Self-Supervised Learning

2021 연구실 하계 세미나

김 기 남

Vision & Display Systems Lab.

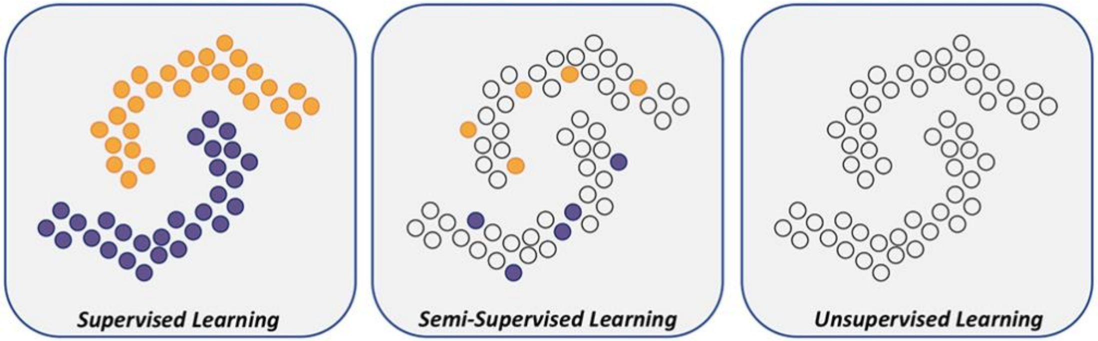
Dept. of Electronic Engineering, Sogang University

Outline

- What is Self-Supervised Learning?
- Why we need Self-Supervised Learning?
- Trend of the self-supervised learning
- Bootstrap your own latent
- Exploring Simple Siamese Representation Learning
- References

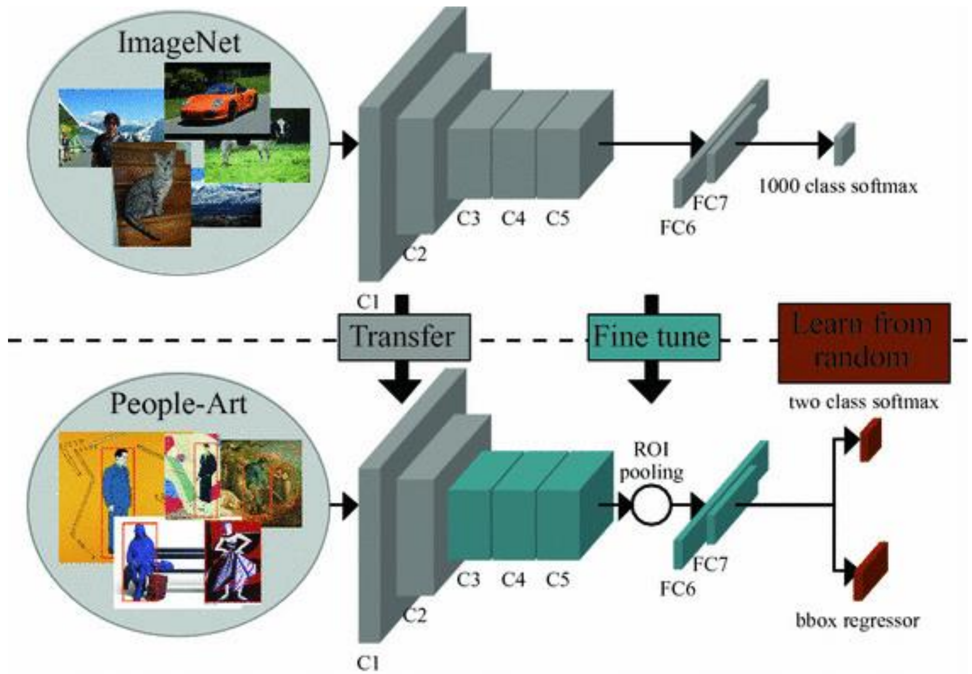
Self-Supervised Learning

- What is Self-Supervised Learning?
  - Supervised Learning
    - 모든 학습 데이터에 label이 존재
  - Semi-Supervised Learning
    - 일부 학습 데이터에 label이 존재
  - Unsupervised Learning
    - 학습 데이터에 label이 존재하지 않음
    - Self-supervised learning은 unsupervised learning의 일부
      - 스스로 Supervision을 주어서 학습을 진행



Self-Supervised Learning

- Why we need Self-Supervised Learning?
  - Visual Representation
    - 일반적으로 deep learning model 학습 시 pretrained weight 를 사용
      - Computer vision task의 경우 일반적으로 ImageNet pretrained weight를 사용
      - Pretrained weight를 사용했을 때와 사용하지 않았을 때의 성능 차이가 큼

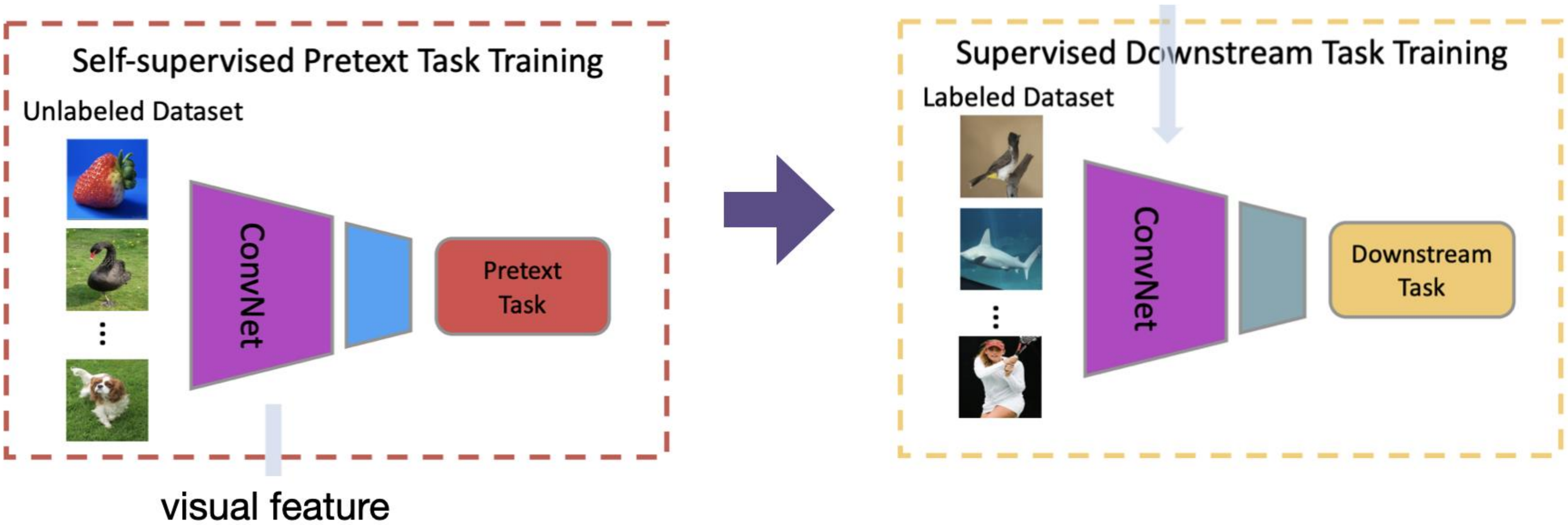


Self-Supervised Learning

- Why we need Self-Supervised Learning?
  - Visual Representation
    - 일반적인 RGB 영상이 아닌 특수한 영상을 사용하는 deep learning model의 경우 ImageNet pretrain이 아닌 해당 data 에 대한 pretrain 후, 학습된 weight 를 사용하는 것이 성능 향상에 유리
      - Pretrained model이 해당 data에 대한 feature extraction을 잘하기 때문
      - Ex) 의료 영상, ToF 영상 등
    - 특수한 data의 경우 양은 많이 존재하나, labeling 되어 있지 않은 경우가 많음
      - Labeling에는 많은 cost가 필요

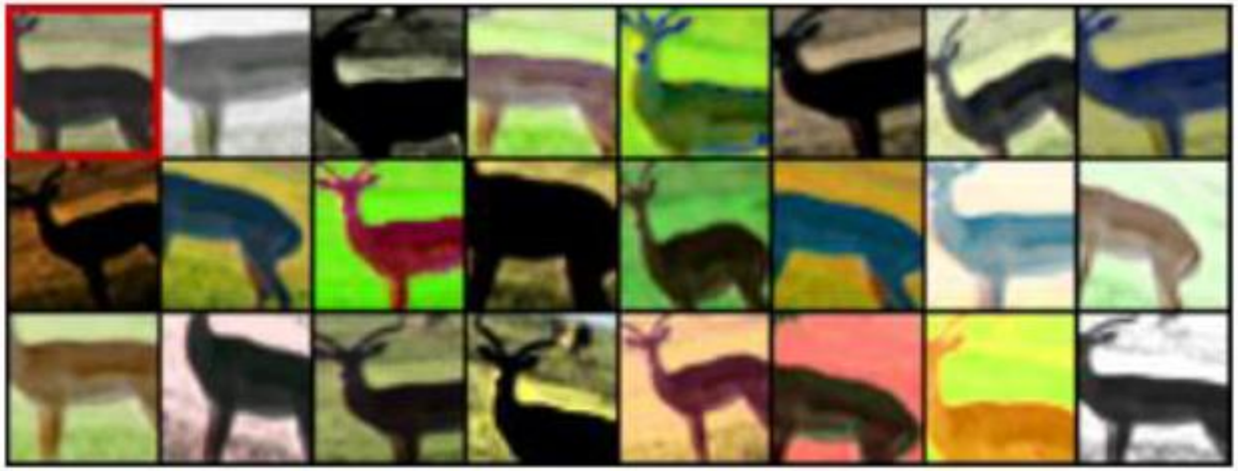
Self-Supervised Learning

- Trend of the self-supervised learning
  - Pretext Tasks (~ 2018)
    - 사용자가 정의한 간단한 문제인 Pretext Tasks 를 통해 visual representation 을 학습한 후, 해당 network 를 실제 사용할 Downstream Task 에 적용하는 방법
    - Supervised learning 으로 학습할 때보다 낮은 성능을 보임
    - Pretext task 를 잘 풀도록 학습되었을 뿐 이미지의 일반적인 시각적 특징을 잡아내지는 못함



Self-Supervised Learning

- Trend of the self-supervised learning
  - Pretext Tasks (~ 2018)
    - Exemplar [1] (NIPS 2014)
      - Image 내에 object 가 존재하는 영역 하나를 patch 로 crop 하여 augmentation
      - Augmented patches 는 전부 같은 class 로 판단하도록 학습
        - ✓ Image 수가 class 수가 되기 때문에 과도한 class 수가 존재
        - ✓ 학습 시 많은 memory 필요

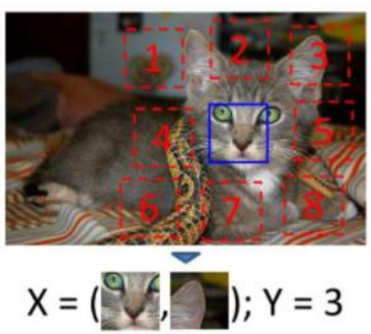


서강대학교 SOGANG UNIVERSITY

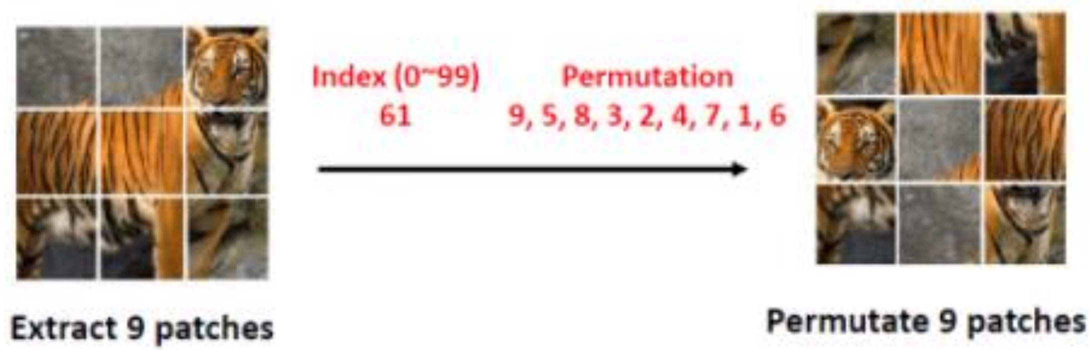
VDS LAB

Self-Supervised Learning

- Trend of the self-supervised learning
  - Pretext Tasks (~ 2018)
    - Context Prediction [2] (ICCV 2015)
      - 9개의 patch 를 가운데 patch 를 기준으로 나머지 patch 의 상대적 위치를 classification 하도록 학습
    - Jigsaw Puzzle [3] (ECCV 2016)
      - 9개의 patch 의 순서를 섞은 후, 원래의 순서를 찾아가도록 학습



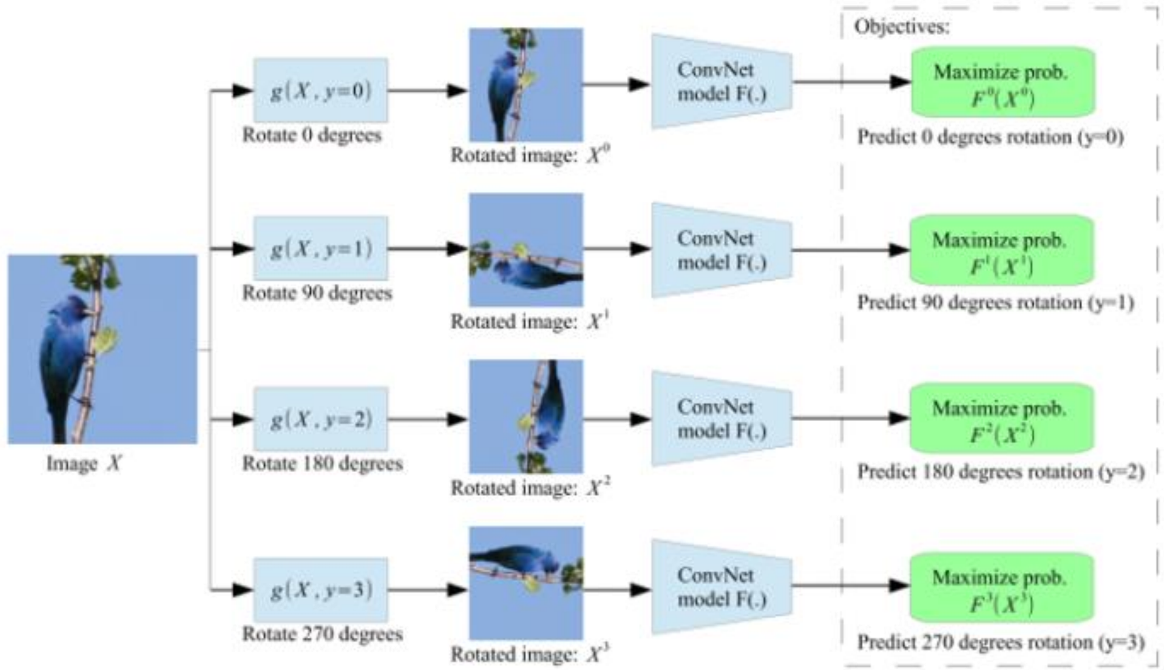
< Context Prediction >



< Jigsaw Puzzle >

Self-Supervised Learning

- Trend of the self-supervised learning
  - Pretext Tasks (~ 2018)
    - Rotation [4] (ICLR 2018)
      - Rotation 된 image가 원래 image에서 얼마나 rotation 되었는지를 classification 하도록 함.

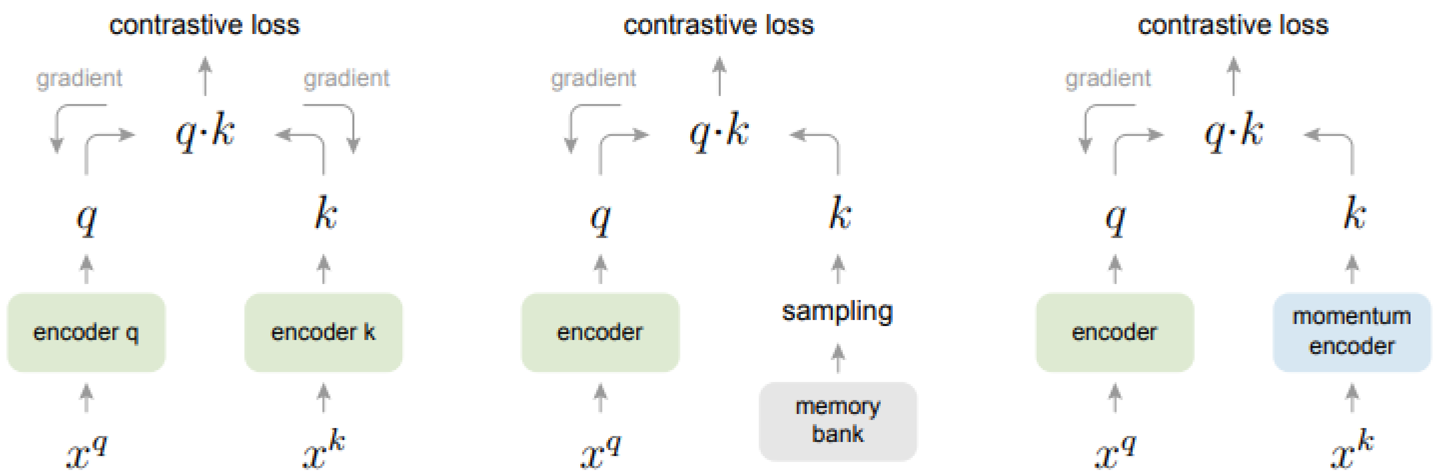


Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - Pretext task 를 정의하지 않고, 같은 image 에 서로 다른 augmentation 을 적용한 positive pair feature representation 는 서로 가깝게, 다른 image 에 다른 augmentation 을 적용한 negative pair 의 feature representation 은 서로 멀어지게 학습하는 것
    - Supervised ImageNet pretrained model 을 사용할 때와 비슷하거나 오히려 높은 성능 달성
    - 여러 가지 문제점 존재
      - Collapsing
        - ✓ 입력을 모두 같은 값으로 수렴시키는 문제 발생
        - ✓ 이를 해결하기 위해 학습 시 large batch size (256 ~ 8192) 사용
      - Negative pair를 정의하는 augmentation 을 어떤 것을 사용하는지에 따라 성능 차이가 심함
        - ✓ 적절한 augmentation 선정 필요

Self-Supervised Learning

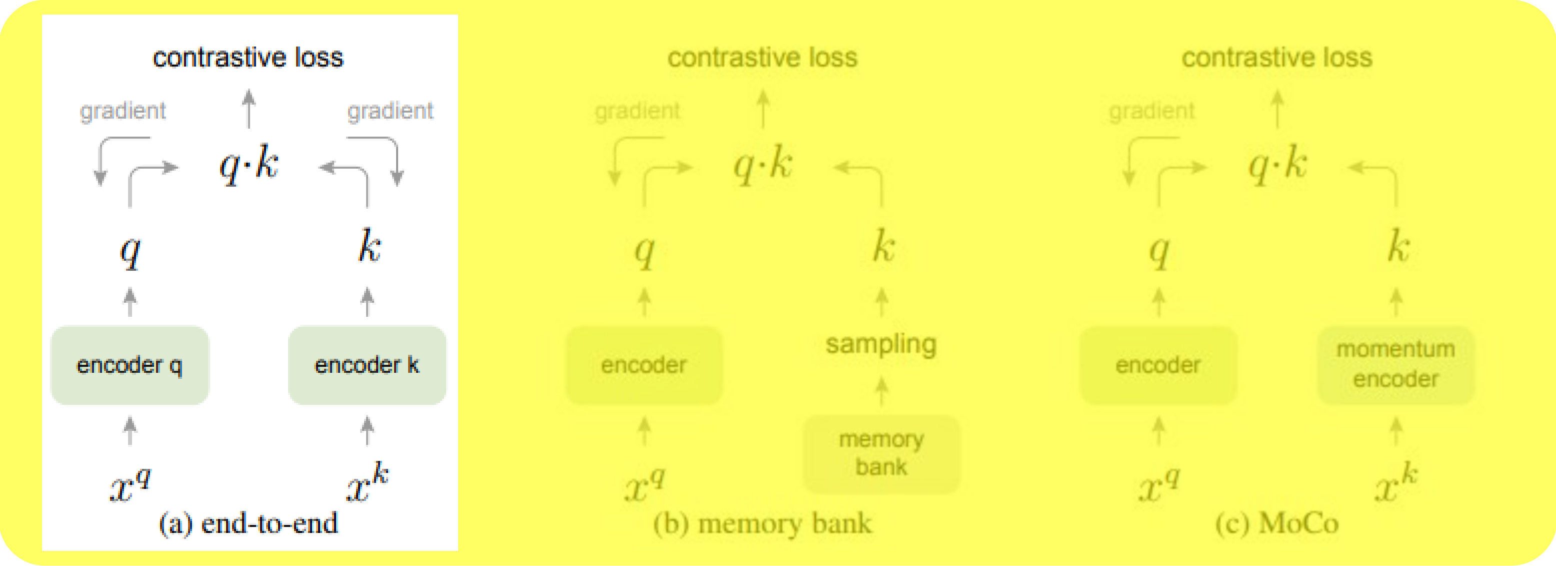
- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - MoCo [5] (CVPR 2020)
      - 기존 방식 중 end-to-end 방식은 large batch size 필요
      - Memory bank 는 inconsistency 발생
      - Momentum encoder 를 사용하여 mini-batch 로 inconsistency 가 발생하지 않는 contrastive learning 방식 제안



(a) end-to-end (b) memory bank (c) MoCo

Self-Supervised Learning

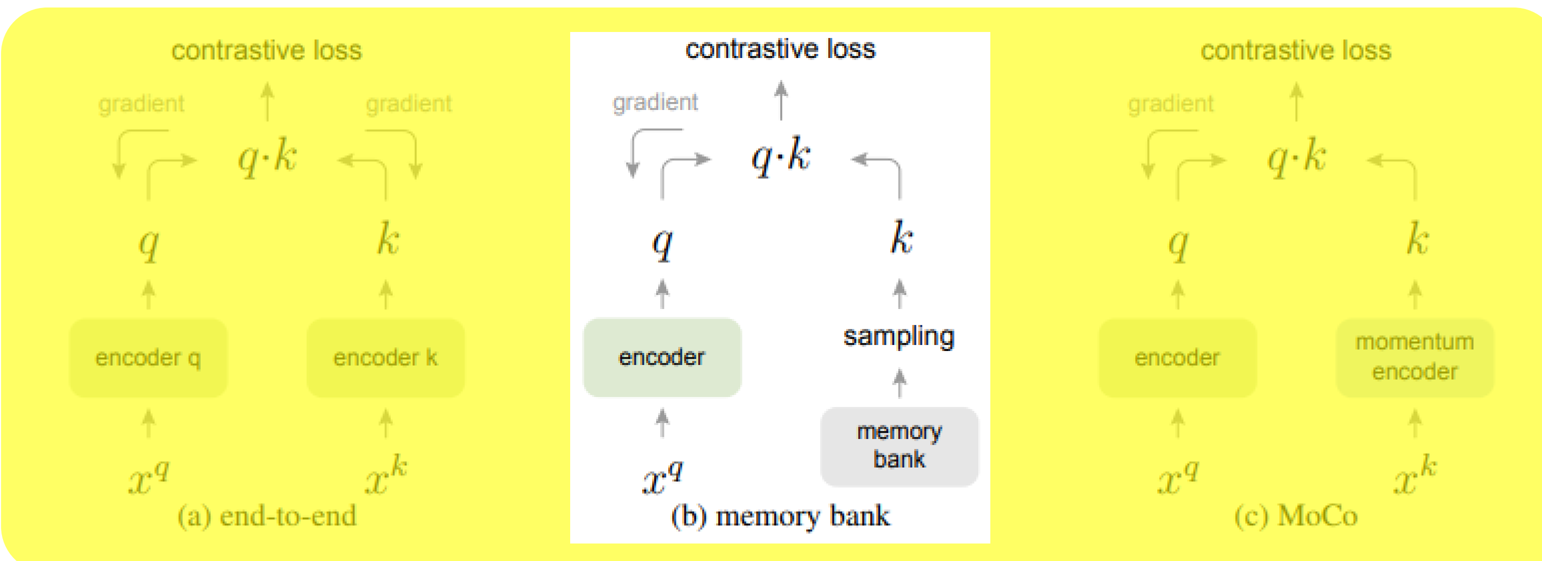
- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - MoCo [5] (CVPR 2020)
      - End-to-end 방식
        - ✓ Batch size 를 크게 가져가 많은 negative sample 을 만들고, 각각의 encoder 를 모두 backpropagation 하는 방식을 채택
          - Batch size 를 크게 가져가야 함 (Memory limit)



(a) end-to-end (b) memory bank (c) MoCo

Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - MoCo [5] (CVPR 2020)
      - Memory bank 방식
        - ✓ 모든 image 에 대해 encoder 를 통과시킨 feature representation 을 추출하고, 해당 vector 를 key 로 하여 memory bank 에 저장 후 sampling 하여 사용
          - Batch size 를 작게 가져갈 수 있으나, inconsistency 발생

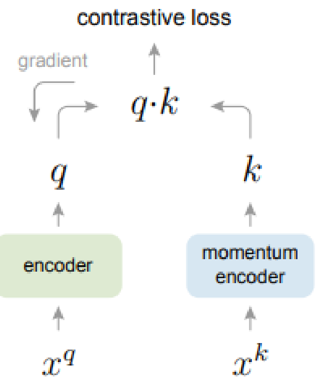


(a) end-to-end (b) memory bank (c) MoCo

Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - MoCo [5] (CVPR 2020)
      - Dynamic dictionary
        - ✓ Mini-batch 보다 큰 size 의 dictionary size 를 지정하고, momentum encoder 를 통과하여 추출된 feature 를 FIFO 방식으로 queuing 하여 사용
      - Momentum encoder
        - ✓ Momentum encoder 는 backpropagation 하지 않음
          - Backpropagation 시 성능이 떨어지는 것을 실험적으로 확인
        - ✓ Encoder 에서 학습된 weight 에 momentum 을 주어 update

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



(c) MoCo

Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - MoCo [5] (CVPR 2020)
      - Downstream task 에서 supervised pretrained model 과 비슷하거나 더 높은 결과를 보임

pre-train	COCO keypoint detection AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
random init.	65.9	86.5	71.7
super. IN-1M	65.8	86.9	71.9
MoCo IN-1M	66.8 (+1.0)	87.4 (+0.5)	72.5 (+0.6)
MoCo IG-1B	66.9 (+1.1)	87.8 (+0.9)	73.0 (+1.1)

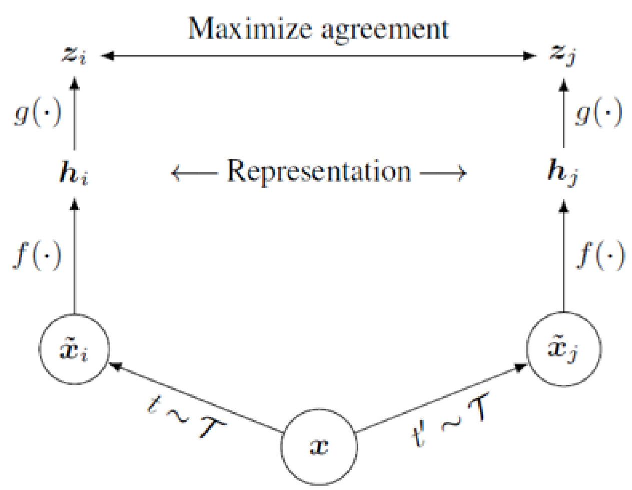
pre-train	LVIS v0.5 instance segmentation AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
random init.	22.5	34.8	23.8
super. IN-1M†	24.4	37.8	25.8
MoCo IN-1M	24.1 (−0.3)	37.4 (−0.4)	25.5 (−0.3)
MoCo IG-1B	24.9 (+0.5)	38.2 (+0.4)	26.4 (+0.6)

pre-train	COCO dense pose estimation AP^{dp}	AP^{dp}_{50}	AP^{dp}_{75}
random init.	39.4	78.5	35.1
super. IN-1M	48.3	85.6	50.6
MoCo IN-1M	50.1 (+1.8)	86.8 (+1.2)	53.9 (+3.3)
MoCo IG-1B	50.6 (+2.3)	87.0 (+1.4)	54.3 (+3.7)

	Cityscapes instance seg.		Semantic seg. (mIoU)	
pre-train	AP^{mk}	AP^{mk}_{50}	Cityscapes	VOC
random init.	25.4	51.1	65.3	39.5
super. IN-1M	32.9	59.6	74.6	74.4
MoCo IN-1M	32.3 (−0.6)	59.3 (−0.3)	75.3 (+0.7)	72.5 (−1.9)
MoCo IG-1B	32.9 (0.0)	60.3 (+0.7)	75.5 (+0.9)	73.6 (−0.8)

Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - SimCLR [6] (ICML 2020)
      - Large batch size 를 사용하는 end-to-end 방식 (batch size 4096 ~ 8192)
      - MoCo 보다 높은 성능 달성
      - Encoder 이후 MLP 로 이루어진 projection head 를 추가
      - Augmentation 이 contrastive learning에 미치는 영향 분석

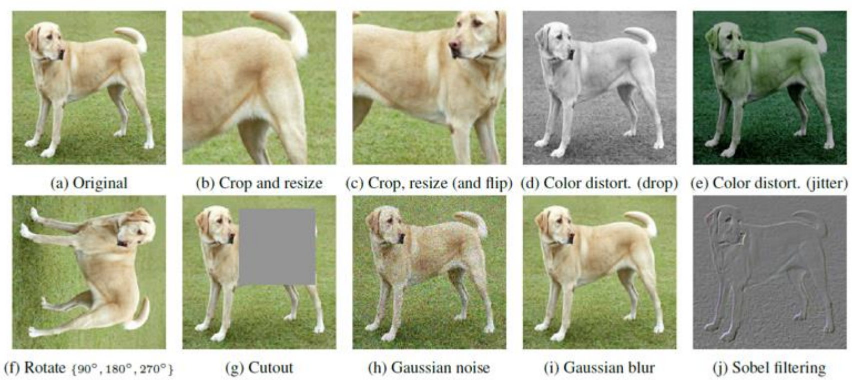


Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - SimCLR [6] (ICML 2020)
      - Large batch size
        - ✓ 4096 를 기본으로 8192 까지 적용
        - ✓ 각 image 마다 2개의 서로 다른 augmentation 을 적용하여 batch size 가 N일 때, 2N 개의 sample 생성
        - ✓ 입력 x 에 대한 positive sample 은 1개, negative sample 은 2N-2개 만큼 생성됨
        - ✓ Batch size 가 클수록 negative sample 개수가 많아져 성능 증가
        - ✓ Large batch 에서 안정적으로 학습이 가능하도록 LARS optimization 적용
        - ✓ Multi- GPU를 이용한 학습 시 각 device 별로 batch normalization 을 진행하지 않고, 모든 device 의 평균, 표준편차를 통합하여 사용
          - Positive sample 이 1개이기 때문에 positive sample 이 존재하는 device 가 1개이기 때문

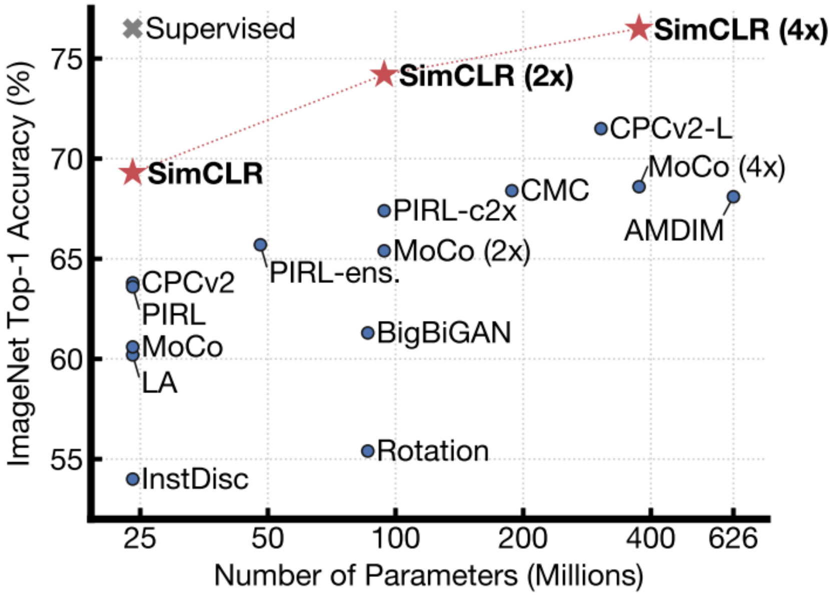
Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - SimCLR [6] (ICML 2020)
      - Augmentation
        - ✓ 2개의 augmentation 을 선택할 때, 어떠한 augmentation 을 사용하는지에 따라 성능 차이가 심함
        - ✓ Crop 과 color distortion 을 선택하였을 때 가장 성능이 높음
          - Color distortion 을 사용하지 않을 경우, augmentation 을 하더라도 augmentation된 영상들 간의 색 배합은 같기 때문에 visual representation 을 찾아내는 것이 아닌 색 배합만을 찾아냄.



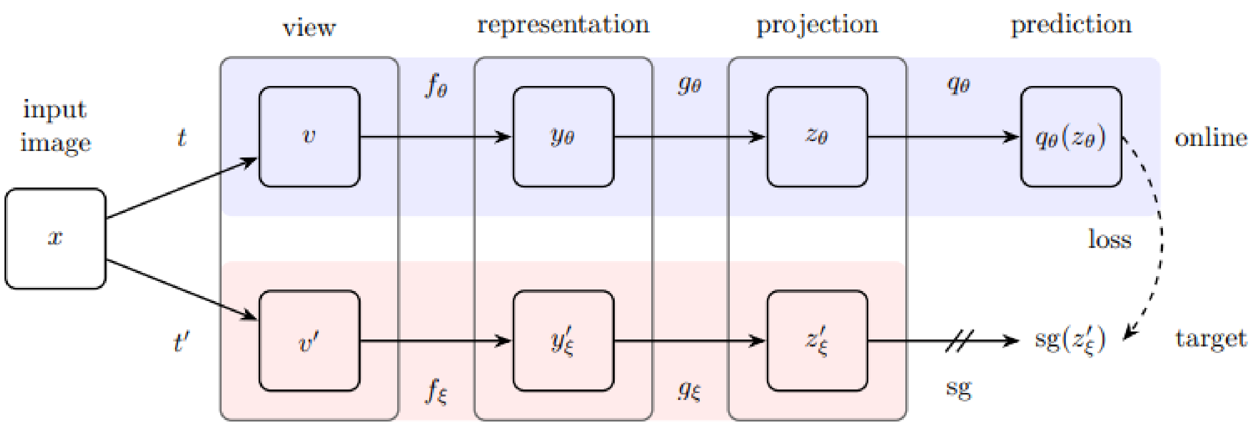
Self-Supervised Learning

- Trend of the self-supervised learning
  - Contrastive Learning (2019 ~ 2020)
    - SimCLR [6] (ICML 2020)
      - MoCo 보다 높은 성능을 보이며, supervised와 근접한 성능을 보임
      - Large batch size 로 인한 memory limit 은 여전히 문제



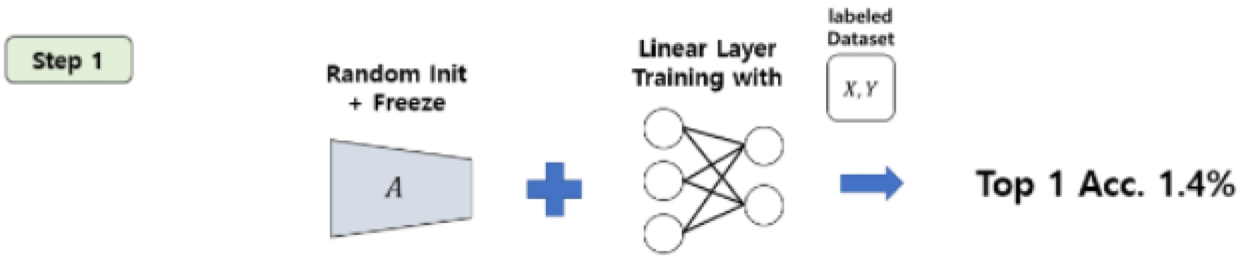
Self-Supervised Learning

- Trend of the self-supervised learning
  - Other (2020 ~)
    - Bootstrap your own latent: A new approach to self-supervised Learning [7] (NIPS 2020)
      - Negative sample 을 사용하지 않는 self-supervised learning 기법 제안
        - ✓ Batch size 를 줄일 수 있음
        - ✓ Negative sample 을 위한 augmentation 종류에 robust 해짐
      - 기존 contrastive learning 과 비슷한 Siamese network 를 이용
      - SOTA 성능 달성



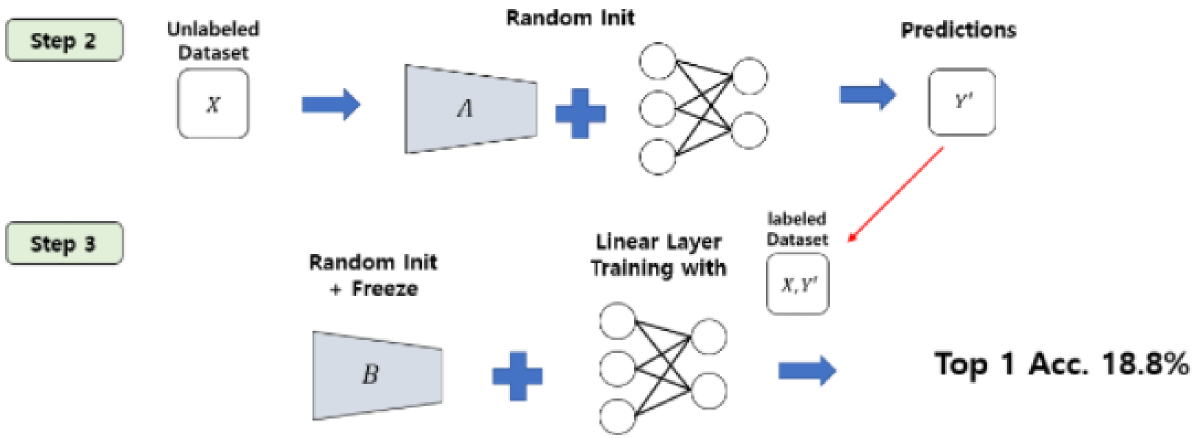
Self-Supervised Learning

- Bootstrap your own latent: A new approach to self-supervised Learning (NIPS 2020)
  - Motivation
    - Negative sample 을 사용하지 않고 image 로부터 visual representation 을 학습하는 것을 목표로 실험을 제안
      - Step 1
        - ✓ Random initialize 된 model 을 학습하지 않고, freeze 한 뒤 linear layer 를 붙여 학습
          - 학습되지 않는 backbone (feature extractor) 를 사용하여 classification 정확도 확인
          - 1.4%의 top-1 accuracy 달성



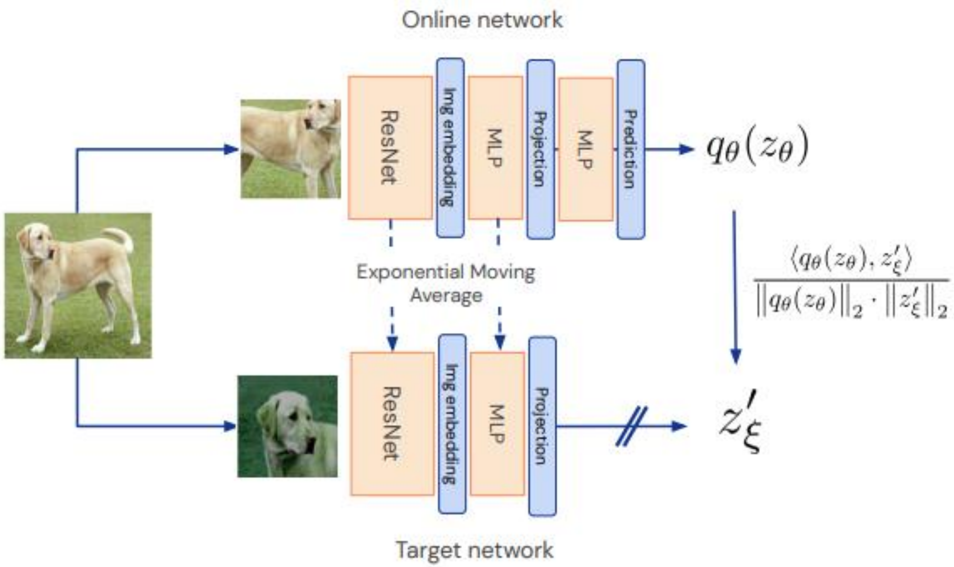
Self-Supervised Learning

- Bootstrap your own latent: A new approach to self-supervised Learning (NIPS 2020)
  - Motivation
    - Step 2
      - ✓ Random initialized backbone + MLP(predictor) 에 unlabeled dataset이 통과한 prediction 확보
    - Step 3
      - ✓ 새로운 random initialized model B를 step 2 에서 확보한 prediction 을 target 으로 학습
        - 18.8% 의 top-1 accuracy 달성
        - 부정확한 target 을 배우도록 학습하더라도 큰 폭으로 성능이 향상



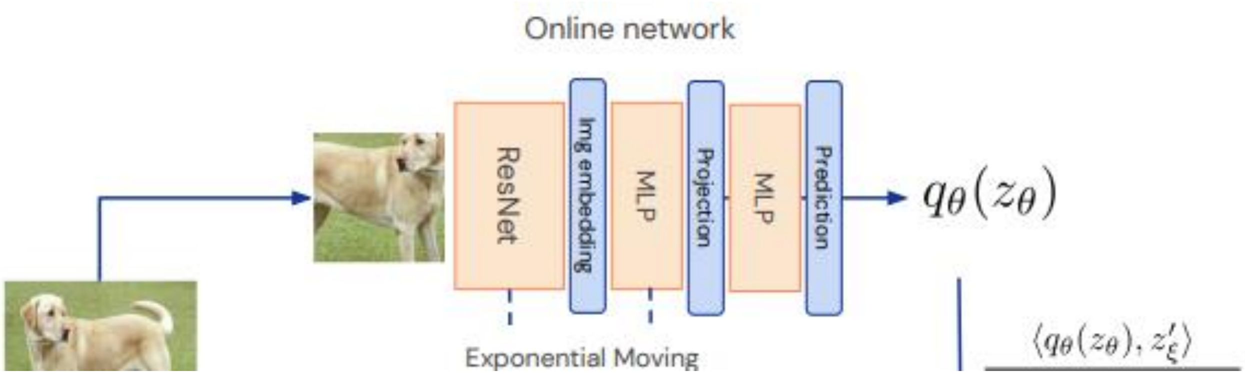
Self-Supervised Learning

- Bootstrap your own latent: A new approach to self-supervised Learning (NIPS 2020)
  - Model Architecture
    - Online network
      - Encoder (feature extractor) + projector + predictor
    - Target network
      - Encoder (feature extractor) + projector



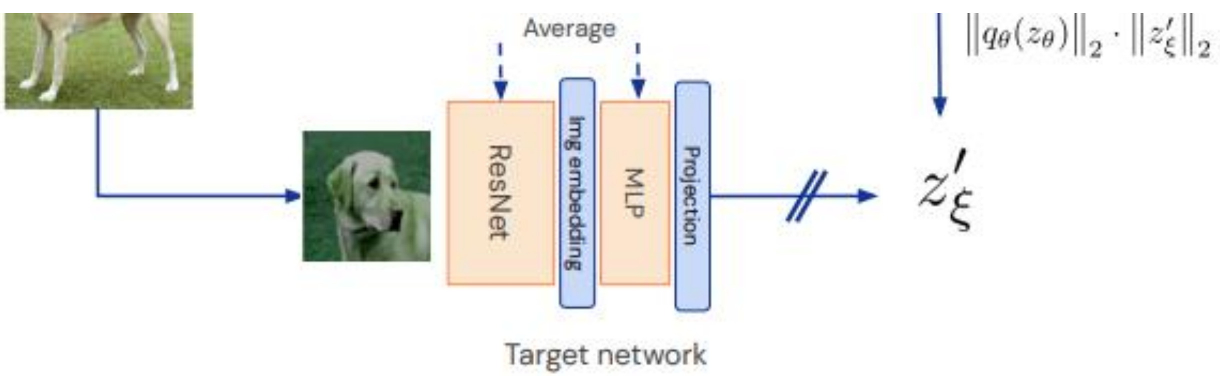
Self-Supervised Learning

- Bootstrap your own latent: A new approach to self-supervised Learning (NIPS 2020)
  - Model Architecture
    - Online network
      - Target network 가 제공한 target 을 prediction 하며 학습하는 network
      - Backpropagation 을 진행
      - Projector 와 predictor 는 모두 MLP 사용



Self-Supervised Learning

- Bootstrap your own latent: A new approach to self-supervised Learning (NIPS 2020)
  - Model Architecture
    - Target network
      - Encoder (feature extractor) + projector
      - Online network 가 학습할 target 제공
      - Stop-gradient
        - ✓ Backpropagation 하지 않음
        - ✓ Online network 의 weight 에 exponential moving average 를 적용하여 weight update
          - Stable 한 학습 가능 (stable 한 target 제공 가능)
          - Stop gradient 미적용시 collapsing 발생



Self-Supervised Learning

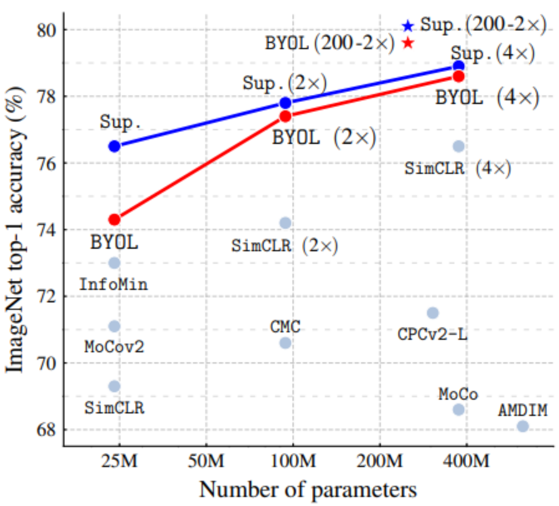
- Bootstrap your own latent: A new approach to self-supervised Learning (NIPS 2020)
  - Model Architecture
    - Symmetric Loss
      - Augmented image 를 online network 와 target network 에 바꿔 입력하여 loss 계산
      - Cosine similarity 사용

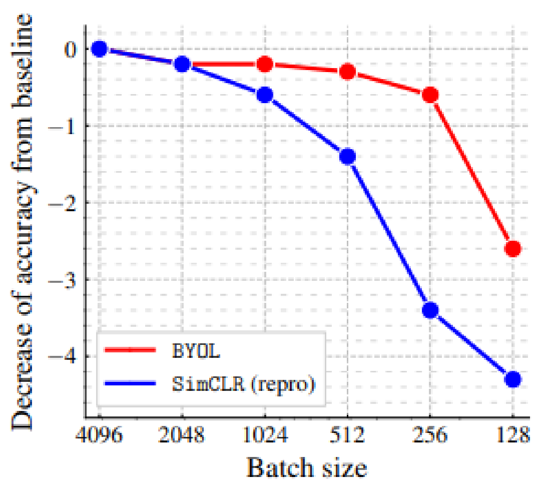
$$\mathcal{L}_{\theta,\xi} \triangleq \left\|\overline{q_\theta}(z_\theta) - \bar{z}'_\xi\right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi\rangle}{\left\|q_\theta(z_\theta)\right\|_2 \cdot \left\|z'_\xi\right\|_2}.$$

$$\mathcal{L}^{\texttt{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$$

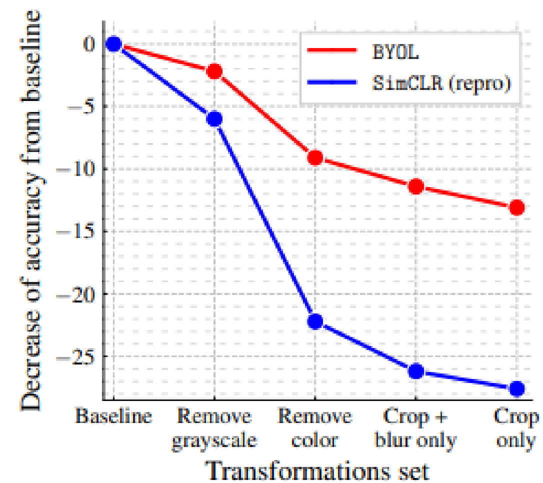
Self-Supervised Learning

- Bootstrap your own latent: A new approach to self-supervised Learning (NIPS 2020)
  - Experimental results
    - MoCo, SimCLR 을 넘어서는 성능을 보임
    - Batch size 가 256까지 감소하여도 성능 유지
    - 기존 contrastive learning 보다 augmentation 의 변화에 robust 함





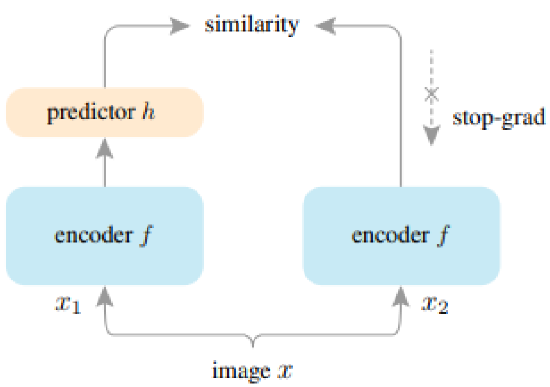
(a) Impact of batch size



(b) Impact of progressively removing transformations

Self-Supervised Learning

- Exploring Simple Siamese Representation Learning [8] (CVPR 2021)
  - Motivation
    - Unsupervised learning 에서 많이 사용되는 Siamese network 를 collapsing 이 발생하지 않게 사용할 수 있는 가장 simple 한 형태 제시
      - Negative sample pairs 를 사용하지 않음
      - large batch 를 사용하지 않음
      - Momentum encoders 를 사용하지 않음
    - Siamese network 에서 collapsing 이 일어나는 원인 실험적 분석
      - Stop-gradient 가 essential 한 role 을 가지고 있음.

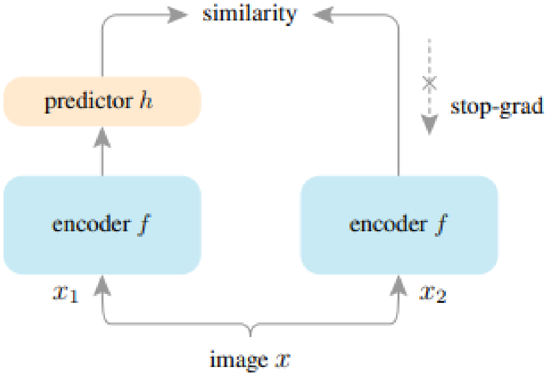


Self-Supervised Learning

- Exploring Simple Siamese Representation Learning (CVPR 2021)
  - Method
    - Siamese network 구조로 되어 있으나, 하나의 network 에만 predictor h 가 존재
    - Predictor 가 존재하지 않는 network 는 stop gradient 적용
    - Encoder 는 feature extractor + projector (MLP) 로 이루어져 있으며, 두 encoder 는 weight share
  - Loss function
    - Cosine similarity 사용
    - Symmetric 하게 cosine similarity 적용

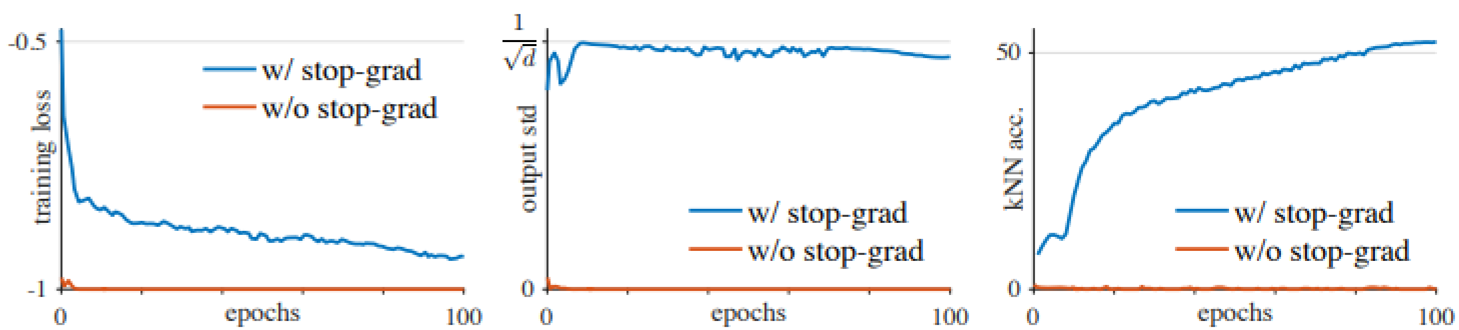
$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}$$

$$\mathcal{L} = \frac{1}{2}\mathcal{D}(p_1, \texttt{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(p_2, \texttt{stopgrad}(z_1))$$



Self-Supervised Learning

- Exploring Simple Siamese Representation Learning (CVPR 2021)
  - Empirical Study
    - Stop Gradient
      - Stop gradient 제거 시 encoder 의 output 이 loss를 낮추기 위해 constant value 만을 출력하는 collapsing 발생
        - ✓ Loss 는 최솟값인 -1로 수렴
        - ✓ Standard deviation 이 0으로 수렴 (collapsing)
      - 모델 구조 등을 변경(MLP layer 수, batch norm 적용 여부) 하여도 stop gradient 가 적용되지 않았을 때에는 항상 collapsing 발생
        - ✓ 원인은 알 수 없으나, stop gradient 가 collapsing 해결에 essential 한 role 을 담당함



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

Self-Supervised Learning

- Exploring Simple Siamese Representation Learning (CVPR 2021)
  - Empirical Study
    - Predictor
      - Predictor 제거 시 collapsing 발생
        - ✓ Loss = $\frac{1}{2}\mathcal{D}(z_1, \texttt{stopgrad}(z_2)) + \frac{1}{2}\mathcal{D}(z_2, \texttt{stopgrad}(z_1))$
          - $\mathcal{D}(z_1, z_2)$ 와 동일한 gradient direction 및 ½ downscale
        - ✓ Stop gradient 를 제거한 것과 동일한 효과
    - Batch size
      - Batch size 는 성능에 영향을 미치나, collapsing 의 원인은 아님
      - SimSiam 에서는 작은 batch size 에서도 좋은 성능을 보임

	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).

batch size	64	128	256	512	1024	2048	4096
acc. (%)	66.1	67.3	68.1	68.1	68.0	67.9	64.0

Table 2. **Effect of batch sizes** (ImageNet linear evaluation accuracy with 100-epoch pre-training).

Self-Supervised Learning

- Exploring Simple Siamese Representation Learning (CVPR 2021)
  - Empirical Study
    - Batch Normalization
      - 성능에 영향을 미치지만, collapsing 의 원인은 아님
        - ✓ Projector 는 모든 MLP 에 batch norm 을 적용
        - ✓ Predictor 는 batch norm 을 적용하지 않음
          - 적용시 unstable 한 학습 결과를 보임
    - Similarity Function
      - Loss function 을 cosine similarity 대신 cross-entropy 를 적용하여도 어느 정도 성능을 보임
    - Symmetrization
      - asymmetric 으로 loss 를 적용하여도 어느 정도 성능을 보임

	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

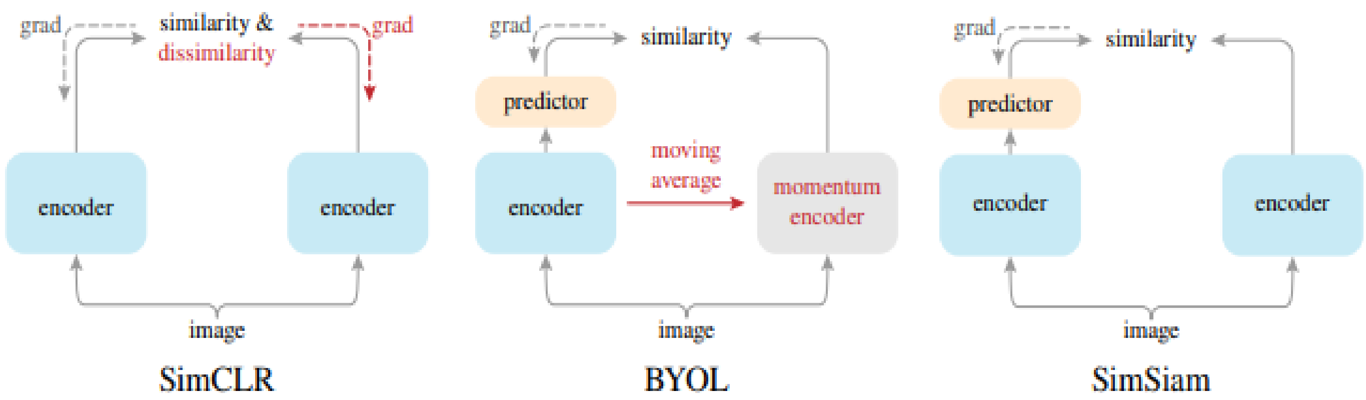
Table 3. **Effect of batch normalization on MLP heads** (ImageNet linear evaluation accuracy with 100-epoch pre-training).

	cosine	cross-entropy
acc. (%)	68.1	63.2

	sym.	asym.	asym. 2×
acc. (%)	68.1	64.8	67.3

Self-Supervised Learning

- Exploring Simple Siamese Representation Learning (CVPR 2021)
  - Methodology Comparisons
    - SimCLR
      - Negative pair 사용, large batch size
      - Stop gradient 미적용
      - Predictor 미적용
    - BYOL
      - Momentum encoder 사용



Self-Supervised Learning

- Exploring Simple Siamese Representation Learning (CVPR 2021)
  - Experimental results
    - 기존 contrastive learning 및 supervised learning 보다 높은 성능 달성

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

Table 4. **Comparisons on ImageNet linear classification**. All are based on **ResNet-50** pre-trained with **two 224×224 views**. Evaluation is on a single crop. All competitors are from our reproduction, and "+" denotes *improved* reproduction *vs*. original papers (see supplement).

pre-train	VOC 07 detection			VOC 07+12 detection			COCO detection			COCO instance seg.		
	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**

Table 5. **Transfer Learning**. All unsupervised methods are based on 200-epoch pre-training in ImageNet. *VOC 07 detection*: Faster

References

[1] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller and Thomas Brox. "Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks" *In NIPS*, 2014.

[2] Carl Doersch, Abhinav Gupta and Alexei A. Efros. "Unsupervised Visual Representation Learning by Context Prediction" *In ICCV*, 2015.

[3] Mehdi Noroozi and Paolo Favaro. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles" *In ECCV*, 2016.

[4] Spyros Gidaris, Praveer Singh and Nikos Komodakis. "Unsupervised Representation Learning By Predicting Image Rotations." *In ICLR*, 2018.

[5] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie and Ross Girshick. "Momentum Contrast for Unsupervised Visual Representation Learning." *In CVPR*, 2020.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi and Geoffrey Hinton. "A Simple Framework for Contrastive Learning of Visual Representations." *In ICML*, 2020.

[7] 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 and Michal Valko. "Bootstrap your own latent: A new approach to self-supervised Learning." *In NIPS*, 2020.

[8] Xinlei Chen and Kaiming He. "Exploring Simple Siamese Representation Learning." *In CVPR*, 2021.

Thank you