Supplementary material

Similarity Contrastive Estimation for Self-Supervised Soft Contrastive Learning

A. Pseudo-Code of SCE

```python
# dataloader: loader of batches of size bsz
# epochs: number of epochs
# T1: weak distribution of data augmentations
# T2: strong distribution of data augmentations
# f1, g1: online encoder and projector
# f2, g2: momentum encoder and projector
# queue: memory buffer
# tau: online temperature
# tau_m: momentum temperature
# lambda_: coefficient between contrastive and relational aspects

for i in range(epochs):
    x1, x2 = T1(x), T2(x)
    z1, z2 = g1(f1(x1)), g2(f2(x2))
    stop_grad(z2)
    sim2_pos = zeros(bsz)
    sim2_neg = einsum("nc,kc->nk", z2, queue)
    sim2 = cat([sim2_pos, sim2_neg]) / tau_m
    s2 = softmax(sim2)
    w2 = lambda_ * one_hot(sim2_pos, bsz + 1) + (1 - lambda_)*s2
    sim1_pos = einsum("nc,nk", z1, z2)
    sim1_neg = einsum("nc,nk", z1, queue)
    sim1 = cat([sim1_pos, sim1_neg]) / tau
    p1 = softmax(sim1)
    loss = cross_entropy(p1, w2)
    loss.backward()
    update(f1.params)
    update(g1.params)
    momentum_update(f2.params, f1.params)
    momentum_update(g2.params, g1.params)
    fifo_update(queue, z2)
```

Algorithm 1: Pseudo-Code of SCE in a pytorch style

B. Proof Proposition 1. in Sec. 3.2

Proposition. $L_{SCE}$ defined as

$$L_{SCE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} w_{ik}^2 \log (p_{ik})$$

can be written as:

$$L_{SCE} = \lambda \cdot L_{InfoNCE} + \mu \cdot L_{ReSSL} + \eta \cdot L_{cNeil},$$

with $\mu = \eta = 1 - \lambda$ and

$$L_{cNeil} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} \left( \frac{\sum_{j=1}^{N} \mathbb{I}_{i \neq j} \cdot \exp(z_{i1}^2 \cdot z_{j2}^2 / \tau)}{\sum_{j=1}^{N} \exp(z_{i1}^2 \cdot z_{j2}^2 / \tau)} \right).$$

Proof. Recall that:

$$p_{ik}^{1} = \frac{\exp(z_{i1}^1 \cdot z_{k2}^2 / \tau)}{\sum_{j=1}^{N} \exp(z_{i1}^1 \cdot z_{j2}^2 / \tau)},$$

$$s_{ik}^{2} = \frac{\prod_{j \neq k} \exp(z_{i1}^2 \cdot z_{j2}^2 / \tau_{m})}{\sum_{j=1}^{N} \prod_{j \neq k} \exp(z_{i1}^2 \cdot z_{j2}^2 / \tau_{m})},$$

$$w_{ik}^{2} = \lambda \cdot \mathbb{I}_{i=k} + (1 - \lambda) \cdot s_{ik}^{2}.$$

We decompose the second loss over $k$ in the definition of $L_{SCE}$ to make the proof:

$$L_{SCE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} w_{ik}^2 \log (p_{ik})$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \left[ w_{i1}^{2} \log (p_{i1}) + \sum_{k=1}^{N} w_{ik}^{2} \log (p_{ik}) \right]$$

$$= -\frac{1}{N} \sum_{i=1}^{N} w_{i1}^{2} \log (p_{i1}) - \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} w_{ik}^{2} \log (p_{ik}).$$

First we rewrite (1) to retrieve the $L_{InfoNCE}$ loss.

$$L_{infoNCE} = -\frac{1}{N} \sum_{i=1}^{N} w_{i1}^{2} \log (p_{i1})$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \lambda \cdot \log (p_{i1})$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \frac{\exp(z_{i1}^1 \cdot z_{i2}^2 / \tau)}{\sum_{j=1}^{N} \exp(z_{j1}^1 \cdot z_{j2}^2 / \tau)}$$

$$= \lambda \cdot L_{infoNCE}. $$

Now we rewrite (2) to retrieve the $L_{ReSSL}$ and $L_{cNeil}$ losses.

$$L_{ReSSL} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} w_{ik}^2 \log (p_{ik})$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} (1 - \lambda) \cdot s_{ik}^2 \cdot \log (p_{ik})$$

$$= - (1 - \lambda) \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} s_{ik}^2 \cdot \log \left( \frac{\exp(z_{i1}^1 \cdot z_{j2}^2 / \tau)}{\sum_{j=1}^{N} \exp(z_{j1}^1 \cdot z_{j2}^2 / \tau)} \right).$$
\[
(1) \quad - (1 - \lambda) \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} \left[ s^2_{ik} \left( \log \left( \frac{\exp(s^2_{ik})}{\sum_{k=1}^{N} \exp(s^2_{ik})} \right) \right) \right] \\
(2) \quad \lambda \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} s^2_{ik} \cdot \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) - \log \left( \frac{\sum_{j=1}^{N} \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) \\
(1 - \lambda) \cdot \frac{1}{N} \sum_{i=1}^{N} \left( \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) \right) = \frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) \\
\]

Because \( s^2_{ii} = 0 \) and \( s^2_k \) is a probability distribution, we have:

\[
\sum_{k=1}^{N} s^2_{ik} \cdot \log \left( \frac{\exp(s^2_{ik})}{\sum_{k=1}^{N} \exp(s^2_{ik})} \right) = \sum_{k=1}^{N} s^2_{ik} \cdot \log \left( \frac{\exp(s^2_{ik})}{\sum_{k=1}^{N} \exp(s^2_{ik})} \right), \\
\sum_{k=1}^{N} s^2_{ik} \cdot \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) = \frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right). \\
\]

Then:

\[
(2) = - (1 - \lambda) \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} \left[ s^2_{ik} \cdot \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) \right] - \lambda \cdot \frac{1}{N} \sum_{i=1}^{N} \left( \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) \right) \\
(1 - \lambda) \cdot \frac{1}{N} \sum_{i=1}^{N} \left( \log \left( \frac{\sum_{j=1}^{N} 1_{ij} \cdot \exp(s^2_{ij})}{\sum_{j=1}^{N} \exp(s^2_{ij})} \right) \right) = (1 - \lambda) \cdot L_{ReSSL} + (1 - \lambda) \cdot L_{CSSL}. \\
\]

Table 1: The 100 classes selected from ImageNet to construct ImageNet100.

D. Data augmentations details for evaluation protocol

The data augmentations used for the evaluation protocol are:

- **training set for large datasets**: random crop to the resolution 224 × 224 and a random horizontal flip with a probability of 0.5.
- **training set for small and medium datasets**: random crop to the dataset resolution with a padding of 4 for small datasets and a random horizontal flip with a probability of 0.5.
- **validation set for large datasets**: resize to the resolution 256 × 256 and center crop to the resolution 224 × 224.
- **validation set for small and medium datasets**: resize to the dataset resolution.

E. Implementation details for pretraining small and medium datasets

Implementation details for small and medium datasets. We use the ResNet-18 encoder and pretrain for
200 epochs. Because the images are smaller, and ResNet is suitable for larger images, typically $224 \times 224$, we follow guidance from SimCLR and replace the first $7 \times 7$ Conv of stride 2 with a $3 \times 3$ Conv of stride 1. We also remove the first pooling layer. The strong data augmentation distribution applied is: random resized crop, color distortion with a strength of 0.5, gray scale with a probability of 0.2, gaussian blur with probability of 0.5, and horizontal flip with probability of 0.5. The weak data augmentation distribution is composed of a random resized crop and a random horizontal flip with the same parameters as the strong data augmentation distribution.

We use 2 GPUs for a total batch size of 256. The memory buffer size is set to 4,096 for small datasets and 16,384 for medium datasets. The projector is a 2 fully connected layer network with a hidden dimension of 512 and an output dimension of 256. A batch normalization is applied after the hidden layer. The SGD optimizer is used during training with a momentum of 0.9 and a weight decay of $5 \times 10^{-4}$. A linear warmup is applied during 5 epochs to reach the initial learning rate of 0.06. The learning rate is scaled using the linear scaling rule: $\text{lr} = \text{initial\_learning\_rate} \times \frac{\text{batch\_size}}{256}$ and then follows the cosine decay scheduler without restart. The momentum value to update the momentum network is 0.99 for small datasets and 0.996 for medium datasets.

F. Temperature influence on small and medium datasets

We made a temperature search on CIFAR10, CIFAR100, STL10 and Tiny-ImageNet by varying $\tau$ in $\{0.1, 0.2\}$ and $\tau_m$ in $\{0.03, ..., 0.10\}$. The results are in Tab. 2. As for ImageNet100, we need a sharper distribution on the output of the momentum encoder. Unlike ResSSL [2], SCE do not collapse when $\tau_m \to \tau$ thanks to the contrastive aspect. For our baselines comparison in Sec. 4.2, we use the best temperatures found for each dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\tau$</th>
<th>$\tau_m = 0.03$</th>
<th>$\tau_m = 0.04$</th>
<th>$\tau_m = 0.05$</th>
<th>$\tau_m = 0.06$</th>
<th>$\tau_m = 0.07$</th>
<th>$\tau_m = 0.08$</th>
<th>$\tau_m = 0.09$</th>
<th>$\tau_m = 0.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>0.1</td>
<td>89.93</td>
<td>90.03</td>
<td>90.06</td>
<td>90.20</td>
<td>90.16</td>
<td>90.06</td>
<td>89.67</td>
<td>88.97</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.2</td>
<td>89.98</td>
<td>90.12</td>
<td>90.12</td>
<td>90.05</td>
<td>90.13</td>
<td>90.09</td>
<td>90.22</td>
<td>90.34</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>0.1</td>
<td>64.49</td>
<td>64.90</td>
<td>65.19</td>
<td>65.33</td>
<td>65.27</td>
<td>65.45</td>
<td>64.89</td>
<td>63.87</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>0.2</td>
<td>63.71</td>
<td>63.74</td>
<td>63.89</td>
<td>64.05</td>
<td>64.24</td>
<td>64.23</td>
<td>64.10</td>
<td>64.30</td>
</tr>
<tr>
<td>STL10</td>
<td>0.1</td>
<td>89.34</td>
<td>89.94</td>
<td>89.87</td>
<td>89.84</td>
<td>89.72</td>
<td>89.52</td>
<td>88.99</td>
<td>88.41</td>
</tr>
<tr>
<td>STL10</td>
<td>0.2</td>
<td>88.4</td>
<td>88.23</td>
<td>88.4</td>
<td>88.35</td>
<td>87.54</td>
<td>88.32</td>
<td>88.80</td>
<td>88.59</td>
</tr>
<tr>
<td>Tiny-IN</td>
<td>0.1</td>
<td>50.23</td>
<td>51.12</td>
<td>51.41</td>
<td>51.66</td>
<td>51.90</td>
<td>51.58</td>
<td>51.37</td>
<td>50.46</td>
</tr>
<tr>
<td>Tiny-IN</td>
<td>0.2</td>
<td>48.56</td>
<td>48.85</td>
<td>48.35</td>
<td>48.98</td>
<td>49.06</td>
<td>49.15</td>
<td>49.66</td>
<td>49.64</td>
</tr>
</tbody>
</table>

Table 2: Effect of varying the temperature parameters $\tau_m$ and $\tau$ on the Top-1 accuracy on small and medium datasets.

References
