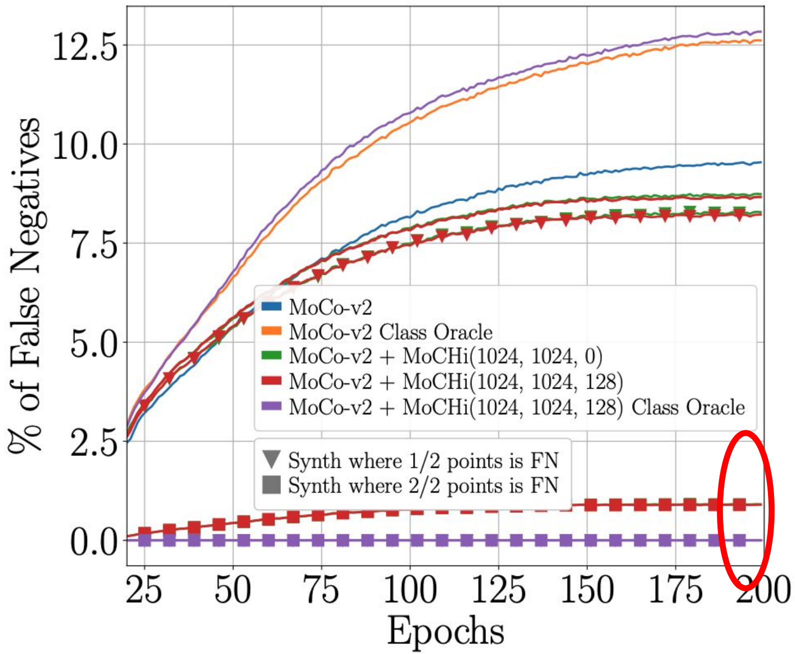
False Negatives (**FN**) are the negatives that are:

- From the same class as the query
- Highly ranked wrt their similarity to the query

Let's first look at the synthetic points:

- *How many of the synthetic points are (definitely) false negatives?*

- Only a small percentage of the points synthesized with MoCHi are definitely **FN**

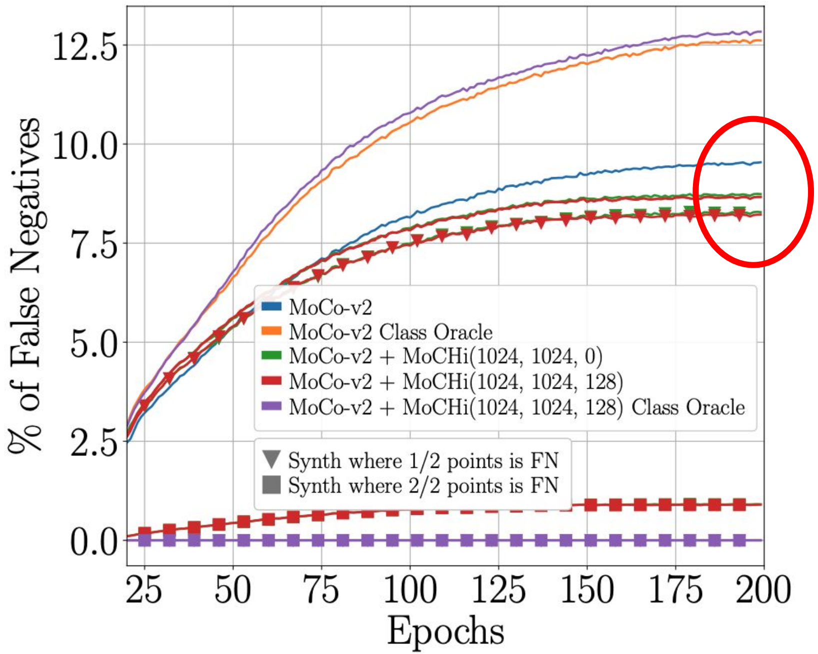


Understanding synthetic negatives

False Negatives (**FN**) are the negatives that are:

- From the same class as the query
- Highly ranked wrt their similarity to the query

But how about the “real” negatives?



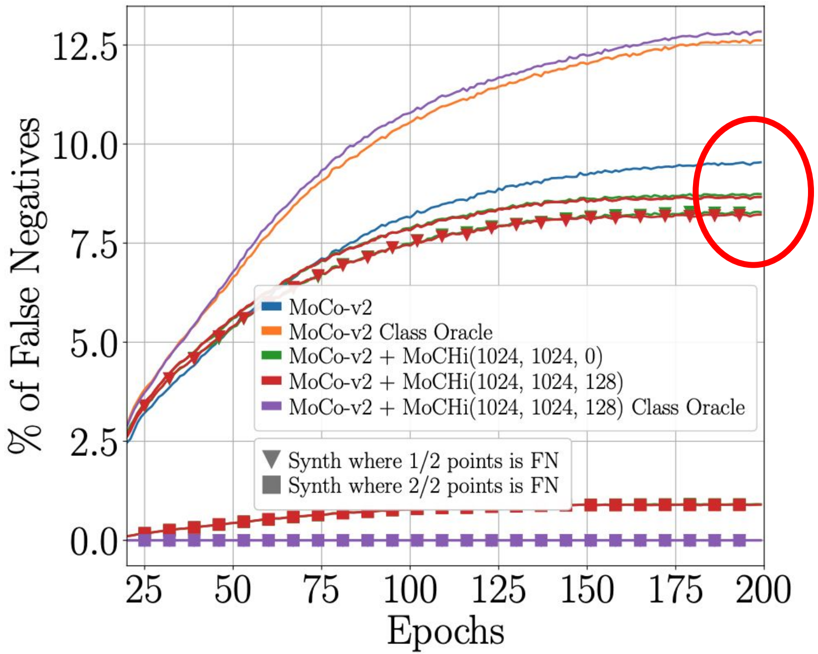
Understanding synthetic negatives

False Negatives (**FN**) are the negatives that are:

- From the same class as the query
- Highly ranked wrt their similarity to the query

But how about the "real" negatives?

- **FN** in the top-k increase with training
- desirable (we are learning a space where features from the same class are closer together)



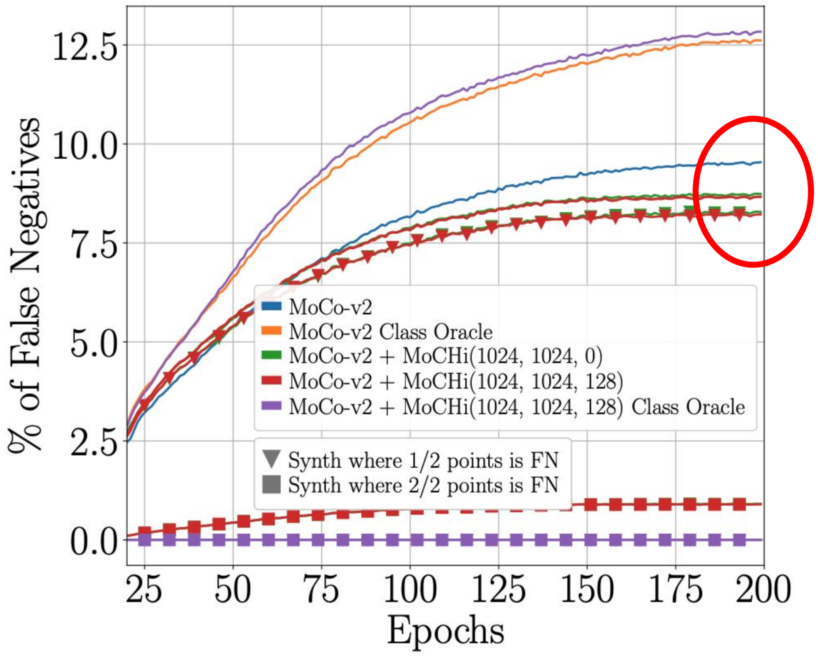
Understanding synthetic negatives

<u>False Negatives</u> (**FN**) are the negatives that are:

- From the same class as the query
- Highly ranked wrt their similarity to the query

But how about the “real” negatives?

- MoCHi has overall a smaller percentage of false negatives!



Understanding synthetic negatives

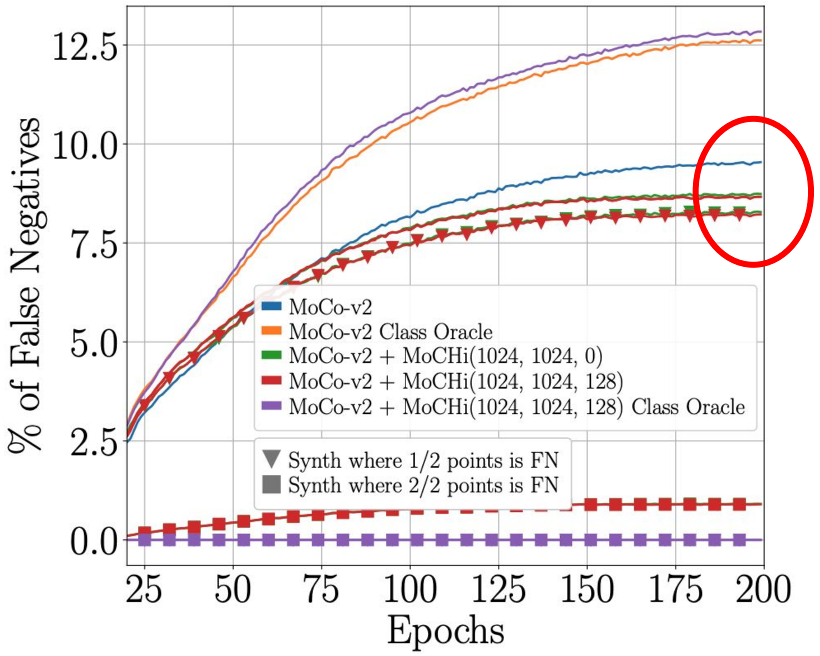
False Negatives (**FN**) are the negatives that are:

- From the same class as the query
- Highly ranked wrt their similarity to the query

But how about the "real" negatives?

- MoCHi has overall a smaller percentage of false negatives!

  ... i.e. MoCo does a better job at bringing embeddings from the same class (in the training set) closer together



Understanding synthetic negatives

<u>False Negatives</u> (**FN**) are the negatives that are:

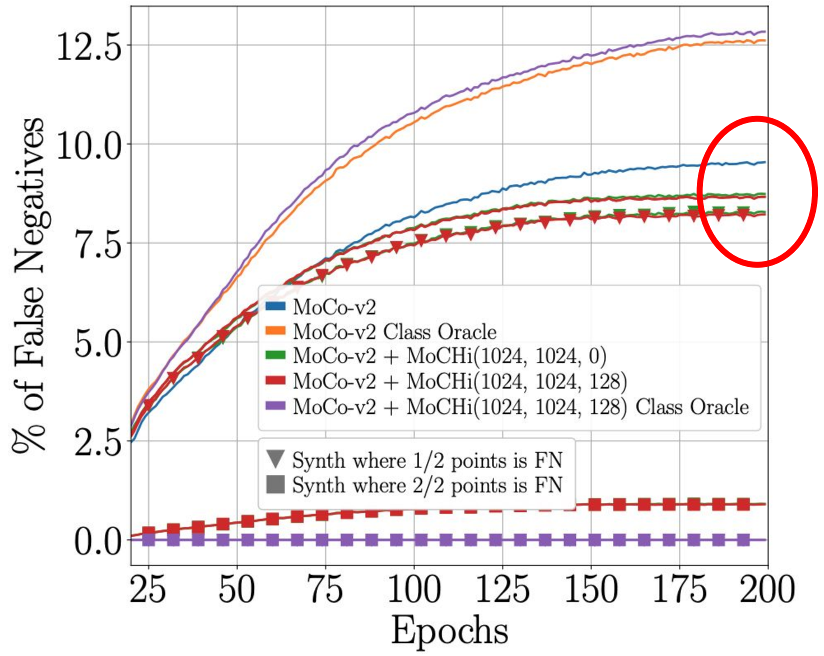
- From the same class as the query
- Highly ranked wrt their similarity to the query

But how about the “real” negatives?

- MoCHi has overall a smaller percentage of false negatives!

  ... i.e. MoCo does a better job at bringing embeddings from the same class (in the training set) closer together

  Why does MoCHi perform better for downstream tasks?



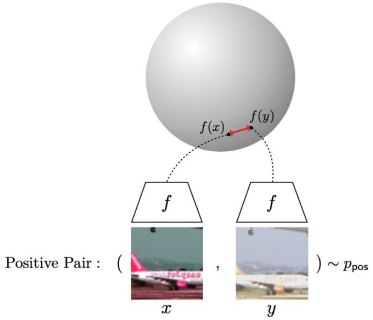
Uniformity and alignment scores [Wang & Isola]

Alignment

- Average distance between representations with the same class

Uniformity

- Average pairwise distance between all embeddings



Alignment: Similar samples have similar features.

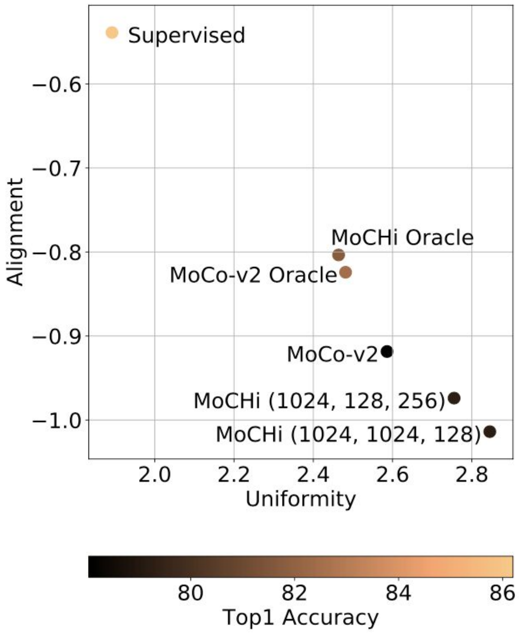
Uniformity: Preserve maximal information.

[Wang & Isola] Wang, Tongzhou, and Phillip Isola. "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere." ICML 2020.

Uniformity and alignment scores [Wang & Isola]

Alignment

Supervised > MoCo > MoCHi

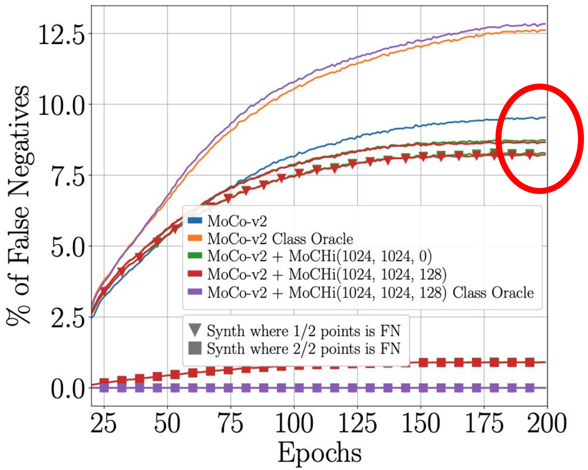


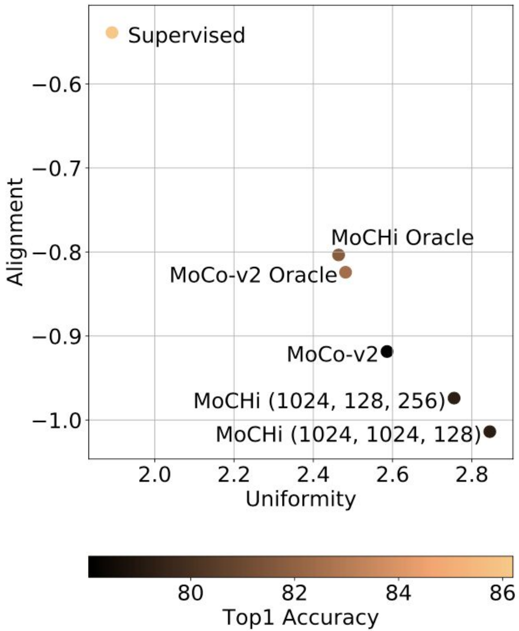
Uniformity and alignment scores [Wang & Isola]

Alignment

Supervised > MoCo > MoCHi

This result confirms the plot

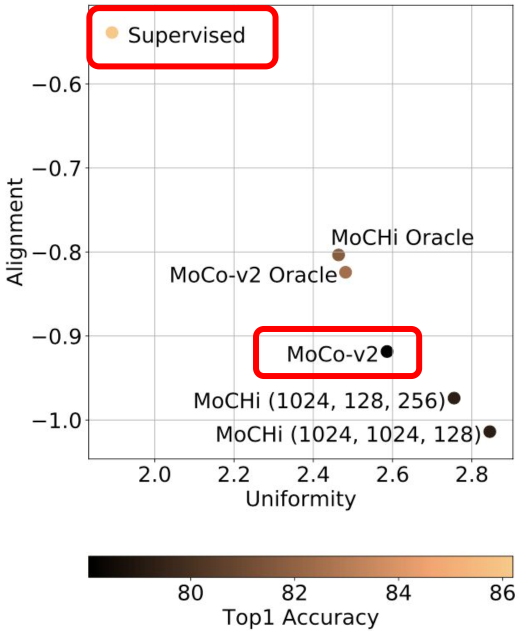




Uniformity

Utilization of the embedding space

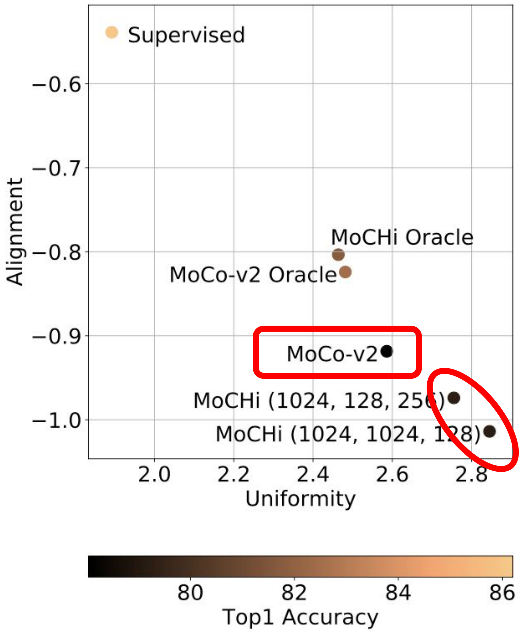
- Contrastive SSL (MoCo) utilizes the embedding space “more” than training with Cross Entropy (supervised)



Uniformity

Utilization of the embedding space

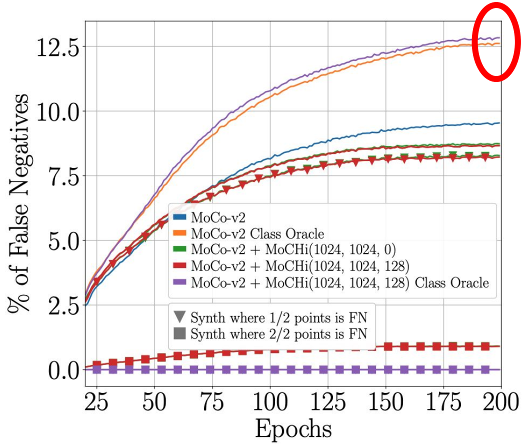
- Contrastive SSL (MoCo) utilizes the embedding space “more” than training with Cross Entropy (supervised)
- Adding synthetic hard negative (MoCHi) results in utilizing the space even more!

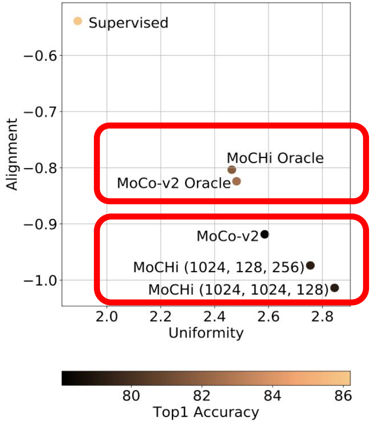


“Oracle” runs

What if we didn’t have **FN**?

- Upper bound: simply discard images with the same label as the query from the negatives
- Oracle runs show:
  - higher percentage of FN
  - higher alignment score





“Oracle” runs

What if we didn’t have **FN**?

- Upper bound: simply discard images with the same label as the query from the negatives
- Oracle runs show:
  - higher percentage of FN
  - higher alignment score
- Performance:
  - Closing the gap with supervised

Linear classification accuracy (ImageNet-100)

	Using Class Oracle
MoCo-v2* (200 epochs)	81.8
+ MoCHi (1024, 1024, 128) (200 epochs)	82.5
+ MoCHi (1024, 1024, 128) (400 epochs)	84.2
+ MoCHi (1024, 1024, 128) (800 epochs)	85.2
Cross-entropy classification (supervised)	86.2

	Acc	AP-50	AP	AP-75
	Using Class Oracle			
Cross-entropy classification (supervised)	76.1	81.3	53.5	58.8
MoCo-v2 [10] + MoCHi (512, 1024, 512)	72.6	*83.3*	*57.7*	*64.6*
	ImageNet-1K	PASCAL VOC		

see also: Khosla, Prannay, et al. "Supervised contrastive learning." NeurIPS 2020.

Take home message

MoCHi

(NeurIPS 2020)

- A more challenging proxy task
- Consistent gains over a state-of-the-art method [MoCo-v2]
- Faster learning
  - +1% AP over MoCo-v2 on PASCAL VOC when pre-training for 100 epochs
  - Match supervised pre-training performance after 100 epochs on COCO
- Better utilization of the embedding space
  - Measured via the Uniformity metric [Wang and Isola]
- Project page with pre-trained models:

https://europe.naverlabs.com/mochi

https://europe.naverlabs.com/mochi