Contrastive Learning

Datapoint 1

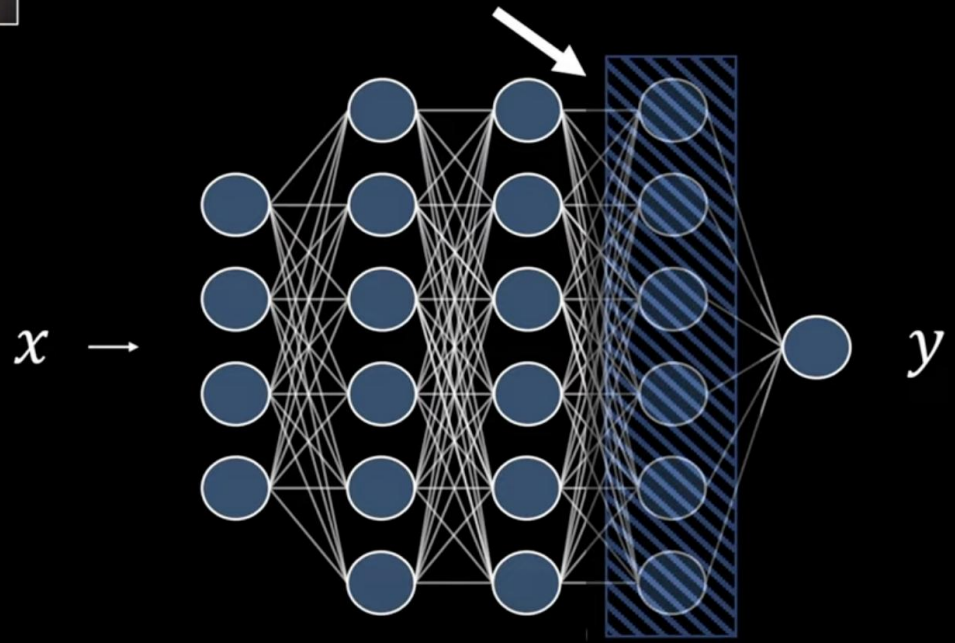
Representation 1

Datapoint 2

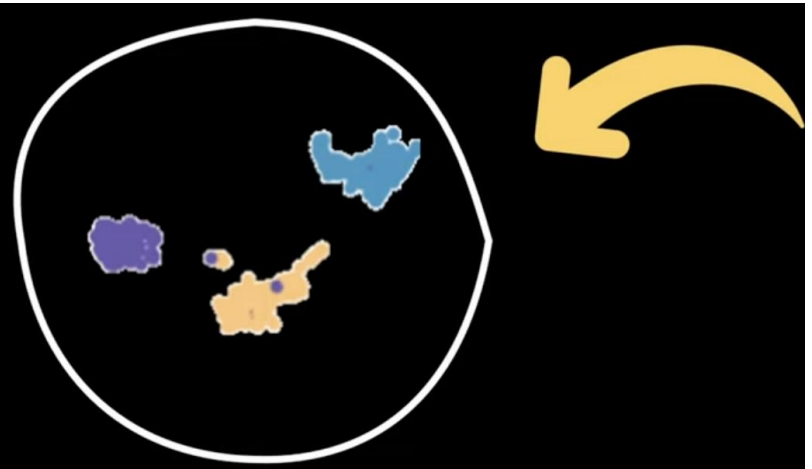
Representation 2

Datapoint 3

Representation 3



Representation learning



Embedding Space

Supervised

Cat

Dog

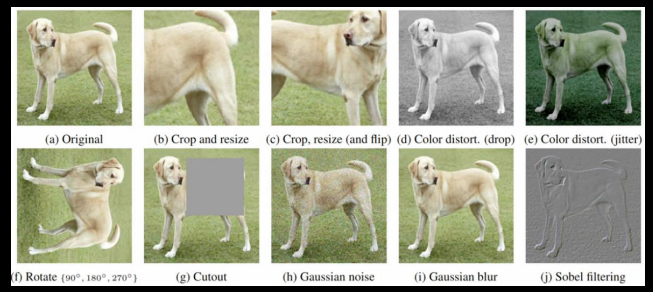
Cat

Loss	Architecture	rel. mCE	mCE
Cross-Entropy (baselines)	AlexNet [28]	100.0	100.0
	VGG-19+BN [44]	122.9	81.6
	ResNet-18 [17]	103.9	84.7
Cross-Entropy (our implementation)	ResNet-50	96.2	68.6
	ResNet-200	69.1	52.4
Supervised Contrastive	ResNet-50	**94.6**	**67.2**
	ResNet-200	**66.5**	**50.6**

Un- / Self-Supervised

Sampled data

Augmentation



Good Augmentations

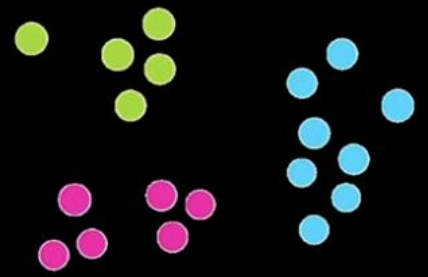
False Negatives

0.13	0.3	0.42	1.2	-0.3	1

0.22	0.6	0.52	1.5	-0.2	-0.3

1.22	-0.26	0.12	0.35	-0.1	-2.3

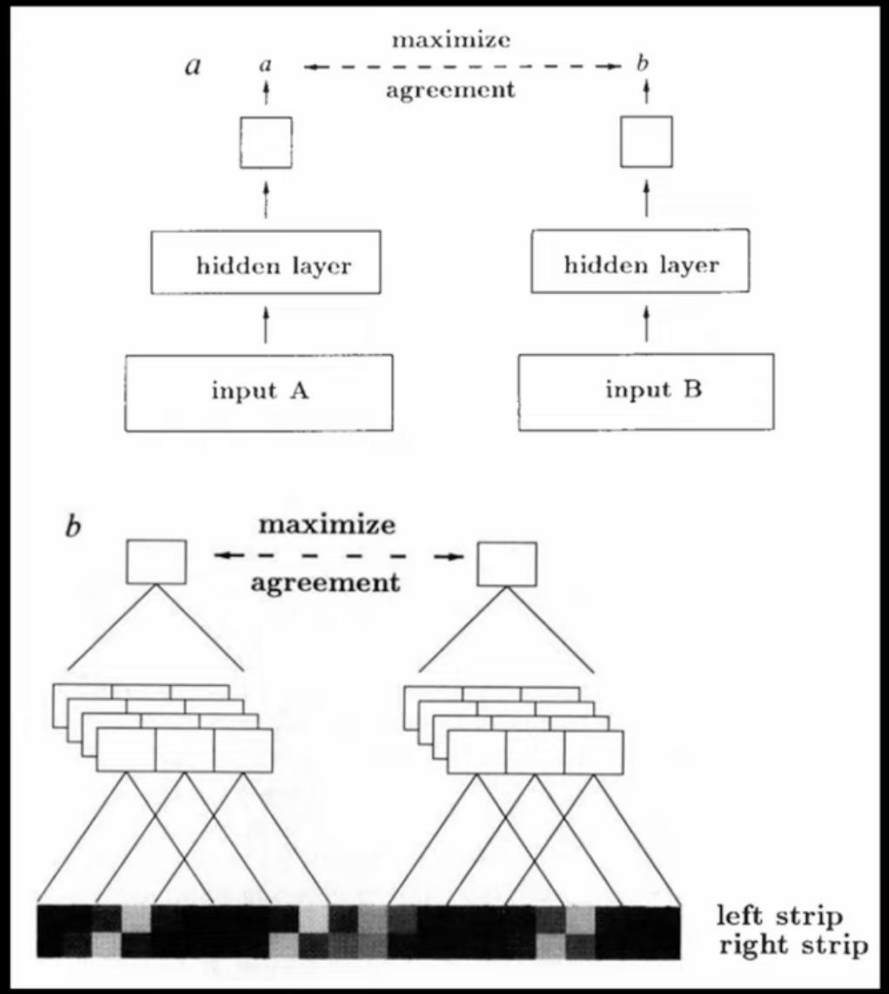
Generalizable representations



Clustering

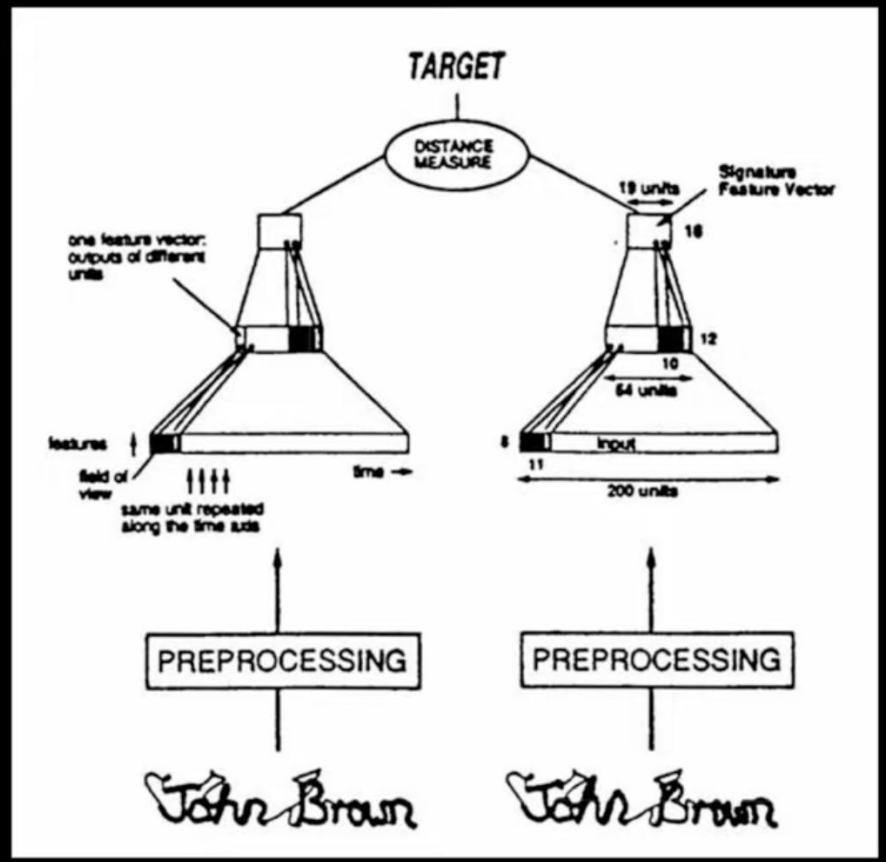
Segmentation

Object Detection



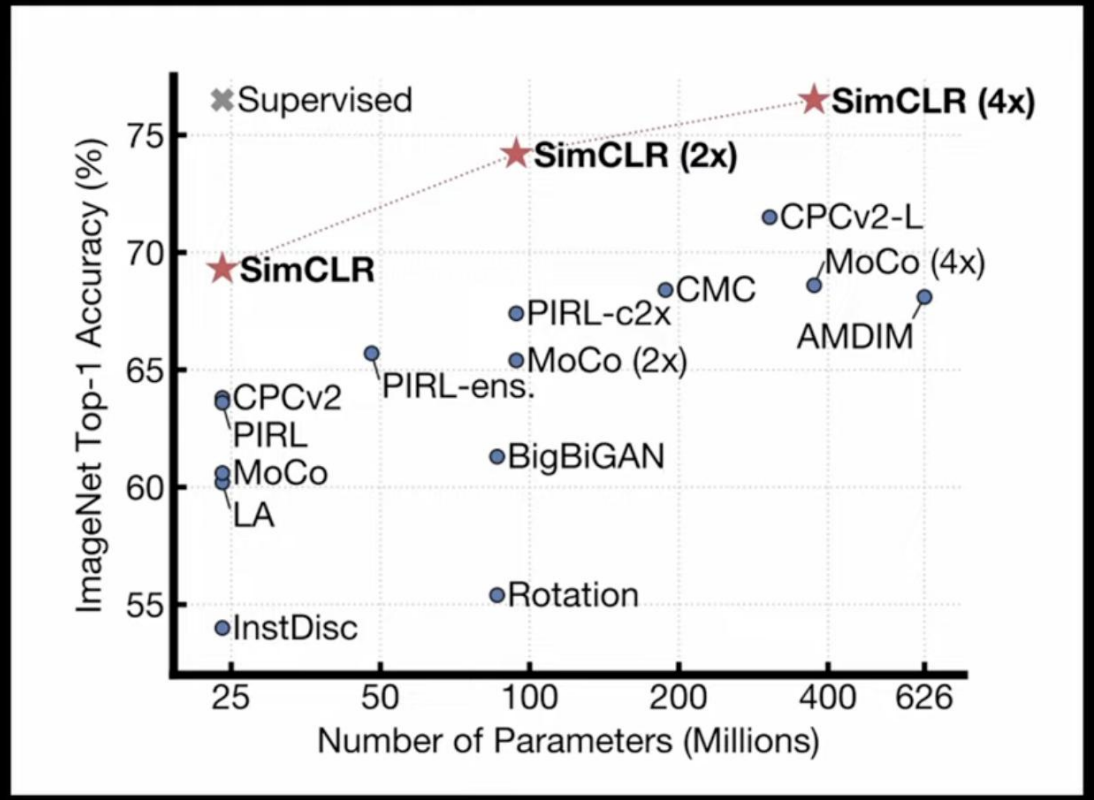
Self-organizing neural network that discovers surfaces in random-dot stereograms (1992)

S Becker, GE Hinton

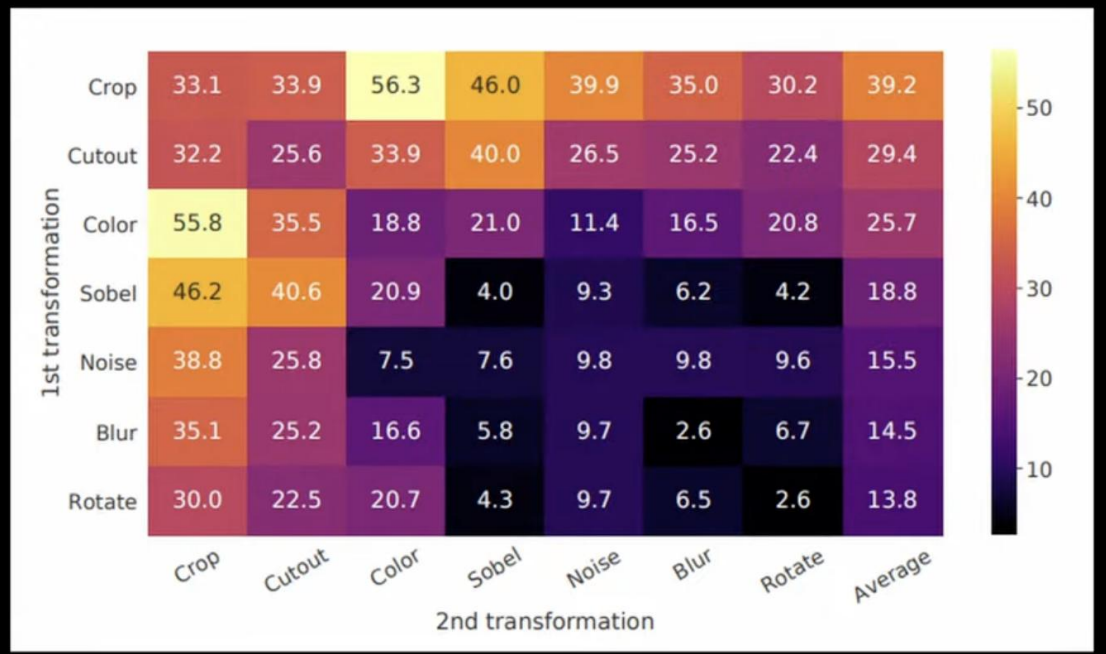


Signature verification using a" siamese" time delay neural network (1993)

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, Roopak Shah



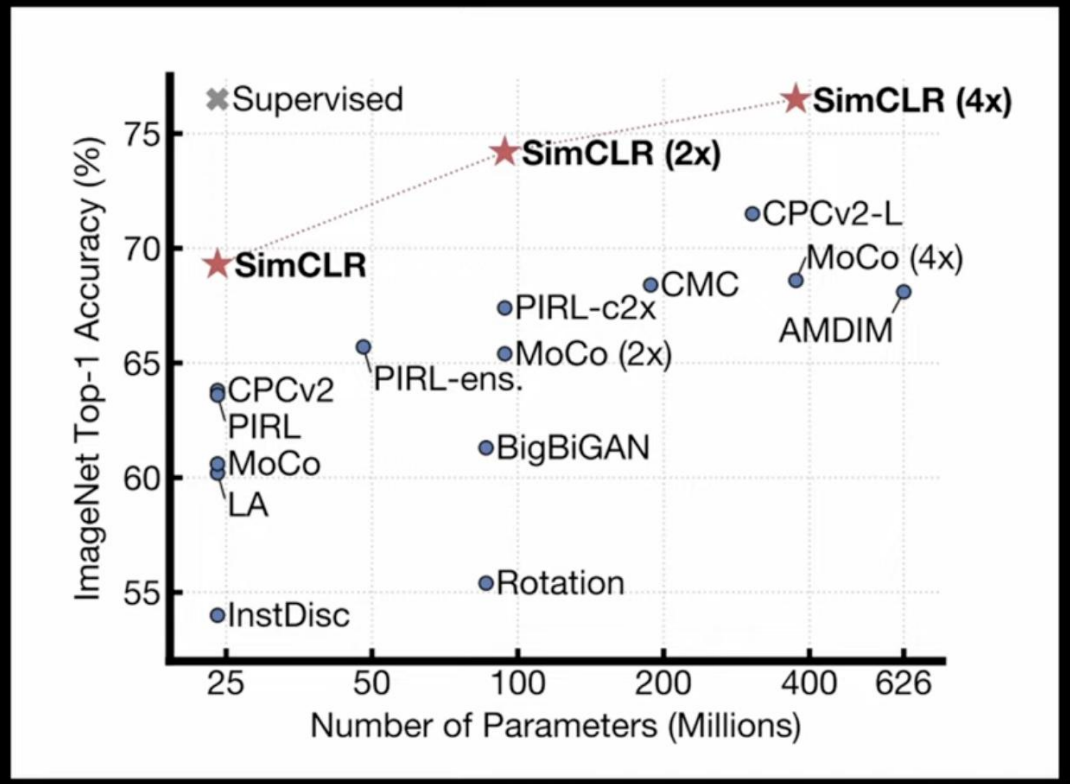
$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$



A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton

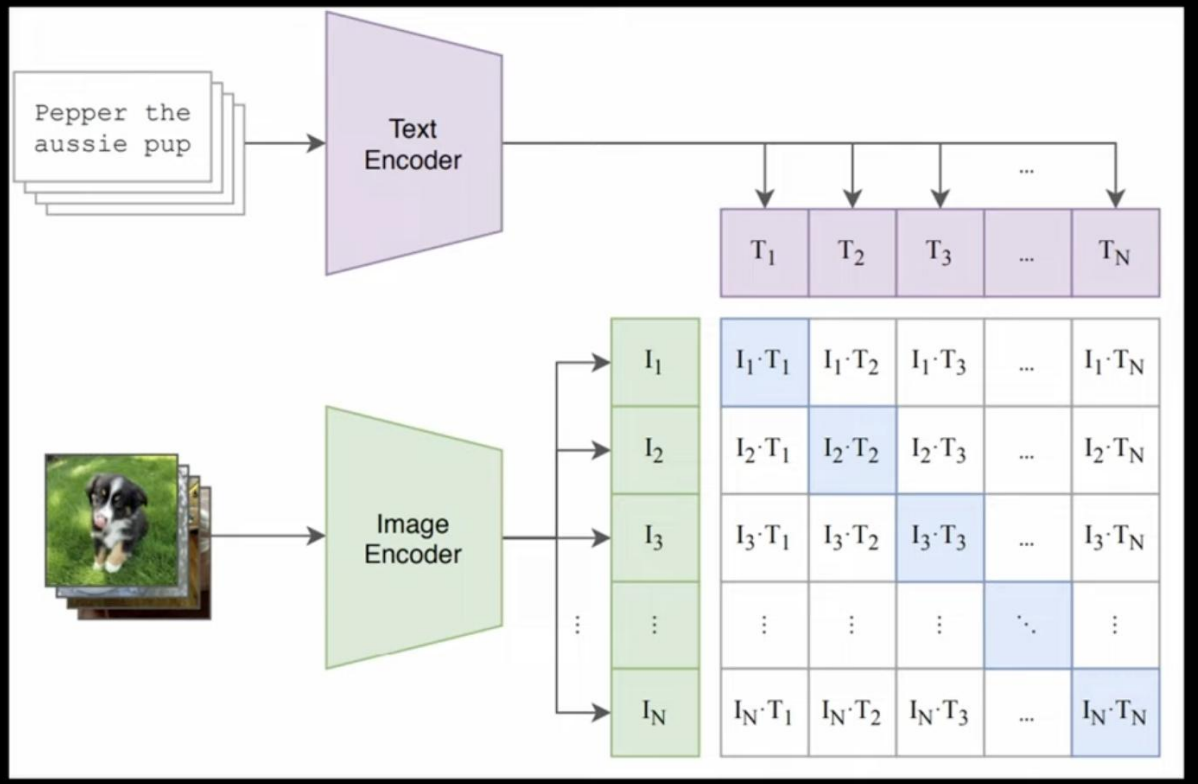
SimCLR



A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton

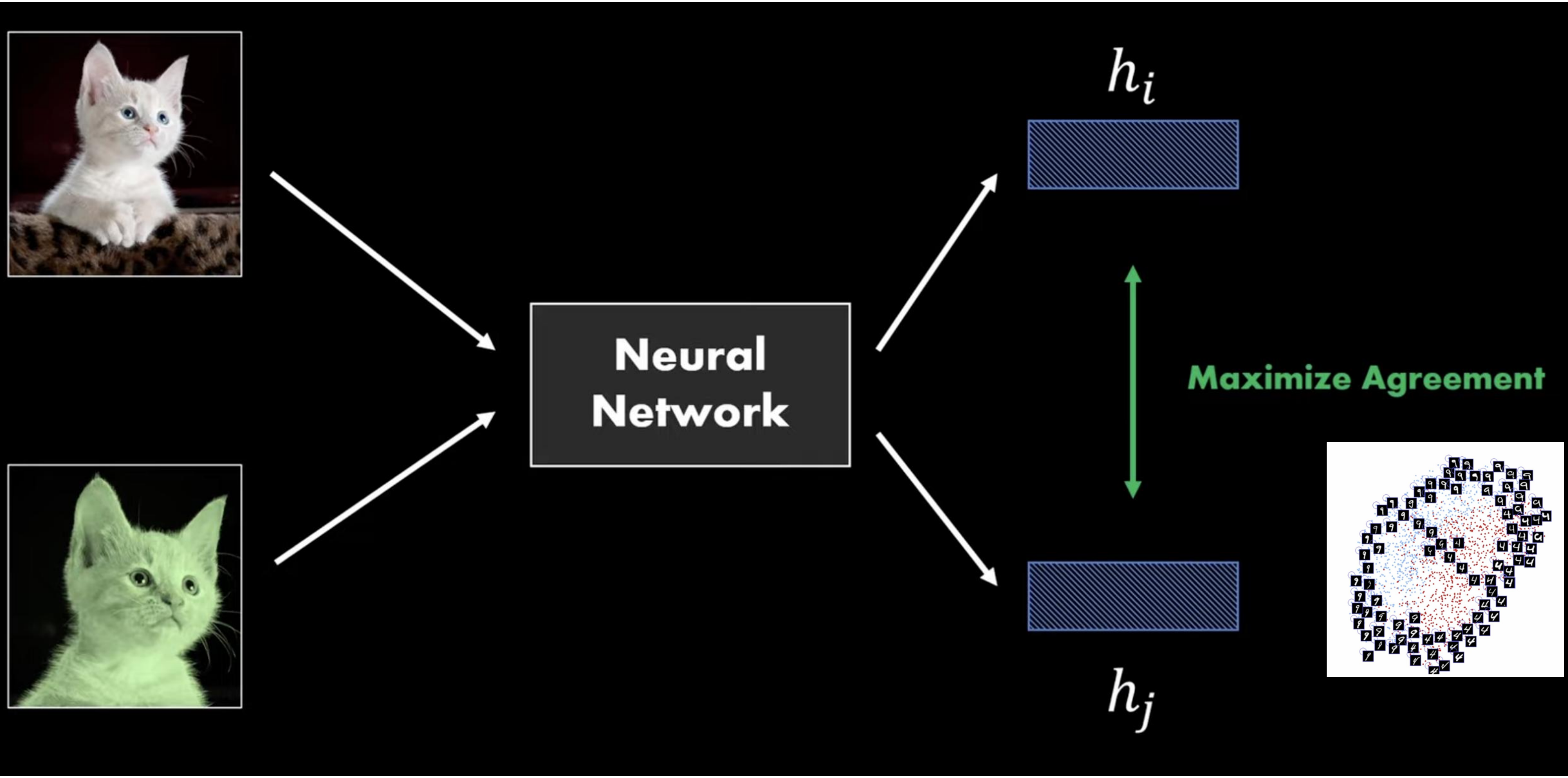
SimCLR



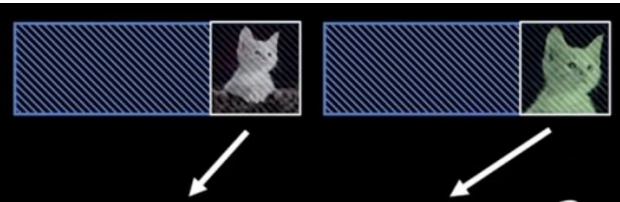
Learning Transferable Visual Models From Natural Language Supervision

Radford, Wook Kim et al.

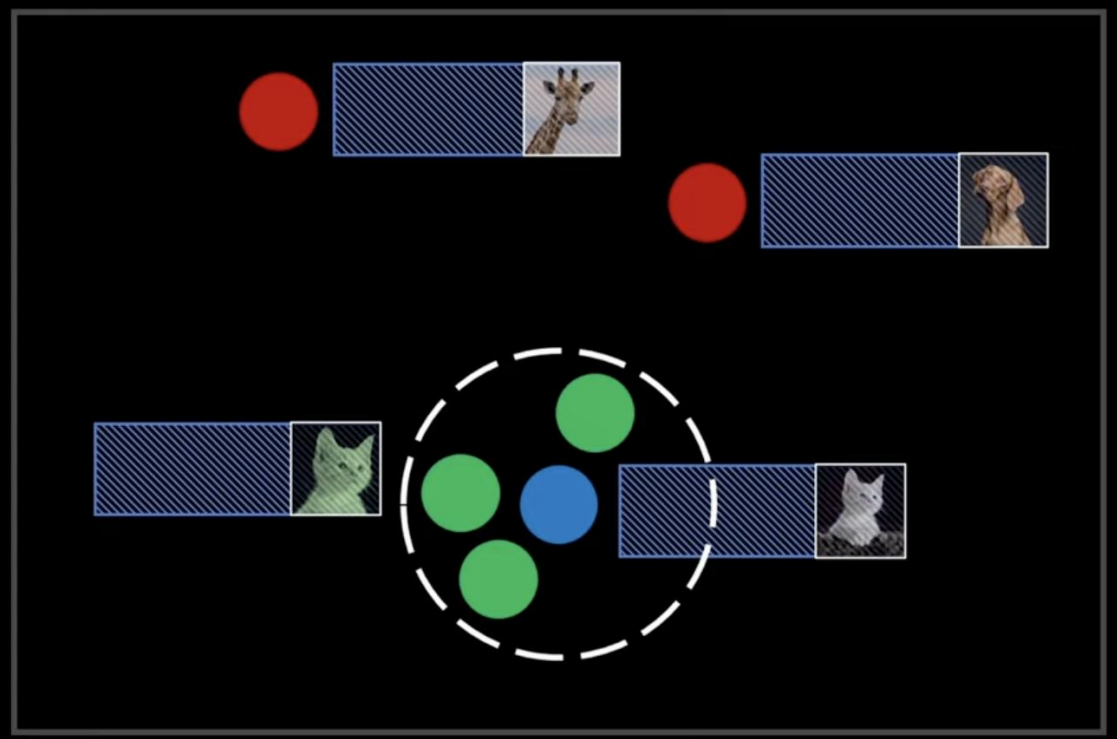
CLIP


h_i
Neural Network
Maximize Agreement
h_j

Contrastive Loss



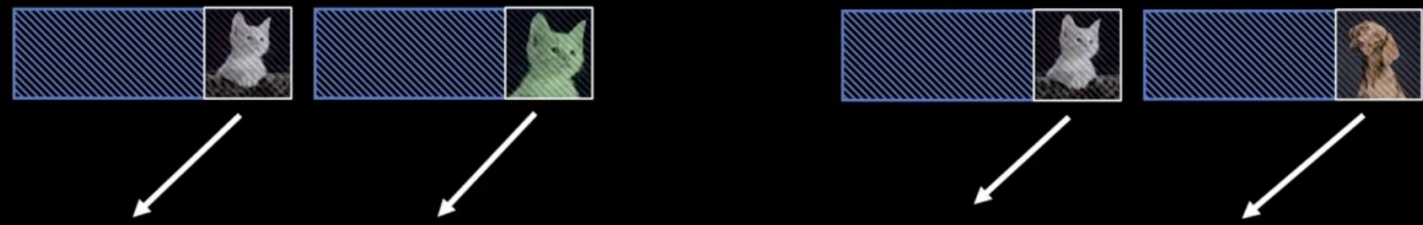
$$L(x_i, x_j) = \mathbb{1}[y_i = y_j]\left\|M(x_i) - M(x_j)\right\|_2^2 +$$
$$\mathbb{1}[y_i \neq y_j]\max(0, \varepsilon - \left\|M(x_i) - M(x_j)\right\|_2)^2$$



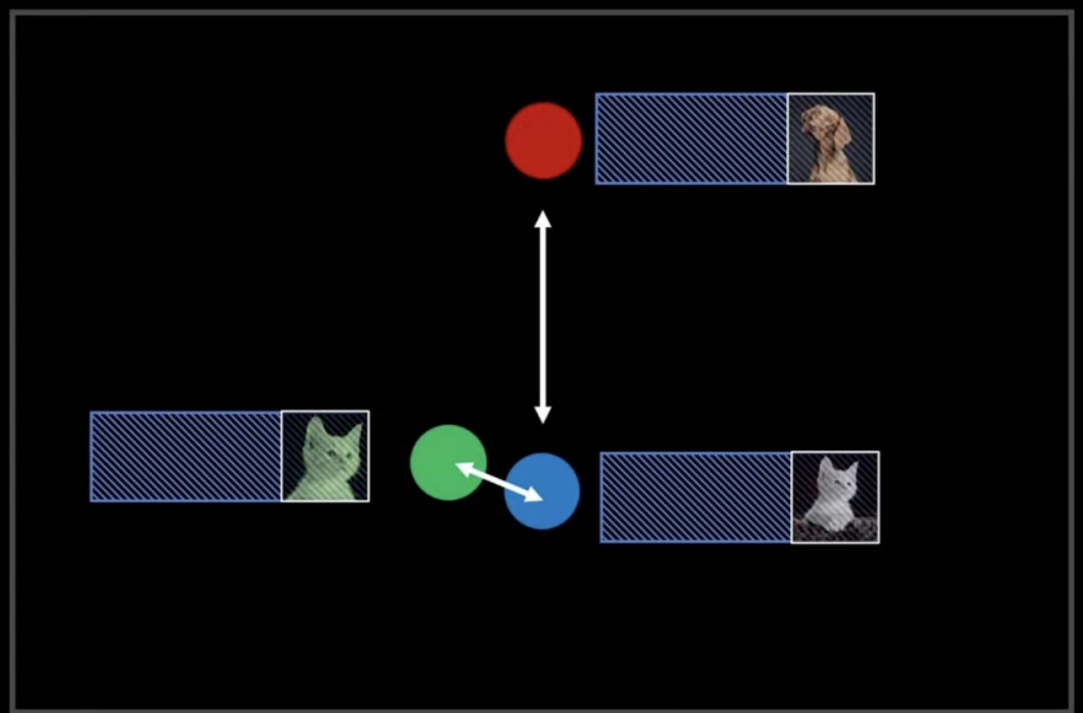
2005

Sumit Chopra, Raia Hadsell, Yann LeCun

Triplet Loss



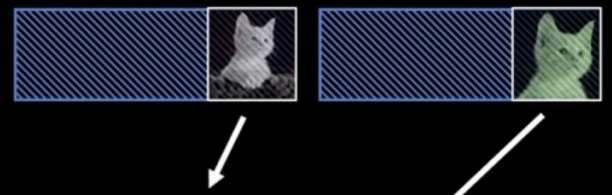
$$L\left(x_i^a, x_i^p, x_i^n\right) = \left\|M(x_i^a) - M(x_i^p)\right\|_2^2 + \alpha - \left\|M(x_i^a) - M(x_i^n)\right\|_2^2$$



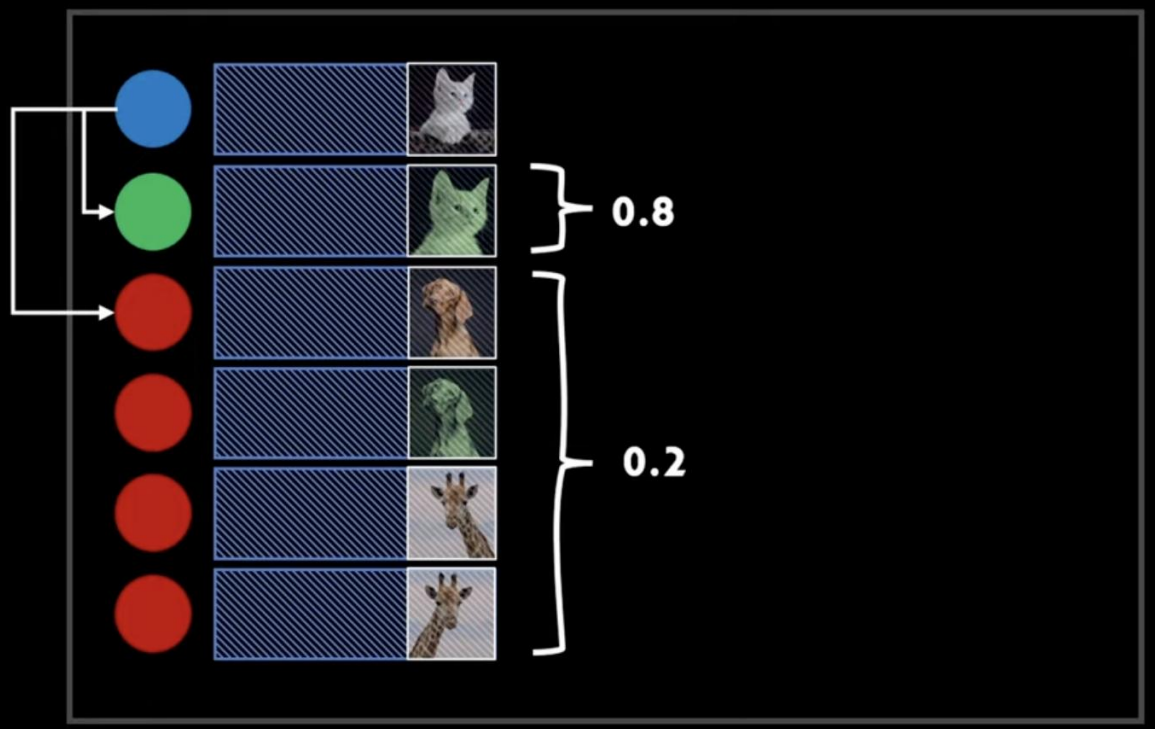
2015

Florian Schroff, Dmitry Kalenichenko, James Philbin

NT-Xent Loss



$$L(x_i, x_j) = -\log \frac{\exp\left(sim\left(M(x_i), M(\mathrm{x}_j)\right)\right)/\mathrm{T}}{\sum_{k=1}^{2N} \mathbb{1}[k \neq i] \exp\left(sim\left(M(x_i), M(\mathrm{x}_k)\right)\right)/\mathrm{T}}$$



2020

Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton

InfoNCE

交叉熵损失(Cross-entropy Loss) 是分类问题中默认使用的损失函数：

$$L_{CE} = -\sum_{c} I\left(y_i = c\right) \log P\left(y = c \mid X_i\right)$$

分类模型中，最后一层一般是linear layer+softmax。所以如果将之前的特征视为$f(X_i)$，linear layer的权重视为W，则有：

$$P(y = c|X_i) = \frac{\exp(W_c^T f(X_i))}{\sum_p \exp(W_p^T f(X_i))}$$

每个权重矩阵 W 事实上代表了每一类样本其特征值的模板（根据向量乘法我们知道越相似的两个向量其内积越大）。

实际上，现有的分类问题是通过一系列深度网络提取特征，然后依据大量的样本学习到一个有关每一类样本特征的模板。在测试的阶段则将这个学到的特征模板去做比对。

对比损失

非参数样本分类：

所谓非参数样本分类，则是将每个计算出的样本特征作为模板，即看做是计算所得的样本特征模板。

$$P(y=c|X_i)=\frac{\exp(f(X_c)^T f(X_i))}{\sum_p \exp(f(X_p)^T f(X_i))}$$

对比损失：

我们最终的目标 $d(f(X),f(X^-)) \gg d(f(X),f(X^+))$

假设已经归一化特征值，则优化上式实际上等同于最大化下式中的softmax概率

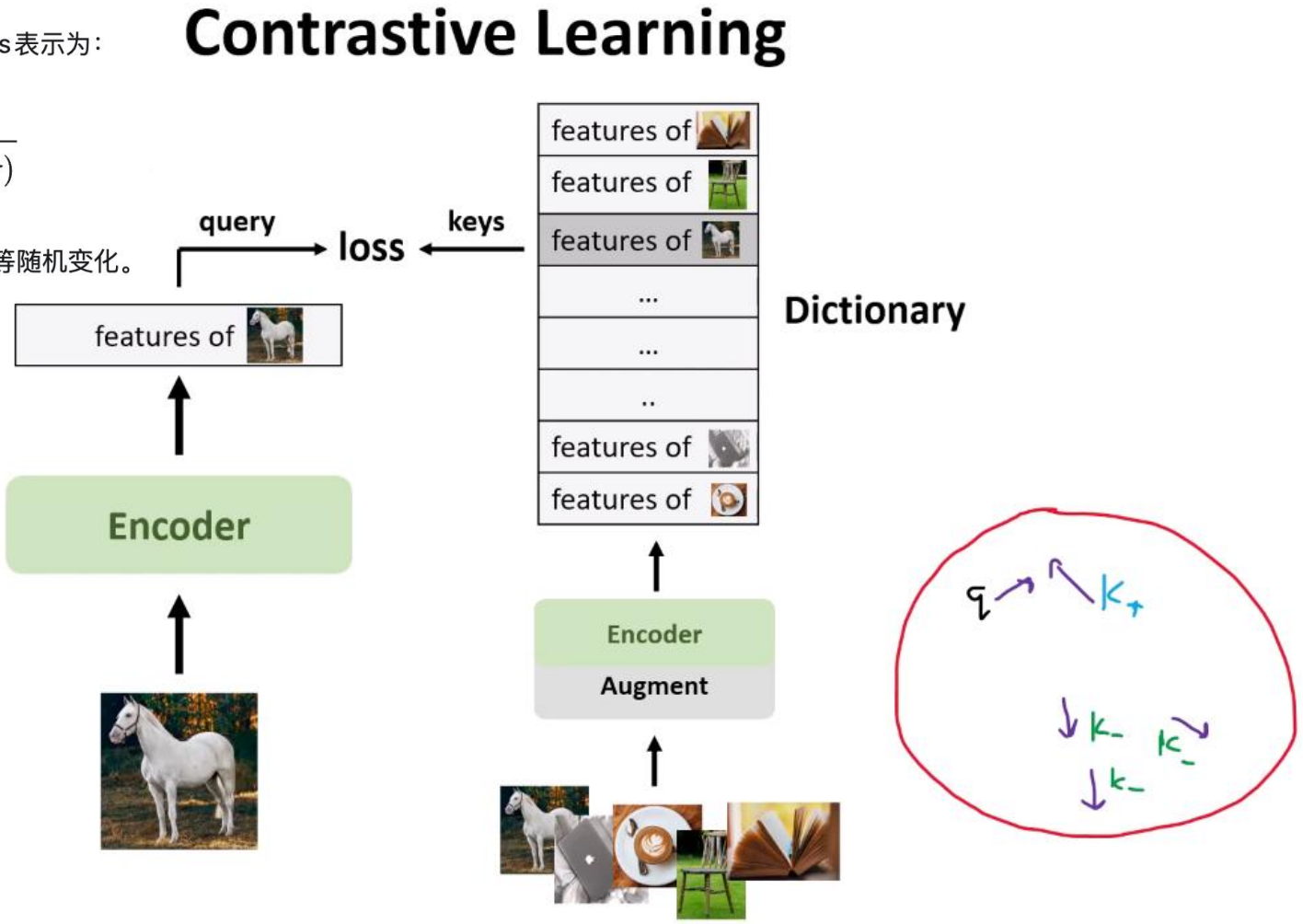
$$P(X,X^+)=\frac{\exp(f(X^+)^T f(X_i))}{\sum_p \exp(f(X_j)^T f(X_i))}$$

对比损失

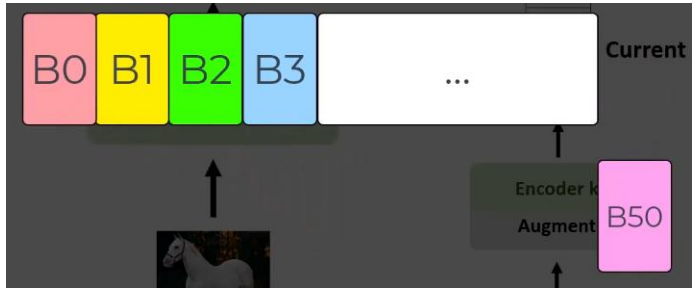
假设其中有一个正样本 其余均是负样本，则根据 InfoNCE Loss表示为：

$$\mathcal{L}_q = -log\frac{exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} exp(q \cdot k_i/\tau)}$$

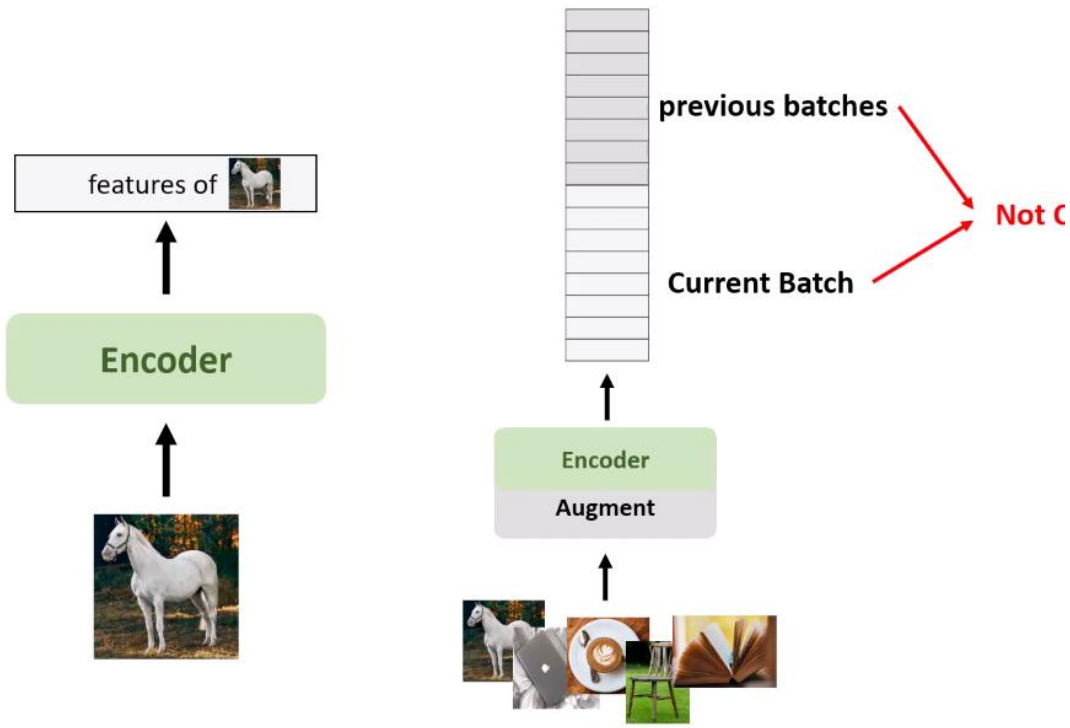
其中 q 和 k^+ 可以有多种构造方式，比如对图像进行裁剪变色等随机变化。

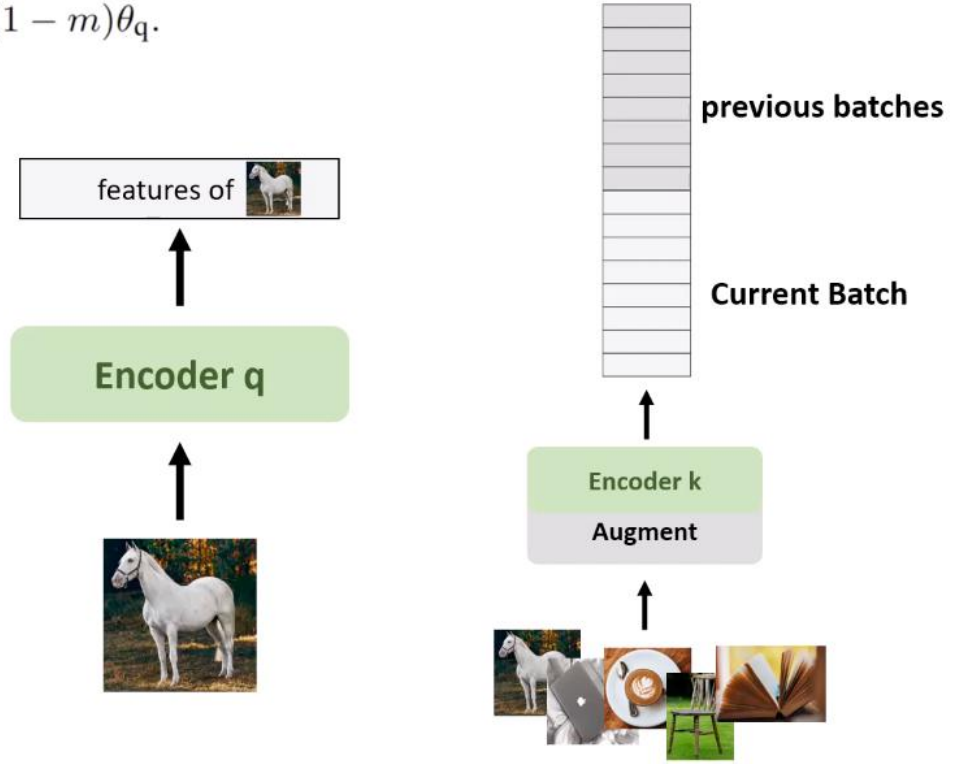


Momentum 更新

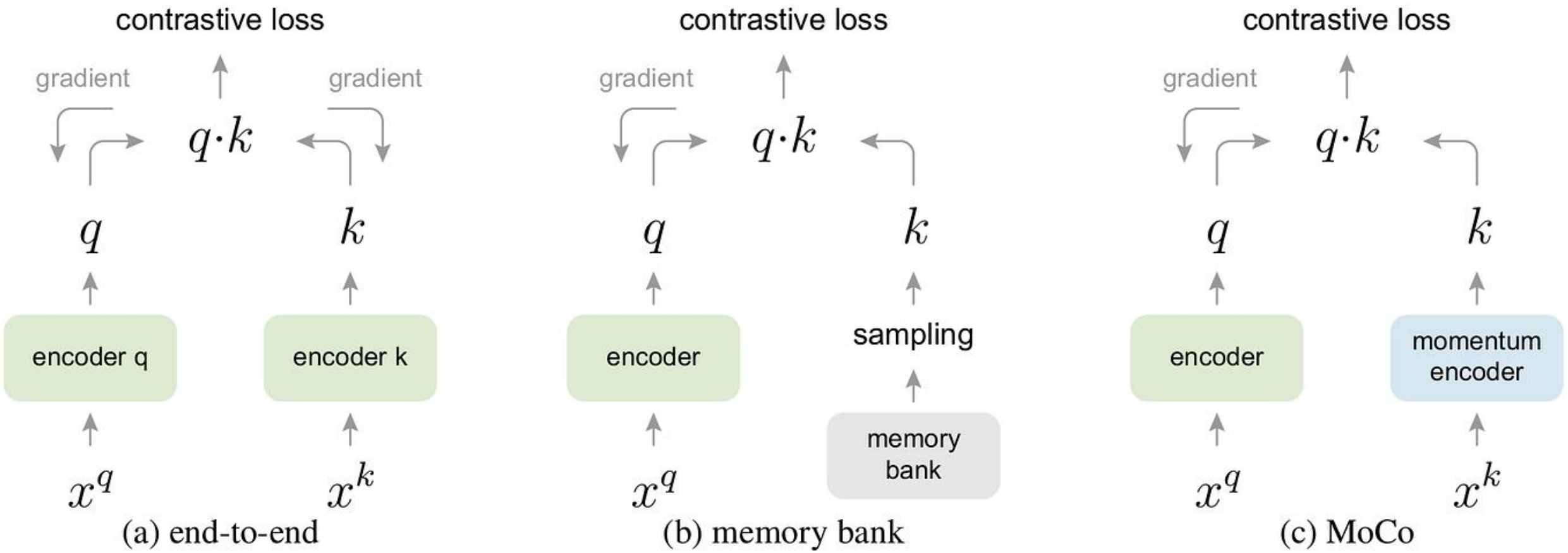


$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}.$$





Momentum 更新


contrastive loss
gradient
gradient
$q \cdot k$
q
k
encoder q
encoder k
x^q
x^k
(a) end-to-end
contrastive loss
gradient
$q \cdot k$
q
k
encoder
sampling
memory bank
x^q
(b) memory bank
contrastive loss
gradient
$q \cdot k$
q
k
encoder
momentum encoder
x^q
x^k
(c) MoCo

从Memory Bank到MOCO

由于对比学习的特性，参与对比学习损失的实例数往往越多越好，但Memory Bank中存储的都是 encoder 编码的特征，容量很大，导致采样的特征具有不一致性（是由不同的encoder产生的）。

所以，对所有参与过momentum encoder的实例建立动态字典(dynamic dictionary)作为Memory Bank，在之后训练过程中每一个batch会淘汰掉字典中最早被编码的数据。

MOCO：Momentum 更新

在参数更新阶段，MOCO只会对encoder中的参数进行更新。

由于Memory Bank，导致引入大量实例的同时，会使反向传播十分困难，而momentum encoder参数更新就依赖于Momentum 更新法，使momentum encoder的参数逐步向encoder参数逼近：

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

其中 $m = 0.999$ ， θ_q 指encoder部分的参数。

下图形式化的表示了三种结构，end-to-end，memory-bank和MoCo的区别。MoCo的特点是：

（1）用于负采样的队列是动态的

（2）用于负样本特征提取的编码器与用于query提取的编码器不一致，是一种Momentum更新的关系。

（3）与Memory Bank类似，NCE Loss只影响 Query ，不更新key。

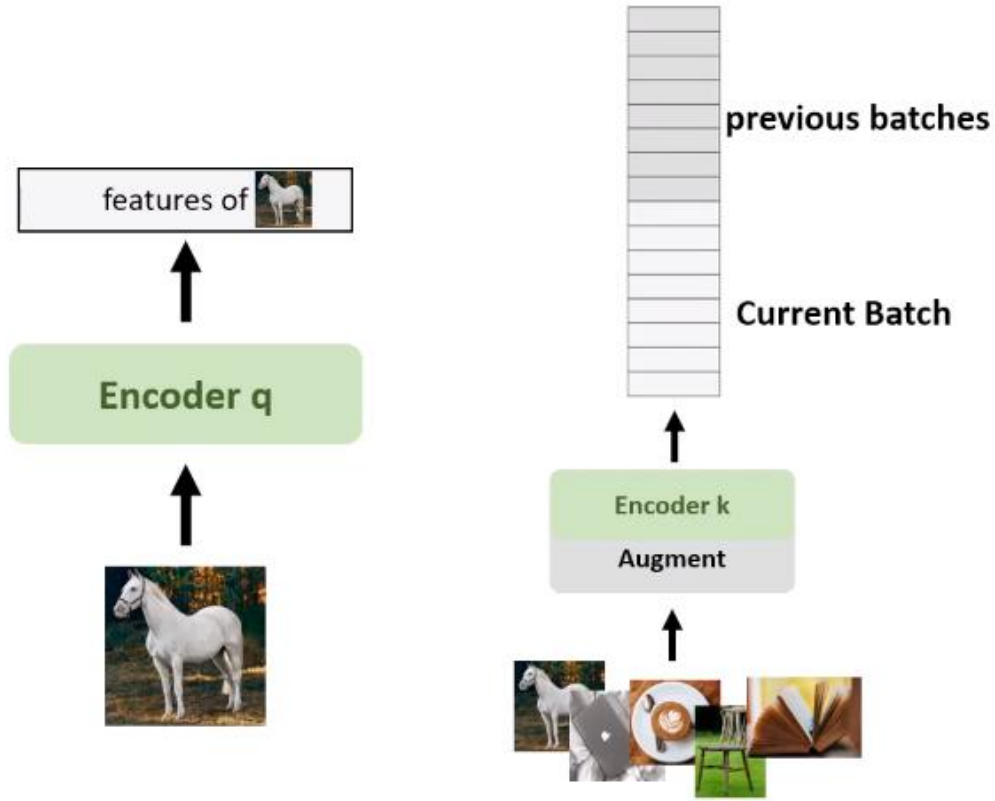
Momentum Contrast (MoCo)

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

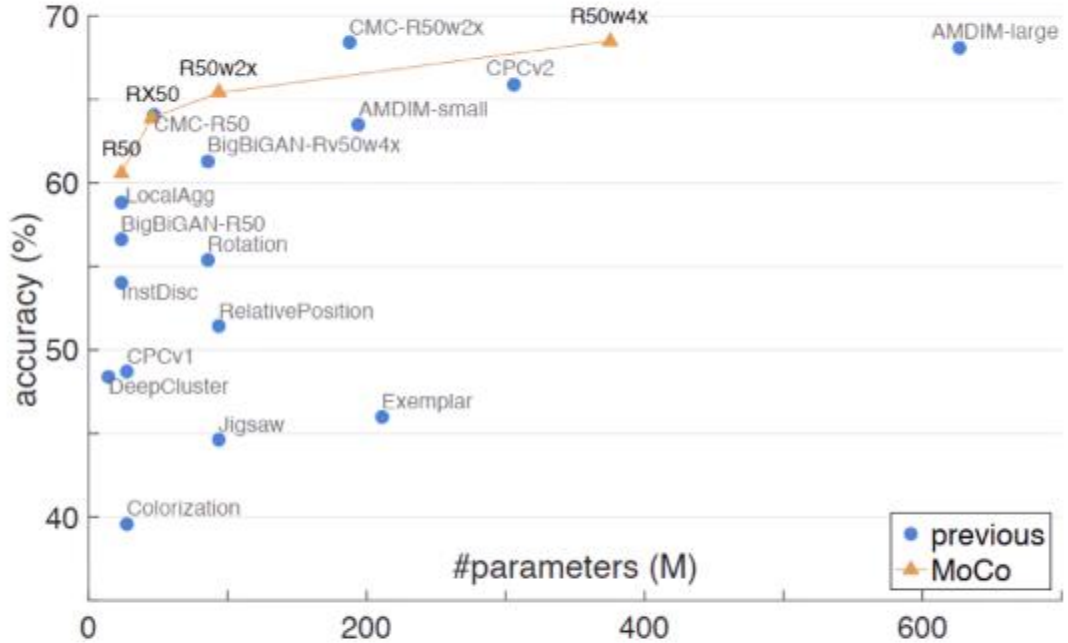
```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}.$$



method	architecture	#params (M)	accuracy (%)
Exemplar [17]	R50w3×	211	46.0 [38]
RelativePosition [13]	R50w2×	94	51.4 [38]
Jigsaw [45]	R50w2×	94	44.6 [38]
Rotation [19]	Rv50w4×	86	55.4 [38]
Colorization [64]	R101*	28	39.6 [14]
DeepCluster [3]	VGG [53]	15	48.4 [4]
BigBiGAN [16]	R50	24	56.6
	Rv50w4×	86	61.3
methods based on contrastive learning follow:			
InstDisc [61]	R50	24	54.0
LocalAgg [66]	R50	24	58.8
CPC v1 [46]	R101*	28	48.7
CPC v2 [35]	$R170^{*}_{wider}$	303	65.9
CMC [56]	$R50_{L+ab}$	47	64.1†
	$R50w2\times_{L+ab}$	188	68.4†
AMDIM [2]	$AMDIM_{small}$	194	63.5†
	$AMDIM_{large}$	626	68.1†
MoCo	R50	24	60.6
	RX50	46	63.9
	R50w2×	94	65.4
	R50w4×	375	**68.6**



MoCo V2

SimCLR

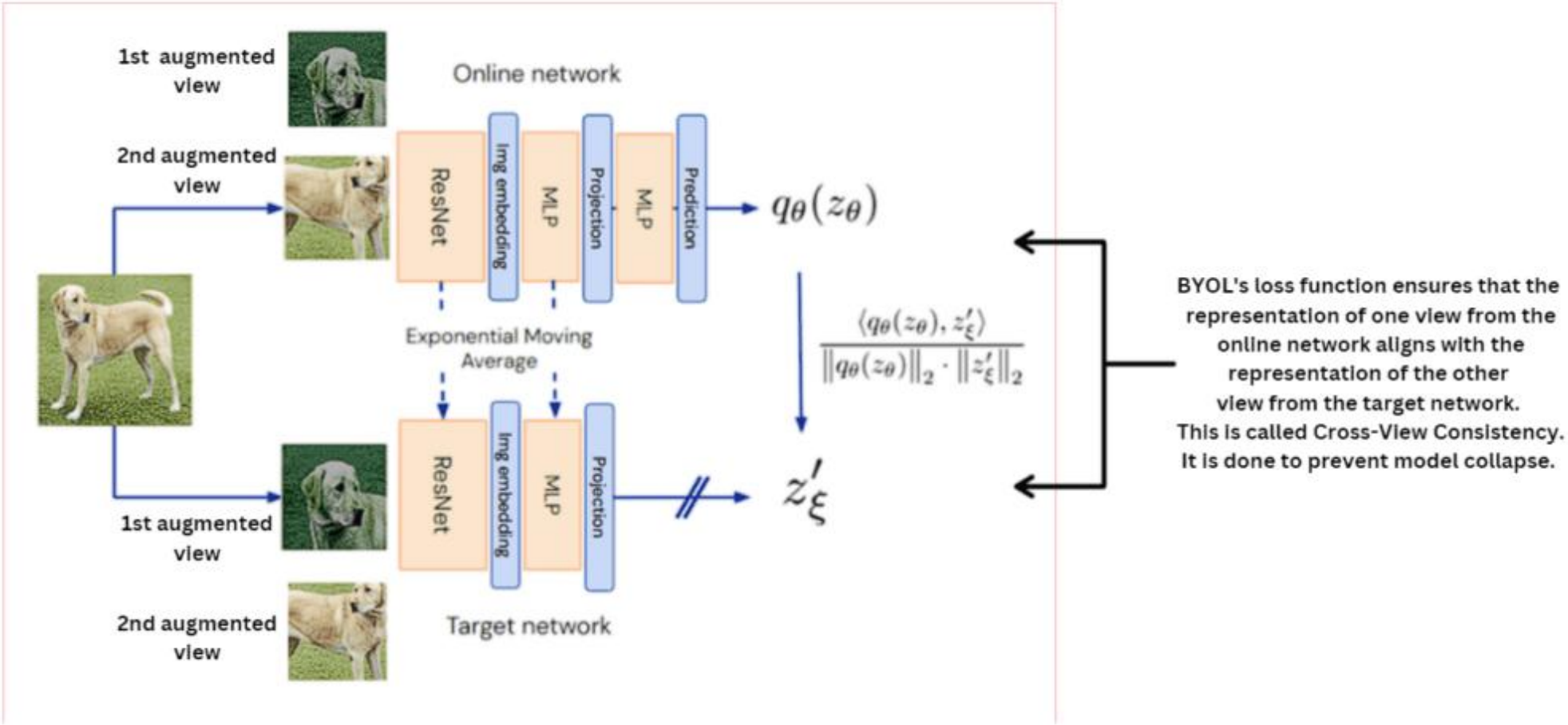
Using MLP head instead of FC layer.

Stronger data augmentations

case	unsup. pre-train MLP	aug+	cos	epochs	ImageNet acc.	VOC detection AP_{50}	AP	AP_{75}
supervised					76.5	81.3	53.5	58.8
MoCo v1				200	60.6	81.5	55.9	62.6
(a)	✓			200	66.2	82.0	56.4	62.6
(b)		✓		200	63.4	82.2	56.8	63.2
(c)	✓	✓		200	67.3	**82.5**	57.2	63.9
(d)	✓	✓	✓	200	67.5	82.4	57.0	63.6
(e)	✓	✓	✓	**800**	**71.1**	**82.5**	**57.4**	**64.0**

Table 1. **Ablation of MoCo baselines**, evaluated by ResNet-50 for (i) ImageNet linear classification, and (ii) fine-tuning VOC object detection (mean of 5 trials). "**MLP**": with an MLP head; "**aug+**": with extra blur augmentation; "**cos**": cosine learning rate schedule.

BYOL: a student-teacher framework

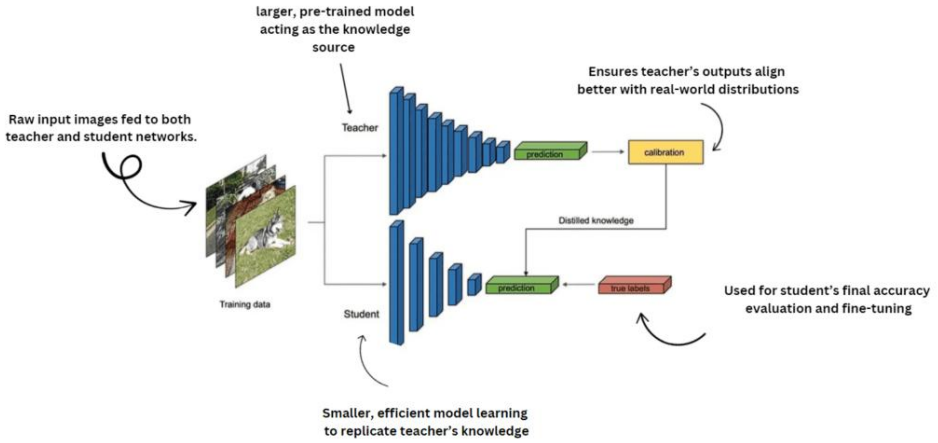


Knowledge Distillation

- The student-teacher model became prominent with the work of Geoffrey Hinton et al. in their paper on Knowledge Distillation:

- The Teacher network is a large, pre-trained model. It provides stable target representations from which the student network can learn. The network evolves slowly based on the student's parameters, ensuring stability and avoiding representation collapse.

- The student network learns by mimicking the teacher's output (soft predictions). It learns to predict the embeddings generated by the teacher.



BYOL vs SimCLR

- BYOL is a true student-teacher framework where the momentum encoder acts as the teacher, providing stable targets for the student to learn from.
- Non-reliance on negative pairs has made BYOL more robust to the choice of image augmentations.

- Comparison:
  - SimCLR authors believed that contrastive loss could focus on aligning representations based on color information alone rather than learning meaningful features (textures and shapes).

  - However, BYOL is still more robust because even if image augmentations share the same color histogram, it is still incentivized to retain additional features in its representation.

BYOL

- BYOL uses two networks – Online (student) and Target (teacher) which interact and learn from each other.

- The learning loop of BYOL is as follows –
  - The student network receives an input (e.g., an augmented view of an image) and learns to predict the target output produced by the teacher network, which processes another augmented view of the same input.
  - By repeatedly minimizing the difference between the student's predictions and the teacher's targets, the student improves its latent representations.

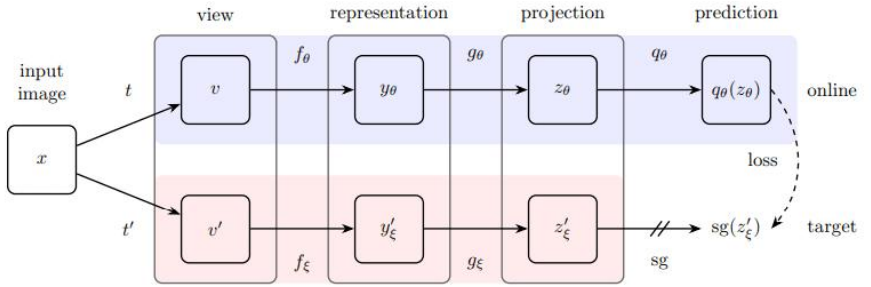


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

Flows in two networks

- **Flow in the Online Network –**

Input $\boldsymbol{v}$ -> Encoder (f_θ) -> Representation (y_θ) -> Projection Head (g_θ) -> Projection (z_θ) -> Prediction Head (q_θ) -> Prediction ($q_\theta(z_\theta)$)

- **Flow in the Target Network –**

Input $\boldsymbol{v}$ -> Encoder (f_ξ) -> Representation (y'_ξ) -> Projection Head (g_ξ) -> Target Projection (z'_ξ)

- **The target network's weights (ξ) are exponentially moving averages (EMA) of the online network's weights (θ):**

$$\xi \leftarrow \tau \cdot \xi + (1-\tau) \cdot \theta$$

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

Key Points

- Momentum coefficient τcontrols how strongly the teacher network weights depend on the online network. Higher Momentum τ= 1.0 corresponds to the target network changing slowly and retaining past information longer while Lower Momentum τ= 0.9 helps the target network adapt more quickly to the online network.
- EMA (Exponential Moving Average) is a mathematical technique used to compute a weighted average of a quantity (here referring to online networks's parameters) over time, giving more weight to recent values while gradually discounting older ones.
- No back-propagation is involved for parametric updates. Removal of gradients is done so that weights evolve more smoothly and consistently.
- Stop-Gradient (sg)
- The online network's gradient flow is stopped at the target network to ensure it remains stable and does not get updated during backpropagation. Without stop-gradient, the online network could attempt to "correct" the target network during training, introducing instability.

Pseudo Cods

Algorithm 1: BYOL: **B**ootstrap **Y**our **O**wn **L**atent

Inputs :

$\mathcal{D}$, $\mathcal{T}$, and $\mathcal{T}'$	set of images and distributions of transformations
θ, f_θ, g_θ, and q_θ	initial online parameters, encoder, projector, and predictor
ξ, f_ξ, g_ξ	initial target parameters, target encoder, and target projector
optimizer	optimizer, updates online parameters using the loss gradient
K and N	total number of optimization steps and batch size
$\{\tau_k\}_{k=1}^{K}$ and $\{\eta_k\}_{k=1}^{K}$	target network update schedule and learning rate schedule

for $k = 1$ **to** K **do**

  $\mathcal{B} \leftarrow \{x_i \sim \mathcal{D}\}_{i=1}^{N}$ // sample a batch of N images

  for $x_i \in \mathcal{B}$ **do**

    $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}'$ // sample image transformations

    $z_1 \leftarrow g_\theta(f_\theta(t(x_i)))$ and $z_2 \leftarrow g_\theta(f_\theta(t'(x_i)))$ // compute projections

    $z'_1 \leftarrow g_\xi(f_\xi(t'(x_i)))$ and $z'_2 \leftarrow g_\xi(f_\xi(t(x_i)))$ // compute target projections

    $l_i \leftarrow -2 \cdot \left(\frac{\langle q_\theta(z_1), z'_1 \rangle}{\|q_\theta(z_1)\|_2 \cdot \|z'_1\|_2} + \frac{\langle q_\theta(z_2), z'_2 \rangle}{\|q_\theta(z_2)\|_2 \cdot \|z'_2\|_2} \right)$ // compute the loss for x_i

  end

  $\delta\theta \leftarrow \frac{1}{N} \sum_{i=1}^{N} \partial_\theta l_i$ // compute the total loss gradient w.r.t. θ

  $\theta \leftarrow \text{optimizer}(\theta, \delta\theta, \eta_k)$ // update online parameters

  $\xi \leftarrow \tau_k \xi + (1 - \tau_k)\theta$ // update target parameters

end

Output : encoder f_θ

Comparison

A. Core Learning Approach

Aspect	SimCLR	BYOL
Method	Contrastive learning using both positive and negative pairs.	Non-contrastive learning, focusing only on positive pairs.
Negative Samples	Requires negative pairs (augmented views of different images).	No negative samples needed.

Comparison

B. Model Architecture

Aspect	SimCLR	BYOL
Networks	Single encoder network.	Two networks: Online (student) and Target (teacher).
Projection Head	Single MLP for projecting representations.	Includes both a projection and prediction head.
Target Update	Not applicable.	Target network updated via EMA of the online network.

Comparison

C. Training Requirements

Aspect	SimCLR	BYOL
Batch Size	Requires very large batch sizes (e.g., 4096).	Works with smaller batch sizes (e.g., 64).
Computational Cost	Higher due to pairwise similarity computations.	Lower due to absence of contrastive comparisons.

Comparison

D. Loss Function

Aspect	SimCLR	BYOL
Loss Type	Contrastive loss, comparing positives and negatives.	L2 distance between predictions and target representations.
Temperature	Used to scale similarity scores.	Not explicitly used.

Comparison

E. Performance and Robustness

Aspect	SimCLR	BYOL
Performance	Performs well but heavily reliant on negative samples and augmentations.	Achieves state-of-the-art performance without negatives.
Robustness	Sensitive to batch size and augmentation quality.	More robust to batch size and augmentation variations.

Graph Contrastive Learning（GCL）

Graph Contrastive Learning（图对比学习）是一类 自监督/无监督图表示学习方法，其核心思想是：

- 通过构造“正样本对”和“负样本对”，学习在表示空间中“相似的图结构/语义更近，不相似的更远”的图表示。

- GCL 的目标是学习：节点表示（node embedding）、子图表示（subgraph embedding）、整图表示（graph embedding），而不依赖人工标注标签。

为什么需要 GCL

- 图数据标注昂贵
- 社交网络、生物分子、知识图谱、推荐系统中的图:
- 标注成本高 标签稀缺或噪声大
- GCL 通过 自监督信号 学习通用表示。
- 图的结构复杂，难以设计监督任务
- 与图像/文本不同，图同时包含： 拓扑结构 节点/边属性 多尺度、层次关系

GCL 的通用框架

- 1.View Generator（视图生成）
  - 常见方式： 随机增强（GRACE / GraphCL） 结构变换（MVGRL） 编码器扰动（SimGRACE） 非对称邻域（GraphACL）
- 2.Encoder（图编码器）
  - GCN / GAT / GraphSAGE / GIN 等 本身不是 GCL 的创新重点，但影响性能与稳定性
- 3.Projection Head（可选）
  - 投影到对比空间 下游任务往往用 encoder 输出
- 4.Contrastive Objective（对比目标）
  - InfoNCE（最主流）
  - MI 最大化（DGI、MVGRL）
  - 非对称一致性（BYOL-style）