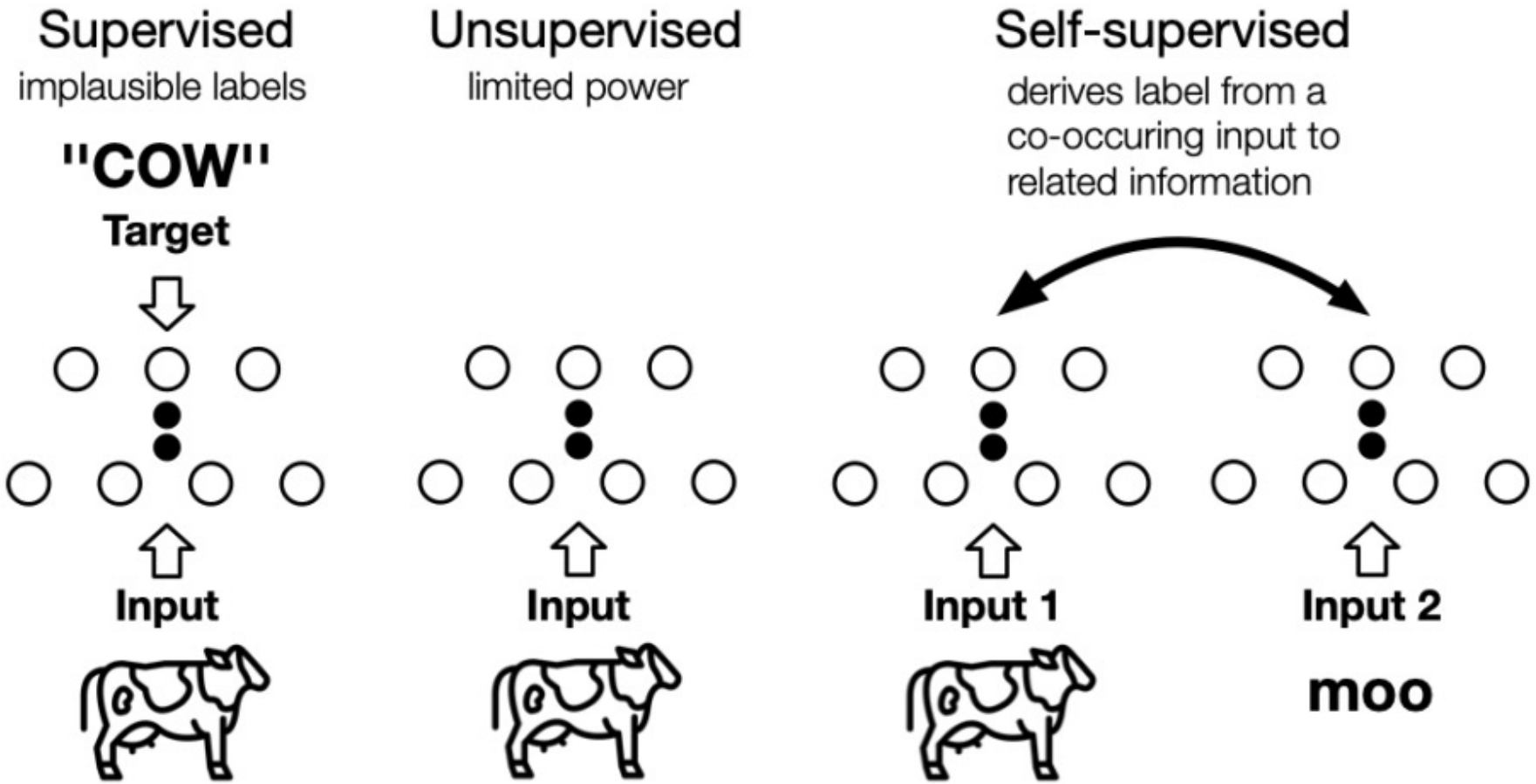
An Introduction to Self-Supervised Learning

Ming Li, Jie Li, Rongrong Ji
Media Analytics and Computing Laboratory
Xiamen University

Updated on November 15, 2021

Learning Paradigms



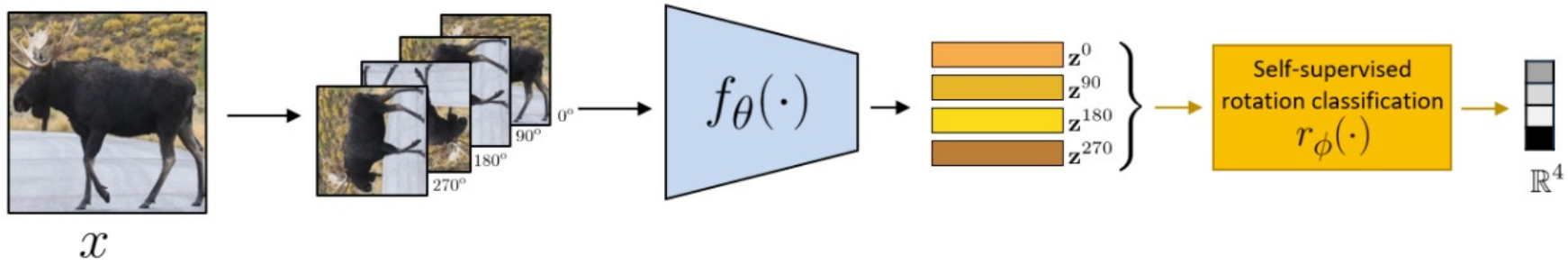
Self-supervised Learning: Generative or Contrastive

Why Self-Supervised Learning ?

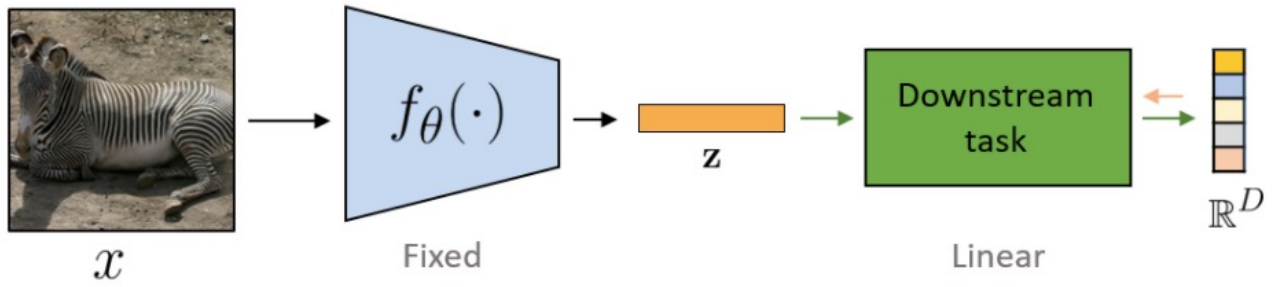
- Labeling data is time-consuming and expensive
- The model learned using supervised learning is highly task-related and may not be well transferred to other tasks
- The value and information of the data itself is much higher than the labels

Self-Supervised Learning Pipeline

Stage 1: Train network on pretext task (without human labels)

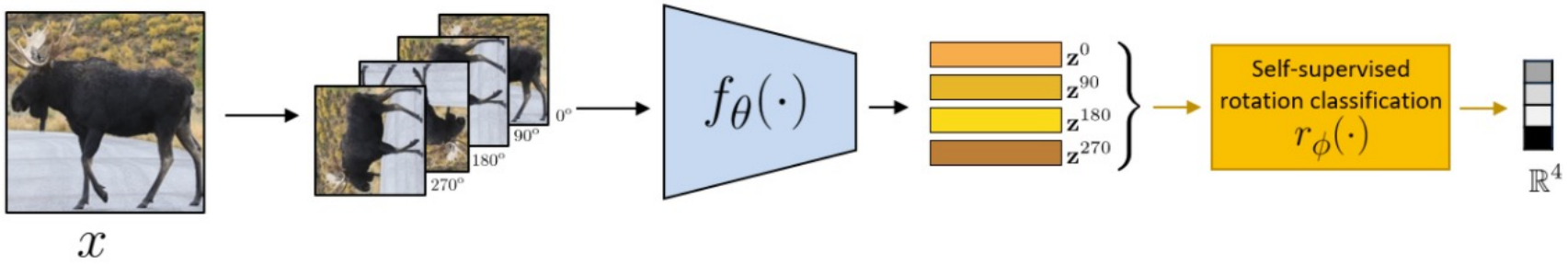


Stage 2: Train classifier on learned features for new task with fewer labels

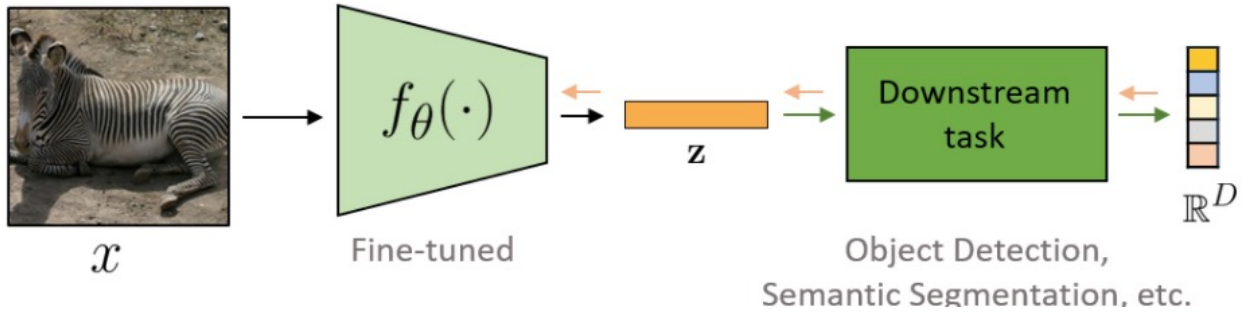


Self-Supervised Learning Pipeline

Stage 1: Train network on pretext task (without human labels)

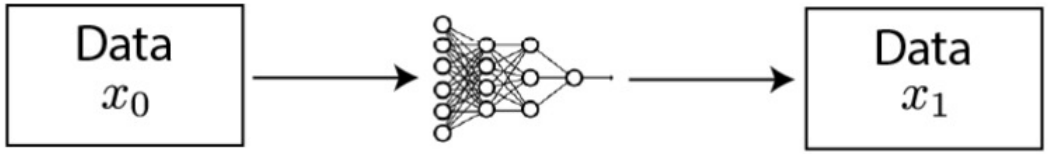


Stage 2: Fine-tune network for new task with fewer labels



Generative SSL and Contrastive SSL

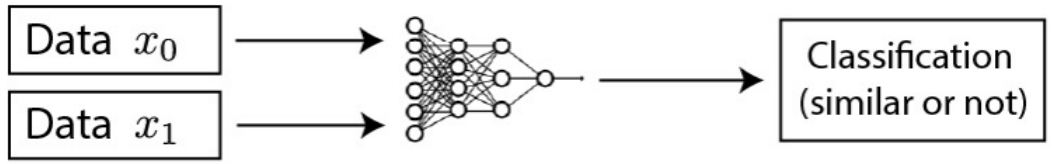
Generative / Predictive



Loss measured in the output space

Examples: Colorization, Auto-Encoders

Contrastive

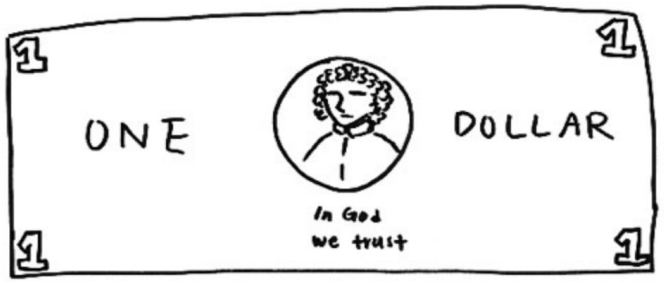


Loss measured in the representation space

Examples: TCN, CPC, Deep-InfoMax

Contrastive Self-Supervised Learning | Ankesh Anand's Blog

What Makes Good for Contrastive SSL?



- A good representation algorithm does not necessarily need to pay attention to every detail of the sample (such as generative SSL), as long as the learned features can distinguish it from other samples.

- For example, in the above pictures, we only need to draw the picture on the left to know that it is a dollar, instead of drawing a dollar completely like the picture on the right.

How Do Contrastive Methods Work?

More formally, for any data point x, contrastive methods aim to learn an encoder f such that:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

- here x^+ is data point similar or congruent to x, referred to as a *positive* sample.
- x^- is a data point dissimilar to x, referred to as a *negative* sample.
- the score function is a metric that measures the similarity between two features.

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{\exp\left(f(x)^T f\left(x^+\right)\right)}{\exp\left(f(x)^T f\left(x^+\right)\right) + \sum_{j=1}^{N-1} \exp\left(f(x)^T f\left(x_j\right)\right)} \right]$$

InfoNCE (Contrastive) Loss

Contrastive Self-Supervised Learning | Ankesh Anand's Blog

Why use InfoNCE Loss ?

- The objective of self-supervised representation learning is maximizing an estimate of the mutual information (MI) between different views of the data

- InfoNCE Loss is a lower bound of mutual information

$$I(X;Y) \geq \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{K}\sum_{j=1}^{K}e^{f(x_i,y_j)}}\right] \triangleq I_{\mathrm{NCE}}(X;Y)$$

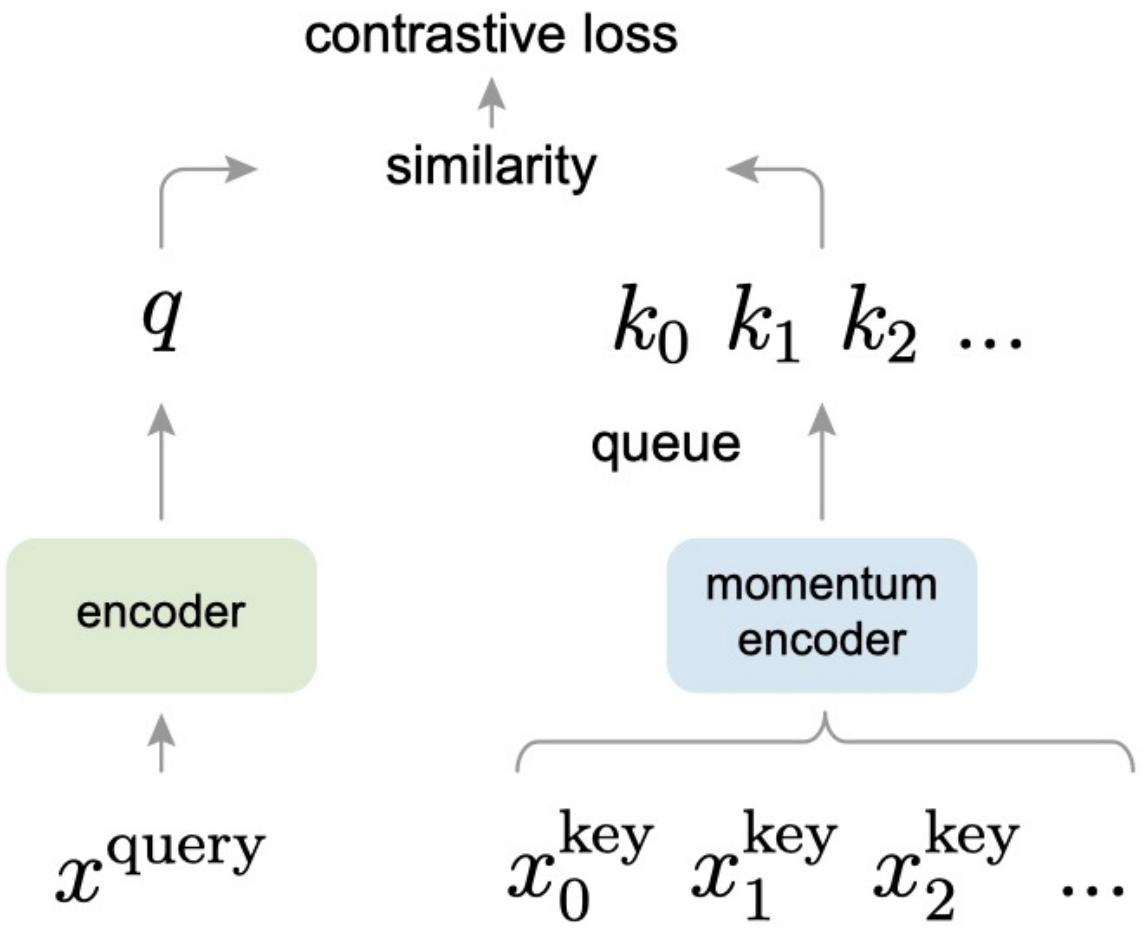
On Mutual Information Maximization for Representation Learning, ICLR 2020

Mainstream Methods

MoCo

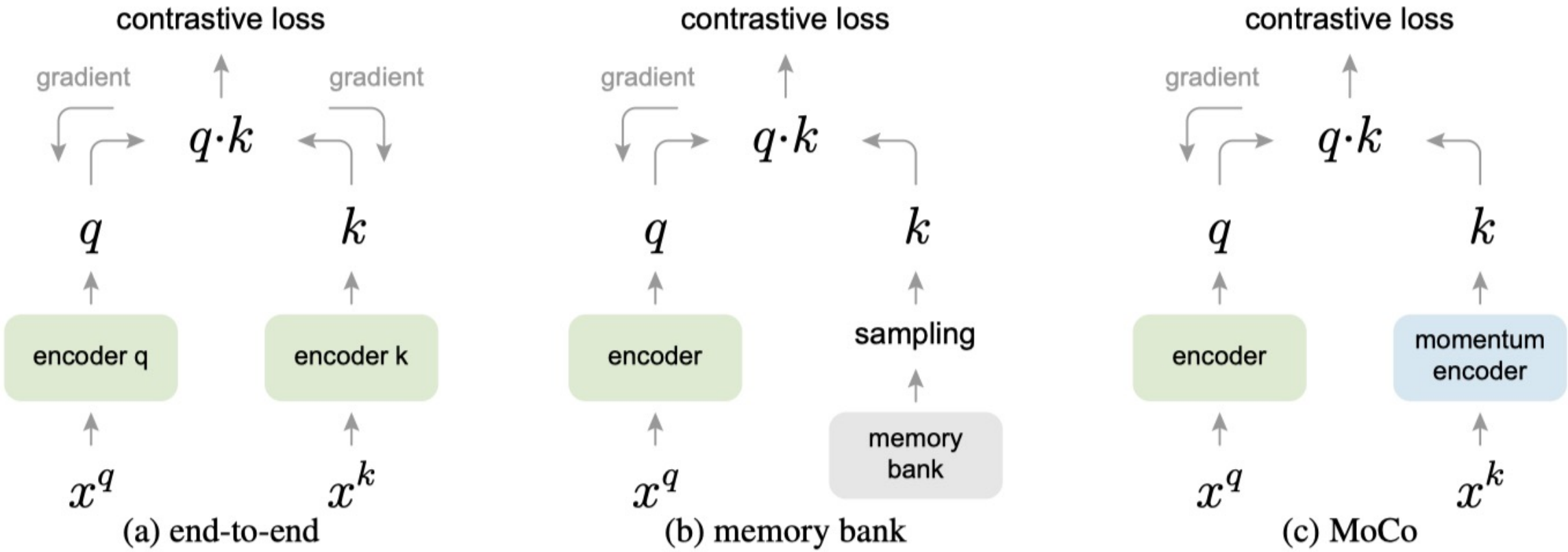
- Contrastive methods tend to work better with more number of negative samples to cover the underlying distribution
- MoCo proposes to decouple the batch size and the number of samples, and use a memory bank to store the previous features to obtain a large number of negative samples
- MoCo uses momentum encoder to maintain the consistency between the features in the memory bank and the features in the current batch

MoCo

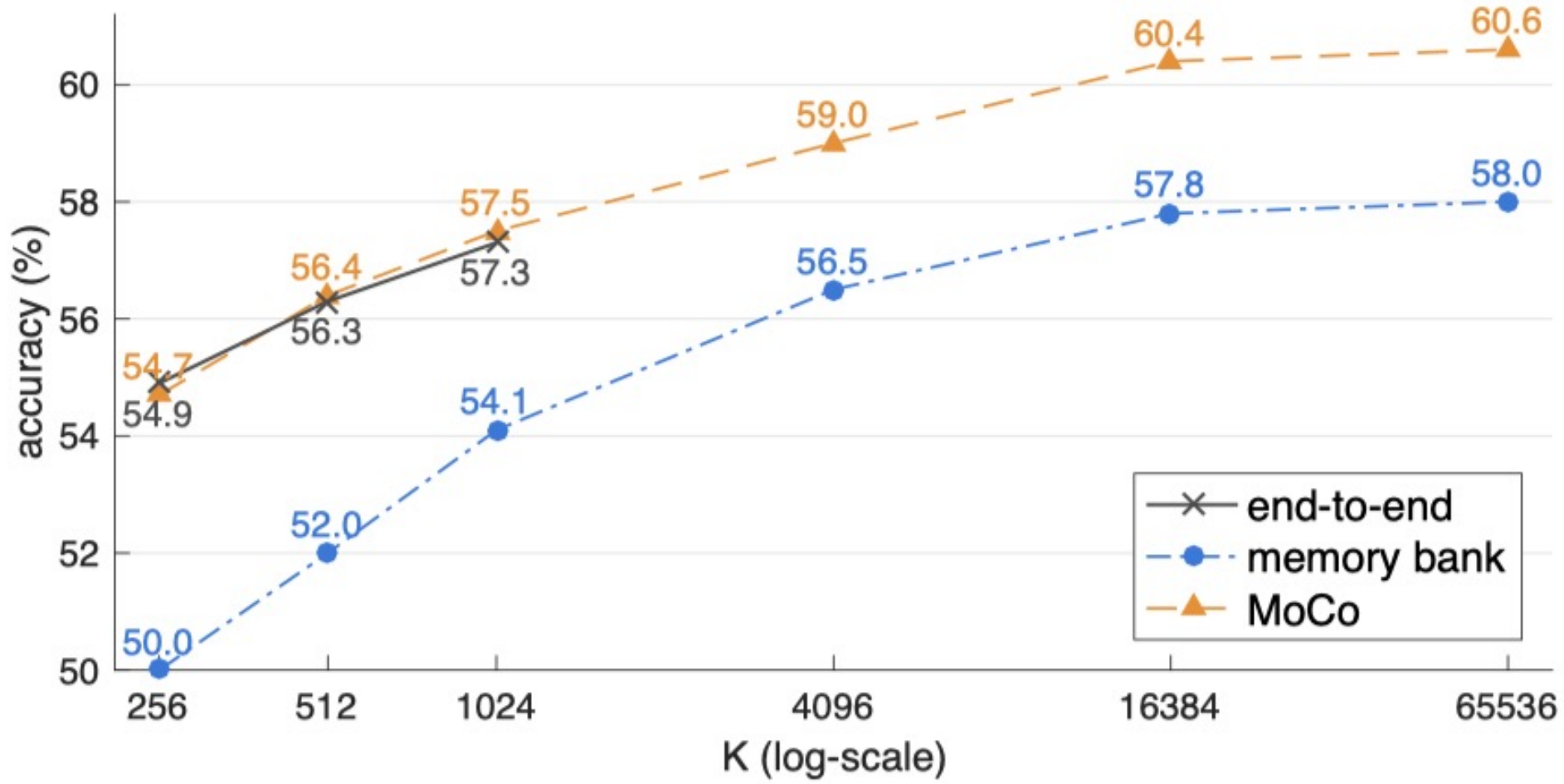


Momentum Contrast for Unsupervised Visual Representation Learning, CVPR 2020

Three Contrastive Loss Mechanisms



Comparison of Three Contrastive Loss Mechanisms



SimCLR

- Composition of data augmentations plays a critical role in defining effective predictive tasks in self-supervised learning
- A learnable nonlinear transformation between the representation and the contrastive loss substantially improves the quality of the learned representations
- Contrastive learning benefits from larger batch sizes and more training steps compared to supervised learning

Data Augmentation for Positive Pairs

(a) Original

(b) Crop and resize

(c) Crop, resize (and flip)

(d) Color distort. (drop)

(e) Color distort. (jitter)

f) Rotate $\{90°, 180°, 270°\}$

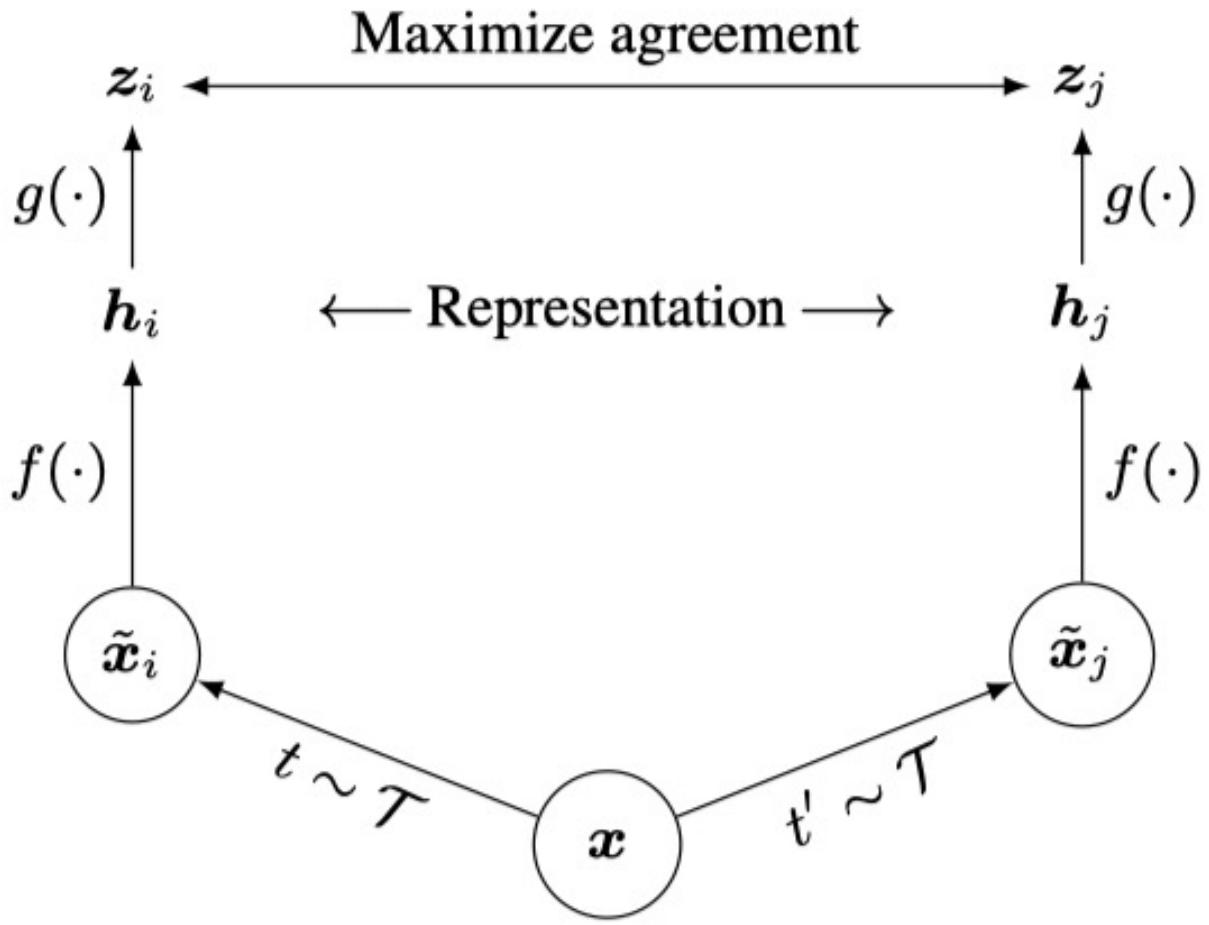
(g) Cutout

(h) Gaussian noise

(i) Gaussian blur

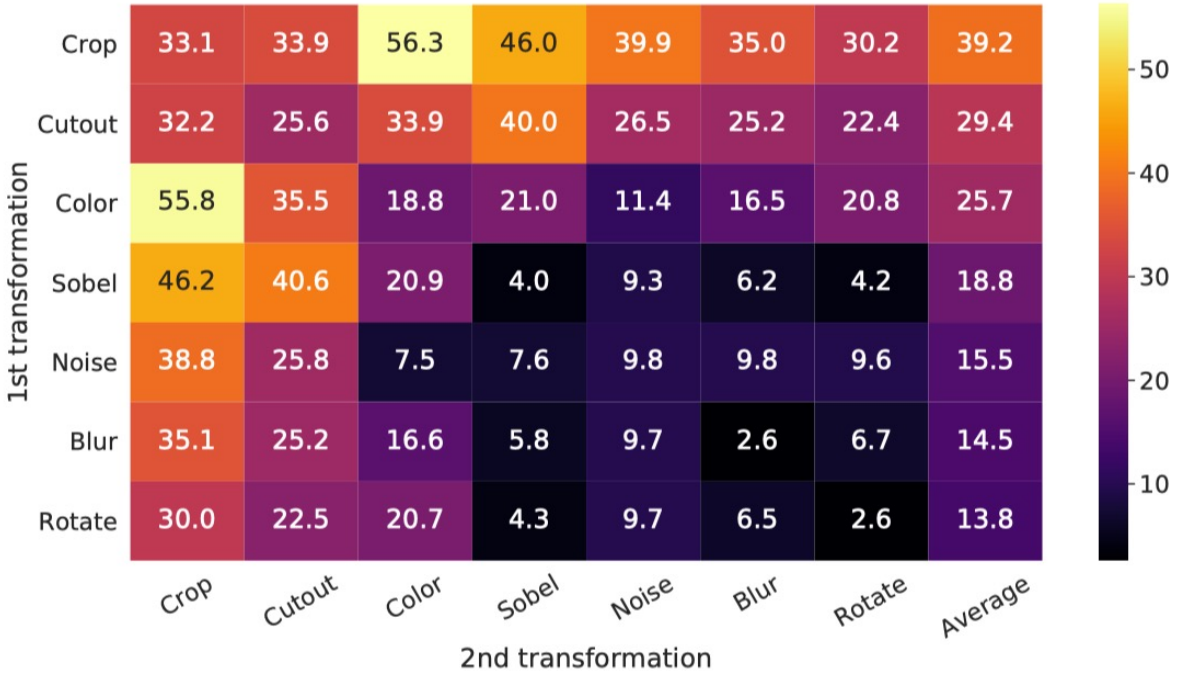
(j) Sobel filtering

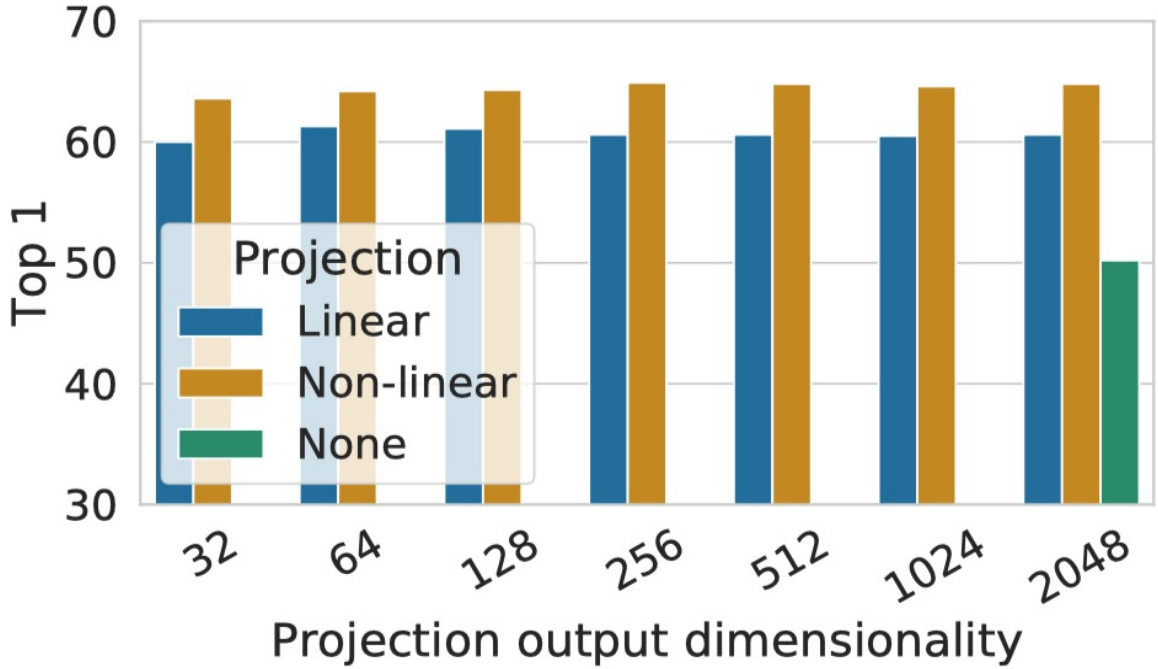
SimCLR Pipeline



A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020

Data Augmentations and Projection Heads Matter





Larger Batch Sizes and longer Epochs Are Better

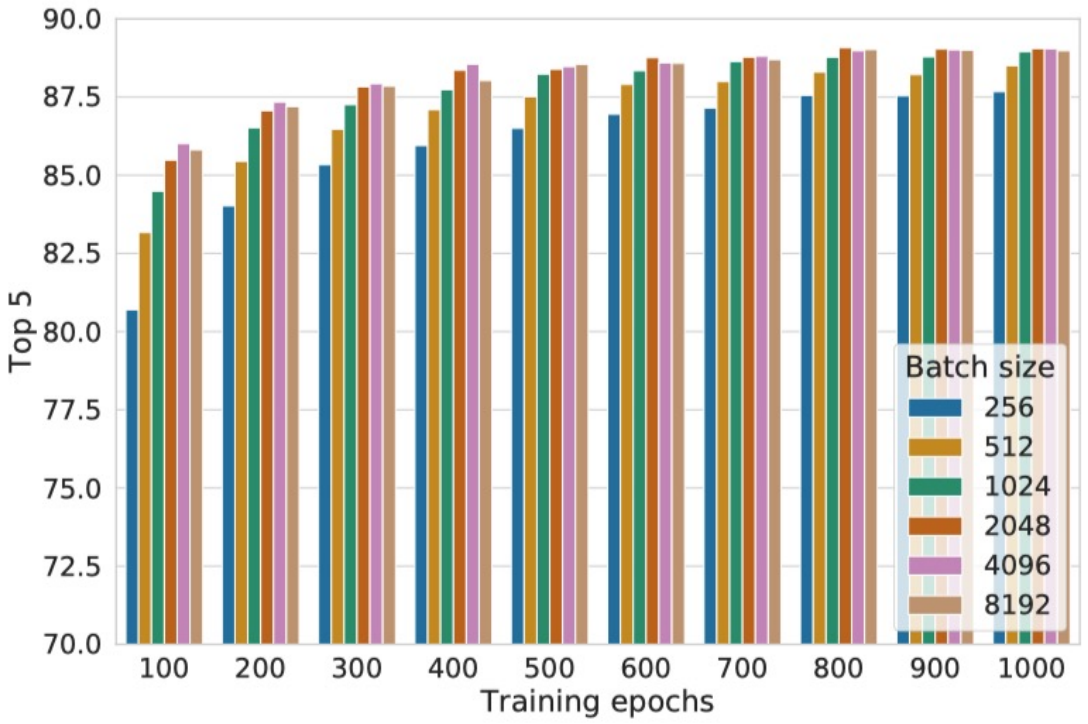


Figure B.1. Linear evaluation (top-5) of ResNet-50 trained with different batch sizes and epochs. Each bar is a single run from scratch. See Figure 9 for top-1 accuracy.

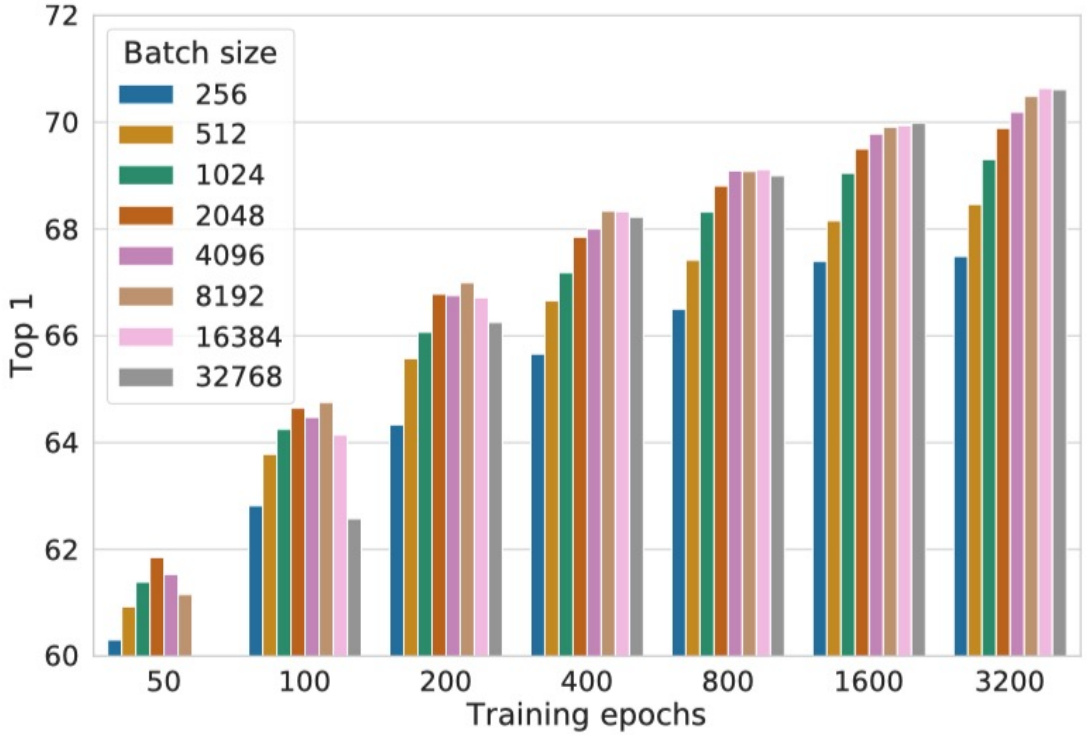
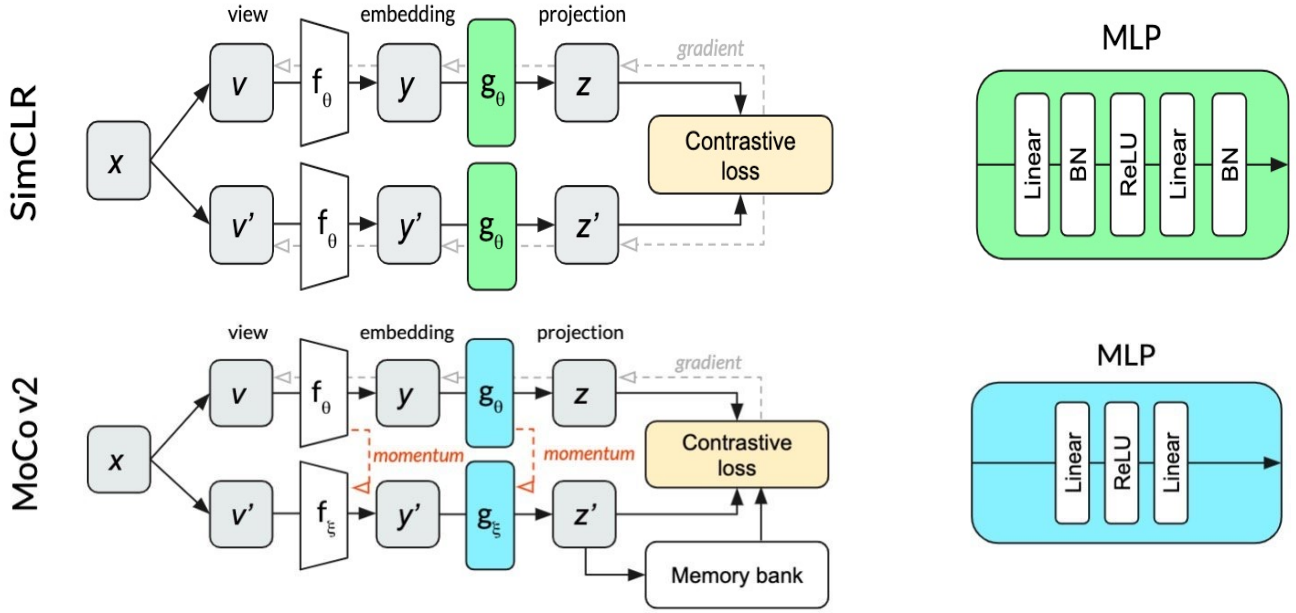


Figure B.2. Linear evaluation (top-1) of ResNet-50 trained with different batch sizes and *longer* epochs. Here a *square root* learning rate, instead of a linear one, is utilized.

MoCo v2

- Inspired by SimCLR, MoCo v2 uses a projection head, stronger data augmentation, cosine learning rate scheduler and longer training epochs to improve MoCo v1



Improved Baselines with Momentum Contrastive Learning, arXiv 2020

generally intelligent

Experiments

- Additional projection head, stronger data enhancement, cosine learning rate and longer training time can all bring improvements

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.

case	unsup. pre-train MLP	aug+	cos	epochs	batch	ImageNet acc.
MoCo v1 [6]				200	256	60.6
SimCLR [2]	✓	✓	✓	200	256	61.9
SimCLR [2]	✓	✓	✓	200	8192	66.6
MoCo v2	✓	✓	✓	200	256	**67.5**
*results of **longer** unsupervised training follow:*						
SimCLR [2]	✓	✓	✓	1000	4096	69.3
MoCo v2	✓	✓	✓	800	256	**71.1**

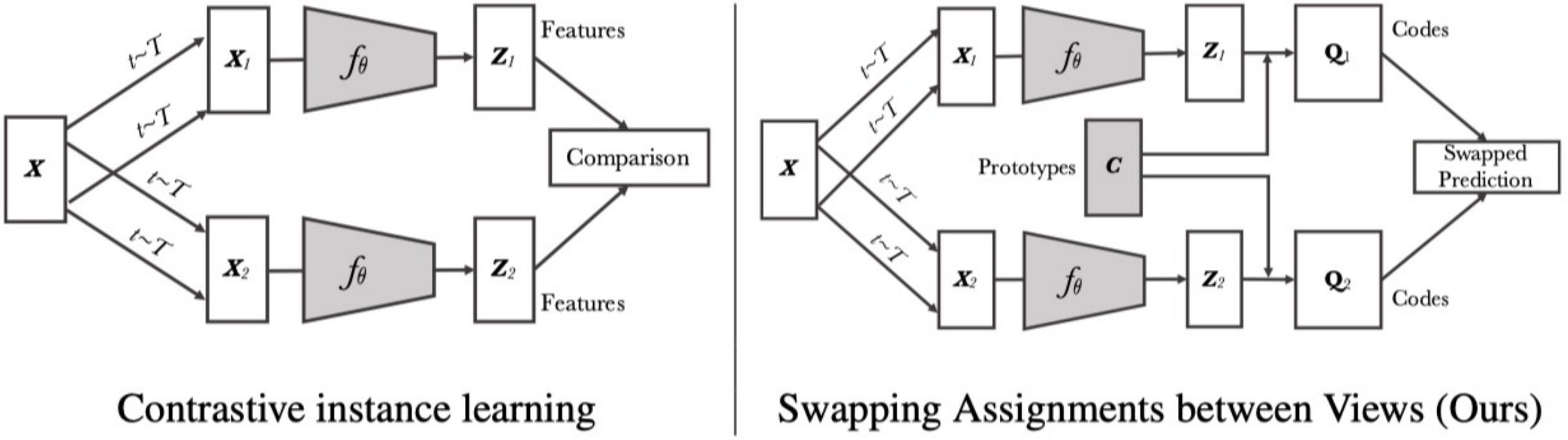
Table 2. **MoCo *vs.* SimCLR**: ImageNet linear classifier accuracy (**ResNet-50, 1-crop 224×224**), trained on features from unsupervised pre-training. "aug+" in SimCLR includes blur and stronger color distortion. SimCLR ablations are from Fig. 9 in [2] (we thank the authors for providing the numerical results).

SwAV

- SwAV clusters the data while enforcing consistency between cluster assignments produced for different augmentations of the same image, instead of comparing features directly as in contrastive learning

- Proposed multi-crop that uses smaller-sized images to increase the number of views while not increasing the memory or computational requirements during training

Contrastive instance learning vs. SwAV



Understanding SwAV

https://github.com/facebookresearch/swav

Unsupervised Learning of Visual Features by Contrasting Cluster Assignments, NeurIPS 2020

Multi-crop

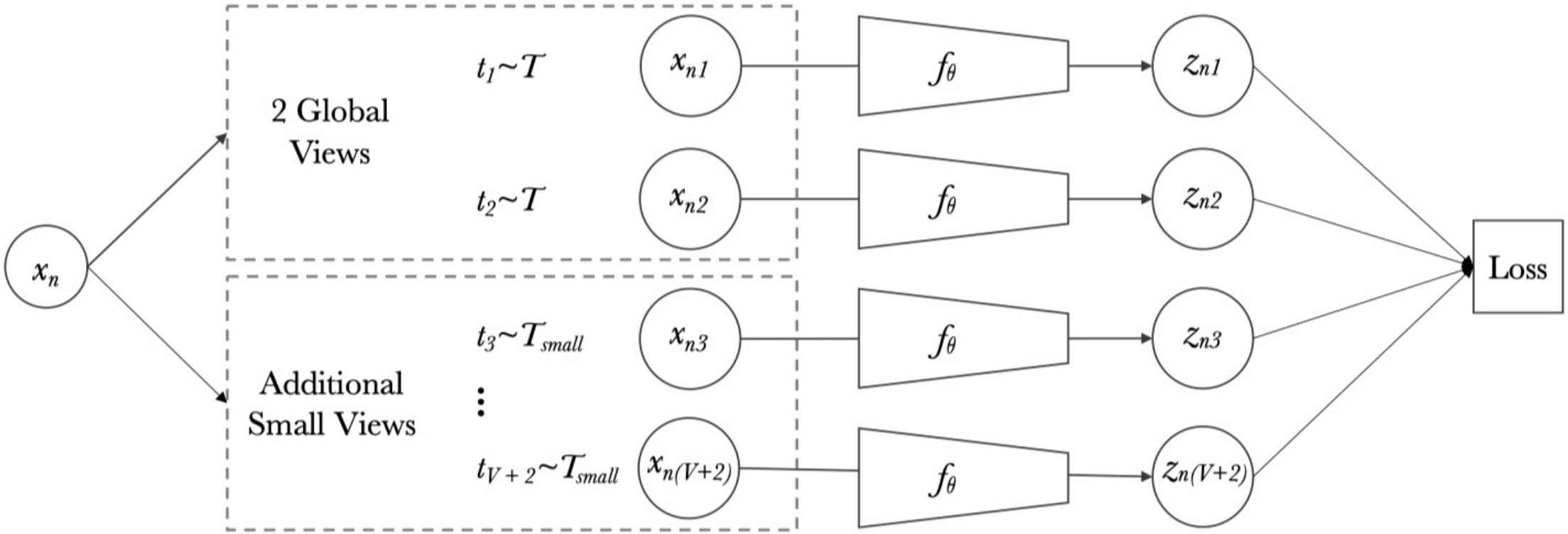


Figure 5: **Multi-crop**: the image x_n is transformed into $V + 2$ views: two global views and V small resolution zoomed views.

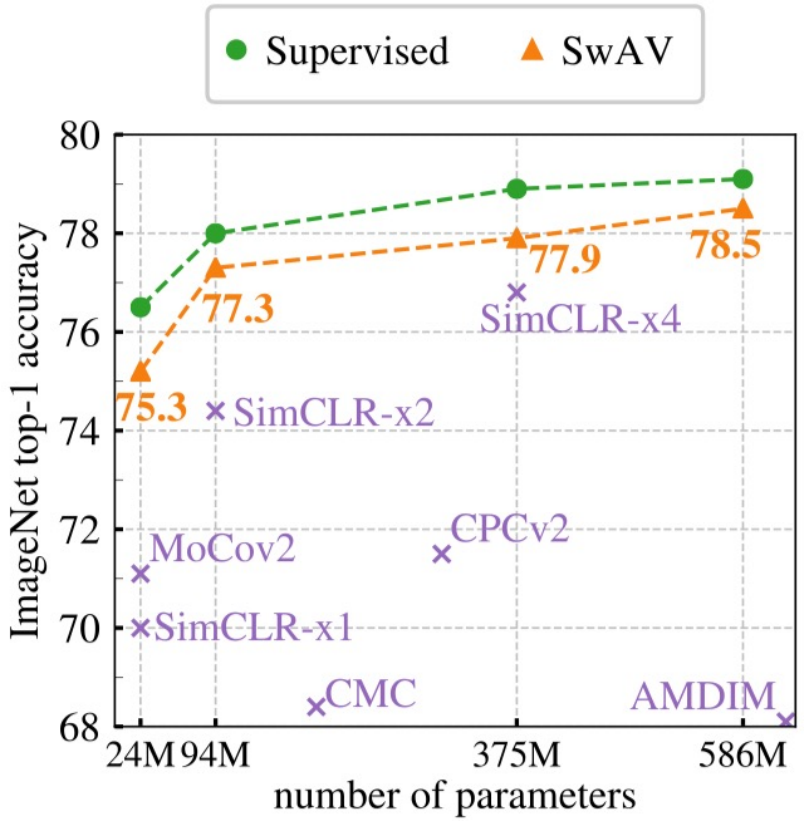
Tricks

- To avoid model collapse, i.e., all features of the current batch are clustered into a prototype, SwAV requires the instances in the current batch to be clustered into different categories evenly

- SwAV uses a mix of views with different resolutions in place of two full-resolution views to avoid the bias produced by downsized images

Experiments

Linear classification on ImageNet



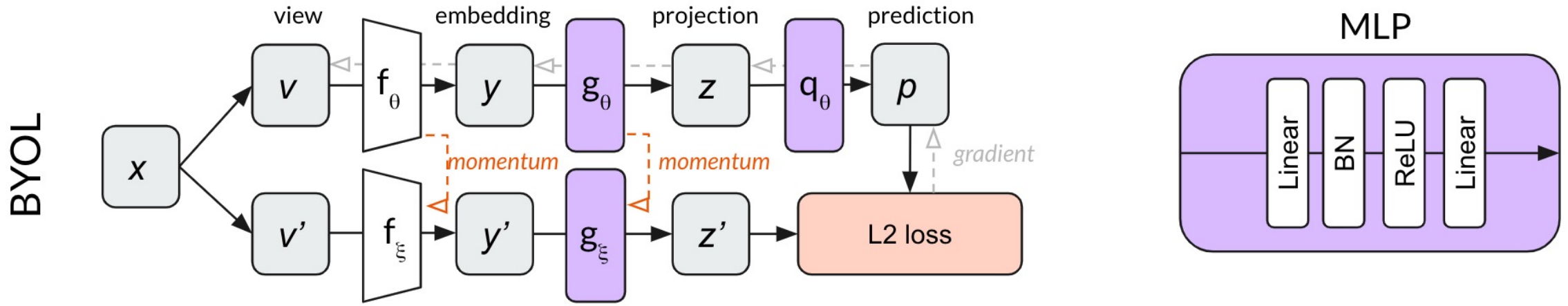
Effectiveness of Multi-crop

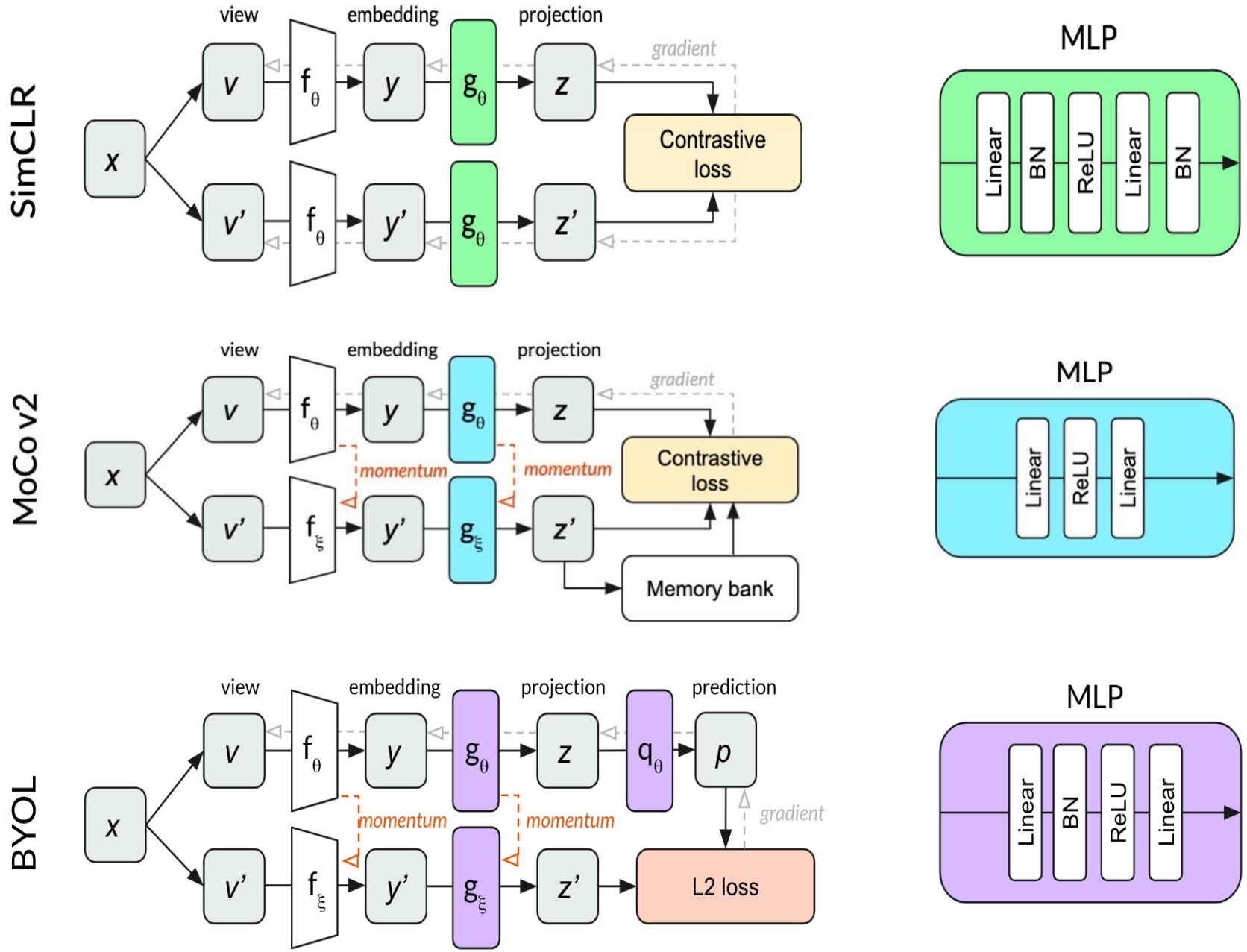
Method	Top-1 2x224	Top-1 2x160+4x96	Δ
Supervised	76.5	76.0	−0.5
Contrastive-instance approaches			
SimCLR	68.2	70.6	+2.4
Clustering-based approaches			
SeLa-v2	67.2	71.8	+4.6
DeepCluster-v2	70.2	74.3	+4.1
SwAV	70.1	74.1	+4.0

BYOL

- BYOL achieves a new state of the art without negative pairs
- BYOL uses an asymmetric network structure (an additional prediction head) and stop-gradient to avoid model collapse

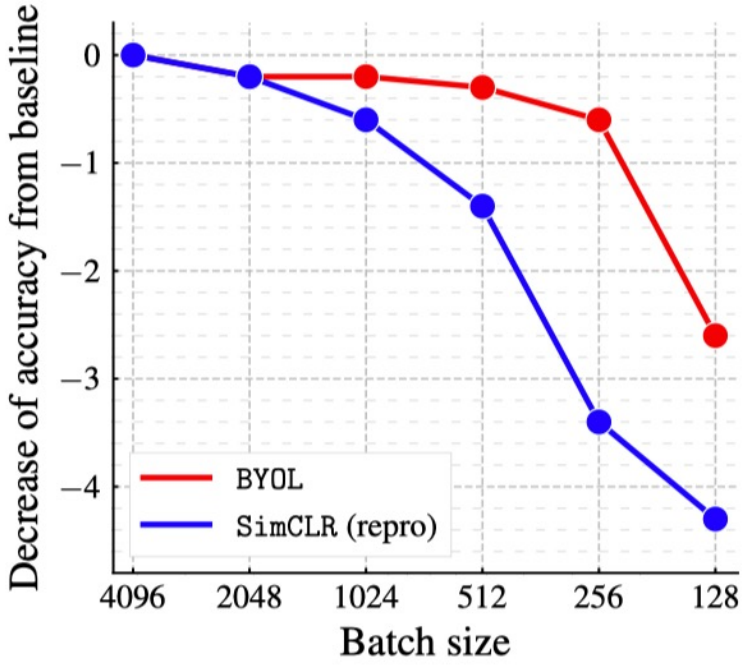
BYOL


BYOL
view
embedding
projection
prediction
x
v
v'
f_θ
f_ξ
y
y'
g_θ
g_ξ
z
z'
q_θ
p
momentum
momentum
gradient
L2 loss
MLP
Linear
BN
ReLU
Linear

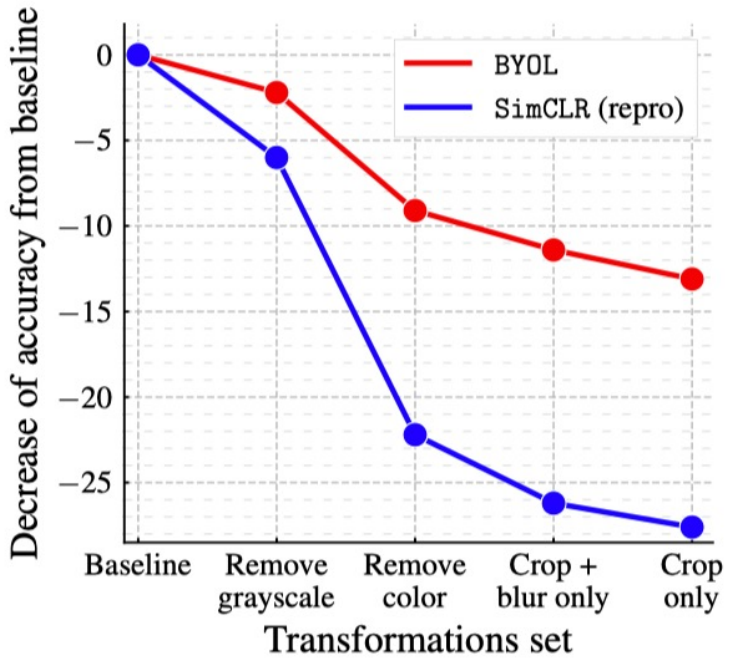

SimCLR
view
embedding
projection
gradient
x
v
v'
f_θ
y
y'
g_θ
z
z'
Contrastive loss
MLP
Linear
BN
ReLU
Linear
BN
MoCo v2
momentum
f_ξ
g_ξ
Memory bank
BYOL
prediction
q_θ
p
L2 loss

Experiments

Compared with SimCLR, BYOL is less sensitive to batch size and transformations because it does not require negative pairs



(a) Impact of batch size



(b) Impact of progressively removing transformations

Experiments

stop-gradient and predictor are both important

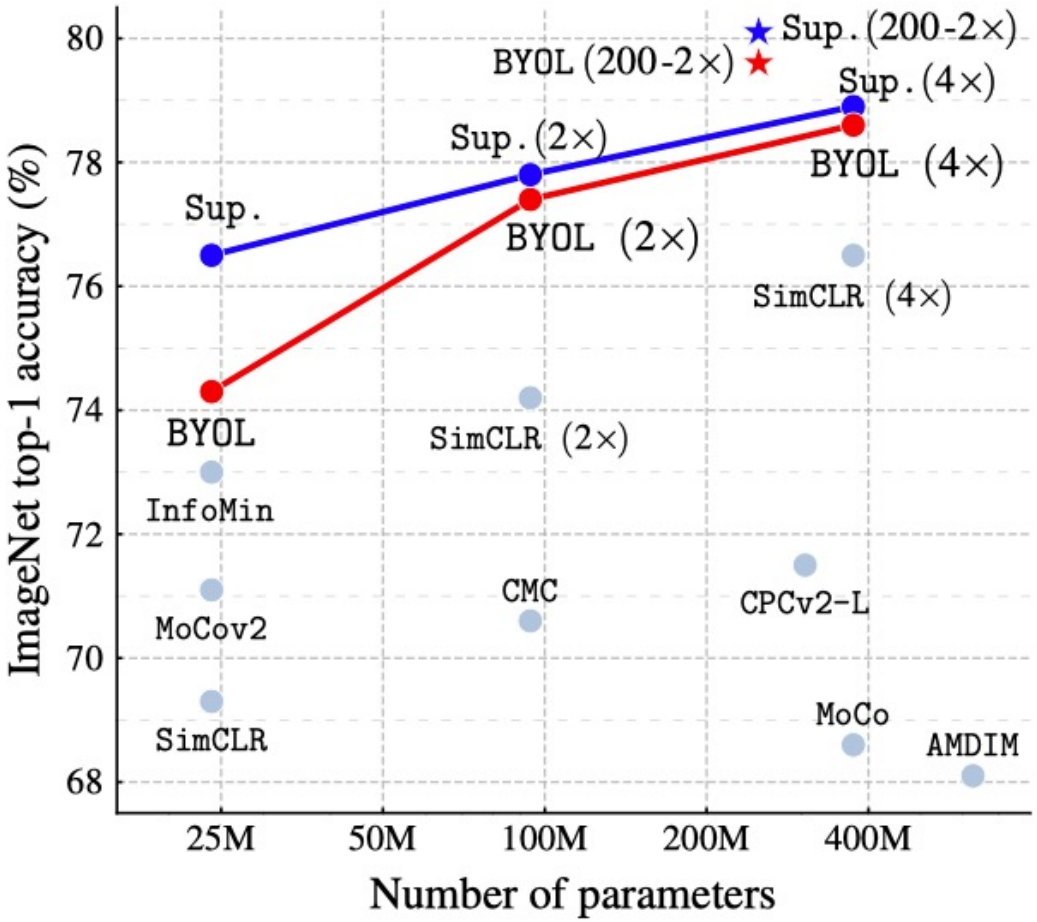
Target	τ_{base}	Top-1
Constant random network	1	$18.8_{\pm 0.7}$
Moving average of online	0.999	69.8
Moving average of online	0.99	**72.5**
Moving average of online	0.9	68.4
Stop gradient of online†	0	0.3

(a) Results for different target modes. †In the *stop gradient of online*, $\tau = \tau_{\text{base}} = 0$ is kept constant throughout training.

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

(b) Intermediate variants between BYOL and SimCLR.

Experiments



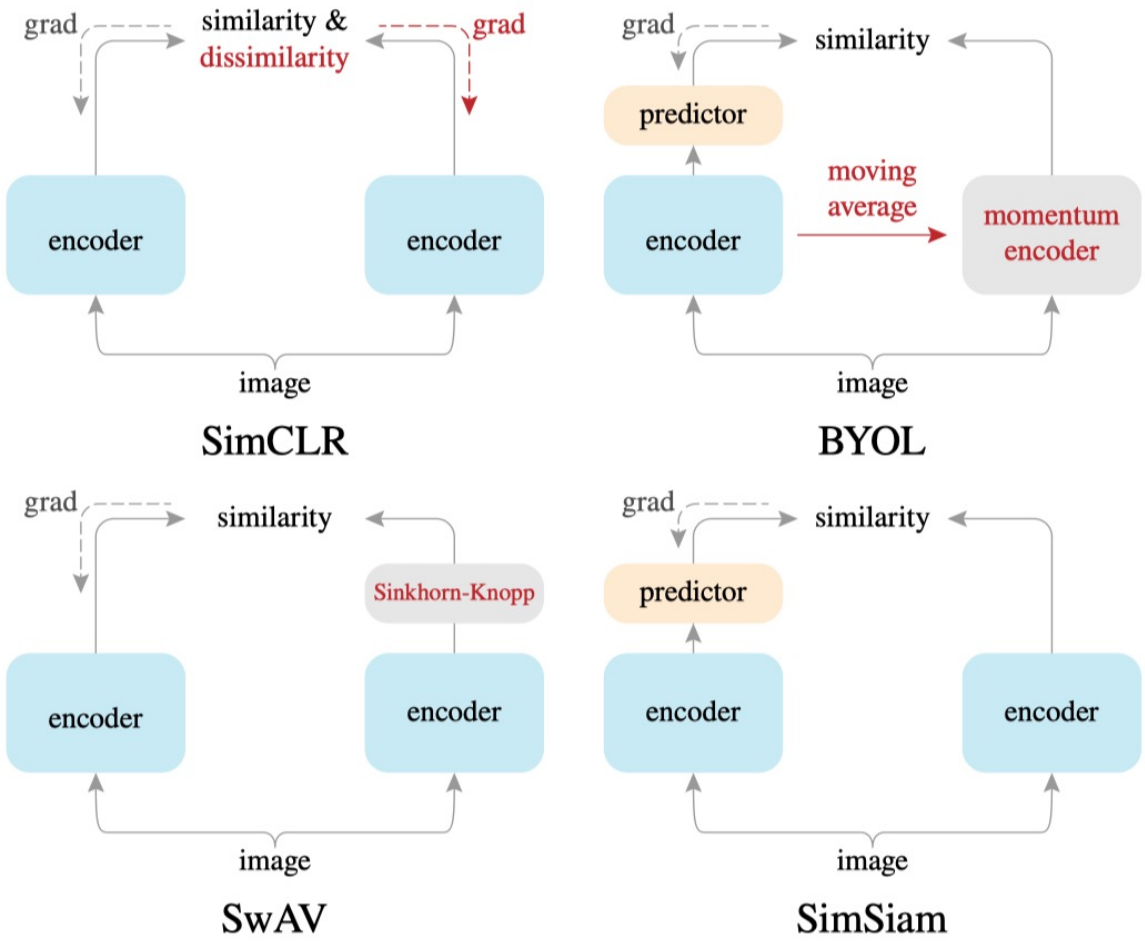
Method	Top-1	Top-5
Local Agg.	60.2	-
PIRL [35]	63.6	-
CPC v2 [32]	63.8	85.3
CMC [11]	66.2	87.0
SimCLR [8]	69.3	89.0
MoCo v2 [37]	71.1	-
InfoMin Aug. [12]	73.0	91.1
BYOL (ours)	**74.3**	**91.6**

(a) ResNet-50 encoder.

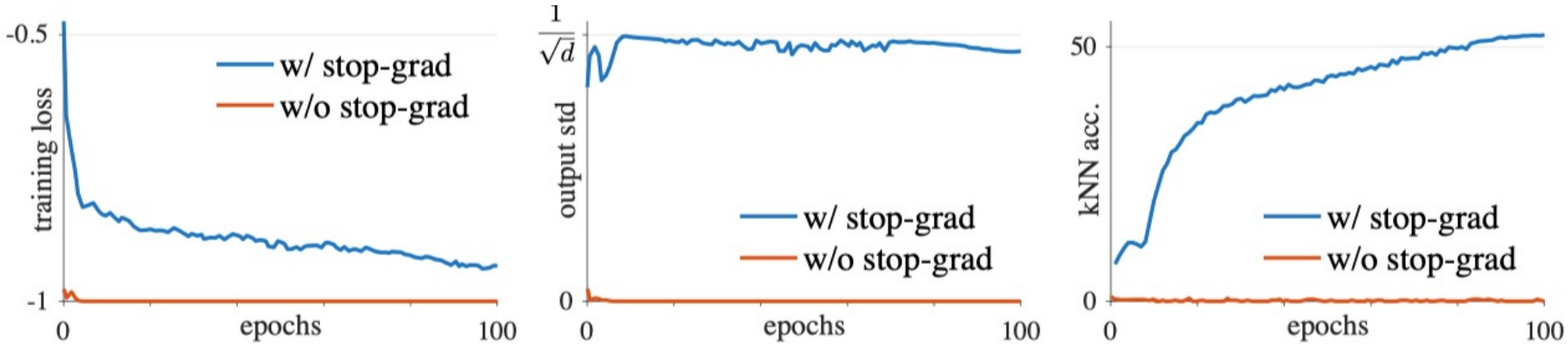
SimSiam

- Simple Siamese networks can learn meaningful representations even using none of the following: (i) negative sample pairs, (ii) large batches, (iii) momentum encoders
- Collapsing solutions do exist for the loss and structure, but a **stop-gradient** operation plays an essential role in preventing collapsing
- Explains the effectiveness of SimSiam from the perspective of the EM algorithm

Comparison on Siamese Architectures



Stop-gradient is Essential to Prevent Model Collapse



	acc. (%)
w/ stop-grad	67.7±0.1
w/o stop-grad	0.1

Figure 2. **SimSiam with *vs*. without stop-gradient**. **Left plot**: training loss. Without stop-gradient it degenerates immediately. **Middle plot**: the per-channel std of the ℓ_2-normalized output, plotted as the averaged std over all channels. **Right plot**: validation accuracy of a kNN classifier [36] as a monitor of progress. **Table**: ImageNet linear evaluation ("w/ stop-grad" is mean±std over 5 trials).

Experiments

	pred. MLP h	acc. (%)
baseline	lr with cosine decay	67.7
(a)	no pred. MLP	0.1
(b)	fixed random init.	1.5
(c)	lr not decayed	68.1

Table 1. **Effect of prediction MLP** (ImageNet linear evaluation accuracy with 100-epoch pre-training). In all these variants, we use the same schedule for the encoder f (lr with cosine decay).

	case	proj. MLP's BN hidden	proj. MLP's BN output	pred. MLP's BN hidden	pred. MLP's BN output	acc. (%)
(a)	none	-	-	-	-	34.6
(b)	hidden-only	✓	-	✓	-	67.4
(c)	default	✓	✓	✓	-	68.1
(d)	all	✓	✓	✓	✓	unstable

Table 3. **Effect of batch normalization on MLP heads** (ImageNet linear evaluation accuracy with 100-epoch pre-training).

Experiments

method	batch size	negative pairs	momentum encoder	100 ep	200 ep	400 ep	800 ep
SimCLR (repro.+)	4096	✓		66.5	68.3	69.8	70.4
MoCo v2 (repro.+)	**256**	✓	✓	67.4	69.9	71.0	72.2
BYOL (repro.)	4096		✓	66.5	**70.6**	**73.2**	**74.3**
SwAV (repro.+)	4096			66.5	69.1	70.7	71.8
SimSiam	**256**			**68.1**	70.0	70.8	71.3

	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance seg.		
pre-train	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
scratch	35.9	16.8	13.0	60.2	33.8	33.1	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet supervised	74.4	42.4	42.7	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR (repro.+)	75.9	46.8	50.1	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo v2 (repro.+)	**77.1**	**48.5**	**52.5**	**82.3**	**57.0**	**63.3**	**58.8**	**39.2**	**42.5**	**55.5**	**34.3**	**36.6**
BYOL (repro.)	**77.1**	47.0	49.9	81.4	55.3	61.1	57.8	37.9	40.9	54.3	33.2	35.0
SwAV (repro.+)	75.5	46.5	49.6	81.5	55.4	61.4	57.6	37.6	40.3	54.2	33.1	35.1
SimSiam, base	75.5	47.0	50.2	**82.0**	56.4	62.8	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam, optimal	**77.3**	**48.5**	**52.5**	**82.4**	**57.0**	**63.7**	**59.3**	**39.2**	**42.1**	**56.0**	**34.4**	**36.7**

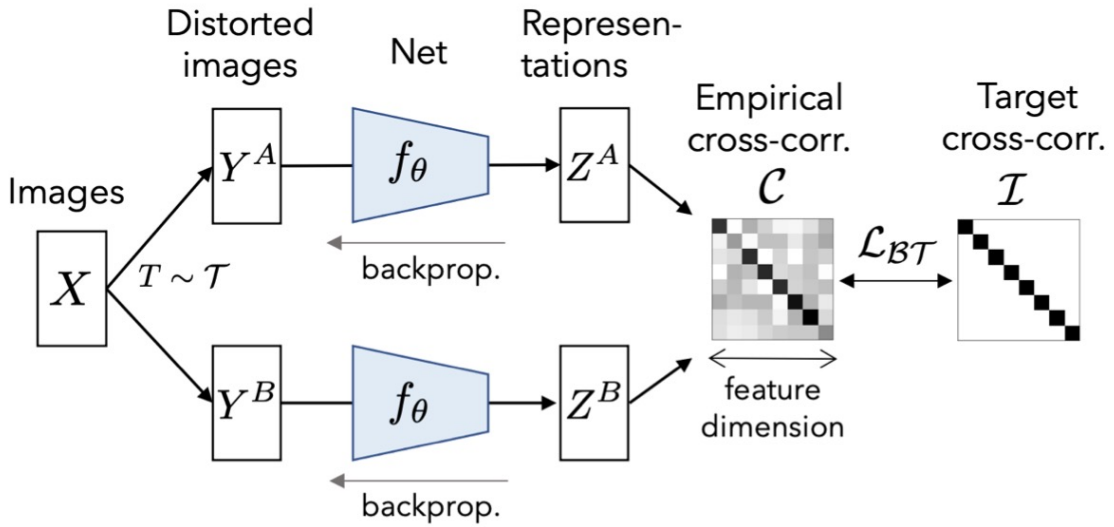
Barlow Twins

- Barlow Twins avoids model collapse by measuring the cross-correlation matrix between two views of a sample and making it as close to the identity matrix as possible

- Barlow Twins can prevent model collapse without the need for negative pairs, asymmetric structure or clustering features

Barlow Twins: Self-Supervised Learning via Redundancy Reduction, ICML 2021

Barlow Twins



Algorithm 1 PyTorch-style pseudocode for Barlow Twins.

```
# f: encoder network
# lambda: weight on the off-diagonal terms
# N: batch size
# D: dimensionality of the representation
#
# mm: matrix-matrix multiplication
# off_diagonal: off-diagonal elements of a matrix
# eye: identity matrix

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    y_a, y_b = augment(x)

    # compute representations
    z_a = f(y_a) # NxD
    z_b = f(y_b) # NxD

    # normalize repr. along the batch dimension
    z_a_norm = (z_a - z_a.mean(0)) / z_a.std(0) # NxD
    z_b_norm = (z_b - z_b.mean(0)) / z_b.std(0) # NxD

    # cross-correlation matrix
    c = mm(z_a_norm.T, z_b_norm) / N # DxD

    # loss
    c_diff = (c - eye(D)).pow(2) # DxD
    # multiply off-diagonal elems of c_diff by lambda
    off_diagonal(c_diff).mul_(lambda)
    loss = c_diff.sum()

    # optimization step
    loss.backward()
    optimizer.step()
```

Experiments

Table 1. **Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet**. All models use a ResNet-50 encoder. Top-3 best self-supervised methods are underlined.

Method	Top-1	Top-5
Supervised	76.5	
MoCo	60.6	
PIRL	63.6	-
SimCLR	69.3	89.0
MoCo v2	71.1	90.1
SimSiam	71.3	-
SwAV (w/o multi-crop)	71.8	-
BYOL	74.3	91.6
SwAV	75.3	-
Barlow Twins (ours)	73.2	91.0

Table 3. **Transfer learning: image classification.** We benchmark learned representations on the image classification task by training linear classifiers on fixed features. We report top-1 accuracy on Places-205 and iNat18 datasets, and classification mAP on VOC07. Top-3 best self-supervised methods are underlined.

Method	Places-205	VOC07	iNat18
Supervised	53.2	87.5	46.7
SimCLR	52.5	85.5	37.2
MoCo-v2	51.8	86.4	38.6
SwAV (w/o multi-crop)	52.8	86.4	39.5
SwAV	56.7	88.9	48.6
BYOL	54.0	86.6	47.6
Barlow Twins (ours)	54.1	86.2	46.5

Ablations

Table 5. **Loss function explorations**. We ablate the invariance and redundancy terms in our proposed loss and observe that both terms are necessary for good performance. We also experiment with different normalization schemes and a cross-entropy loss and observe reduced performance.

Loss function	Top-1	Top-5
Baseline	71.4	90.2
Only invariance term (on-diag term)	57.3	80.5
Only red. red. term (off-diag term)	0.1	0.5
Normalization along feature dim.	69.8	88.8
No BN in MLP	71.2	89.7
No BN in MLP + no Normalization	53.4	76.7
Cross-entropy with temp.	63.3	85.7

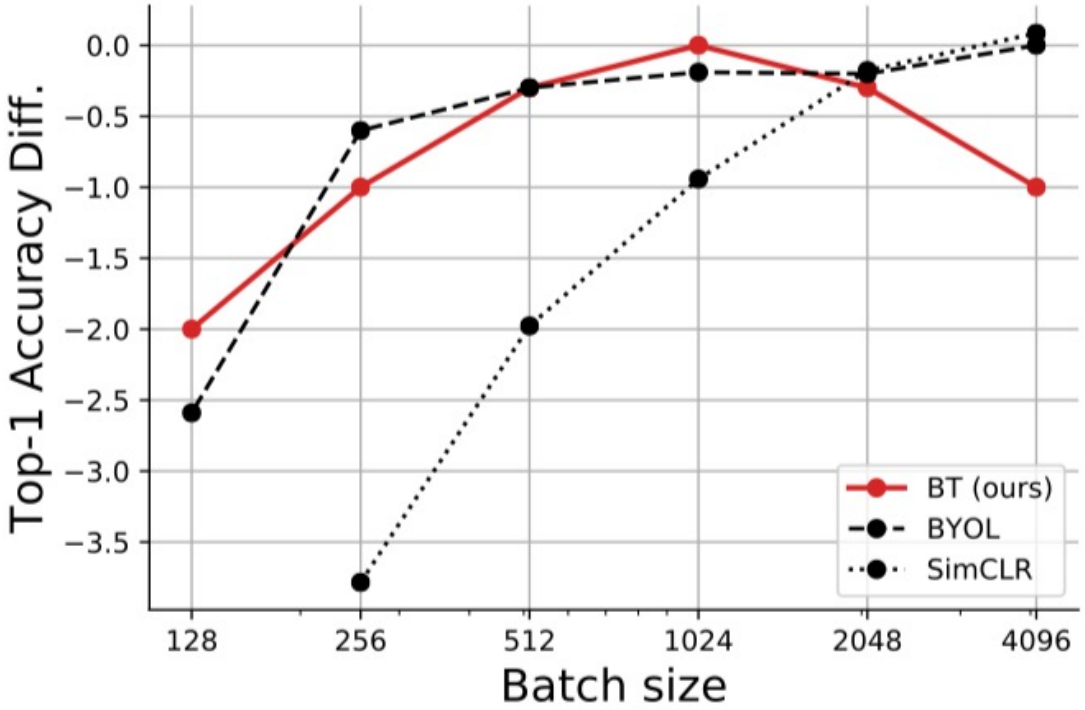


Figure 2. Effect of batch size. To compare the effect of the batch size across methods, for each method we report the difference between the top-1 accuracy at a given batch size and the best obtained accuracy among all batch size tested. BYOL: best accuracy is 72.5% for a batch size of 4096 (data from (Grill et al., 2020) fig. 3A). SIMCLR: best accuracy is 67.1% for a batch size of 4096 (data from (Chen et al., 2020a) fig. 9, model trained for 300 epochs). BARLOW TWINS: best accuracy is 71.7% for a batch size of 1024.

Ablations

Table 6. Effect of asymmetric settings

case	stop-gradient	predictor	Top-1	Top-5
Baseline	-	-	71.4	90.2
(a)	✓	-	70.5	89.0
(b)	-	✓	70.2	89.0
(c)	✓	✓	61.3	83.5

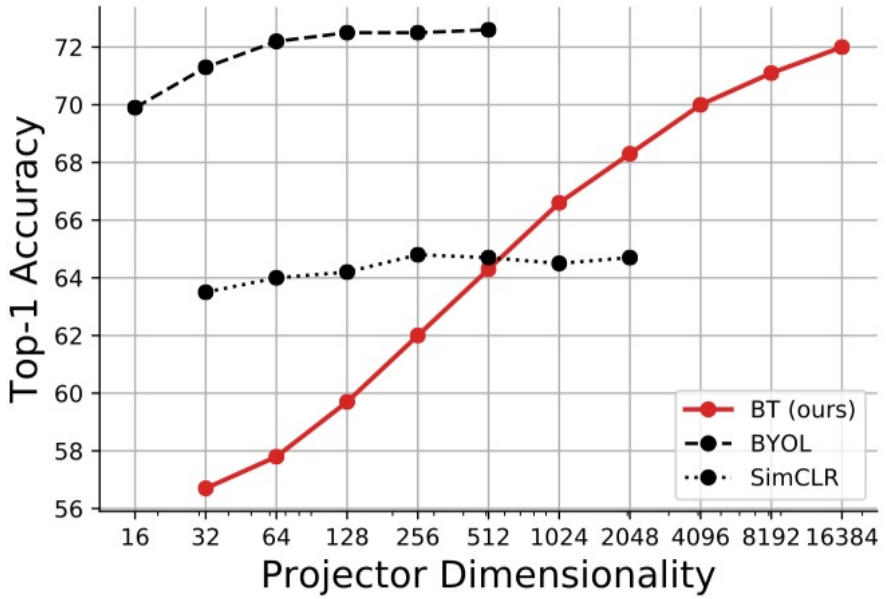


Figure 4. Effect of the dimensionality of the last layer of the projector network on performance. The parameter λ is kept fix for all dimensionalities tested. Data for SIMCLR is from (Chen et al., 2020a) fig 8; Data for BYOL is from (Grill et al., 2020) Table 14b.

Connection with InfoNCE

$$\mathcal{L}_{infoNCE} \triangleq \underbrace{-\sum_b \frac{\langle z_b^A, z_b^B \rangle_i}{\tau \left\| z_b^A \right\|_2 \left\| z_b^B \right\|_2}}_{\text{similarity term}} + \underbrace{\sum_b \log \left(\sum_{b' \neq b} \exp \left(\frac{\langle z_b^A, z_{b'}^B \rangle_i}{\tau \left\| z_b^A \right\|_2 \left\| z_{b'}^B \right\|_2} \right) \right)}_{\text{contrastive term}}$$

$$\mathcal{L}_{\mathcal{BT}} = \underbrace{\sum_i \left(1 - \frac{\langle z_{\cdot,i}^A, z_{\cdot,i}^B \rangle_b}{\left\| z_{\cdot,i}^A \right\|_2 \left\| z_{\cdot,i}^B \right\|_2} \right)^2}_{\text{invariance term}} + \underbrace{\lambda \sum_i \sum_{j \neq i} \left(\frac{\langle z_{\cdot,i}^A, z_{\cdot,j}^B \rangle_b}{\left\| z_{\cdot,i}^A \right\|_2 \left\| z_{\cdot,j}^B \right\|_2} \right)^2}_{\text{redundancy reduction term}}$$

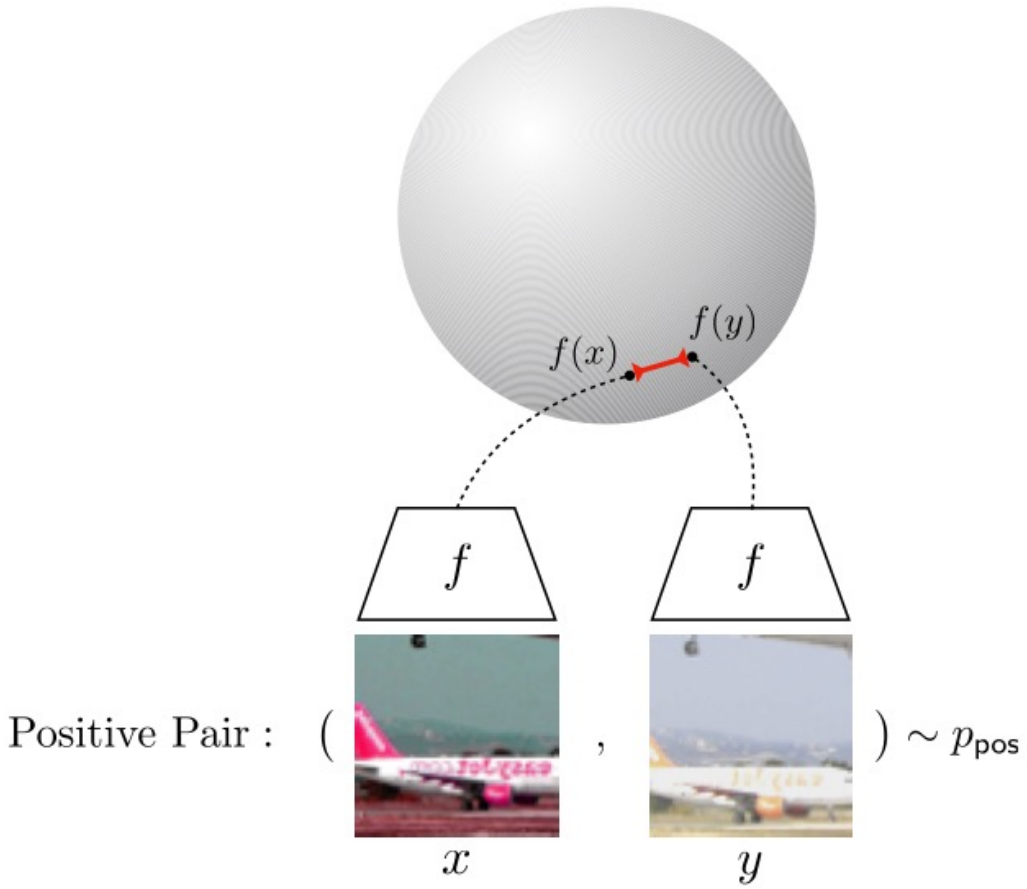
Barlow Twins: Self-Supervised Learning via Redundancy Reduction, ICML 2021

Understanding Contrastive SSL

Alignment and Uniform

- Alignment: two samples forming a positive pair should be mapped to nearby features, and thus be (mostly) invariant to unneeded noise factors
- Uniformity: feature vectors should be roughly uniformly distributed on the unit hypersphere, preserving as much information of the data as possible
- Both alignment and uniformstrongly agree with downstream task performance

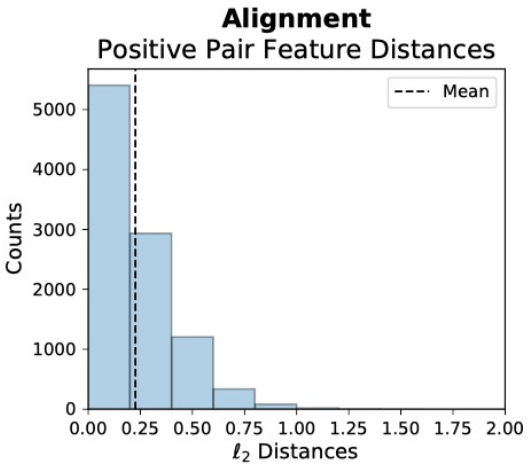
Understanding Alignment and Uniform

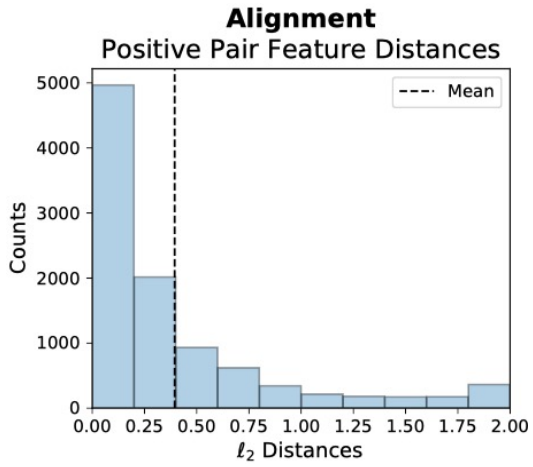


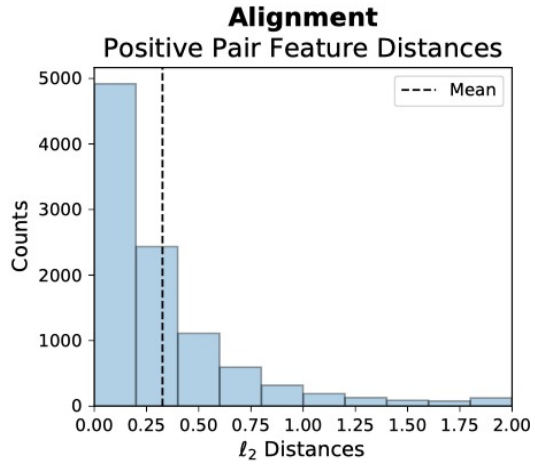
Alignment: Similar samples have similar features.

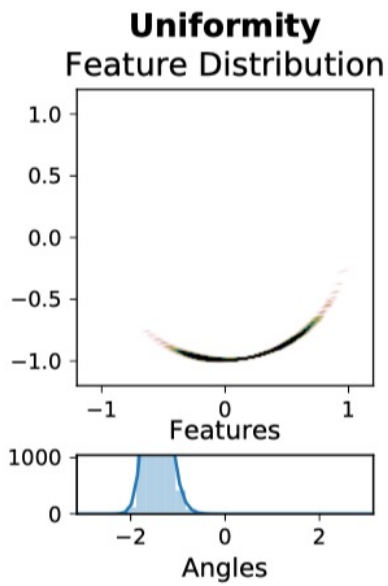
Uniformity: Preserve maximal information.

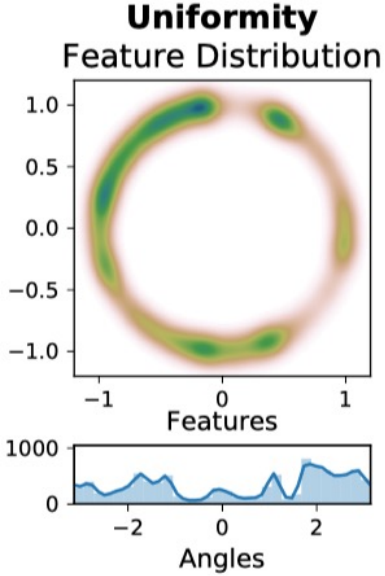
Alignment and Uniform Analysis

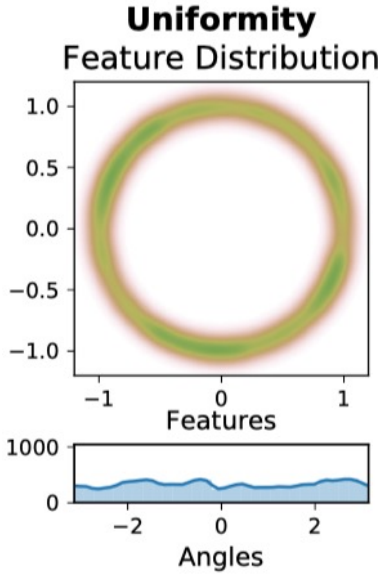












Random Initialization **Supervised Predictive Learning** **Unsupervised Contrastive Learning**

Optimizing Alignment and Uniform

$$\mathcal{L}_{\text{align}}(f;\alpha) \triangleq \mathbb{E}_{(x,y)\sim p_{\text{pos}}}\left[\|f(x)-f(y)\|_2^{\alpha}\right]$$

$$\mathcal{L}_{\text{uniform}}(f;t) \triangleq \log \mathbb{E}_{x,y \overset{\text{i.i.d.}}{\sim} p_{\text{data}}}\left[e^{-t\|f(x)-f(y)\|_2^2}\right]$$

$$\mathcal{L}_{\text{contrastive}}(f;\tau,M) = \underbrace{\mathop{\mathbb{E}}_{(x,y)\sim p_{\text{pos}}}\left[-f(x)^{\mathsf{T}}f(y)/\tau\right]}_{\text{Alignment}} + \underbrace{\mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{x_i^-\}_{i=1}^{M} \overset{\text{i.i.d.}}{\sim} p_{\text{data}}}}\left[\log\left(e^{f(x)^{\mathsf{T}}f(y)/\tau}+\sum_i e^{f(x_i^-)^{\mathsf{T}}f(x)/\tau}\right)\right]}_{\text{Uniform}}$$

Experiments

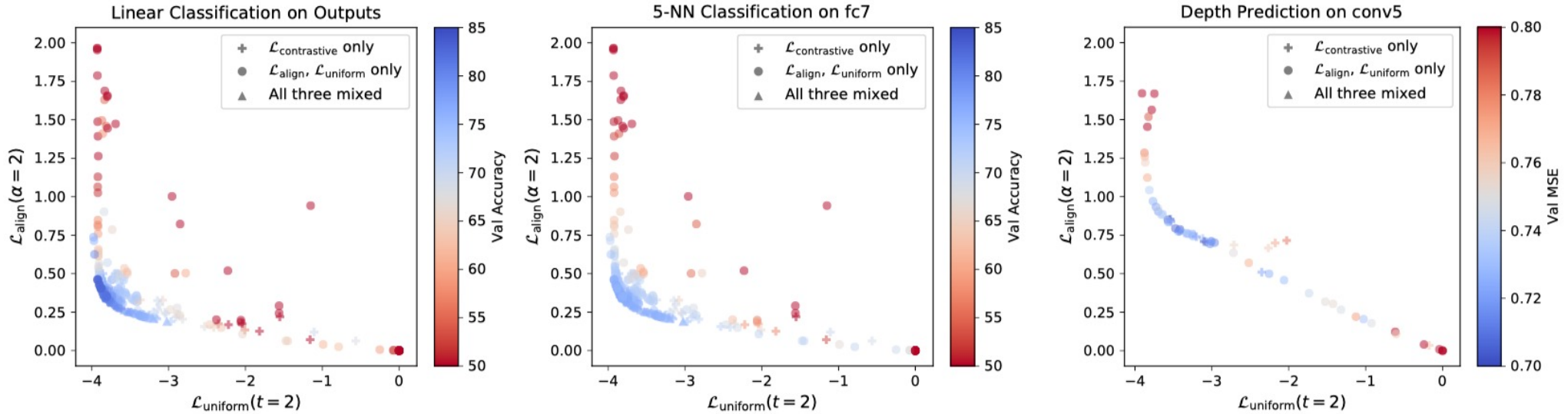
- Directly optimizing only alignment and uniform can lead to better representations

	STL-10 Validation Set Accuracy ↑				NYU-DEPTH-V2 Validation Set MSE ↓	
	Output + Linear	Output + 5-NN	fc7 + Linear	fc7 + 5-NN	conv5	conv4
Best $\mathcal{L}_{\text{contrastive}}$ only	80.46%	78.75%	83.89%	76.33%	0.7024	**0.7575**
Best $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ only	**81.15%**	78.89%	**84.43%**	**76.78%**	**0.7014**	0.7592

	IMAGENET-100 MoCo-based Encoders		BOOKCORPUS Quick-Though-Vectors-based Encoders	
	top1 Val. Accuracy ↑	top5 Val. Accuracy ↑	MR Val. Accuracy ↑	CR Val. Accuracy ↑
Best $\mathcal{L}_{\text{contrastive}}$ only	72.80%	91.64%	**77.51%**	**83.86%**
Best $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ only	**74.60%**	**92.74%**	73.76%	80.95%

Experiments

- Both alignment and uniform strongly agree with downstream task performance.



(a) 306 STL-10 encoders are evaluated with linear classification on output features and 5-nearest neighbor (5-NN) on fc7 activations. Higher accuracy (blue color) is better.

(b) 64 NYU-DEPTH-V2 encoders are evaluated with CNN depth regressors on conv5 activations. Lower MSE (blue color) is better.

Experiments

- Both alignment and uniform causally affect downstream task performance

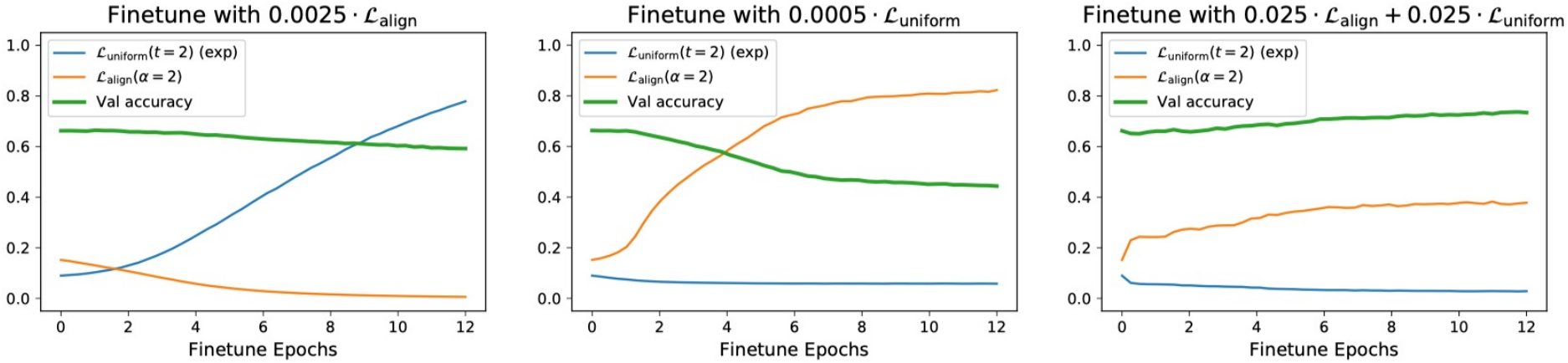


Figure 7: Finetuning trajectories from a STL-10 encoder trained with $\mathcal{L}_{\text{contrastive}}$ using a suboptimal temperature $\tau = 2.5$. Finetuning objectives are weighted combinations of $\mathcal{L}_{\text{align}}(\alpha=2)$ and $\mathcal{L}_{\text{uniform}}(t=2)$. For each intermediate checkpoint, we measure $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ metrics, as well as validation accuracy of a linear classifier trained from scratch on the encoder outputs. $\mathcal{L}_{\text{uniform}}$ is exponentiated for plotting purpose. **Left and middle:** Performance degrades if only one of alignment and uniformity is optimized. **Right:** Performance improves when both are optimized.

The Behavior of Contrastive Loss

- The contrastive loss is a hardness-aware loss function, and the temperature τ controls the strength of penalties on hard negative samples
- The hardness-aware property is significant to the success of the softmax-based contrastive loss
- Too low temperature or too high temperature lead to worse representation

Simple Loss and Contrastive Loss

$$\mathcal{L}(x_i) = -\log\left[\frac{\exp(s_{i,i}/\tau)}{\sum_{k\neq i}\exp(s_{i,k}/\tau) + \exp(s_{i,i}/\tau)}\right]$$

$$\mathcal{L}_{simple}(x_i) = -s_{i,i} + \lambda\sum_{i\neq j} s_{i,j}$$

The Role of Temperature

- τ -> 0, only focus on the most hard negative sample

$$= \lim_{\tau \to 0^+} \frac{1}{\tau} max[s_{max} - s_{i,i}, 0]$$

- τ -> ∞, the contrastive loss degraded to simple loss

$$= \lim_{\tau \to +\infty} -\frac{N-1}{N\tau} s_{i,i} + \frac{1}{N\tau} \sum_{k \neq i} s_{i,k} + \log N$$

Hard Negative Mining

数据集	Contrastive Loss (Eq 1)	Simple Loss (Eq 2)
CIFAR-10	79.75	74.83
CIFAR-100	51.82	39.31
ImageNet-100	71.53	48.09
SVHN	92.55	70.83

数据集	Contrastive Loss (Eq1)	Simple Loss (Eq2) + Hard
CIFAR-10	79.75	84.84
CIFAR-100	51.82	55.71
ImageNet-100	71.53	74.31
SVHN	92.55	94.99

Verify the Role of Temperature

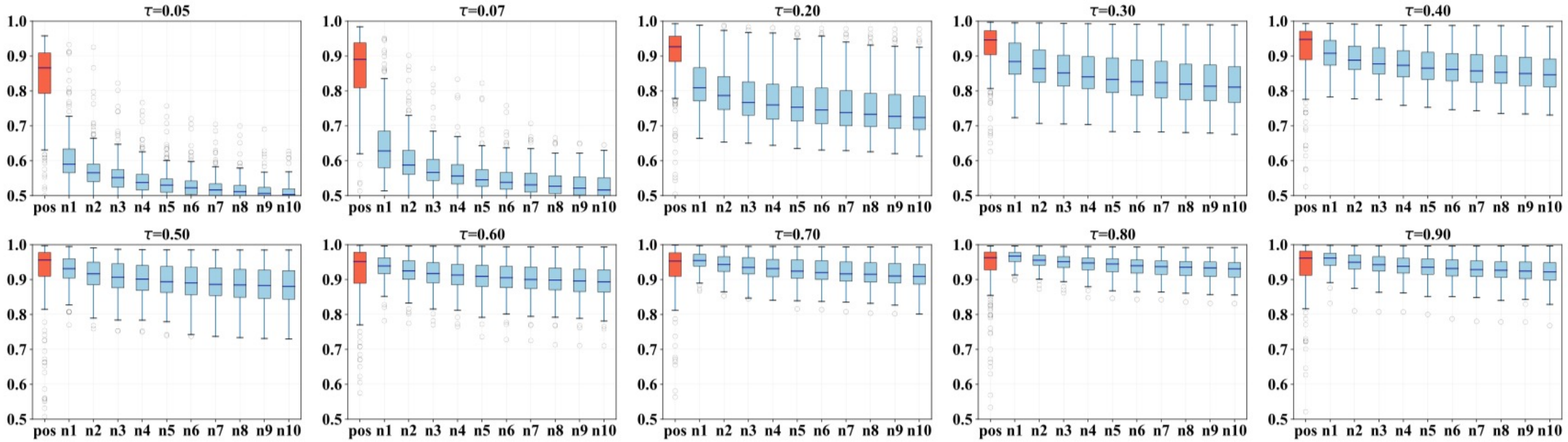


Figure 8. We display the similarity distribution of positive samples and the top-10 nearest negative samples that are marked as 'pos' and 'ni' for the i-th nearest neighbour. All models are trained on CIFAR100. For models trained on other datasets, they present the same pattern with the above figure, and we display them in the supplementary material.

Better Sampling and Harder Samples

Debiased Sampling

- Debiased contrastive learning develop a new, debiased contrastive objective that corrects for the sampling bias of negative examples, while only assuming access to positive examples and the unlabeled data

- With debiased sampling, the proposed objective consistently outperforms the state-of-the-art for representation learning in vision, language, and reinforcement learning benchmarks.

Sampling Bias

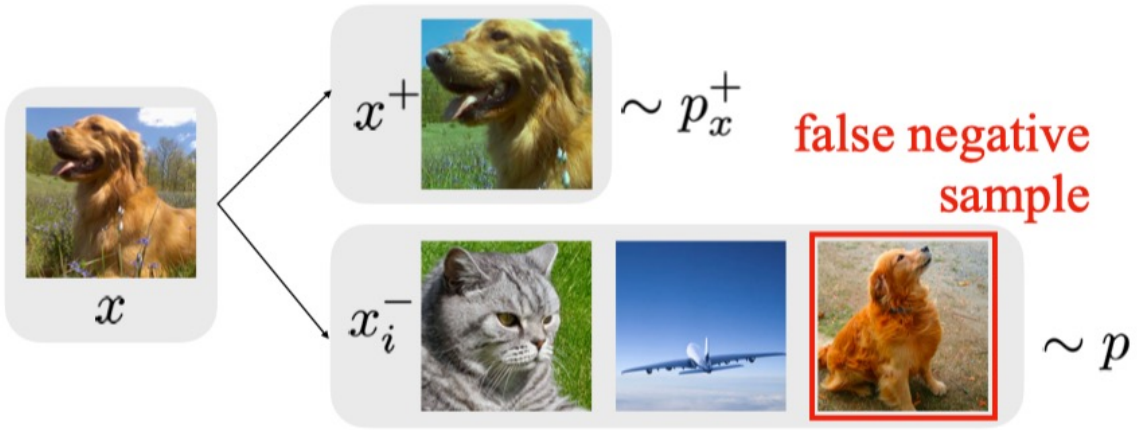


Figure 1: **"Sampling bias":** The common practice of drawing negative examples x_i^- from the data distribution $p(x)$ may result in x_i^- that are actually similar to x.

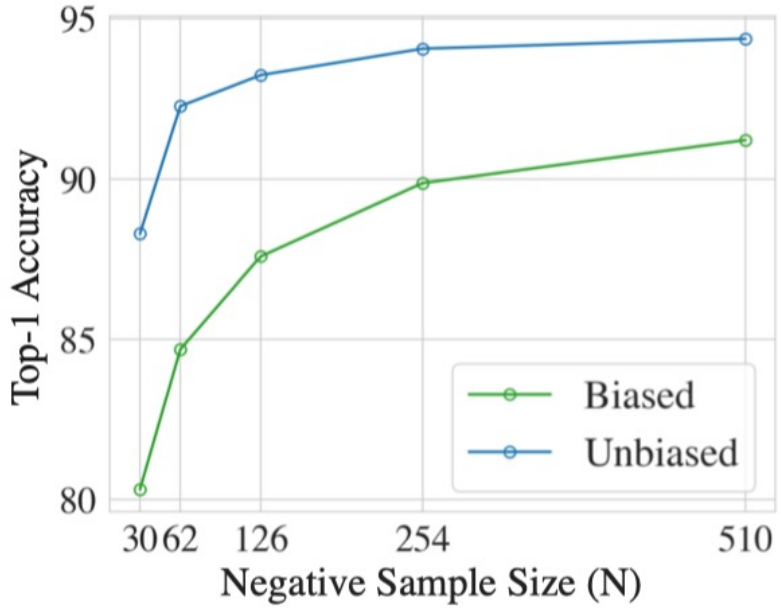
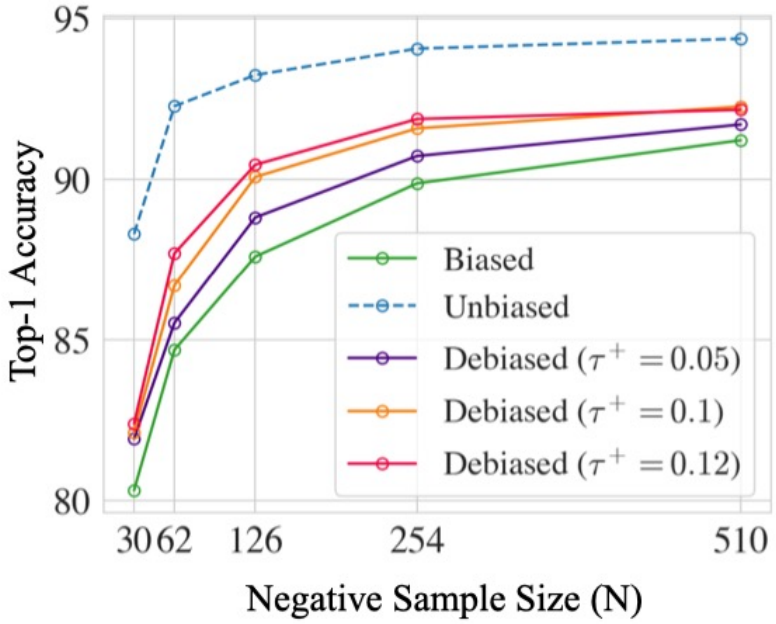
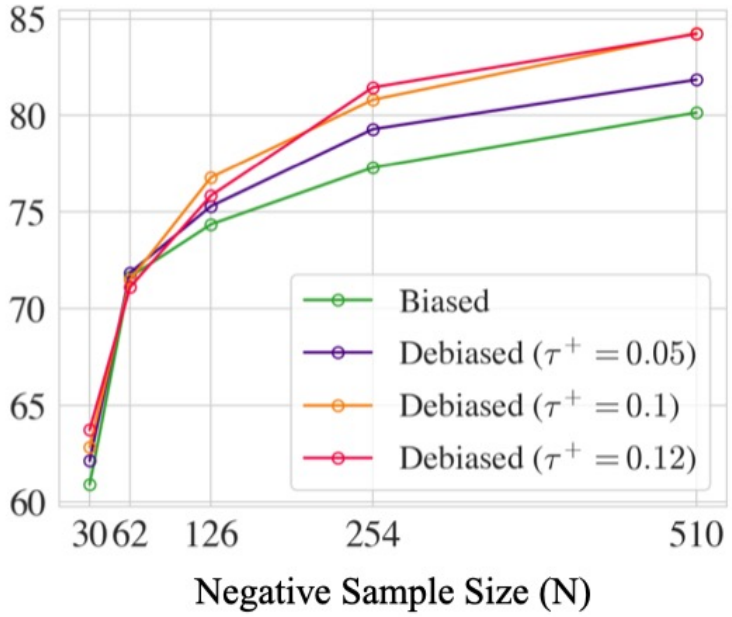


Figure 2: **Sampling bias leads to performance drop:** Results on CIFAR-10 for drawing x_i^- from $p(x)$ (biased) and from data with different labels, i.e., truly semantically different data (unbiased).

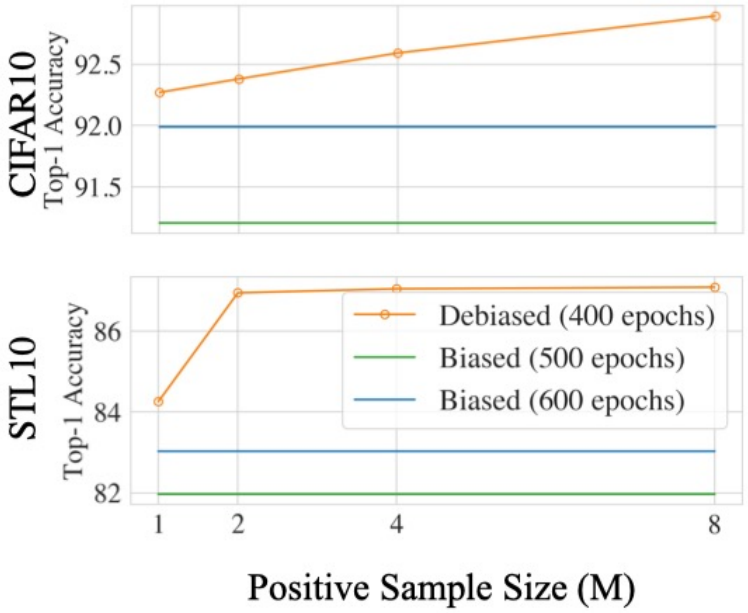
Experiments



(a) CIFAR10 (M=1)



(b) STL10 (M=1)



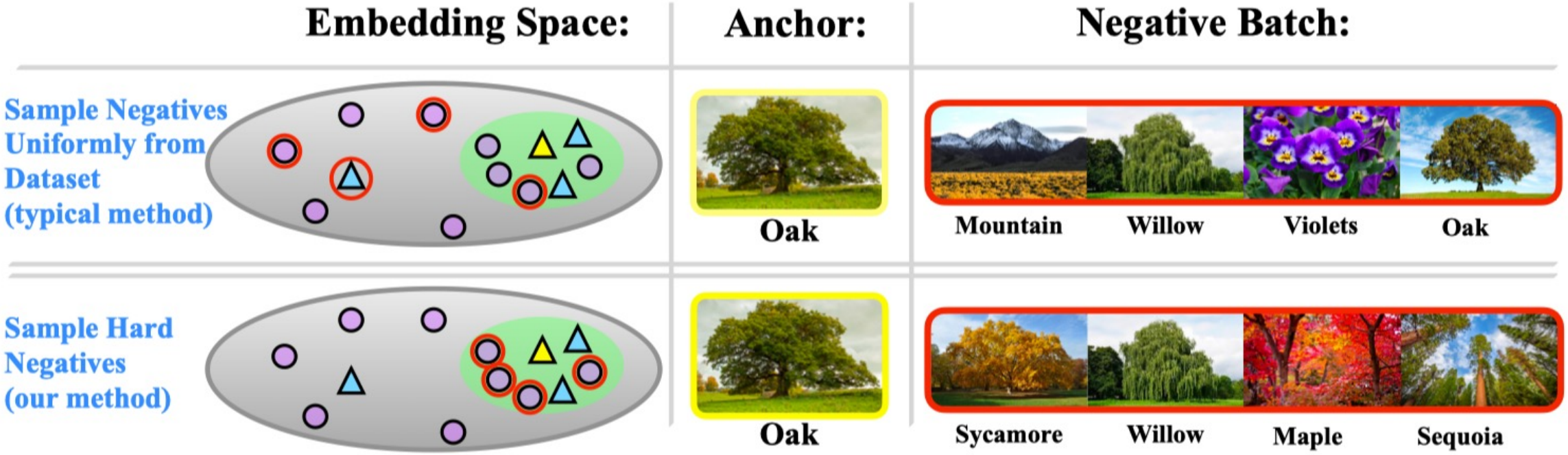
(c) Effect of Positive Samples

Hard Negatives

- The key challenge toward using hard negatives is that contrastive methods must remain unsupervised, making it infeasible to adopt existing negative sampling strategies that use true similarity information

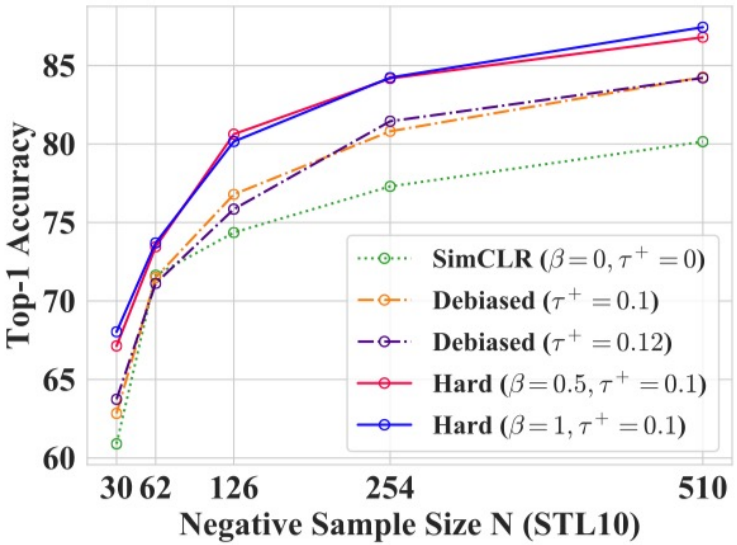
- This paper develop a new family of unsupervised sampling methods for selecting hard negative samples where the user can control the hardness

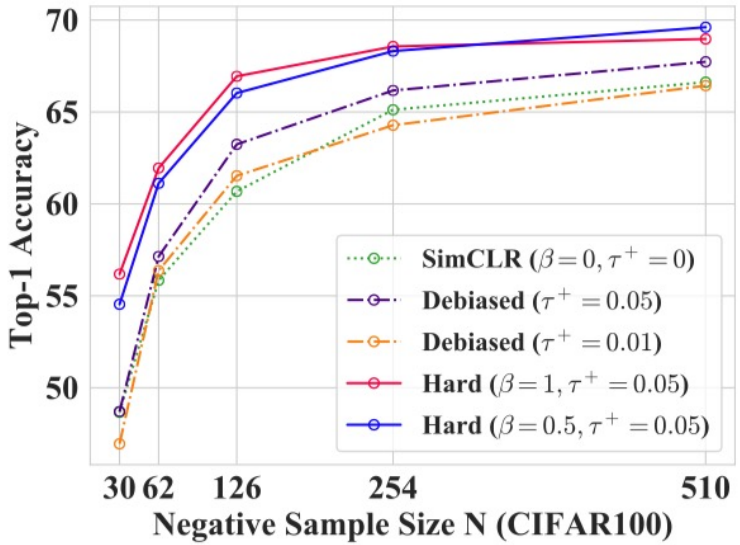
Hard Negative Mining

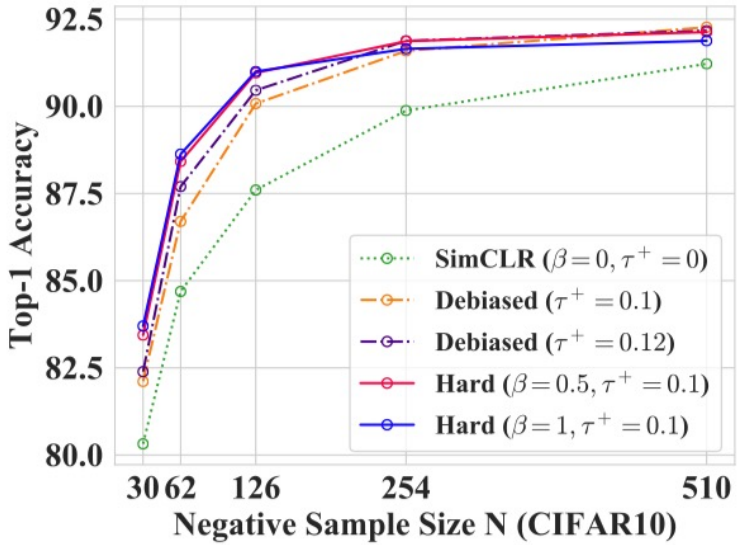


Contrastive learning with hard negative samples, ICLR 2021

Experiments





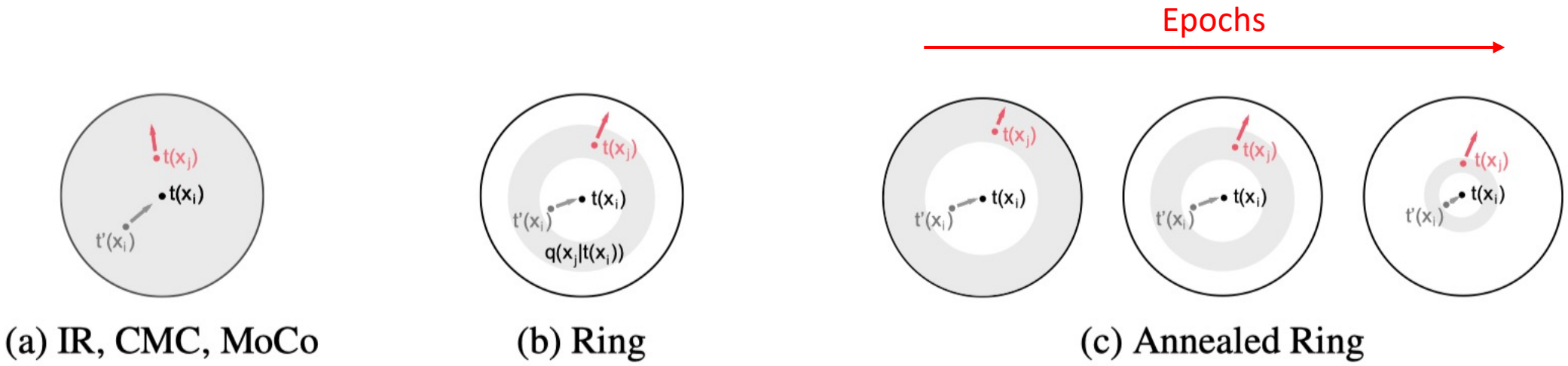


Contrastive learning with hard negative samples, ICLR 2021

MoCoRing

- Randomly sampled negative examples often include many uninformative examples either because they are too easy or too hard to discriminate
- Semi-hard negatives can yield stronger contrastive representations
- The naive strategy of sampling hard negatives throughout training can be detrimental. We then show that slowly introducing harder negatives yields good performance.

MoCoRing



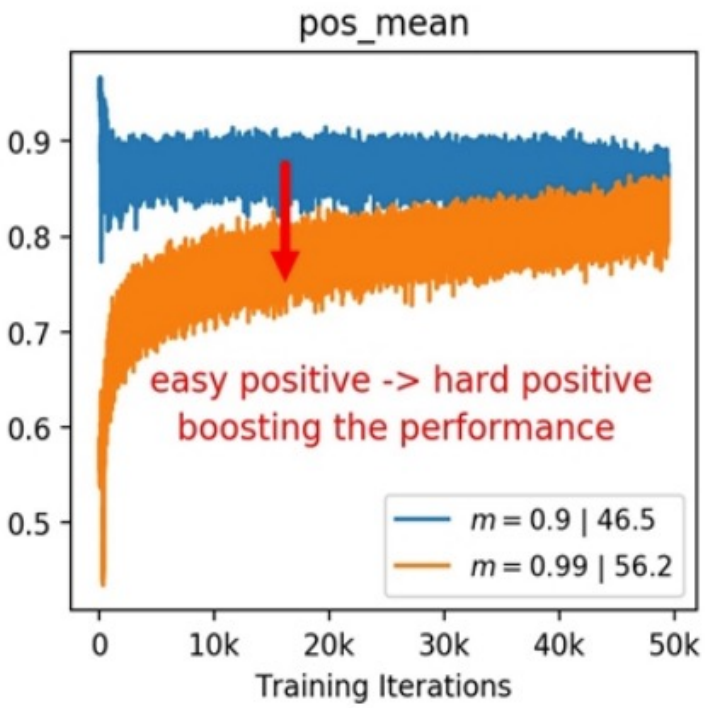
(a) IR, CMC, MoCo (b) Ring (c) Annealed Ring

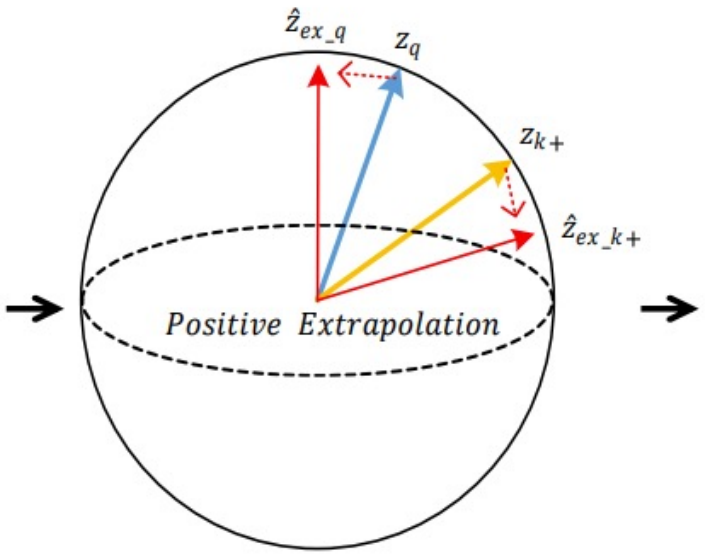
Figure 1: Visual illustration of Ring Discrimination. Black: view of example x_i; gray: second view of x_i; red: negative samples; gray area: distribution $q(x|t(x_i))$. In subfigure (c), the negative samples are annealed to be closer to $t(x_i)$ through training. In other words, the support of q shrinks.

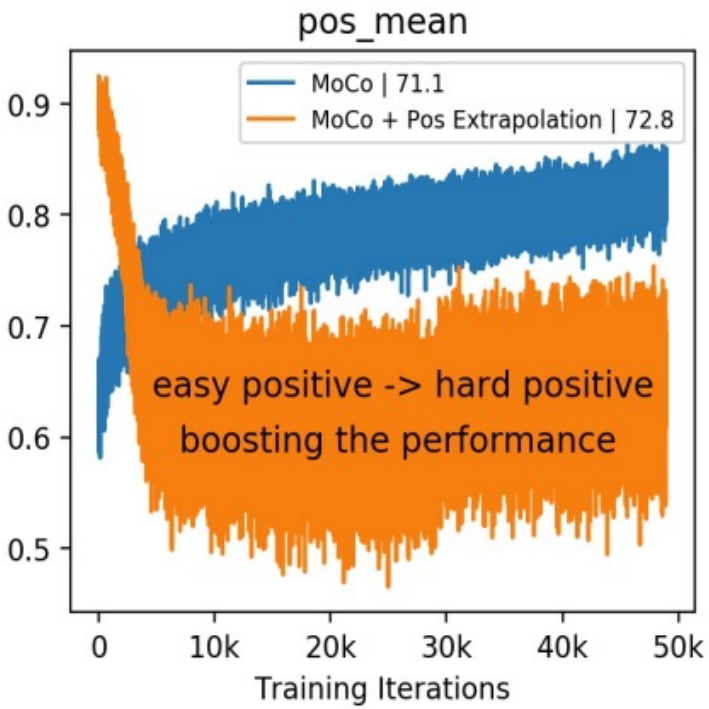
Feature Transformation

- Harder positives can used to boost the learning because hard positives enable the model to be more view-invariant
- Diversified negatives make the model more discriminative
- This paper proposes to create hard positives and diversified negatives by feature transformation

Hard Positives Can Learn Better Representation

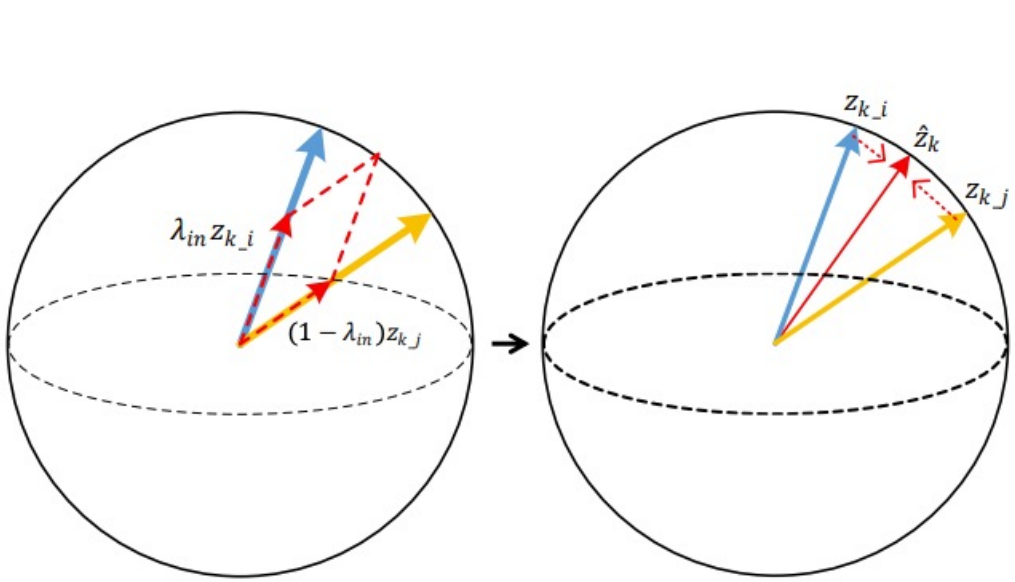




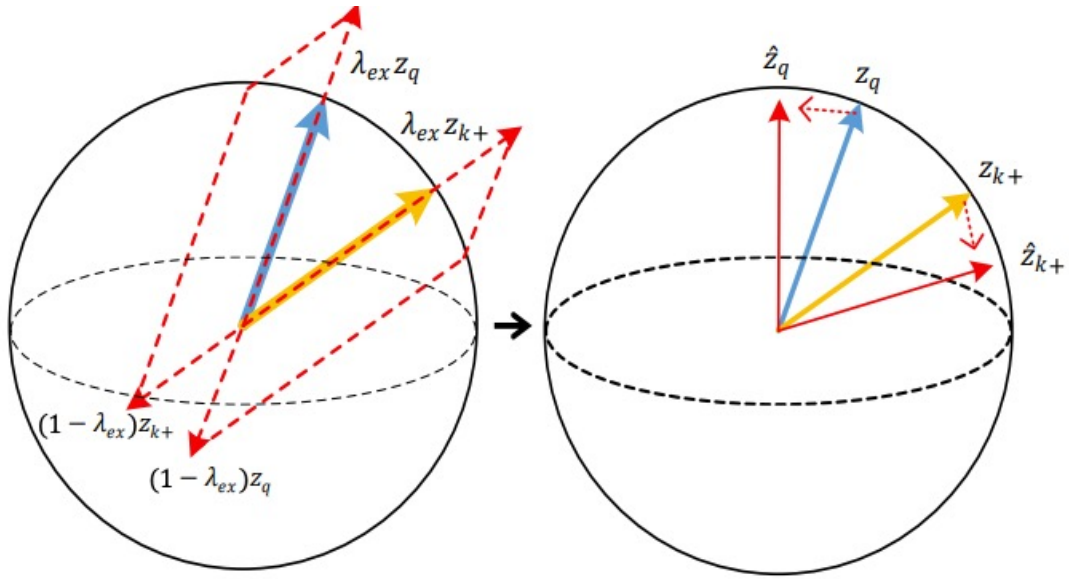


(a) Observation (b) Proposed Method (c) Performance Gain

Interpolation for Positives and Negatives



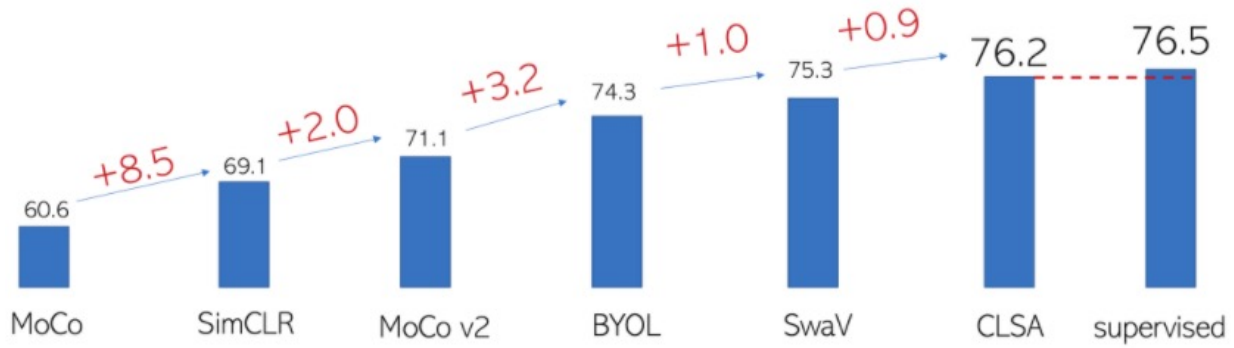
(a) Negative Interpolation



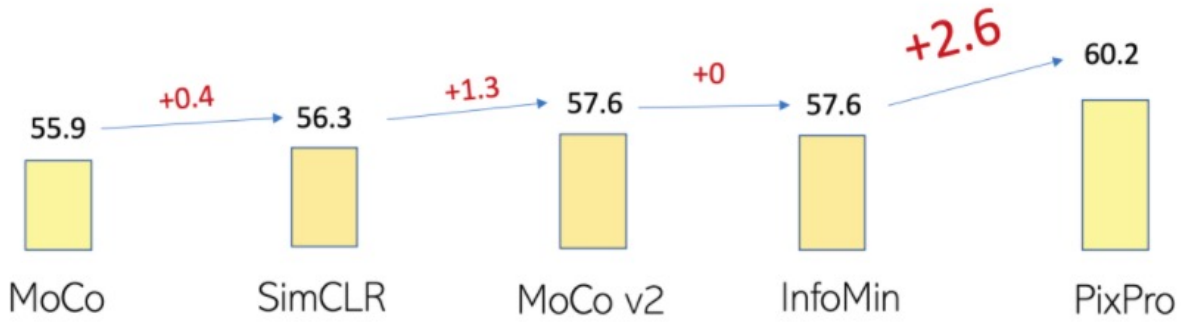
(b) Positive Extrapolation

Focus on Downstream Tasks

Is Image-Level Contrast Learning Good Enough?

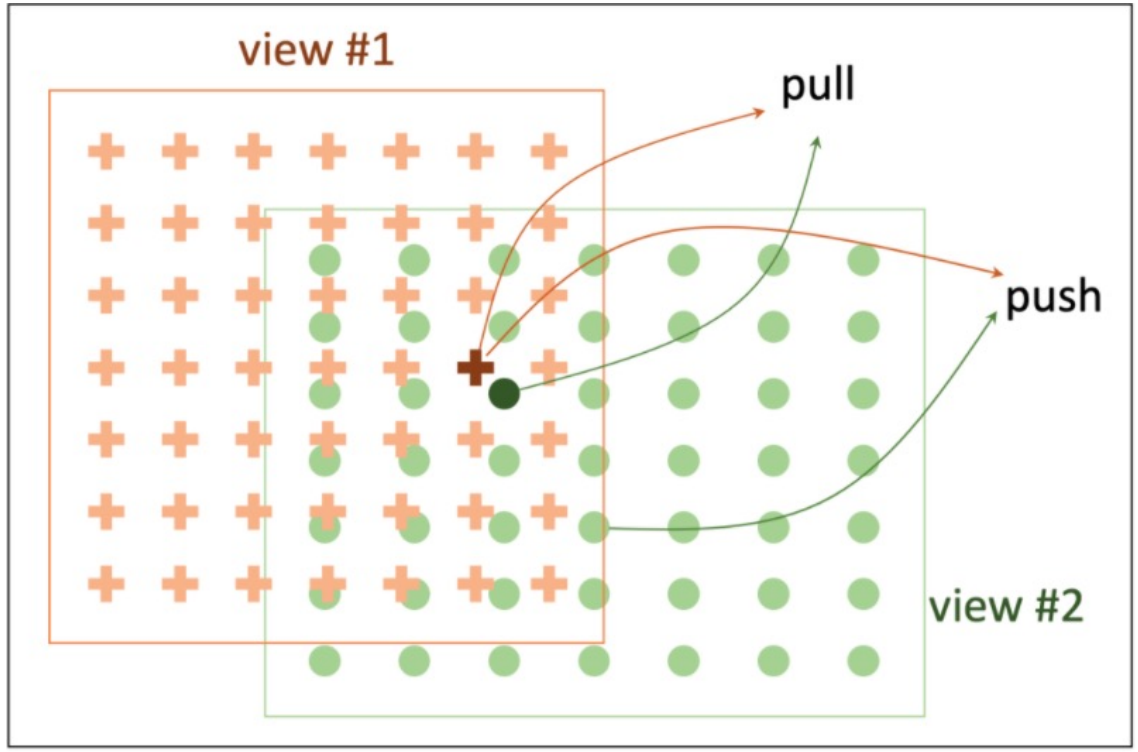


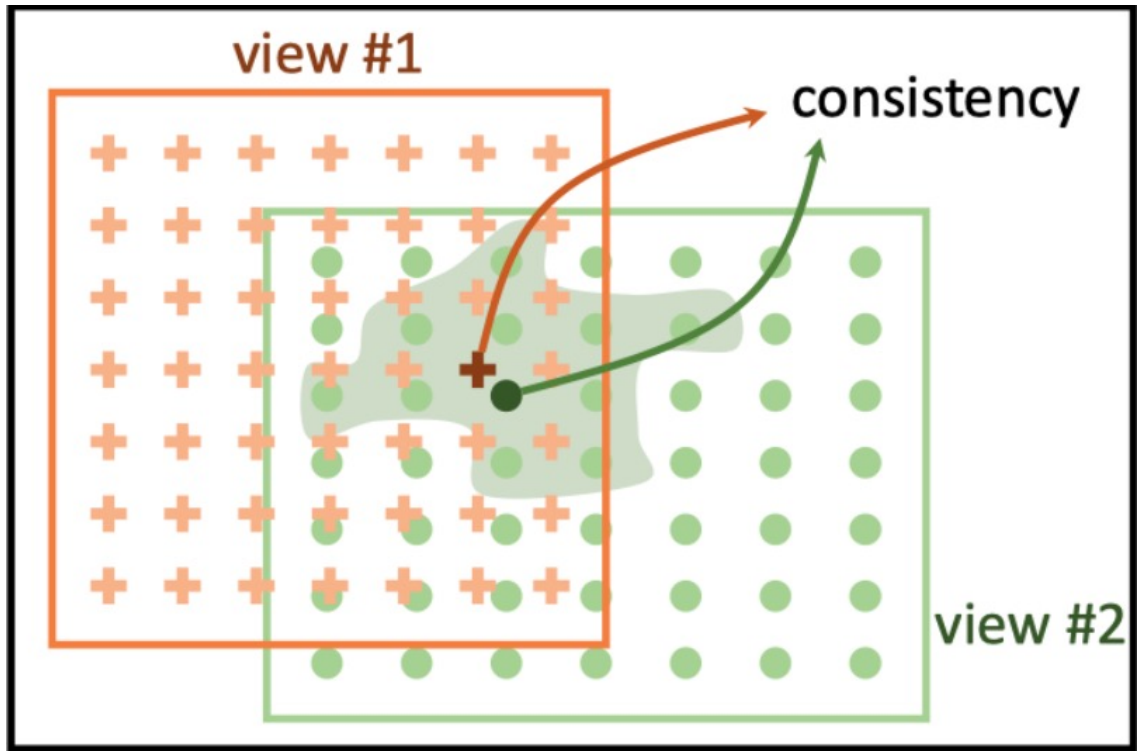
Totally 15.6% absolute improvements in 1 year!



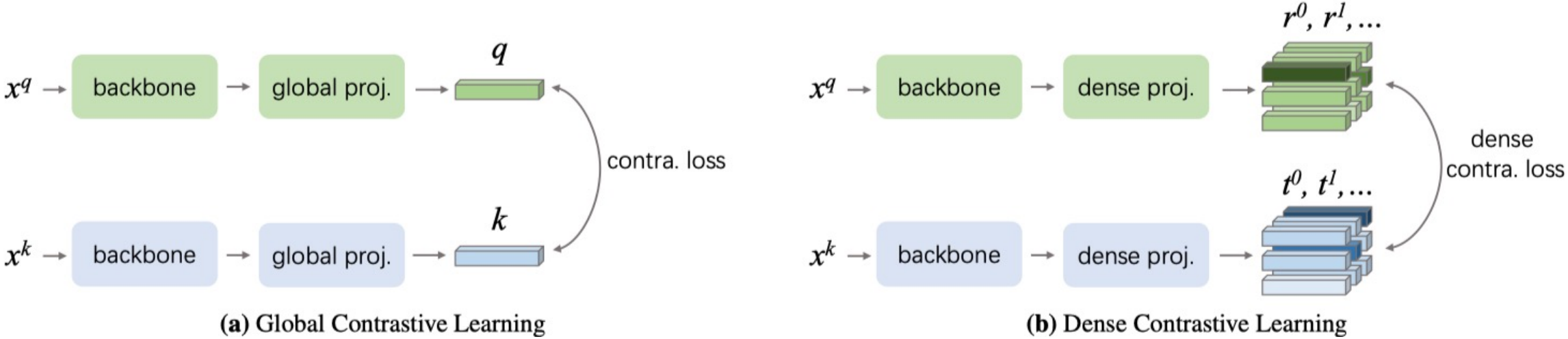
Totally 1.7% absolute improvements in 1 year!

Pixel-level Contrastive Learning





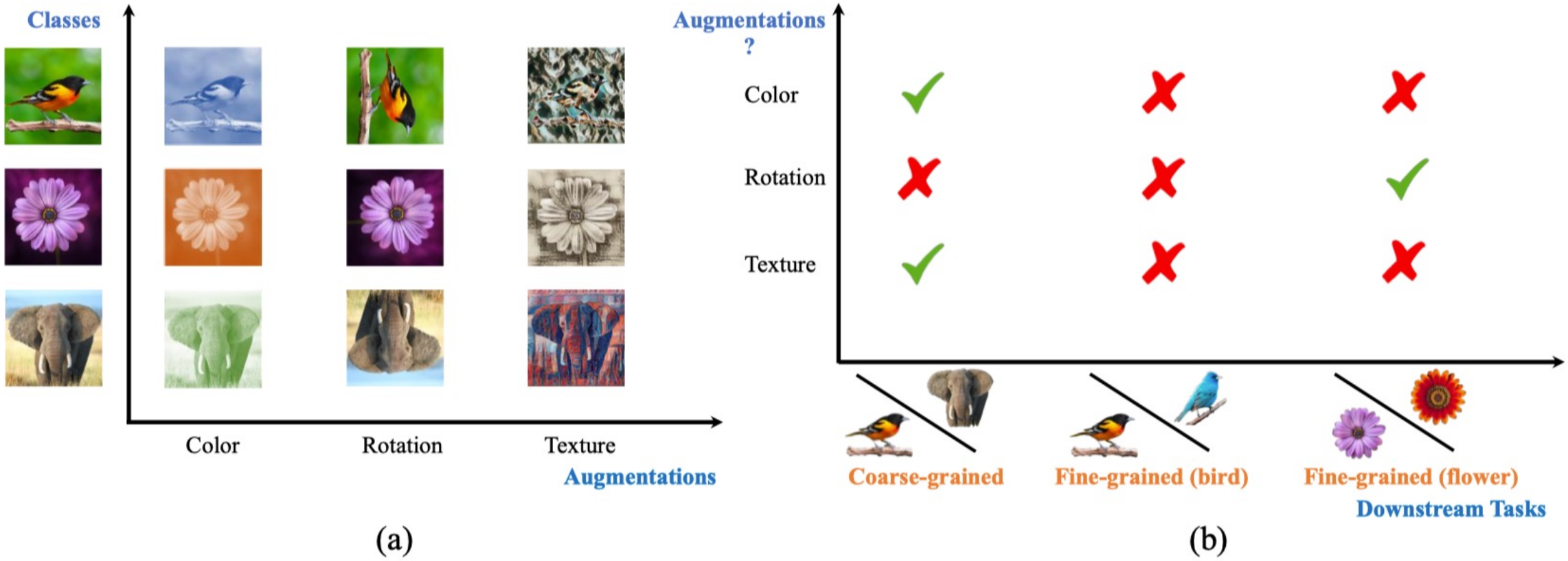
Patch-level Contrastive Learning



(a) Global Contrastive Learning

(b) Dense Contrastive Learning

Different Augmentations for Different Tasks



Beyond Image Contrastive Learning

Multi-modal Self-Supervised Learning

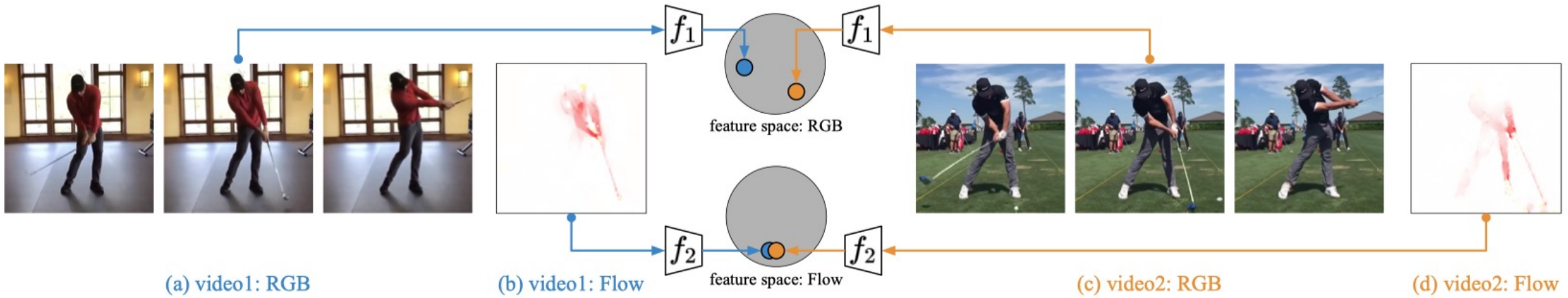
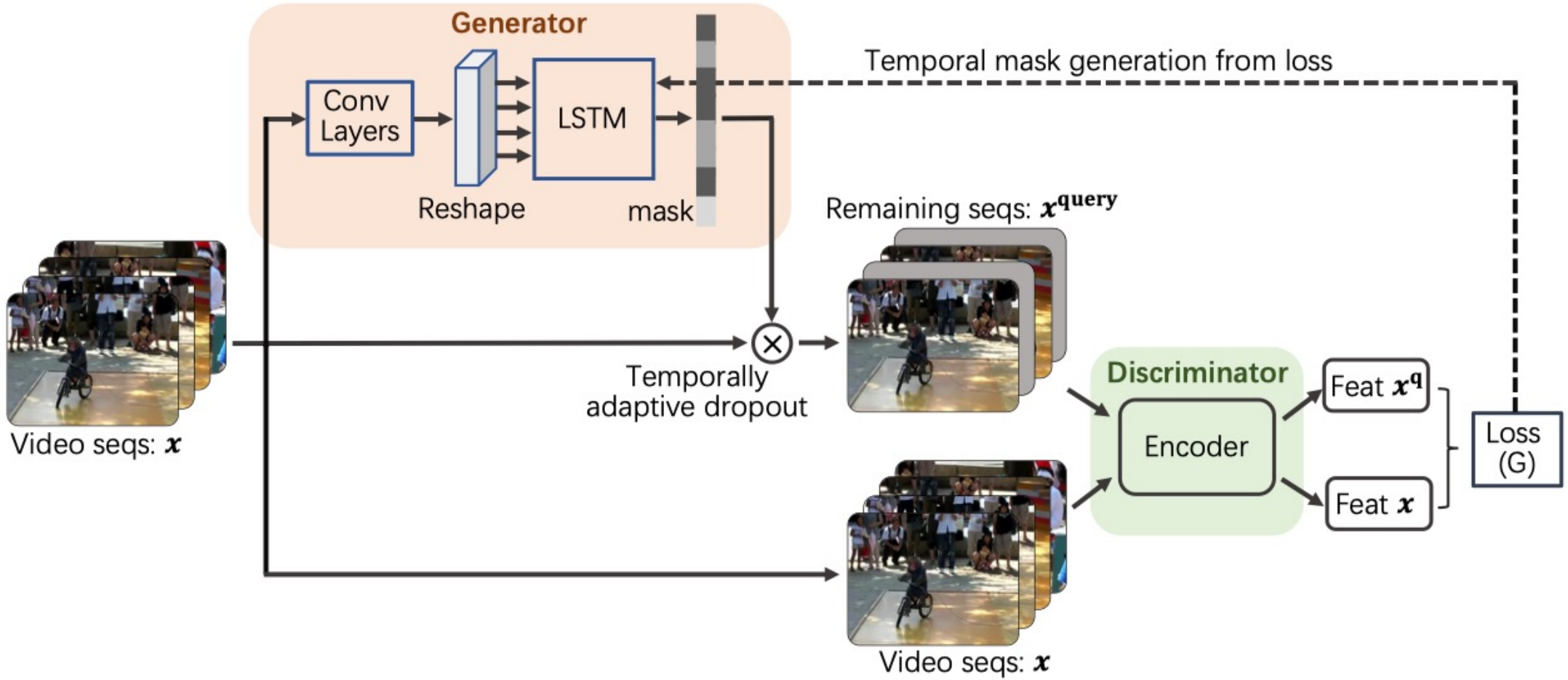


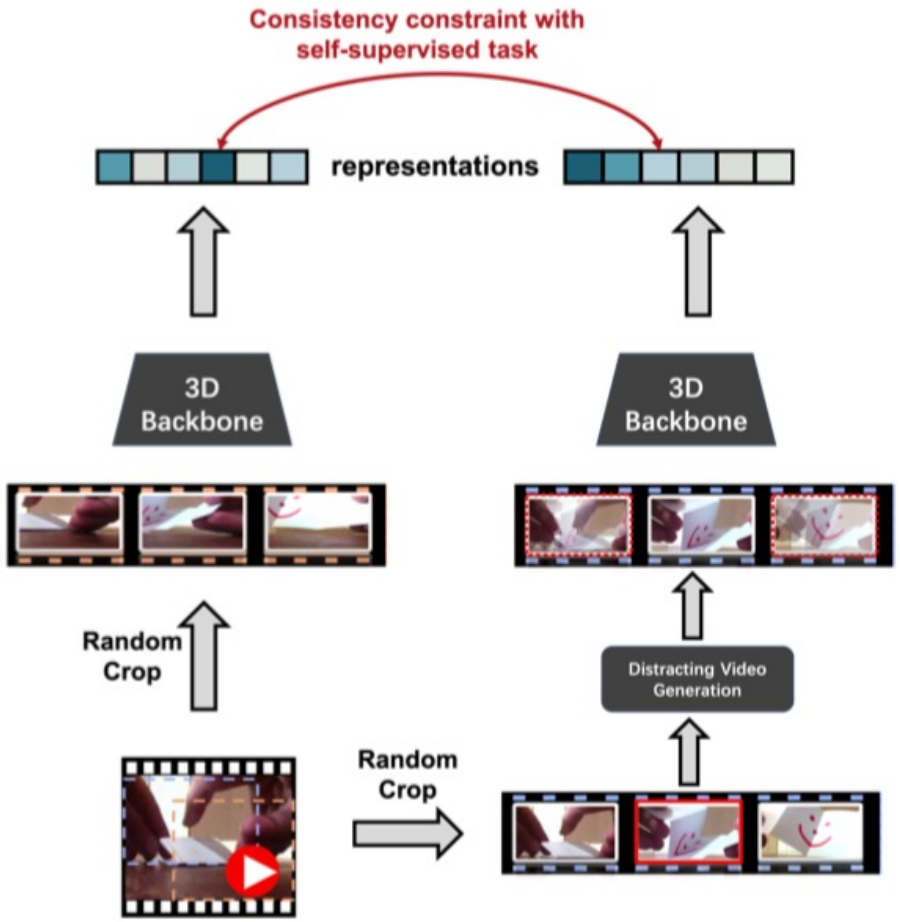
Figure 1: Two video clips of a golf-swing action and their corresponding optical flows. In this example, the flow patterns are very similar across different video instances despite significant variations in RGB space. This observation motivates the idea of co-training, which aims to gradually enhance the representation power of both networks, $f_1(\cdot)$ and $f_2(\cdot)$, by mining hard positives from one another.

Video Classification



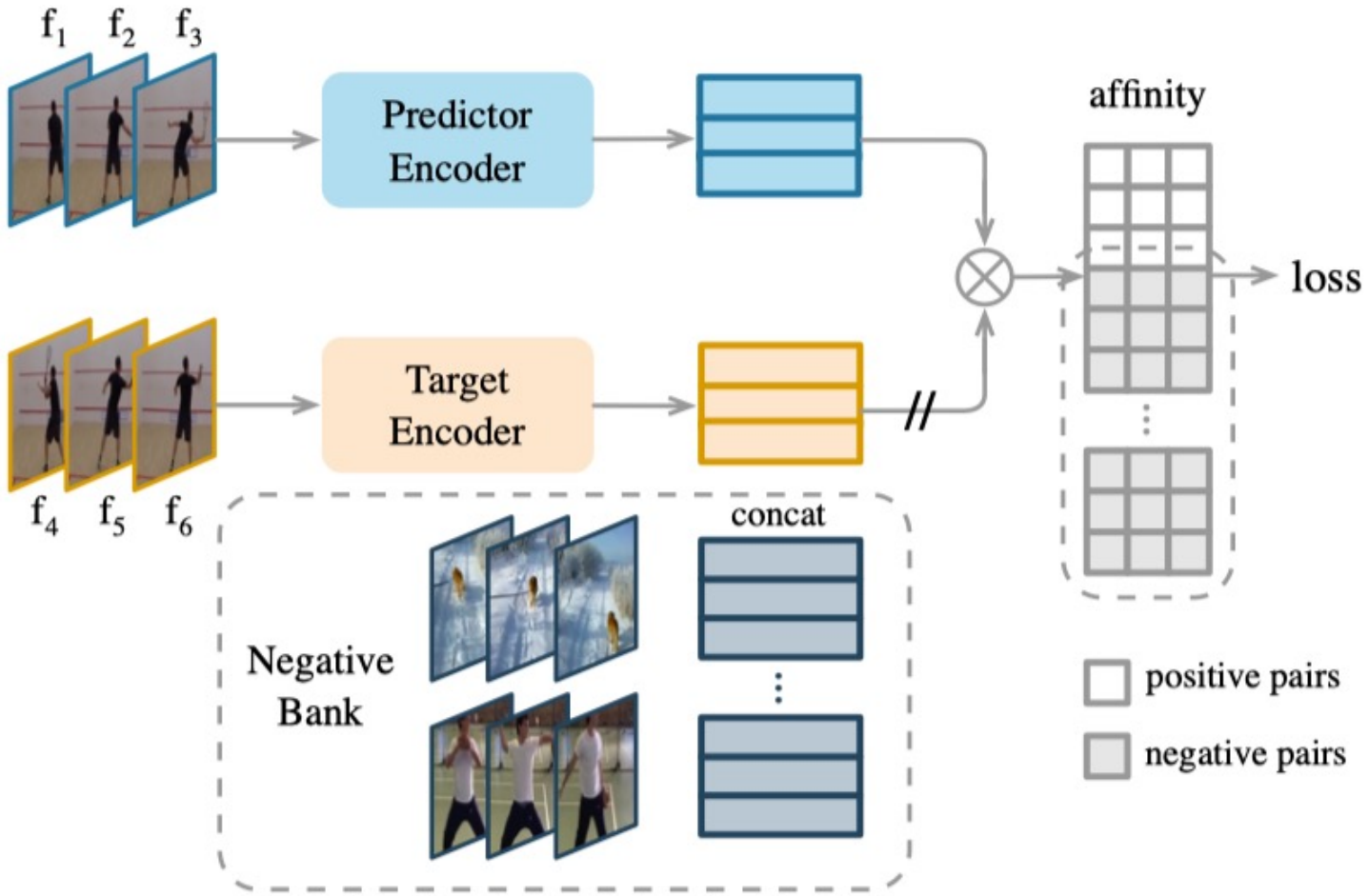
VideoMoCo: Contrastive Video Representation Learning with Temporally Adversarial Examples, CVPR 2021

Video Self-supervised Learning



Removing the Background by Adding the Background: Towards Background Robust Self-supervised Video Representation Learning, CVPR 2021

Object Tracking



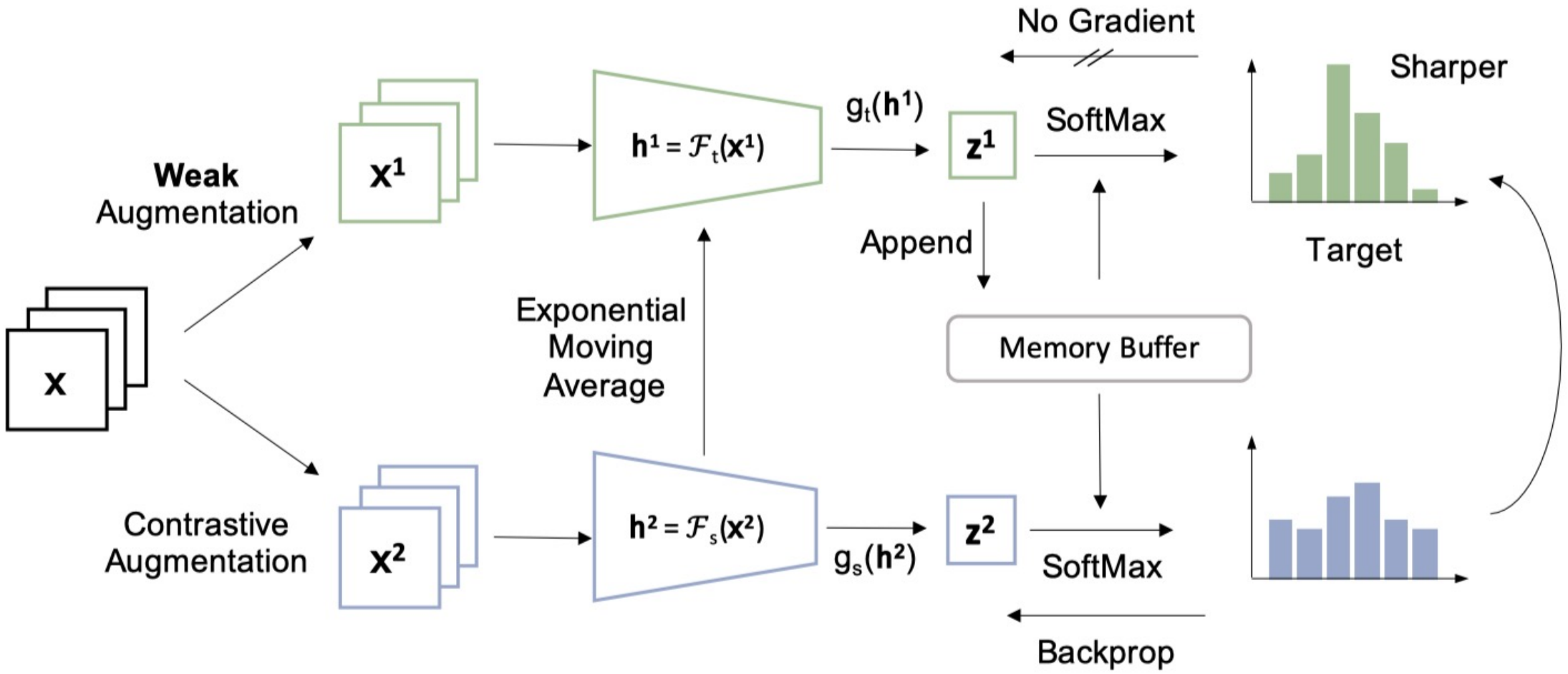
Rethinking Self-supervised Correspondence Learning: A Video Frame-level Similarity Perspective, ICCV 2021

What’s New in Self-Supervised Learning?

ReSSL

- A distribution-based(relation-based) contrastive learning paradigm, similar to SwAV
- SwAV uses multiple prototypes to calculate the similarity distribution, while ReSSL uses other samples of the current batch to calculate the similarity distribution
- Introduce the consistency regularization paradigm in semi-supervised learning to self-supervised learning

ReSSL



ReSSL: Relational Self-Supervised Learning with Weak Augmentation, NeurIPS 2021

Experiments

Table 1: Compare to other SSL algorithms on small and medium dataset.

Method	BackProp	EMA	CIFAR-10	CIFAR-100	STL-10	Tiny ImageNet
Supervised	-	-	94.22	74.66	82.55	59.26
SimCLR [7]	2x	No	84.92	59.28	85.48	44.38
BYOL [23]	2x	Yes	85.82	57.75	87.45	42.70
SimSiam [10]	2x	No	88.51	60.00	87.47	37.04
MoCoV2 [9]	1x	Yes	86.18	59.51	85.88	43.36
ReSSL (Ours)	1x	Yes	**90.20**	**63.79**	**88.25**	**46.60**

Consistency Regularization is Very Important

Table 2: Effect of different τ_t and τ_s for ReSSL

Dataset	τ_s	$\tau_t = 0.01$	$\tau_t = 0.02$	$\tau_t = 0.03$	$\tau_t = 0.04$	$\tau_t = 0.05$	$\tau_t = 0.06$	$\tau_t = 0.07$
CIFAR-10	0.1	89.35	89.74	90.09	90.04	**90.20**	90.18	88.67
CIFAR-10	0.2	89.52	89.67	89.24	89.50	89.22	89.40	89.50
CIFAR-100	0.1	62.34	62.79	62.71	**63.79**	63.46	63.20	61.31
CIFAR-100	0.2	60.37	60.05	60.24	60.09	59.09	59.12	59.76
STL-10	0.1	86.65	86.96	87.16	87.32	**88.25**	87.83	87.08
STL-10	0.2	85.17	86.12	85.01	85.67	85.21	85.51	85.28
Tiny ImageNet	0.1	45.20	45.40	46.30	**46.60**	45.08	45.24	44.18
Tiny ImageNet	0.2	43.28	42.98	43.58	42.12	42.70	42.76	42.60

Table 3: Effect of weak augmentation guided ReSSL

Teacher Aug	Student Aug	CIFAR-10	CIFAR-100	STL-10	Tiny ImageNet
Contrastive	Contrastive	86.17	57.60	84.71	40.38
Weak	Contrastive	**90.20**	**63.79**	**88.25**	**46.60**

Decoupled Contrastive Learning

- InfoNCE based contrastive learning can be seen as a coupling of two tasks, i.e., pull the positive closer and push the negative apart.
- The negative-positive-coupling (NPC) effect, leading to unsuitable learning efficiency with respect to the batch size.
- By decoupling the NPC effect, we can significantly improving SSL efficiency.

Decoupled Contrastive Learning, arXiv 2021

Negative-Positive-Coupling Effect

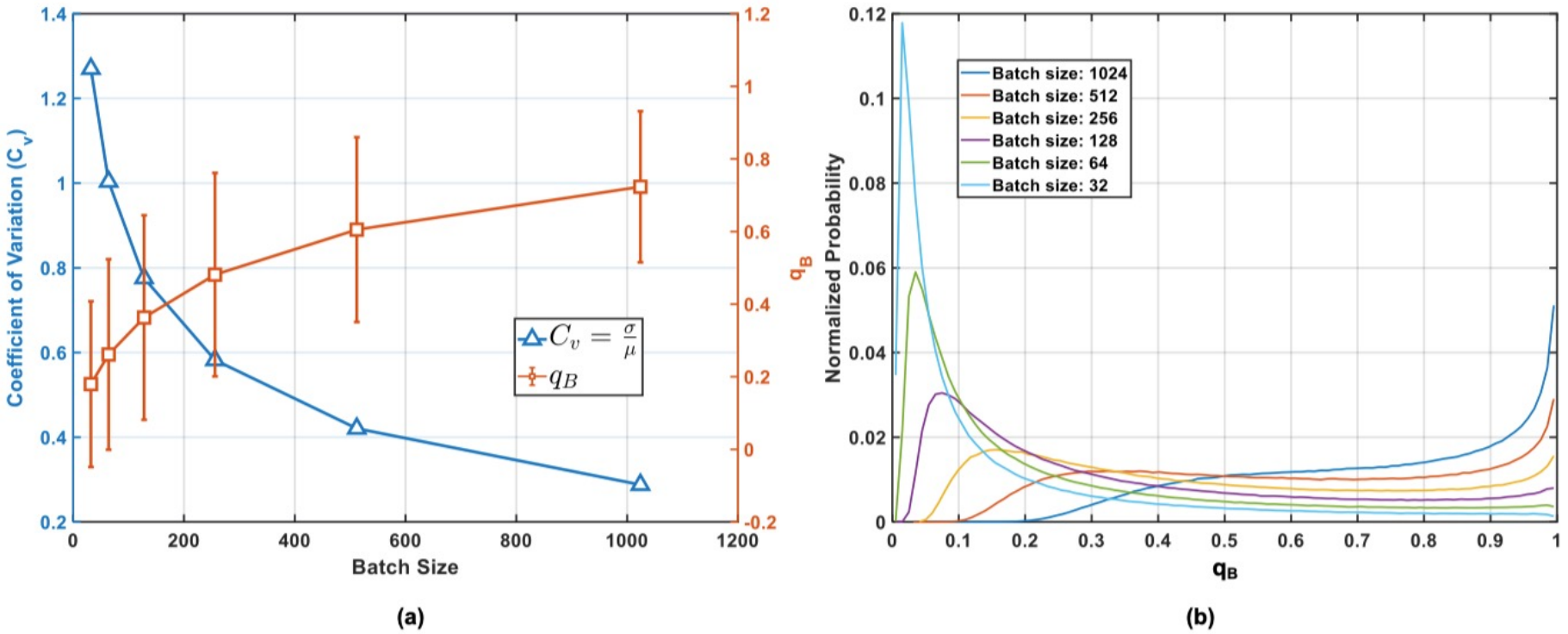


Figure 1: An overview of the batch size issue in the general contrastive approaches: (a) shows the NPC multiplier q_B in different batch sizes. As the large batch size increasing the q_B will approach 1 with a small coefficient of variation ($C_v = \sigma/\mu$). (b) illustrates the distribution of q_B.

Decoupled Contrastive Learning Loss

- InfoNCE

$$L_i^{(k)} = -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)}{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \sum_{l \in \{1,2\}, j \in [\![1,N]\!], j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)}$$

- Decoupled Contrastive Learning Loss

$$L_{DC,i}^{(k)} = -\log \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)}{\cancel{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)} + \sum_{l \in \{1,2\}, j \in [\![1,N]\!], j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)}$$

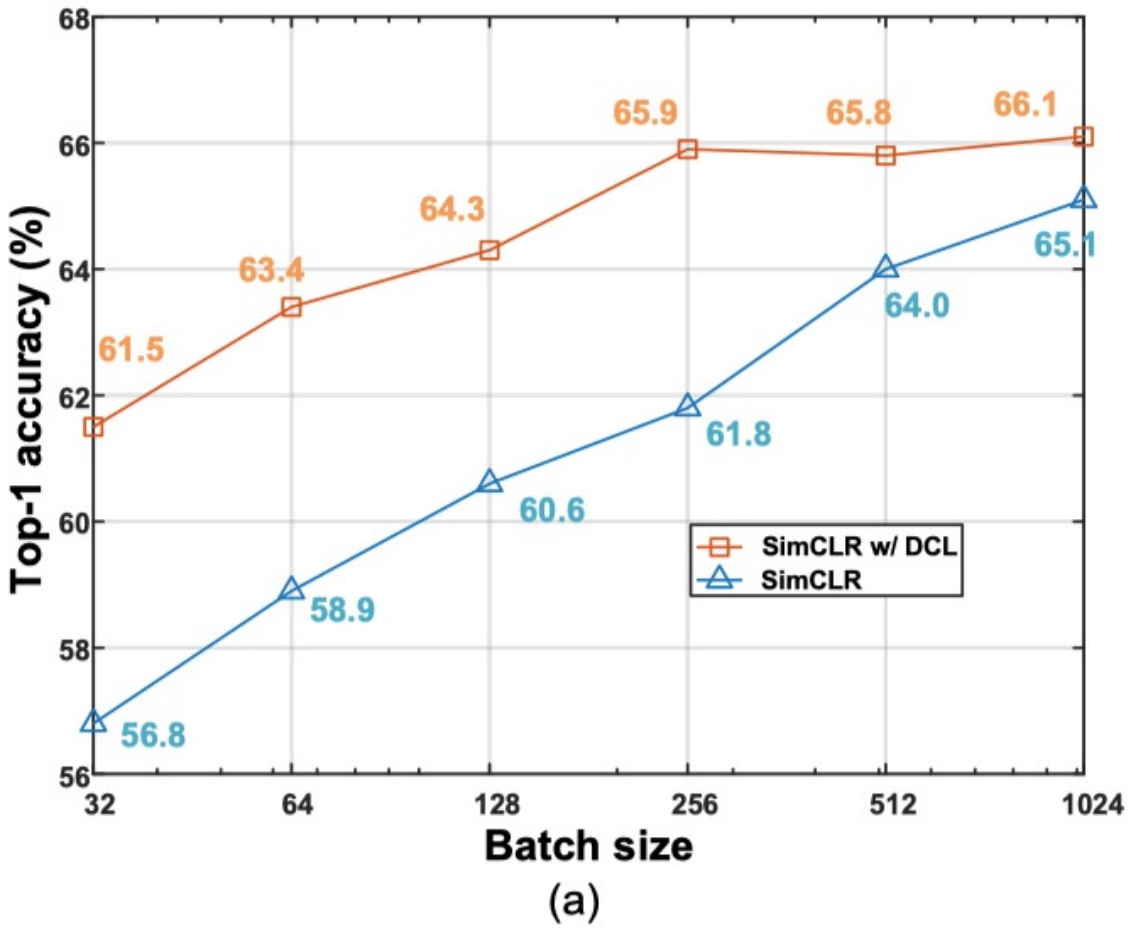
$$= -\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau + \log \sum_{l \in \{1,2\}, j \in [\![1,N]\!], j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)$$

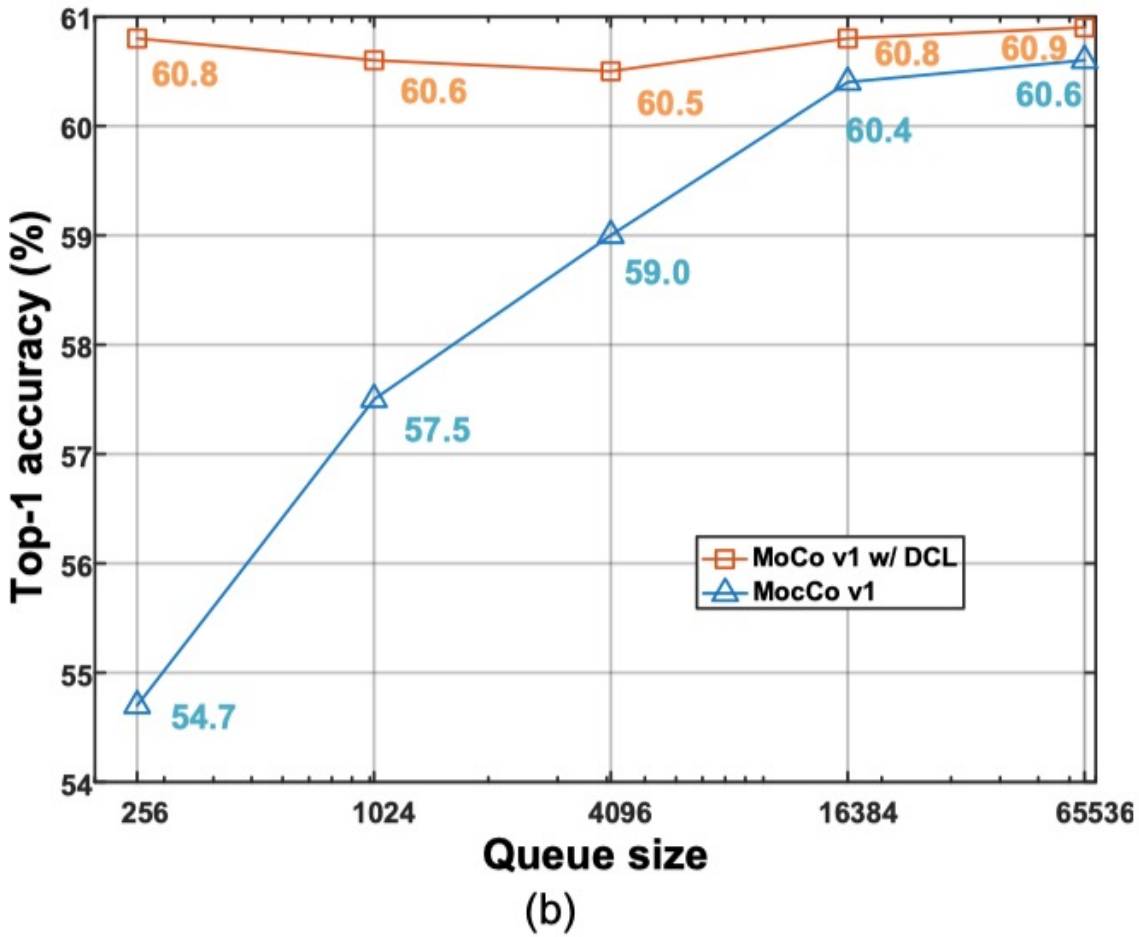
- Weighted DCL Loss

$$L_{DCW,i}^{(k)} = -w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \log \sum_{l \in \{1,2\}, j \in [\![1,N]\!], j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)$$

Decoupled Contrastive Learning, arXiv 2021

Experiments



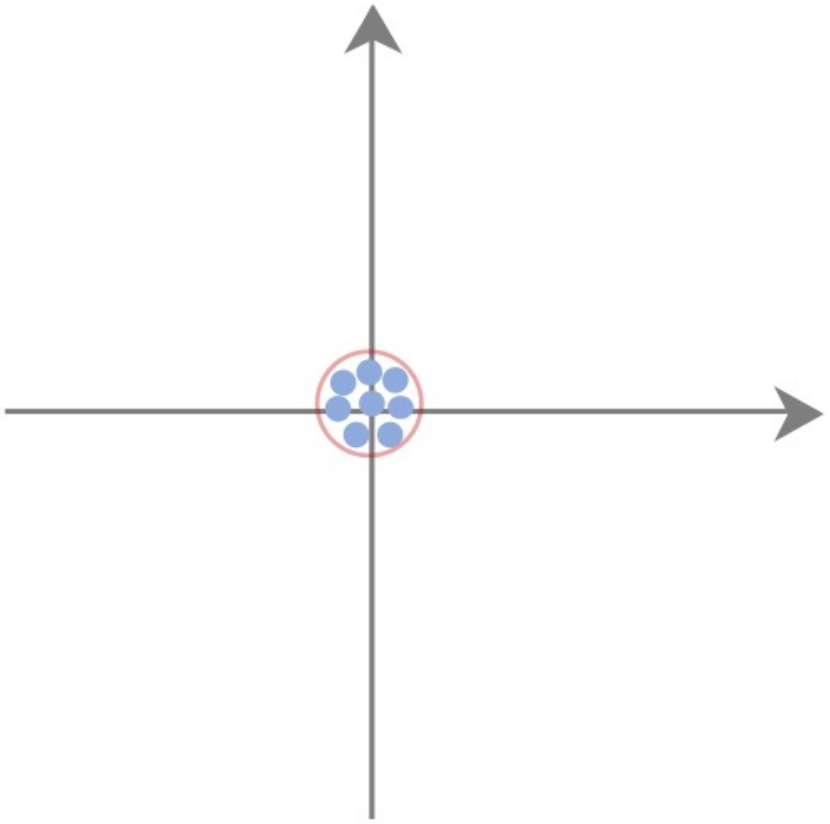


Experiments

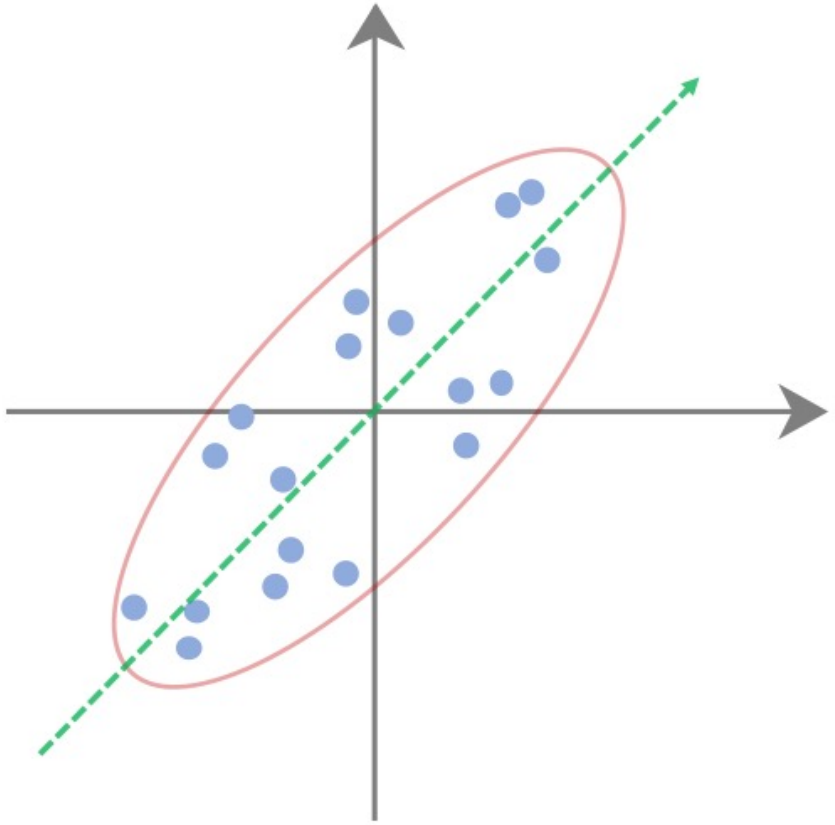
Table 1: Comparisons with/without DCL under different numbers of batch sizes from 32 to 512. Results show the effectness of DCL on four widely used benchmarks. The performance of DCL keeps steadier than the SimCLR baseline while the batch size is varied.

Architecture@epoch	ResNet-18@200 epoch									
Dataset	ImageNet-100 (linear)					STL10 (kNN)				
Batch Size	32	64	128	256	512	32	64	128	256	512
SimCLR	74.2	77.6	79.3	80.7	81.3	74.1	77.6	79.3	80.7	81.3
SimCLR w/ DCL	**80.8**	**82.0**	**81.9**	**83.1**	**82.8**	**82.0**	**82.8**	**81.8**	**81.2**	**81.0**
Dataset	CIFAR10 (kNN)					CIFAR100 (kNN)				
Batch Size	32	64	128	256	512	32	64	128	256	512
SimCLR	78.9	80.4	81.1	81.4	81.3	49.4	50.3	51.8	52.0	52.4
SimCLR w/ DCL	**83.7**	**84.4**	**84.4**	**84.2**	**83.5**	**51.1**	**54.3**	**54.6**	**54.9**	**55.0**
Architecture@epoch	ResNet-50@500 epoch									
SimCLR	82.2	-	88.5	-	89.1	49.8	-	59.9	-	61.1
SimCLR w/ DCL	**86.1**	-	**89.9**	-	**90.3**	**54.3**	-	**61.6**	-	**62.2**

Multiple Collapses of SSL



(a) complete collapse



(b) dimensional collapse

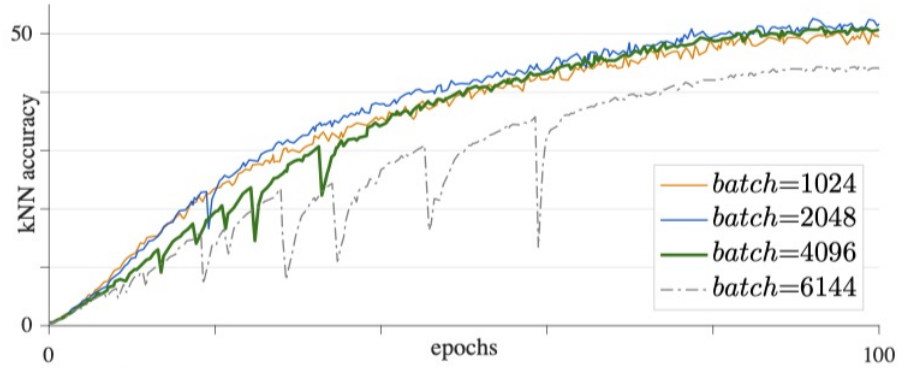
On Feature Decorrelation in Self-Supervised Learning, ICCV 2021

SSL for Visual Transformers

MoCo v3

- Study the self-supervised learning of visual transformers (ViT).
- Training instability that caused by patch projection (first layer in ViT) is a major issue that degrades accuracy, and it can be hidden by apparently good results.
- A fixed random patch projection is proposed to make training more stable and achieve higher accuracy.

The Instability of Training



batch	1024	2048	4096	6144
linear acc.	71.5	72.6	72.2	69.7

Figure 1. **Training curves of different batch sizes** (MoCo v3, ViT-B/16, 100-epoch ImageNet, AdamW, lr=1.0e-4).

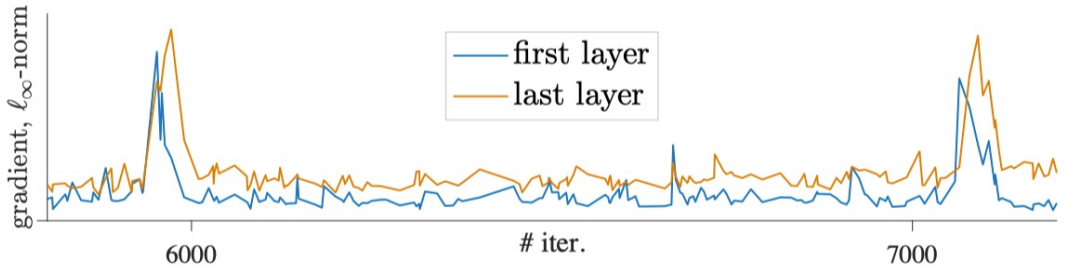
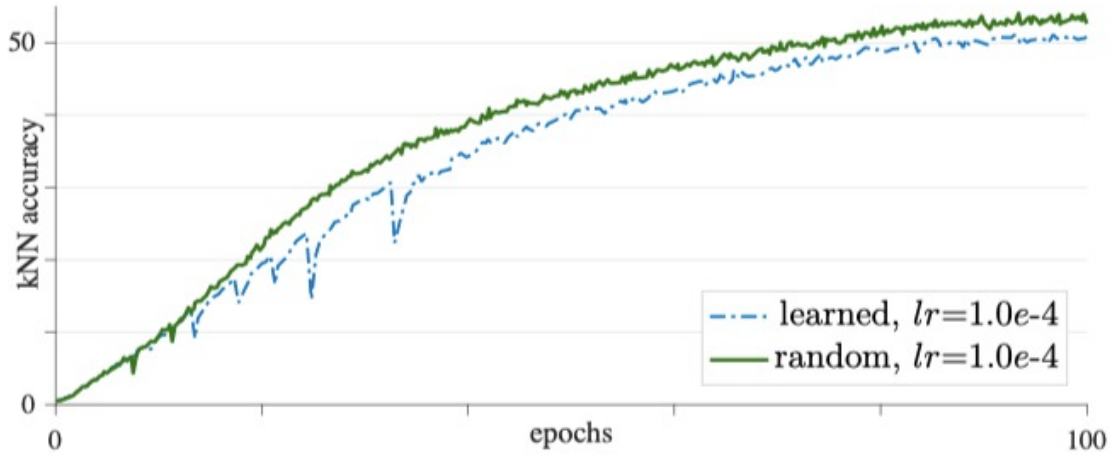
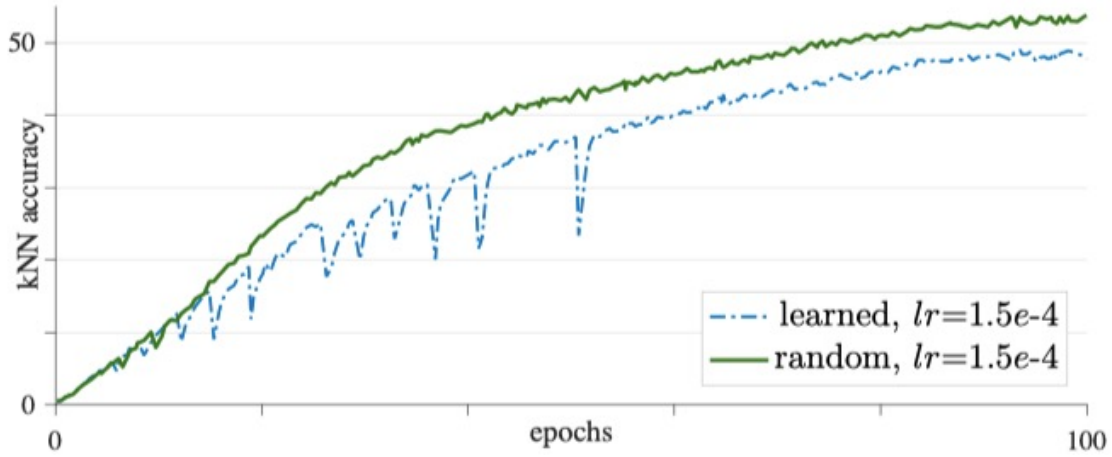
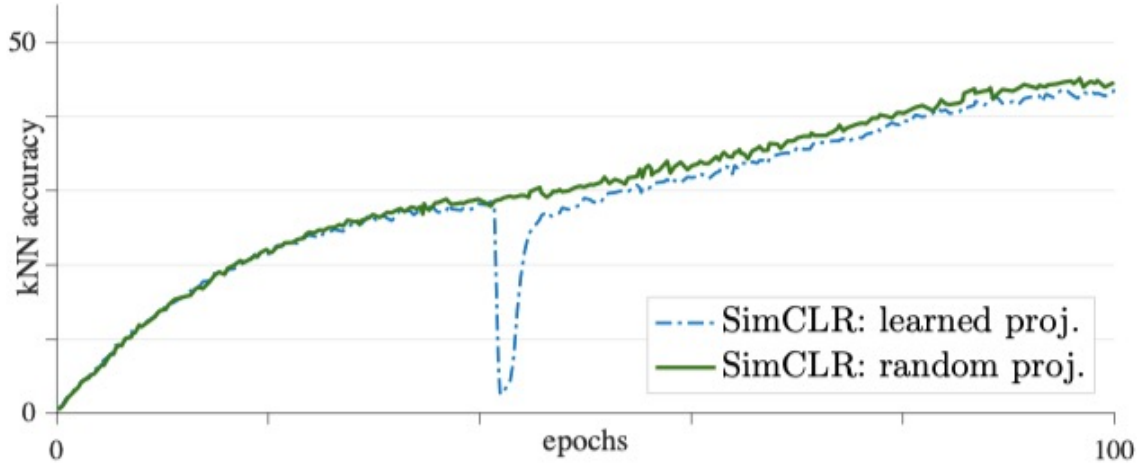


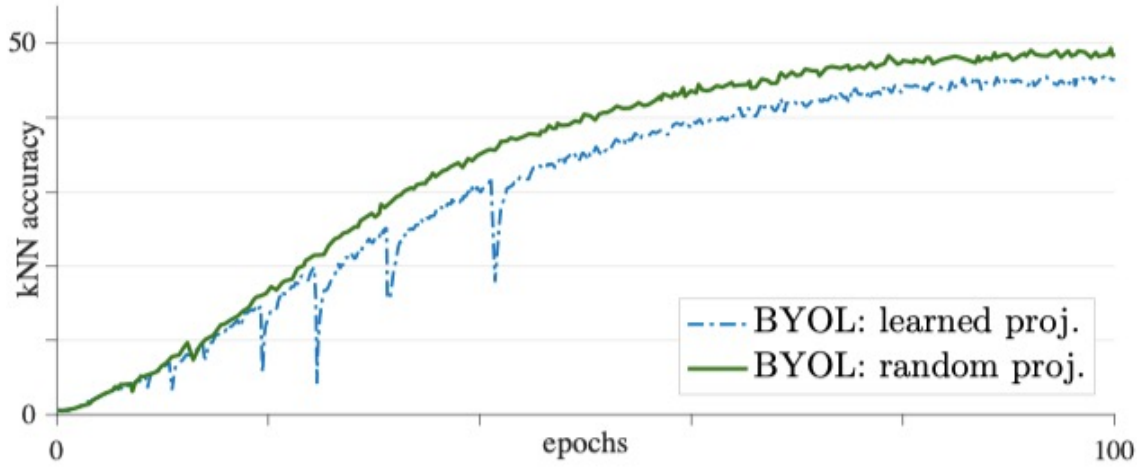
Figure 4. We monitor the gradient magnitude, shown as relative values for the layer. A "spike" in the gradient causes a "dip" in the training curve. We observe that a spike happens earlier in the first layer, and are delayed by tens of iterations in the last layers.

Fixed Random Patch Projection Lear to Higher Accuracy and More Stable





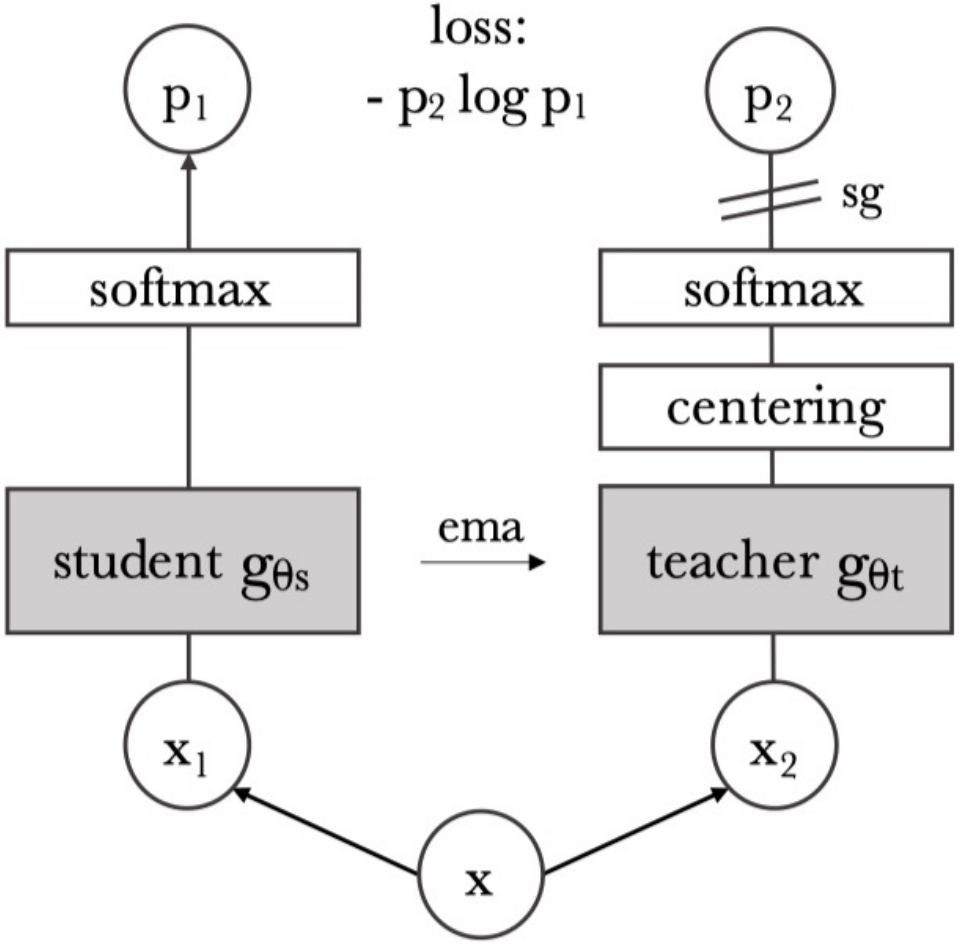




DINO

- Study the difference of self-supervised learning between ViT and convolution.
- Self-supervised ViT features contain explicit information about the semantic segmentation of an image, which does not emerge as clearly with supervised ViTs, nor with convnets.
- Features learned by self-supervised ViT are also excellent k-NN classifiers.

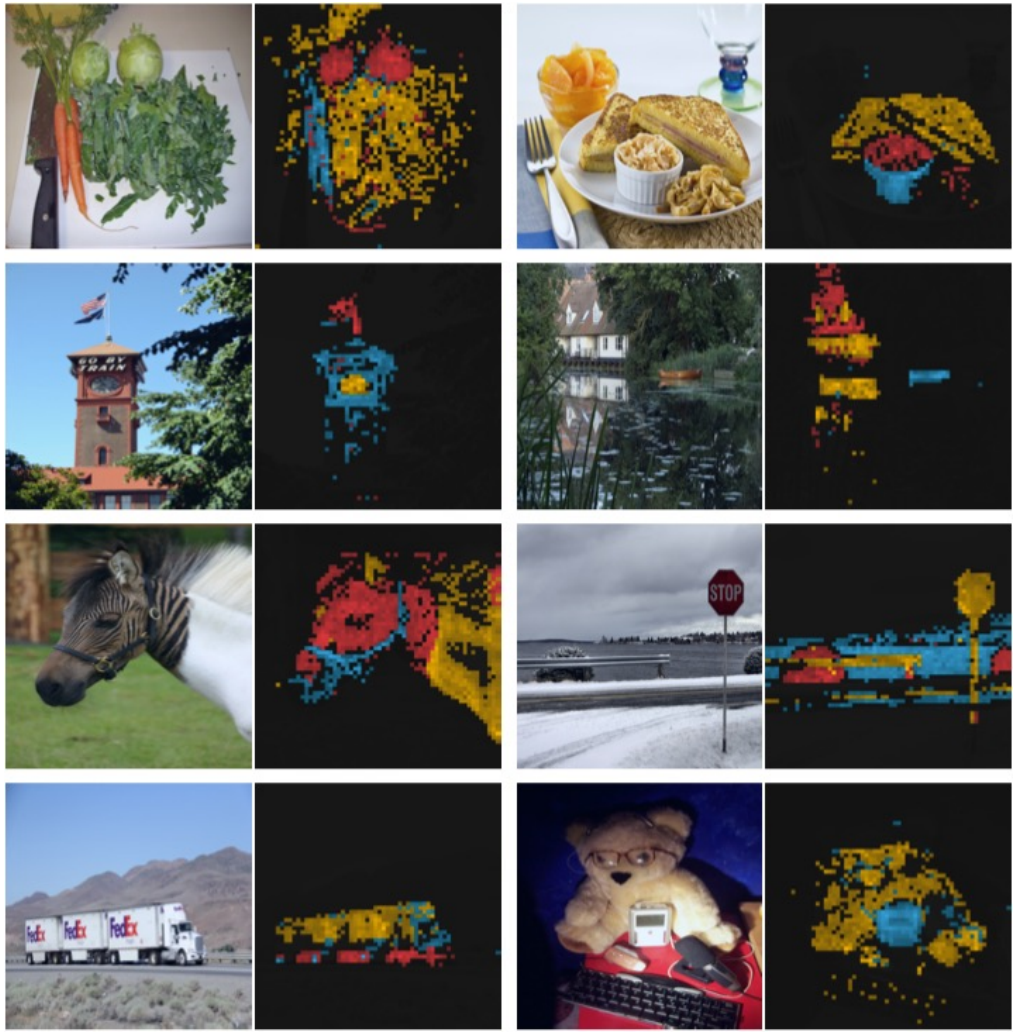
Method

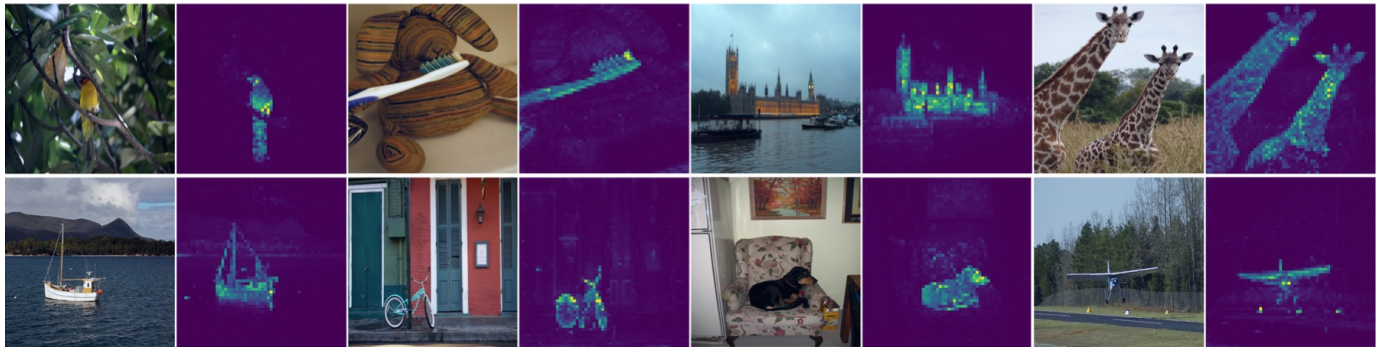


Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# 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()
```

Attention Maps





Emerging Properties in Self-Supervised Vision Transformers, arXiv 2021

Experiments

Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	**75.3**	65.7
DINO	RN50	23	1237	**75.3**	**67.5**
Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	**77.0**	**74.5**

	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

SK: Sinkhorn-Knopp, MC: Multi-Crop, Pred.: Predictor
CE: Cross-Entropy, MSE: Mean Square Error, INCE: InfoNCE

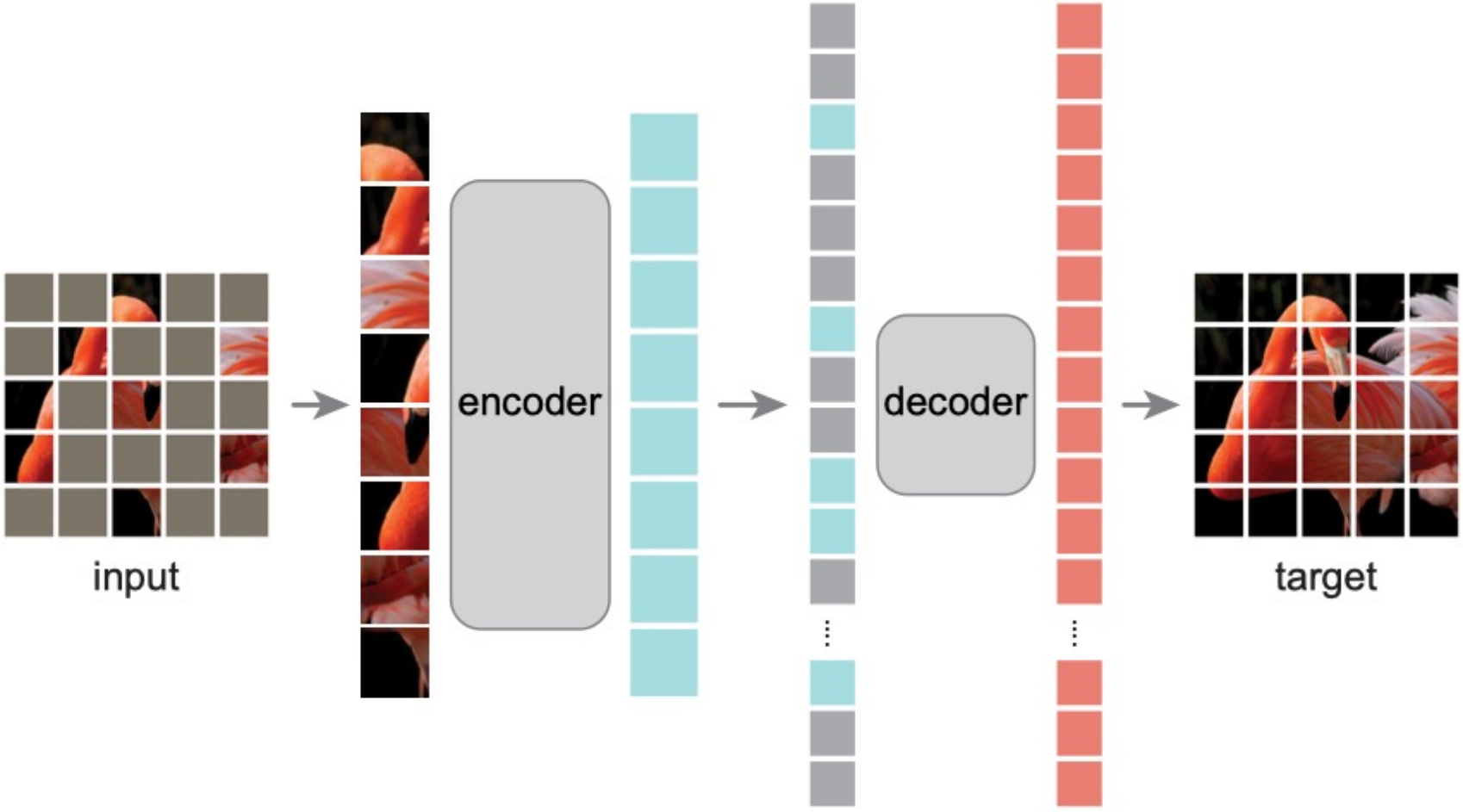
Generative SSL in Computer Vision

Masked AutoEncoder

- A generative SSL paradigm that can surpass contrastive SSL in computer vision domain.
- A asymmetric encoder-decoder that allows the encoder to operate only on the partial, observed signal (without mask tokens) and a lightweight decoder that reconstructs the full signal from the latent representation and mask tokens.
- Masking a high proportion of the input image to learn semantic reconstruction instead of pixel reconstruction.

Masked Autoencoders Are Scalable Vision Learners, arXiv 2021

MAE Pipeline



Masked Autoencoders Are Scalable Vision Learners, arXiv 2021

MAE

- MAE Encoder
  - A ViT but applied only on visible, unmasked patches.
- MAE Decoder
  - The input is the full set of tokens with positional embeddings.
  - A light-weight network architecture than the encoder.
- MAE Reconstruction Target
  - MAE reconstructs the input by predicting the normalized pixel values for each masked patch with MSE loss.

High Masking Ratio is Important

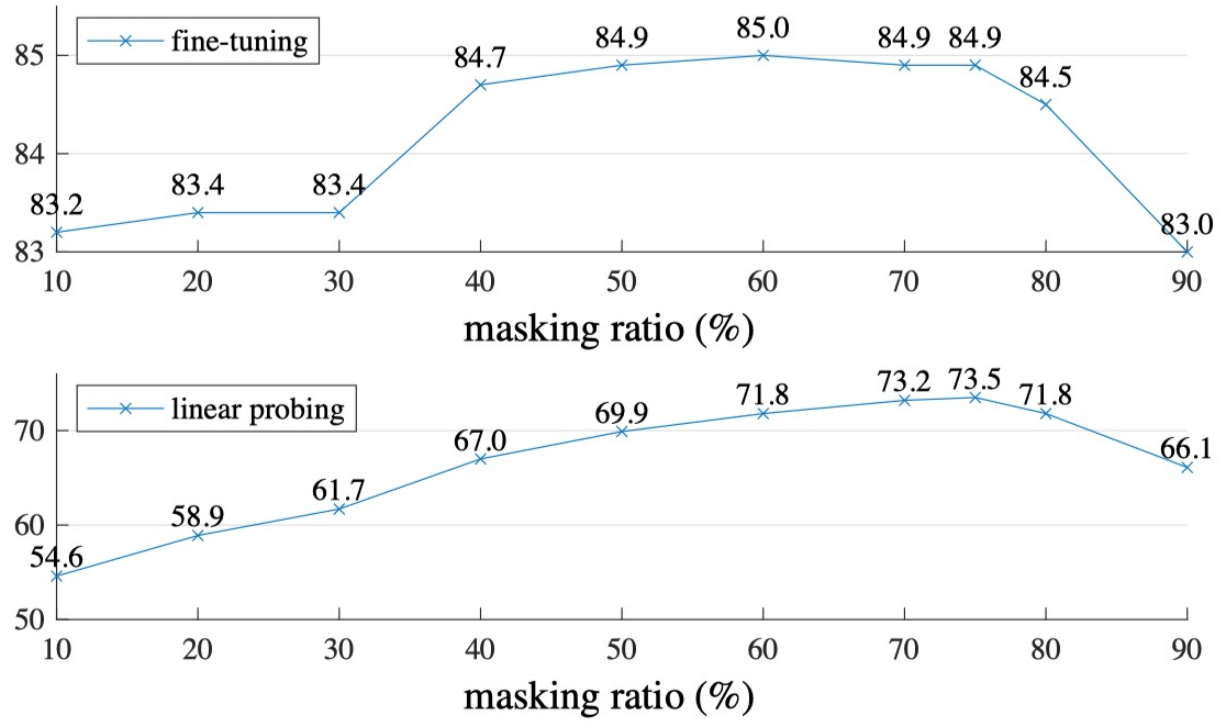


Figure 5. **Masking ratio**. A high masking ratio (75%) works well for both fine-tuning (top) and linear probing (bottom). The y-axes are ImageNet-1K validation accuracy (%) in all plots in this paper.

Ablation Studies

blocks	ft	lin
1	84.8	65.5
2	**84.9**	70.0
4	**84.9**	71.9
8	**84.9**	**73.5**
12	84.4	73.3

(a) **Decoder depth**. A deep decoder can improve linear probing accuracy.

dim	ft	lin
128	**84.9**	69.1
256	84.8	71.3
512	**84.9**	**73.5**
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width**. The decoder can be narrower than the encoder (1024-d).

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	**84.9**	**73.5**	**1**×

(c) **Mask token**. An encoder without mask tokens is more accurate and faster (Table 2).

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	**85.4**	**73.9**
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target**. Pixels as reconstruction targets are effective.

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

(e) **Data augmentation**. Our MAE works with minimal or no augmentation.

case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

(f) **Mask sampling**. Random sampling works the best. See Figure 6 for visualizations.

Comparisons with Previous Results

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H$_{448}$
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	83.6	85.9	86.9	**87.8**

Table 3. **Comparisons with previous results on ImageNet-1K**. The pre-training data is the ImageNet-1K training set (except the tokenizer in BEiT was pre-trained on 250M DALLE data [43]). All self-supervised methods are evaluated by end-to-end fine-tuning. The ViT models are B/16, L/16, H/14 [16]. The best for each column is underlined. All results are on an image size of 224, except for ViT-H with an extra result on 448. Here our MAE reconstructs normalized pixels and is pre-trained for 1600 epochs.

		AP^{box}		AP^{mask}	
method	pre-train data	ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	**53.3**	44.4	47.1
MAE	IN1K	**50.3**	**53.3**	**44.9**	**47.2**

Table 4. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline. All entries are based on our implementation. Self-supervised entries use IN1K data *without* labels. Mask AP follows a similar trend as box AP.

method	pre-train data	ViT-B	ViT-L
supervised	IN1K w/ labels	47.4	49.9
MoCo v3	IN1K	47.3	49.1
BEiT	IN1K+DALLE	47.1	53.3
MAE	IN1K	**48.1**	**53.6**

Table 5. **ADE20K semantic segmentation** (mIoU) using UperNet. BEiT results are reproduced using the official code. Other entries are based on our implementation. Self-supervised entries use IN1K data *without* labels.