Emerging Properties in Self-Supervised Vision Transformers

Mathilde Caron , Hugo Touvron , Ishan Misra , Herve Jegou , Julien Mairal, Piotr Bojanowski, Armand Joulin

Facebook AI Research **Inria** **Sorbonne University**

Presenter: Huy Hoang Dang

Supervisor: David Hoffman

Quick Recap

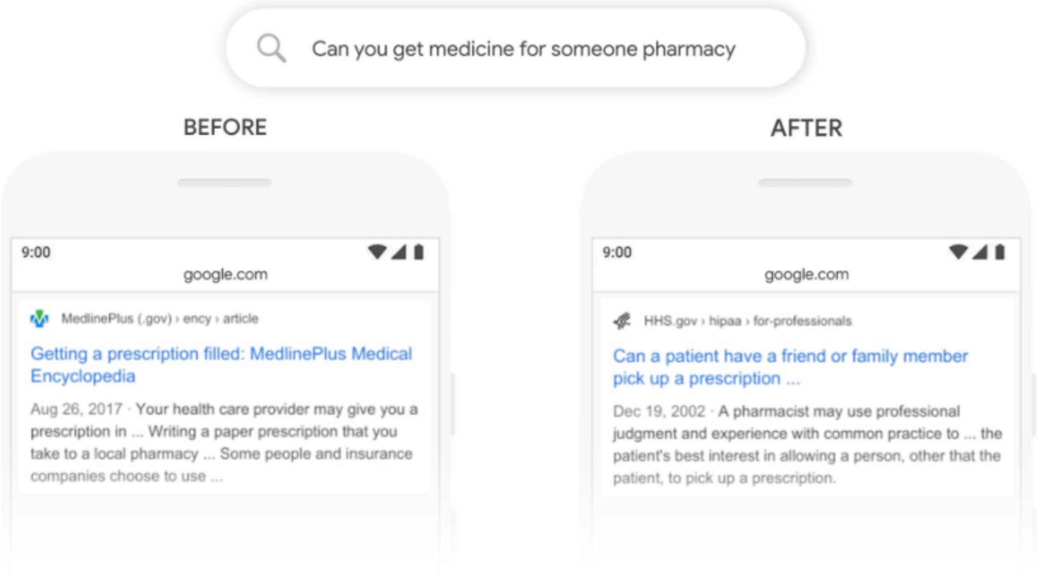
Why Self-Supervised Learning (SSL) ?

- Supervised learning requires a lot of labelled data
- Getting good quality labelled data is usually expensive and time-consuming

→ Motivation for SSL: Learning useful representations of the data by leveraging unlabelled data pool, which is easier to acquire

SSL has been successfully applied in NLP field
(e.g: BERT, GPT-3, etc.)

Why Self-Supervised Learning (SSL) ?

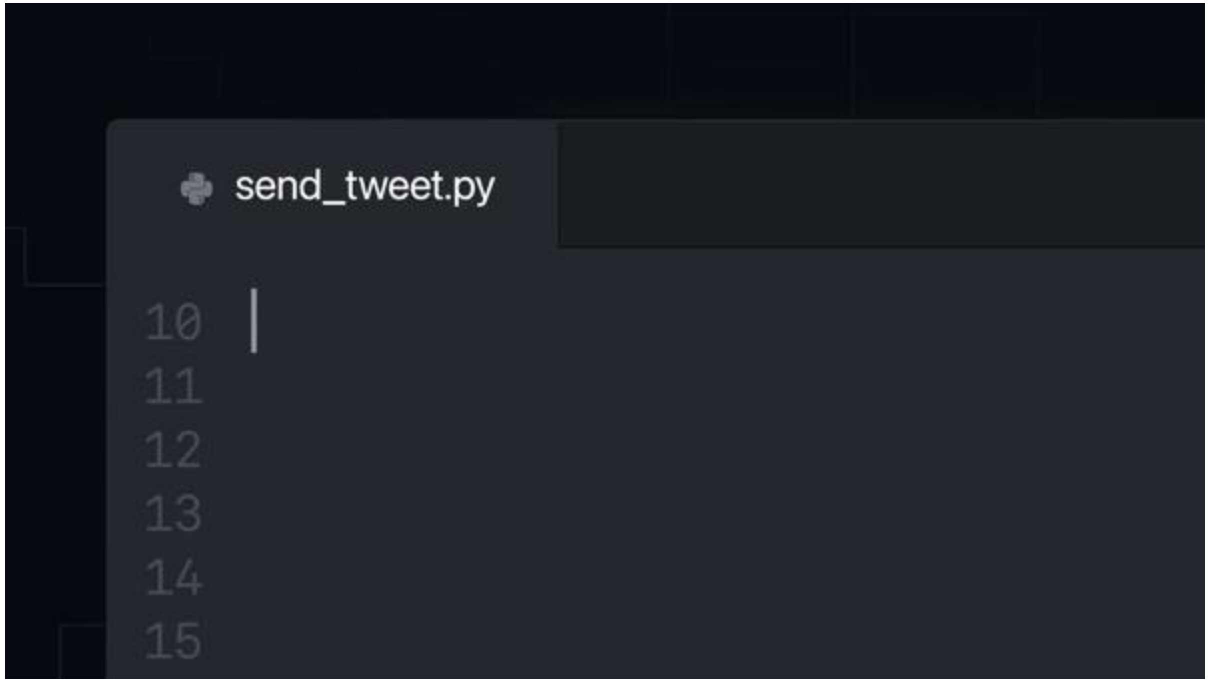


With the BERT model, we can better understand that “for someone” is an important part of this query, whereas previously we missed the meaning, with general results about filling prescriptions.

BERT model introduced by Google at the end of 2018

Image source:
https://blog.google/products/search/search-language-understanding-bert/

Why Self-Supervised Learning (SSL) ?



OpenAI Codex – descendant of *GPT-3*, used and finetuned for code generation in *Github Copilot*

Source:
https://copilot.github.com/

Contrastive Learning

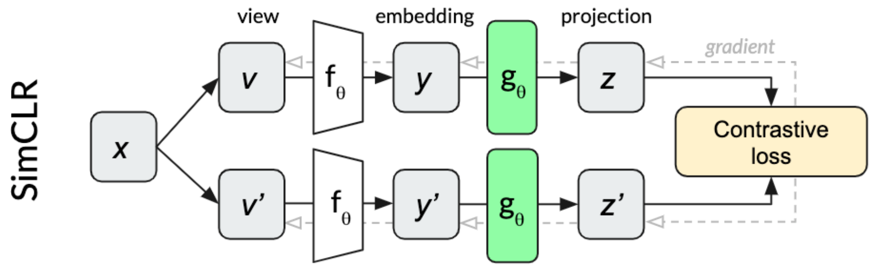
One direction for SSL is **contrastive learning**

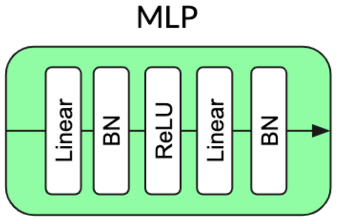
Goal of contrastive learning : To learn such an embedding space in which **similar sample pairs** **stay close** to each other while **dissimilar ones** are **far apart**

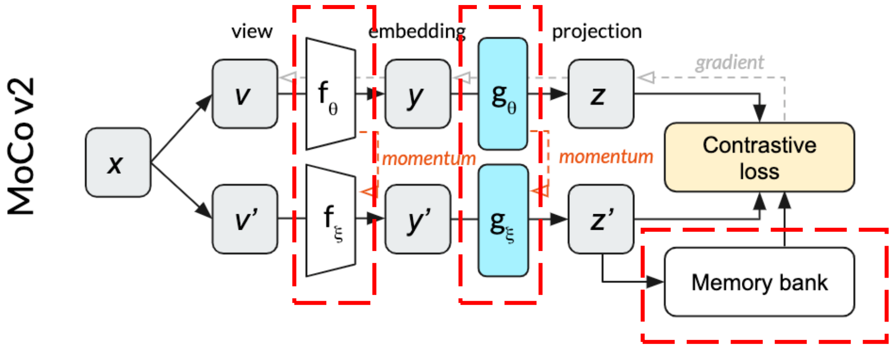
Examples of contrastive learning frameworks:

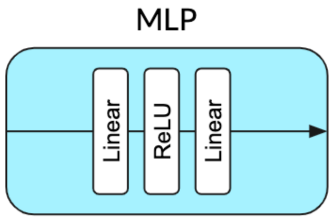
- SimCLR (Cheng et al, 2020) & SimCLR-v2 (2020)
- MoCo (He et al, 2019) & MoCo-v2 (2020)

SimCLR & MoCo (MoCo-v2)









SimCLR and **MoCo** have four major components:

- *Data augmentation*
- *Base Encoder* f *(ResNet)*
- *Projection head* g
- *Contrastive Loss*

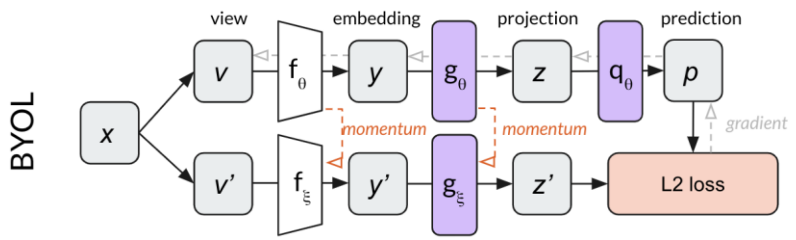
However, **MoCo** is more *computing-efficient* due to:

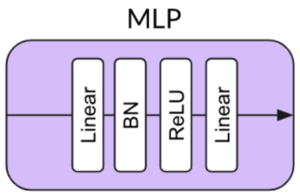
- *Slow-moving average network (momentum encoder)*
- *Dynamic dictionary look-up (memory bank)*

These methods need careful treatments of negative pairs

Image source:
https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning/

BYOL – Bootstrap Your Own Latent



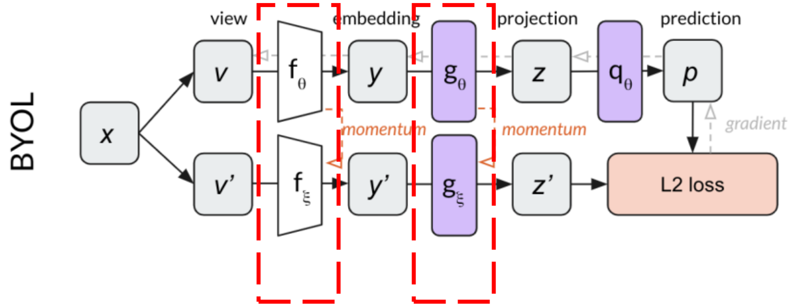


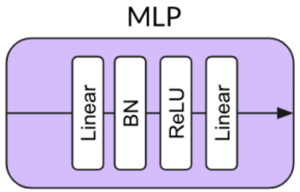
Two neural networks, referred to as *online* and *target networks*, that interact and learn from each other

- No negative pairs required

Image source:
https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning/

BYOL – Bootstrap Your Own Latent



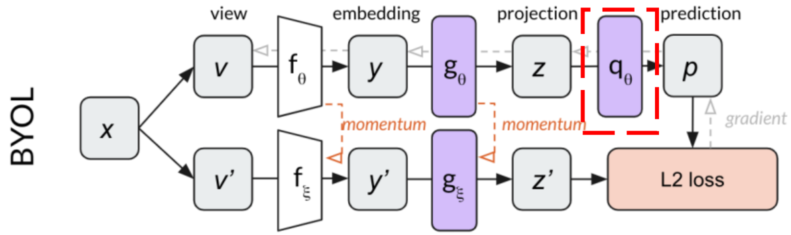


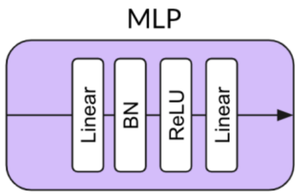
Two neural networks, referred to as _online_ and _target networks_, that interact and learn from each other

- No negative pairs required
- Momentum encoder concept based on **MoCo**

Image source:
https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning/

BYOL – Bootstrap Your Own Latent



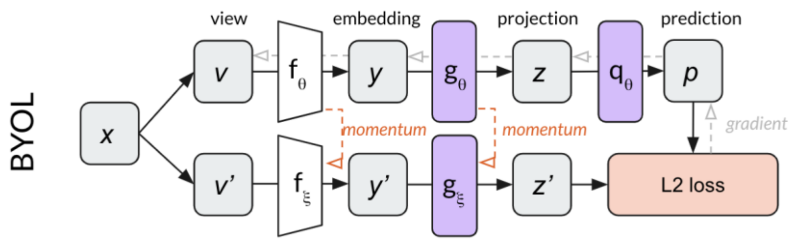


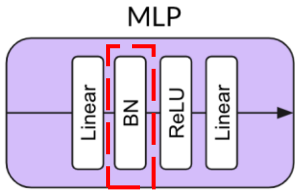
Two neural networks, referred to as *online* and *target networks*, that interact and learn from each other

- No negative pairs required
- Momentum encoder concept based on **MoCo**
- Online network has a *predictor*

Image source:
https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning/

BYOL – Bootstrap Your Own Latent



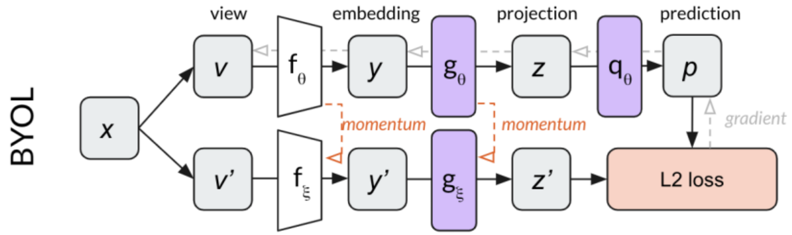


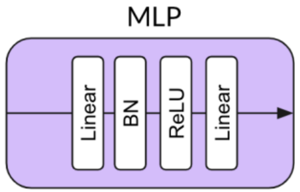
Two neural networks, referred to as *online* and *target networks*, that interact and learn from each other

- No negative pairs required
- Momentum encoder concept based on **MoCo**
- Online network has a *predictor*
- Batch normalization helps avoid *dimensional collapse* (predicting the same code for every image)

Image source:
https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning/

BYOL – Bootstrap Your Own Latent





Two neural networks, referred to as *online* and *target networks*, that interact and learn from each other

- No negative pairs required
- Momentum encoder concept based on **MoCo**
- Online network has a *predictor*
- Batch normalization helps avoid *dimensional collapse* (predicting the same code for every image)

→ Inspiration for DINO!!!

Image source:
https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning/

BYOL – Bootstrap Your Own Latent

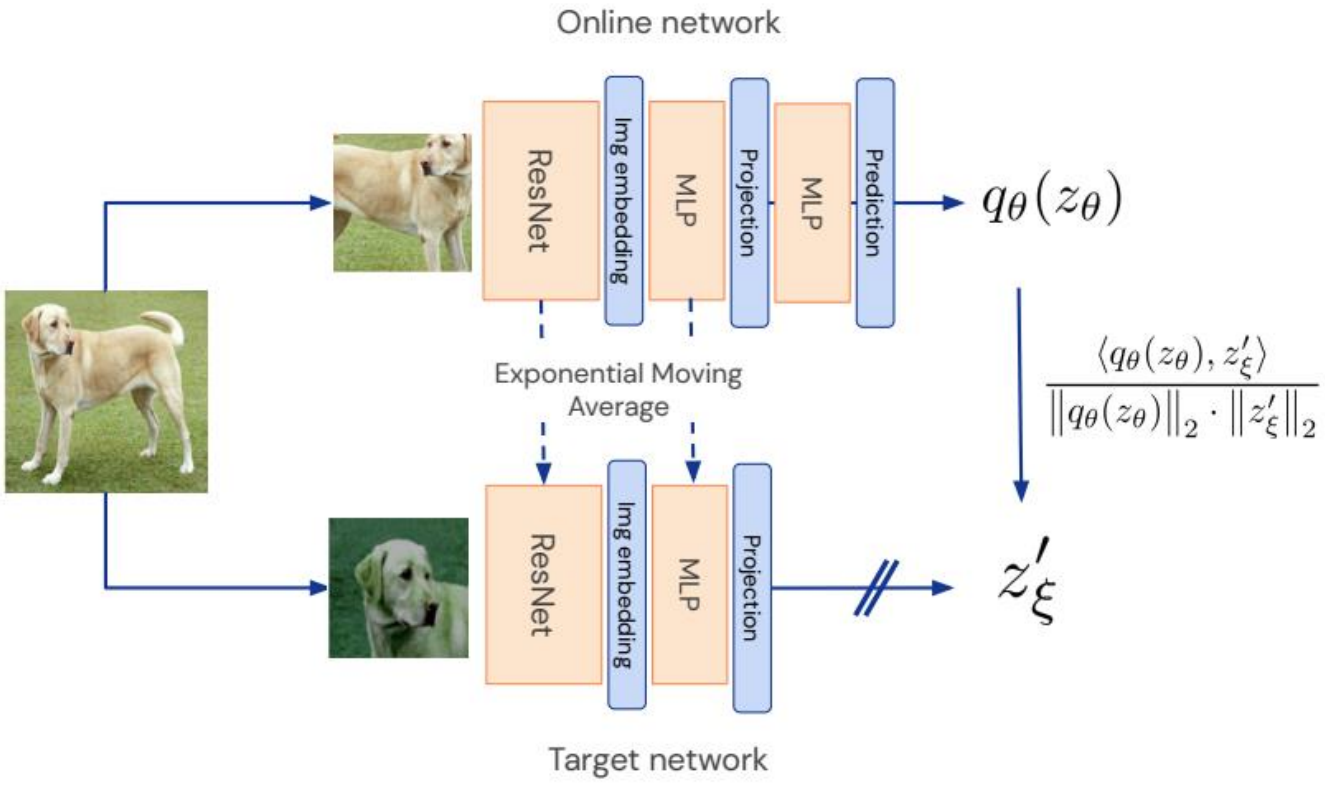


Image source:
[BYOL paper, Jean-Bastien Grill et al. 2020]

ViT Architecture overview

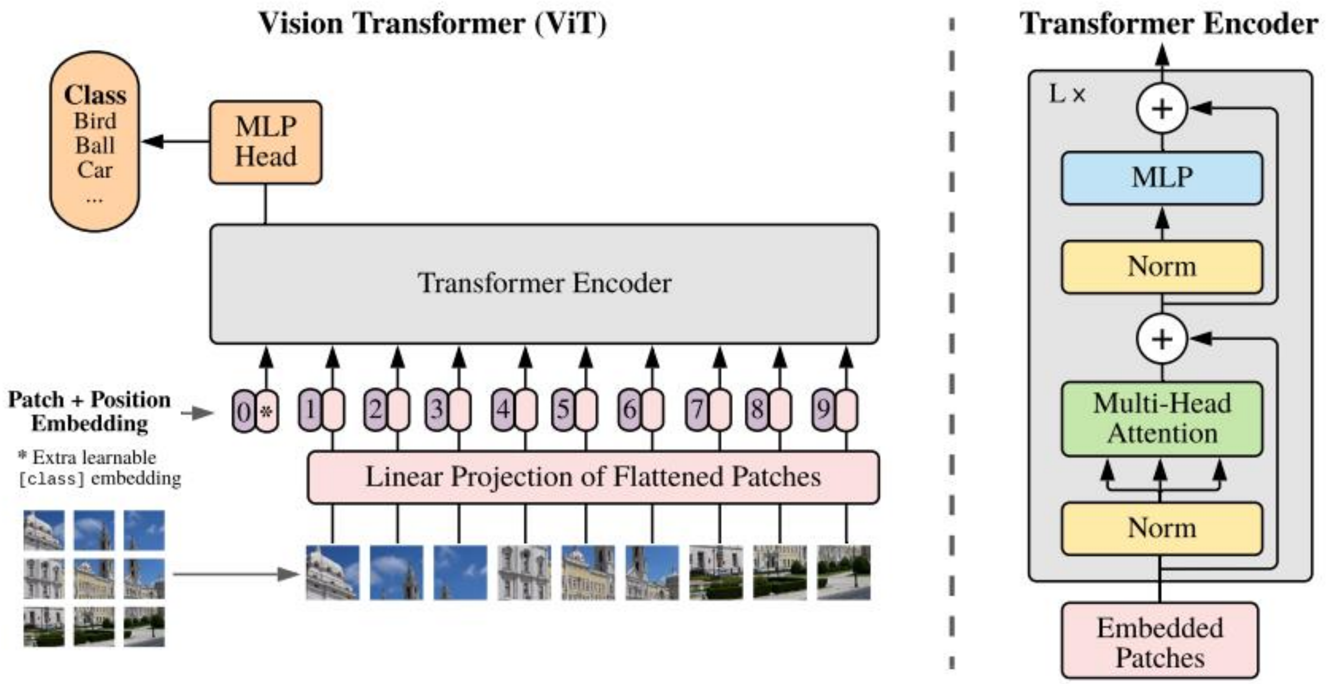
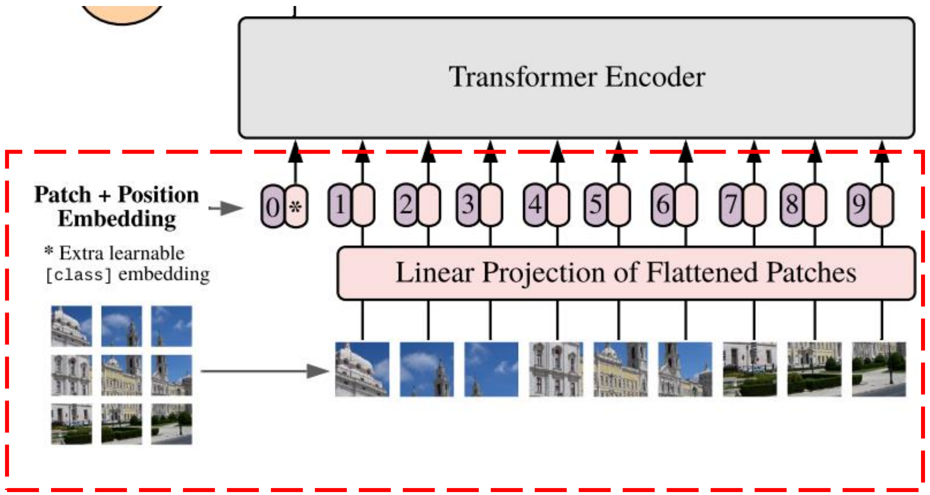


Image source:
[An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale]

ViT Architecture overview – Patch + Position Embedding



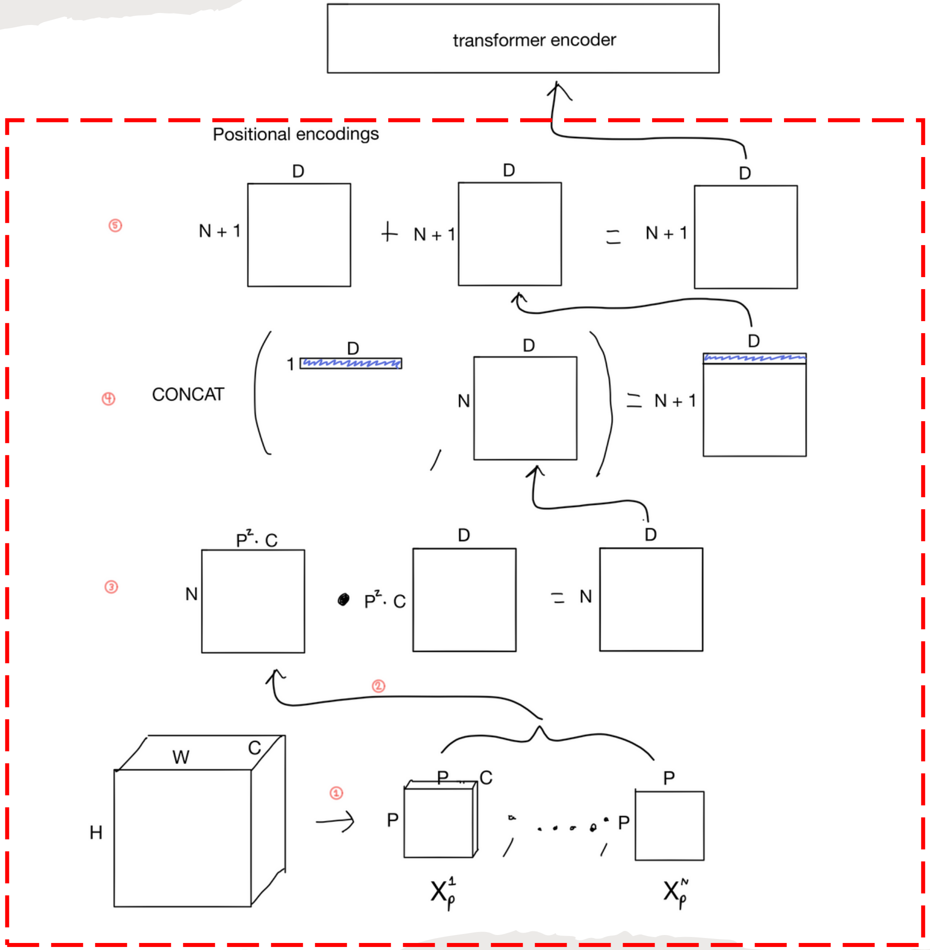
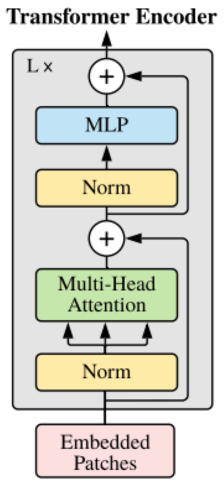


Image source:
[An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale]()

ViT Architecture overview – Transformer Encoder



$P = 16$
$N = 14 * 14$
$D = P^2C = 16 * 16 * 3$

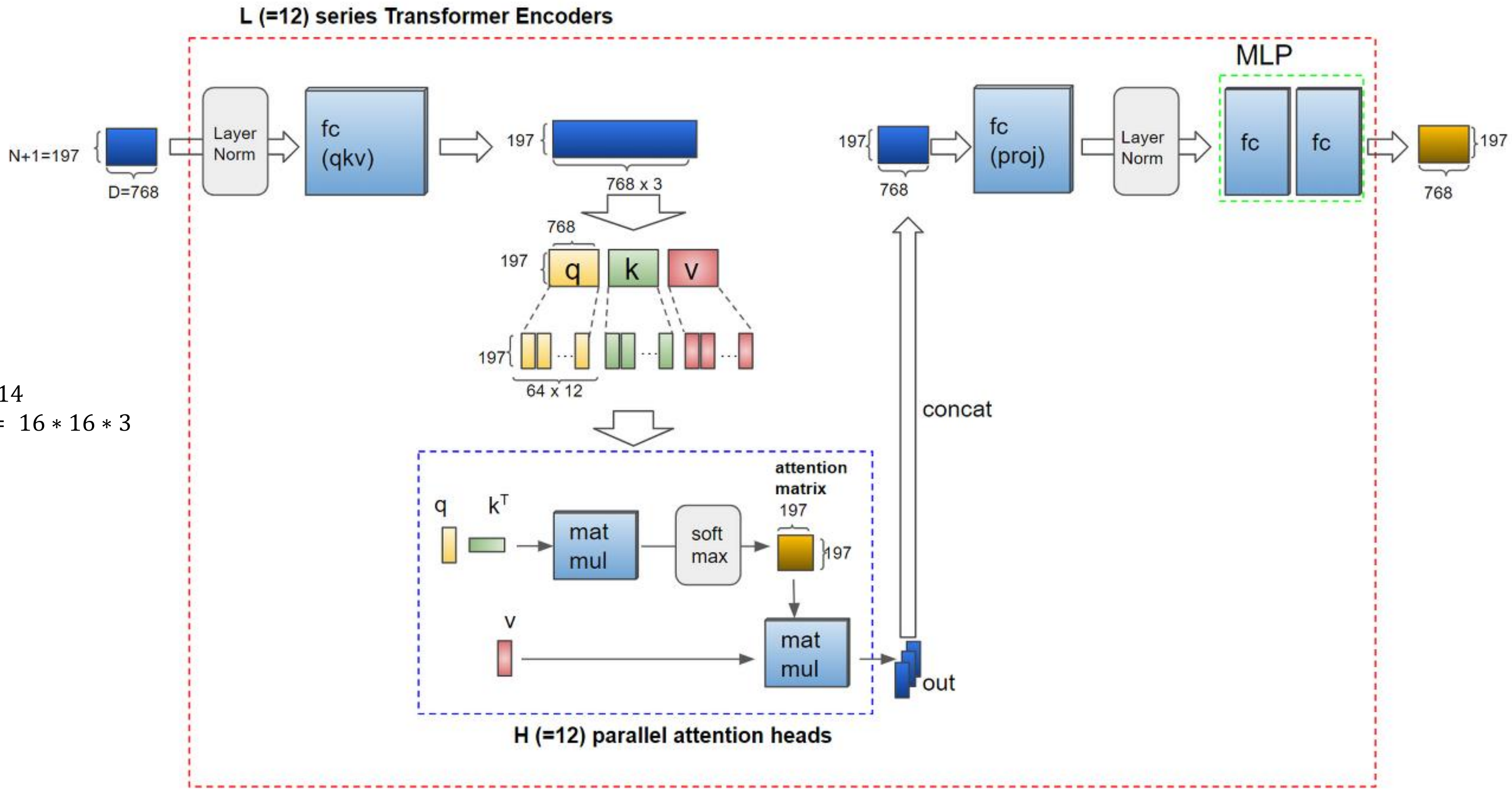


Image source:
[An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale]
[https://hafoom.com/39]

DINO – Self-distillation with No Labels

DINO (Self-Distillation with No Labels)

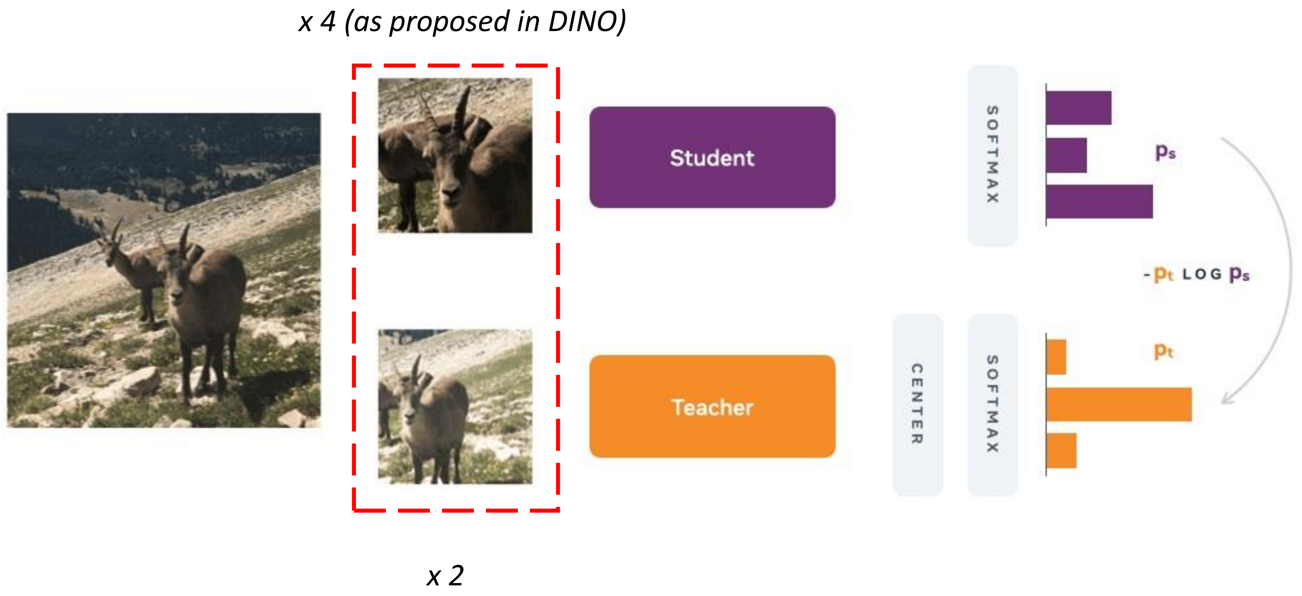
DINO interprets the existing **BYOL** framework with a few changes:

- **Core architecture** is Vision Transformer (can be flexible)

- **Self-distillation**:
  - Two separate networks - student and teacher (same as target and online network in BYOL)
  - However, the final loss is *cross-entropy* (as proposed in *Knowledge Distillation paper*)

- **Collapse prevention** by centering and sharpening of the teacher output

How DINO works

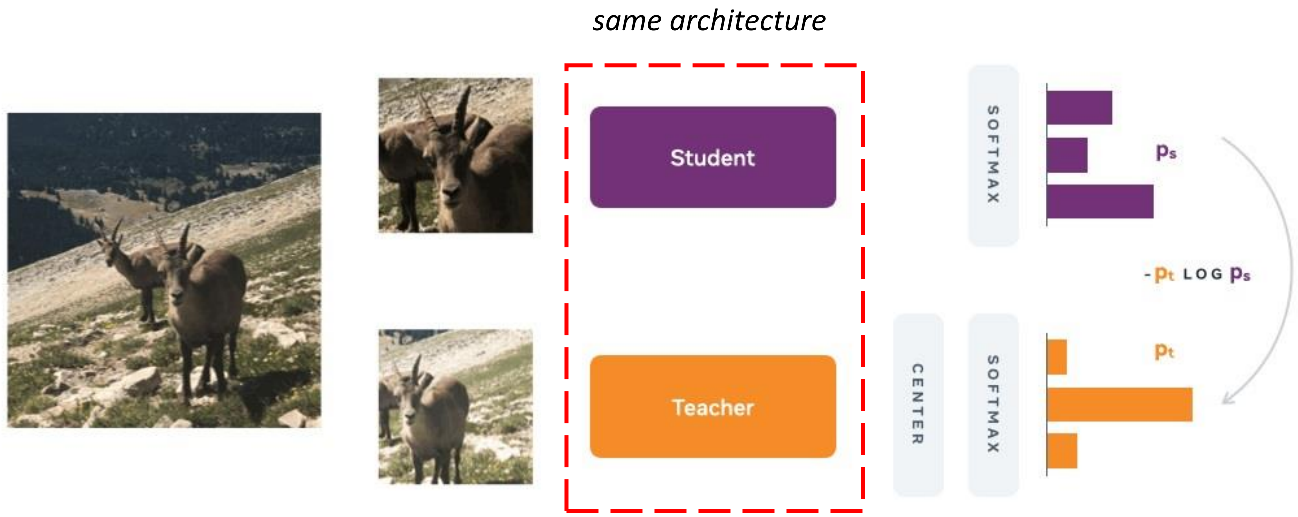


1. Forward Training Phase

Different crops of an image are created

- Two ***global* views**, $\boldsymbol{x_1^g}$ and $\boldsymbol{x_2^g}$
- Several ***local* views** of smaller resolution

How DINO works



1. Forward Training Phase

- **All crops** are passed through **the student**
- **Only** the global views are passed through **the teacher**

How DINO works

Intuition behind *centering and sharpening* of **DINO**:

Unlike standard convnets (as in **BYOL**), ViT architectures do not use *batch normalizations* (BN) by default

Thus, to avoid collapse, **DINO** uses two separate operations:

- <u>Centering</u> by subtracting the mean feature → prevents collapse to constant 1-hot target
- <u>Sharpening</u> by using low softmax temperature → prevents collapse to a uniform target vector

How DINO works

1. Forward Training Phase

The output from teacher network g_{θ_t} is then go through **centering stage**:

Centering can be defined as adding the bias term c to the teacher

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

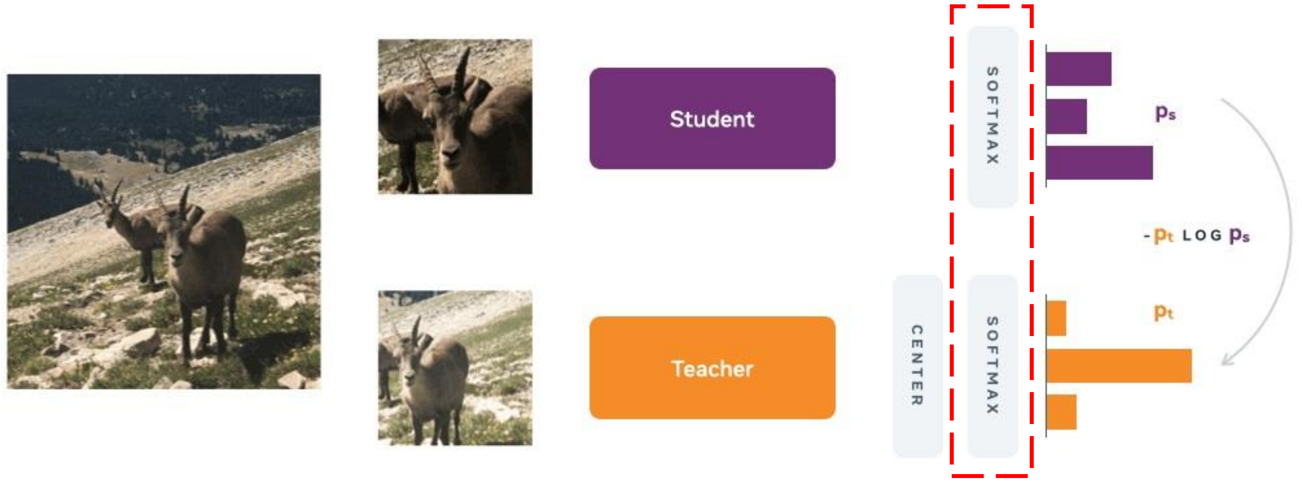
parameter controlling how much to update the center

old center

teacher output

The update rule for parameter c is : $c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i)$

How DINO works



1. Forward Training Phase

Both g_{θ_t} and g_{θ_s} go through **sharpening stage (softmax)**

- g_{θ_s} is normalized to output the following distribution

temperature

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

- g_{θ_t} : similar formula holds

Note: τ_s usually has higher temperature (less sharpness) than τ_t

How DINO works

1. Forward Training Phase

For fixed teacher network g_{θ_t}, we learn to match these distributions by minimizing the cross-entropy loss w.r.t. the parameters of the student network θ_s :

$$\min_{\theta_s} H(P_t(x), P_s(x))$$

Where $H\big(P_t(x)\,, P_s(x)\big) = -\,P_t(x)\log P_s(x)$

How DINO works

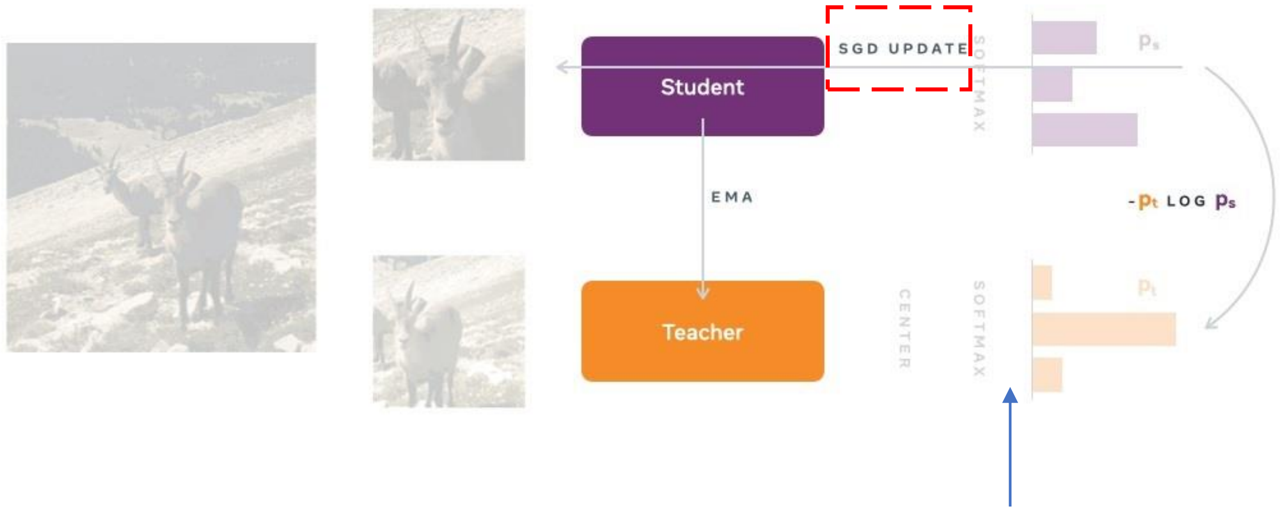
1. Forward Training Phase

Combine with the **multi-crop technique (MC)** mentioned earlier, we get the final loss to minimize:

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

θ_s can be learned by minimizing the above equation with *stochastic gradient descent*

How DINO works

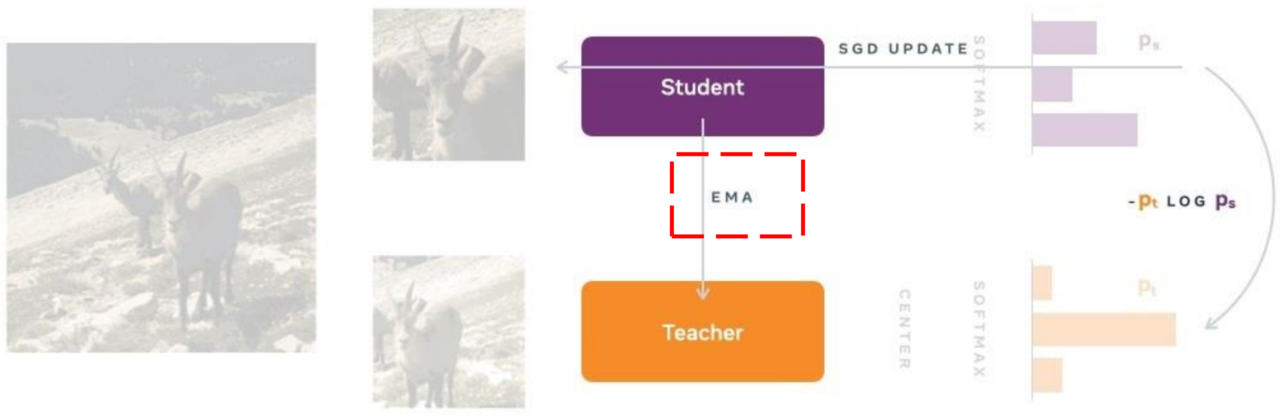


Stop gradient prevents updating the teacher network

2. Backpropagation Phase

The backpropagation is performed **only** through the student network to update the student parameter θ_s

How DINO works



2. Backpropagation Phase

The learned θ_s is used to update θ_t via exponential moving average (similar to **MoCo** & **BYOL**):

$$\theta_t \longleftarrow \lambda\theta_t + (1 - \lambda)\theta_s$$

λ : cosine schedule from 0.996 to 1

Key takeaways

- Teacher and Student network have the same architecture but different parameters
- Teacher is not trained beforehand but is trained along with the student network
- Multi-cropping is used to create global and local views of the image
  - Encourage *“local-to-global” correspondences*
- Student parameters are updated via backpropagation of the loss function
- Teacher parameters are updated via EMA using earlier student parameters
- Sharpening and Centering in **DINO** has the effect of preventing mode collapse, analogous to batch normalization in **BYOL**

Main Results

Comparing with other SSL frameworks

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
BYOL [30]	RN50	23	1237	74.4	64.8
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**

- **DINO** performance is second best (only behind supervised learning
- Can be combined with ResNet50 and manage to achieve remarkable results
- When combining **DINO** with ViT, k-NN classifier is almost on par with linear classifier

Probing the Self-Attention Map

Attention maps from multiple heads

DINO can attend to different semantic regions of an image, even when they are *small* or *occluded*

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

DINO can attend well to objects in *presence of clutter*

Probing the Self-Attention Map

Source:
https://ai.facebook.com/blog/dino-paws-computer-vision-with-self-supervised-transformers-and-10x-more-efficient-training/

Probing the Self-Attention Map

Image source:
https://ai.facebook.com/blog/dino-paws-computer-vision-with-self-supervised-transformers-and-10x-more-efficient-training/

Why is DINO so powerful?

It turns out, applying self-supervision to Vision Transformers leads to the following desirable properties:

- The model learns to <u>semantically segment</u> the object and create boundaries. This information is accessible in the self-attention modules
- The learned feature representations i.e., the output vector of the model, is very useful to perform clustering
    - ➢ Allow fast classification (i.e: k-NN)

Clustering ability of DINO

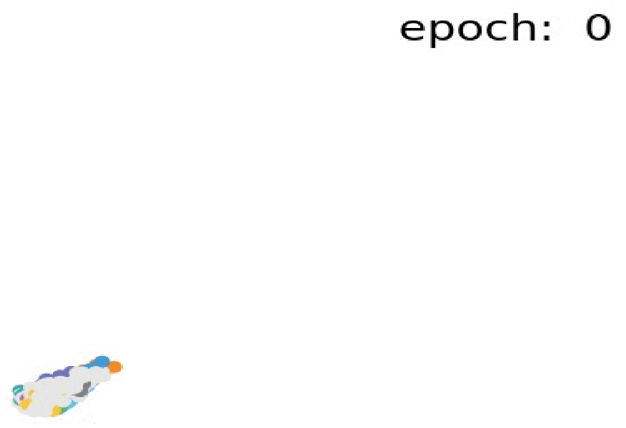
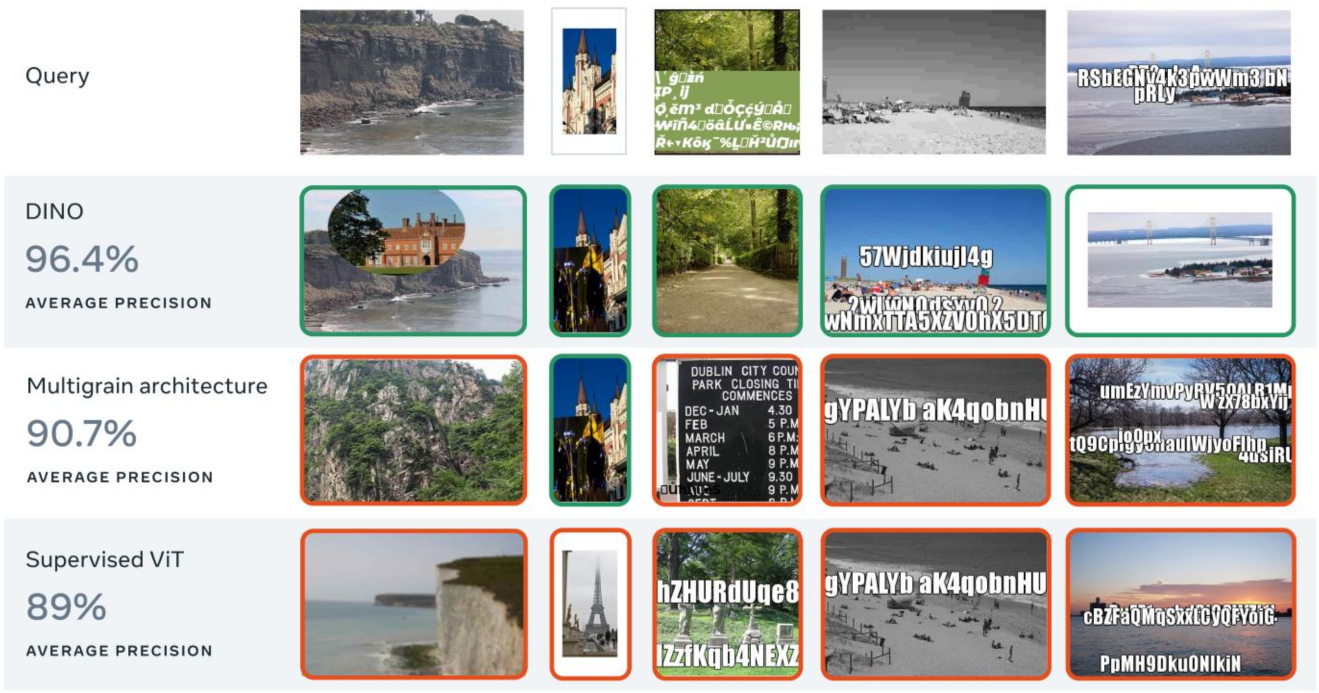


Image source:
https://ai.facebook.com/blog/dino-paws-computer-vision-with-self-supervised-transformers-and-10x-more-efficient-training/

Copy Detection



DINO outperforms other models in determining whether an image is a modified copy of any image in a database, without being designed to perform such task

Image source:
https://ai.facebook.com/blog/dino-paws-computer-vision-with-self-supervised-transformers-and-10x-more-efficient-training/

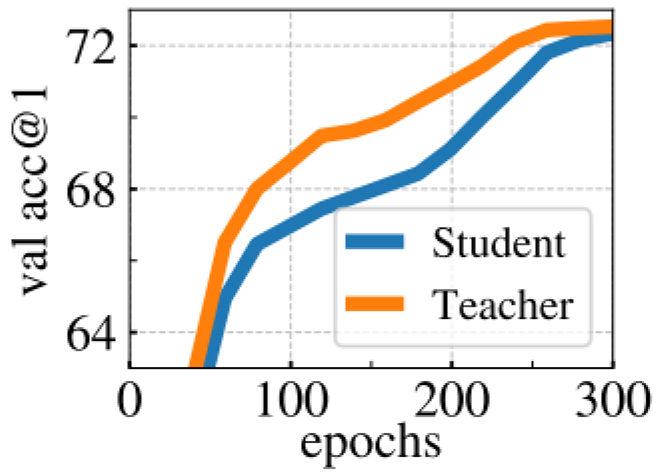
Ablation Study of DINO

Importance of the different components

Method	Mom.	MC	Loss	Pred.	k-NN	Lin.
DINO	✓	✓	CE	✗	72.8	76.1
	✗	✓	CE	✗	0.1	0.1
	✓	✗	CE	✗	67.9	72.5
	✓	✓	MSE	✗	52.6	62.4
	✓	✓	CE	✓	71.8	75.6
BYOL	✓	✗	MSE	✓	66.6	71.4
MoCov2	✓	✗	INCE	✗	62.0	71.6

- Without *Momentum*, **DINO** failed to converge.
- *Predictor* is required for **BYOL** to avoid collapse, but not for **DINO**
- *Multi-crop* and *cross-entropy loss* are important components to obtain good features in **DINO**

Impact of the choice of Teacher Network



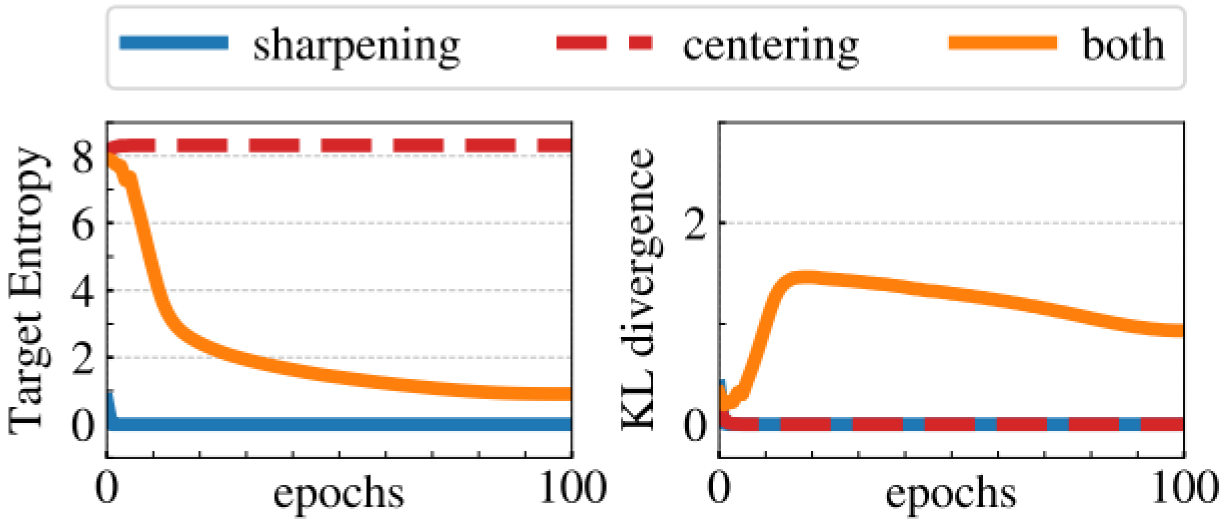
Teacher	Top-1
Student copy	0.1
Previous iter	0.1
Previous epoch	66.6
Momentum	72.8

☹ **Student copy** and **Previous iter** fail to converge.

😎 **Momentum encoder** yields the best result, followed by **Previous epoch**.

- Teacher always outperforms **student** when using momentum encoder => Help guide the student

Impact of the choice of Teacher Network



Two forms of collapse:

- Model output is uniform along all dimensions (high entropy)
- Model output is dominated by one dimension (low entropy)

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

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

Summary

Self-supervised learning such as DINO

- Can learn SOTA self-supervised representations without requiring negatives
  - ➢ Surpassing even supervised learning for segmentation
  - ➢ Nearly surpassing supervised learning for classification

- Have emerged properties that can be leveraged in future applications
  1. Features quality has potential for k-NN classification and image retrieval
  2. Scene layout information can also benefit weakly supervised image segmentation

- Manage to achieve performance comparable with the best convnets with the same setting
  - ➢ Could be the key to developing a BERT-like model based on ViT

References

- **Knowledge Distillation:**
  - Geoffrey Hinton, Oriol Vinyals, Jeff Dean : Distilling the Knowledge in a Neural Network , *arXiv:1503.02531* , 2015

- **SimCLR:**
  - Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton : A Simple Framework for Contrastive Learning of Visual Representations , *arXiv:2002.05709* , 2020.

  - Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, Geoffrey Hinton : Big Self-Supervised Models are Strong Semi-Supervised Learners , *arXiv:2006.10029* , 2020.

- **MoCo:**
  - Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick : Momentum Contrast for Unsupervised Visual Representation Learning , *arXiv:1911.05722* , 2019.

  - Xinlei Chen, Haoqi Fan, Ross Girshick, Kaiming He : Improved Baselines with Momentum Contrastive Learning , *arXiv:2003.04297* , 2020.

- **BYOL:**
  - Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond et al : Bootstrap your own latent: A new approach to self-supervised Learning , *arXiv:2006.07733* , 2020

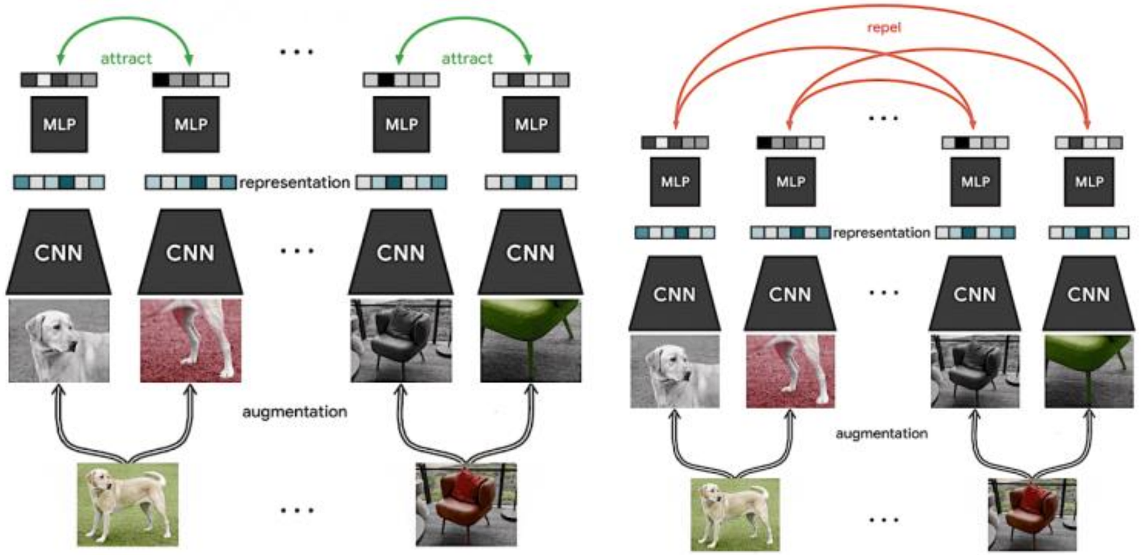
- **Misc:**
  - Antti Tarvainen, Harri Valpola : Mean teachers are better role models : Weight-averaged consistency targets improve semi-supervised deep learning results , *arXiv:1703.01780* , 2017

  - Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin : Unsupervised feature learning via non-parametric instance discrimination , *arXiv:1805.01978* , 2018

  - Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, Armand Joulin : Unsupervised Learning of Visual Features by Contrasting Cluster Assignments , *arXiv:2006.09882* , 2020

  - Three mysteries in deep learning: Ensemble, knowledge distillation, and self-distillation , https://www.microsoft.com/en-us/research/blog/three-mysteries-in-deep-learning-ensemble-knowledge-distillation-and-self-distillation/

SimCLR



Two transformations **v** and **v'** of **x** are fed through *the same network* to produce two projections **z** and **z’**

The **contrastive loss** will:

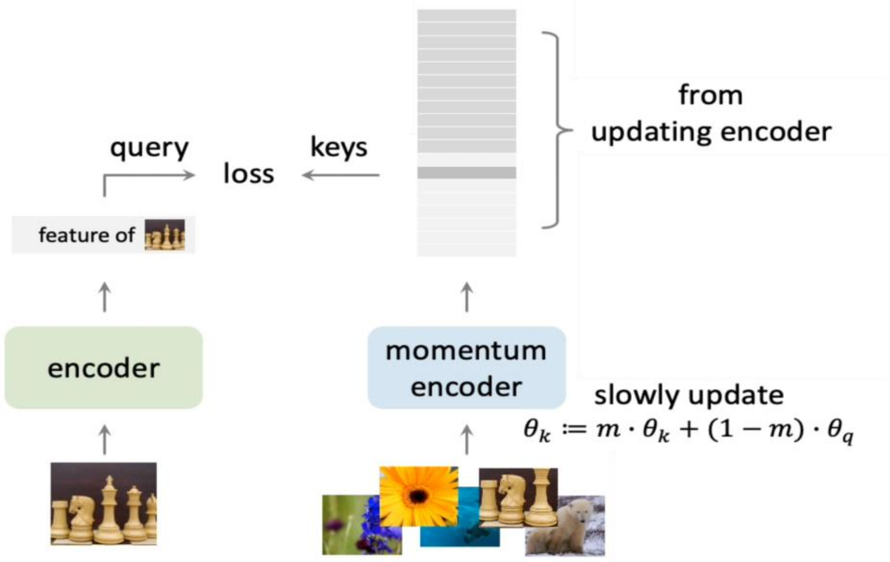
- Maximize the similarity of **z** and **z’** from the same input x
- Minimize the similarity to projections of other images within *the same mini-batch*

SimCLR relies on large batch training, in order to ensure a sufficiently diverse set of negatives

➡ ***computationally demanding*** ✕

Image source:
[Google AI Blog: Advancing Self-Supervised and Semi-Supervised Learning with SimCLR (googleblog.com)]

MoCo & MoCo-v2 - Momentum Contrast



- **Dynamic dictionary look-up** keeps a queue of encoded feature representations from the *current* and *previous* batches.
  ➡ Solve the batch size problem ✓
- **Slow-moving average network (*momentum encoder*)** is adopted to improve the representation consistency between the *current* and *earlier* keys.

Better results compared to **SimCLR** ✓

However, similar to **SimCLR**, **MoCo** also requires careful treatment of negative pairs ✕

Image source:
https://www.youtube.com/watch?v=4VVGtYPM8JE

Knowledge Distillation

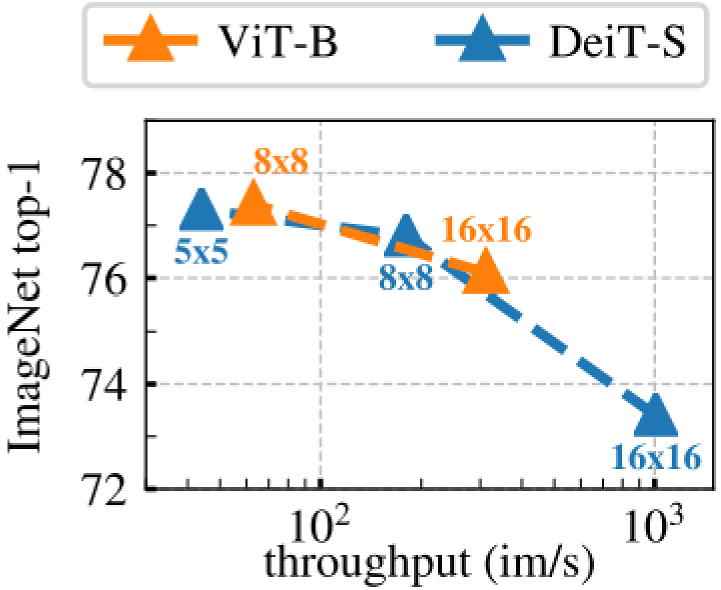
- A primary mechanism that enables humans to quickly learn new complex concepts when given only small training sets with the same or different categories
- In deep learning, KD is an effective technique that has been widely used to transfer information from one network to another network whilst training constructively
- KD was first defined and generalized by *Hinton et al*
- KD has been broadly applied to two distinct fields: model compression and knowledge transfer

Self-Distillation

- **Definition:** The goal of self-distillation is to learn a student model by distilling knowledge in itself without referring to other models
- **Another definition:** When both the student and teacher are the same network, then it is called as *Self-Distillation*

Important of the different components

- **Importance of the patch size.**



The performance improves as the size of the patch is decreased.

=> **Tradeoff:** between performance and throughput (images per second)