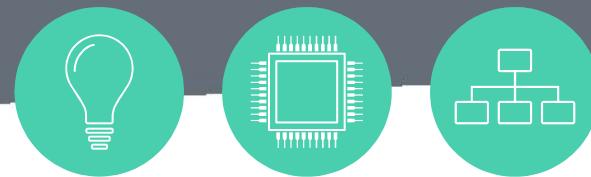


# یادگیری ماشین

مجتبی نافذ



# مباحث این جلسه

یادگیری بازنمایی (Representation Learning)

یادگیری خود نظارتی (self-supervised learning)

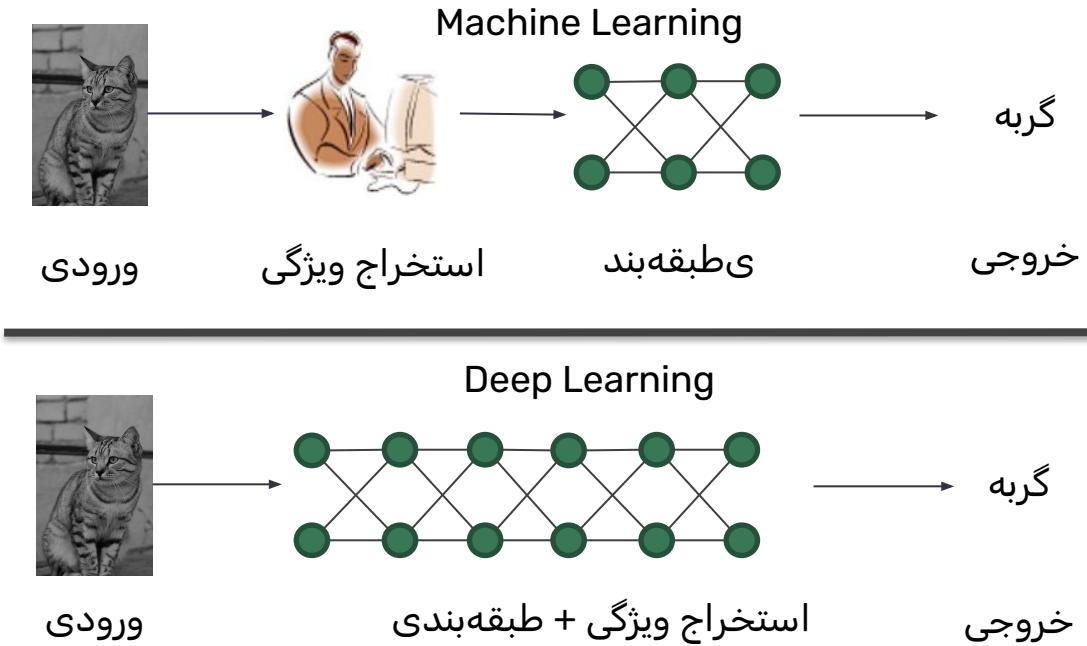
یادگیری بازنمایی به صورت خودناظارتی

یادگیری متقابل (Contrastive learning)



# یادگیری بازنمایی (Representation learning)

- تعریف : فرایند ساده‌سازی داده خام به الگوهای قابل فهم برای ماشین



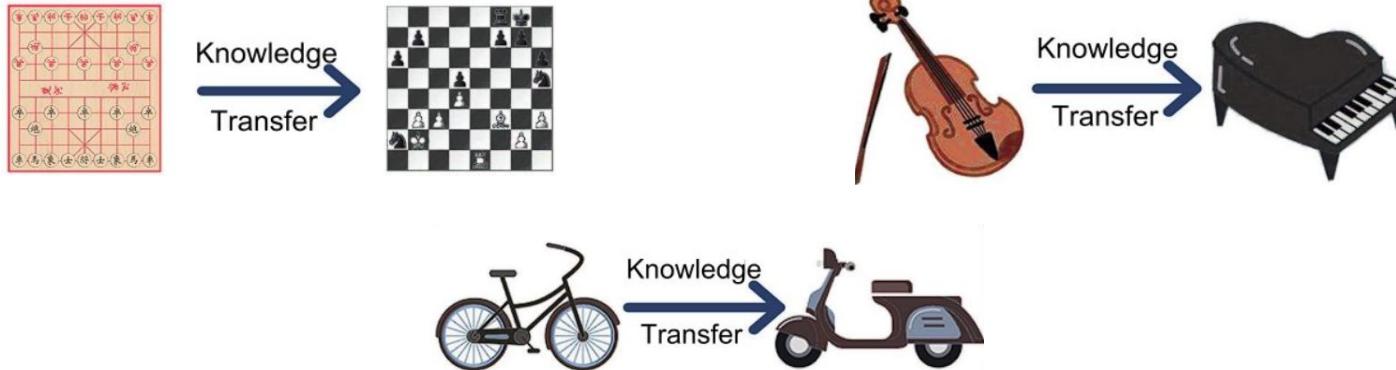
# یادگیری بازنمایی (Representation learning)

- **ویژگی‌های بازنمایی خوب**

- اطلاعات: حفظ ویژگی‌های مهم داده

- فشردگی: کاهش ابعاد و حجم نسبت به داده خام

- قابل تعمیم و انتقال بودن بازنمایی با استفاده از یادگیری ویژگی‌های معنایی: **Generalization**



# روش‌های یادگیری بازنمایی (Representation learning)

یادگیری نظارتی  
supervised learning

یادگیری خود نظارتی  
self-supervised learning

استفاده از داده برچسب گذاری شده

بدون برچسب  
(برچسب گذاری به صورت اتومات)

توجه روی یک وظیفه خاص

شناسایی الگوهای درون خود داده ها



# روش‌های یادگیری بازنمایی (Representation learning)

- ایراد روش‌های یادگیری بازنمایی نظارتی (supervised)
  - کمبود داده برچسب گذاری شده (تصاویر پزشکی)
  - عملکرد شبکه‌های عصبی عمیق روی داده‌ی بسیار زیاد قابل ملاحظه می‌شود
  - مدل‌های supervised ممکن است برای وظیفه طراحی شده راه میانبر (shortcut) یاد بگیرند (یادگیری همبستگی جعلی)

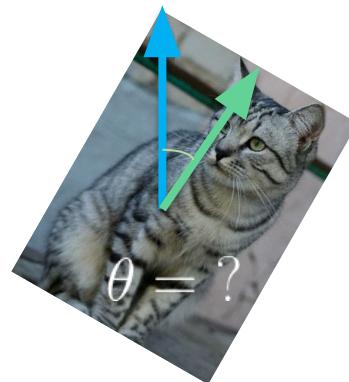


# یادگیری خود نظارتی (SSL)

مثال: پیش‌بینی / کامل کردن تصویر image transformation



تکمیل عکس



پیش‌بینی چرخش



پازل



رنگ‌آمیزی

یادگیری این وظایف (Tasks) بازنمایی خوبی از داده‌ها تولید می‌کند.  
به صورت خودکار برای این وظایف دستاویز برچسب تولید می‌کنیم.

- 
-

# ارزیابی SSL

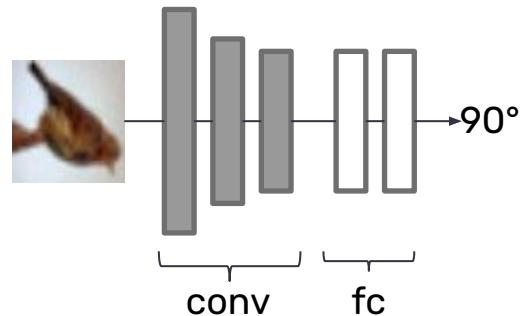
- معمولاً در SSL عملکرد مدل روی وظیفه طراحی شده مهم نیست  
( اهمیتی ندارد مدل چرخش تصویر را درست پیشビینی کند )
- ارزیابی مدل بدست آمده روی وظایف پایین دستی (Downstream tasks)



# ارزیابی SSL

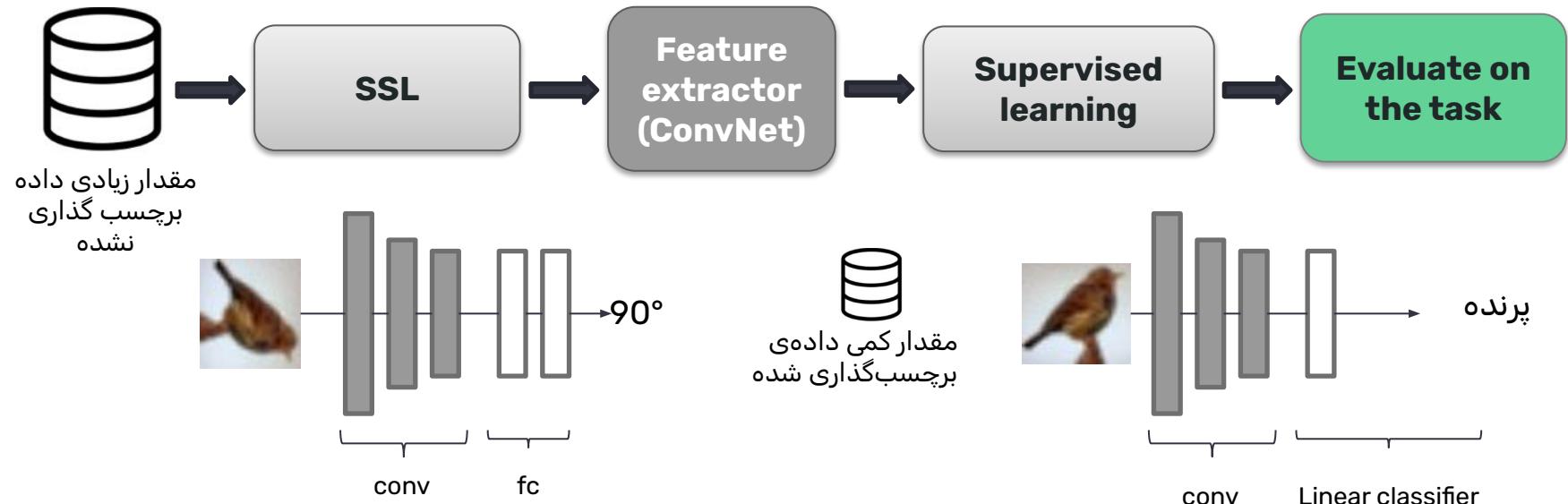


مقدار زیادی  
داده برچسب  
گذاری نشده



یادگیری بازنمایی خوب  
از طریق SSL  
روی وظایف  
مثل چرخش تصویر

# ارزیابی SSL



وصل کردن یک شبکه کم عمق روی شبکه آموزش مدل روی وظیفه مورد نظر



# انواع یادگیری خود نظارتی

Pretext Tasks

Contrastive Learning

Generative Models



# Pretext Tasks : Rotation



90° rotation



270° rotation



180° rotation



0° rotation



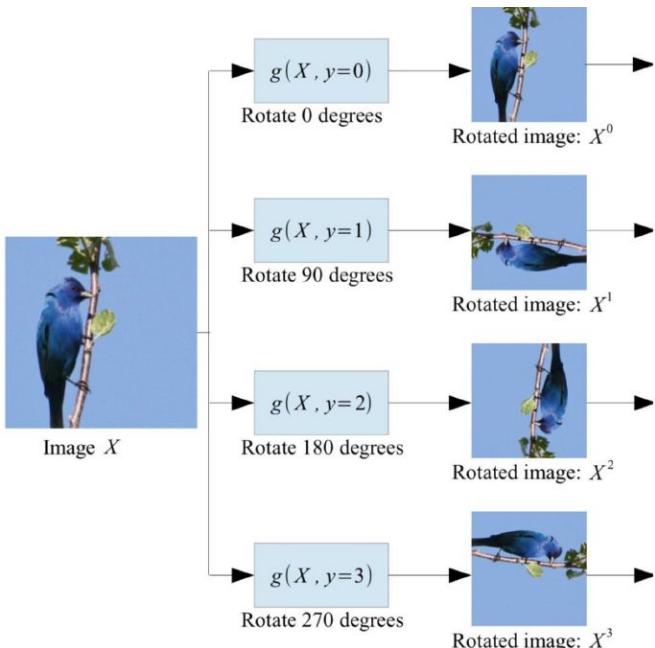
270° rotation

چرا آموزش مدل روی وظایف دستاویز منجر به بازنمایی بهتر میشود؟

- مدلی میتواند این وظایف (مثلا چرخش) را به درستی پیش بینی کند که درک بالایی از تصویر رسیده باشد.

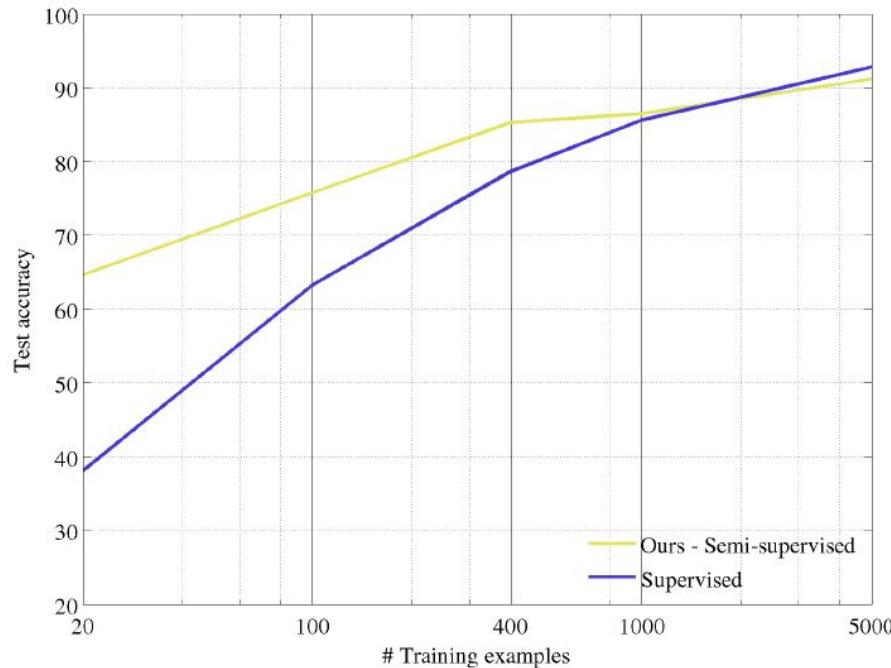


# Pretext Tasks : Rotation



- یادگیری خود نظارتی از طریق ایجاد چرخش‌های مختلف و تولید برچسب‌های متناظر
- مدل یاد می‌گیرد هر تصویر دچار کدام چرخش شده است (یک مدل طبقه‌بندی چهار کلاسه)

# Pretext Tasks : Rotation



مدل نظارتی روی دادگان CIFAR10 — —  
مدل آموزش دیده روی وظیفه  
چرخش به صورت خود نظارتی روی  
كل دادگان آموزش CIFAR10 —



# Pretext Tasks : Rotation

	Classification (%mAP)	Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all
ImageNet labels	78.9	79.9	56.8
Random		53.3	43.4
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4
Context (Doersch et al., 2015)	55.1	65.3	51.1
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7
ColorProxy (Larsson et al., 2017)	65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4
(Ours) RotNet	<b>70.87</b>	<b>72.97</b>	<b>54.4</b>
			<b>39.1</b>

پیشآموزش  
روی دادگان  
Image Net

بدون  
پیشآموزش

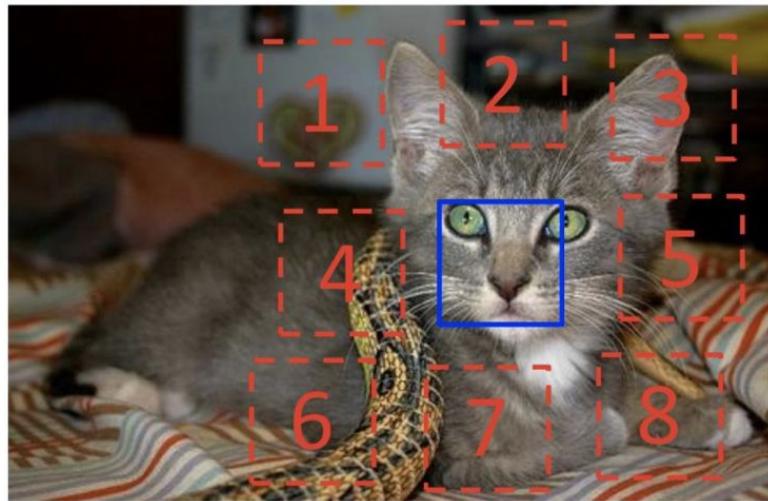
تعمیم پذیری به بقیه  
وظایف و دادگان

یادگیری خود نظارتی روی  
کل دادگان آموزش  
ImageNet

Fine Tune  
برچسب گذاری شده  
Pascal VOC 2007

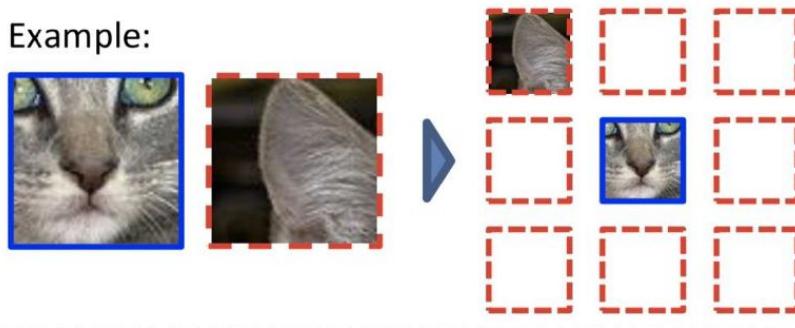


# Pretext Tasks : Relative Patch Location



$$X = (\underset{\text{Patch 1}}{\text{cat eyes}}, \underset{\text{Patch 2}}{\text{cat ear}}); Y = 3$$

Example:



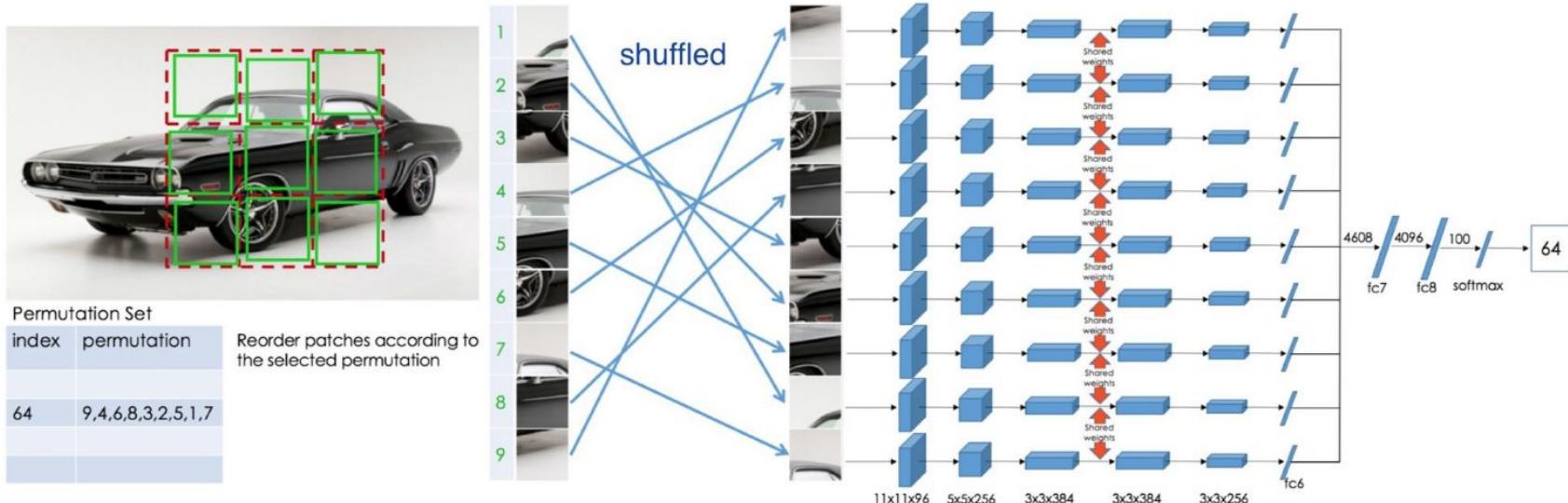
Question 1:



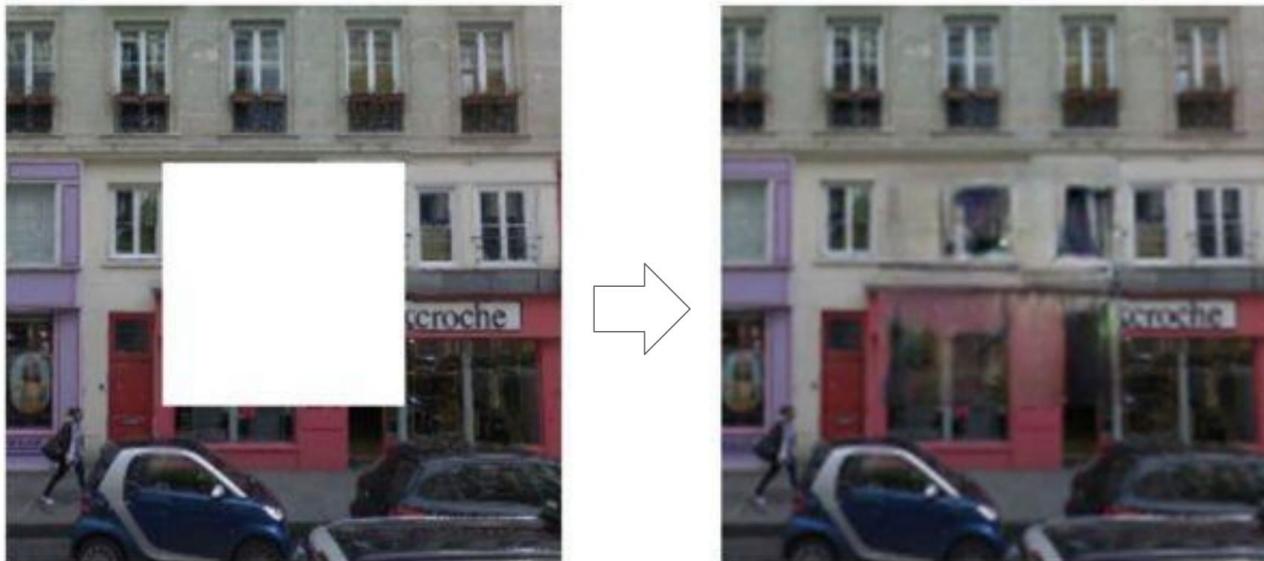
Question 2:



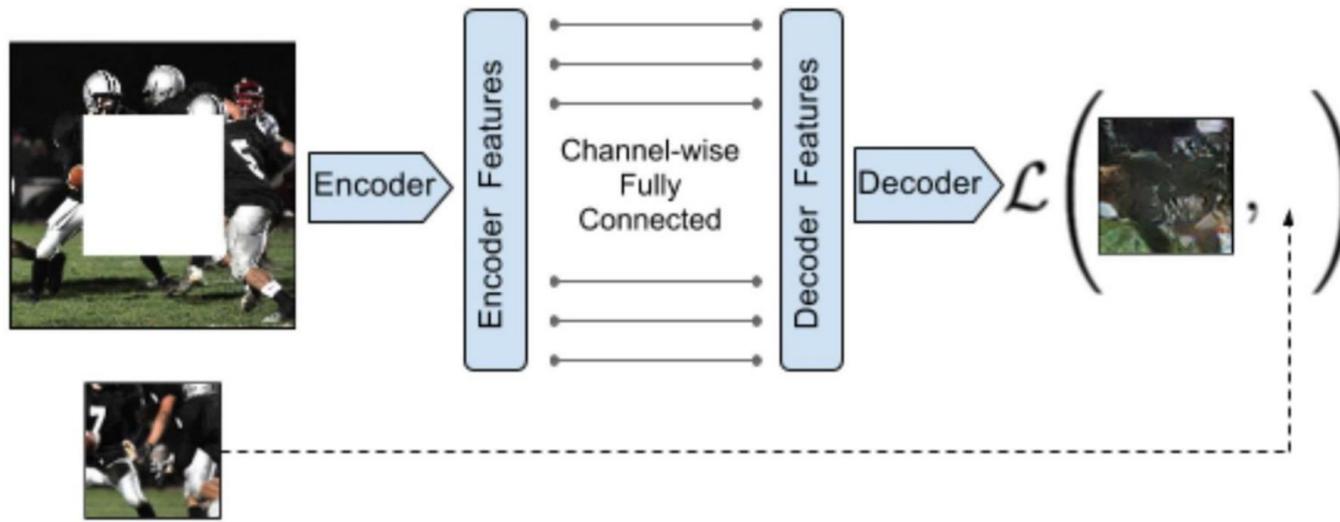
# Pretext Tasks : Jigsaw Puzzle



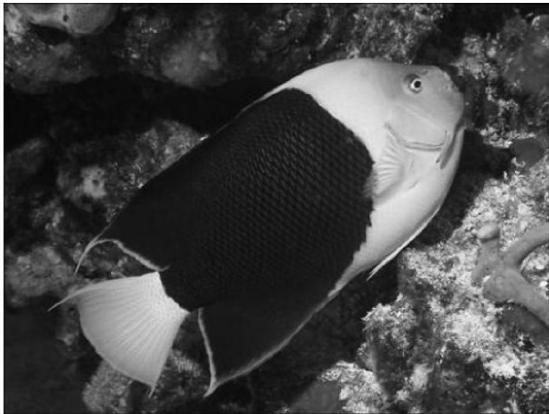
# Pretext task: predict missing pixels (inpainting)



# Pretext task: predict missing pixels (inpainting)

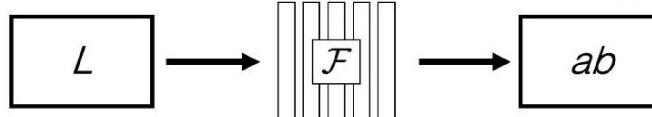


# Pretext Tasks : Image Coloring



Grayscale image:  $L$  channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

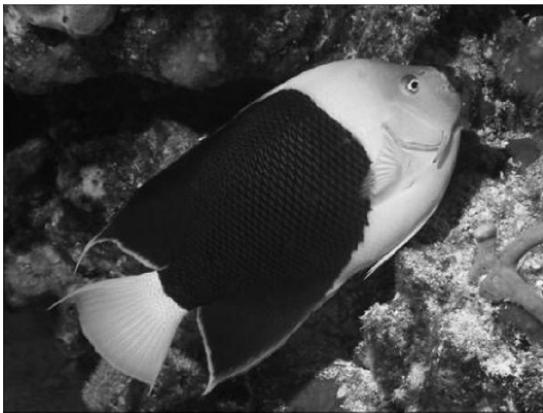


Color information:  $ab$  channels

$$\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$$

5

# Pretext Tasks : Image Coloring



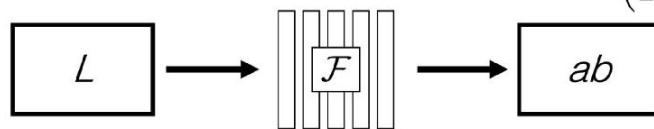
Grayscale image:  $L$  channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$



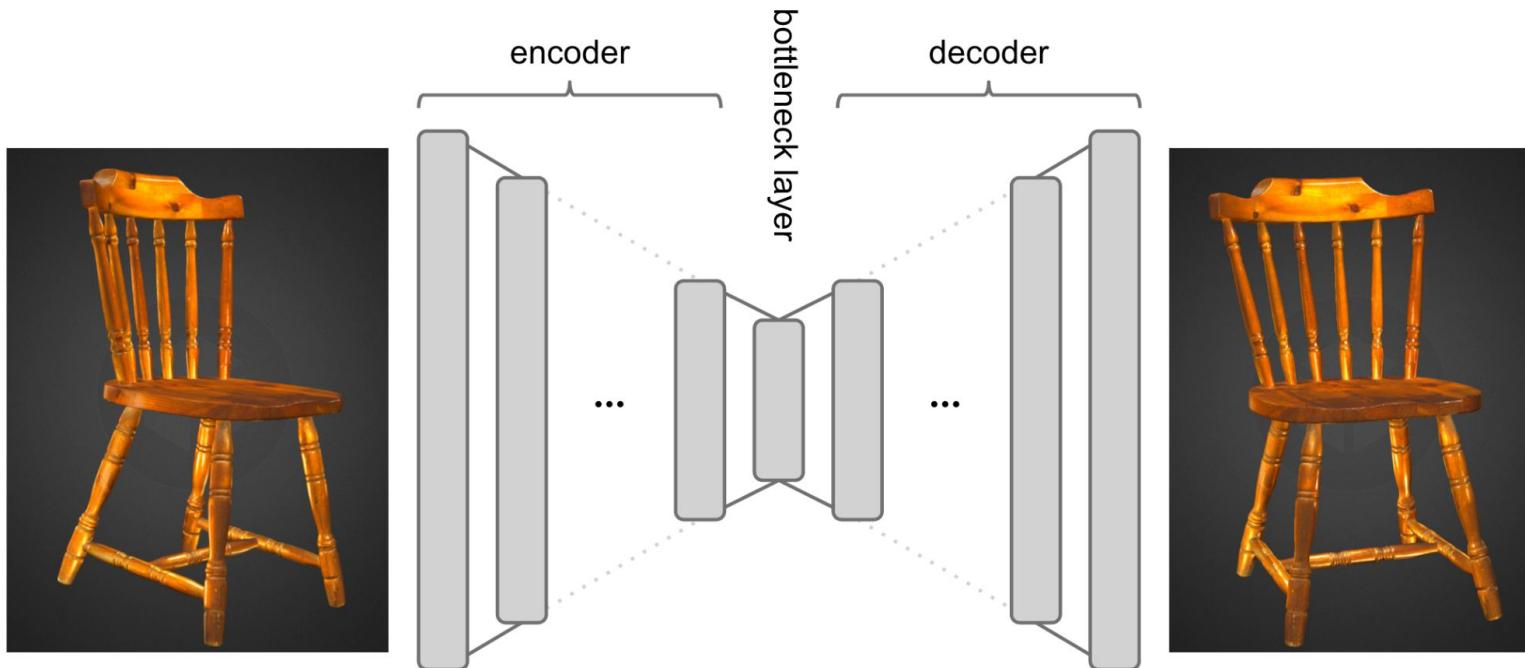
Concatenate ( $L, ab$ ) channels

$$(\mathbf{X}, \widehat{\mathbf{Y}})$$

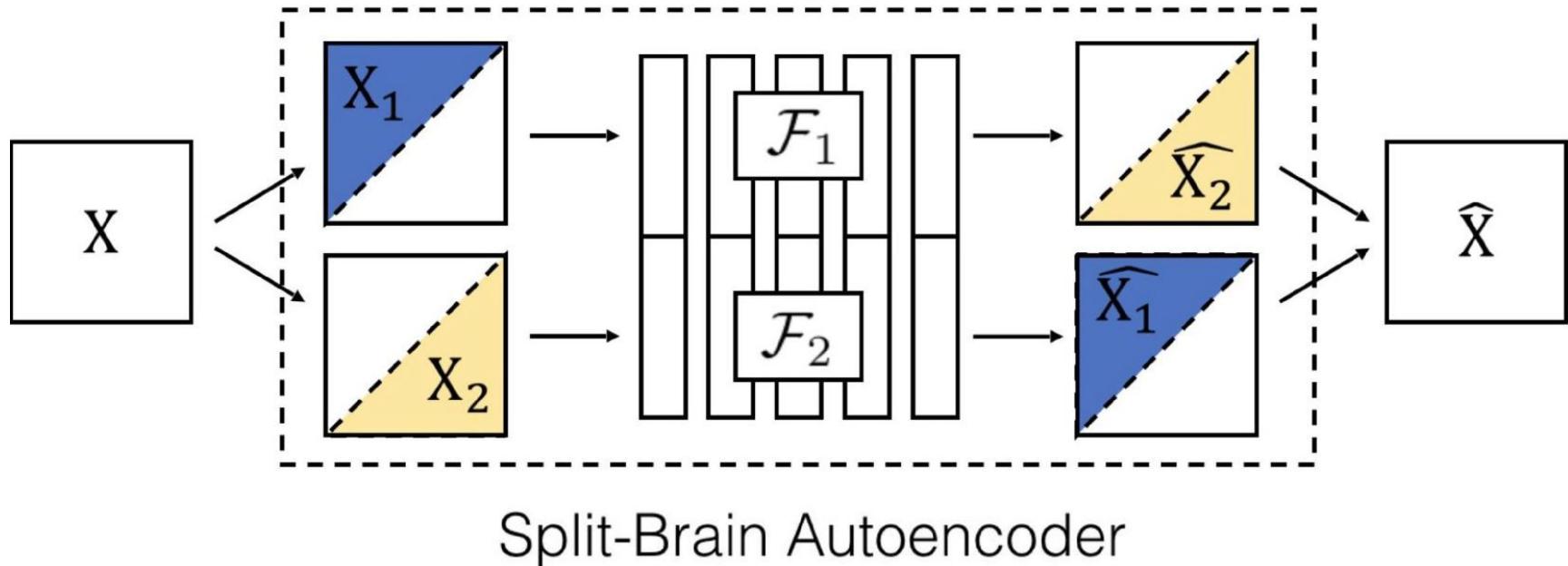


□

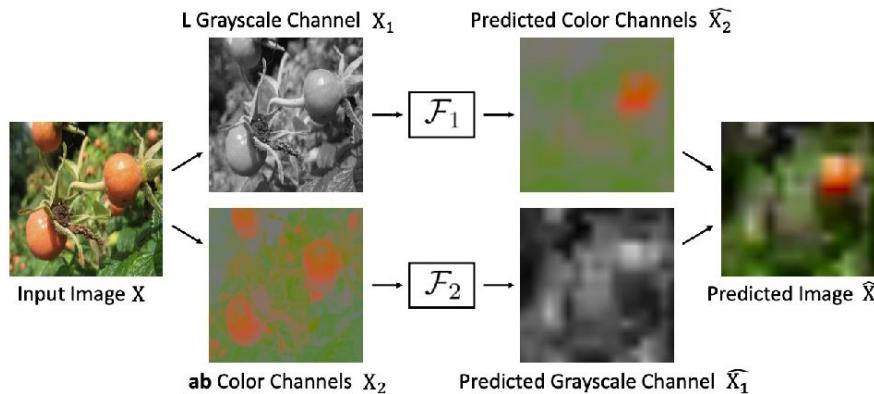
# Pretext Tasks : Image Reconstruction



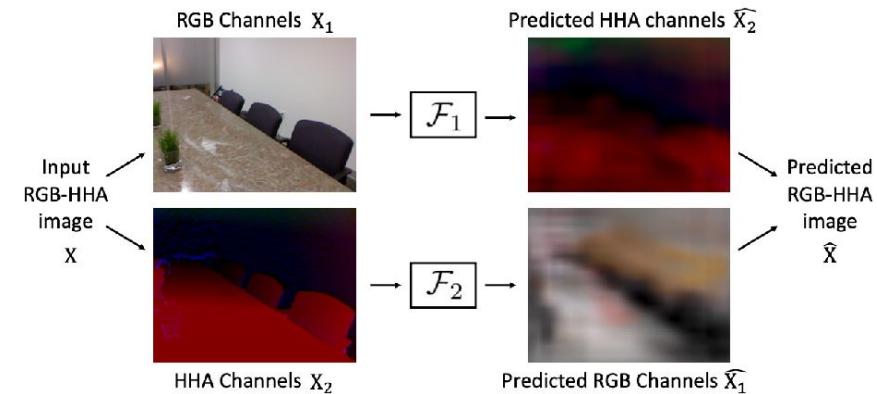
# Pretext Tasks : Split - Brain



# Pretext Tasks : Split - Brain



(a) **Lab Images**



(b) **RGB-D Images**



# خلاصه

- وظایف دستاویز با تمرکز بر ویژگی‌های بصری عمل می‌کنند مثل چرخش، رنگ‌آمیزی و موقعیت نسبی
- مدل‌ها برای حل این وظایف دستاویز مجبور به یادگیری بازنمایی خوبی از تصاویر به لحاظ معنایی می‌شوند
- ارزیابی مدل‌ها روی وظایف دستاویز اهمیت چندانی ندارند بلکه عملکرد مدل با استفاده از بازنمایی‌های بدست آمده روی وظایف پایین دستی است که اهمیت دارد (classification, detection, segmentation)



# مشکلات

- پیدا کردن وظیفه‌های دستاویز(Pretext) خوب طاقت فرسا است
- بازنمایی‌های یادگرفته شده ممکن است قابل تعمیم باشند، و مختص یک Pretext task باشند.



# Contrastive Learning

مفهوم و تعریف تابع هزینه

مدل های MoCo و SimCLR

بیان روش های اخیرتر



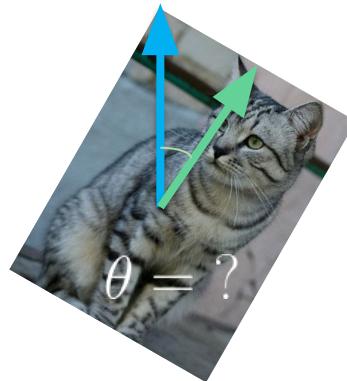
# یادگیری خود نظارتی (SSL)

مثال: پیش‌بینی / کامل کردن تصویر image transformation



؟

تکمیل عکس



پیش بینی چرخش



پازل

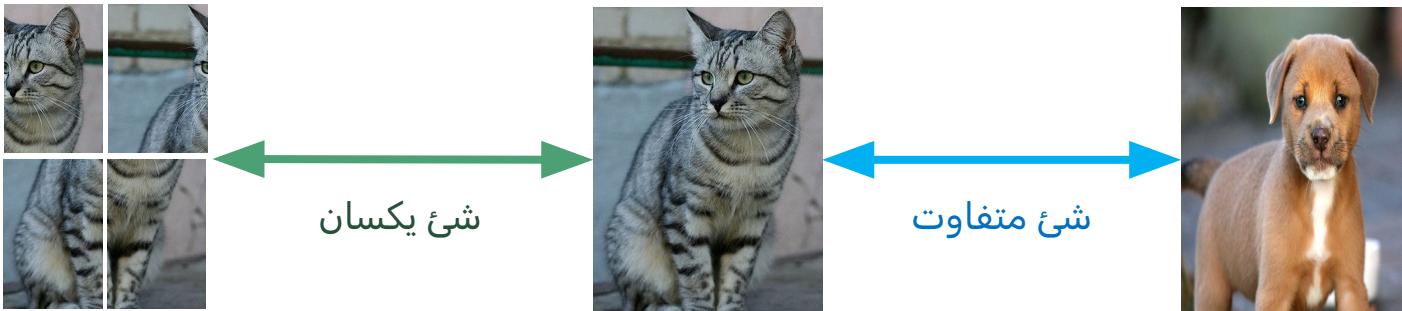


رنگ آمیزی

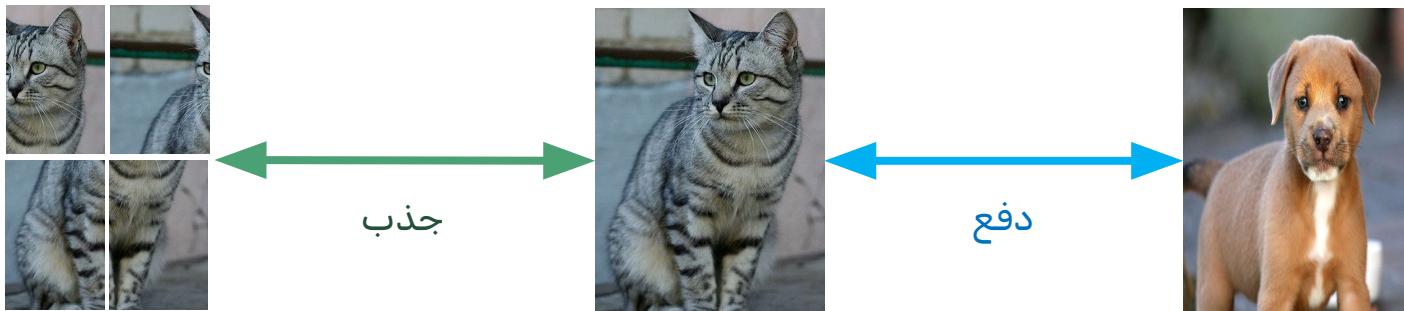
یادگیری این وظایف (Tasks) بازنمایی خوبی از داده‌ها تولید می‌کند.  
به صورت خودکار برای این وظایف دستاویز برچسب تولید می‌کنیم.

- 
-

# Contrastive Learning



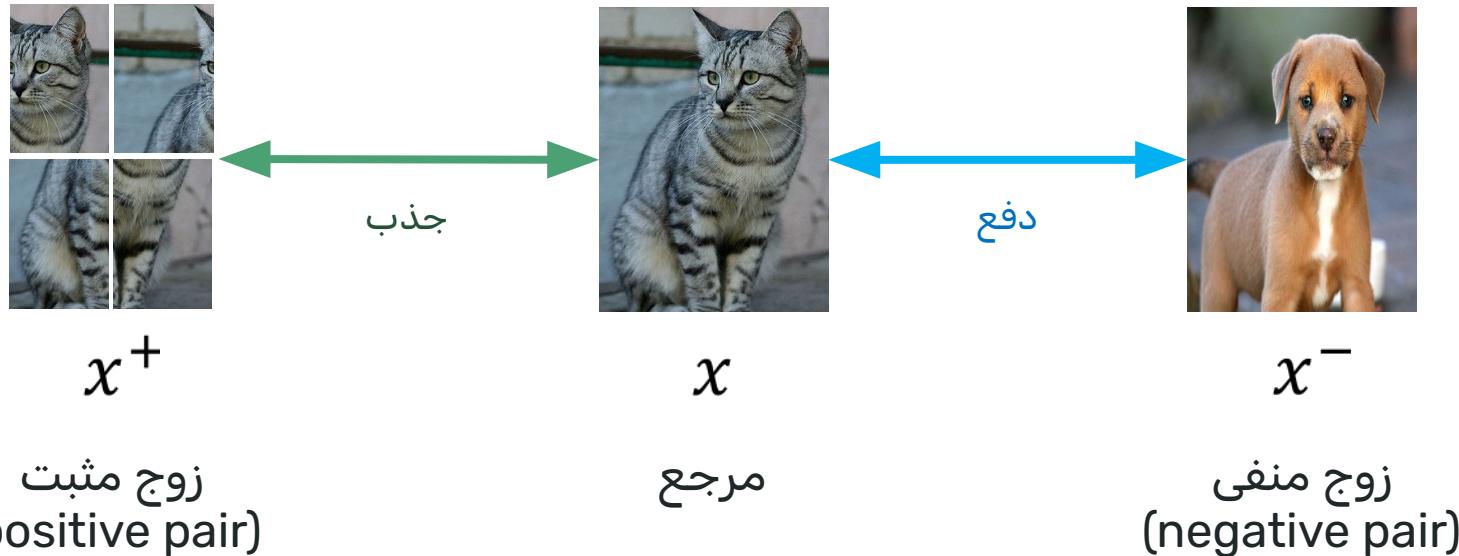
# Contrastive Learning



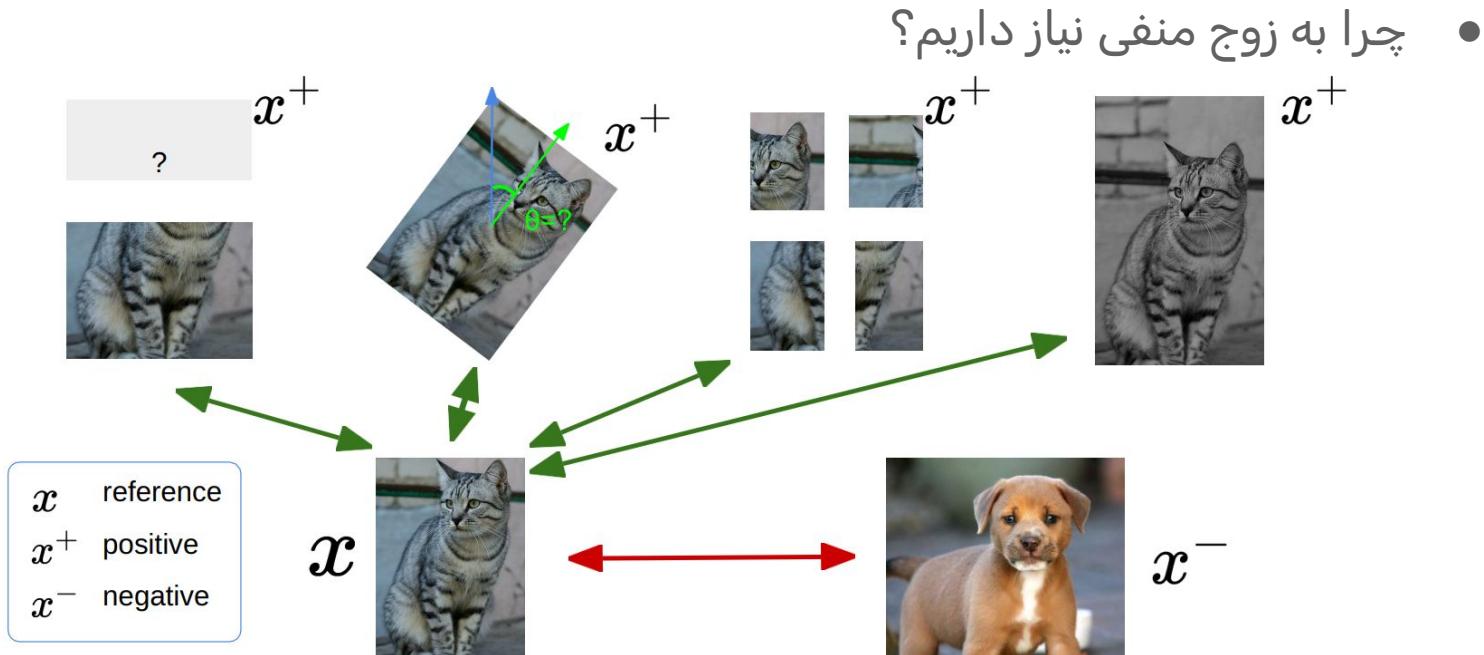
ایده : بازنمایی‌های یک شئ بهم نزدیک شوند و از بقیه اشیا دور شود



# Contrastive Learning



# Contrastive Learning



# Contrastive Learning

$$s(u, v) = \frac{u^t v}{\|u\| \|v\|}$$

- تابع score یک تابع تعیین شده برای محاسبه شباهت دو بازنمایی
- هدف پیدا کردن یا آموزش شبکه ( $f$ ) است که با اعمال تابع امتیاز روی بازنمایی‌های خاص برای جفت‌های مثبت ( $x, x^+$ ) عدد خیلی بزرگتری نسبت به تمام جفت‌های منفی ( $x, x^-$ ) باشد

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



# Loss Function

$$\mathcal{L} = -\frac{1}{N} \sum_{x=x_1}^{x_N} \log \frac{e^{score(f(x), f(x^+))}}{e^{score(f(x), f(x^+))} + \sum_{k=1}^{N-1} e^{score(f(x), f(x_k^-))}}$$

# Contrastive Learning

- یک جفت مثبت و  $N-1$  جفت منفی

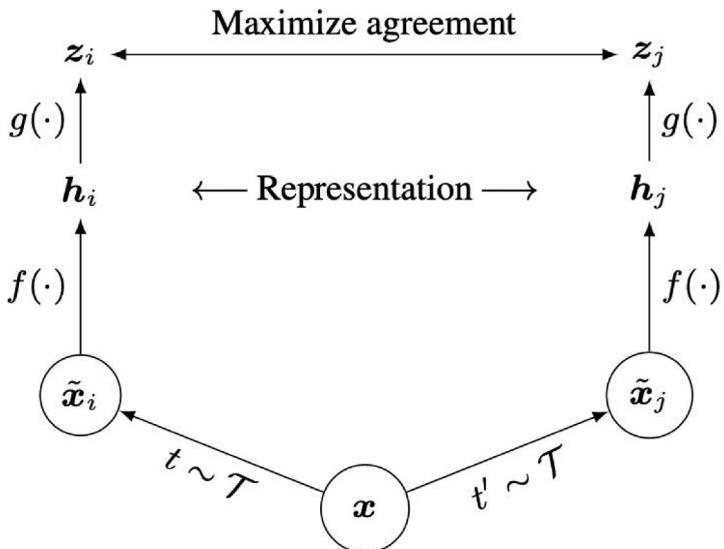
$$\mathcal{L} = -\frac{1}{N} \sum_{x=x_1}^{x_N} \log \frac{e^{\text{score}(f(x), f(x^+))}}{e^{\text{score}(f(x), f(x^+))} + \sum_{k=1}^{N-1} e^{\text{score}(f(x), f(x_k^-))}}$$

- تابع هزینه به اندازه Batch بستگی دارد.
- هرچه اندازه Batch بزرگتر بهتر است. (ایده آل چیست؟)



# SimCLR

- تابع  $s(u, v) = \frac{u^t v}{\|u\| \|v\|}$  score بیان می کند چقدر هم جهت هستند.



- فضای ویژگی ( $h$ ) با یک شبکه کم عمق ( $g$ ) به فضای  $z$  برده میشود و فرایнд آموزش متضاد روی این فضا انجام می شود
- جفت های مثبت از طریق data augmentation نظیر برش تصادفی و بلور تصادفی ساخته می شود

# SimCLR : Data Augmentation



(a) Original



(b) Crop and resize



(c) Crop, resize (and flip)



(d) Color distort. (drop)



(e) Color distort. (jitter)



(f) Rotate  $\{90^\circ, 180^\circ, 270^\circ\}$



(g) Cutout



(h) Gaussian noise



(i) Gaussian blur



(j) Sobel filtering



# SimCLR : Algorithm

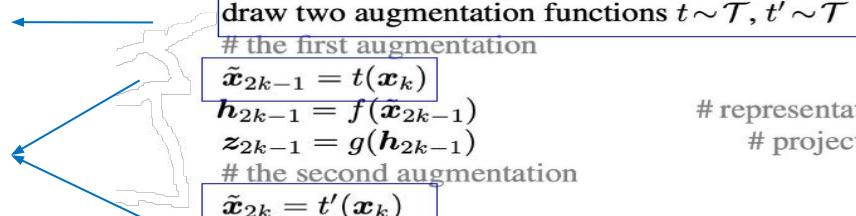
**Algorithm 1** SimCLR's main learning algorithm.

```

input: batch size  $N$ , constant  $\tau$ , structure of  $f, g, \mathcal{T}$ .
for sampled minibatch  $\{\mathbf{x}_k\}_{k=1}^N$  do
    for all  $k \in \{1, \dots, N\}$  do
        draw two augmentation functions  $t \sim \mathcal{T}, t' \sim \mathcal{T}$ 
        # the first augmentation
         $\tilde{\mathbf{x}}_{2k-1} = t(\mathbf{x}_k)$ 
         $\mathbf{h}_{2k-1} = f(\tilde{\mathbf{x}}_{2k-1})$  # representation
         $\mathbf{z}_{2k-1} = g(\mathbf{h}_{2k-1})$  # projection
        # the second augmentation
         $\tilde{\mathbf{x}}_{2k} = t'(\mathbf{x}_k)$ 
         $\mathbf{h}_{2k} = f(\tilde{\mathbf{x}}_{2k})$  # representation
         $\mathbf{z}_{2k} = g(\mathbf{h}_{2k})$  # projection
    end for
    for all  $i \in \{1, \dots, 2N\}$  and  $j \in \{1, \dots, 2N\}$  do
         $s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \|\mathbf{z}_j\|)$  # pairwise similarity
    end for
    define  $\ell(i, j)$  as  $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$ 
     $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$ 
    update networks  $f$  and  $g$  to minimize  $\mathcal{L}$ 
end for
return encoder network  $f(\cdot)$ , and throw away  $g(\cdot)$ 

```

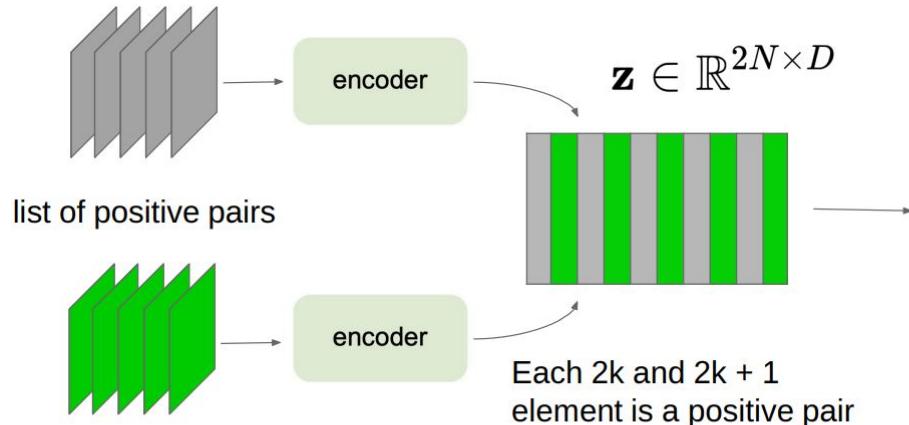
انتخاب دو تابع



تولید جفت‌های مثبت با  
تابع انتخاب شده

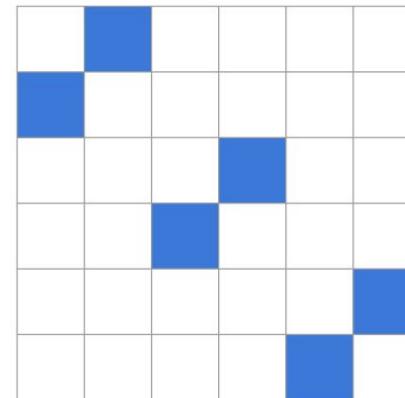
پیمایش روی  $2N$  نمونه به  
عنوان مرجع و محاسبه  
تابع هزینه

# SimCLR



$$s_{i,j} = \frac{\mathbf{z}_i^T \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|}$$

"Affinity matrix"

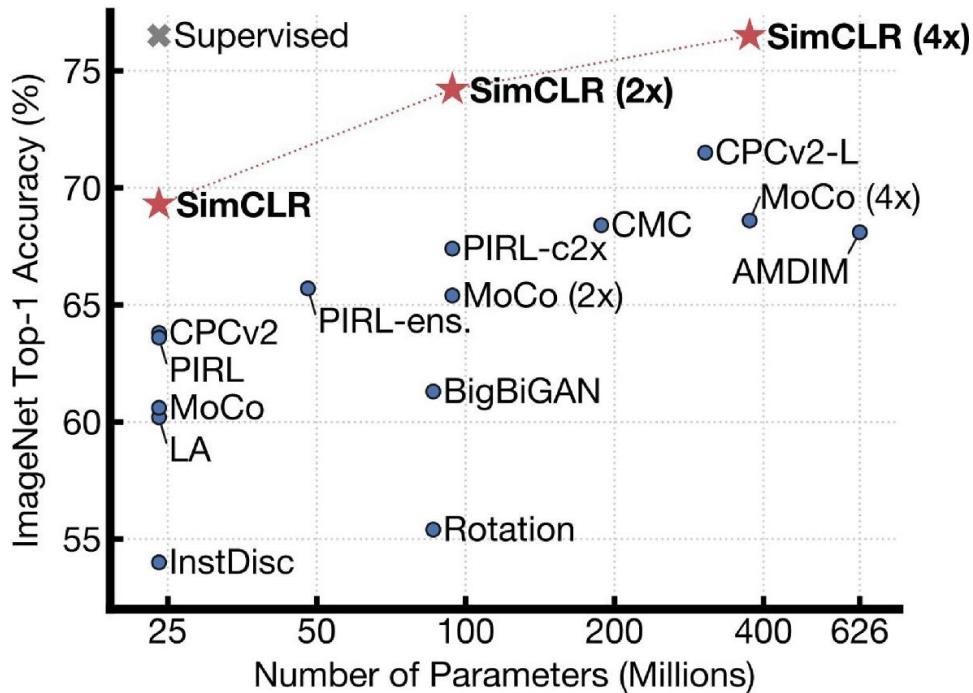


$2N$

$2N$

= classification label for each row

# SimCLR : Evaluation



- آموزش انکودر SimCLR روی کل داده‌های آموزش ImageNet
- استفاده انکودر فریز شده و آموزش یک طبقه‌بند خطی با استفاده از داده برچسب‌گذاری شده

# SimCLR : Evaluation

Method	Architecture	Label fraction		
		1%	10%	Top 5
Supervised baseline	ResNet-50	48.4	80.4	
<i>Methods using other label-propagation:</i>				
Pseudo-label	ResNet-50	51.6	82.4	
VAT+Entropy Min.	ResNet-50	47.0	83.4	
UDA (w. RandAug)	ResNet-50	-	88.5	
FixMatch (w. RandAug)	ResNet-50	-	89.1	
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2	
<i>Methods using representation learning only:</i>				
InstDisc	ResNet-50	39.2	77.4	
BigBiGAN	RevNet-50 (4×)	55.2	78.8	
PIRL	ResNet-50	57.2	83.8	
CPC v2	ResNet-161(*)	77.9	91.2	
SimCLR (ours)	ResNet-50	75.5	87.8	
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2	
SimCLR (ours)	ResNet-50 (4×)	<b>85.8</b>	<b>92.6</b>	

Table 7. ImageNet accuracy of models trained with few labels.

- آموزش مدل SimCLR روی کل

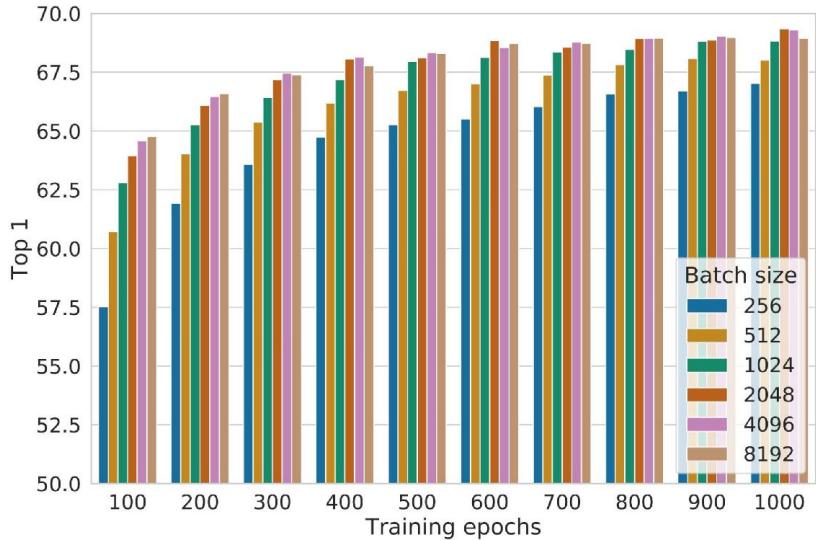
- داده‌های آموزش ImageNet

- تنظیم دقیق شبکه آموزش داده شده

- روی یک و ده درصد داده‌های

- برچسب‌گذاری شده ImageNet

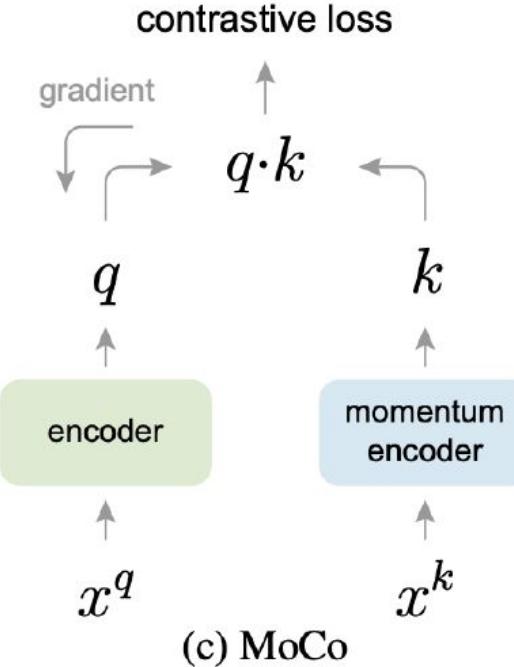
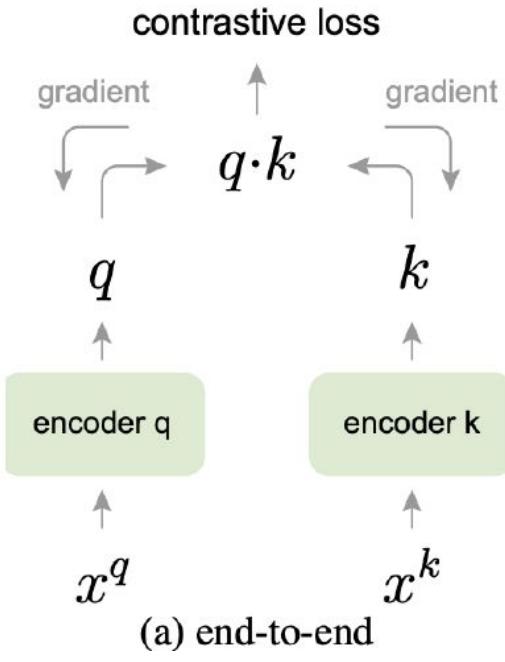
# SimCLR : Batch Size



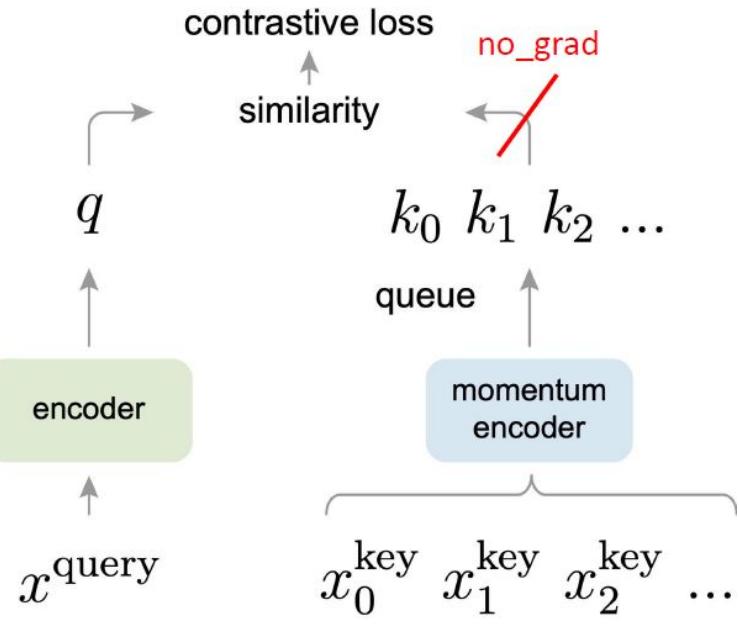
- حیاتی بودن اندازه Batch برای آموزش متضاد (چرا؟)
- بزرگ نیازمند حافظه بسیار بزرگ نیز است و سخت افزار قوی است.

Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.<sup>10</sup>

# MoCo (Momentum Contrastive Learning)



# تفاوت های SimCLR با MoCo



- نگهداری صفر از بازنمایی‌های نمونه‌های منفی (Key)
- محاسبه گرادیان و بروزرسانی وزن‌های encoder از طریق داده‌های query
- مجزا کردن داده‌های منفی و سایز batch (پشتیبانی از Batch Size بزرگ)
- آپدیت وزن‌های Encoder بصورت  $w_k = mw_k + (1 - m)w_q$



# MoCo : Algorithm

**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
```

تولید جفت‌های مثبت با نمونه  
گیری از توابع Augmentation

جلوگیری از تشکیل گرادیان

بروزرسانی صفت نمونه‌های منفی

استفاده از صفت نمونه‌های منفی (keys)

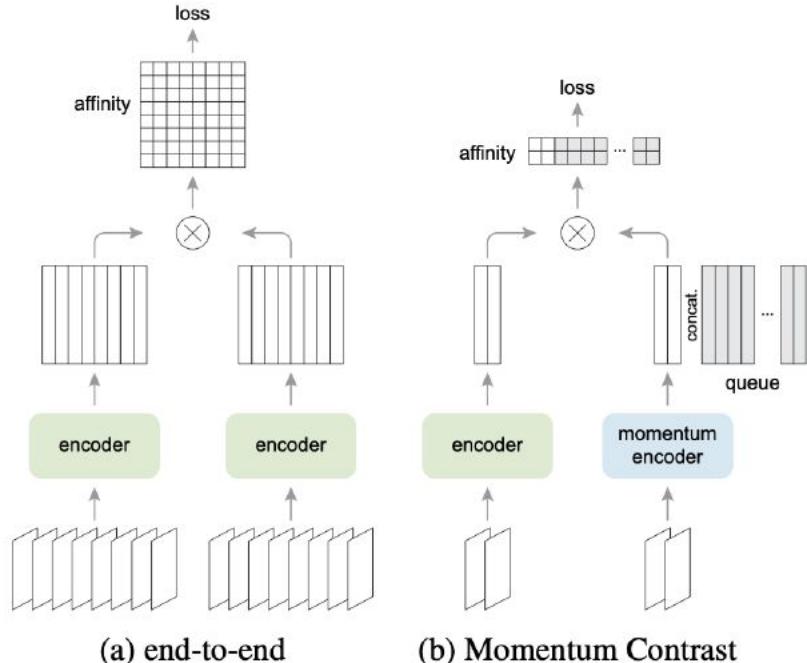
تشکیل تابع هزینه

بروزرسانی شبکه f\_k از طریق momentum

bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.



# MoCo V2



ترکیب ایده‌های **MoCo** و **SimCLR**

- استفاده از سر غیر خطی برای آموزش متضاد
- استفاده از بروزرسانی **MoCo** که اجازه میدهد تعداد نمونه‌های منفی افزایش یابد.



سر غیر خطی و Augmentation ضروري است

case	unsup. pre-train				ImageNet acc.	VOC detection		
	MLP	aug+	cos	epochs		AP <sub>50</sub>	AP	AP <sub>75</sub>
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	<b>82.5</b>	57.2	63.9
(d)	✓	✓	✓	200	67.5	82.4	57.0	63.6
(e)	✓	✓	✓	<b>800</b>	<b>71.1</b>	<b>82.5</b>	<b>57.4</b>	<b>64.0</b>

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.



# MoCo : Evaluation

case	MLP	aug+	cos	unsup. pre-train epochs	batch	ImageNet acc.
MoCo v1 [6]				200	256	60.6
SimCLR [2]	✓	✓	✓	200	256	61.9
SimCLR [2]	✓	✓	✓	200	8192	66.6
<b>MoCo v2</b>	✓	✓	✓	200	256	<b>67.5</b>

*results of longer unsupervised training follow:*

SimCLR [2]	✓	✓	✓	1000	4096	69.3
<b>MoCo v2</b>	✓	✓	✓	800	256	<b>71.1</b>

Table 2. **MoCo vs. SimCLR**: ImageNet linear classifier accuracy (**ResNet-50, 1-crop  $224 \times 224$** ), 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).

- سر غیر خطی و Augmentation ضروری است

- جداسازی نمونه‌های منفی از داخل بچ اجازه

- می دهد که MoCo-V2 نسبت به SimCLR با

- اندازه بچ کوچکتر عملکرد بهتری داشته باشد

- (vs 8192 256)



# MoCo : Evaluation

mechanism	batch	memory / GPU	time / 200-ep.
MoCo	256	<b>5.0G</b>	<b>53 hrs</b>
end-to-end	256	7.4G	65 hrs
end-to-end	4096	93.0G <sup>†</sup>	n/a

Table 3. **Memory and time cost** in 8 V100 16G GPUs, implemented in PyTorch. <sup>†</sup>: based on our estimation.

- سر غیر خطی و Augmentation ضروری است
- جداسازی نمونه‌های منفی از داخل بچ اجازه می‌دهد که MoCo-V2 نسبت به SimCLR با اندازه بچ کوچکتر عملکرد بهتری داشته باشد (vs 8192 256)
- حافظه مصرفی بشدت کاهش پیدا کرده است و لی نتیجه بهبود یافته است

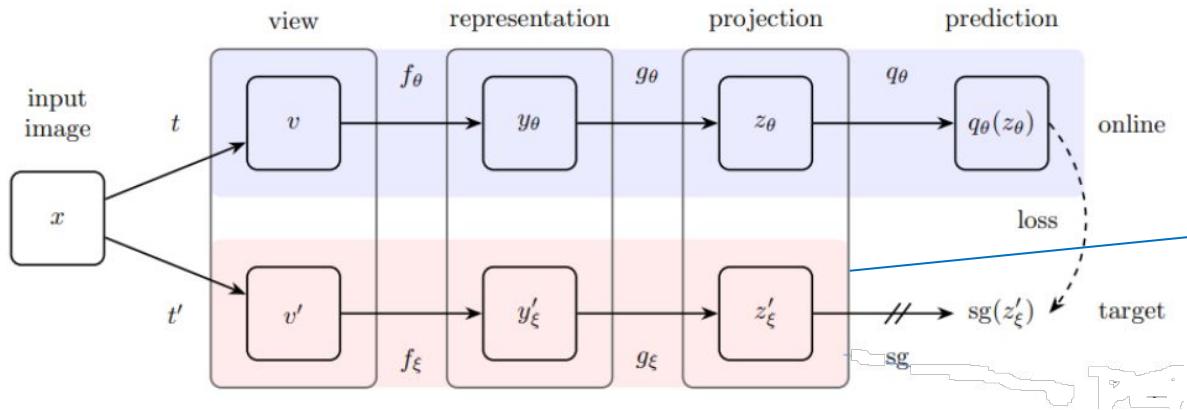


# BYOL : Bootstrap Your Own Latent

- احتیاجی به داده منفی ندارد
- از دو شبکه استفاده می کند (online,target) که باهم در تعامل هستند و از هم یاد میگیرند
- یک شبکه پیشビینی کننده به شبکه آنلاین اضافه شده است
- استفاده از شبکه target باعث تشویق مدل online به یادگیری اطلاعات بیشتر می شود

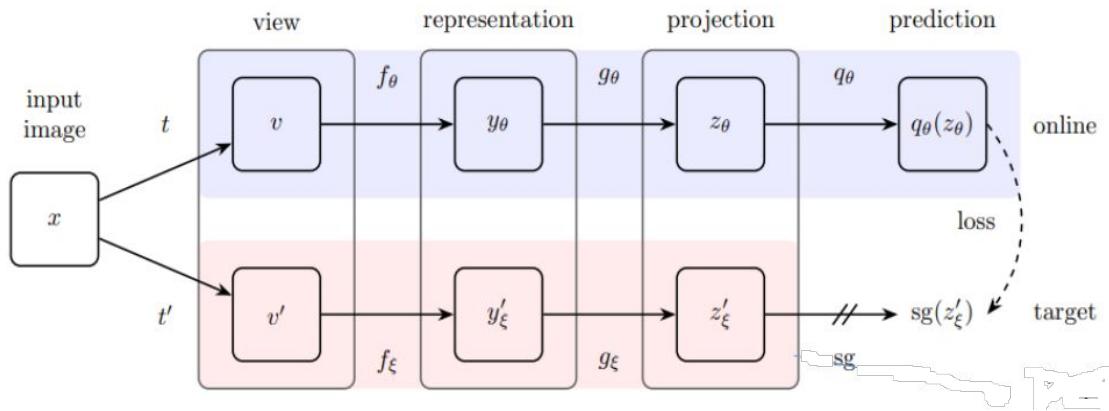


# BYOL : Bootstrap Your Own Latent



شبکه یک moving target online از شبکه average است.

# BYOL : LOSS



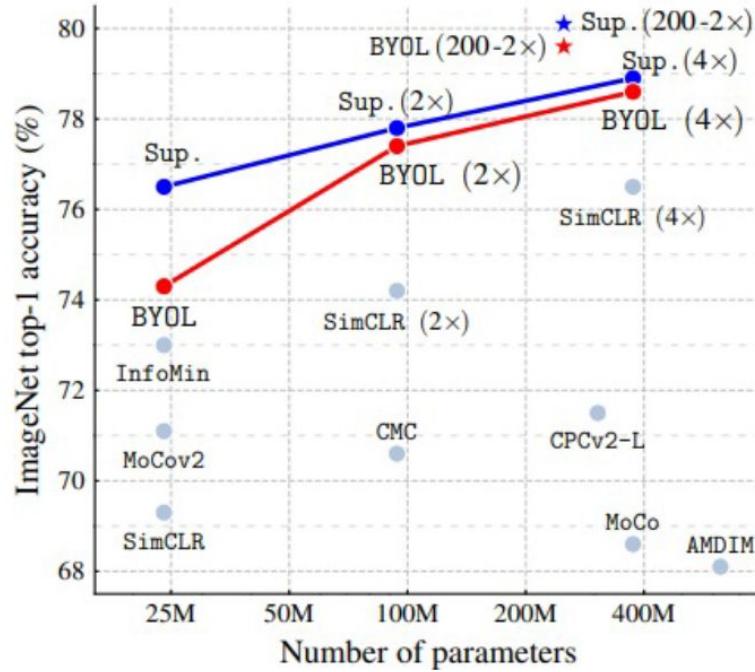
$$\overline{q}_\theta(z_\theta) \triangleq q_\theta(z_\theta)/\|q_\theta(z_\theta)\|_2$$

$$\overline{z}'_\xi \triangleq z'_\xi / \|z'_\xi\|_2$$

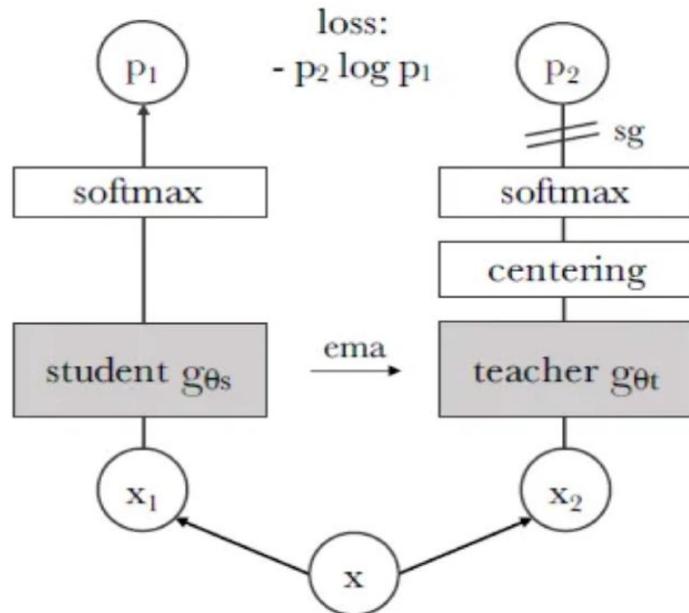
$$\mathcal{L}_{\theta,\xi}^{\text{BYOL}} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$$

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

# BYOL : Evaluation



# Dino



# Dino : Result

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

# خلاصه

- یادگیری خودناظارتی (self-supervised learning)
  - وظایف دستاویز (pretext tasks)
    - ... rotation, jigsaw puzzle, image coloring
  - یادگیری تقابلی (contrastive learning)
    - SimCLR, MoCo, BYOL, Dino
  - یادگیری با استفاده از مدل های مولد (generative models)



# References

<https://viso.ai/deep-learning/representation-learning/>

[Gidaris et al. Unsupervised Representation Learning by Predicting Image Rotations, ICLR 2018 May](#)

[Doersch et al., Unsupervised Visual Representation Learning by Context Prediction, 2015](#)

[Noroozi & Favaro, Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, 2016](#)

[Zhang, Isola & Efros, Colorful Image Colorization, 2016](#)

[Zhang, Isola & Efros, Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction, CVPR 2017](#)

[Chen et al., A Simple Framework for Contrastive Learning of Visual Representations, 2020](#)

[He et al., Momentum Contrast for Unsupervised Visual Representation Learning, 2020](#)

[J.B. Grill et al., Bootstrap Your Own Latent, 2020](#)

[Caron et al., Emerging Properties in Self-Supervised Vision Transformers, 2022](#)

Deep Learning Course, Dr.Soleimani 2023

Understanding Deep Learning, 2023

