Faculty of Computer Science

Research Seminar

2022

# DINO

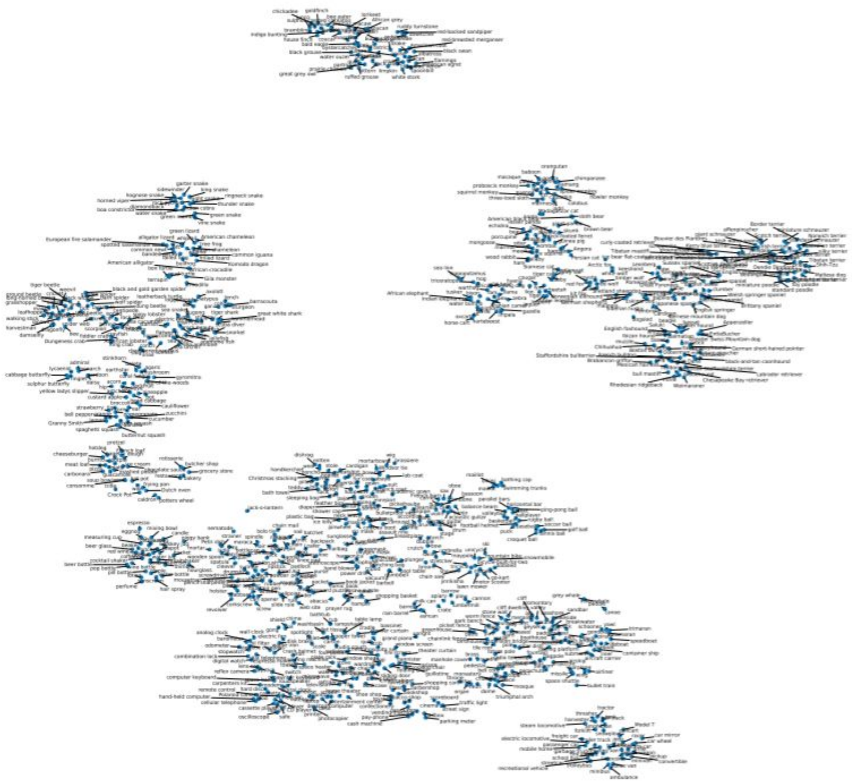
## Emerging Properties in Self-Supervised Vision Transformers

# BYOL

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

Akulov Dmitry

# Achieved results with DINO



# Negative sampling

Previous SOTA self-supervised methods used negative sampling approach for image representation tasks
Used to prevent collapsed solutions
This approach is based on two key ideas

- Reducing distance between positive pairs
- Decreasing distance between negative pairs

One of the problem is process of generating negative samples

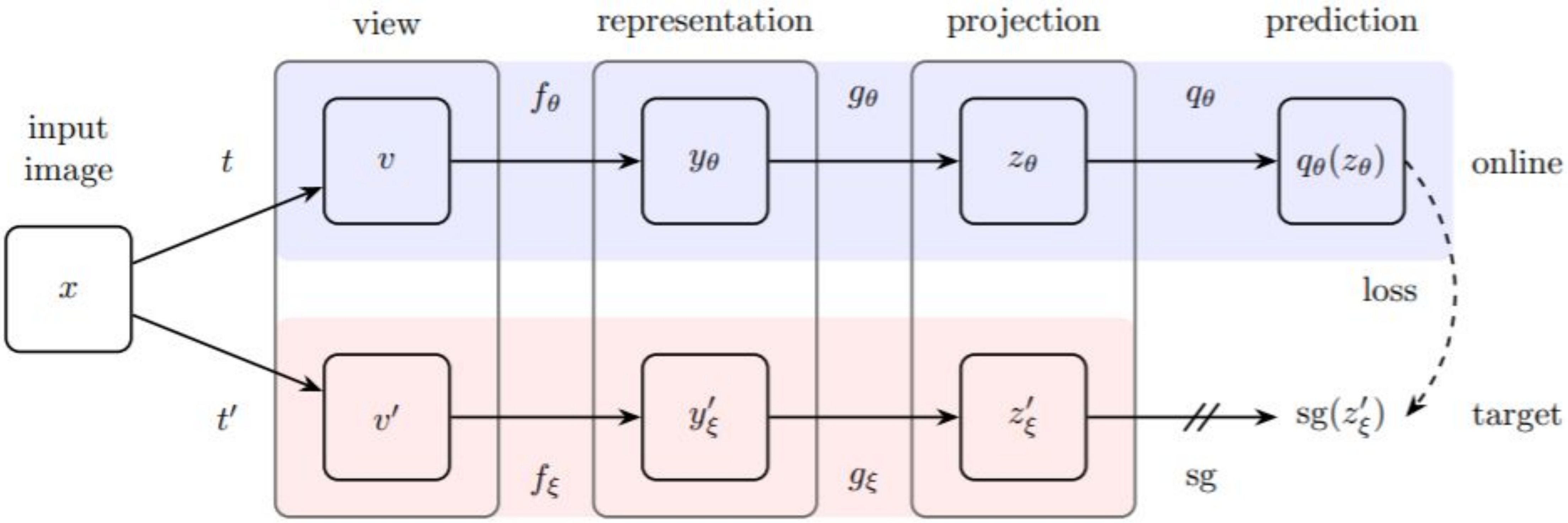
Some non-obvious problems as critical dependence on augmentation types

# BYOL don’t use negative samples

There is a Risk of getting collapsed solutions
It is empirically shown that BYOL does not converge to such solutions.

- BYOL achieves state-of-the-art results under the linear evaluation protocol on ImageNet without using negative pairs.
- BYOL learned representation outperforms the state of the art on semi-supervised and transfer benchmarks.
- BYOL is more resilient to changes in the batch size and in the set of image augmentations compared to its contrastive counterparts.

# BYOL model



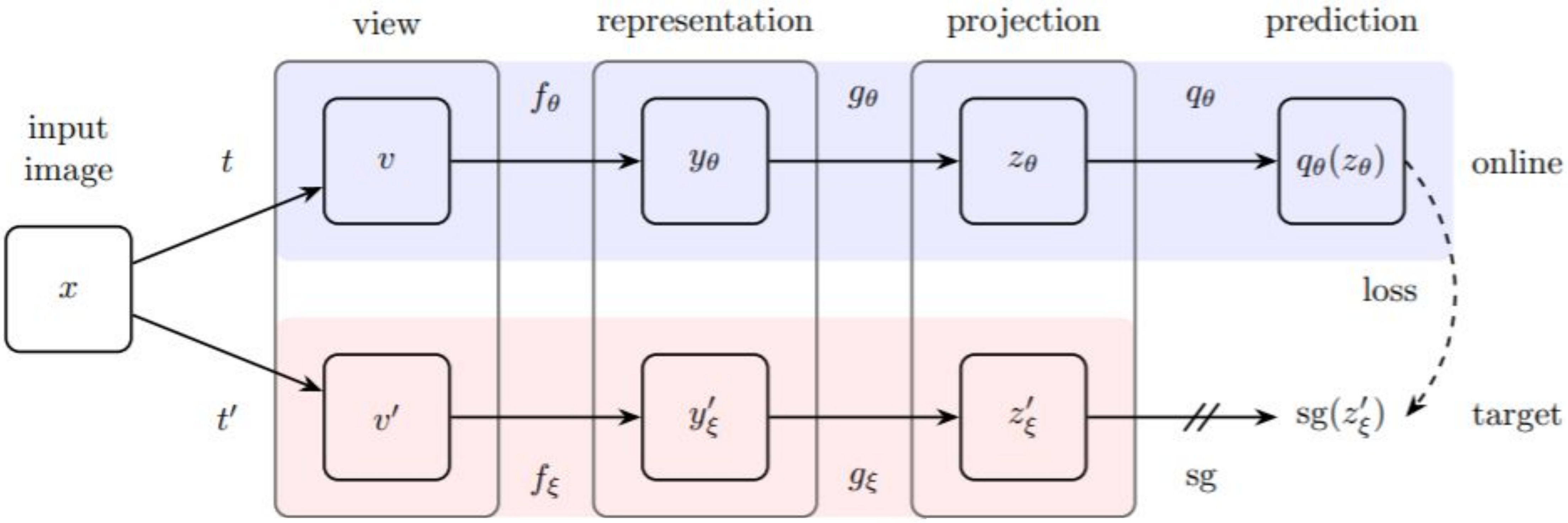

f - ResNet

q - MLP

t, t' - Augmentation

g - MLP

# BYOL model

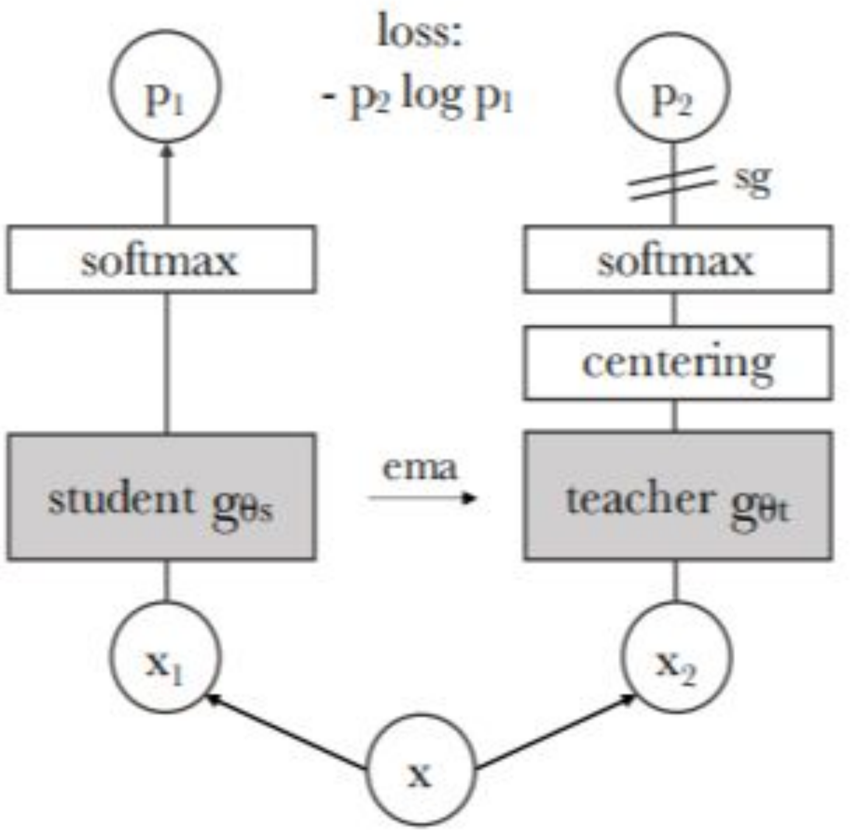



$$\mathcal{L}_{\theta,\xi} \triangleq \left\|\overline{q_\theta}(z_\theta) - \overline{z}'_\xi\right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi\rangle}{\left\|q_\theta(z_\theta)\right\|_2 \cdot \left\|z'_\xi\right\|_2}.$$

$$\theta \leftarrow \mathrm{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\mathtt{BYOL}}_{\theta,\xi}, \eta\right),$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta,$$

# DINO model

- Knowledge **di**stillation with **no** labels
- Co-distillation process
- Framework is flexible and works on both convnets and ViTs without the need to modify the architecture, nor adapt internal normalizations
- Inspired by BYOL
- Another loss function + Centering



# DINO code

**Algorithm 1** DINO PyTorch pseudocode w/o multi-crop.

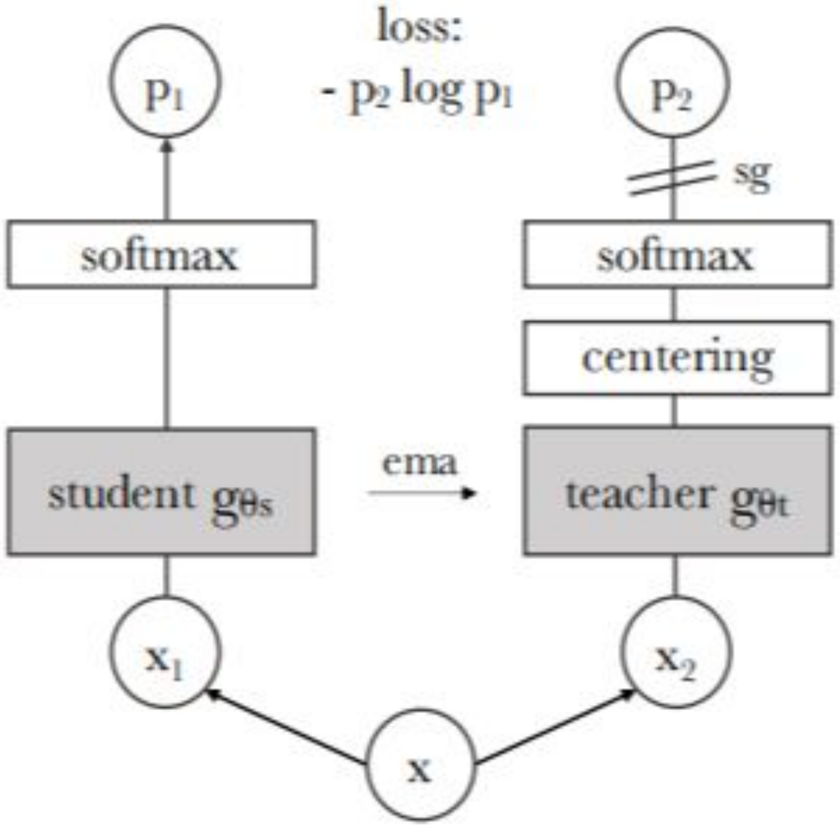
```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```



# Avoiding collapse

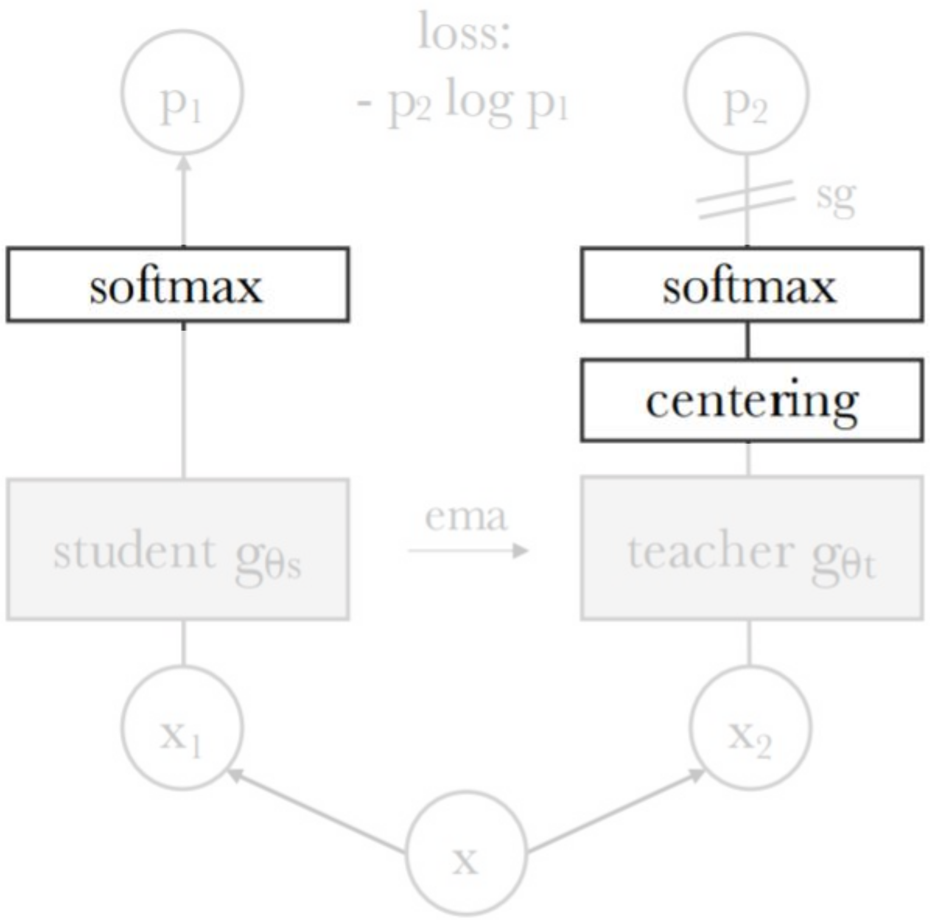
- Centering + Sharpening
- Sharpening:

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)},$$

- Centering:

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i),$$

$$g_t(x) \leftarrow g_t(x) + c.$$



# Avoiding collapse

- Centering + Sharpening
- Sharpening:

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)},$$

- Centering:

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i),$$

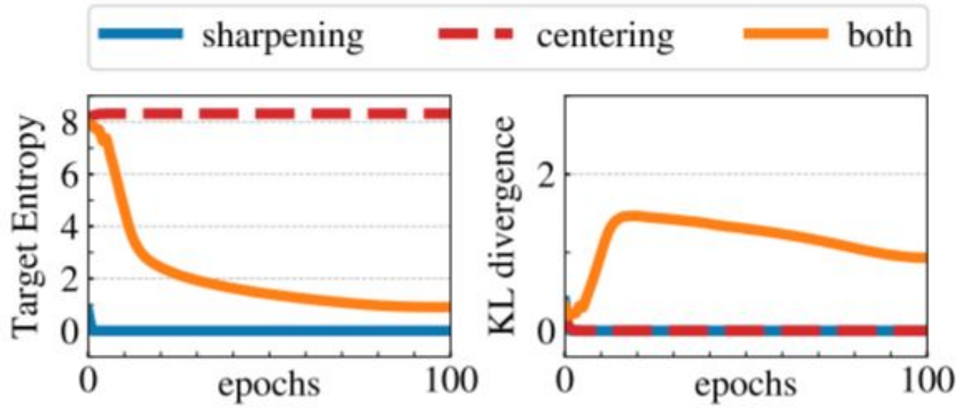
$$g_t(x) \leftarrow g_t(x) + c.$$

There are two forms of collapse: regardless of the input,

- the model output is uniform along all the dimensions or
- dominated by one dimension

The centering avoids the collapse induced by a dominant dimension, but encourages an uniform output.
Sharpening induces the opposite effect.

$$H(P_t, P_s) = h(P_t) + D_{KL}(P_t|P_s).$$



# DINO results

- Best performance among self-supervised methods
- DINO with ViT performance with a simple k-NN classifier is almost on par with a linear classifier
- Comparing across architectures shows that ViT with 8 × 8 patches trained with DINO achieves 80.1% top-1 in linear classification and 77.4% with a k-NN classifier with
  - 10× less parameters and
  - 1.4× faster run time

  than previous SOTA

| Method | Arch. | Param. | im/s | Linear | k-NN |
|---|---|---|---|---|---|
| Supervised | RN50 | 23 | 1237 | 79.3 | 79.3 |
| SCLR [12] | RN50 | 23 | 1237 | 69.1 | 60.7 |
| MoCov2 [15] | RN50 | 23 | 1237 | 71.1 | 61.9 |
| InfoMin [67] | RN50 | 23 | 1237 | 73.0 | 65.3 |
| BarlowT [81] | RN50 | 23 | 1237 | 73.2 | 66.0 |
| OBoW [27] | RN50 | 23 | 1237 | 73.8 | 61.9 |
| BYOL [30] | RN50 | 23 | 1237 | 74.4 | 64.8 |
| DCv2 [10] | RN50 | 23 | 1237 | 75.2 | 67.1 |
| SwAV [10] | RN50 | 23 | 1237 | **75.3** | 65.7 |
| DINO | RN50 | 23 | 1237 | **75.3** | **67.5** |
| Supervised | ViT-S | 21 | 1007 | 79.8 | 79.8 |
| BYOL* [30] | ViT-S | 21 | 1007 | 71.4 | 66.6 |
| MoCov2* [15] | ViT-S | 21 | 1007 | 72.7 | 64.4 |
| SwAV* [10] | ViT-S | 21 | 1007 | 73.5 | 66.3 |
| DINO | ViT-S | 21 | 1007 | **77.0** | **74.5** |
| *Comparison across architectures* | | | | | |
| SCLR [12] | RN50w4 | 375 | 117 | 76.8 | 69.3 |
| SwAV [10] | RN50w2 | 93 | 384 | 77.3 | 67.3 |
| BYOL [30] | RN50w2 | 93 | 384 | 77.4 | – |
| DINO | ViT-B/16 | 85 | 312 | 78.2 | 76.1 |
| SwAV [10] | RN50w5 | 586 | 76 | 78.5 | 67.1 |
| BYOL [30] | RN50w4 | 375 | 117 | 78.6 | – |
| BYOL [30] | RN200w2 | 250 | 123 | 79.6 | 73.9 |
| DINO | ViT-S/8 | 21 | 180 | 79.7 | **78.3** |
| SCLRv2 [13] | RN152w3+SK | 794 | 46 | 79.8 | 73.1 |
| DINO | ViT-B/8 | 85 | 63 | **80.1** | 77.4 |

# Attention maps from multiple heads

# Video Segmentation

- The 2017 DAVIS Challenge on Video Object Segmentation

- They evaluated the quality of frozen features on video instance tracking.
- They compared with existing self-supervised methods and a supervised ViT-S/8 trained on ImageNet.
- They thus did not train any model on top of the features, nor finetune any weights for the task
- They observed that even though their training objective nor our architecture are designed for dense tasks, the performance is competitive on this benchmark.

| Method | Data | Arch. | $(\mathcal{J}\&\mathcal{F})_m$ | $\mathcal{J}_m$ | $\mathcal{F}_m$ |
|---|---|---|---|---|---|
| *Supervised* | | | | | |
| ImageNet | INet | ViT-S/8 | 66.0 | 63.9 | 68.1 |
| STM [48] | I/D/Y | RN50 | 81.8 | 79.2 | 84.3 |
| *Self-supervised* | | | | | |
| CT [71] | VLOG | RN50 | 48.7 | 46.4 | 50.0 |
| MAST [40] | YT-VOS | RN18 | 65.5 | 63.3 | 67.6 |
| STC [37] | Kinetics | RN18 | 67.6 | 64.8 | 70.2 |
| DINO | INet | ViT-S/16 | 61.8 | 60.2 | 63.4 |
| DINO | INet | ViT-B/16 | 62.3 | 60.7 | 63.9 |
| DINO | INet | ViT-S/8 | **69.9** | **66.6** | **73.1** |
| DINO | INet | ViT-B/8 | **71.4** | **67.9** | **74.9** |

# Conclusion

- Self-supervised pretraining can achieve comparable to convolutional networks performance
- Generated features have high quality for k-NN classification
- The presence of information about the scene layout in the features can also benefit weakly supervised image segmentation