26TH INTERNATIONAL CONFERENCE ON MEDICAL IMAGE COMPUTING AND COMPUTER ASSISTED INTERVENTION
8-12 OCTOBER 2023
VANCOUVER / CANADA

Additional Positive Enables Better Representation Learning for Medical Images

Dewen Zeng[1], Yawen Wu[2], Xinrong Hu[1], Xiaowei Xu[3], Jingtong Hu[2], Yiyu Shi[1]

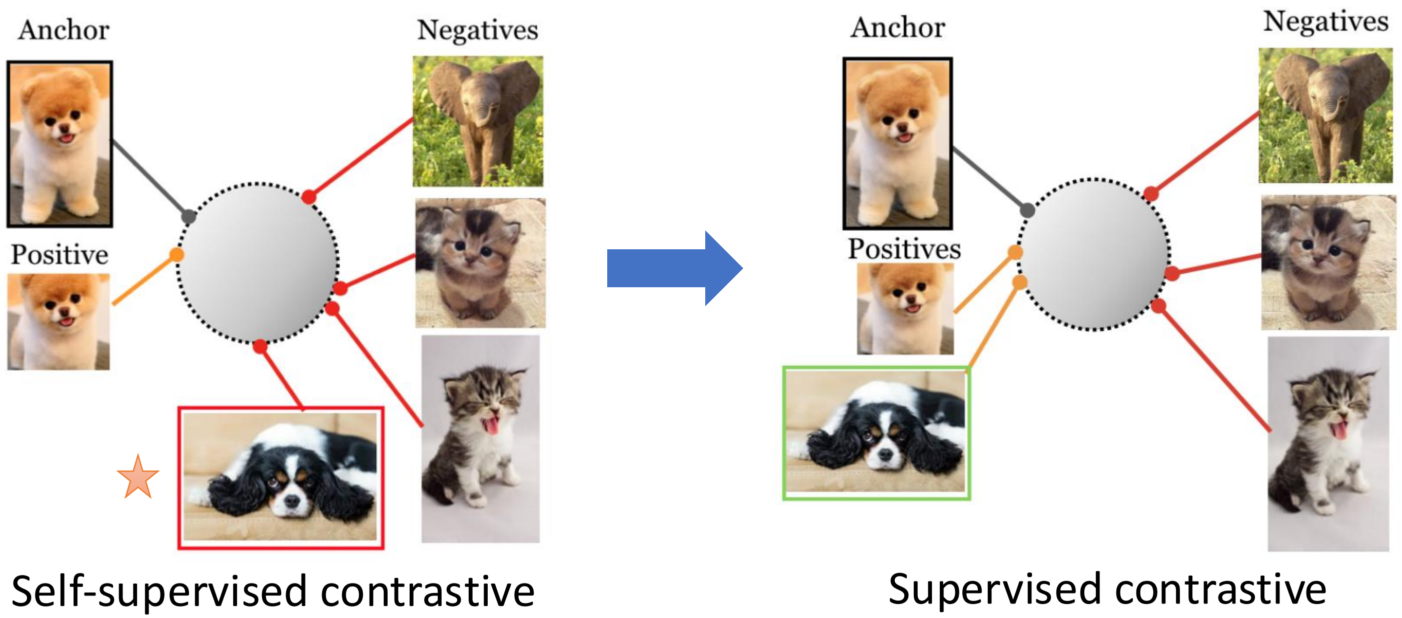
[1] University of Notre Dame

[2] University of Pittsburgh

[3] Guangdong Provincial People's Hospital

Limitations in SOTA self-supervised learning framework

- ❑ Positive pairs are generated using data augmentations
- ❑ Instance with the same label from other images are not utilized



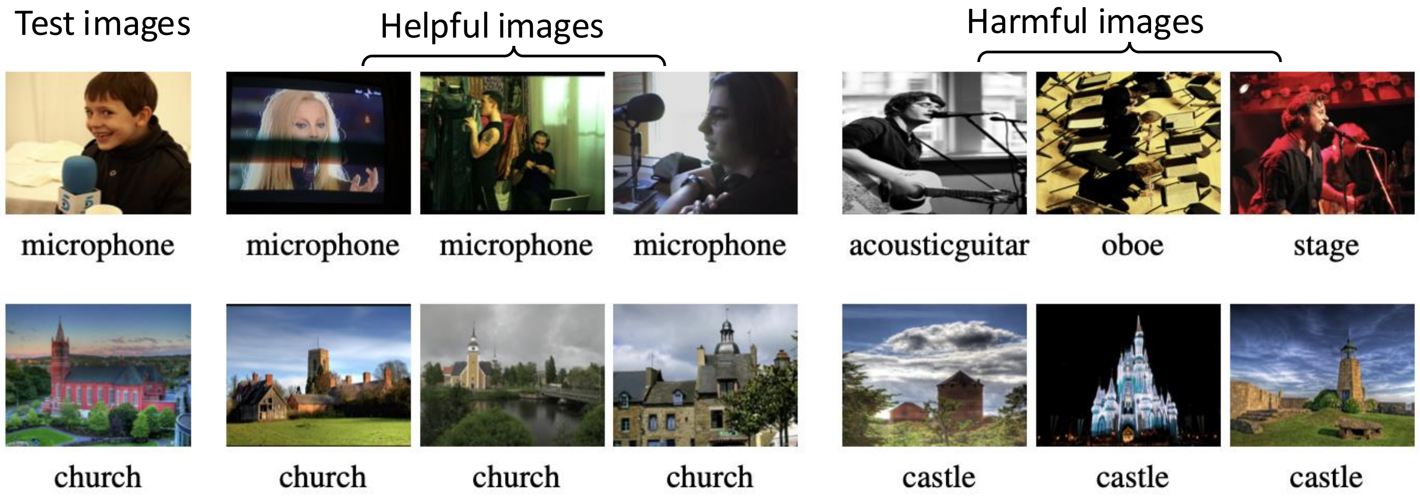
Can we identify such positives without label?

Influence Function: TracIn

- Measure how the loss of a test sample changes with the training process of a training sample
- A **positive** TracIn value means **helpful** example that reduces loss, a **negative** TracIn value means **harmful** example that increases loss

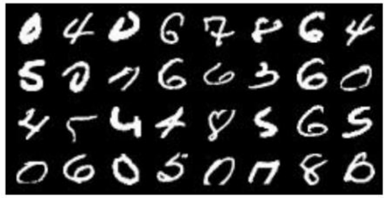
$$\mathrm{TracIn}(x_i, x_k) = \sum_{t: x_t = x_i}^{\overbrace{T}^{\text{total number of iterations}}} \overbrace{\eta_t}^{\text{learning rate at iteration } t} \underbrace{\nabla \ell(w_t, x_k)}_{\text{gradient of sample } x_k \text{ at iteration } t} \cdot \nabla \ell(w_t, x_i).$$

Model interpretation

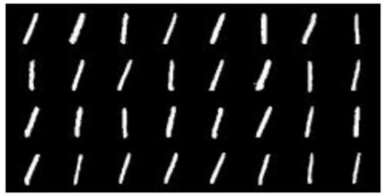


Mislabel identification (self influence)

Highest TracIn
MNIST

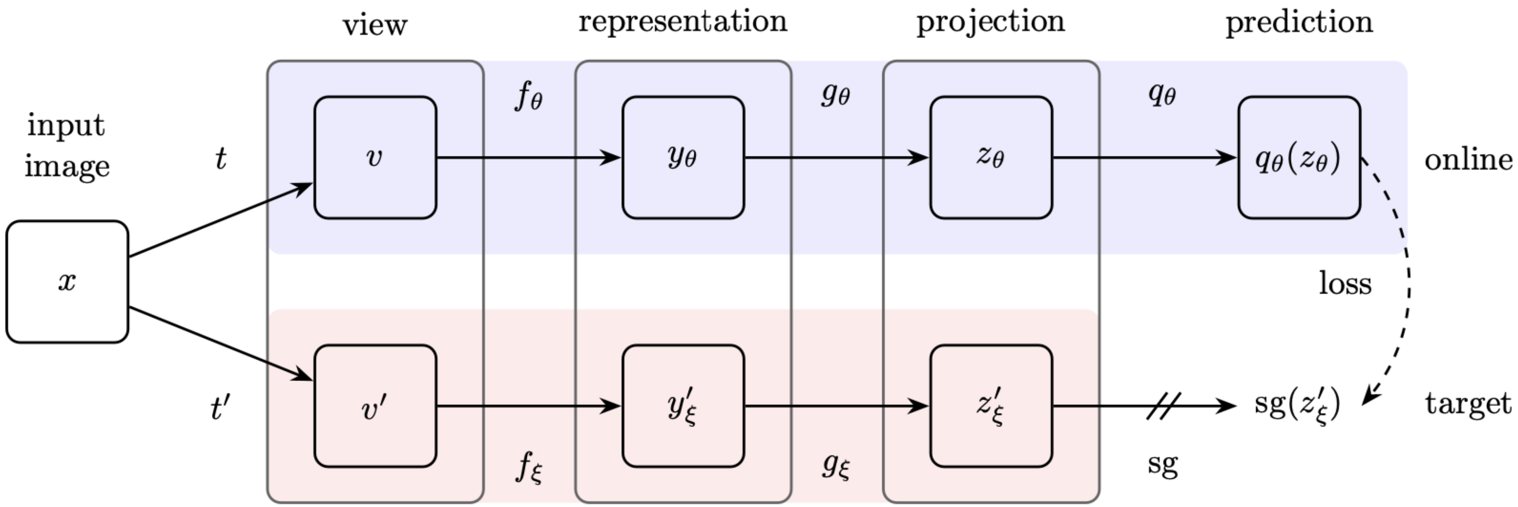


Lowest TracIn
MNIST



BYOL: Bootstrap your own latent

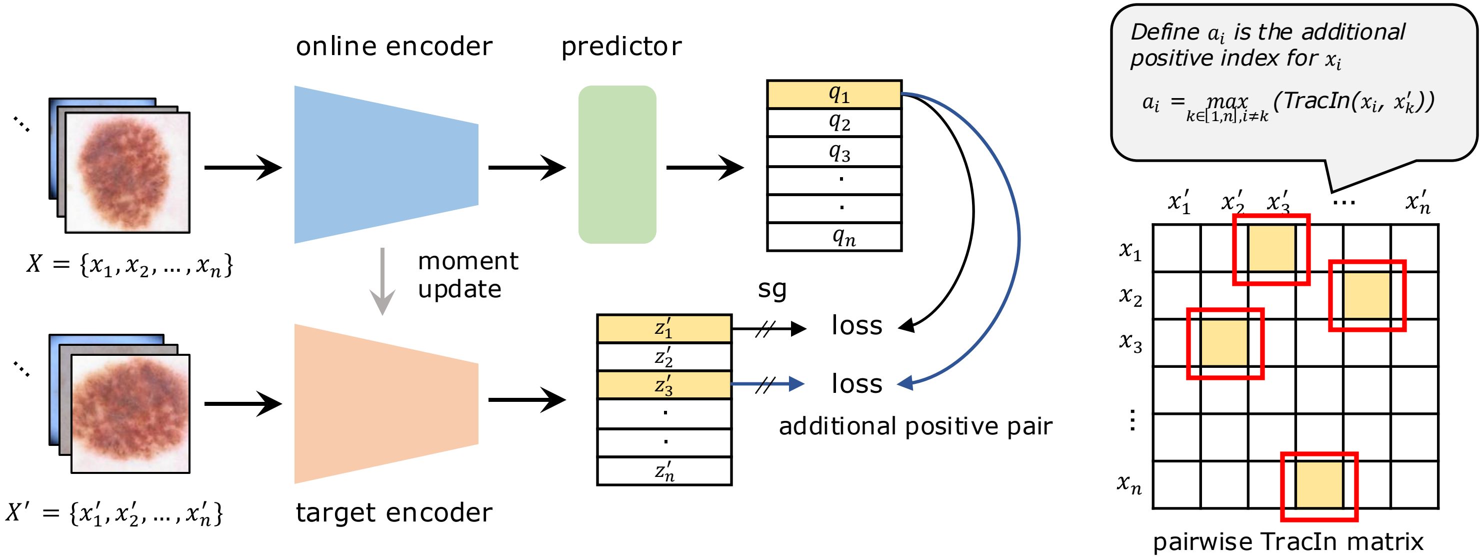
- ❑ Only positive pair is required (no negatives)
- ❑ Easy to compute per-sample gradient



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

BYOL-TracIn

- Use BYOL loss function without label
- Selecting another sample with the largest TracIn value
- Use the gradient of the last linear layer to estimate (save resources)



Efficient per-sample gradient

- TracIn requires the gradient of each sample.
- PyTorch or Tensorflow do not support batch-wise per-sample gradient

For linear layer:

linear layer input

$$\underbrace{q}_{\text{output logits}} = \underbrace{W}_{\text{weights}} \overbrace{a}^{\text{linear layer input}}$$

loss function

$$\nabla_W \overbrace{f(q)}^{\text{loss function}} = \nabla_q f(q) \cdot a^T$$

target network output (constant)

$$f(q) = 2 - 2 \cdot \frac{\langle q, \overbrace{z}^{\text{target network output (constant)}} \rangle}{(\|q\|_2 \cdot \|z\|_2)}$$

gradient

$$\nabla_q f(q) = 2 \cdot \left(\frac{\langle q, z \rangle \cdot q}{(\|q\|_2^3 \cdot \|z\|_2)} - \frac{z}{\|q\|_2 \cdot \|z\|_2}\right)$$

$$\begin{aligned} \text{TracIn}(x_i, x_k) &\approx \eta_t \nabla_W f(q_i) \cdot \nabla_W f(q_k) \\ &= \eta_t \left(\nabla_q f(q_i) a_i^T\right) \cdot \left(\nabla_q f(q_k) a_k^T\right) \\ &= \eta_t \left(\underbrace{\nabla_q f(q_i)}_{\text{gradient of the logits}} \cdot \nabla_q f(q_k)\right) \cdot \underbrace{(a_i}_{\text{linear layer input}} \cdot a_k) \end{aligned}$$

We only need to save the inputs and outputs of the linear layer in one forward operation

Semi-supervised results

- Dataset: ISIC 2019 (25,331 images), ChestX-ray (108,948 images)
- Our method: **BYOL-TracIn**, upper bound: **BYOL-Sup**
- BYOL-TracIn-pretrained: using a pretrained model for computing TracIn

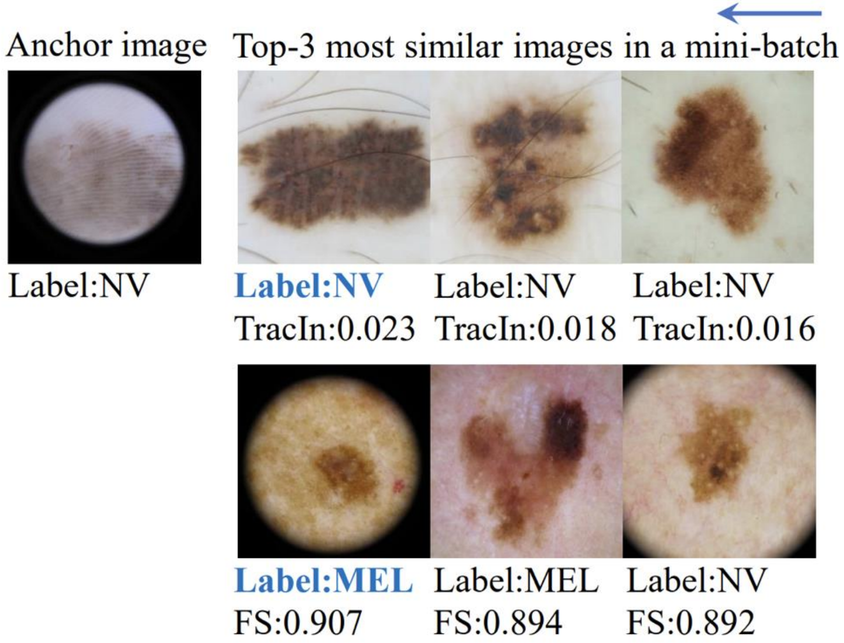
Method	ISIC 2019 10%	ISIC 2019 50%	ISIC 2019 100%	ChestX-ray 10%	ChestX-ray 50%	ChestX-ray 100%
	BMA ↑			AUC ↑		
Random	0.327(.004)	0.558(.005)	0.650(.004)	0.694(.005)	0.736(.001)	0.749(.001)
BYOL [12]	0.399(.001)	0.580(.006)	0.692(.005)	0.699(.004)	0.738(.003)	0.750(.001)
FNC [17]	0.401(.004)	0.584(.004)	0.694(.005)	0.706(.001)	0.739(.001)	0.752(.002)
FT [30]	0.405(.005)	0.588(.008)	0.695(.005)	0.708(.001)	0.743(.001)	0.751(.002)
FS	0.403(.006)	0.591(.003)	0.694(.004)	0.705(.003)	0.738(.001)	0.752(.002)
FS-pretrained	0.406(.002)	0.596(.004)	0.697(.005)	0.709(.001)	0.744(.002)	0.752(.002)
BYOL-TracIn	0.403(.003)	0.594(.004)	0.694(.004)	0.705(.001)	0.742(.003)	0.753(.002)
BYOL-TracIn -pretrained	**0.408(.007)**	**0.602(.003)**	**0.700(.006)**	**0.712(.001)**	**0.746(.002)**	**0.754(.002)**
BYOL-Sup	0.438(.006)	0.608(.007)	0.705(.005)	0.714(.001)	0.748(.001)	0.756(.003)

baselines

Transfer learning results

- Dataset: ISIC 2016 (900 images), Shenzhen (662 images)
- FS: Feature Similarity

Method	ISIC 2016 Precision ↑	Shenzhen AUC ↑
Random	0.400(.005)	0.835(.010)
BYOL [12]	0.541(.008)	0.858(.003)
FNC [17]	0.542(.007)	0.862(.006)
FT [30]	0.559(.011)	0.876(.005)
FS	0.551(.003)	0.877(.004)
FS-pretrained	0.556(.004)	0.877(.006)
BYOL-TracIn	0.555(.012)	0.880(.007)
BYOL-TracIn -pretrained	**0.565(.010)**	**0.883(.001)**
BYOL-Sup	0.592(.008)	0.893(.006)



Conclusions

1. BYOL-TracIn provides a way to select additional positive pair for self-supervised pre-training, which could increase the diversity of features in positive pairs.
2. BYOL-TracIn shows significant improvements in both semi-supervised and transfer learning settings