**Digest**

# Self-distillation with no labels (DINO)

**Paper: Emerging Properties in Self-Supervised Vision Transformers**

M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, A. Joulin

ICCV (2021)

**Digest** by Samuel Albanie, June 2022

# Outline

- Motivation
- Related work
- DINO framework
- Evaluation protocols
- Experiments
- Ablations
- Summary

# Motivation

## Vision transformers and self-supervision

Transformers have seen tremendous success in NLP

Vision Transformer (ViT) demonstrated competitive performance with CNNs, but did not show dramatic benefits

A key factor in NLP successes was the use of self-supervision:

BERT (Devlin et al., 2019) - clozes/next sentence prediction

GPT (Radford et al., 2019) - language modelling

However, ViT is trained in a fully-supervised manner

**Would Vision Transformers also benefit from self-supervision?**

## Emergent properties

Transformers encode a different set of inductive biases to CNNs

Without convolutions, they do not enforce the principle of locality

It is possible that Transformers behave differently under self-supervision

They may encode scene layout or object boundaries differently

**Do different properties emerge from Transformers than CNNs?**

## Which factors matter?

Many ideas in the self-supervised literature have improved performance

- momentum encoders (He et al., 2020)
- multi-crop augmentation (Caron et al., 2020)

**How do these components affect feature properties?**

**Reference:**
(Transformers) A. Vaswani et al., "Attention is all you need", NeurIPS (2017)
(ViT) A. Dosovitskiy, et al. "An image is worth 16x16 words: Transformers for image recognition at scale", ICLR (2021)
(BERT) J. Devlin et al., "Bert: Pre-training of deep bidirectional transformers for language understanding", NAACL-HLT (2019)
(GPT) A. Radford et al., "Language models are unsupervised multitask learners" (2019)
K. He et al., "Momentum contrast for unsupervised visual representation learning", *CVPR* (2020)
M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", *NeurIPS* (2020)

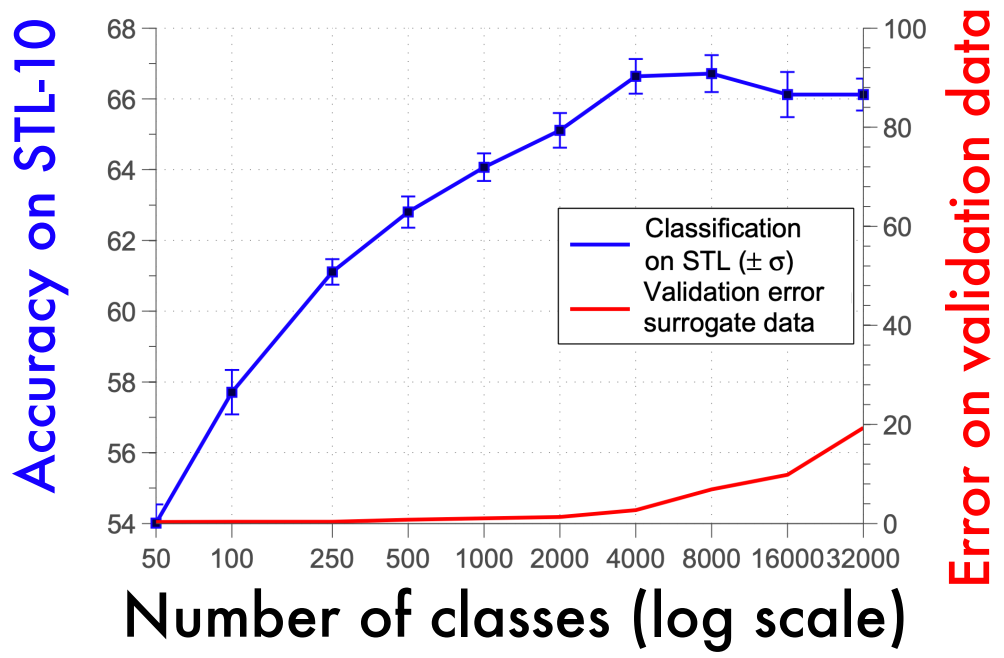
# Outline

- Motivation
- Related work
- DINO framework
- Evaluation protocols
- Experiments
- Ablations
- Summary

# Related Work

## Dosovitskiy et al., (2014)

Classes from patches

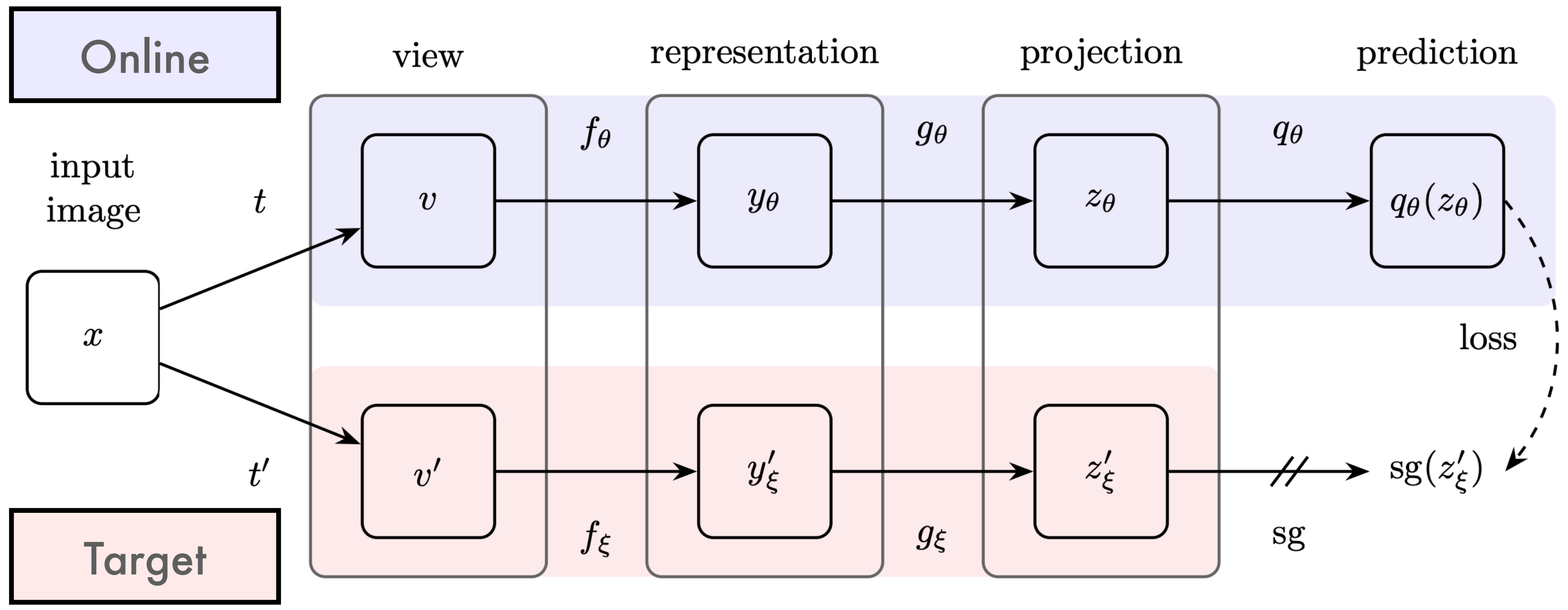



Influence of classes

*best augmentation depends on downstream data*

Exemplar-CNN

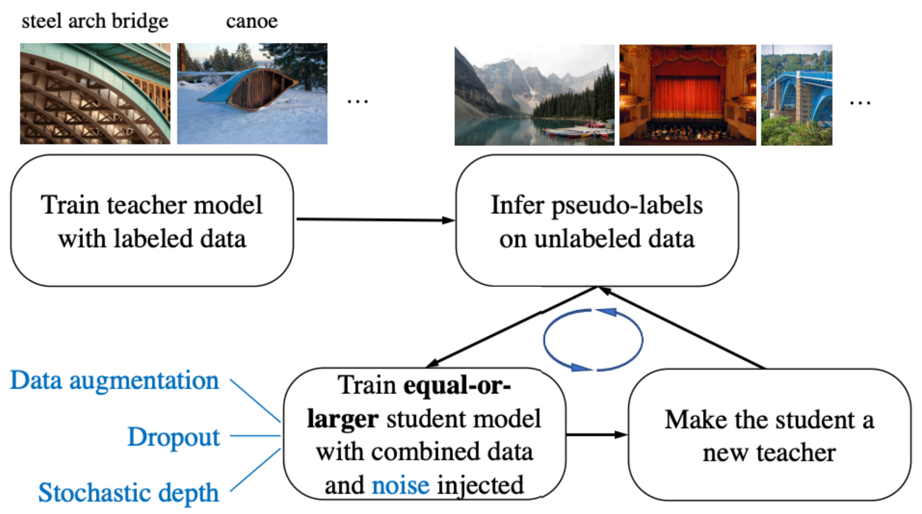
## Grill et al., (2020)



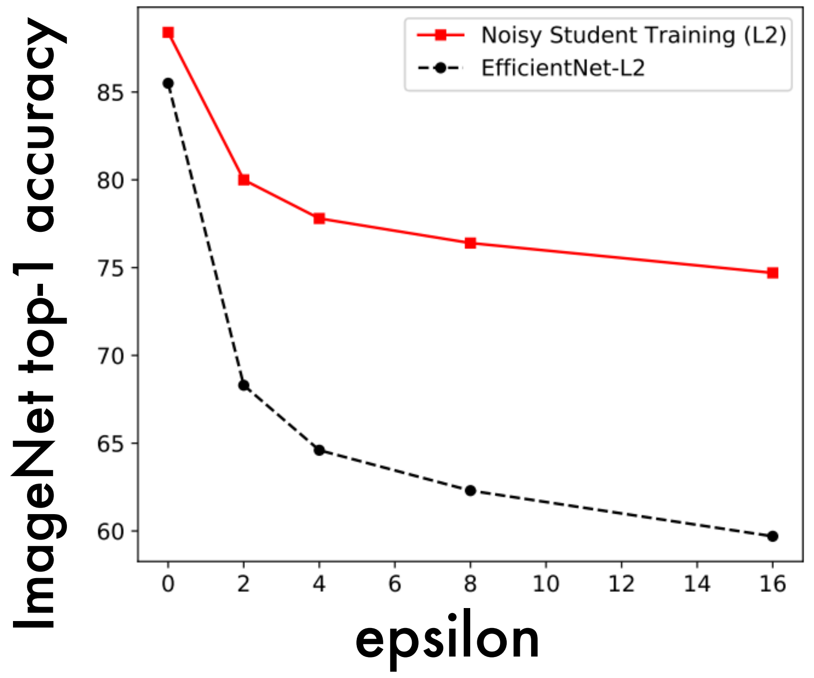

Framework

BYOL

## Xie et al., (2020)



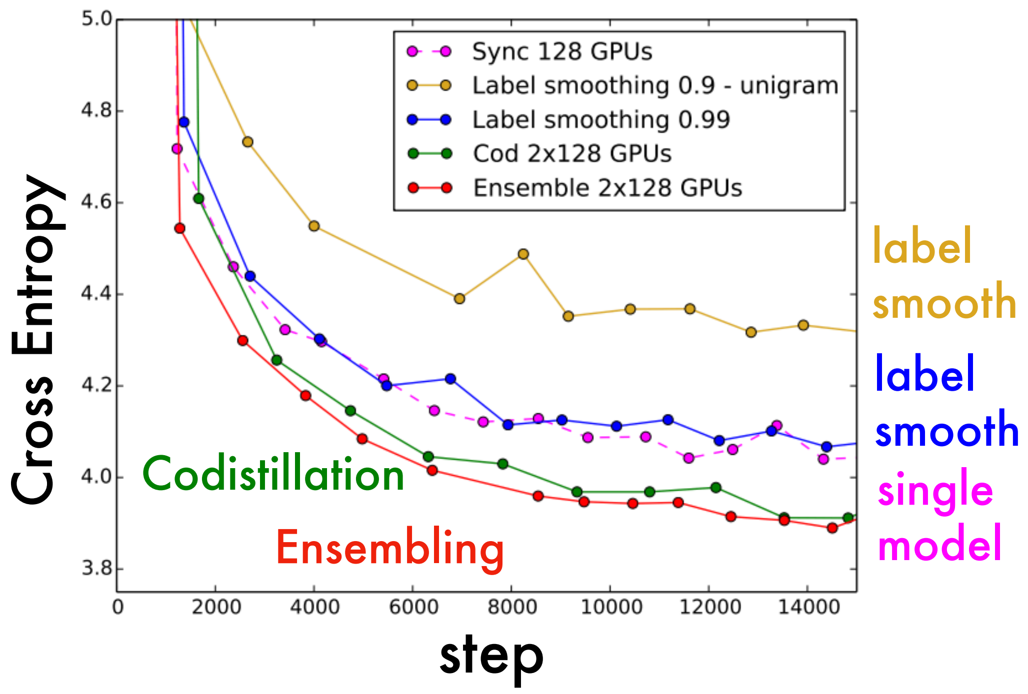

Distillation framework



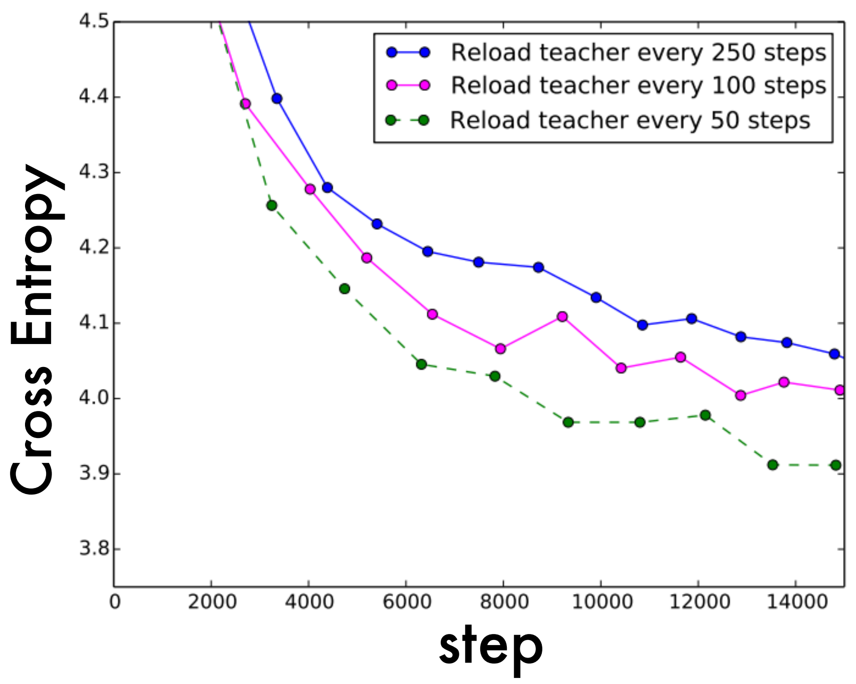

Robustness against FGSM

Noisy Student

## Anil et al., (2018)




Co-distillation vs baselines (LM task)




Staleness

Codistillation

**References/Image credits:**

A. Dosovitskiy et al., "Discriminative unsupervised feature learning with convolutional neural networks", NeurIPS (2014)

Q. Xie et al., "Self-training with noisy student improves imagenet classification", CVPR (2020)

J-B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)

R. Anil et al., "Large scale distributed neural network training through online distillation", arxiv (2018)

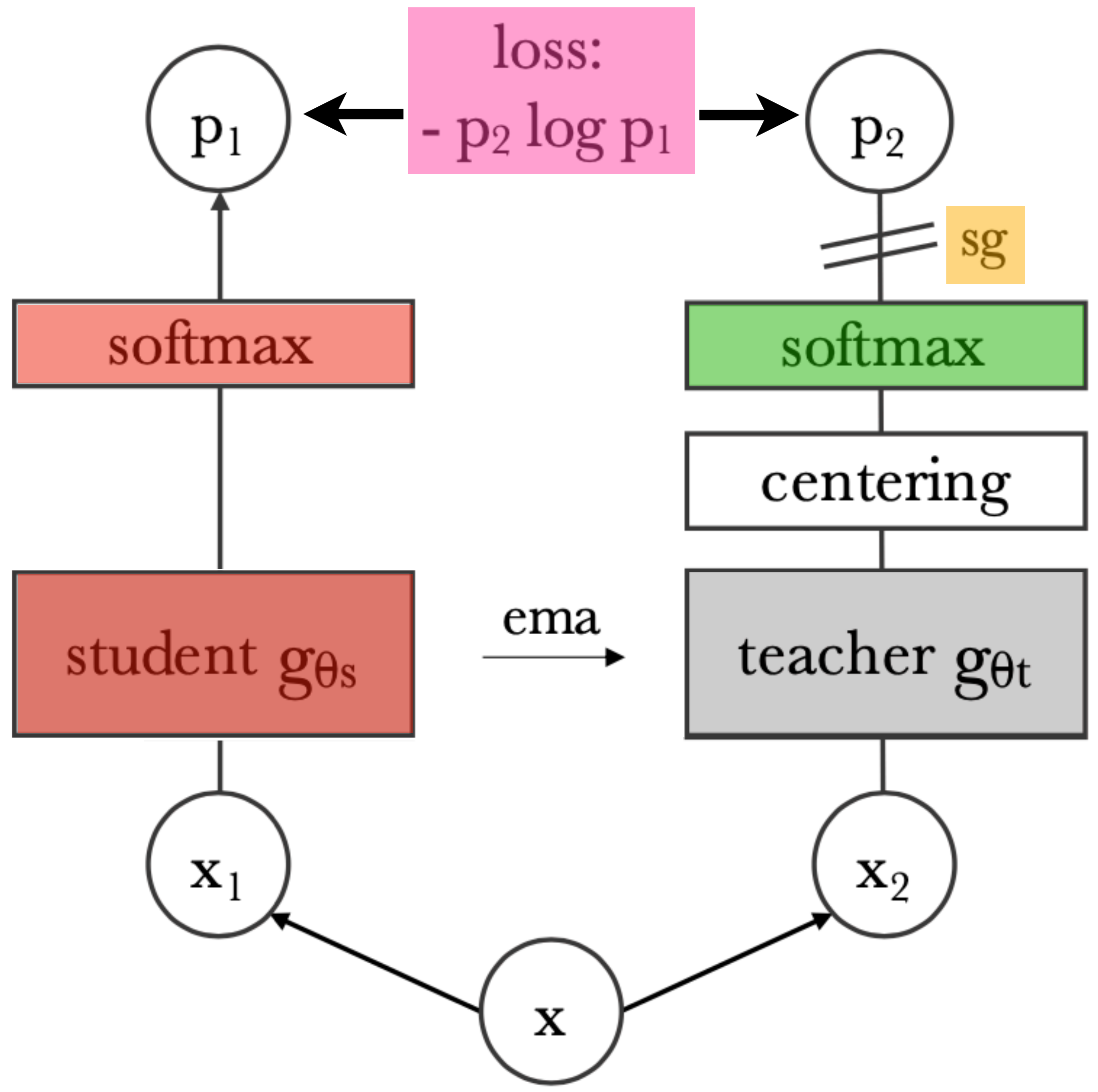
# Outline

- Motivation
- Related work
- DINO framework
- Evaluation protocols
- Experiments
- Ablations
- Summary

# Framework: Self-Supervised Learning with Knowledge Distillation

## Overview

Knowledge distillation: student learns from teacher




## Distribution matching

A student distribution is produced via a softmax:

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

where $\tau_s > 0$ is a temperature hyperparameter

Similarly a teacher distribution is produced via:

$$P_t(x)^{(i)} = \frac{\exp(g_{\theta_t}(x)^{(i)}/\tau_t)}{\exp(\sum_{k=1}^{K} g_{\theta_t}(x)^{(k)}/\tau_t)}$$

where $\tau_t > 0$ is another temperature hyperparameter

Distributions are matched via cross-entropy loss:

$\min_{\theta_s} H(P_t(x), P_s(x))$ where $H(a, b) = -a \log b$

Minimised w.r.t student parameters

## Pseudocode

**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()
```

**Reference/Image credits:**


M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

# Framework: Self-Supervised Learning with Knowledge Distillation

## Global and local views

In practice, $V \geq 2$ views of each image are used

Inspired by multicrop strategy of SwAV

The set of views $V$ contains:

- 2 global views $x_1^g, x_2^g$
- several local views of smaller resolution

Only global views are passed to the teacher

All crops (global and local) are passed to the student

This encourages local-to-global correspondences

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x'))$$

both local and global

Global views are crops at $224^2$ resolution (> 50% area)

Local views are crops at $96^2$ resolution ( $\leq 50\,\%$ area)

## Teacher network

We do not have access to a supervised teacher

Instead is built from past iterations of the student

It is found that a momentum encoder works well with exponential moving average (EMA)

Update rule for teacher: $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$

where $\lambda$ follows a cosine schedule from 0.996 to 1

**Note:** the role of the momentum encoder in DINO is different to its role in MoCo (a queue for consistency)

It may be closer to that of *Mean Teacher* (model parameter ensembling)

Similar to Ruppert-Polyak model averaging to improve performance, resulting in a teacher that performs better than the student during training

## Network architecture

The neural network $g$ consists of:

- a backbone $f$ (ViT or ResNet)
- a projection head $h$

These are composed $g = h \circ f$

The features from $f$ are used for downstream tasks

The projection head is a 3-layer MLP (similar to SwAV)

- 2048 dimensional hidden layer with $l_2$ normalisation
- a fully connected layer with weight norm with K dims

No predictor used (unlike BYOL, identical teacher/student)

Since ViT architectures do not use batch norm by default, DINO with ViT backbone is free from batch norm


**References:**

(SwAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)

(Momentum Encoder, MOCO) K. He et al., "Momentum contrast for unsupervised visual representation learning", CVPR (2020)

(Cosine schedule) J-B Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)

(MeanTeacher) A. Tarvainen et al., "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results", *NeurIPS* (2017)

D. Ruppert, "Efficient estimations from a slowly convergent Robbins-Monro process" (1988)

B. T. Polyak et al., "Acceleration of stochastic approximation by averaging." SICON (1992)

(ViT) A. Dosovitskiy, et al. "An image is worth 16x16 words: Transformers for image recognition at scale", ICLR (2021)

(ResNet) K. He et al., "Deep residual learning for image recognition", CVPR (2016)

(WeightNorm) T. Salimans et al., "Weight normalization: A simple reparameterization to accelerate training of deep neural networks", NeurIPS (2016)

# DINO: Avoiding collapse

**Normalisation constraints to prevent collapse**

A key problem for self-supervised methods is the prevention of representation collapse to a single vector

Different mechanisms have been used to prevent collapse:

- Contrastive loss (e.g. Instance Discrimination)
- Clustering constraints (e.g. DeepCluster, SwAV)
- Predictor & Batch Norm (e.g. BYOL)
- Batch Norm alternatives such as Group Norm and Weigh Norm (BYOL-variant)

DINO is found to work well with a combination of centring and sharpening of the teacher outputs

This approach trades stability in exchange for reduced dependence on the batch

Centring (unlike batch norm) only depends on first-order batch statistics

This operation can be interpreted as adding a bias term $c$ to the teacher: $g_t(x) \leftarrow g_t(x) + c$

The centre $c$ is updated with an EMA, so it works well across different batch sizes:

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

where $m > 0$ is a rate parameter and $B$ is the batch size

Sharpening is achieved by using a low softmax temperature $\tau_t$ for the teacher

**References:**
(InstanceDisc) Z. Wu et al., "Unsupervised feature learning via non-parametric instance discrimination", CVPR (2018)
(DeepCluster) M. Caron et al., "Deep clustering for unsupervised learning of visual features", (ECCV) 2018
(SWAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)
(BYOL) J-B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(BYOL-variant) P. Richemond et al., "BYOL works even without batch statistics", arxiv (2020)
(Batch Norm) S. Ioffe et al., "Batch normalization: Accelerating deep network training by reducing internal covariate shift", ICML (2015)
(Group Norm) Y. Wu et al., "Group normalization", ECCV (2018)
(Weight Norm) T. Salimans et al., "Weight normalization: A simple reparameterization to accelerate training of deep neural networks", NeurIPS (2016)

# DINO: Nuts and bolts

## Vision Transformer (ViT) for DINO

| Name | ResNet-50 | ViT-S/16 | ViT-S/8 | ViT-B/16 | ViT-B/8 |
|---|---|---|---|---|---|
| blocks | - | 12 | 12 | 12 | 12 |
| dim | 2048 | 384 | 384 | 768 | 768 |
| heads | - | 6 | 6 | 12 | 12 |
| #tokens | - | 197 | 785 | 197 | 785 |
| #params | 23M | 21M | 21M | 85M | 85M |
| im/s | 1237 | 1007 | 180 | 312 | 63 |

DINO Vision Transformer implementation follows DeiT

blocks - number of transformer blocks

dim - channel dimension for representation

heads - number of heads in multi-head attention

#tokens - length of token sequence for $224^2$ pixel inputs

#params - total number of parameters (excluding projection head)

im/s - timings on an NVIDIA V100 GPU with 128 samples in the minibatch

As with prior work [CLS] token aggregates information - this is projected via the projection head $h$

## DINO optimisation details

Pretraining is performed on ImageNet without labels

Models are optimised with AdamW with a batch size of 1024 on 16 GPUs (ViT-S/16)

Use linear scaling learning rate warmup, then decays with a cosine schedule

Weight decay follows a cosine schedule from 0.04 to 0.4

The temperature of the student $\tau_s$ is set to 0.1, while the teacher temperature $\tau_t$ is warmed up linearly from 0.04 to 0.07 over the first 30 epochs

Use BYOL data augmentation (colour jitter, Gaussian blur, solarisation)

Bicubic interpolation is used to adapt the position embeddings across scales

Models/code available on GitHub

**References:**
H. Touvron et al., "Training data-efficient image transformers & distillation through attention", ICML (2021)
(ImageNet) O. Russakovsky et al., "Imagenet large scale visual recognition challenge", IJCV (2015)
(AdamW) I. Loshchilov et al., "Decoupled weight decay regularization", arxiv (2017)
(LR warmup) P. Goyal et al., "Accurate, large minibatch sgd: Training imagenet in 1 hour", arxiv (2017)
(Cosine schedule) I. Loshchilov et al., "Sgdr: Stochastic gradient descent with warm restarts", arxiv (2016)
(BYOL) J-B. Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(Github) https://github.com/facebookresearch/dino

# Outline

- Motivation
- Related work
- DINO framework
- Evaluation protocols
- Experiments
- Ablations
- Summary

# Evaluation protocols

## Evaluating DINO features

Typically, self-supervised features are evaluated via linear probes (on frozen features) and finetuning

Linear probe protocol:

- Data augmentation (random resize crops and horizontal flips) to train the probe
- Evaluate on a central crop

Finetuning protocol:

- Initialise network with pretrained weights and adapt them during training

**Note:** both protocols are sensitive to hyperparameter choices

So, also evaluate under a k-NN protocol (Wu et al., 2018):

- Freeze the pretrained model and compute features for training sets on downstream tasks
- Nearest neighbour classifier matches k nearest neighbours on training set and assigns label by votes

Sweep over different numbers of nearest neighbours - a value of 20 NN is found to work well

k-NN protocol requires no other hyperparameters or data augmentation

It also requires only one pass over the downstream dataset (simplifying the evaluation procedure)

**References:**
(k-NN self-sup protocol) Z. Wu et al., "Unsupervised feature learning via non-parametric instance discrimination", CVPR (2018)

# Evaluation protocols - details

## k-NN weighted evaluation protocol

Features are computed for training data of downstream task (frozen model)

Classify a test image $x$ by comparing features against training features $T$

Representation of an image is the [CLS] token

- 384 dimensional for ViT-S
- 768 dimensional for ViT-B

The top $k$ nearest neighbours, $\mathcal{N}_k$ make a class prediction by voting

Class $c$ is assigned a weight of $\sum_{i \in \mathcal{N}_k} \alpha_i \mathbf{1}_{c_i = c}$

The contribution weight $\alpha_i$ for each neighbour is defined via:

$\alpha_i = \exp(T_i x / \tau)$ with $\tau = 0.07$ following Wu et al. (2018)

A choice of $k = 20$ works consistently well

## Linear classification protocol

Remove projection head and train a supervised linear classifier on features

Classifier is trained with SGD and a batch size of 1024 for 100 epochs on ImageNet

No weight decay is applied

For each model, the learning rate is set by sweeping

During training: resize crops and horizontal flips, during testing: central crops

ViT-S *feature-based*

concatenate [CLS] tokens

| concatenate $l$ last layers | 1 | 2 | 4 | 6 |
|---|---|---|---|---|
| representation dim | 384 | 768 | 1536 | 2304 |
| ViT-S/16 linear eval | 76.1 | 76.6 | 77.0 | 77.0 |

ViT-B *CNN-style*

[CLS] & AVG POOL

| pooling strategy | `[CLS]` tok. only | concatenate `[CLS]` tok. and avgpooled patch tok. |
|---|---|---|
| representation dim | 768 | 1536 |
| ViT-B/16 linear eval | 78.0 | 78.2 |

**References:**
(k-NN self-sup protocol) Z. Wu et al., "Unsupervised feature learning via non-parametric instance discrimination", CVPR (2018)
(BERT) J. Devlin et al., "Bert: Pre-training of deep bidirectional transformers for language understanding", NAACL-HLT (2019)

# Outline

- Motivation
- Related work
- DINO framework
- Evaluation protocols
- Experiments
- Ablations
- Summary

# Experiments - ImageNet

## Comparing with the same architecture

Compare to existing self-supervised methods with the same architecture

| Method | Arch. | Param. | im/s | Linear | k-NN |
|---|---|---|---|---|---|
| Supervised | RN50 | 23 | 1237 | 79.3 | 79.3 |
| SCLR | RN50 | 23 | 1237 | 69.1 | 60.7 |
| MoCov2 | RN50 | 23 | 1237 | 71.1 | 61.9 |
| InfoMin | RN50 | 23 | 1237 | 73.0 | 65.3 |
| BarlowT | RN50 | 23 | 1237 | 73.2 | 66.0 |
| OBoW | RN50 | 23 | 1237 | 73.8 | 61.9 |
| BYOL | RN50 | 23 | 1237 | 74.4 | 64.8 |
| DCv2 | RN50 | 23 | 1237 | 75.2 | 67.1 |
| SwAV | 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* | ViT-S | 21 | 1007 | 71.4 | 66.6 |
| MoCov2* | ViT-S | 21 | 1007 | 72.7 | 64.4 |
| SwAV* | ViT-S | 21 | 1007 | 73.5 | 66.3 |
| DINO | ViT-S | 21 | 1007 | **77.0** | **74.5** |

For RN50 DINO is competitive

major gap

For ViT-S DINO yields gains

minor gap

* baseline re-implemented by DINO authors

Param. - model parameters (millions) im/s - on NVIDA V100 (batch size 128)

## Comparing across architectures

Compare to existing self-supervised methods with different architectures

| Method | Arch. | Param. | im/s | Linear | k-NN |
|---|---|---|---|---|---|
| *Comparison across architectures* | | | | | |
| SCLR | RN50w4 | 375 | 117 | 76.8 | 69.3 |
| SwAV | RN50w2 | 93 | 384 | 77.3 | 67.3 |
| BYOL | RN50w2 | 93 | 384 | 77.4 | – |
| DINO | ViT-B/16 | 85 | 312 | 78.2 | 76.1 |
| SwAV | RN50w5 | 586 | 76 | 78.5 | 67.1 |
| BYOL | RN50w4 | 375 | 117 | 78.6 | – |
| BYOL | RN200w2 | 250 | 123 | 79.6 | 73.9 |
| DINO | ViT-S/8 | 21 | 180 | 79.7 | **78.3** |
| SCLRv2 | RN152w3+SK | 794 | 46 | 79.8 | 73.1 |
| DINO | ViT-B/8 | 85 | 63 | **80.1** | 77.4 |

reduced patch size, fewer params

reduced patch size


**Image credits/References:**

M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
H. Touvron et al., "Training data-efficient image transformers & distillation through attention", ICML (2021)
(SCLR) T. Chen et al. "A simple framework for contrastive learning of visual representations" ICML (2020)
(MoCov2) X. Chen et al., "Improved baselines with momentum contrastive learning", arxiv (2020)
(InfoMin) Y. Tian et al. "What makes for good views for contrastive learning?" NeurIPS (2020)
(BarlowT) J. Zbontar et al., "Barlow twins: Self-supervised learning via redundancy reduction", ICML (2021)
(OBoW) S. Gidaris et al., "Obow: Online bag-of-visual-words generation for self-supervised learning", CVPR (2021)
(BYOL) J-B Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(DCv2/SwAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)

# Experiments - Properties of Self-Supervised ViT

## Image Retrieval

Compare on Revisited Oxford and Revisited Paris retrieval datasets

| Pretrain | Arch. | Pretrain | $\mathcal{R}$Ox | | $\mathcal{R}$Par | |
|---|---|---|---|---|---|---|
| | | | M | H | M | H |
| Sup. | RN101+R-MAC | ImNet | 49.8 | 18.5 | 74.0 | **52.1** |
| Sup. | ViT-S/16 | ImNet | 33.5 | 8.9 | 63.0 | 37.2 |
| DINO | ResNet-50 | ImNet | 35.4 | 11.1 | 55.9 | 27.5 |
| DINO | ViT-S/16 | ImNet | 41.8 | 13.7 | 63.1 | 34.4 |
| DINO | ViT-S/16 | GLDv2 | **51.5** | **24.3** | **75.3** | 51.6 |

M - medium split  H - hard split  GLDv2: Google Landmarks Dataset v2

## Copy detection

Evaluate on copy detection (recognise images distorted by blur, insertions etc.)

**Benchmark:** Copydays dataset (strong subset) 10K YFC100M distractors

DINO features: concat [CLS] token with GeM pooled patch tokens whiten

| Method | Arch. | Dim. | Resolution | mAP |
|---|---|---|---|---|
| Multigrain | ResNet-50 | 2048 | $224^2$ | 75.1 |
| Multigrain | ResNet-50 | 2048 | largest side 800 | 82.5 |
| Supervised | ViT-B/16 | 1536 | $224^2$ | 76.4 |
| DINO | ViT-B/16 | 1536 | $224^2$ | 81.7 |
| DINO | ViT-B/8 | 1536 | $320^2$ | **85.5** |


**Image credits/References:**

M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

(Paris) J. Philbin et al., "Lost in quantization: Improving particular object retrieval in large scale image databases", CVPR (2008)

(Revisited Oxford/Paris) F. Radenović et al., "Revisiting oxford and paris: Large-scale image retrieval benchmarking", *CVPR (2018)*

(Retrieval baseline features) J. Revaud et al., "Learning with average precision: Training image retrieval with a listwise loss", ICCV (2019)

(GLDv2) T. Weyand et al., "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval", CVPR (2020)

(Copydays dataset) M. Douze et al., "Evaluation of gist descriptors for web-scale image search", CIVR (2009)

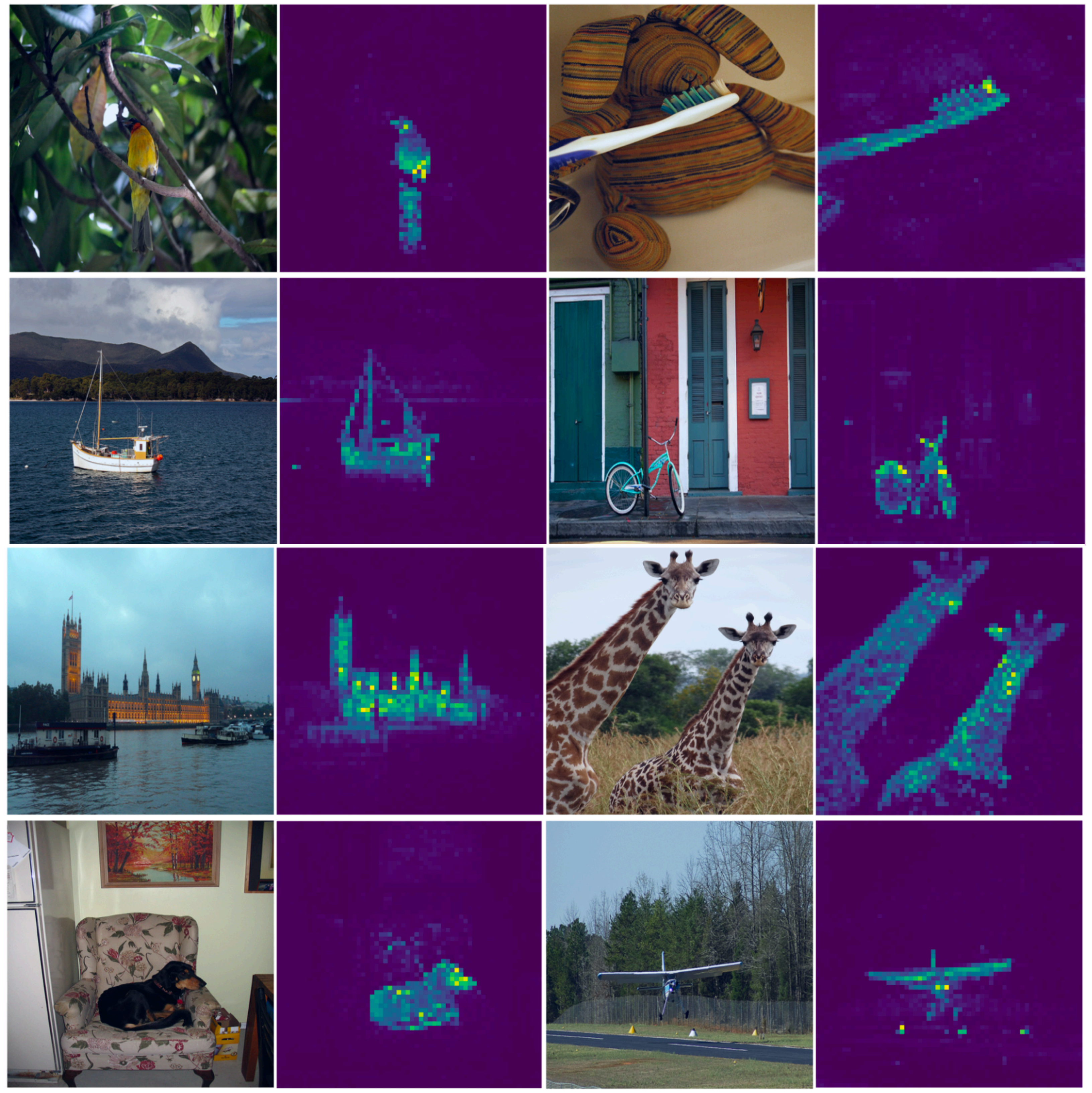
(GeM) F. Radenović, et al. "Fine-tuning CNN image retrieval with no human annotation", TPAMI (2018)

(MultiGrain) M. Berman et al., "Multigrain: a unified image embedding for classes and instances", arxiv (2019)

# Experiments - Semantic Layout of Scenes

## Qualitative Results

Self-attention from [CLS] token on heads of the last layer of ViT-S/8



Attention appears to be class specific

## Video Instance segmentation

Evaluate video instance segmentation on DAVIS-2017

**Protocol:** segment scenes with nearest neighbours between frames

| 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 | I/D/Y | RN50 | 81.8 | 79.2 | 84.3 |
| *Self-supervised* | | | | | |
| CT | VLOG | RN50 | 48.7 | 46.4 | 50.0 |
| MAST | YT-VOS | RN18 | 65.5 | 63.3 | 67.6 |
| STC | 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** |

$\mathcal{J}_m$ - mean region similarity $\mathcal{F}_m$ - mean contour-based accuracy

Frame resolution: 480 pixels

As DINO is not fine-tuned it must have retained some spatial information


(DAVIS-2017) J. Pont-Tuset et al., "The 2017 DAVIS challenge on video object segmentation", arxiv (2017)
(STM) S. W. Oh et al., "Video object segmentation using space-time memory networks", ICCV (2019)
(CT) X. Wang et al., "Learning correspondence from the cycle-consistency of time," CVPR (2019)
(MAST) Z. Lai et al., "MAST: A memory-augmented self-supervised tracker", CVPR (2020)
(STC) A. Jabri, "Space-time correspondence as a contrastive random walk", NeurIPS (2020)

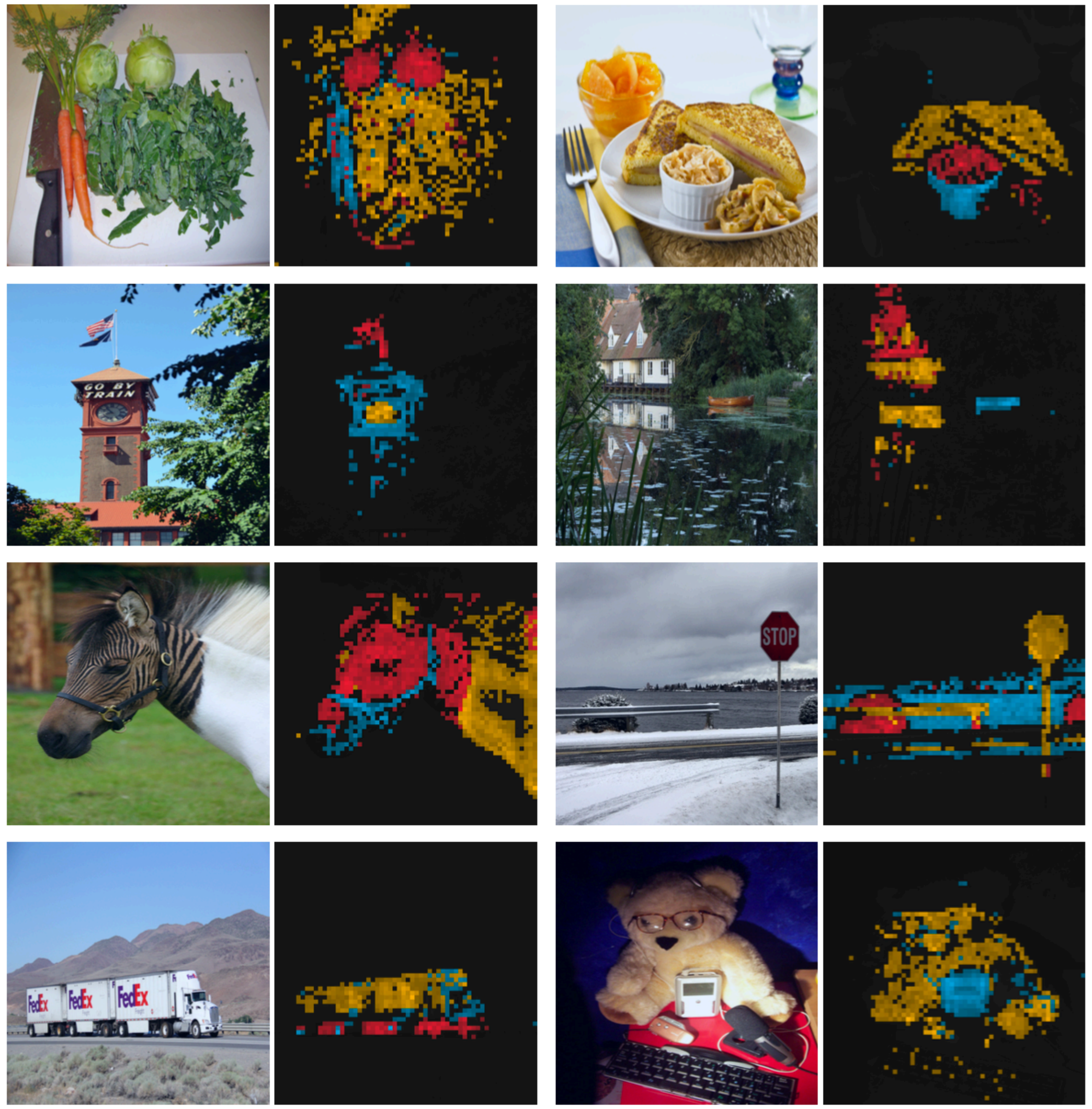


**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

# Experiments - probing the self-attention map

## Qualitative Results

Self-attention from [CLS] token (different heads, different colours) taken from the last layer of ViT-S/8




## Comparing supervised vs DINO segmentation

Visualise masks by thresholding [CLS] self-attention maps to keep 60% of mass


Quantify mask quality via Jaccard similarity between ground-truth and masks on Pascal VOC 2012

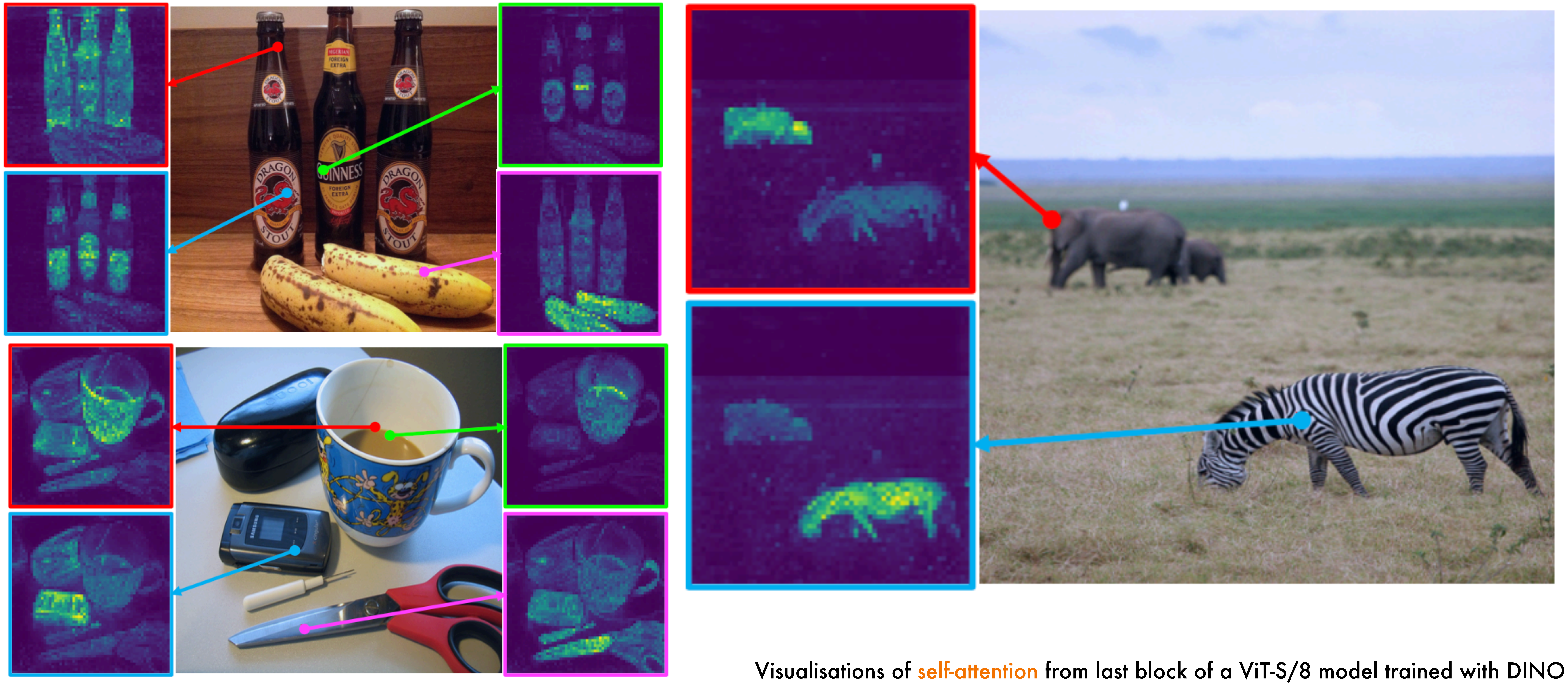
| | Random | Supervised | DINO |
|---|---|---|---|
| ViT-S/16 | 22.0 | 27.3 | 45.9 |
| ViT-S/8 | 21.8 | 23.7 | 44.7 |

**Note:** can obtain segmentations from self-sup. CNNs, but need dedicated methods e.g. using gradients/attribution propagation, Gur et al. (2021)


**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
(Pascal VOC) M. Everingham et al., "The pascal visual object classes (voc) challenge", IJCV (2010)
S. Gur et al., "Visualization of supervised and self-supervised neural networks via attribution guided factorization", AAAI (2021)

# Experiments - visualisation of reference points

Qualitative Results




Visualisations of self-attention from last block of a ViT-S/8 model trained with DINO

**Image credits/References:**

M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

# Experiments - class visualisation with t-SNE

## Qualitative Results

Represent each ImageNet class with average feature for validation images

Reduce to 30D with PCA

Run t-SNE for 5000 iterations (perplexity 20, LR 200)

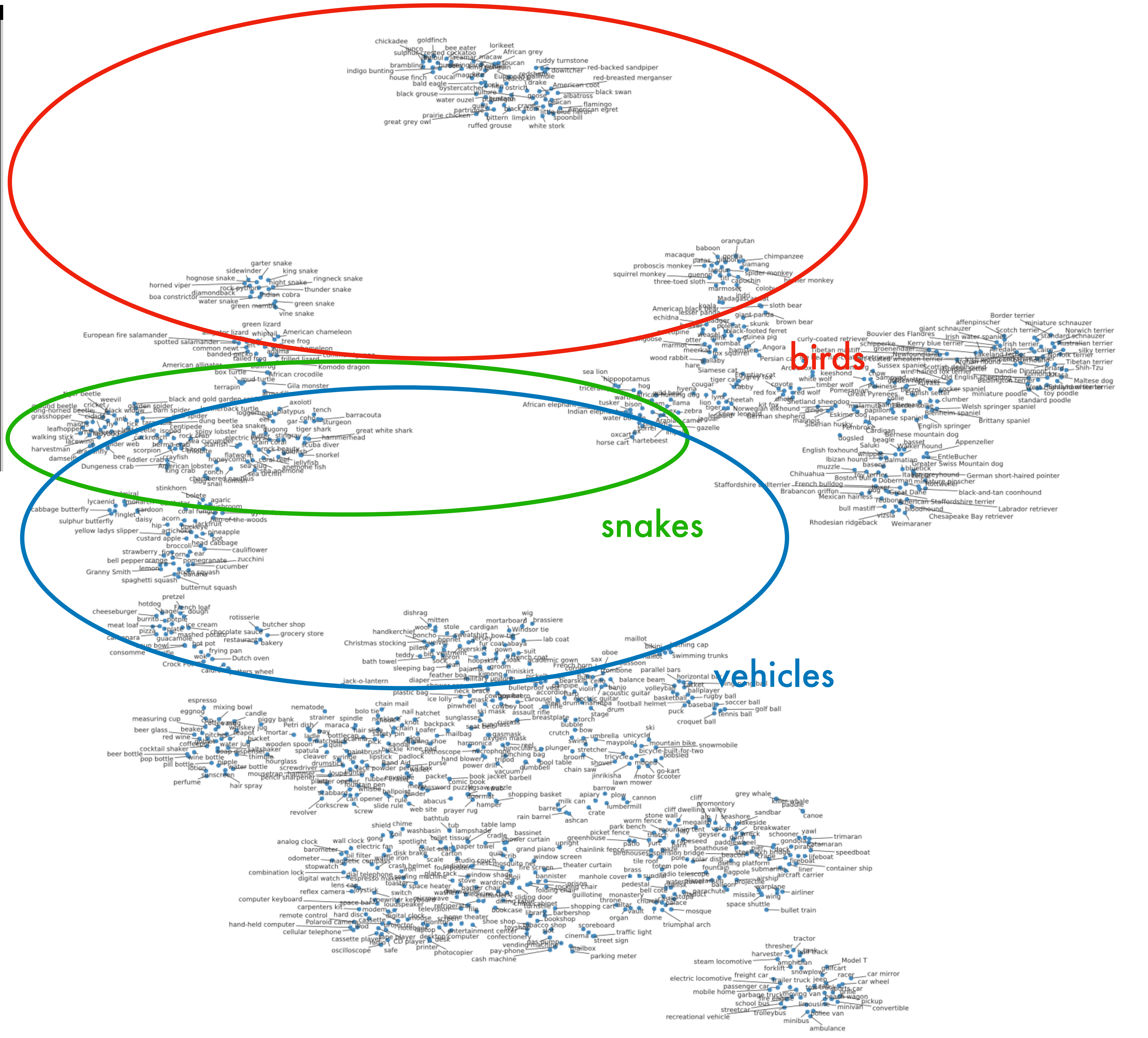



**Image credits/References:**

L. Van der Maaten et al., "Visualizing data using t-SNE", JMLR (2008)

M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

# Experiments - transfer learning on downstream tasks

## Transfer learning

To evaluate feature quality, DINO features are compared to supervised features with the same architecture trained with ImageNet labels

The transfer learning protocol follows DeiT across 8 tasks and compares to the supervised baseline provided by DeIT

| | $\text{Cifar}_{10}$ | $\text{Cifar}_{100}$ | $\text{INat}_{18}$ | $\text{INat}_{19}$ | Flwrs | Cars | INet |
|---|---|---|---|---|---|---|---|
| *ViT-S/16* | | | | | | | |
| Sup. | **99.0** | 89.5 | 70.7 | 76.6 | 98.2 | 92.1 | 79.9 |
| DINO | **99.0** | **90.5** | **72.0** | **78.2** | **98.5** | **93.0** | **81.5** |
| *ViT-B/16* | | | | | | | |
| Sup. | 99.0 | 90.8 | **73.2** | 77.7 | 98.4 | 92.1 | 81.8 |
| DINO | **99.1** | **91.7** | 72.6 | **78.6** | **98.8** | **93.0** | **82.8** |

As observed in previous works, self-supervised features appear to transfer better than supervised features

DINO attains notable gains on ImageNet itself

**Image credits/References:**
(DeiT) H. Touvron et al., "Training data-efficient image transformers & distillation through attention", ICML (2021)
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
($\text{Cifar}_{10}$/$\text{Cifar}_{100}$) A. Krizhevsky, "Learning multiple layers of features from tiny images", (2009)
($\text{INat}_{18}$/$\text{INat}_{19}$) G. Horn et al., "The inaturalist challenge 2018 dataset". arxiv (2018)
(Flwrs) M-E. Nilsback et al., "Automated flower classification over a large number of classes" ICVGIP (2008)
(Cars) J. Krause et al., "3d object representations for fine-grained categorization", ICCVW (2013)
(INet) O. Russakovsky et al., "Imagenet large scale visual recognition challenge", IJCV (2015)

# Experiments: low-shot learning on ImageNet

## Low-shot learning on ImageNet

Evaluate features on low-shot learning on ImageNet

Train logistic regression (using `cyanure`) on frozen features

| Method | Arch | Param. | Top 1 1% | Top 1 10% |
|---|---|---|---|---|
| *Self-supervised pretraining with finetuning* | | | | |
| UDA | RN50 | 23 | – | 68.1 |
| SimCLRv2 | RN50 | 23 | 57.9 | 68.4 |
| BYOL | RN50 | 23 | 53.2 | 68.8 |
| SwAV | RN50 | 23 | 53.9 | 70.2 |
| SimCLRv2 | RN50w4 | 375 | 63.0 | 74.4 |
| BYOL | RN200w2 | 250 | 71.2 | 77.7 |
| *Semi-supervised methods* | | | | |
| SimCLRv2+KD | RN50 | 23 | 60.0 | 70.5 |
| SwAV+CT | RN50 | 23 | – | 70.8 |
| FixMatch | RN50 | 23 | – | 71.5 |
| MPL | RN50 | 23 | – | 73.9 |
| SimCLRv2+KD | RN152w3+SK | 794 | 76.6 | 80.9 |
| *Frozen self-supervised features* | | | | |
| DINO -FROZEN | ViT-S/16 | 21 | 64.5 | 72.2 |


**Image credits/References:**

J. Mairal, "Cyanure: An open-source toolbox for empirical risk minimization for python, c++, and soon more", arxiv (2019)
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
(UDA) Q. Xie, et al., "Unsupervised data augmentation for consistency training", NeurIPS (2020)
(SimCLRv2) T. Chen et al., "Big self-supervised models are strong semi-supervised learners", NeurIPS (2020)
(BYOL) J-B Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(SwAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)
(SwAV+CT) M. Assran et al., "Recovering petaflops in contrastive semi-supervised learning of visual representations", arxiv (2020)
(FixMatch) K. Sohn et al., "Fixmatch: Simplifying semi-supervised learning with consistency and confidence", NeurIPS (2020)
(MPL) H. Pham et al., "Meta pseudo labels", CVPR (2021)

# Outline

- Motivation
- Related work
- DINO framework
- Evaluation protocols
- Experiments
- Ablations
- Summary

# Ablation studies

## Framework components

Which components contribute to DINO's performance?

Train ViT-S/16 for 300 epochs on ImageNet

| | Method | Mom. | SK | MC | Loss | Pred. | $k$-NN | Lin. |
|---|---|---|---|---|---|---|---|---|
| 1 | DINO | ✓ | ✗ | ✓ | CE | ✗ | 72.8 | 76.1 |
| 2 | | ✗ | ✗ | ✓ | CE | ✗ | 0.1 | 0.1 |
| 3 | | ✓ | ✓ | ✓ | CE | ✗ | 72.2 | 76.0 |
| 4 | | ✓ | ✗ | ✗ | CE | ✗ | 67.9 | 72.5 |
| 5 | | ✓ | ✗ | ✓ | MSE | ✗ | 52.6 | 62.4 |
| 6 | | ✓ | ✗ | ✓ | CE | ✓ | 71.8 | 75.6 |
| 7 | BYOL | ✓ | ✗ | ✗ | MSE | ✓ | 66.6 | 71.4 |
| 8 | MoCov2 | ✓ | ✗ | ✗ | INCE | ✗ | 62.0 | 71.6 |
| 9 | SwAV | ✗ | ✓ | ✓ | CE | ✗ | 64.7 | 71.8 |

Mom. - Momentum
SK - Sinkhorn-Knopp
MC - Multi-Crop
Pred. - Student Predictor
Lin. - Linear probe

## Self-supervised backbone influence

Backbones: Train both ResNet-50 and ViT-S/16 for 300 epochs on ImageNet

| | ResNet-50 | | ViT-small | |
|---|---|---|---|---|
| Method | Linear | $k$-NN | Linear | $k$-NN |
| MoCo-v2 | 71.1 | 62.9 | 71.6 | 62.0 |
| BYOL | 72.7 | 65.4 | 71.4 | 66.6 |
| SwAV | 74.1 | 65.4 | 71.8 | 64.7 |
| DINO | **74.5** | **65.6** | **76.1** | **72.8** |

DINO is particularly effective for self-supervised training of vision transformers.


**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
(SwAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)
(BYOL) J-B Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(MoCov2) X. Chen et al., "Improved baselines with momentum contrastive learning", arxiv (2020)

# Ablation studies - methodology comparison

## Relationship to MoCo-v2 and BYOL

Fine-grained analysis of components (top-1 linear probe accuracy)

| | Method | Loss | multi-crop | Center. | BN | Pred. | Top-1 |
|---|---|---|---|---|---|---|---|
| 1 | DINO | CE | ✓ | ✓ | | | 76.1 |
| 2 | – | MSE | ✓ | ✓ | | | 62.4 |
| 3 | – | CE | ✓ | ✓ | | ✓ | 75.6 |
| 4 | – | CE | | ✓ | | | 72.5 |
| 5 | MoCov2 | INCE | | | ✓ | | 71.4 |
| 6 | | INCE | ✓ | | ✓ | | 73.4 |
| 7 | BYOL | MSE | | | ✓ | ✓ | 71.4 |
| 8 | – | MSE | | | ✓ | | 0.1 |
| 9 | – | MSE | | ✓ | | | 52.6 |
| 10 | – | MSE | ✓ | | ✓ | ✓ | 64.8 |

Center. - Centering operator
BN - Batch Normalization in the projection heads
Pred. - Student Predictor

## Relationship to SwAV

Effect of momentum and teacher output operation

| | Method | Momentum | Operation | Top-1 |
|---|---|---|---|---|
| 1 | DINO | ✓ | Centering | 76.1 |
| 2 | – | ✓ | Softmax(batch) | 75.8 |
| 3 | – | ✓ | Sinkhorn-Knopp | 76.0 |
| 4 | – | | Centering | 0.1 |
| 5 | – | | Softmax(batch) | 72.2 |
| 6 | SwAV | | Sinkhorn-Knopp | 71.8 |

## Details on Softmax(batch) variant

Implementation of Sinkhorn-Knopp used in SwAV:

```
# x is n-by-K
# tau is Sinkhorn regularization param
x = exp(x / tau)
for _ in range(num_iters): # 1 iter of Sinkhorn
  # total weight per dimension (or cluster)
  c = sum(x, dim=0, keepdim=True)
  x /= c

  # total weight per sample
  n = sum(x, dim=1, keepdim=True)
  # x sums to 1 for each sample (assignment)
  x /= n
```

Softmax(batch) variant (equivalent to `num_iters=1`):

```
x = softmax(x / tau, dim=0)
x /= sum(x, dim=1, keepdim=True)
```


**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
(BYOL) J-B Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(MoCov2) X. Chen et al., "Improved baselines with momentum contrastive learning", arxiv (2020)
(SwAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)
(Sinkhorn-Knopp) M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport", NeurIPS (2013)

# Ablation studies - k-NN performance and pretraining

## k-NN classification vs linear probe performance

Compare ResNet-50 and ViT-S (frozen DINO features)

No data augmentation is used when extracting features

| | Logistic | | | k-NN | | |
|---|---|---|---|---|---|---|
| | RN50 | ViT-S | Δ | RN50 | ViT-S | Δ |
| Inet 100% | 72.1 | 75.7 | 3.6 | 67.5 | 74.5 | 7.0 |
| Inet 10% | 67.8 | 72.2 | 4.4 | 59.3 | 69.1 | 9.8 |
| Inet 1% | 55.1 | 64.5 | 9.4 | 47.2 | 61.3 | 14.1 |
| Pl. 10% | 53.4 | 52.1 | -1.3 | 46.9 | 48.6 | 1.7 |
| Pl. 1% | 46.5 | 46.3 | -0.2 | 39.2 | 41.3 | 2.1 |
| VOC07 | 88.9 | 89.2 | 0.3 | 84.9 | 88.0 | 3.1 |
| FLOWERS | 95.6 | 96.4 | 0.8 | 87.9 | 89.1 | 1.2 |
| Average Δ | | | 2.4 | | | 5.6 |

DINO ViT-S features yield a particularly good k-NN classifier

## Self-supervised ImageNet pretraining of ViT

Compare supervised ViT-B/16 on ImageNet

| Pretraining | | | | | |
|---|---|---|---|---|---|
| method | data | res. | tr. proc. | Top-1 | |
| *Pretrain on additional data* | | | | | |
| MPP | JFT-300M | 384 | ViT | 79.9 | |
| Supervised | JFT-300M | 384 | ViT | 84.2 | |
| *Train with additional model* | | | | | |
| Rand. init. | - | 224 | DeiT | 83.4 | (RegNetY) |
| *No additional data nor model* | | | | | |
| Rand. init. | - | 224 | ViT | 77.9 | |
| Rand. init. | - | 224 | DeiT | 81.8 | |
| Supervised | ImNet | 224 | DeiT | 81.9 | |
| DINO | ImNet | 224 | DeiT | 82.8 | |

res. - image resolution

tr. proc. - training procedure (data augmentation and optimisation)


**Image credits/References:**

M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

(Inet) O. Russakovsky et al., "Imagenet large scale visual recognition challenge", IJCV (2015)

(Pl) B. Zhou et al., "Learning deep features for scene recognition using places database", NeurIPS (2014)

(VOC07) M. Everingham et al., "The pascal visual object classes (voc) challenge", IJCV (2010)

(FLOWERS) M-E. Nilsback et al., "Automated flower classification over a large number of classes" ICVGIP (2008)

(ViT) A. Dosovitskiy, et al. "An image is worth 16x16 words: Transformers for image recognition at scale", ICLR (2021)

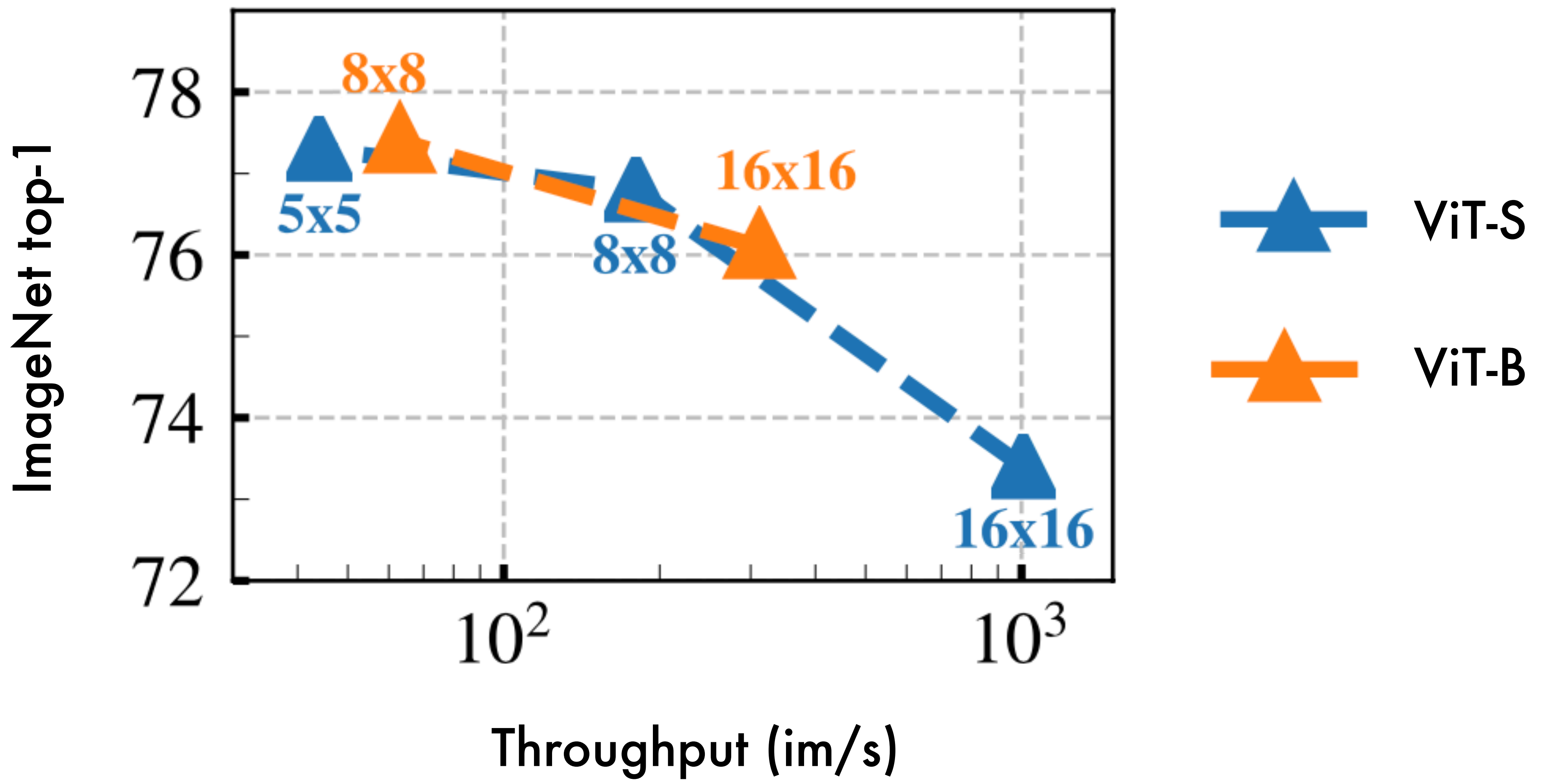
(DeiT) H. Touvron et al., "Training data-efficient image transformers & distillation through attention", ICML (2021)

(RegNetY) I. Radosavovic et al., "Designing network design spaces", CVPR (2020)

# Ablation studies - patch size

## Influence of patch size

Compare k-NN performance of ViT-S and ViT-B at different patch sizes




Smaller patches yield gains with no additional parameters

Smaller patches result in lower throughput: 44 ims/s (5x5) vs 180 ims/s (8x8)

**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

# Ablation studies - projection heads

## Overview

Like SimCLR, DINO benefits from a projection head

Follow an approach inspired by SwAV:

- n-layer MLP (2048D hidden units, GELU activations)
- Last layer (no GELU) $l_2$ norm, WeightNorm on FC

## BN-free system

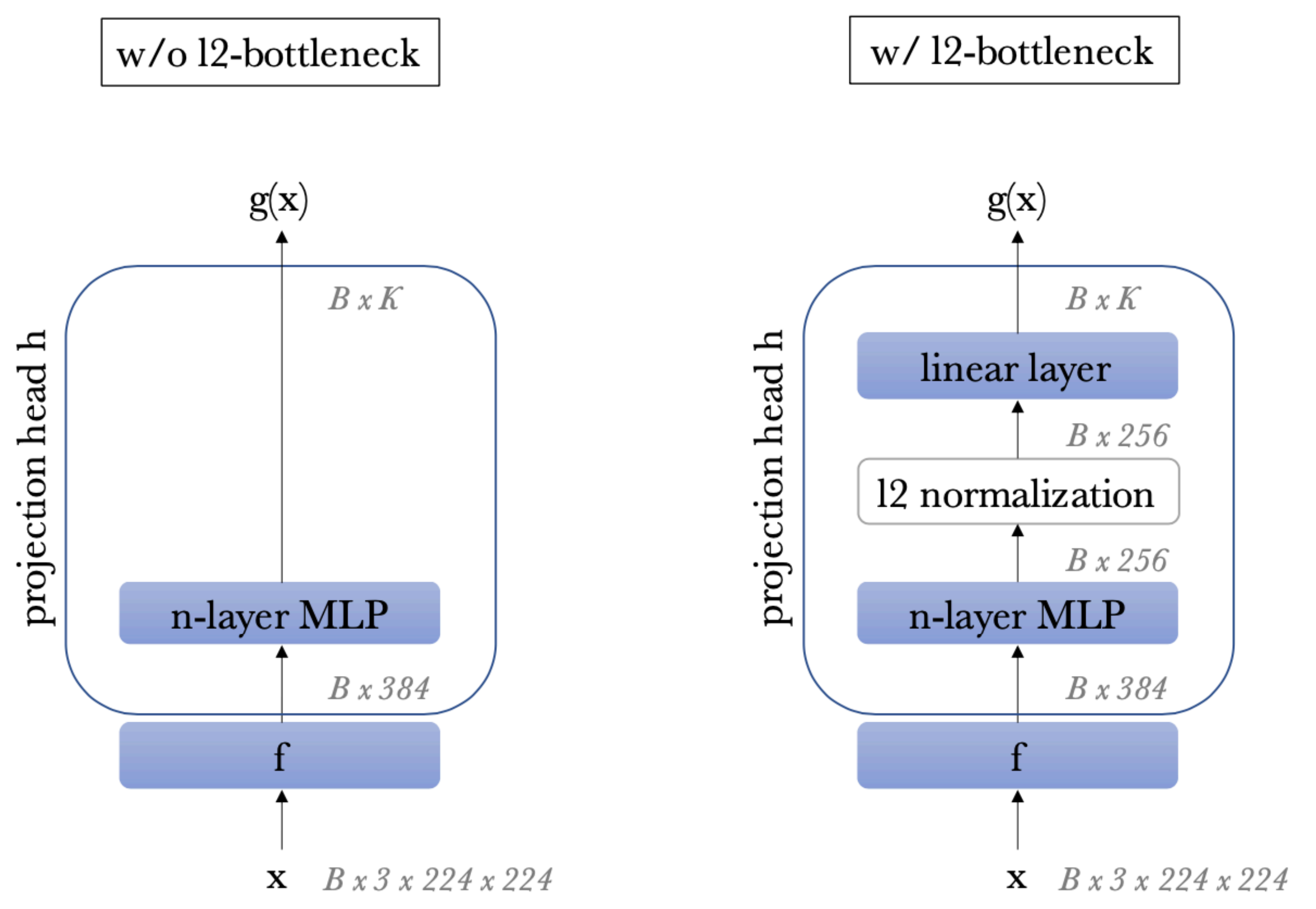
No batch norm is used in DINO ViT projection heads

System is therefore "BN-free"

| ViT-S, 100 epochs | heads w/o BN | heads w/ BN |
|---|---|---|
| $k$-NN top-1 | 69.7 | 68.6 |

BN-free: simpler and no need for BN synchronisation

## L2-normalisation bottleneck




Evaluate DINO ViT-S/16 on ImageNet (K = 4096)

| # proj. head linear layers | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| w/ l2-norm bottleneck | – | 62.2 | 68.0 | 69.3 |
| w/o l2-norm bottleneck | 61.6 | 62.9 | 0.1 | 0.1 |

Takeaway: the l2 bottleneck is essential

## Output dimension

Compare projection head output dimensions

For each output dimension size, bottleneck is 256D

| $K$ | 1024 | 4096 | 16384 | 65536 | 262144 |
|---|---|---|---|---|---|
| $k$-NN top-1 | 67.8 | 69.3 | 69.2 | 69.7 | 69.1 |

Using a large dimensionality helps (up to a point)

## GELU activations

Compare projection head activation functions

**Note:** by default GELU is used in ViT

| ViT-S, 100 epochs | heads w/ GELU | heads w/ ReLU |
|---|---|---|
| $k$-NN top-1 | 69.7 | 68.9 |

GELU is preferable to ReLU for the projection head


**Image credits/References:**
(SimCLR) T. Chen et al., "A simple framework for contrastive learning of visual representations", ICML (2020)
(WeightNorm) T. Salimans et al., "Weight normalization: A simple reparameterization to accelerate training of deep neural networks" NeurIPS (2016)
(Batch Norm) S. Ioffe et al., "Batch normalization: Accelerating deep network training by reducing internal covariate shift", ICML (2015)

# Ablation studies - choice of teacher network

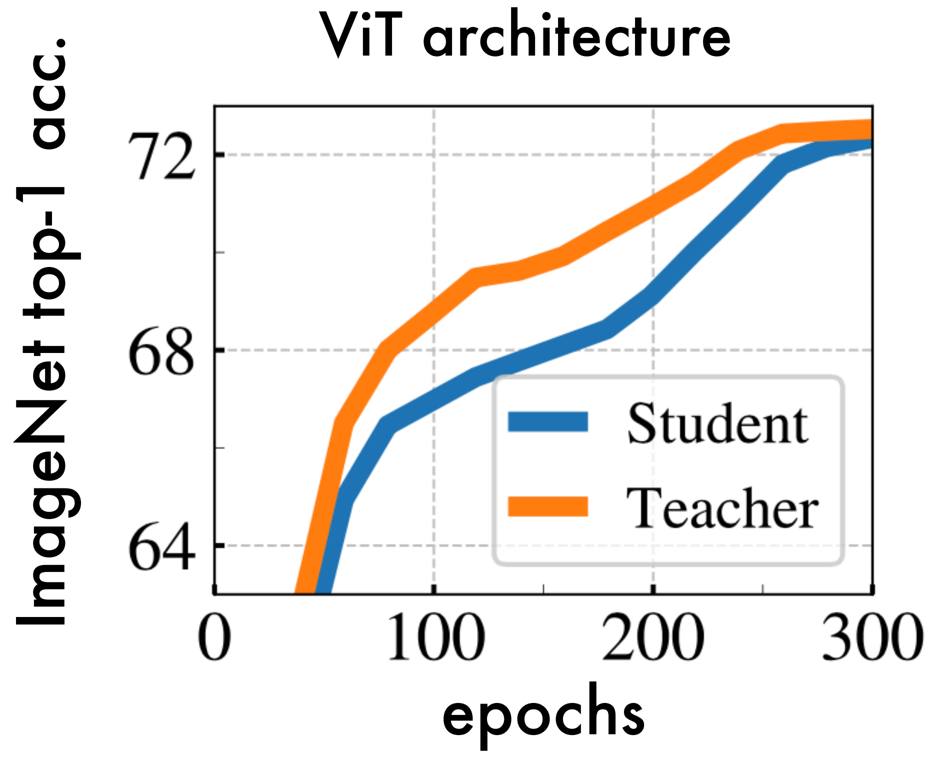
## Building the teacher from the student

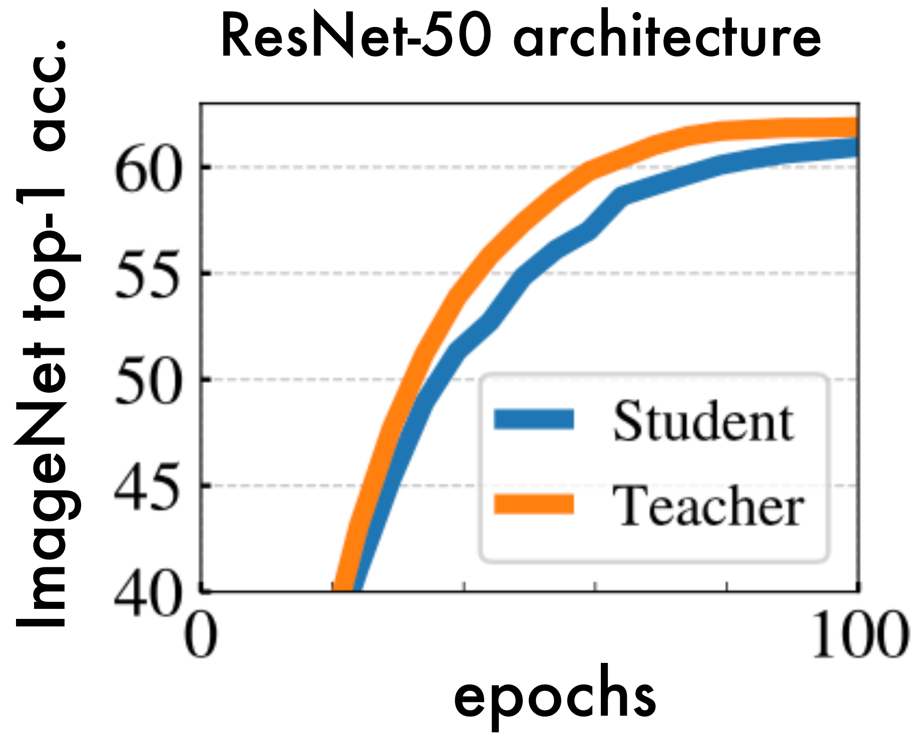
Various strategies can be used to build the teacher from the student

Performance is compared on ImageNet top-1accuracy (with k-NN)

| Teacher | Top-1 |
|---|---|
| Student copy | 0.1 |
| Previous iter | 0.1 |
| Previous epoch | 66.6 |
| Momentum | 72.8 |

## Training dynamics




Interpretation: momentum teacher in DINO is a form of Polyak-Ruppert averaging

This provides a (higher-quality) model ensemble that guides the student


**Image credits/References:**

M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

D. Ruppert, "Efficient estimations from a slowly convergent Robbins-Monro process" (1988)

B. T. Polyak et al., "Acceleration of stochastic approximation by averaging", SICON (1992)

# Ablation studies - avoiding collapse

## Avoiding the collapse of representations

There are **two** forms of collapse that can occur during pretraining:

- collapse to a uniform output along all dimensions
- collapse to a vector dominated by only one dimension

Centring avoids collapse along one dimension but encourages uniform output

Sharpening avoids uniform output but encourages collapse along one dimension

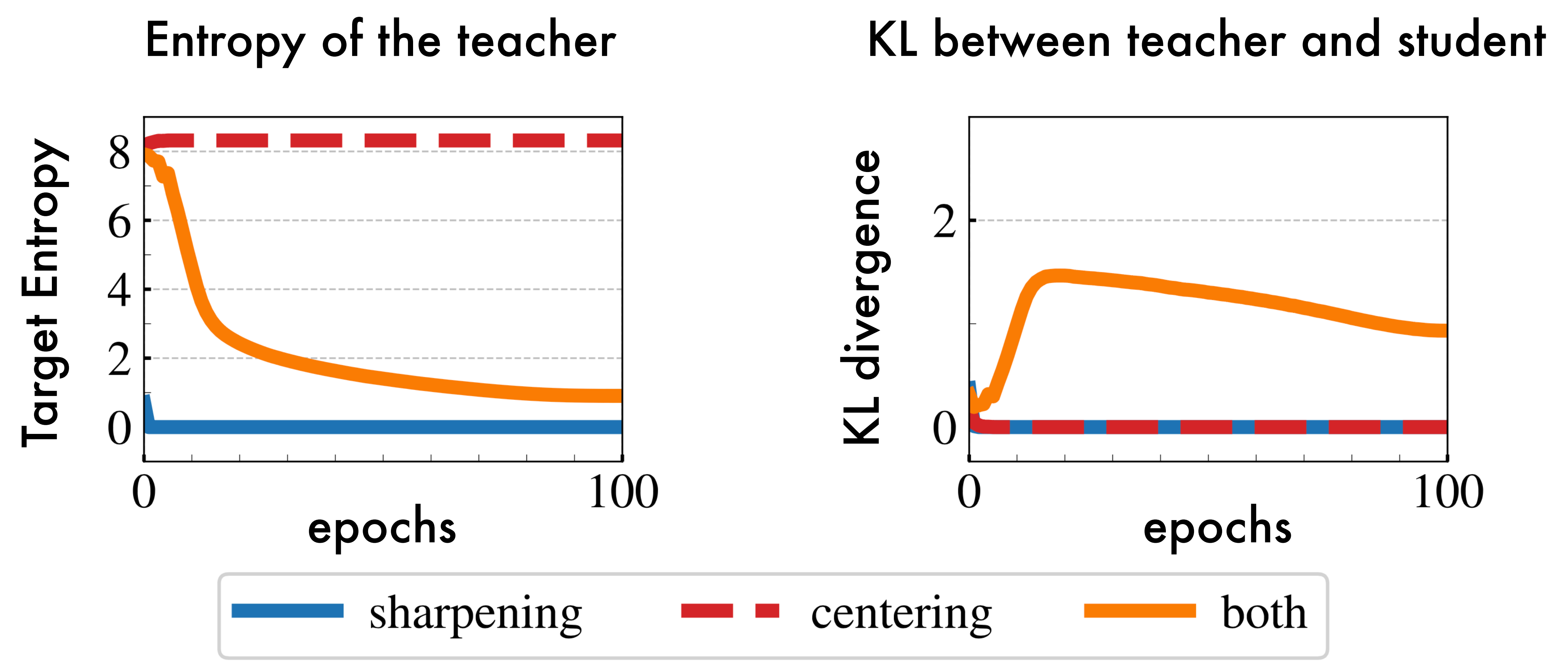
This can be seen by decomposing the cross-entropy between the distributions:

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

When KL term is equal to zero, the two distributions are identical

This indicates the outputs are constant, so a collapse has occurred

## Evolution of distributions




The entropy converges to either 0 (no centring) or $-\log(1/K)$ (no sharpening)

KL-divergence converges to zero if either operation is missing


**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

# Ablation studies: optimisation hyperparameters

## Online centring

Influence of the momentum hyperparameter for centre updates:

| $m$ | 0 | 0.9 | 0.99 | 0.999 |
|---|---|---|---|---|
| $k$-NN top-1 | 69.1 | 69.7 | 69.4 | 0.1 |

Collapse!

## Sharpening

Influence of the teacher softmax temperature $\tau_t$

| $\tau_t$ | 0 | 0.02 | 0.04 | 0.06 | 0.08 | $0.04 \rightarrow 0.07$ |
|---|---|---|---|---|---|---|
| $k$-NN top-1 | 43.9 | 66.7 | 69.6 | 68.7 | 0.1 | 69.7 |

linear warmup for 30 epochs

## Longer training

Influence of training more epochs

| DINO ViT-S | 100-ep | 300-ep | 800-ep |
|---|---|---|---|
| $k$-NN top-1 | 70.9 | 72.8 | 74.5 |

Note: for main comparison BYOL is only trained for 300 epochs

## Supervised vs self-supervised self-attention maps

Compare supervised vs self-supervised ViT-S/16 self-attention for segmentation

Evaluate on Pascal VOC 2012

Threshold to keep a fixed % of mass

Compute Jaccard similarity to ground truth

| ViT-S/16 weights | |
|---|---|
| Random weights | 22.0 |
| Supervised | 27.3 |
| DINO | 45.9 |
| DINO w/o multicrop | 45.1 |
| MoCo-v2 | 46.3 |
| BYOL | 47.8 |
| SwAV | 46.8 |

Key ingredient: Self-supervision + ViT

## Number of ViT-S heads

Influence of number of ViT-S heads on accuracy and throughput

| # heads | dim | dim/head | # params | im/sec | $k$-NN |
|---|---|---|---|---|---|
| 6 | 384 | 64 | 21 | 1007 | 72.8 |
| 8 | 384 | 48 | 21 | 971 | 73.1 |
| 12 | 384 | 32 | 21 | 927 | 73.7 |
| 16 | 384 | 24 | 21 | 860 | 73.8 |

For all other experiments in the paper, 6 heads are used.

**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
(Pascal VOC) M. Everingham et al., "The pascal visual object classes (voc) challenge", IJCV (2010)
(MoCo-v2) X. Chen et al., "Improved baselines with momentum contrastive learning", arxiv (2020)
(BYOL) J-B Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(SwAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)

# Ablations: multi-crop strategy

## Range of scales

Generate views with `RandomResizedCrop`

Select a scale hyperparameter, $s$:

- 2 global views in scale $(s, 1)$, resize to $224 \times 224$
- 6 local views with scale $(0.05, s)$, resize to $96 \times 96$

Arbitrary choice: non-overlapping scale ranges

| $(0.05, s), (s, 1), s$: | 0.08 | 0.16 | 0.24 | 0.32 | 0.48 |
|---|---|---|---|---|---|
| $k$-NN top-1 | 65.6 | 68.0 | 69.7 | 69.8 | 69.5 |

Note: best value ( $\approx 0.3$) is higher than SwAV ( $\approx 0.14$)

## Multi-crop for different frameworks

ViT-S/16 for 300 epochs with various frameworks

| crops | $2 \times 224^2$ | | $2 \times 224^2 + 6 \times 96^2$ | |
|---|---|---|---|---|
| eval | $k$-NN | linear | $k$-NN | linear |
| BYOL | 66.6 | 71.4 | 59.8 | 64.8 |
| SwAV | 60.5 | 68.5 | 64.7 | 71.8 |
| MoCo-v2 | 62.0 | 71.6 | 65.4 | 73.4 |
| DINO | **67.9** | **72.5** | **72.7** | **75.9** |

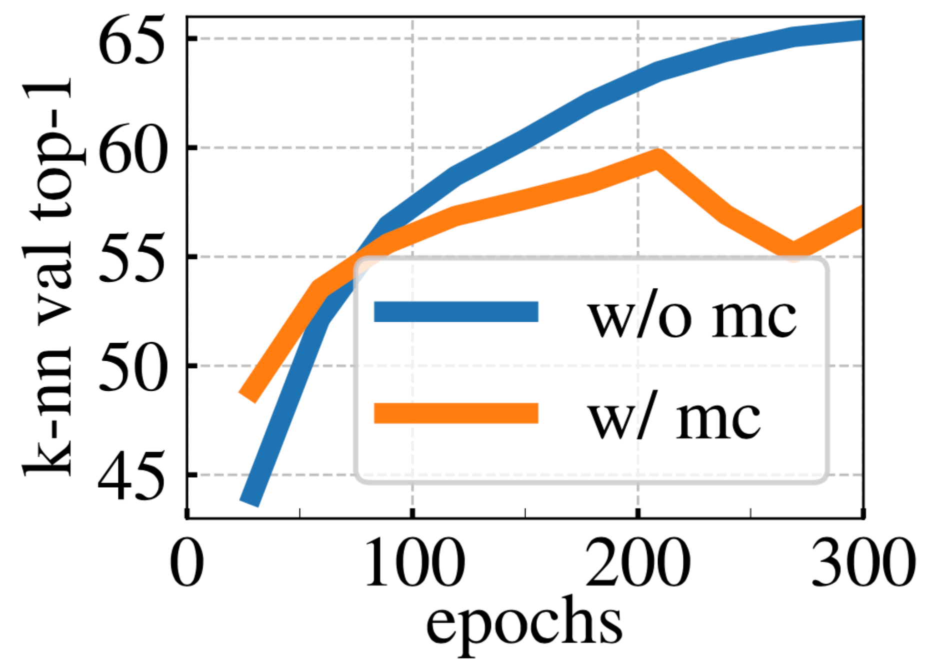
Multi-crop does not benefit all frameworks equally

DINO sees a major boost, while BYOL does worse

## Multi-crop with BYOL

Study BYOL performance with/without multi-cropping

BYOL (ViT-S)




Consistent effect across a range of hyperparameters


**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
(SwAV) M. Caron et al., "Unsupervised learning of visual features by contrasting cluster assignments", NeurIPS (2020)
(BYOL) J-B Grill et al., "Bootstrap your own latent-a new approach to self-supervised learning", NeurIPS (2020)
(MoCo-v2) X. Chen et al., "Improved baselines with momentum contrastive learning", arxiv (2020)

# Compute requirements and batch sizes

## Computational requirements for DINO

Measure time/GPU memory used to run ViT-S/16 on two 8-GPU machines

| | 100 epochs | | 300 epochs | | |
|---|---|---|---|---|---|
| multi-crop | top-1 | time | top-1 | time | mem. |
| $2\times224^2$ | 67.8 | 15.3h | 72.5 | 45.9h | 9.3G |
| $2\times224^2 + 2\times96^2$ | 71.5 | 17.0h | 74.5 | 51.0h | 10.5G |
| $2\times224^2 + 6\times96^2$ | 73.8 | 20.3h | 75.9 | 60.9h | 12.9G |
| $2\times224^2 + 10\times96^2$ | 74.6 | 24.2h | 76.1 | 72.6h | 15.4G |

Multi-crop improves the accuracy/running-time trade off (with extra memory)

Gains due to additional views see diminishing returns

## Training with small batches

Investigate the influence of batch size on feature quality

Evaluate ImageNet top-1 with k-NN after 100 epochs without multi-crop

Scale learning rate linearly with batch size (Goyal et al., 2019)

| bs | 128 | 256 | 512 | 1024 |
|---|---|---|---|---|
| top-1 | 57.9 | 59.1 | 59.6 | 59.9 |

DINO still works well with smaller batch sizes (some re-tuning required)

Note: this differs from contrastive approaches (for which batch size is critical)


**Image credits/References:**
M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)
P. Goyal et al., "Accurate, large minibatch sgd: Training imagenet in 1 hour", arxiv (2017)

# Outline

- Motivation
- Related work
- DINO framework
- Evaluation protocols
- Experiments
- Ablations
- Summary

# Summary

**DINO summary**

DINO can train a ViT with self-supervision to reach a comparable performance with the best CNNs

Two additional properties emerge from DINO:

- high-quality features for k-NN classification
- features contain information about scene layout (useful for segmentation)

DINO may provide a route to build a BERT-like model on ViT

Future work: self-supervised pretraining on uncurated images (Goyal et al., 2022)


**References:**

M. Caron et al., "Emerging properties in self-supervised vision transformers", ICCV (2021)

(BERT) J. Devlin et al., "Bert: Pre-training of deep bidirectional transformers for language understanding", NAACL-HLT (2019)

P. Goyal et al., "Vision models are more robust and fair when pretrained on uncurated images without supervision", arxiv (2022)