Photogrammetry & Robotics Lab

Machine Learning for Robotics and Computer Vision Tutorial

Pre-training & Self-supervision

Jens Behley

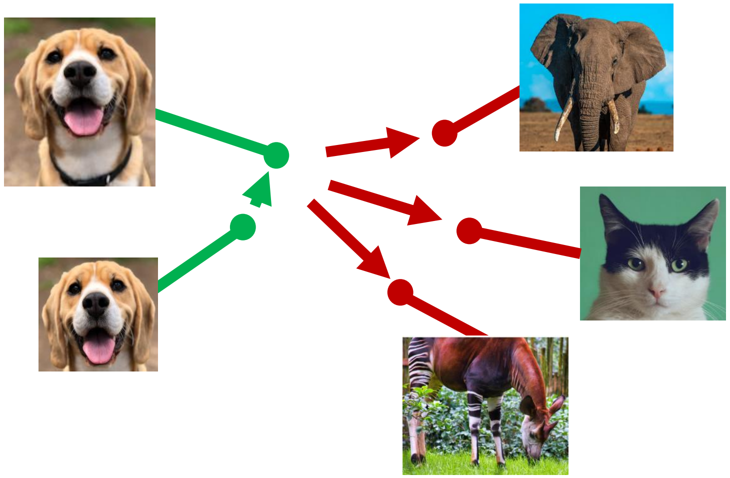
Exam Dates

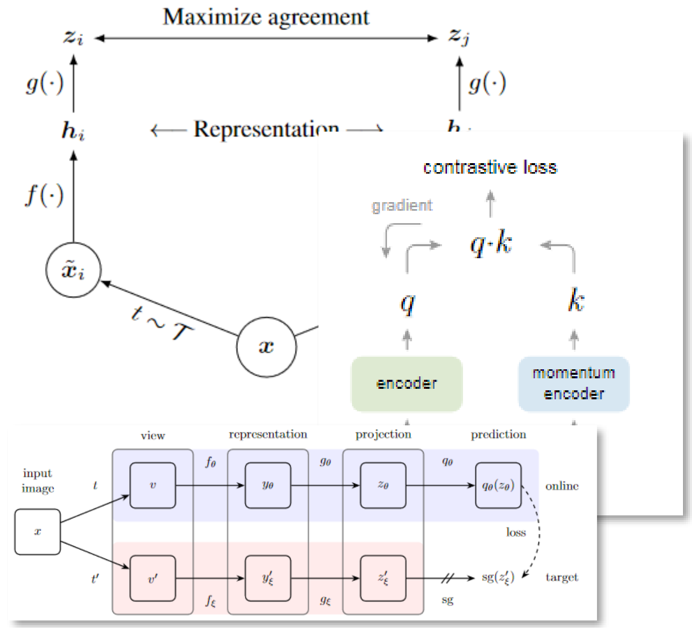
- Oral Exam via Zoom in English
- Webcam must be on all the time and alone in room
- No other windows besides Zoom open.

- Date from the voting: **Wed, 25.08.2021**

- If this date still doesn't fit, contact us and we provide one alternative date

This week's lecture





- Purely supervised training does not scale
- Using pre-trained models allows to get away with less labels!
- Self-supervised pretraining shows strong performance without any labels!

Pre-training & Fine-tuning

Stage 1:
Pre-training

(ImageNet)

Stage 2:
Fine-tuning
(Targeted dataset)

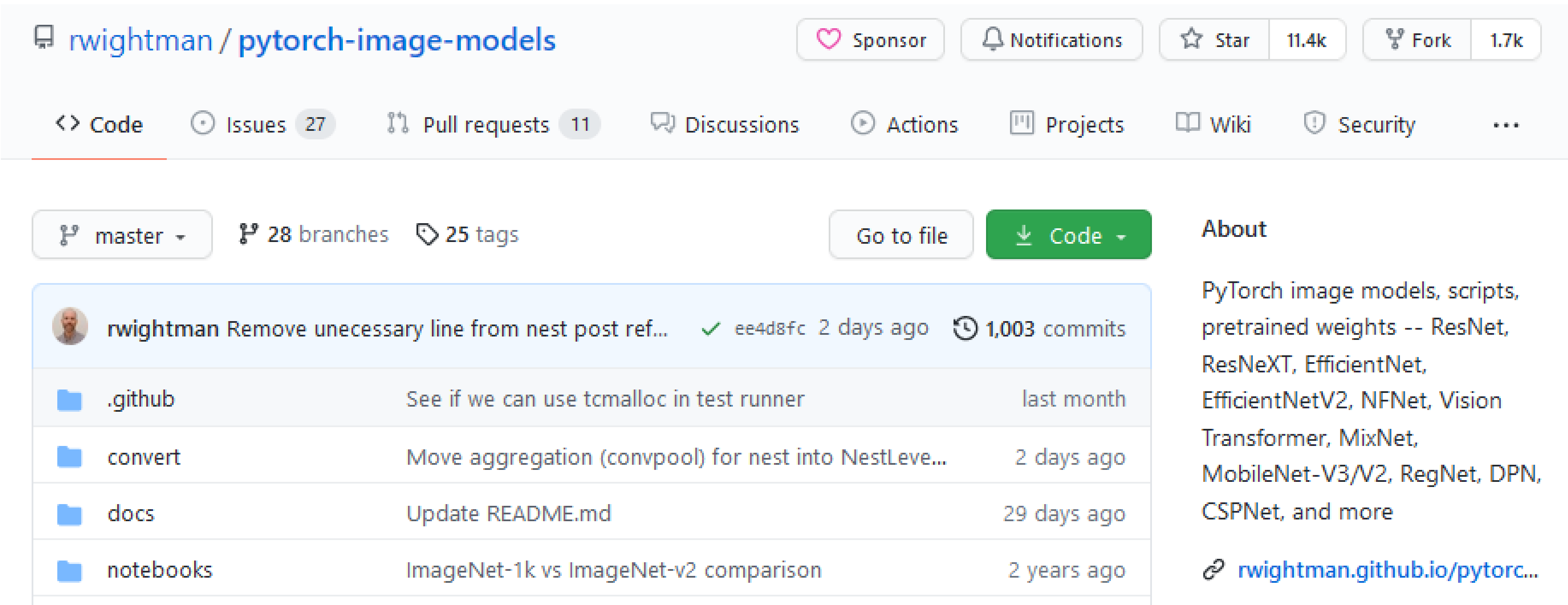
- **Idea:** Take weights from ImageNet and train only part of the network for novel task/dataset
- Training with pre-trained weights is faster and less data intensive!

Where to get pre-trained models?

- PyTorch vision contains pre-trained models for classification, segmentation, detection

- Repository of other often contain pre-trained versions of their approach

Pytorch-image-models Repo



- Maintained by Ross Wightman
- Up-to-date implementation and pre-trained weights of state-of-the-art backbones
- 452(!) pretrained models/variants of common models

Feature Extraction

- The timm library provides handy methods to get just the features, see Docs: https://rwightman.github.io/pytorch-image-models/feature_extraction/

forward_features()

```
import torch
import timm
m = timm.create_model('xception41', pretrained=True)
o = m(torch.randn(2, 3, 299, 299))
print(f'Original shape: {o.shape}')
o = m.forward_features(torch.randn(2, 3, 299, 299))
print(f'Unpooled shape: {o.shape}')
```

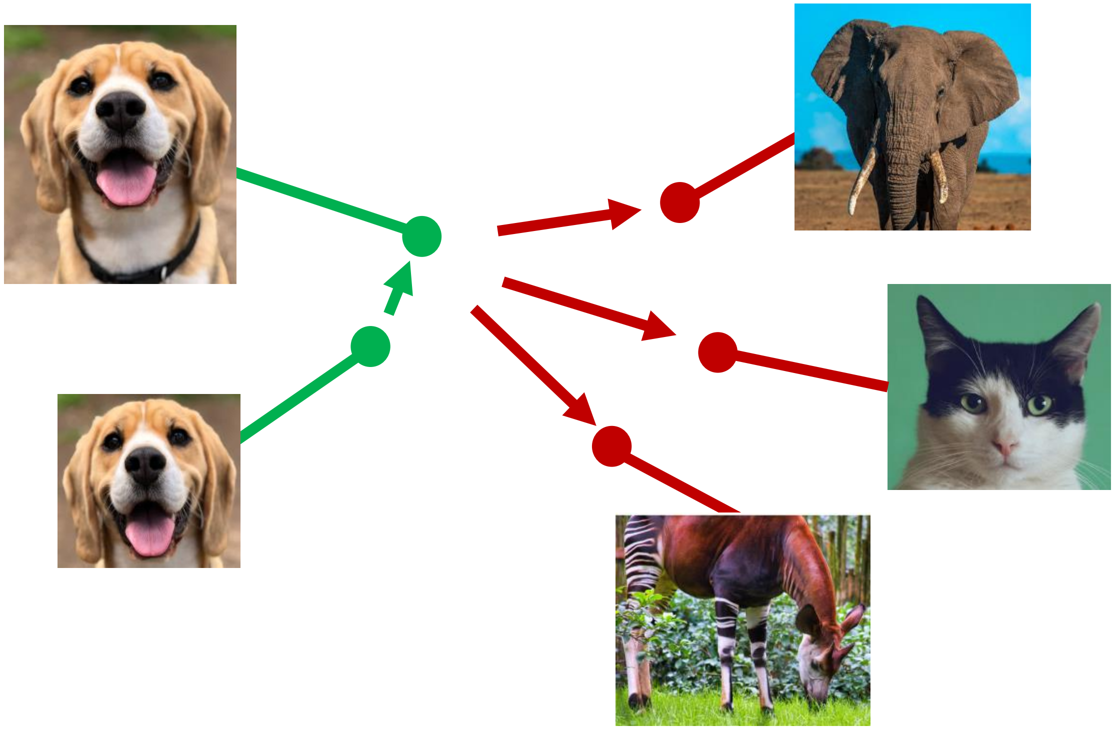
Output:

```
Original shape: torch.Size([2, 1000])
Unpooled shape: torch.Size([2, 2048, 10, 10])
```

Other Domains or Modalities

- ImageNet features or characteristics not always the best fit → **Self-supervised Learning**
- Specifically: **Contrastive Learning**

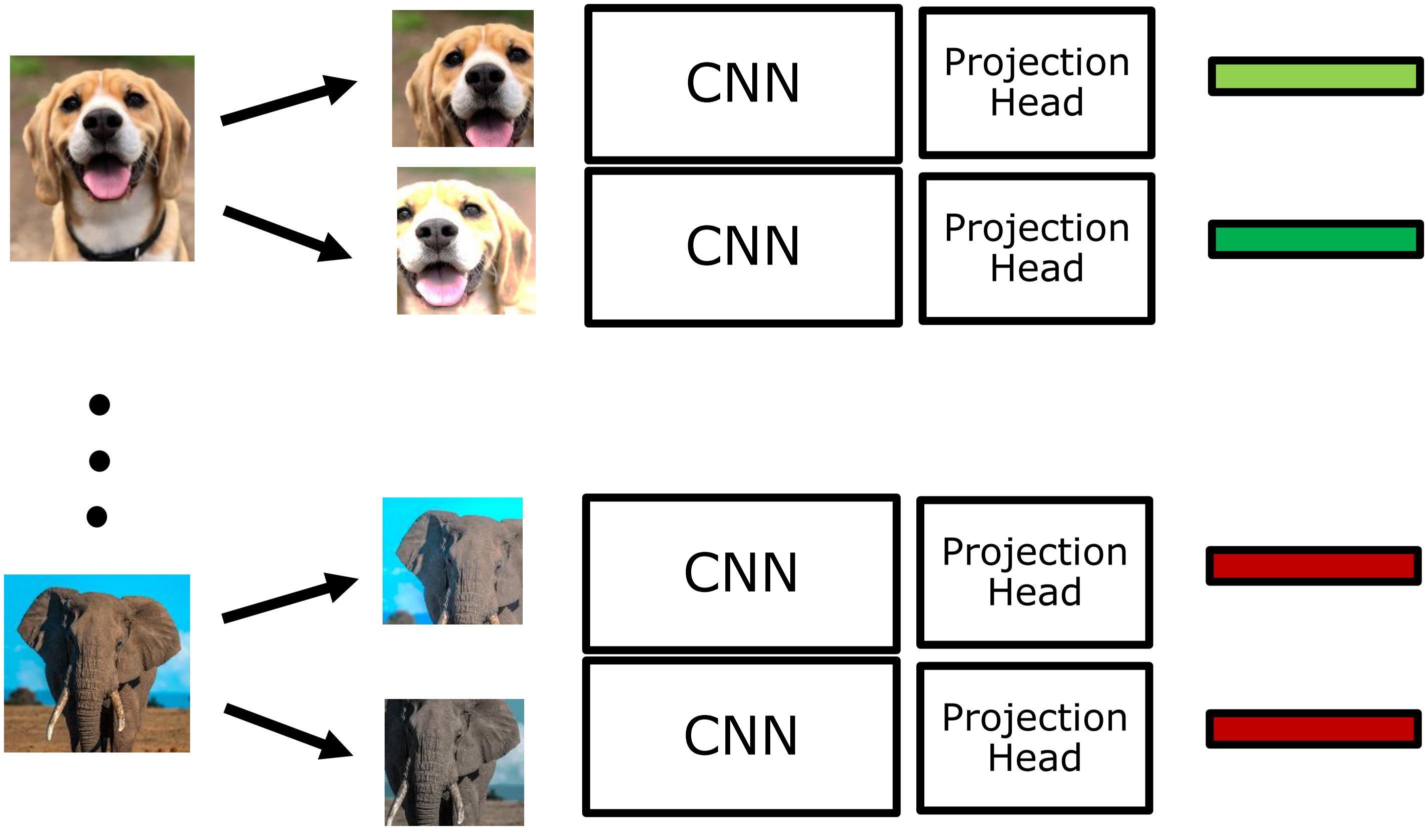
Contrastive Learning



- **Idea:** Learn representations such that similar examples (positives) are closer than representations of different examples (negatives)

Photos from unsplash.com

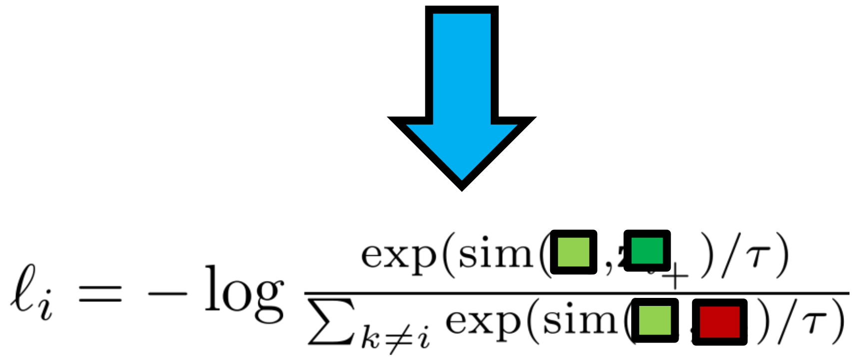
Common framework



Contrastive Loss

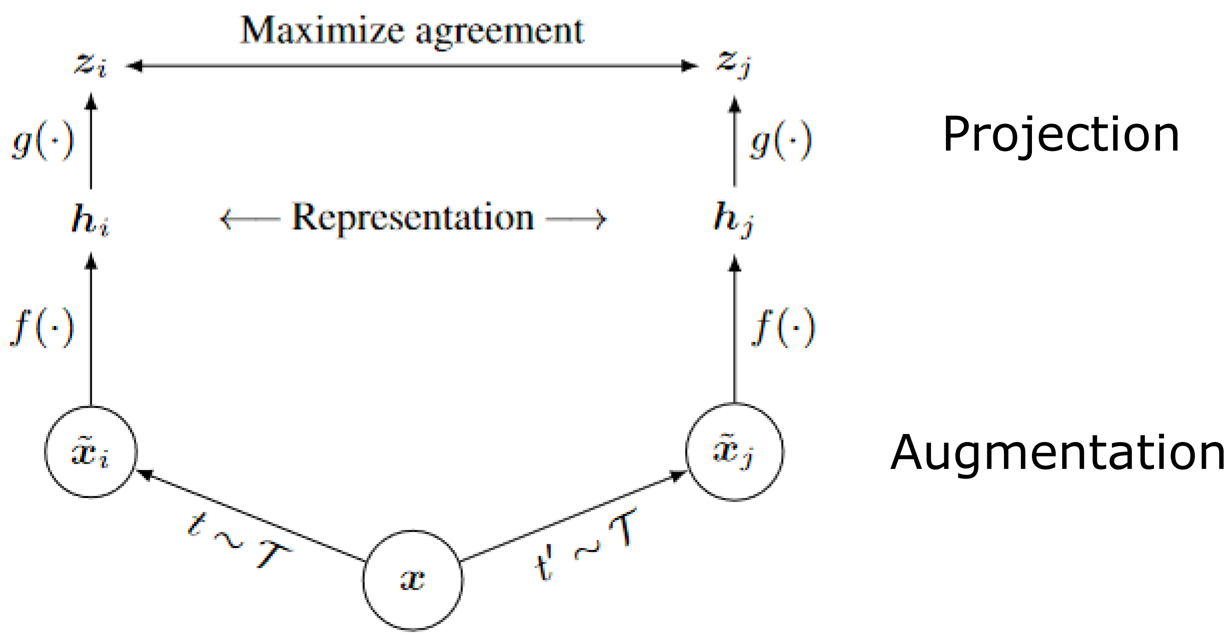
- Temperature scaled **contrastive loss**:

$$\ell_i = -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_{i_+})/\tau)}{\sum_{k \neq i} \exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$



- Cosine Similarity: $\mathrm{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$

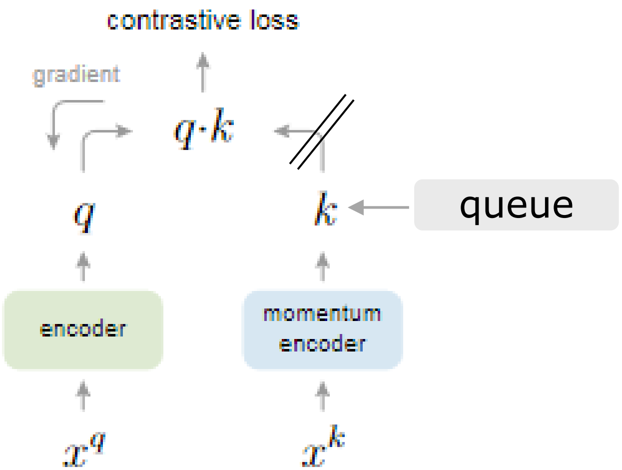
SimCLR



- **Idea:** Learn representations by finding agreement between *projected* features
- Compute contrastive loss over projections/latents z
- Projection $g(\cdot)$ via FC → ReLU → FC

[Chen, 2020]

Momentum Encoder



MoCo

- Only updated with weighted average between parameters of encoder θ_q and parameters of momentum encoder θ_k :

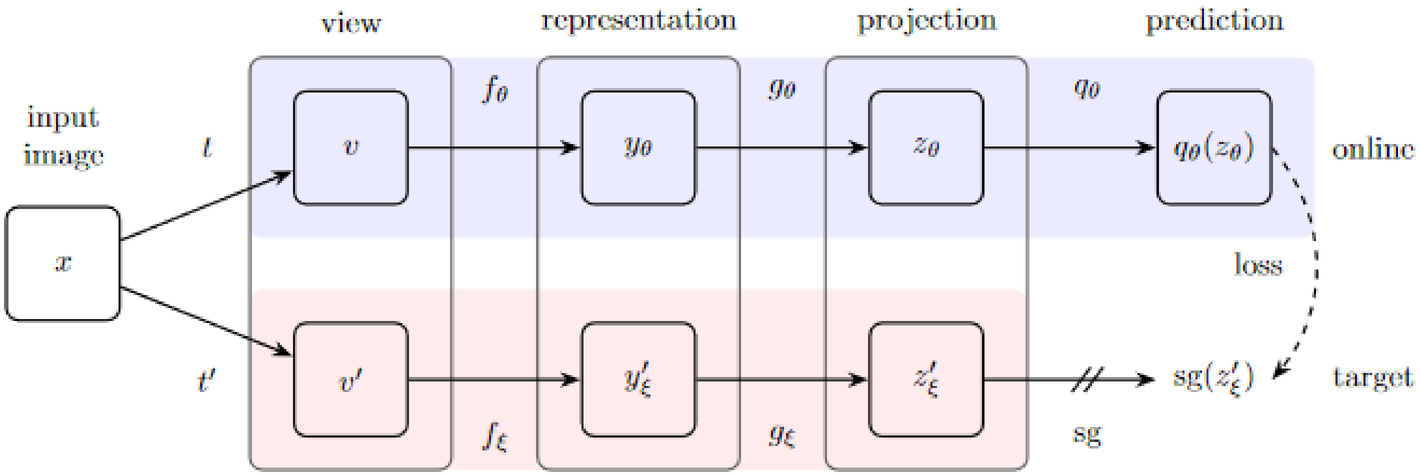
$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q$$

- Typically, large values (e.g., m = 0.999) better then smaller values (e.g., m = 0.9)

[He, 2020]

Figure from [He, 2020]

Boostrap your own latent (BYOL)



- Augmented views are passed through online and target network
- Online network predicts output of the target network

- Important: There are no negative examples involved!

[Grill, 2020]

Figure from [Grill, 2020]

BYOL training and update

- Loss measures difference between prediction $q(z_\theta)$ and output of target network z'_ξ :

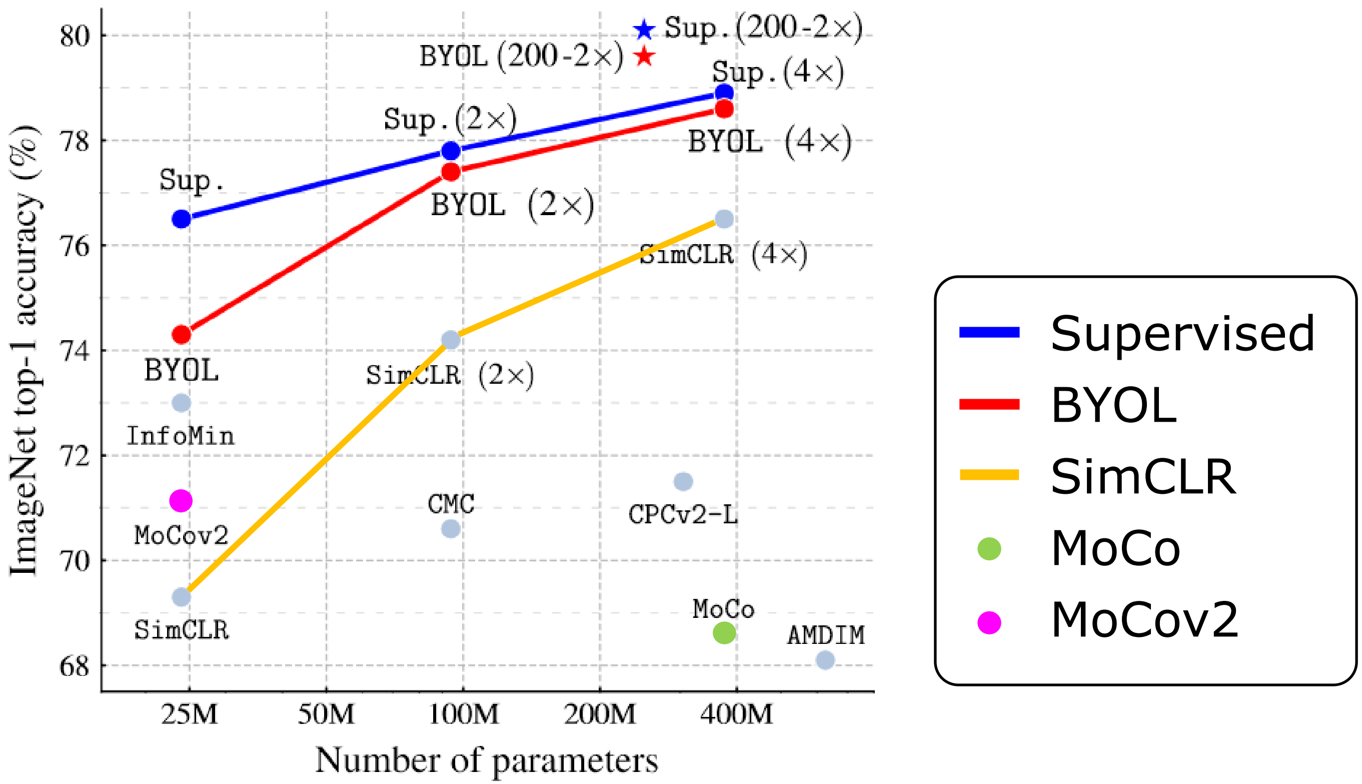
$$\ell = \left\| \frac{q(z_\theta)}{\|q(z_\theta)\|_2} - \frac{z'_\xi}{\|z'_\xi\|_2} \right\|_2^2 = 2 - 2 \cdot \frac{q(z_\theta)^\top z'_\xi}{\|q(z_\theta)\|_2 \|z'_\xi\|_2}$$

- Only online network is directly updated via backpropagation
- Target network parameters ξ are updated via momentum:

$$\xi \leftarrow m\xi + (1-m)\theta$$

[Grill, 2020]

Comparison on ImageNet



- Results for ResNet50 with different widths (=number of channels), e.g., 2x, 4x
- BYOL approaches supervised training

[Grill, 2020]

Figure from [Grill, 2020]

See you next week!