

رسالة محمد



# یادگیری عمیق

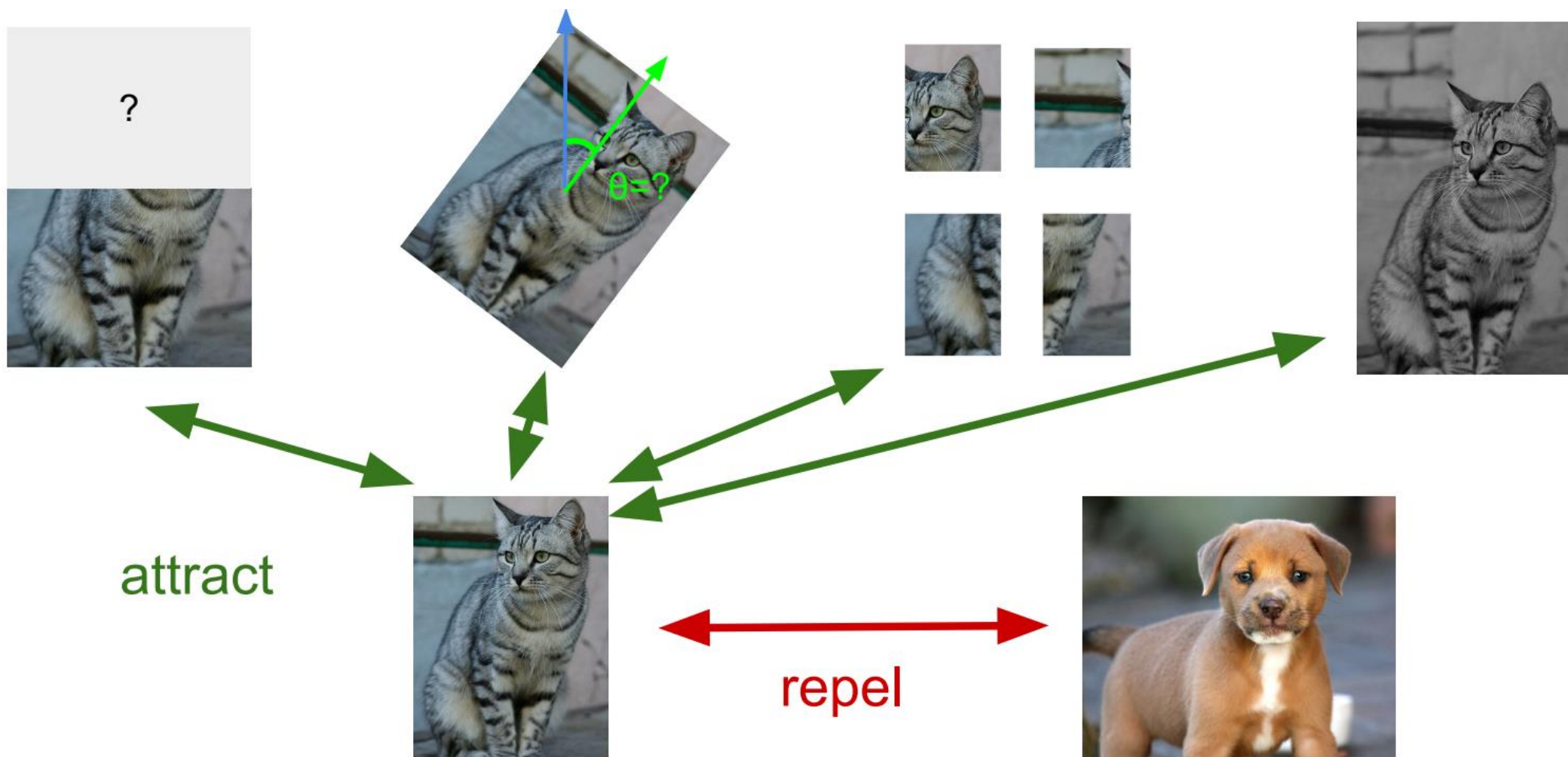
مدرس: محمدرضا محمدی

بهار ۱۴۰۲

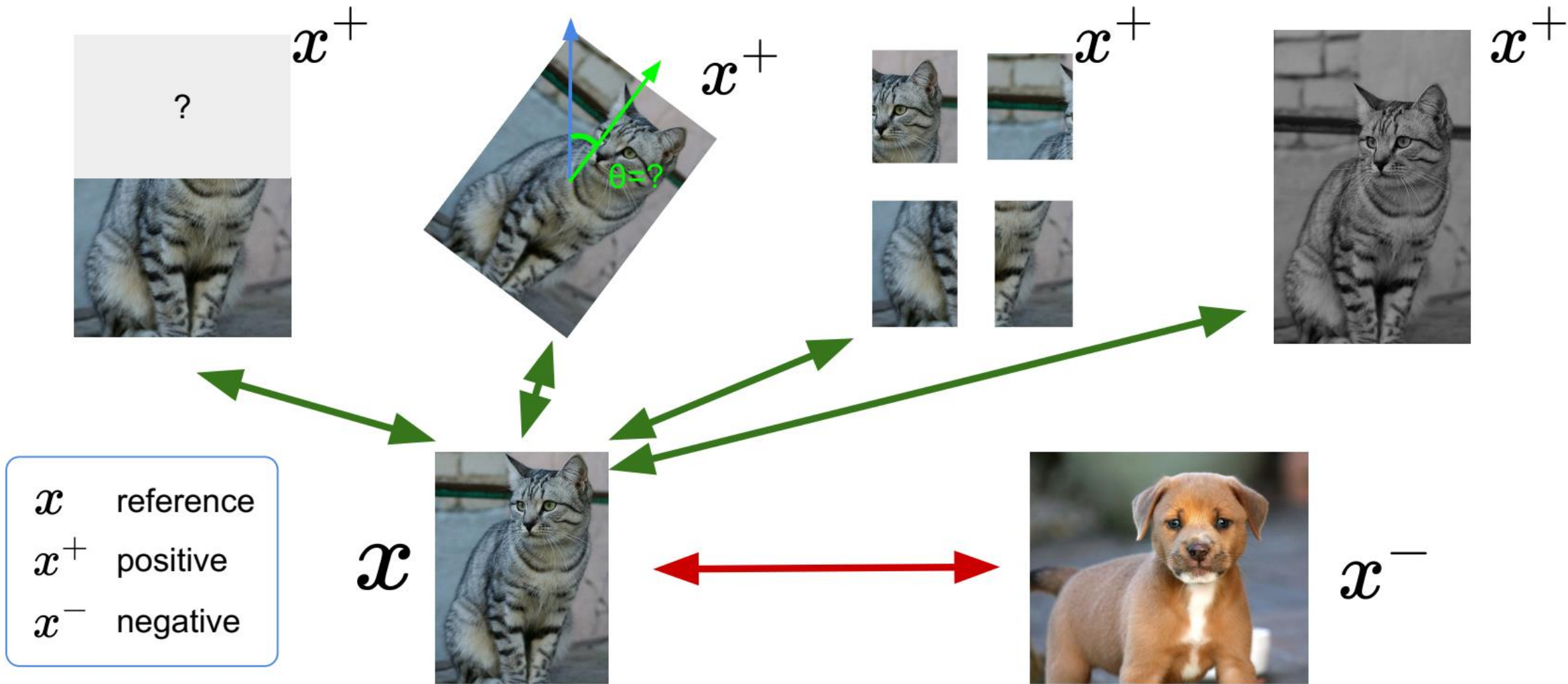
یادگیری بازنمایی

Representation Learning

# یادگیری بازنمایی مقابله‌ای (Contrastive)



# یادگیری بازنمایی مقابله‌ای (Contrastive)



# یک فرمولاسیون برای یادگیری مقابله‌ای



- می‌خواهیم با استفاده از آموزش یک کدگذار، ویژگی‌های مربوط به تصویر مرجع و مثبت به هم نزدیک باشند اما ویژگی‌های تصویر مرجع و تصویر منفی دور باشند

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



$x$  reference

$x^+$  positive

$x^-$  negative

$x$



$x^-$



# یک فرمولاسیون برای یادگیری مقابله‌ای

- تابع ضرر برای حالتی که یک نمونه مثبت و  $N-1$  نمونه منفی داشته باشیم:

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



$x$



$x^+$



$x$



$x_1^-$



$x_2^-$



$x_3^-$

...

# یک فرمولاسیون برای یادگیری مقابله‌ای

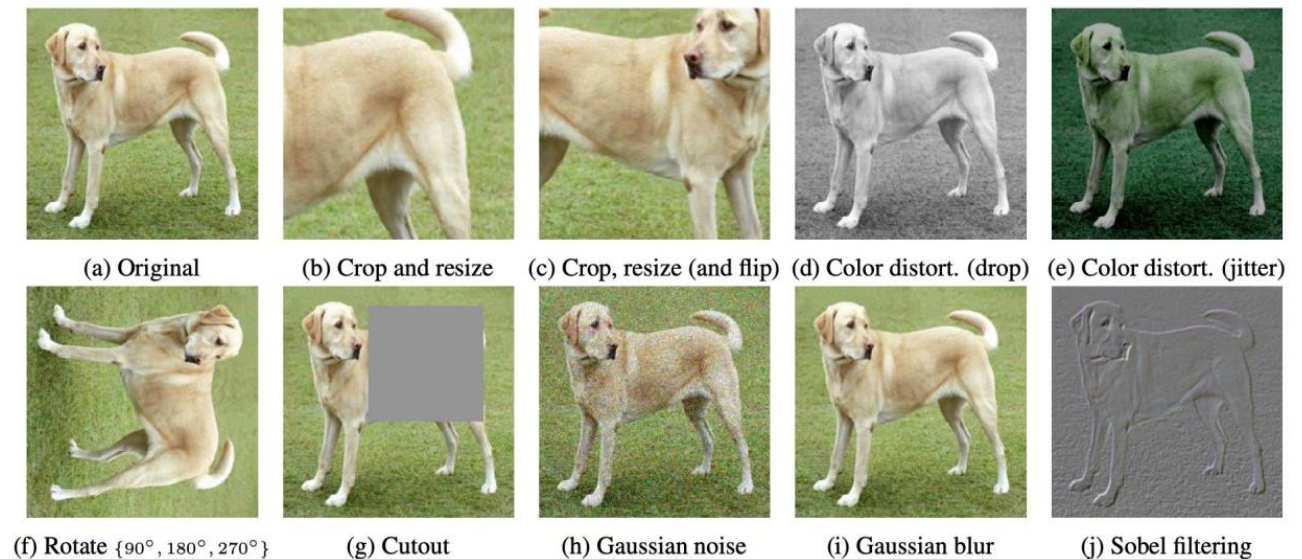
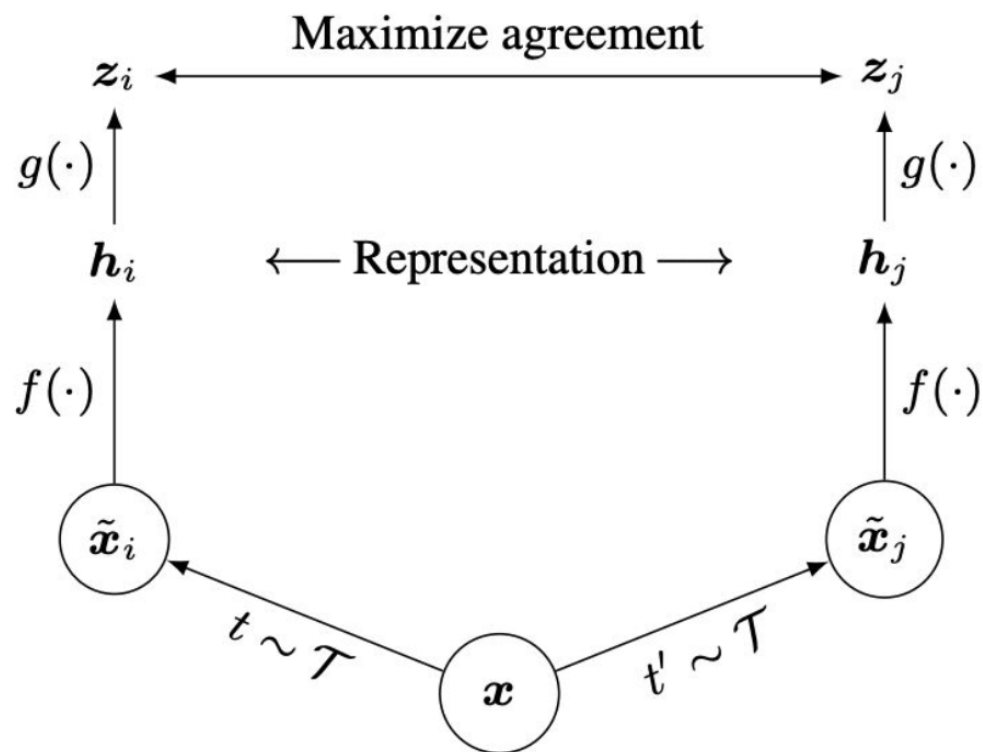
- تابع ضرر برای حالتی که یک نمونه مثبت و  $N-1$  نمونه منفی داشته باشیم:
  - شبیه به Softmax و Crossentropy است
  - به نام ضرر InfoNCE شناخته می‌شود

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



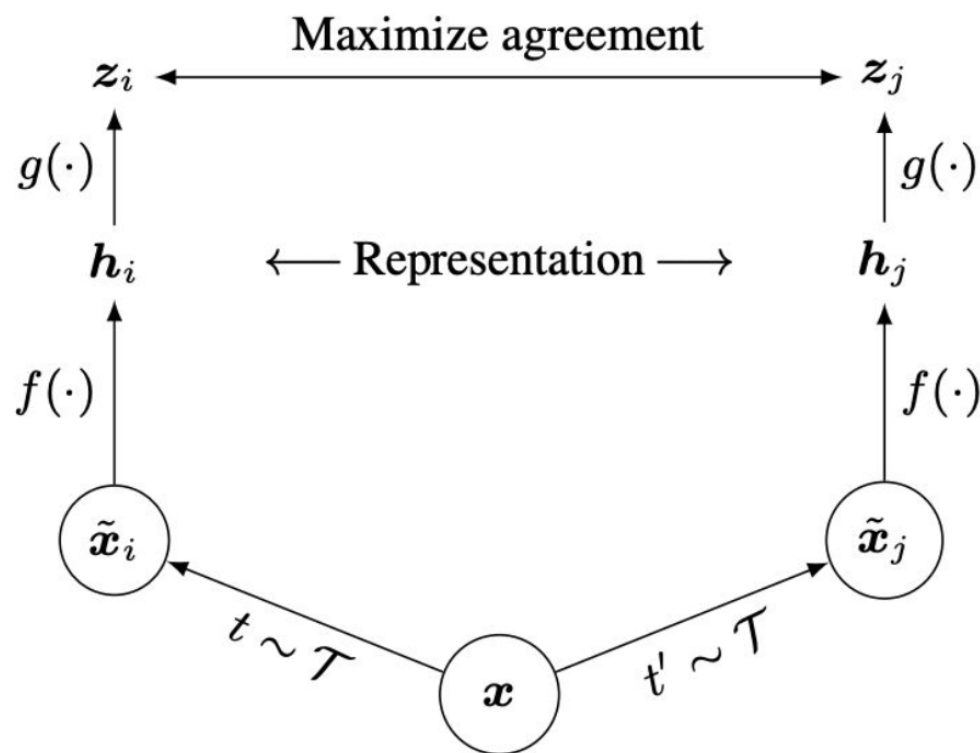
# SimCLR

- از فاصله کسینوسی به عنوان تابع امتیاز استفاده می کند
- نمونه های مثبت با استفاده از داده افزایی تولید می شوند



- از شبکه  $g$  (Projection Network) برای نگاشت ویژگی ها به فضای که مقایسه انجام شود استفاده می کند

# SimCLR




---

**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**  $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$

$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$

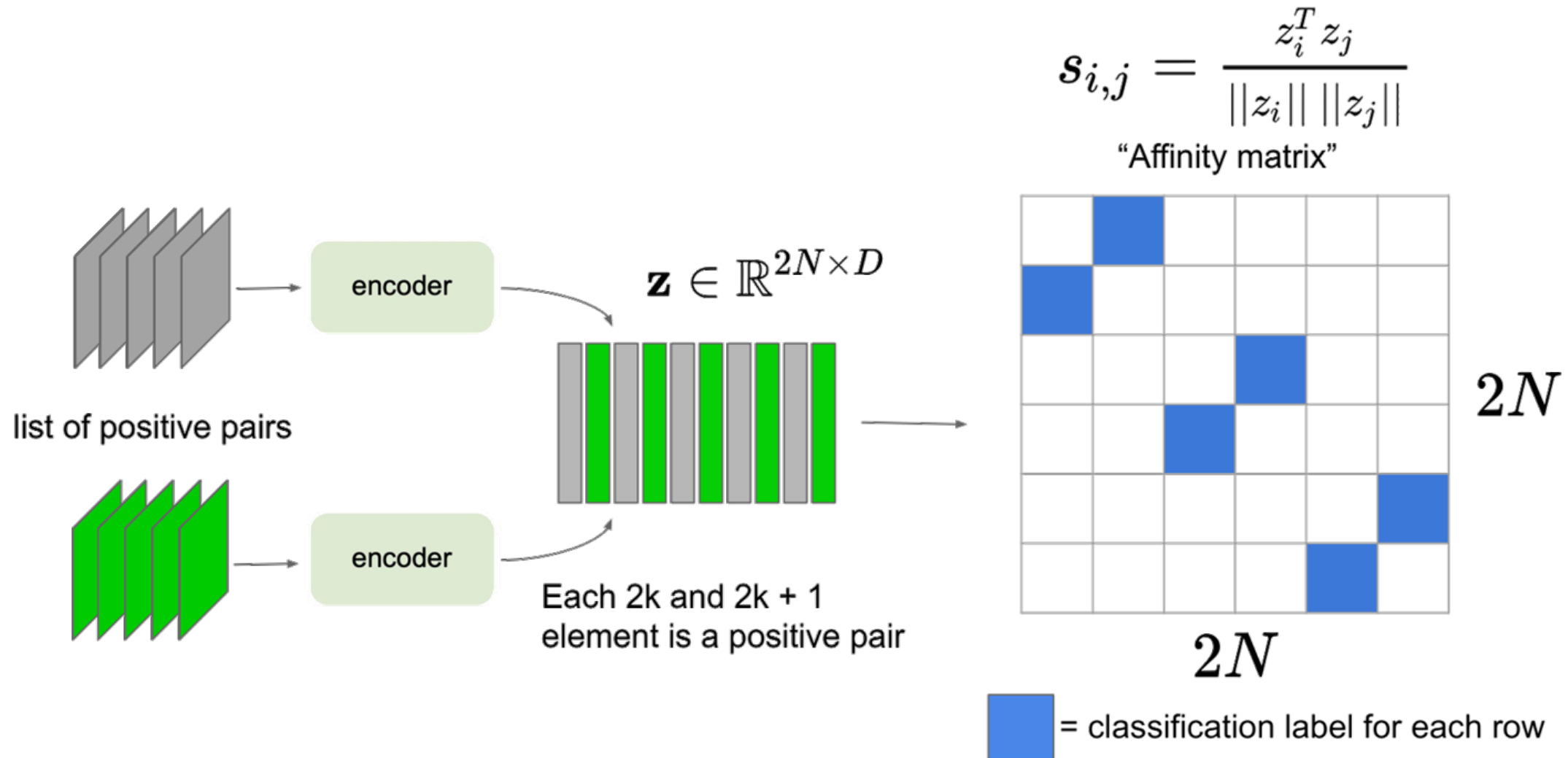
    update networks  $f$  and  $g$  to minimize  $\mathcal{L}$

**end for**

**return** encoder network  $f(\cdot)$ , and throw away  $g(\cdot)$

---

# SimCLR



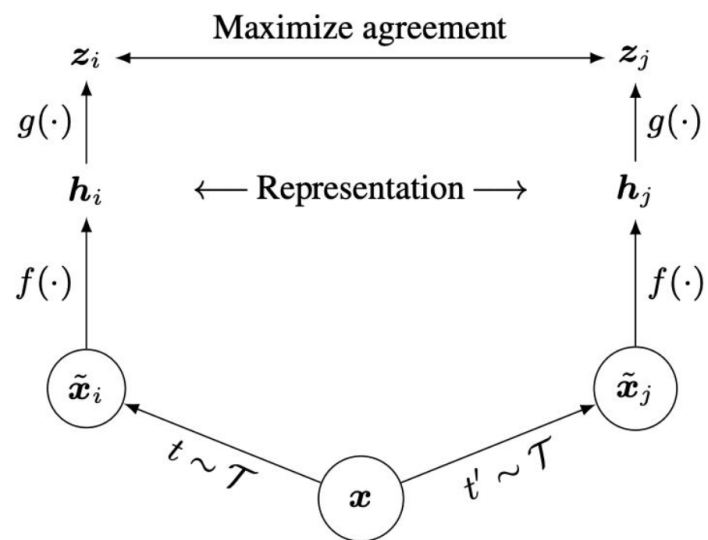
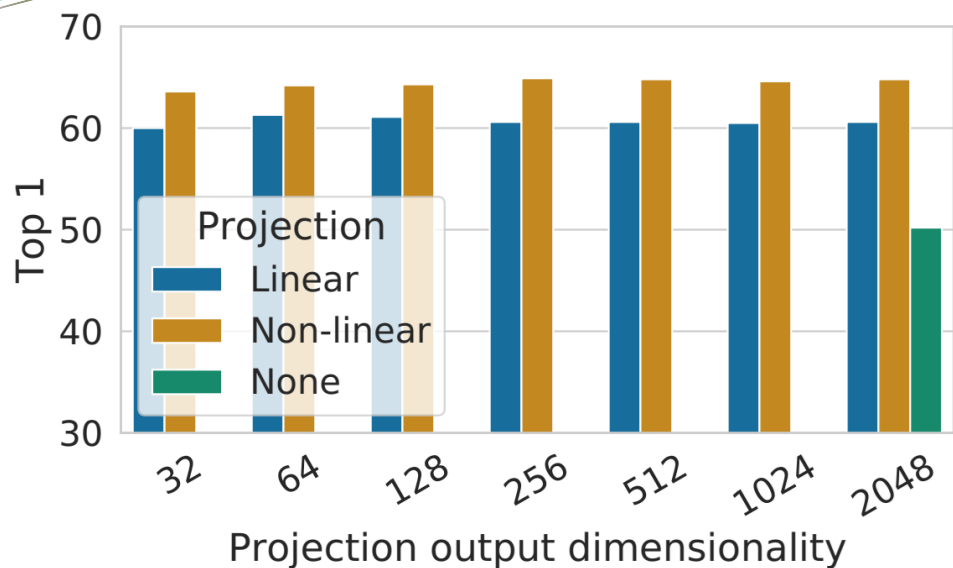
# SimCLR: یادگیری نیمه نظارتی

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.

- یادگیری بازنمایی توسط تمام تصاویر ImageNet انجام شده است
- سپس، با درصد کمی از نمونه‌های دارای برچسب تنظیم دقیق شده است

# Projection Head :SimCLR



- نگاشت خطی و غیرخطی باعث بهبود یادگیری بازنمایی شده‌اند

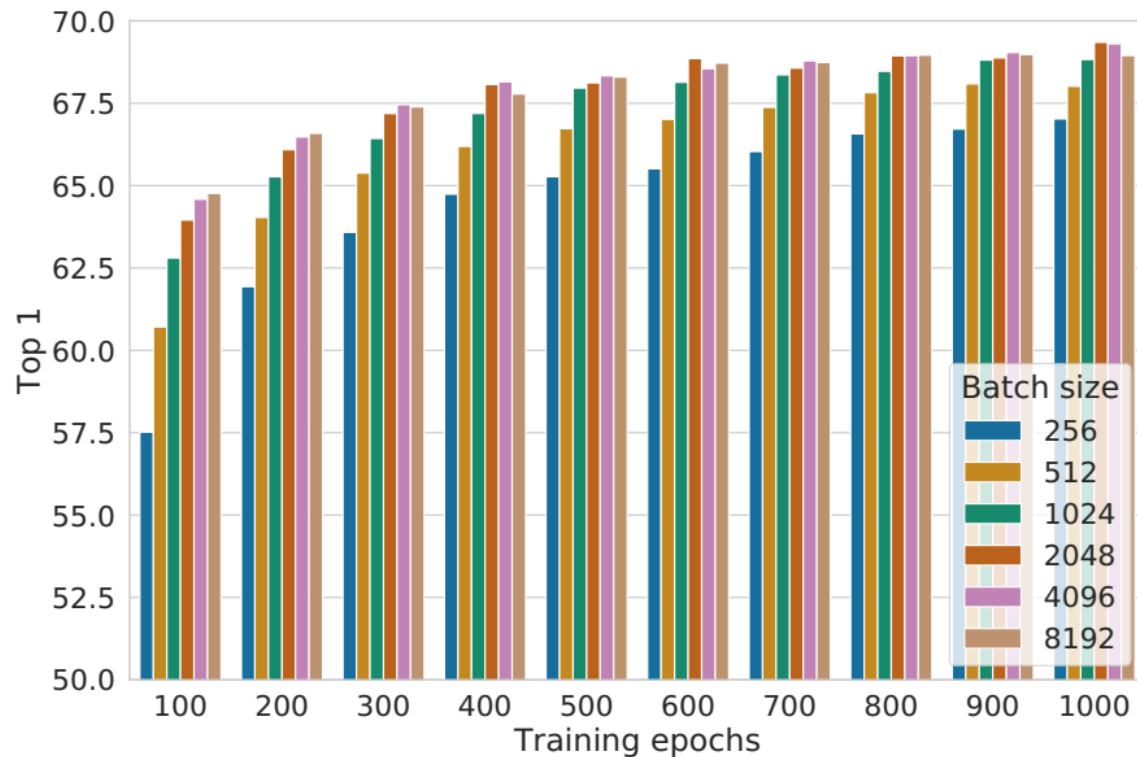
- توضیح احتمالی:

- هدف یادگیری مقابله‌ای ممکن است اطلاعات مفید را برای کارهای پایین‌دستی (downstream) حذف کند

- فضای بازنمایی  $z$  به گونه‌ای آموزش داده شده است که نسبت به تبدیل داده‌ها نامتغیر باشد

- با حذف  $g$ ، اطلاعات بیشتری را می‌توان در فضای نمایش  $h$  حفظ کرد

# Batch Size :SimCLR

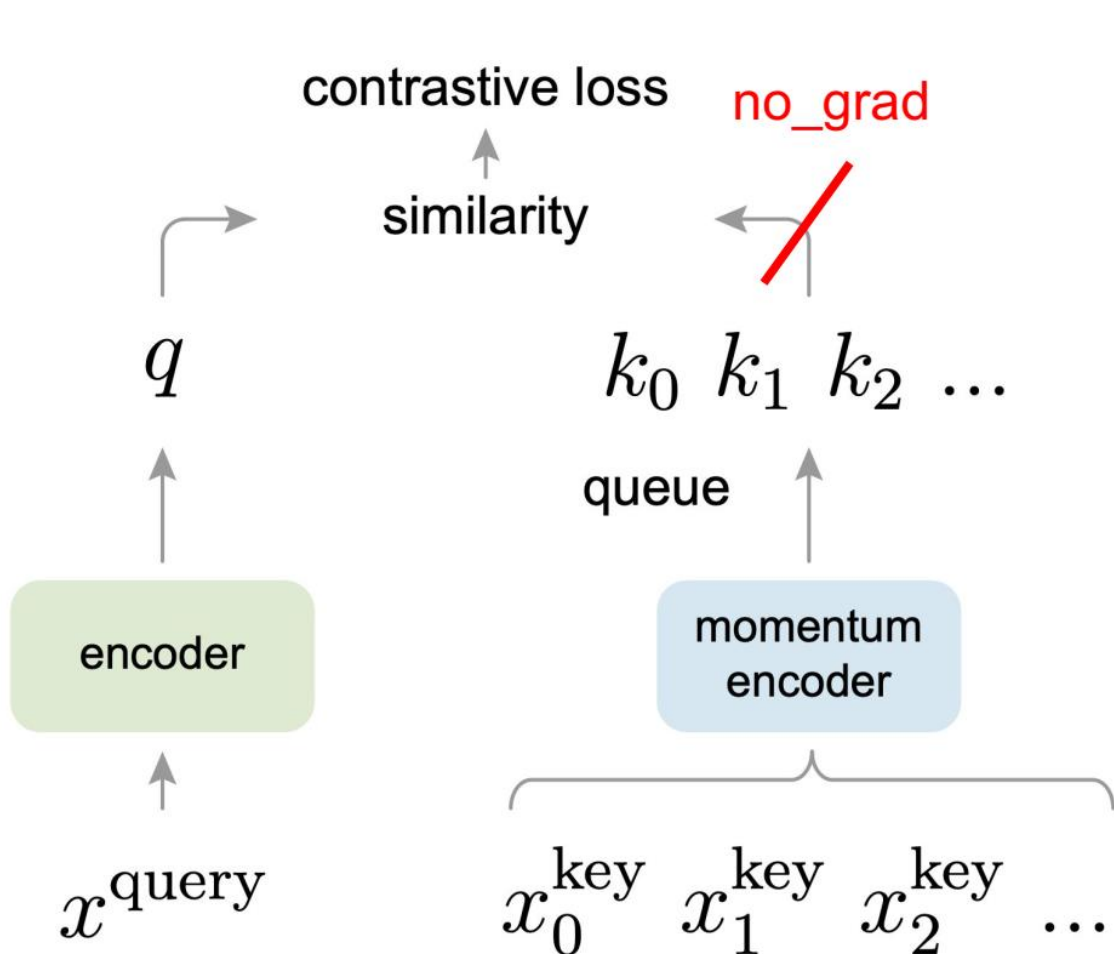


- اندازه دسته و همچنین تعداد دوره اثر زیادی در عملکرد بازنمایی‌های آموخته شده دارند
- اندازه دسته بزرگ نیاز به حافظه بسیار زیاد در زمان آموزش ایجاد می‌کند

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>



# یادگیری مقابله‌ای تکانه‌ای (MOCO)



## • تفاوت‌های کلیدی با SimCLR:

- یک صف پویا از کلیدها (نمونه های منفی) نگهداری می‌شود
- محاسبه گرادیان‌ها و به‌روزرسانی تنها برای کدگذار query انجام می‌شود
- اندازه mini-batch از تعداد کلیدها جدا می‌شود؛ می‌تواند تعداد زیادی نمونه منفی را پشتیبانی کند
- کدگذار مربوط به کلیدها به صورت آهسته و با استفاده از قاعده به‌روزرسانی تکانه‌ای به‌روز می‌شود:

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



## 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: Nx1
    k = f_k.forward(x_k) # keys: Nx1
    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
```

Generate N positive pairs by  
sampling data augmentation  
functions

Encode samples

Use the running queue of  
keys as the negative samples

InfoNCE loss

Update f\_k through  
momentum

Update the FIFO negative  
sample queue

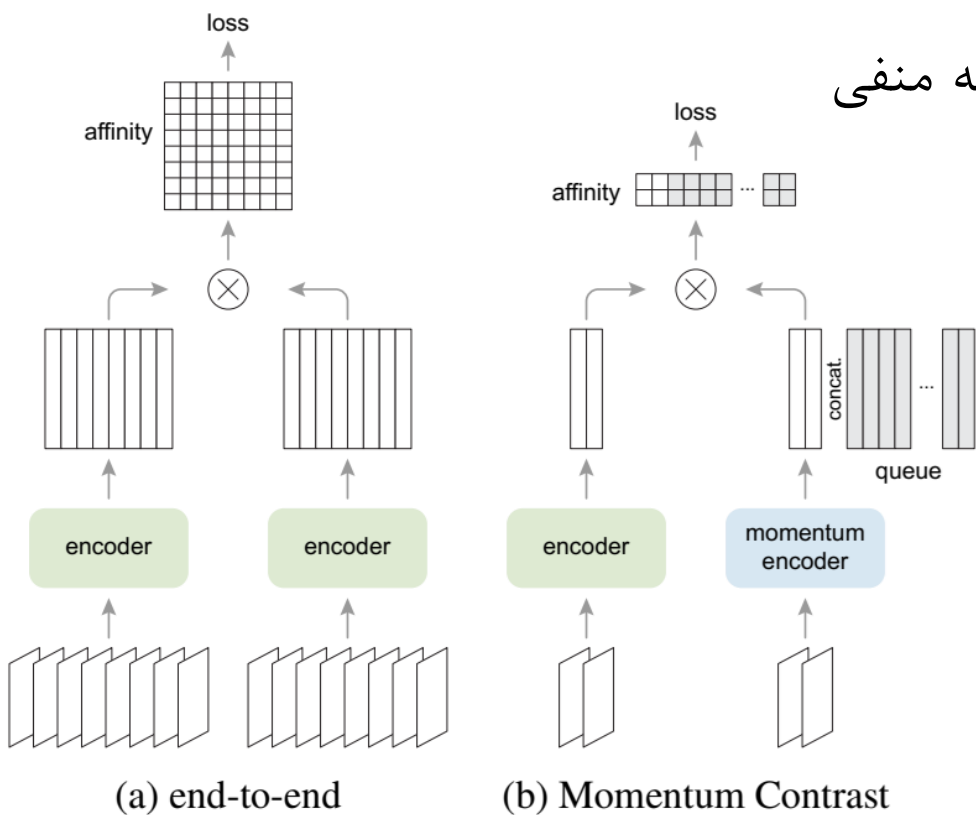
bmm: batch matrix multiplication; mm: matrix multiplication; cat: concatenation.

X

# MOCO V2

## • یک ایده ترکیبی از SimCLR و MOCO:

- از SimCLR: نگاشت غیرخطی و داده‌افزایی قوی
- از MOCO: به‌روزرسانی مبتنی بر تکانه که اجازه آموزش با تعداد نمونه منفی زیاد بر روی حافظه محدود را می‌دهد



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 V2

• یک ایده ترکیبی از SimCLR و MOCO:

- از SimCLR: نگاشت غیرخطی و داده‌افزایی قوی

- از MOCO: به‌روزرسانی مبتنی بر تکانه که اجازه آموزش با تعداد نمونه منفی زیاد بر روی حافظه محدود را می‌دهد

case	unsup. pre-train					ImageNet acc.
	MLP	aug+	cos	epochs	batch	
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>

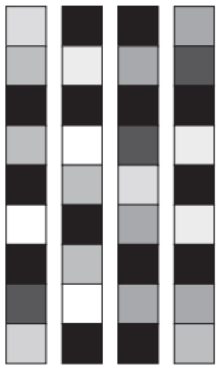
Table 2. **MoCo vs. SimCLR**: ImageNet linear classifier accuracy (**ResNet-50, 1-crop 224×224**), trained on features from unsupervised pre-training. “aug+” in SimCLR includes blur and stronger color distortion. SimCLR ablations are from Fig. 9 in [2] (we thank the authors for providing the numerical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

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.

# جانمایی کلمات (Word Embedding)

- جانمایی کلمات اطلاعات بیشتر را در ابعاد بسیار کمتری قرار می‌دهد
- این بردارها را می‌توان با استفاده از حجم زیادی از متن پیش‌آموزش داد و در مجموعه داده‌های کوچک از آنها استفاده کرد



One-hot word vectors:

- Sparse
- High-dimensional
- Hardcoded

Word embeddings:

- Dense
- Lower-dimensional
- Learned from data

# مدل زبان طبیعی

- در این مدل می‌خواهیم کلمه بعدی را پیش‌بینی کنیم

- هدف ما دستیابی به بردارهای  
جانمایی است و وزن‌های دیگر شبکه  
هدف نیستند

