Supplementary material

Proof of Proposition 1 Let $\mathbf{Y}^* = \hat{y}(\mathbf{S}_m)$ be the solution of the optimal linear assignment problem given the similarity matrix $\mathbf{S}_m$. With this, we can rewrite $L(\mathbf{S}_m, \mathbf{Y}_{gt})$ as follows:

$$L(\mathbf{S}_m, \mathbf{Y}_{gt}) = \sum_i \sum_j ([\mathbf{Y}_{gt}]_{ij} - [\mathbf{Y}^*]_{ij})[\mathbf{S}_m]_{ij} = \sum_i \sum_j q_{ij}[\mathbf{S}_m]_{ij} \tag{1}$$

Note that the term q_{ij} can only take the values in $\{-1, 0, 1\}$ as both matrices $\mathbf{Y}_{gt}$ and $\mathbf{Y}^*$ are binary. Consider the case when $q_{ij} = 0$, which indicates that the objects i and j are matched correctly and that $[\mathbf{S}]_{ij} + m \leq [\mathbf{S}]_{ik}$ for any $k \neq j$. The latter implies that the distance between the positive pair (i, j) is at least m smaller than the distance with all other pairs in the batch.

Next, note that the optimal value of the loss function $L(\mathbf{S}_m, \mathbf{Y}_{gt}) = 0$ is achieved when all q_{ij} equate to zero, which implies that at the optimum, distances in all positives pairs are at least m smaller than the distances in all negative pairs. This optimality condition reassembles the formulation of margin Triplet loss [6]. □

Proof of Proposition 2 Recall that the batch-hard mining emerges from relaxing constraint $\mathbf{Y} \in \mathit{\Pi}$ into $\mathbf{Y} \in \mathcal{R}$, where $\mathcal{R}$ is a set of binary row-stochastic matrices. We can rewrite $L_\tau(\mathbf{S}_m, \mathbf{Y}_{gt})$ with batch-hard mining as:

$$\begin{aligned} \hat{L}_\tau(\mathbf{S}, \mathbf{Y}_{gt}) &= \sum_{(p,k)\in\mathbf{Y}_{gt}} [\mathbf{S}]_{pk} + \tau \sum_i log\big(\sum_j exp(-\frac{1}{\tau}[\mathbf{S}]_{ij})\big) \\ &= \sum_{(p,k)\in\mathbf{Y}_{gt}} \phi(p,k) + \tau \sum_i log\big(\sum_j exp(-\frac{1}{\tau}\phi(i,j))\big) \end{aligned} \tag{2}$$

where $\phi(p, k)$ is a distance measure between the samples p and k, and τ is the parameter controlling a degree of smoothness. Note that in (2) we assume that the distance $\phi(p, k)$ decreases when the similarity between the samples increases.

The first and the second terms in (2) relates to the alignment and distribution terms [3] and reassemble the formulation of the normalized temperature cross entropy loss (InfoNCE) [9, 1] up to a normalization constant. □

Formulations of contrastive losses with batch-hard mining

Here we present formulations of the contrastive losses with batch-hard mining strategy. We show the derivation for the case of the structured linear assignment loss, while the derivations for the smoothed structured linear assignment and SparseCLR losses can be done analogously.

We start from the original structured linear assignment loss with one-to-one mining:

$$L(\mathbf{S}, \mathbf{Y}_{gt}) = tr(\mathbf{S}\mathbf{Y}_{gt}^T) - \min_{\mathbf{Y} \in \mathit{\Pi}} tr(\mathbf{S}\mathbf{Y}^T) \tag{3}$$

To derive the loss under the batch-hard mining strategy, the structural constraint $\mathbf{Y} \in \mathit{\Pi}$ is relaxed to $\mathbf{Y} \in \mathcal{R}$, where $\mathcal{R}$ is a set of row-stochastic binary matrices, i.e. $[\mathbf{Y}]_{ij} \in \{0, 1\}$ for $\forall(i, j)$ and $\sum_j [\mathbf{Y}]_{ij} = 1$ for $\forall i$. With this, the structured linear assignment loss with the batch-hard mining is defined as follows:

$$\begin{aligned}
\hat{L}(\mathbf{S}, \mathbf{Y}_{gt}) &= tr(\mathbf{S}\mathbf{Y}_{gt}^T) - \min_{\mathbf{Y} \in \mathcal{R}} tr(\mathbf{S}\mathbf{Y}^T) \\
&= tr(\mathbf{S}\mathbf{Y}_{gt}^T) - \min_{u_1 \ldots u_N} \sum_i (\sum_j [\mathbf{S}]_{ij} [u_i]_j) \\
&= tr(\mathbf{S}\mathbf{Y}_{gt}^T) - \sum_i \min_{u_i} (\sum_j [\mathbf{S}]_{ij} [u_i]_j) \\
&= tr(\mathbf{S}\mathbf{Y}_{gt}^T) - \sum_i \min_j [\mathbf{S}_{ij}]
\end{aligned} \tag{4}$$

where the first inequality follows from the fact that the rows $u_1 \ldots u_N$ of $\mathbf{Y} \in \mathcal{R}$ are independent of each other, and the last inequality is due to u_i is a binary vector containing a one-hot encoding of the minimum index.

▶ Smoothed structured linear assignment loss $L_\tau(\mathbf{S}, \mathbf{Y}_{gt})$ with batch-hard-mining:

$$\hat{L}_\tau(\mathbf{S}, \mathbf{Y}_{gt}) = tr(\mathbf{S}\mathbf{Y}_{gt}^T) + \tau \sum_i \log \sum_j (-\frac{1}{\tau} [\mathbf{S}]_{ij}) \tag{5}$$

▶ SparseCLR contrastive loss $L_{sparse}(\mathbf{S}, \mathbf{Y}_{gt})$ with batch-hard mining:

$$\hat{L}_{sparse}(\mathbf{S}, \mathbf{Y}_{gt}) = tr(\mathbf{S}\mathbf{Y}_{gt}^T) - \frac{1}{2} \sum_i \sum_{j \in \Omega(h_i)} \left([h_i]_j^2 - \mathcal{T}^2(-h_i) \right) \tag{6}$$

where $h_i \in \mathbb{R}^N$ correspond to a i-th row of the similarity matrix $\mathbf{S}$, and $\Omega(X) = \{j \in X : \ sparsemax(X)_j > 0\}$ is the support of *sparsemax* function [7]. The thresholding operator $\mathcal{T}$ is defined as:

$$\mathcal{T}(z) = \frac{(\sum_{j \in \Omega(z)} z_j) - 1}{|\Omega(z)|} \tag{7}$$

And the *sparsemax* function is defined as:

$$sparsemax(z) = \underset{p \in \mathit{\Delta}^{N-1}}{\operatorname{argmin}} \ ||p - z||^2 \tag{8}$$

with $\mathit{\Delta}^{N-1}$ being $N - 1$ dimensional simplex.

QARe for a positive similarity measure

In the derivations in the main paper, we adopt the metric notation commonly used in assignment problems, where the maximum similarity between objects is indicated by the minimum distance between their representations (e.g. Euclidean distance). In practice, however, the task may require to use of other types of similarity measures, for which the opposite holds (e.g. cosine distance). For this case, the derivation takes the analogous path, but with a switch of signs in Equation **??** and with min replaced by max in the LAP/QAP objectives. With this, the structured quadratic assignment loss equates to:

$$\begin{aligned} L_{QAP} &= \max_{\mathbf{Y}\in\Pi}\big\{tr(\mathbf{S}\mathbf{Y}^T) + tr(\mathbf{S}_{\mathcal{A}}\mathbf{Y}\mathbf{S}_{\mathcal{B}}^T\mathbf{Y}^T)\big\} - tr(\mathbf{S}\mathbf{Y}_{gt}) \\ &\leq \langle\lambda_{\mathcal{A}},\lambda_{\mathcal{B}}\rangle_+ + \underbrace{\max_{\mathbf{Y}\in\Pi}\big\{tr(\mathbf{S}\mathbf{Y}^T)\big\} - tr(\mathbf{S}\mathbf{Y}_{gt})}_{structured\ linear\ assignment\ loss} \end{aligned} \tag{9}$$

where $\langle\lambda_{\mathcal{A}},\lambda_{\mathcal{B}}\rangle_+$ corresponds to a maximum dot product between eigenvalues of the matrices S_A and S_B.

Practically, modifying the QARe computation presented in Algorithm 1 in the main paper from the Euclidean distance to the cosine similarity requires 3 steps: (i) similarities are scaled to be non-negative, (ii) computing a maximum dot product instead of a minimum dot product, (iii) the maximum dot product of eigenvalues is added to a contrastive objective. Note that step (i) is extra compared to the Euclidean distance case and is needed because the QAP formulation requires non-negative distances as an input. The algorithm is summarized in Algorithm 2.

Experiments

Time and memory complexity

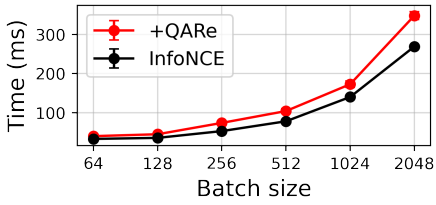

Fig. 1. A training iteration time complexity of InfoNCE with and without the quadratic assignment regularization.

In addition to computational complexity, we provide an empirical analysis of how the proposed regularization influences the time and memory complexity

of a baseline method (InfoNCE). We observe 13% increase in training time for the batch size of 256, and up to 29% for the batch size of 2048 (Figure 1). The increase in memory consumption is negligible (+0.78% for the batch size of 2048).

Instance matching

Augmentations. During the training, we apply a simple augmentation strategy: horizontal flip with a 50% chance and color jittering. For the latter, the brightness, contrast, saturation, and hue are sampled from a uniform distribution $U[0, 0.1]$.

Hyper-parameters of the losses. We use 4 baseline contrastive losses: margin Triplet [6], InfoNCE [9], NT-Logistic [8] and the proposed SparseCLR losses, which we extend with the Quadratic Assignment Regularization (QARe). The margin parameter for Triplet loss is set to 0.5, the temperature parameters for InfoNCE and NT-Logistic are both set to 0.05. We combine QARe and the backbone contrastive learning loss by taking their convex combination, where the QARe term is weighted by the constant $\beta \geq 0$. The exact values of β are 0.4/0.5/0.2/0.3 for margin Triplet / InfoNCE / NT-Logistic / SparseCLR.

Self-supervised classification

Encoder architectures. We use two models: Conv-4 and ResNet-32. The Conv-4 model involves 4 blocks. The first 3 blocks consist of 8, 16, and 32 feature maps respectively. Each performs a convolution with a kernel size of 3, a stride of 1, and a padding of 1 pixel, followed by a batch-normalization layer, a ReLU, and an average pooling layer with a kernel size of 2 and a stride of 2 pixels. The fourth block performs the same operations but instead of an average pooling, an adaptive average pooling is used. For ResNet-32, we use off-the-shelf Pytorch implementation as described in [5]. We initialize the model using standard Xavier initialization [4] and set batch-normalization weights and biases to 1 and 0 respectively.

Augmentations. For self-supervised training we apply the following augmentations to images: horizontal flip with a 50% chance, random crop-resize, grayscale conversion with a 20% chance, and color jitter with an 80% chance. Random crop-resize consists of cropping the given image from 0.08 to 1.0 of the original size, then changing its aspect ratio from 3/4 to 4/3 of the original, and finally resizing

	Conv-4			**ResNet-32**		
Method	**CIFAR-10**	**CIFAR-100**	**tiny-ImageNet**	**CIFAR-10**	**CIFAR-100**	**tiny-ImageNet**
(β) SimCLR+QARe	0.5	0.375	0.875	1.125	1.125	1.125
(β) SparseCLR+QARe	0.125	1.25	0.5	1	0.875	1.25

Table 1. QARe weighting for the architectures and the datasets for self-supervised classification experiment.

to input shape using a bilinear interpolation. For color jitter, the brightness, contrast and saturation are sampled from $U[0, 0.8]$, and the hue is sampled from $U[0, 0.2]$.

Hyper-parameters of the losses. We combine quadratic assignment regularization and the backbone contrastive learning loss by summing with weighting the QARe term by a constant $\beta \geq 0$. The values of β for each architecture and dataset are listed in Table 1. The temperature parameter for both SimCLR [2] and SimCLR+QARe is set to 0.05.

Algorithm 2 Pseudocode for set-based InfoNCE with cosine similarity and Quadratic Assignment Regularization (QARe).

```
# f: encoder network
# alpha: weighting for the pairwise contrastive part (linear assignment)
# beta: weighting for the set contrastive part
# N: batch size
# E: dimensionality of the embeddings

for x in loader: # load a batch with N samples
    # two randomly augmented views of x
    y_a, y_b = augment(x)

    # compute embeddings
    z_a = f(y_a) # NxE
    z_b = f(y_b) # NxE

    # compute inter-set and intra-set similarities
    S_AB = similarity(z_a, z_b) # NxN
    S_A  = similarity(z_a, z_a) # NxN
    S_B  = similarity(z_b, z_b) # NxN

    # compute pairwise contrastive InfoNCE loss
    pairwise_term = infonce(S_AB)

    # shift to non-negative & compute eigenvalues
    eigs_a = eigenvalues(1 + S_A) #N
    eigs_b = eigenvalues(1 + S_B) #N

    # compute QARe from maximum dot product of eigenvalues
    eigs_a_sorted = sort(eigs_a, descending = True) #N
    eigs_b_sorted = sort(eigs_b, descending = True) #N
    qare = eigs_a_sorted.T@eigs_b_sorted

    # combine pairwise contrastive loss with QARe
    loss = alpha*pairwise_term + beta*qare/(N^2)

    # optimization step
    loss.backward()
    optimizer.step()
```

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607 (2020)

2. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. Advances in neural information processing systems **33**, 22243–22255 (2020)
3. Chen, T., Luo, C., Li, L.: Intriguing properties of contrastive losses. In: Advances in Neural Information Processing Systems. vol. 34, pp. 11834–11845 (2021)
4. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
6. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
7. Martins, A.F.T., Astudillo, R.F.: From softmax to sparsemax: A sparse model of attention and multi-label classification. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML) (2016)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
9. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)