DeepMind

Bootstrap Your Own Latent:
A new approach to self-supervised learning

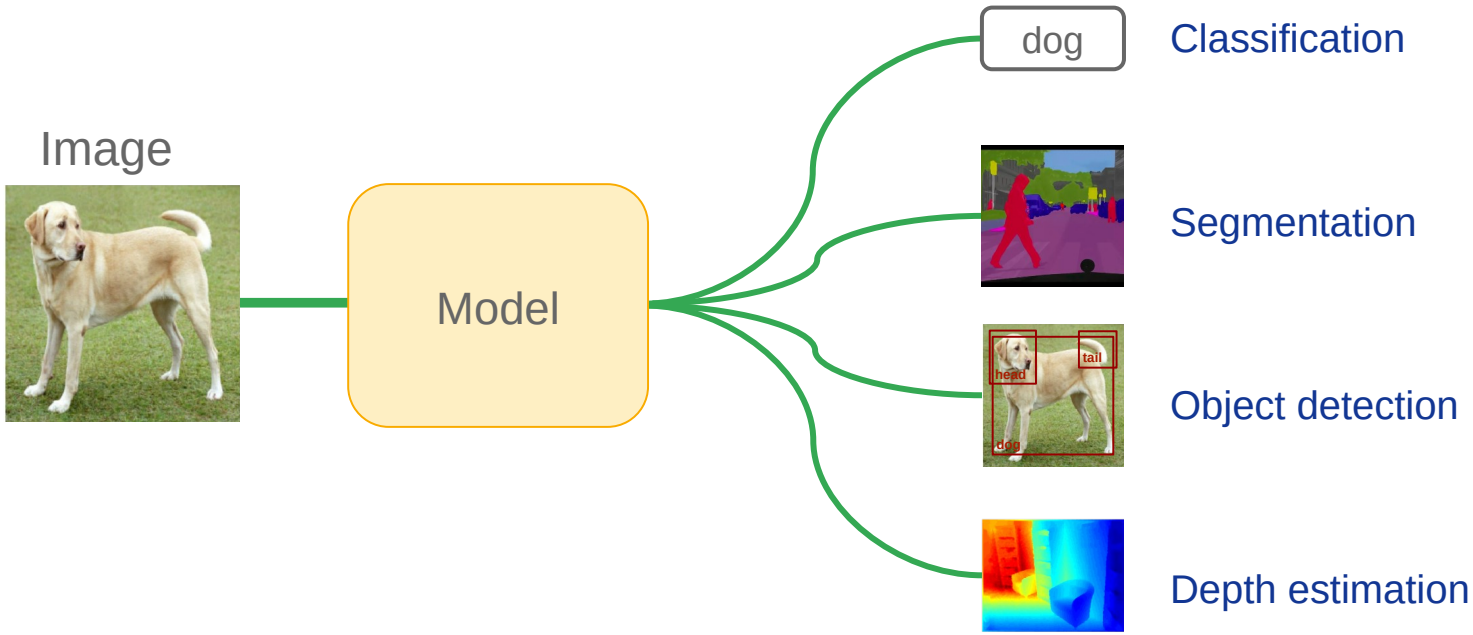
Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond
Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi
Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, **Michal Valko**

NeurIPS 2020

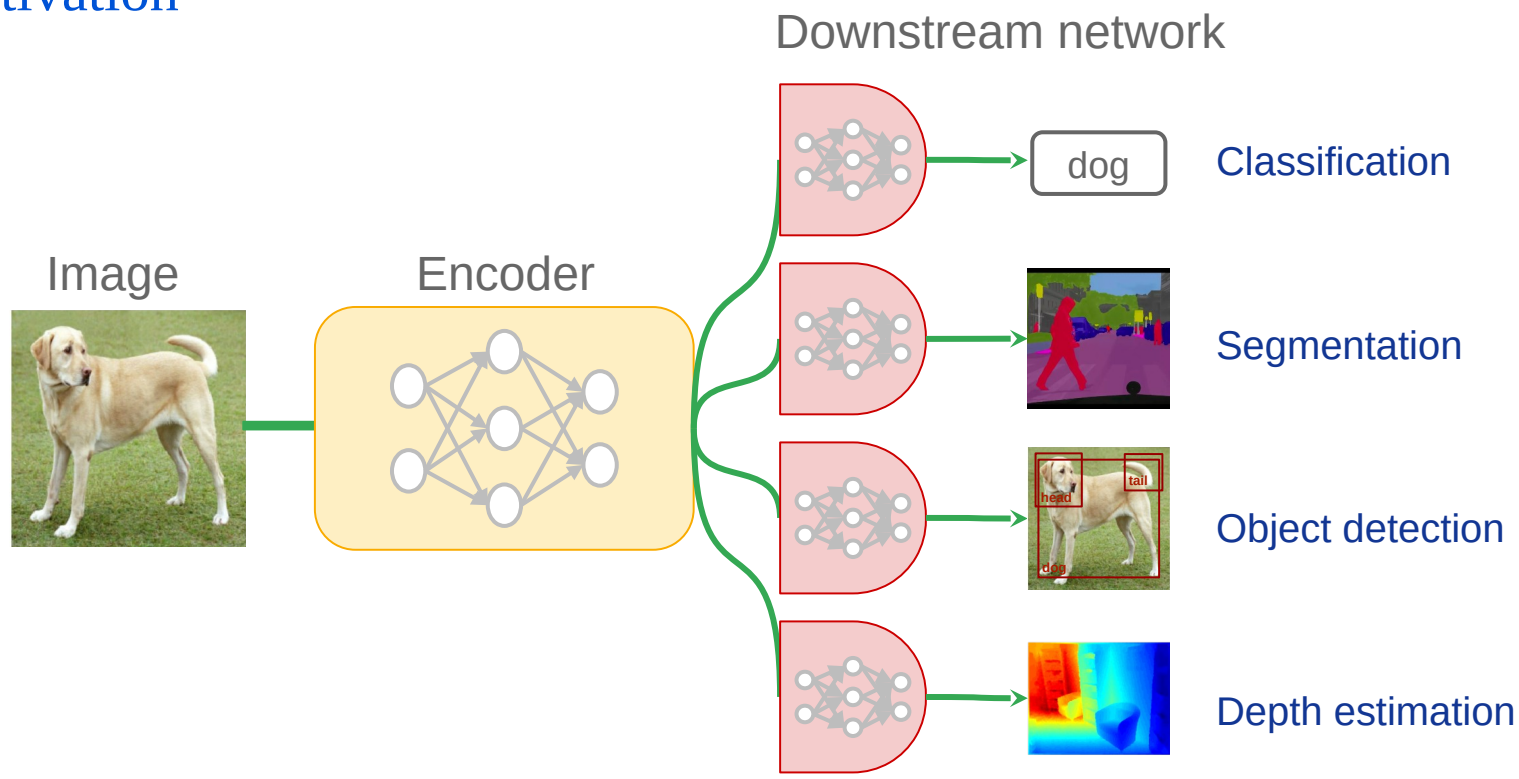
DeepMind

1 Self-Supervised Learning

Computer Vision Goal

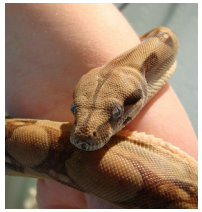


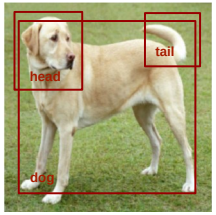
Motivation



How to train the encoder?

Motivation

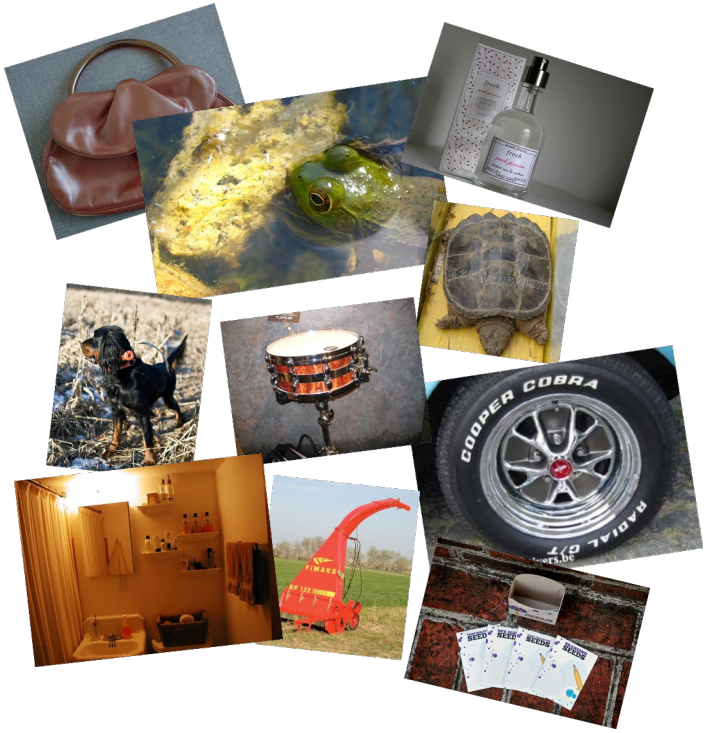




Dog

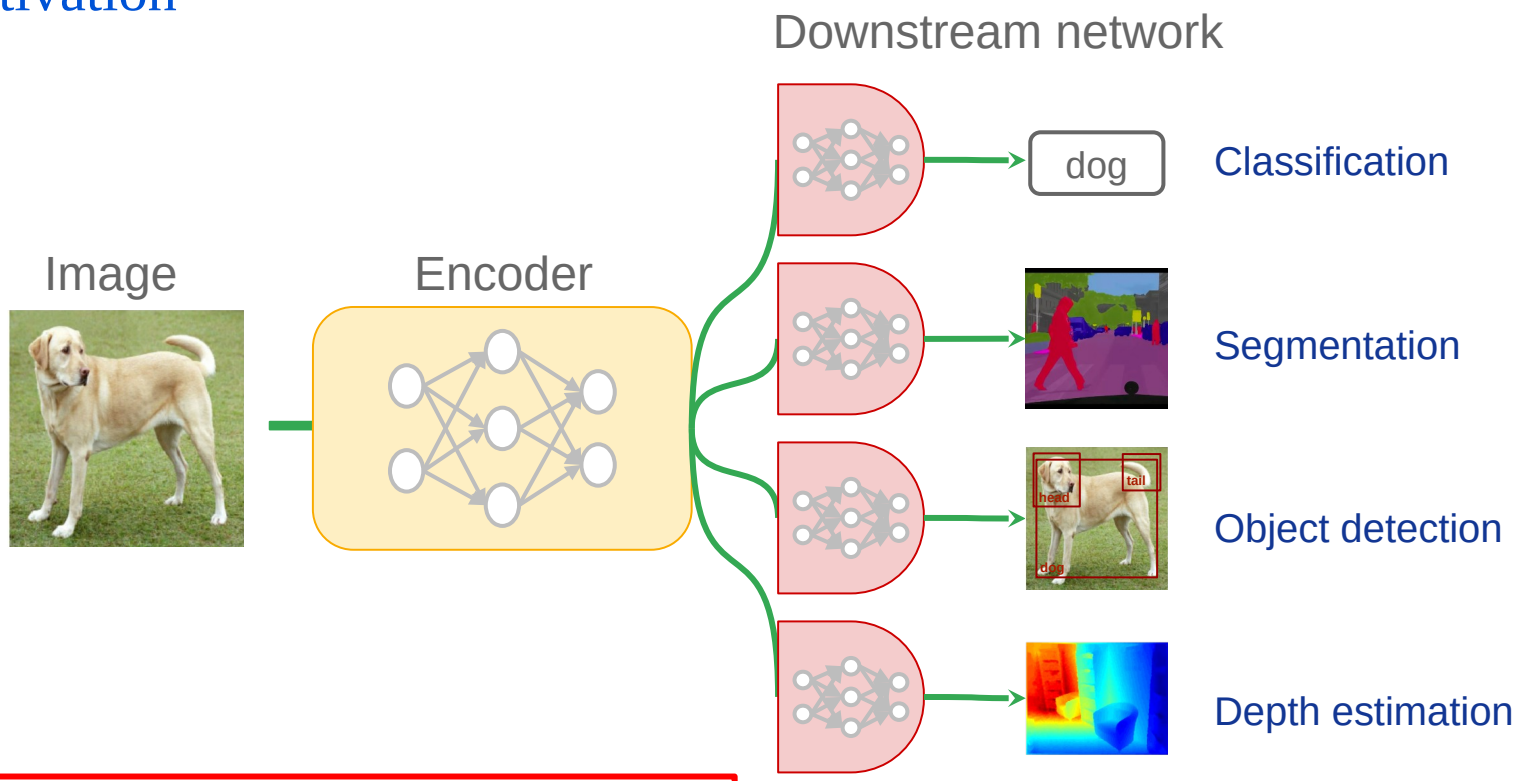
Snake

Labelled, but costly/few data



Unlabelled, **free** data!

Motivation


Downstream network
Image
Encoder
dog
Classification
Segmentation
head
tail
dog
Object detection
Depth estimation
BYO
Self-supervised
Free unlabeled data
Supervised
Few labeled data

DeepMind

2 Method

Intuition: Two different views (augmentations) of the same picture should be predictive of each other.

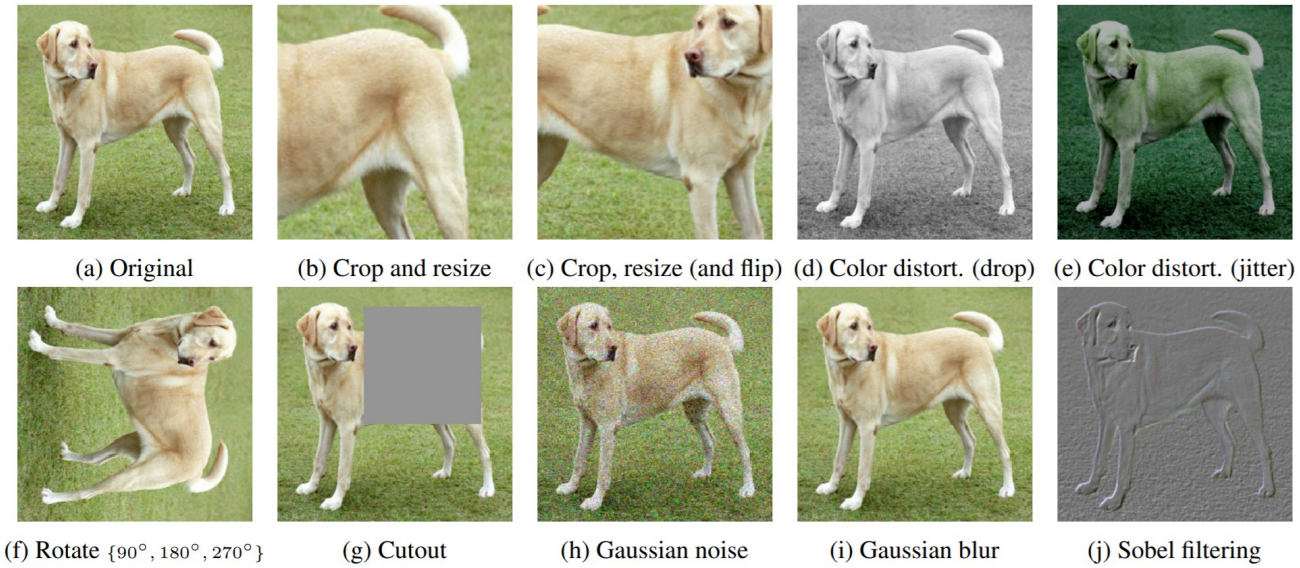
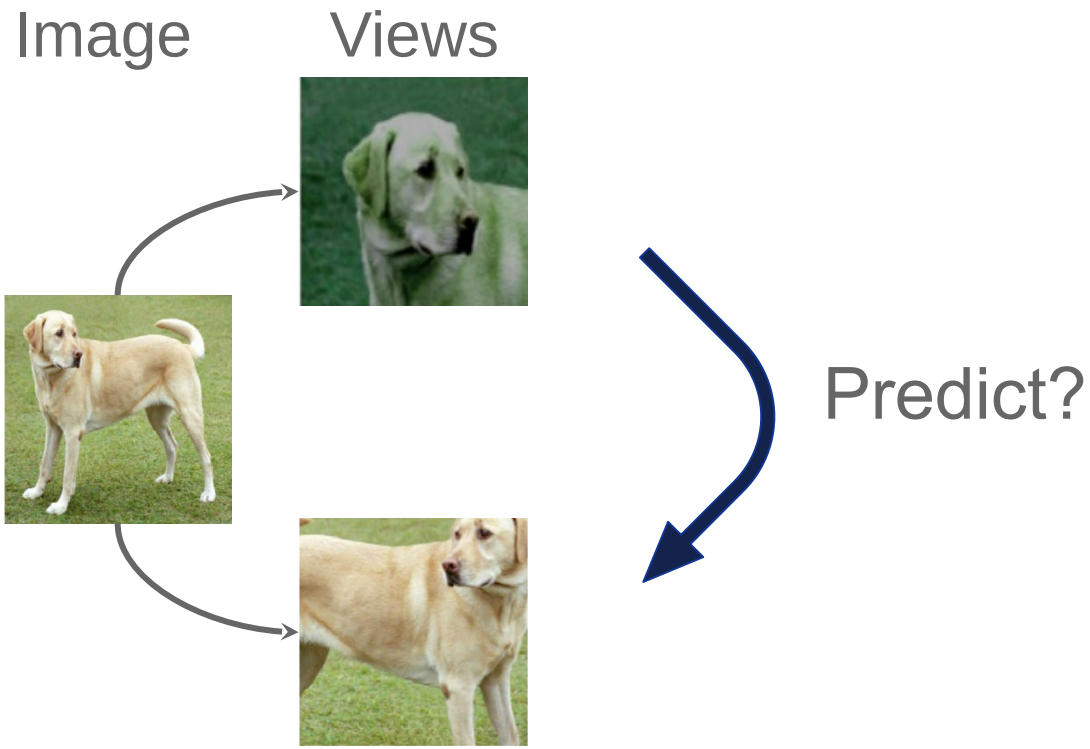


Figure from SimCLR[1]

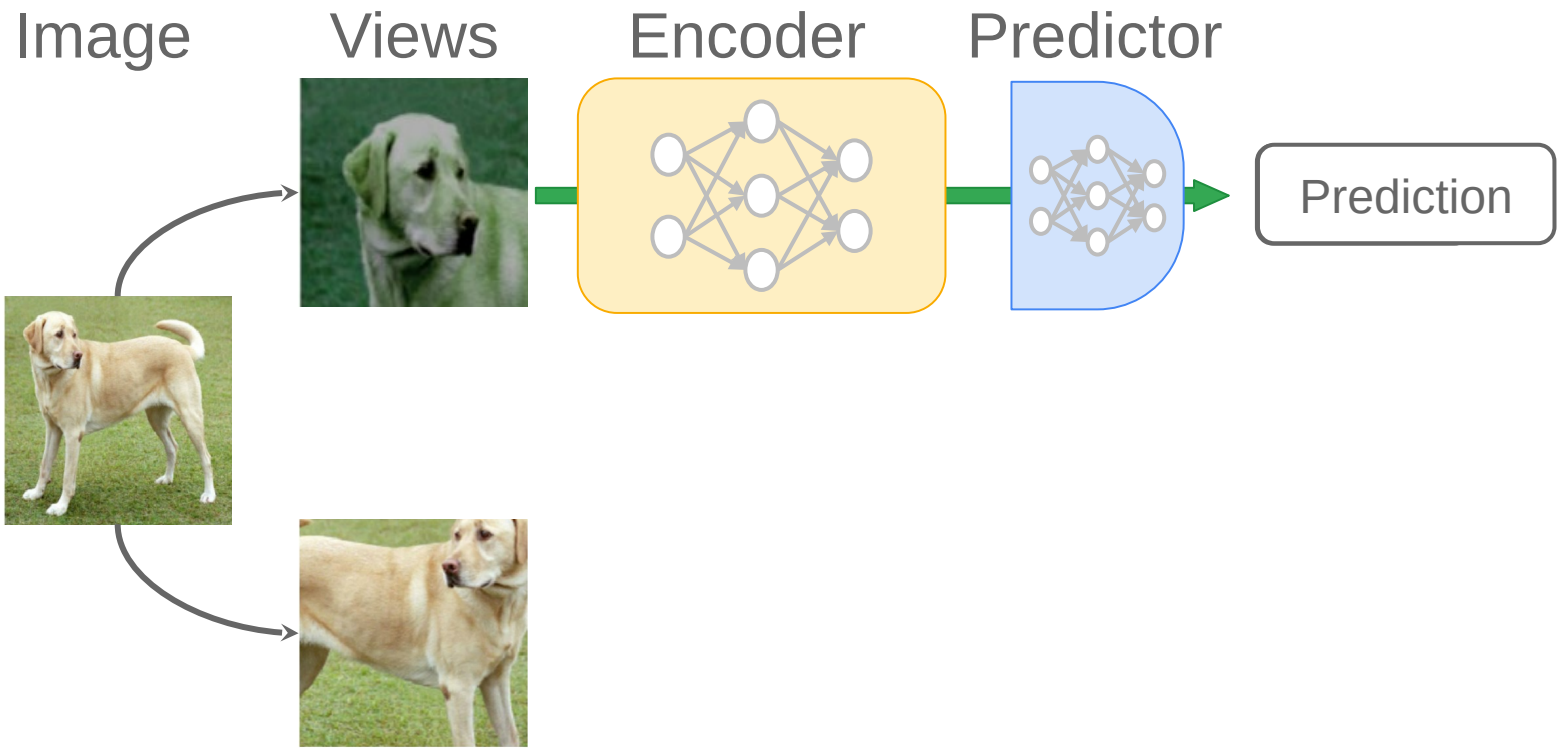
A view of a dog is still a dog, i.e. semantic information is **invariant** to transformations.

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

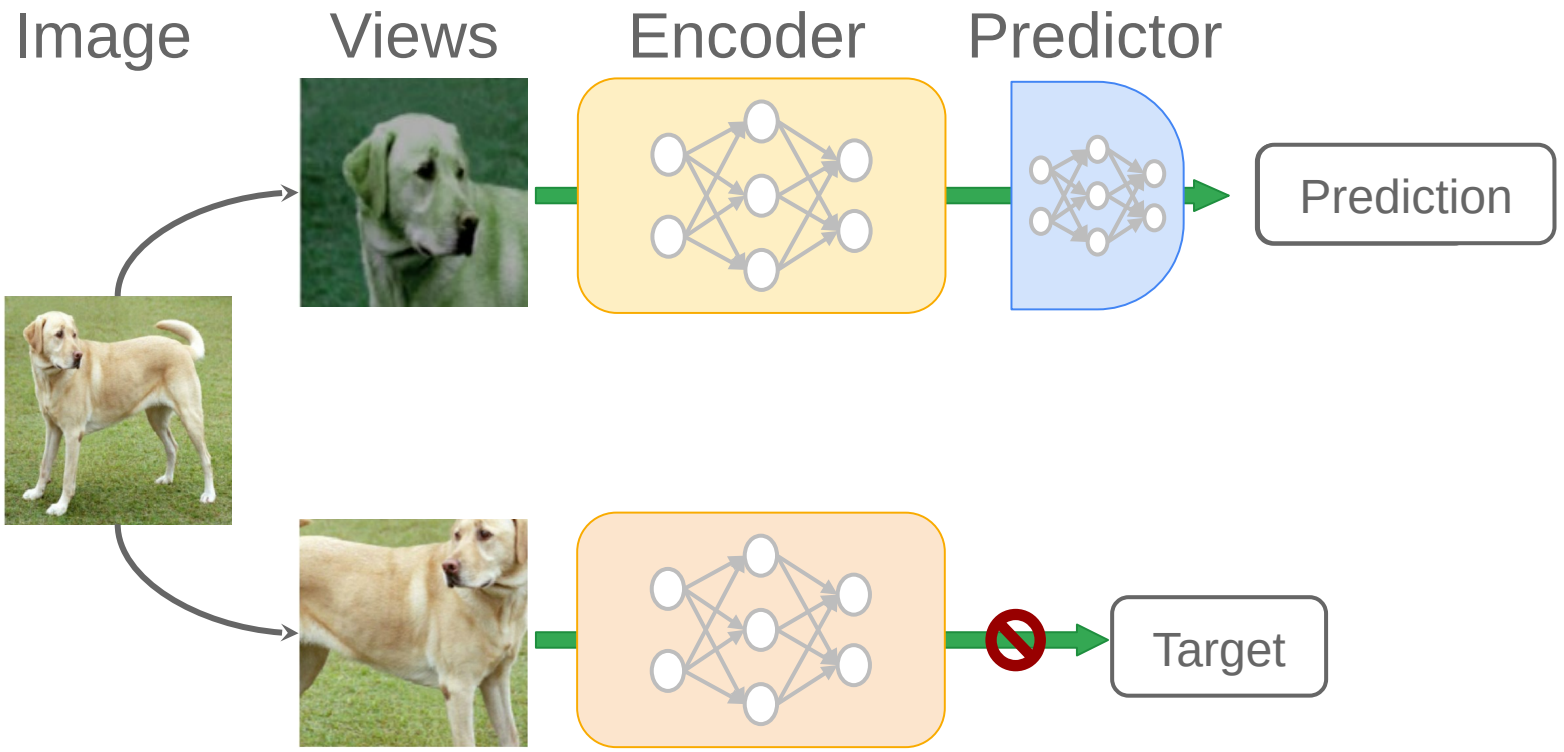
BYOL main intuition



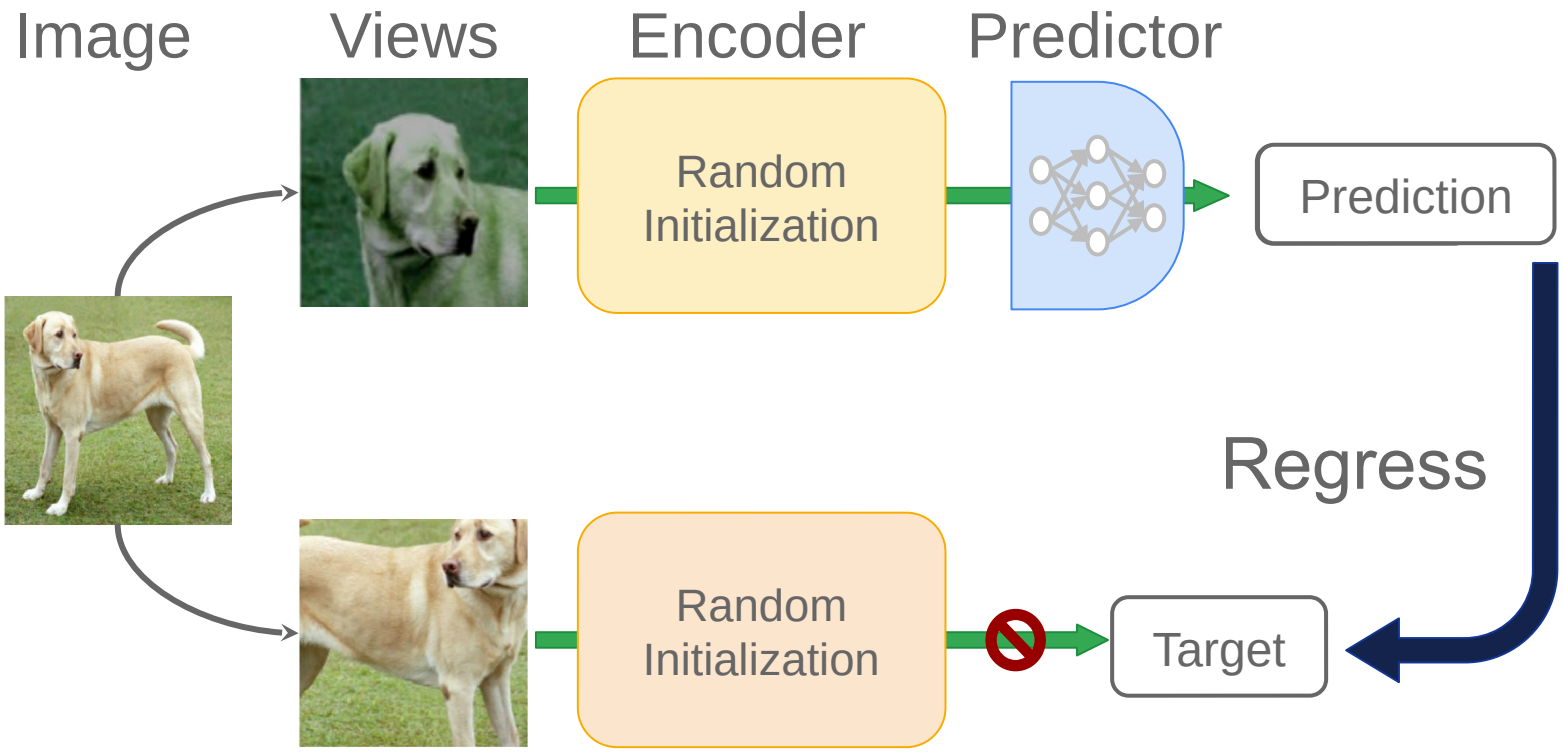
BYOL main intuition



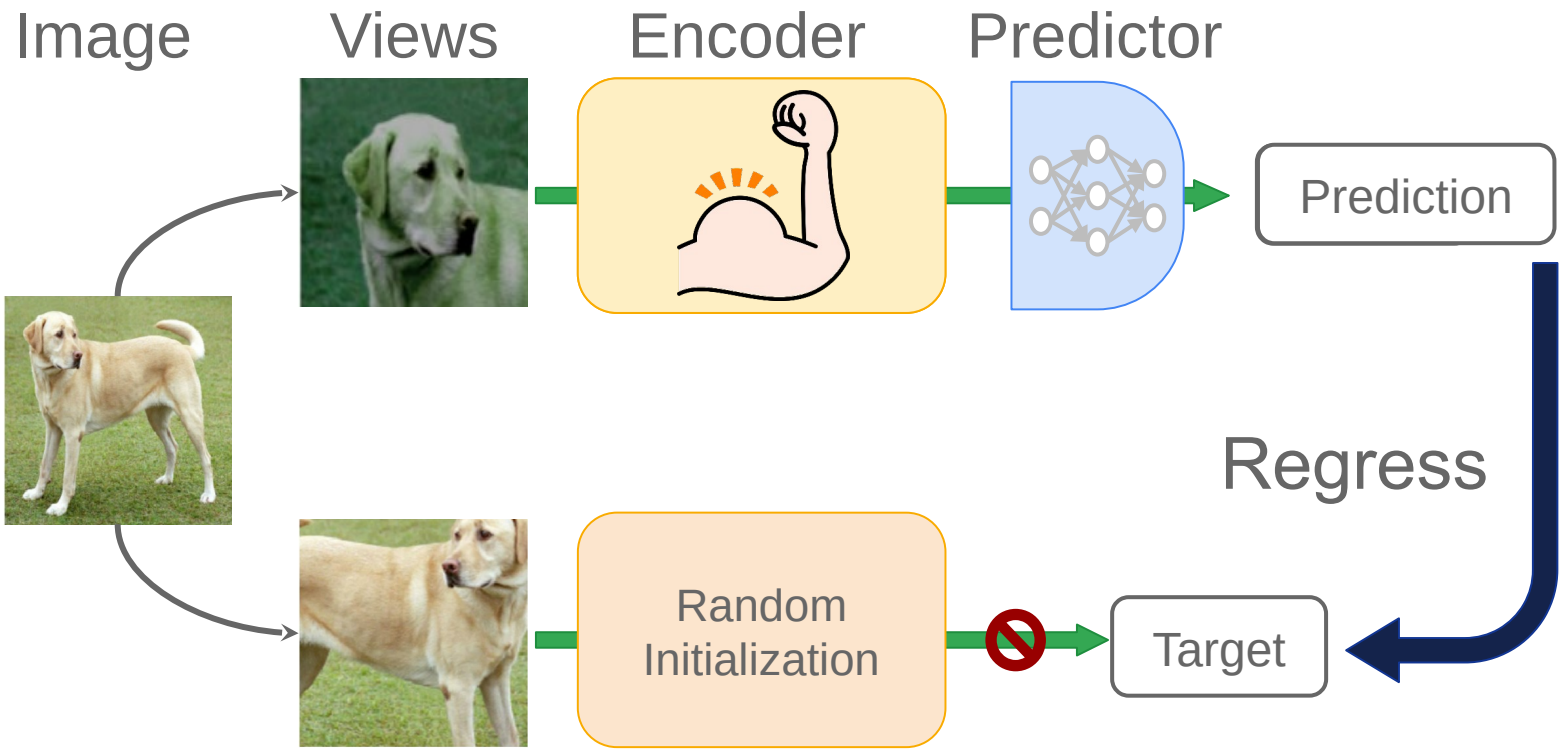
BYOL main intuition



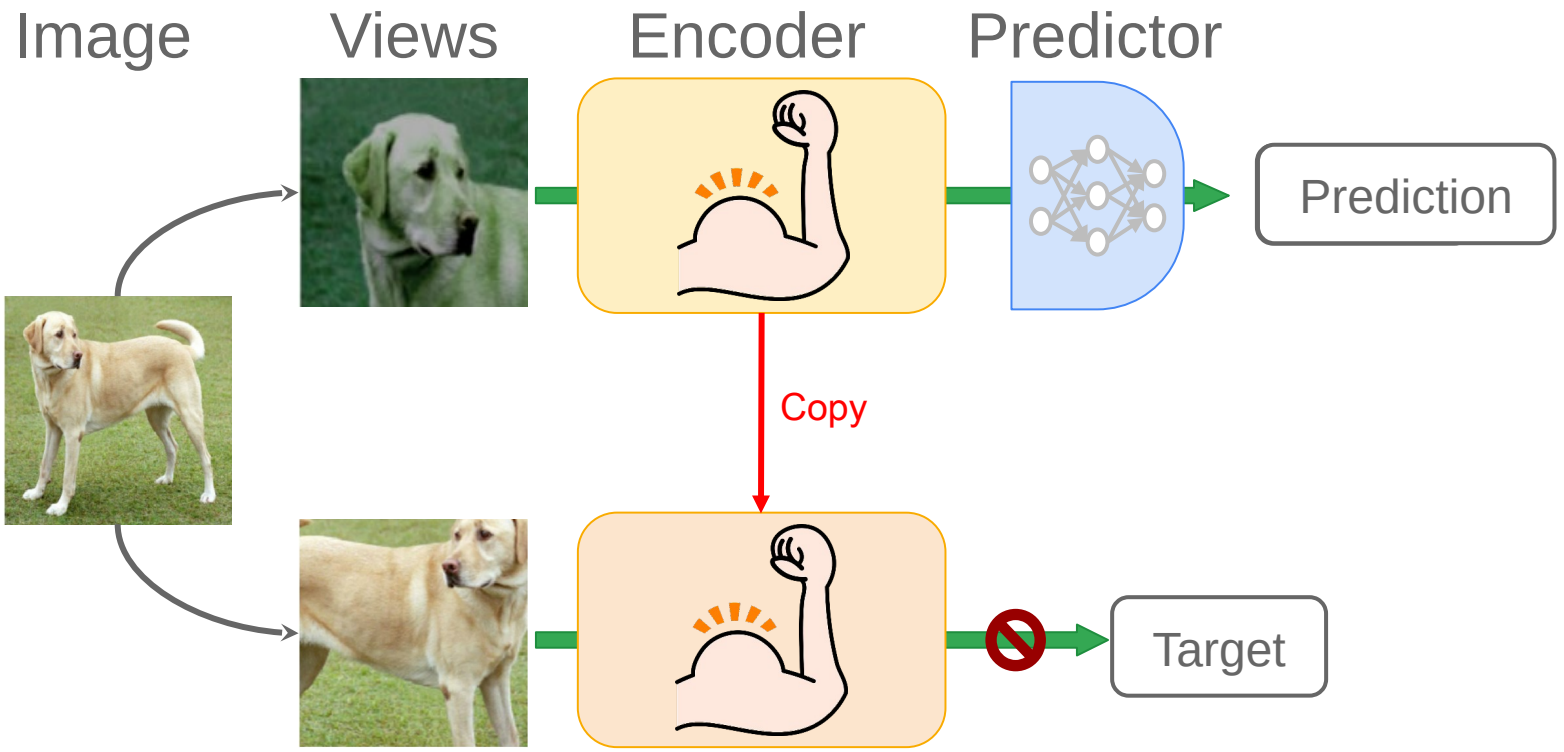
BYOL main intuition



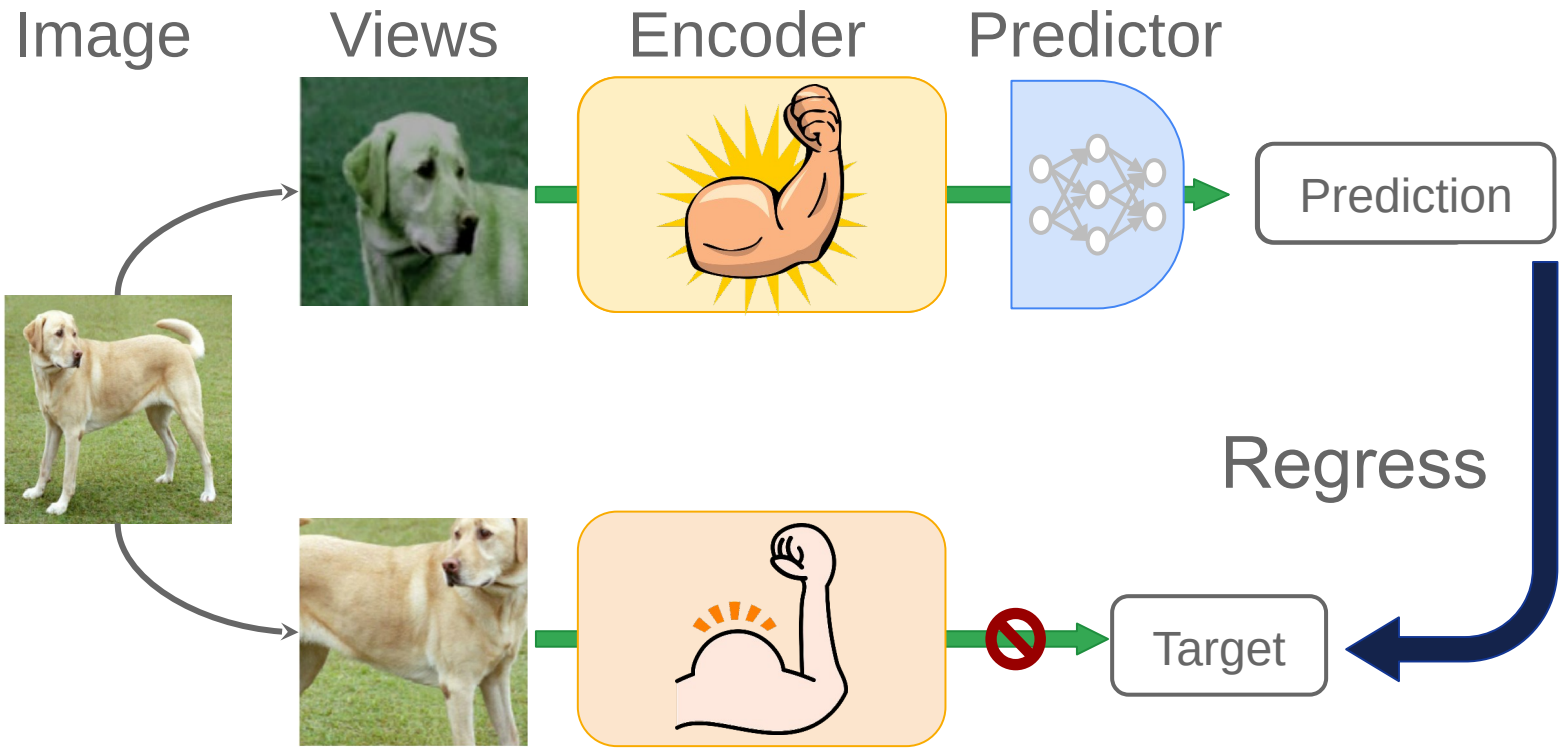
BYOL main intuition



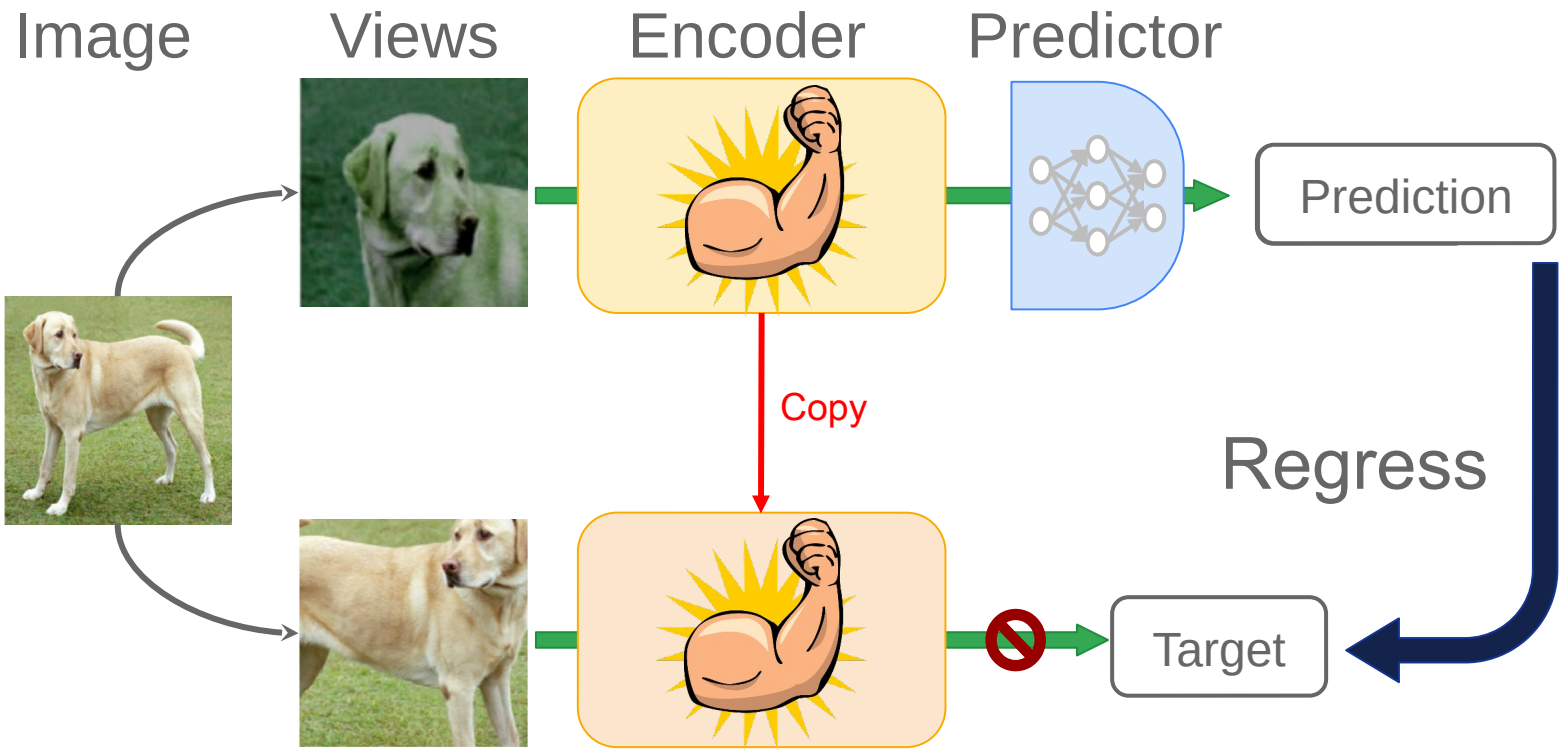
BYOL main intuition



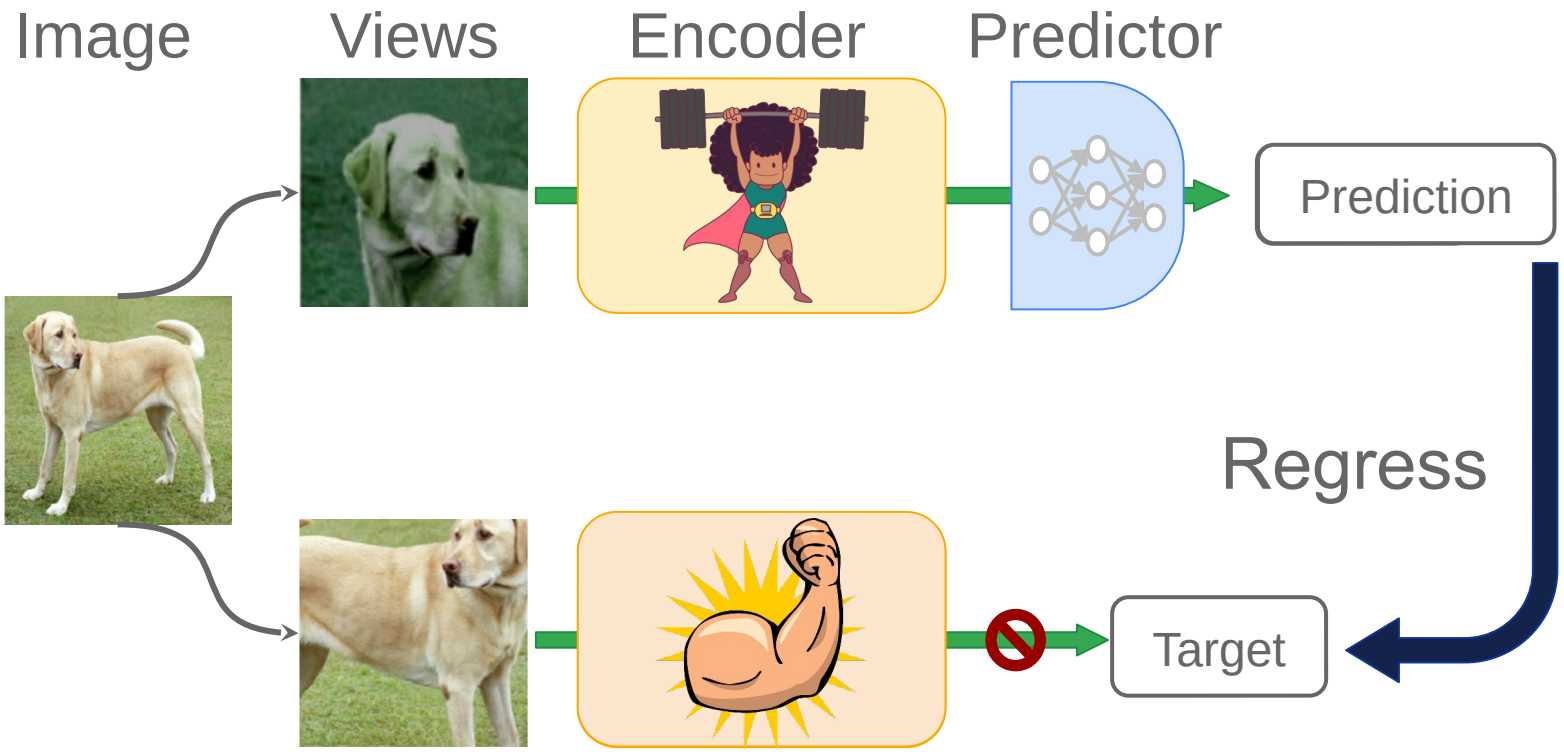
BYOL main intuition



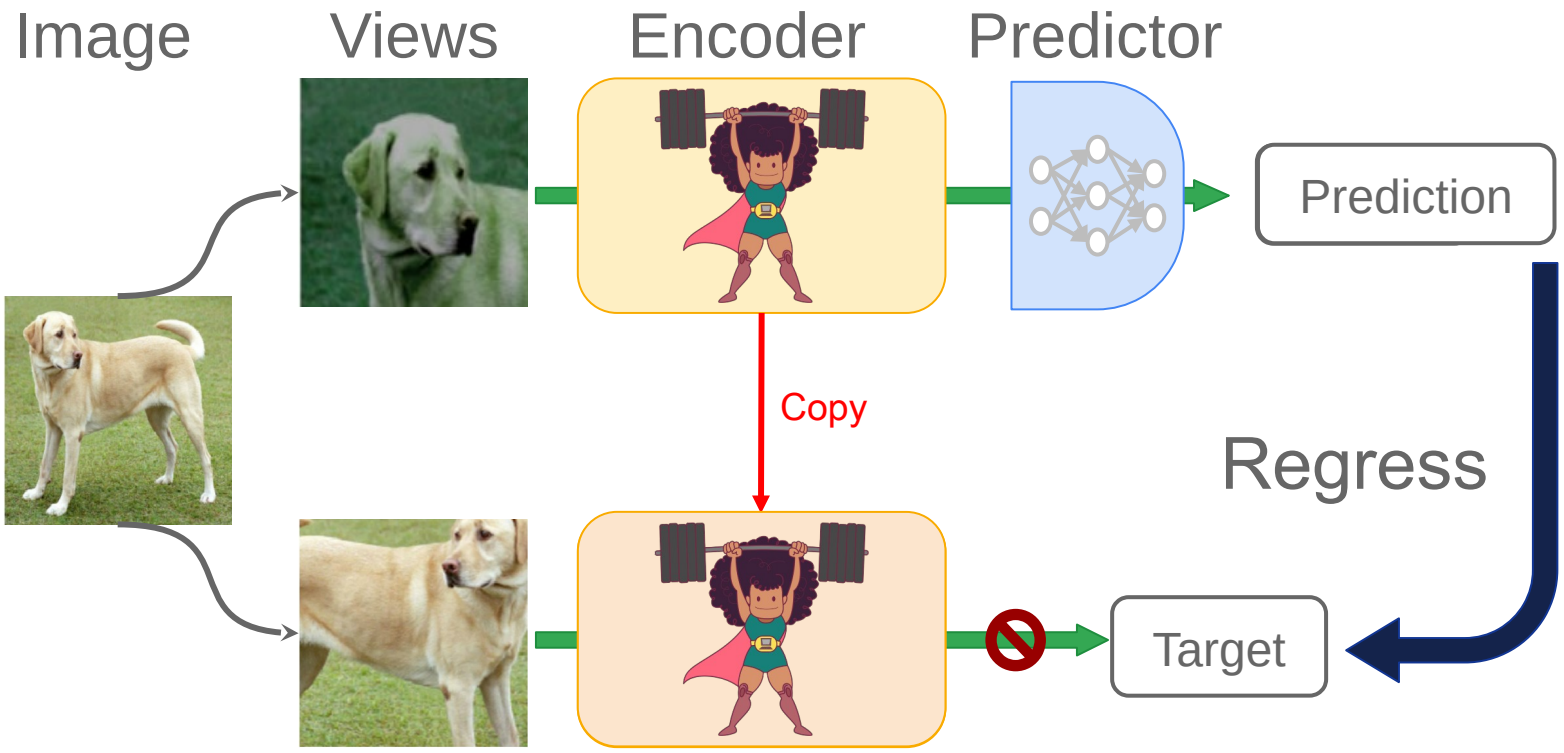
BYOL main intuition



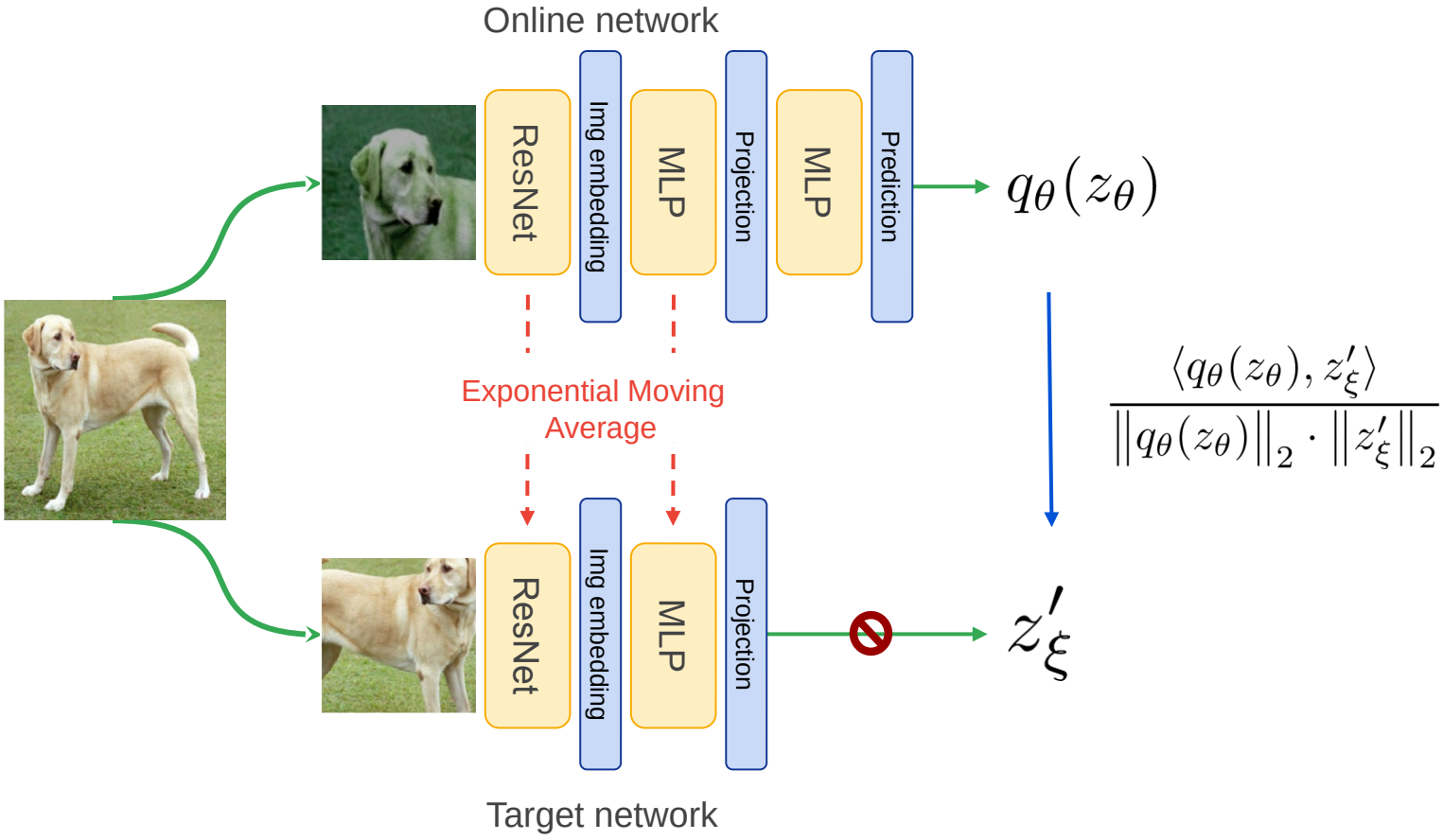
BYOL main intuition

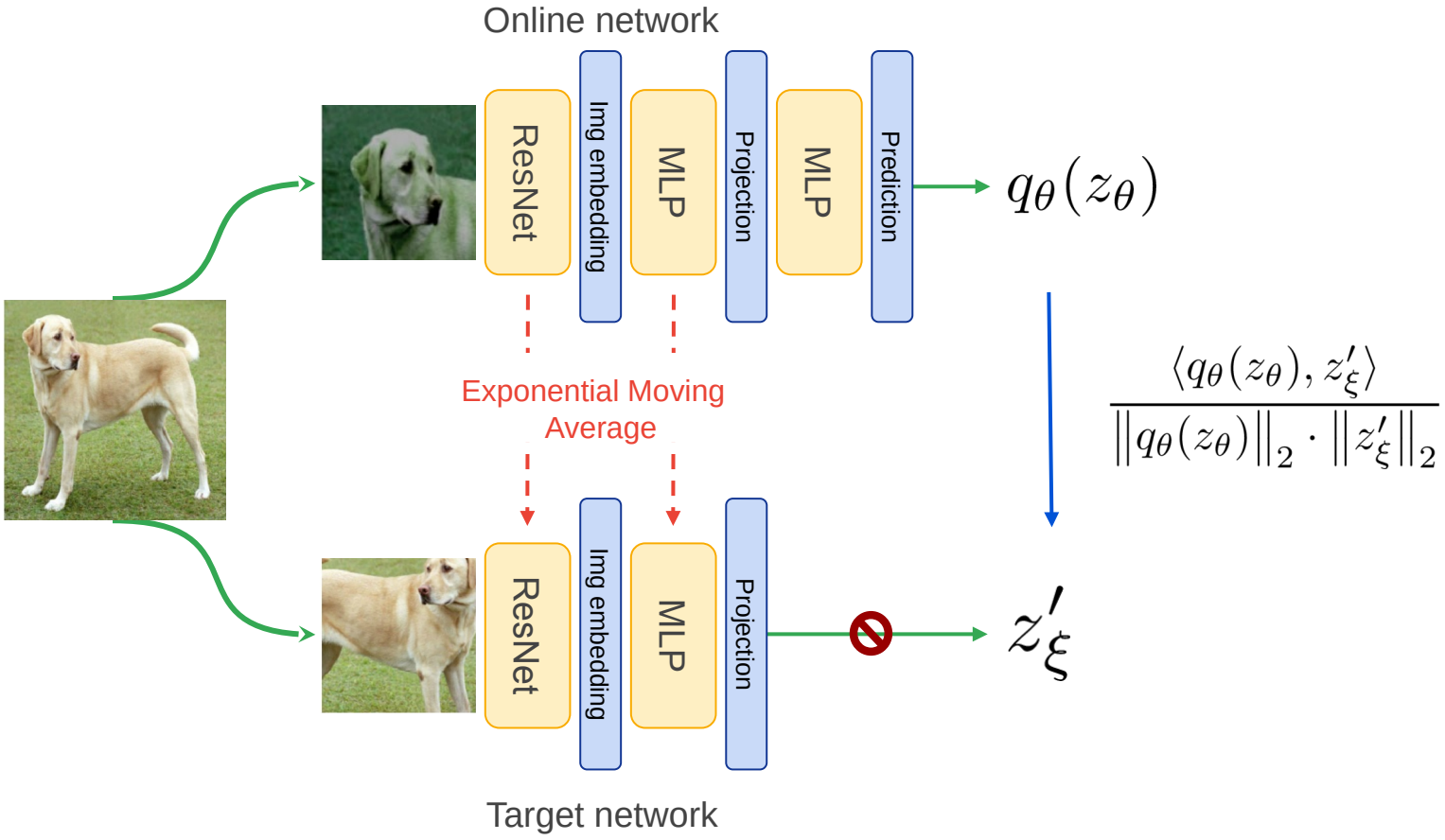


BYOL main intuition

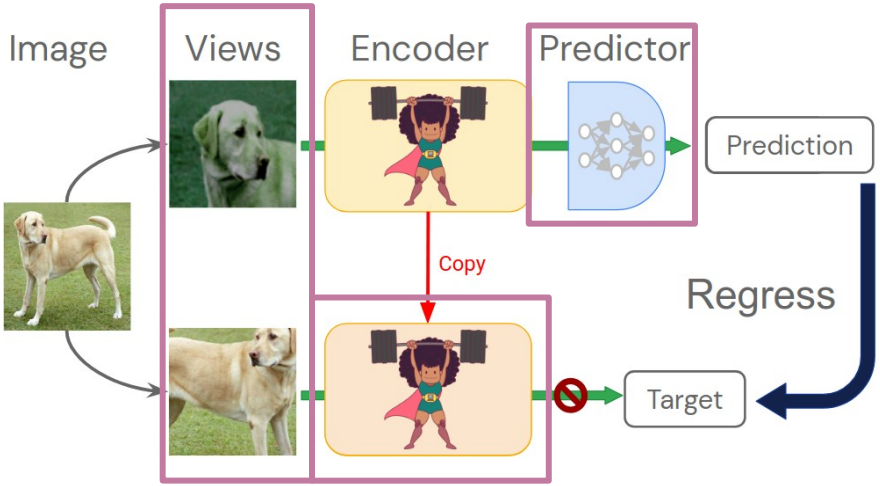


BYOL Architecture




Online network
ResNet
Img embedding
MLP
Projection
MLP
Prediction
$q_\theta(z_\theta)$
Exponential Moving Average
$\frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$
ResNet
Img embedding
MLP
Projection
z'_ξ
Target network

BYOL's highlight



Key ingredients:

- Image transformations.
- Target network.
- Additional predictor on top of online network.

Interest of the method:

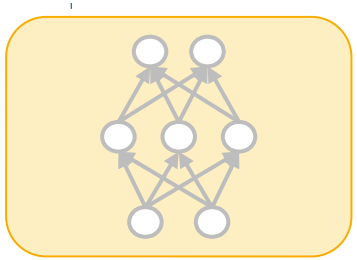
- Simple training procedure.
- No negative examples.
- Work at the embedding level, e.g. no-pseudo labels.

DeepMind

3 Performance

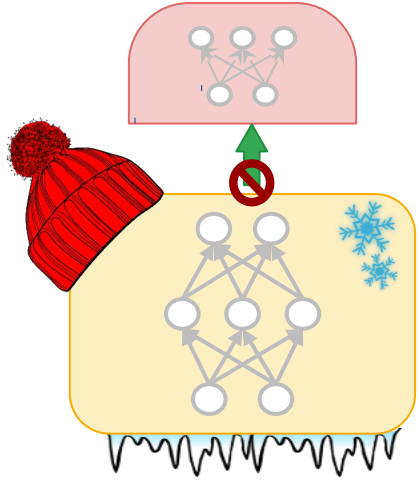
Linear Evaluation Protocol on ImageNet

Step 1: Train a “representation” on ImageNet without any labels.



ResNet

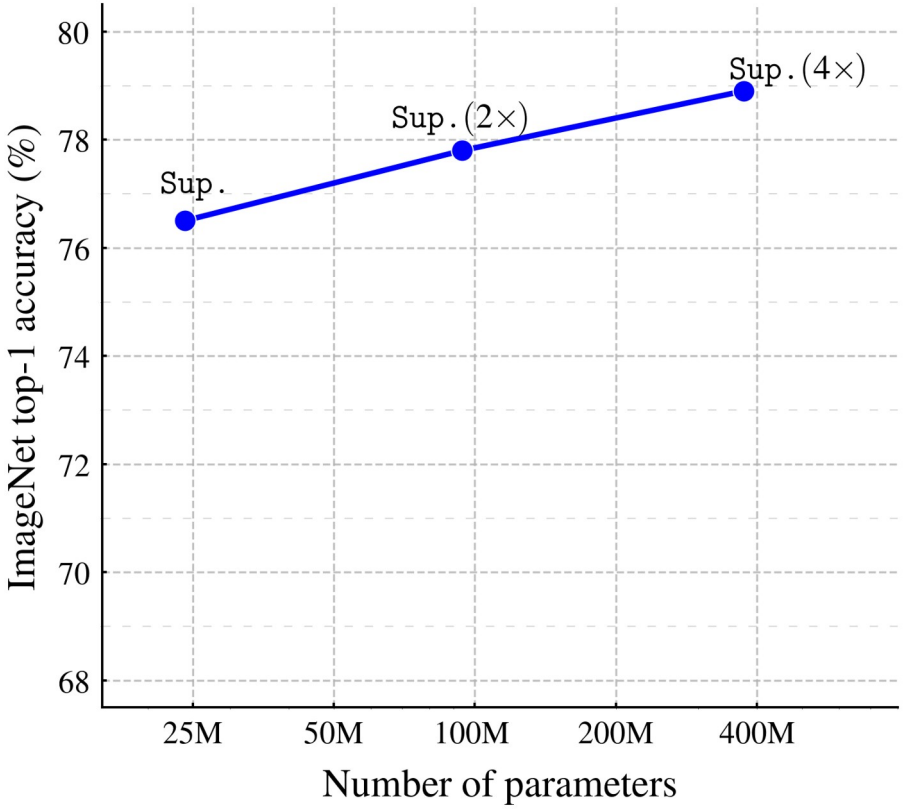
Step 2: On top of the frozen representation, train a linear classifier on ImageNet with label information.



Linear Classifier

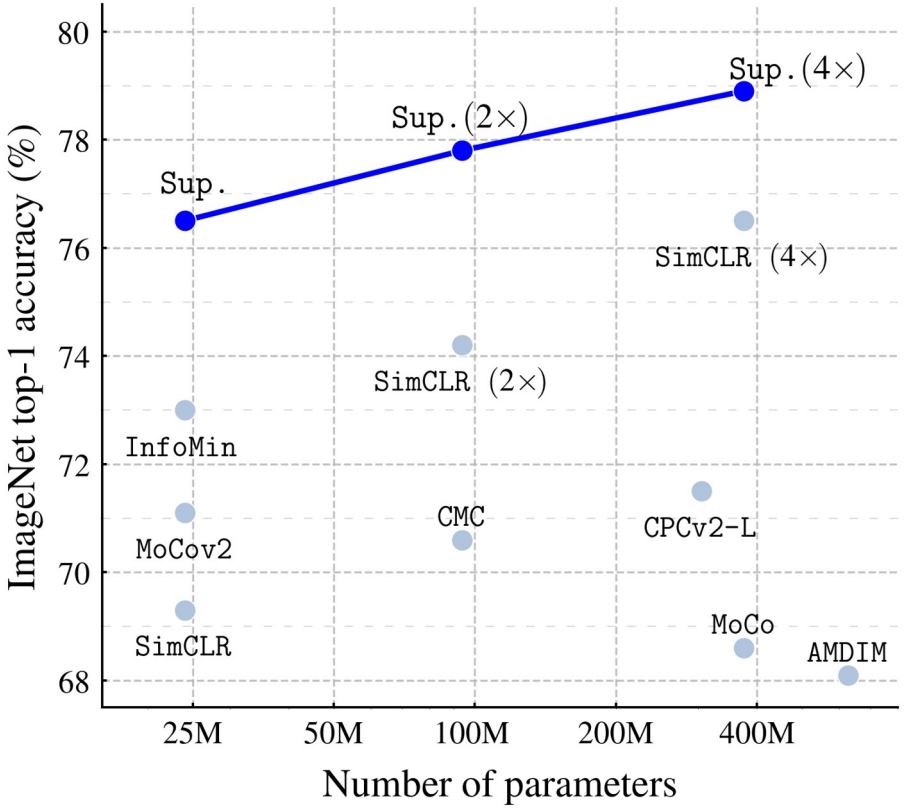
ResNet

Linear Evaluation Performance on ImageNet



Note: these supervised baselines are from SimCLR (Chen et al., ICML 2020)

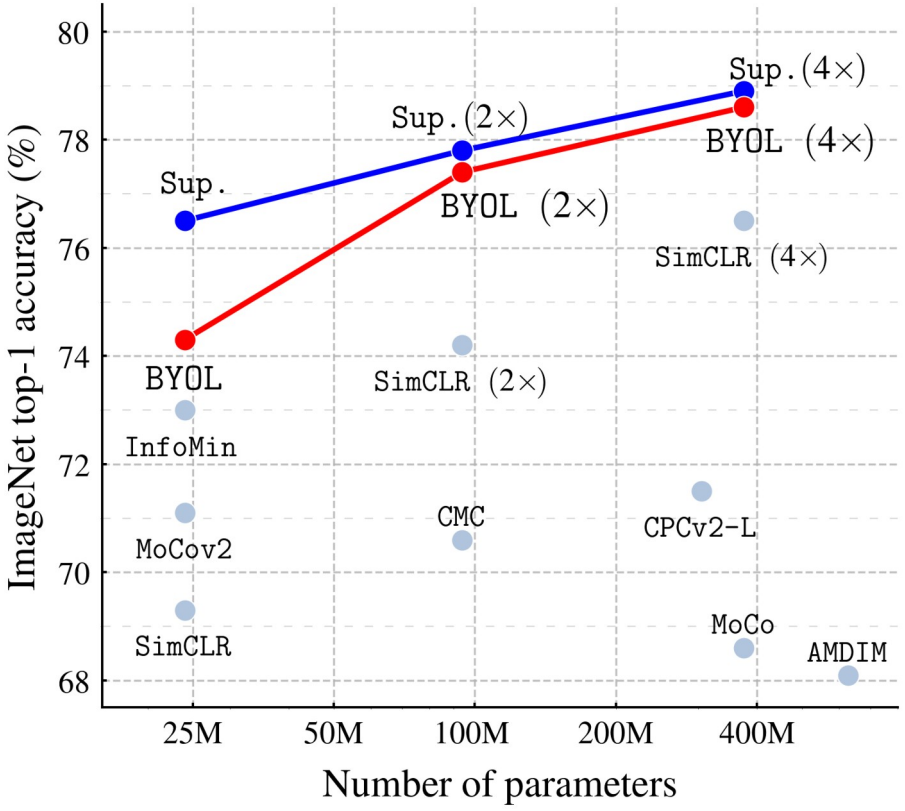
Linear Evaluation Performance on ImageNet



Note: these supervised baselines are from SimCLR (Chen & Hinton, ICML 2020)

CPCv2: van den Oord et al., *Representation learning with contrastive predictive coding*. 2018
AMDIM: Bachman et al., *Learning representations by maximizing mutual information across views*. 2019
CMC: Tian et al.,*Contrastive multiview coding*. 2019.
MoCo: He et al., *Momentum contrast for unsupervised visual representation learning.* 2019
InfoMin: Tian et al., *What makes for good views for contrastive learning*. 2020
MoCov2: Jain et al., *Improved baselines with momentum contrastive learning.* 2020
SimCLR: Chen et al., *A simple framework for contrastive learning of visual representations*. 2020

Linear Evaluation Performance on ImageNet



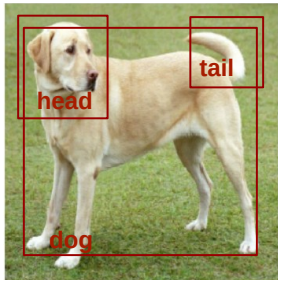
Note: these supervised baselines are from SimCLR (Chen & Hinton, ICML 2020)

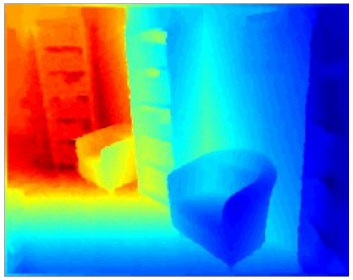
CPCv2: van den Oord et al., *Representation learning with contrastive predictive coding*. 2018
AMDIM: Bachman et al., *Learning representations by maximizing mutual information across views*. 2019
CMC: Tian et al.,*Contrastive multiview coding*. 2019.
MoCo: He et al., *Momentum contrast for unsupervised visual representation learning.* 2019
InfoMin: Tian et al., *What makes for good views for contrastive learning*. 2020
MoCov2: Jain et al., *Improved baselines with momentum contrastive learning.* 2020
SimCLR: Chen et al., *A simple framework for contrastive learning of visual representations*. 2020

Transfer Results

Semantic segmentation and object detection:

Method	AP_{50}	mIoU
Supervised-IN	74.4	74.4
MoCo	74.9	72.5
SimCLR (repro)	75.2	75.2
BYOL (ours)	**77.5**	**76.3**





Depth estimation:

Method	Higher better pct.<1.25	pct.$<1.25^2$	pct.$<1.25^3$	Lower better rms	rel
Supervised-IN	81.1	95.3	98.8	0.573	**0.127**
SimCLR (repro)	83.3	96.5	99.1	0.557	0.134
BYOL (ours)	**84.6**	**96.7**	**99.1**	**0.541**	0.129

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

Further comparison with SimCLR

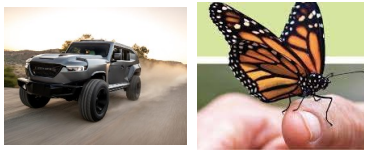
BYOL outperforms other self-supervised learning methods on the following benchmarks:

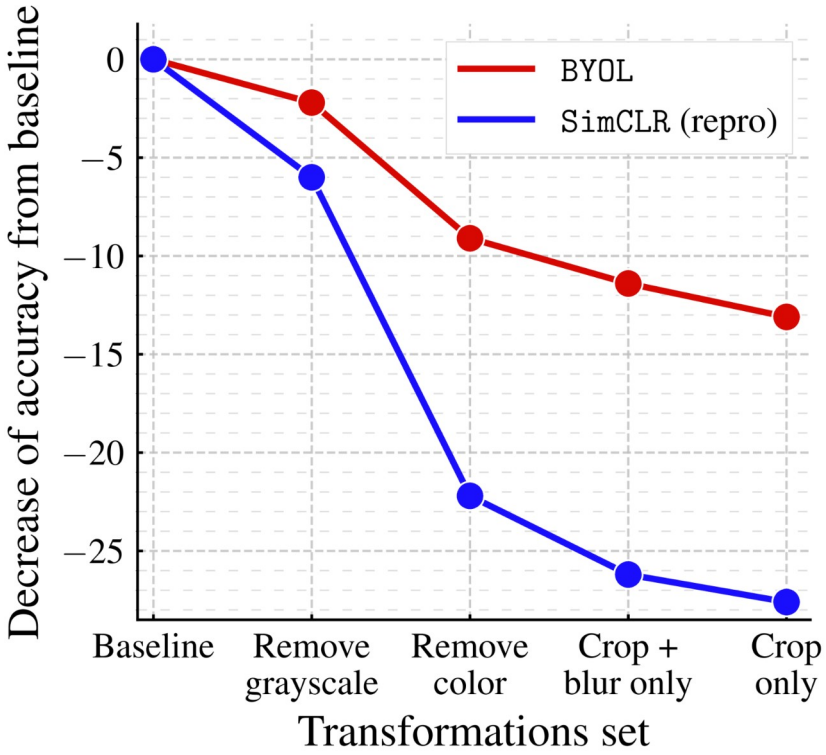
- Semi-supervised learning on ImageNet
- Fine-tuning on small classification datasets (such as CIFAR or Flowers)
- Transfer tasks when pretraining on Places365 instead of ImageNet

BYOL vs. Contrastive methods:

- BYOL is less sensitive to the choice of image transformations
- BYOL is more robust to smaller batch sizes

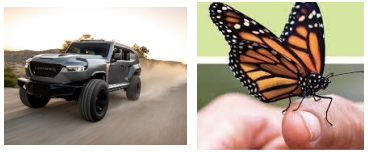
Sensitivity to augmentation choice

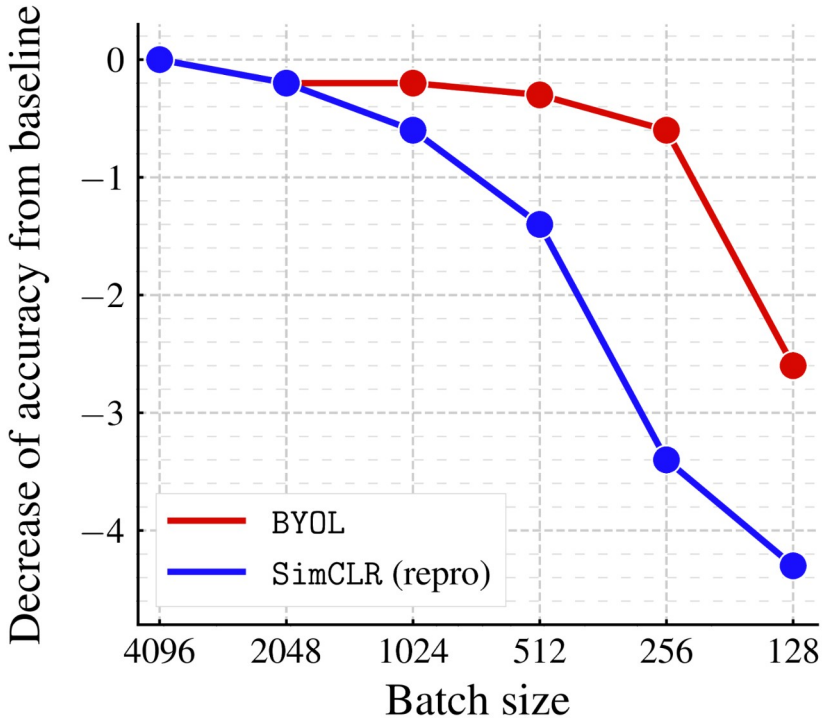




BYOL is **predictive** rather than **contrastive** ⇒ lower sensitivity to transformation set.

Sensitivity to batch size





No negative examples ⇒ lower sensitivity to batch size.

DeepMind

4 Building intuitions

Should BYOL collapse?

BYOL "loss"

$$\mathcal{L}_{\theta,\xi} := \|\bar{q}_\theta(z_\theta) - \bar{z}'_\xi\|^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$$

This loss has **trivial global minima** in the form of **collapsed constant projections and predictions**. But

Target update

$$\xi_{t+1} = \underbrace{(1-\eta)\xi_t + \eta\theta_{t+1}}_{\text{EMA update}} \neq \underbrace{\xi_t - \alpha\nabla_\xi \mathcal{L}_{\theta_t,\xi_t}}_{\text{GD update}}$$

BYOL is not optimizing $\mathcal{L}_{\theta,\xi}$

Constant representations are **equilibria** but may not be **stable** or **attractive**.

What factors prevent collapsing?

ImageNet top-1 accuracy @300 epochs

Base BYOL	**72.5**
- Remove predictor*	**Collapse**
- Remove EMA of target network (and keep the stop gradient)	**Barely learns**
- Add explicit negative examples	**72.7**

Remark: BYOL **without predictor** → Mean Teacher[1] but without supervised signal.

[1]Mean Teacher: Tarvainen et al., *Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2017.*

From BYOL to SimCLR & CPC-like methods

Adding the negative examples (NE) back in the loss

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_{i+N})/\tau)}{\sum_{k=1}^{2N} 1_{k \neq i} \exp(\text{sim}(z_i, z_k)/\tau)}$$

$$= \mathcal{L}_{\text{BYOL}} + \beta \cdot \mathcal{L}_{\text{NE}}$$

Three differences between SimCLR and BYOL:

- Use of a predictor
- Use of a target network
- No negative pairs (β= 0)

Method	Predictor	Target network	β	Top-1
BYOL	✓	✓	0	**72.5**
	✓	✓	1	70.9
		✓	1	70.7
SimCLR			1	69.4
	✓		1	69.1
	✓		0	0.3
		✓	0	0.2
			0	0.1

Can we remove the target network?

BYOL **with near optimal predictor** → Works **without** target network:

- Optimal linear predictor on the batch → 50.3% top1 accuracy
- Increased predictor learning rate

λ	Top-1
0	0.01
1	5.5
2	$62.8_{\pm 1.5}$
10	66.6
20	$66.3_{\pm 0.3}$
Baseline	72.5

$$\lambda = \frac{\text{predictor lr}}{\text{network lr}}$$

Hypothesis: Near-optimal predictor is key.

Why is it important to keep the predictor optimal?

Optimal predictor: Conditional expectation of targets w.r.t. online

$$q^*(z_\theta) = \mathbb{E}\left[z'_\xi \mid z_\theta\right]$$

Online projection objective: Conditional variance of targets w.r.t. Online

$$\mathcal{L}_{\theta,\xi} = \mathrm{Var}\left[z'_\xi \mid z_\theta\right]$$

z'_ξ: Target projection

z_θ: Online projection

To reduce conditional variance:

- **Collapse target** representation.
- **Increase** information in the online projection.

BYOL only plays on second one, (stop gradient in targets)

→ Always **increase the variability** of online projections!

DeepMind

Thank you!

The code and checkpoints are available:
https://github.com/deepmind/deepmind-research

Follow-up work on BYOL and BatchNorm:
https://arxiv.org/abs/2010.10241

BYOL works even without batch statistics

Result 1: BYOL indeed performs very poorly when all BN are removed (projection + prediction + encoder).

Hypothesis: BN provides a good init, doubly crucial for BYOL, both for optim and for providing good initial targets.

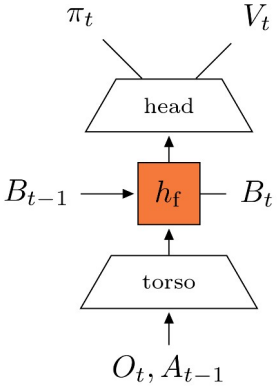
Experiments to test hypothesis: Can we recover perf with better inits and no batch statistics.

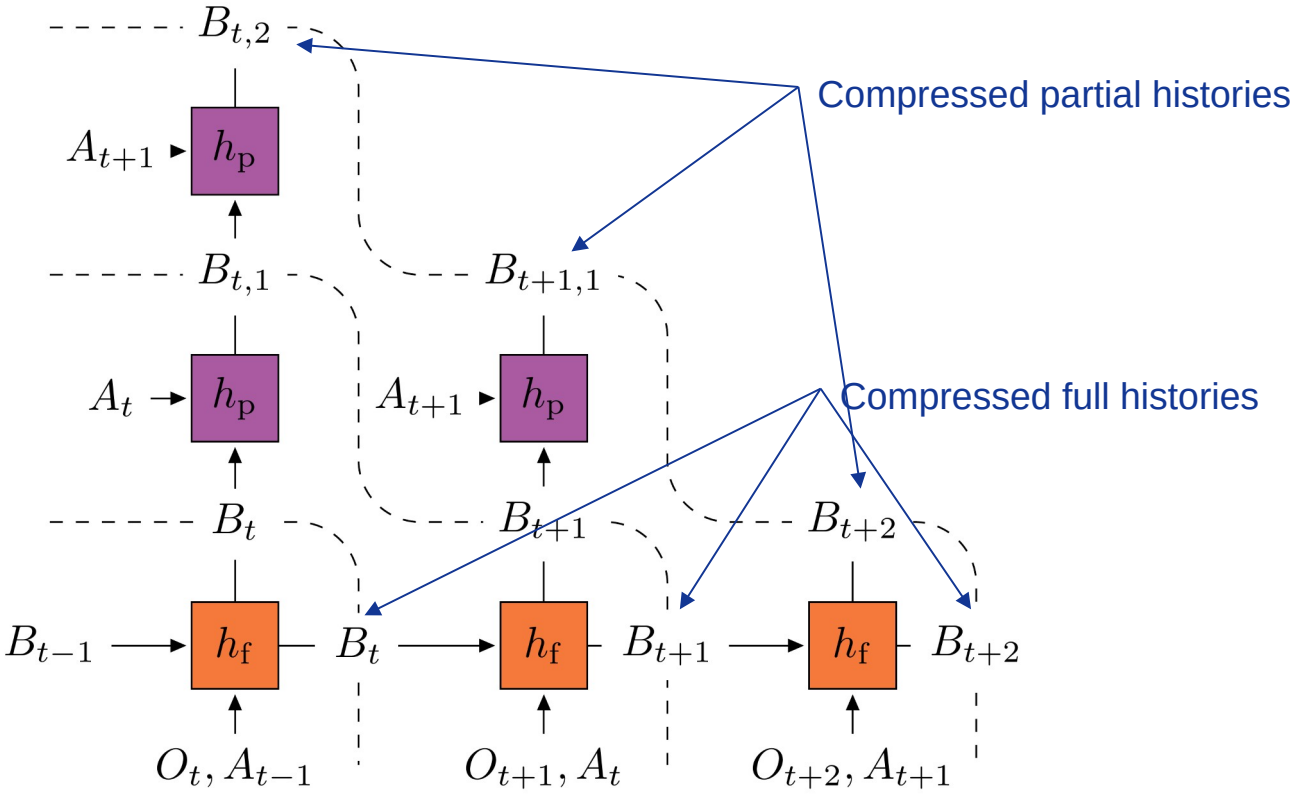
Result 2: BYOL does not collapse and works well with **better initialization**

Result 3: BYOL with **GroupNorm** and **WeightStandardization** (no batch stats) performs the same as BYOL with **BatchNorm**.

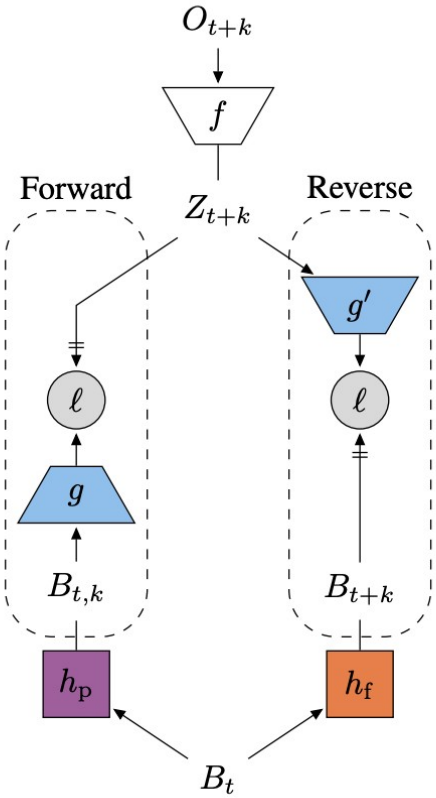
BYOL variant	Vanilla BN	No BN	Modified init.	GN + WS
Uses batch statistics	Yes	No	No	No
Top-1 accuracy (%)	74.3	0.1	65.7	73.9

PBL


π_t
V_t
head
B_{t-1}
h_f
B_t
torso
O_t, A_{t-1}


$B_{t,2}$
A_{t+1}
h_p
Compressed partial histories
$B_{t,1}$
$B_{t+1,1}$
A_t
h_p
A_{t+1}
h_p
Compressed full histories
B_t
B_{t+1}
B_{t+2}
B_{t-1}
h_f
B_t
h_f
B_{t+1}
h_f
B_{t+2}
O_t, A_{t-1}
O_{t+1}, A_t
O_{t+2}, A_{t+1}

PBL


O_{t+k}
f
Forward
Z_{t+k}
Reverse
g'
ℓ
ℓ
g
$B_{t,k}$
B_{t+k}
h_{p}
h_{f}
B_t

BYOL components

1) Compute view1 and view2

1) Retrieve the projection z_θ and z'_ξ

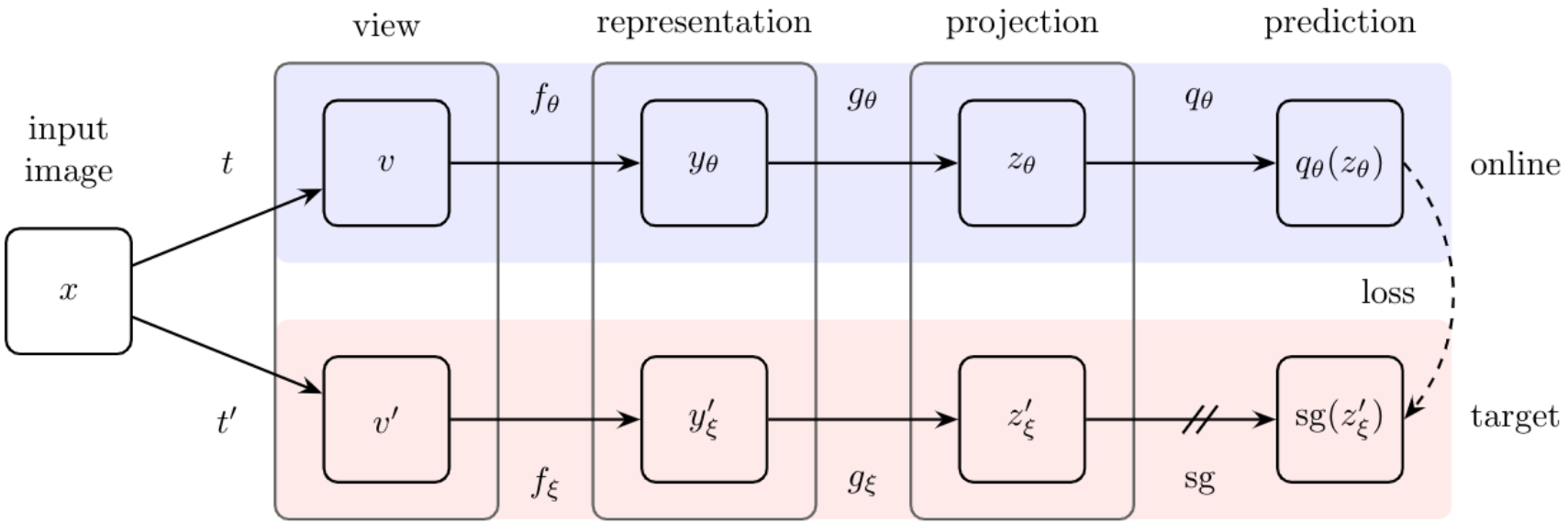
2) Learn to predict z'_ξ with a predictor $q_\theta(z_\theta)$ by maximizing cosine similarity

$$\mathcal{L}_\theta^{\texttt{BYOL}} = \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$$

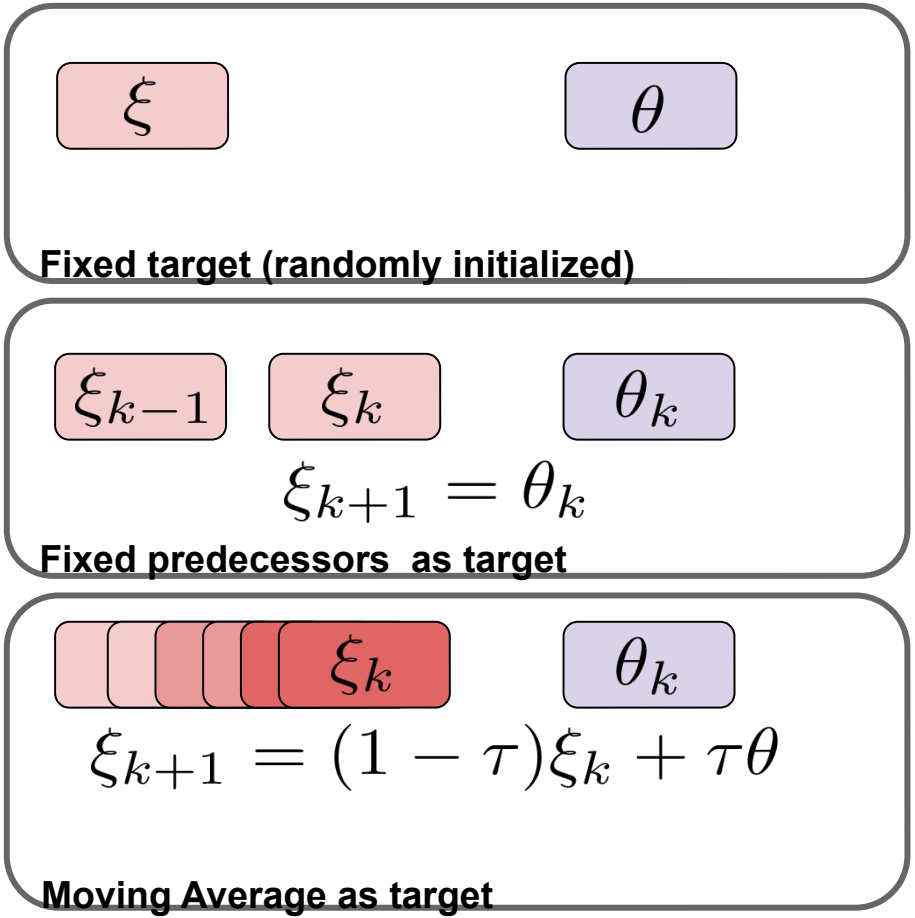
4) Update the target network

$$\xi_{k+1} = (1-\tau)\xi_k + \tau\theta$$

Architecture



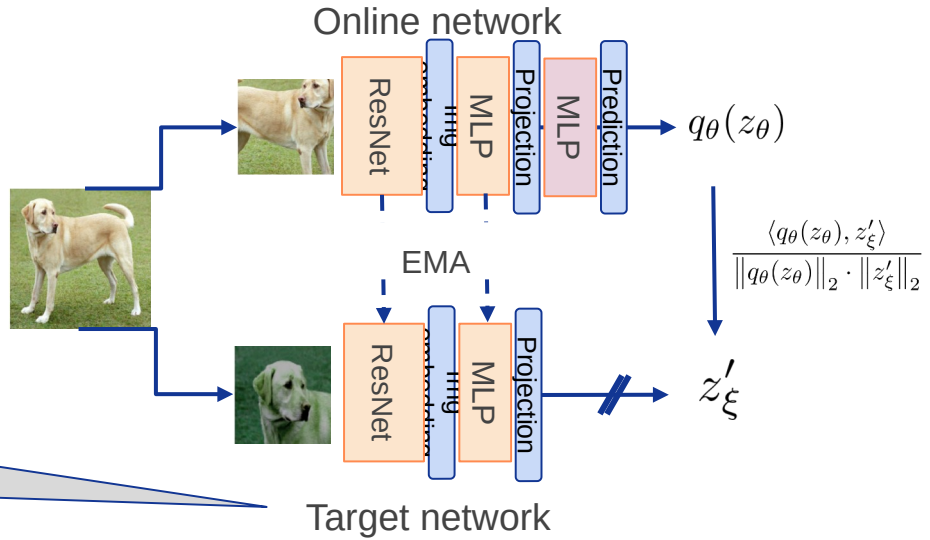
Update


ξ
θ
Fixed target (randomly initialized)
ξ_{k-1}
ξ_k
θ_k
$\xi_{k+1} = \theta_k$
Fixed predecessors as target
ξ_k
θ_k
$\xi_{k+1} = (1-\tau)\xi_k + \tau\theta$
Moving Average as target

Why BYOL does not collapse

Core component: **Predictor optimal**

Target network can be removed if we increase the learning rate of the predictor



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

The predictor optimal forces to increase activation variance
-> No constant !!!

$$\nabla_\theta \mathbb{E}\left[\left\| q^\star(z_\theta) - z'_\xi \right\|_2^2 \right] = \nabla_\theta \mathbb{E}\left[\left\| \mathbb{E}\left[z'_\xi | z_\theta \right] - z'_\xi \right\|_2^2 \right] = \nabla_\theta \mathbb{E}\left[\sum_i \mathrm{Var}(z'_{\xi,i} | z_\theta) \right]$$