With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations

Johannes Dienert

January 21, 2022

Unsupervised learning?

[1] Image source: github.com/tzutalin

Unsupervised learning?

- labeling is expensive
- ImageNet
  - 14 million samples
  - 49 thousand human annotators
- unlabelled data
  - nearly unlimited
  - free

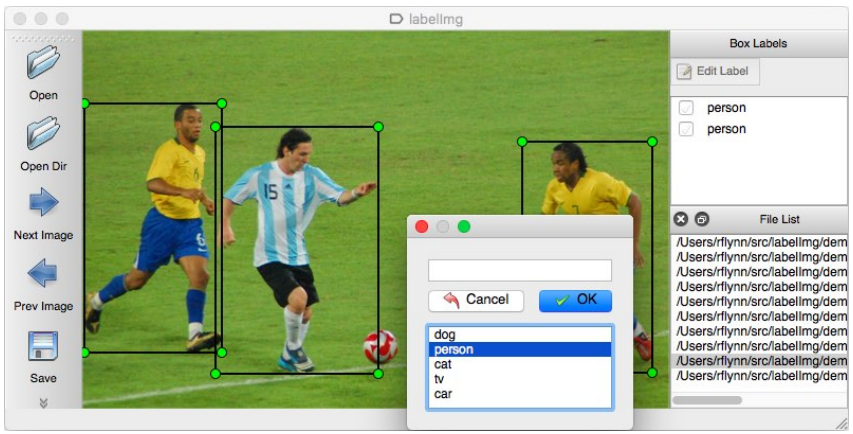


Figure: Manual labeling[1]

[1] Image source: github.com/tzutalin

Contrastive Learning

- self-supervised representation learning

Contrastive Learning

- self-supervised representation learning
- label/ground truth: 'similar' vs 'not-similar'

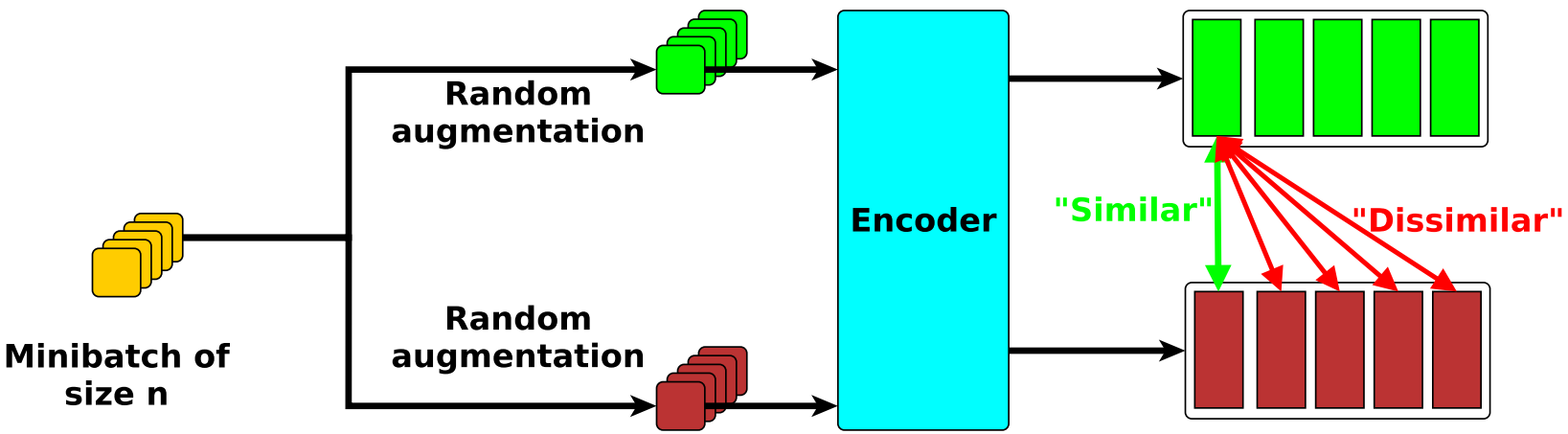


Figure: SimCLR training pipeline

Limitations

- generalization depends on augmentation

Limitations

- generalization depends on augmentation
- no positive pairs for
  - different viewpoints
  - similar objects

Better positive pairs

- beyond random augmentation

Better positive pairs

- beyond random augmentation
  - class labels
  - clustering

SimCLR

- positive pair: two random augmentations
- negative pairs: other samples from batch

SimCLR

- positive pair: two random augmentations
- negative pairs: other samples from batch
- focus on augmentation

BYOL

- no negative pairs
- two networks (online & target)

BYOL

- no negative pairs
- two networks (online & target)

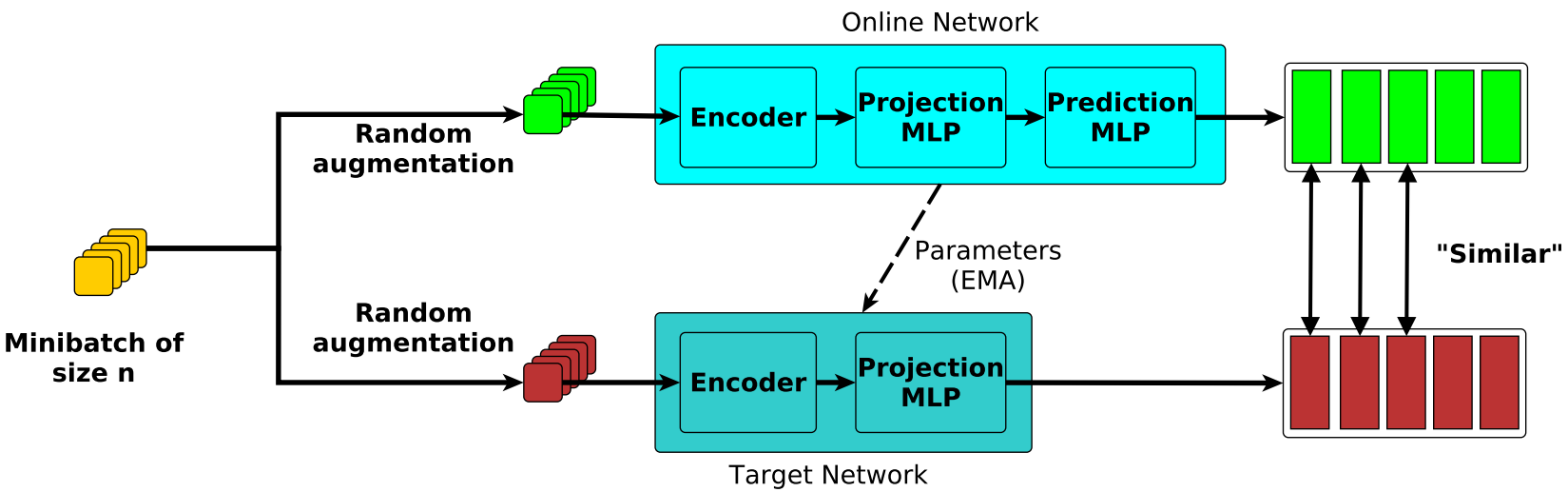


Figure: BYOL training pipeline

MoCo v1

- maintain support set (as queue)

MoCo v1

- maintain support set (as queue)

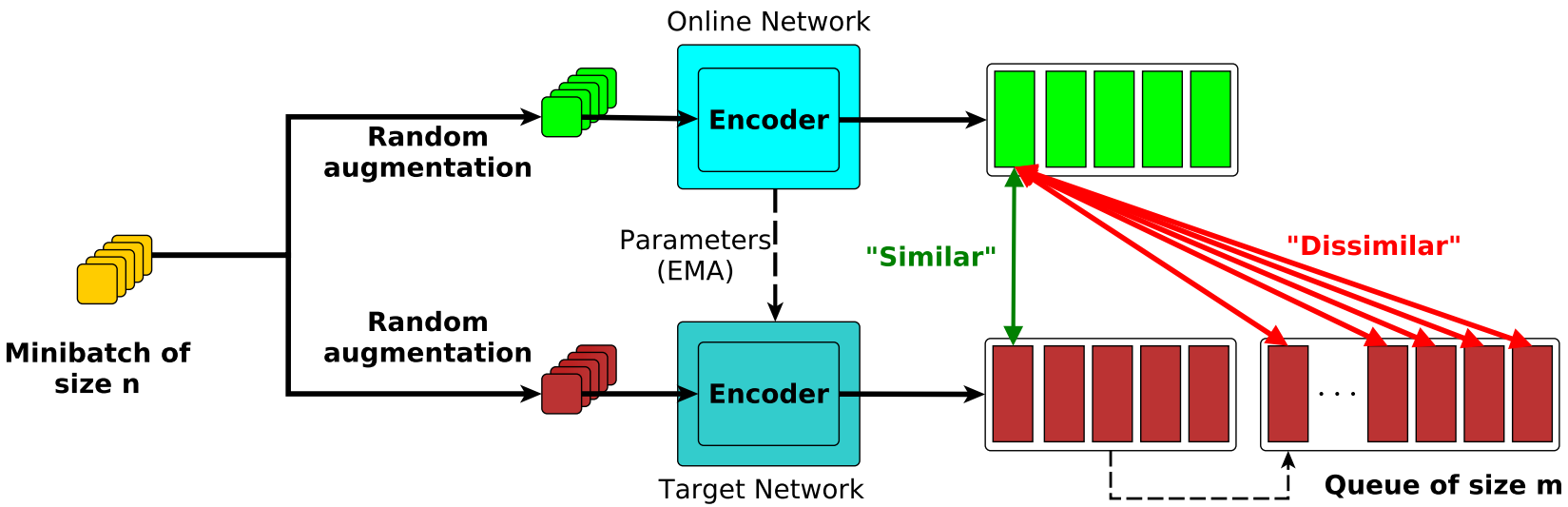


Figure: MoCo training pipeline

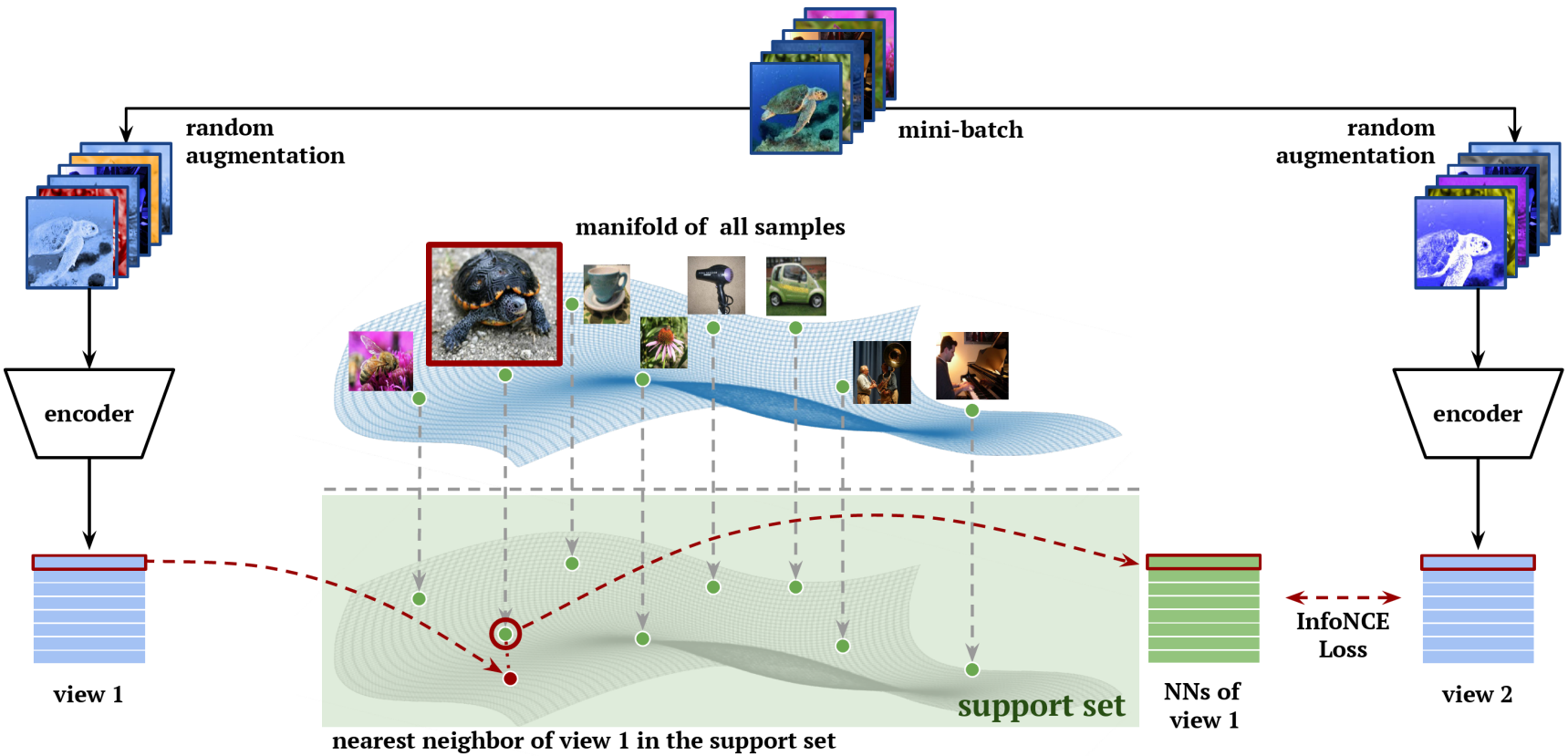
Idea

- nearest-neighbor as positive

Idea

- nearest-neighbor as positive
- compared to MoCo
  - one encoder
  - positive sample from queue

NNCLR



[1]Image credit: Dwibedi et Al.

NNCLR Training - Pseudocode

```
d = 512      # embedding dim
n = 1024     # batch size
m = 65536    # queue length
Q = queue(m, d)
```

NNCLR Training - Pseudocode

```
d = 512      # embedding dim
n = 1024     # batch size
m = 65536    # queue length
Q = queue(m, d)

for batch in loader:
        view1 = aug(batch)                    # n x 1
        view2 = aug(batch)                    # n x 1
        z = encoder.forward(view1)            # n x 1
        z_plus = encoder.forward(view2)       # n x 1
        nns = NN(z, Q)                        # n x 1
        loss = LNNCLR(nns, z_plus)
        encoder.update(loss)
        Q.update(z_plus)
```

NN Selection

$$NN(z, Q) = \arg\min_{q \in Q} ||z - q||_2$$

- $L2$ normalization
- Q: support set
- z: embedding

NNCLR Loss

$$\mathcal{L}_i^{NNCLR} = -log \frac{exp(NN(z_i, Q) \cdot z_i^+ / \tau)}{\sum_{k=1}^{n} exp(NN(z_i, Q) \cdot z_k^+ / \tau)}$$

- $L2$ normalization before dot product

Implementation details (1)

- symmetric loss $\mathcal{L}_i^{NNCLR}$:

$$-log\frac{exp(NN(z_i,Q)\cdot z_i^+/\tau)}{\sum_{k=1}^{n} exp(NN(z_i,Q)\cdot z_k^+/\tau)} - log\frac{exp(NN(z_i,Q)\cdot z_i^+/\tau)}{\sum_{k=1}^{n} exp(NN(z_k,Q)\cdot z_i^+/\tau)}$$

Implementation details (2)

- prediction head g *(optional)*
  - additional MLP g
  - process embeddings $p_i^+ = g(z_i^+)$ and $p_i = g(z_i)$

Implementation details (2)

- prediction head g *(optional)*
  - additional MLP g
  - process embeddings $p_i^+ = g(z_i^+)$ and $p_i = g(z_i)$

$$-log\frac{exp(NN(p_i, Q) \cdot p_i^+/\tau)}{\sum_{k=1}^{n} exp(NN(p_i, Q) \cdot p_k^+/\tau)} - log\frac{exp(NN(p_i, Q) \cdot p_i^+/\tau)}{\sum_{k=1}^{n} exp(NN(p_k, Q) \cdot p_i^+/\tau)}$$

Experimental setup

Experimental setup

- ResNet-50 encoder
- projection head
- embeddings $d = 256$
- batch size $bs = 4096$
- queue size *98304*

Experimental setup

- ResNet-50 encoder
- projection head
- embeddings $d = 256$
- batch size $bs = 4096$
- queue size *98304*
- cosine annealing schedule
- learning rate $lr = 0.3$
- weight-decay

ImageNet linear evaluation procedure

- self-supervised representation learning

ImageNet linear evaluation procedure

- self-supervised representation learning
- **freeze weights**
- linear classifier (supervised)

ImageNet evaluations (1)

Method	Top-1	Top-5
PIRL	63.6	-
CPC v2	63.8	85.3
MoCo v2	71.1	-
SimCLR v2	71.7	-
SwAV	71.8	N/A
InfoMin Aug.	73.0	91.1
BYOL	74.3	91.6
NNCLR	**75.4**	**92.3**
SwAV (multi crop)	75.3	N/A
NNCLR (multi crop)	**75.6**	**92.4**

Table: Comparison with other self-supervised learning methods on ResNet-50 encoder. Methods on the top section use two views only.

ImageNet evaluations (2)

Method	ImageNet 1% Top-1	ImageNet 1% Top-5	ImageNet 10% Top-1	ImageNet 10% Top-5
Supervised	25.4	48.4	56.4	80.4
PIRL	-	57.2	-	83.8
SimCLR	48.3	75.5	65.6	87.8
BYOL	53.2	78.4	68.8	89.0
NNCLR	**56.4**	**80.7**	**69.8**	**89.3**
SwAV (multi crop)	53.9	78.5	70.2	89.9

Table: **Semi-Supervised** learning results on ImageNet. Performances are reported on fine-tuning a pre-trained ResNet-50 with ImageNet 1% and 10% datasets.

Transfer learning evaluations

Method	Food101	CIFAR10	SUN397	Cars	DTD
BYOL	75.3	91.3	62.2	**67.8**	75.5
SimCLR	72.8	90.5	60.6	49.3	**75.7**
Sup.-IN	72.3	93.6	61.9	66.7	74.9
NNCLR	**76.7**	**93.7**	**62.5**	67.1	75.5

Table: Selection of the **transfer learning** evaluation results. All results reported as Top-1 classification accuracy.

Transfer learning evaluations

Method	Food101	CIFAR10	SUN397	Cars	DTD
BYOL	75.3	91.3	62.2	**67.8**	75.5
SimCLR	72.8	90.5	60.6	49.3	**75.7**
Sup.-IN	72.3	93.6	61.9	66.7	74.9
NNCLR	**76.7**	**93.7**	**62.5**	67.1	75.5

Table: Selection of the **transfer learning** evaluation results. All results reported as Top-1 classification accuracy.

- best performance in 8 / 12
- better than features from supervised learning in 11 / 12

Dependence on Augmentation

Method	SimCLR	BYOL	NNCLR
Full aug.	67.9	72.5	72.9
Only crop	40.3 (-27.6)	59.4 (-13.1)	68.2 (-4.7)

Table: Effect of limited data augmentation methods

Only my best friend?

Only my best friend?

k in Top-k NN	1	2	4	8	16	32
Top-1 perf.	**74.9**	74.1	73.8	73.8	73.8	73.2
Top-5 perf.	**92.1**	91.6	91.5	91.4	91.3	91.2

Table: Effect of randomly taking one of the k best neighbors.

Soft vs. Hard NN

- convex combination of embeddings
- weighted by similarity to z_i

NN Type	Top-1 perf.	Top-5 perf.
Soft NN	71.4	90.4
Hard NN	74.9	92.1

Table: Soft vs. Hard nearest neighbor selection

Conclusions

- nearest-neighbours to increase diversity
- state-of-the-art performance
- reduce reliance on data augmentation

Questions

- Thank you for your audience!

Pure effect of NN

Mom. Enc.	Positive sample	Top-1 perf.	Top-5 perf.
No	View 1	71.4	90.4
No	NN of View 1	**74.5**	**91.9**
Yes	View 1	72.5	91.3
Yes	NN of View 1	**74.9**	**92.1**

Table: Effect of using the nearest-neighbors as positives

Support set size

Queue size	8192	16384	32768	65536	98304
Top-1 perf.	73.6	74.2	74.9	75.0	**75.4**
Top-5 perf.	91.2	91.7	92.1	92.2	**92.3**

Table: Effect of different sized support set (queue length)

Class of my NN

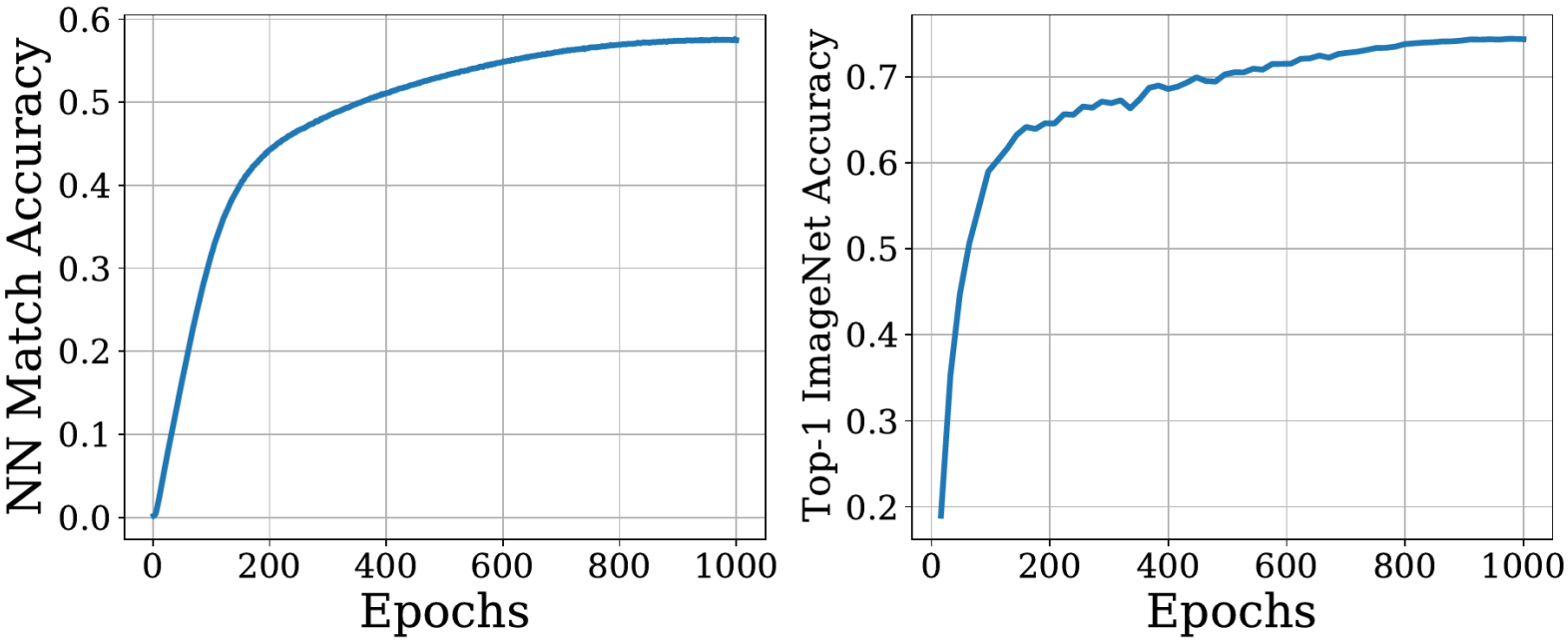


Figure: Accuracy of the NN belonging to the same class vs. Top-1 Accuracy

MoCo vs MoCo v2

- projection head: replaced 1 layer MLP by 2 layers with ReLu
- data augmentation: added blurring
- learning schedule: cosine

Embedding size

d	128	256	512	1024	2048
Top-1 perf.	74.9	74.9	74.8	74.9	**74.6**
Top-5 perf.	92.1	92.1	92.0	92.0	**92.0**

Table: Effect of embedding dimensionality d

Prediction Head

Prediction MLP	Top-1	Top-5
No	74.5	92.0
Yes	74.9	92.1

Table: Prediction head provides a small boost