# A More on generalized contrastive loss

## A.1 Experimental setup

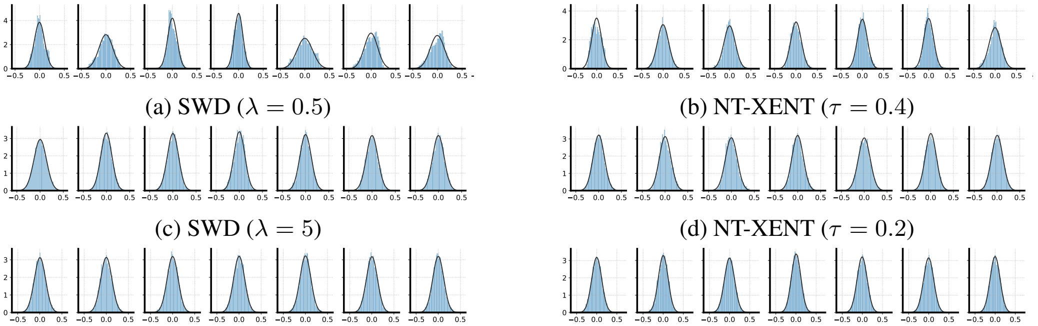
We follow [13, 14] for the use of augmentations and architectures. By default, we use ResNet-50 [39] and a 2-layer projection head [13, 14] after the ResNet's average pooling layer. We set the output ($\boldsymbol{z}$) dimensionality to 64 for CIFAR10 and 128 for ImageNet, since increasing them has little effect on the performance. We use a square root learning rate scaling with batch size with a LARS optimizer [21], i.e., $\mathrm{LearningRate} = 0.075 \times \sqrt{\mathrm{BatchSize}}$ for ImageNet and $\mathrm{LearningRate} = 0.2 \times \sqrt{\mathrm{BatchSize}}$ for CIFAR-10. The batch size and training epoch will be specified for each experiment. We use the linear evaluation protocol, i.e. the accuracy of a trained linear classifier on the learned features is used as a proxy for representation quality.

When comparing the standard contrastive loss (i.e. NT-Xent in Eq. 1) and other instantiations of the generalized contrastive loss (in Table 1), we optimize the hyper-parameters for different losses (for NT-Xent loss, we set $\tau = 0.2$; for decoupled NT-Xent loss, we set $\tau = 1.0, \lambda = 0.1$; for SWD based losses, we set $\lambda = 5$ ; and since we use mean squared error instead of $\ell_2$ distance in alignment loss for losses in Table 1, we find it helpful to scale the loss by 1000 when the hidden vector $\boldsymbol{z}$ is normalized). A batch size of 128 is used for CIFAR-10, and 1024 is used for ImageNet.

## A.2 Temperature $\tau$ is (within a range) inversely correlated to weighting $\lambda$ of distribution loss

**Both temperature $\tau$ and weighting $\lambda$ control how well the representations fit the prior.** To see how well the learned distribution matches the prior distribution (e.g. Gaussian), we randomly project the (high-dimensional) representation vectors into 1-D space and plot the histogram distribution. For prior distribution of Gaussian or uniform in hypersphere, these random projections in 1-D space should be Gaussian like.

Figure A.1 shows random orthogonal projection of representation from CIFAR-10 test set. We see that both weighting ($\lambda$ in Eq. 2) and the temperature scaling ($\tau$ in Eq. 1) have the effect of controlling distribution matching term, but they have an inverse correlation. In other words, using a higher temperature has similar effect as setting a larger weighting of distribution matching term.



(a) SWD ($\lambda = 0.5$)

(b) NT-XENT ($\tau = 0.4$)

(c) SWD ($\lambda = 5$)

(d) NT-XENT ($\tau = 0.2$)

(e) SWD ($\lambda = 50$)

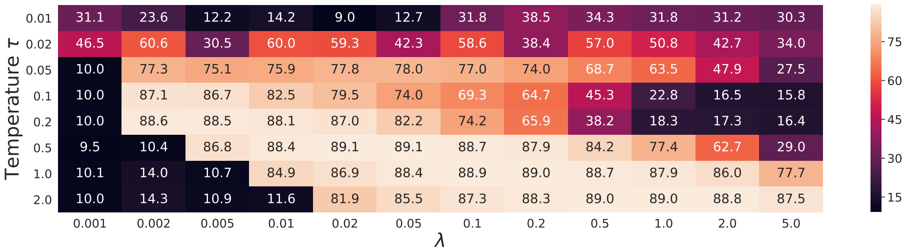
(f) NT-XENT ($\tau = 0.1$)

Figure A.1: Distribution of random orthogonal projection of output vectors on CIFAR-10 test set (each small plot has its own random projection direction). For SWD (uniform hypersphere) loss, distribution becomes more Gaussian as $\lambda$ increases. For NT-Xent loss, the distribution becomes more Gaussian as $\tau$ decreases.

**Decoupled NT-Xent loss.** It is worth noting that temperature $\tau$ in the rewritten NT-Xent loss (Eq. 3) appears in two places, one as the scaling of the distribution loss term, and the other as the width of Gaussian kernel. They do not necessarily need to be the same, so we could decouple them as follows.

$$\mathcal{L}^{\text{Decoupled NT-Xent}} = -\frac{1}{n}\sum_{i,j}\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j) + \lambda\frac{1}{n}\sum_{i}\log\sum_{k=1}^{2n}\mathbb{1}_{[k\neq i]}\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau) \tag{4}$$

The decoupling allows us to study the effects of them separately. So we tune $\tau$ and $\lambda$ separately for the decoupled NT-xent loss. Figure A.2 shows the linear evaluation of ResNet-18 trained in 200 epochs. We see that the temperature $\tau$ and the weighting $\lambda$ are inversely correlated for most range. In practice one could simply fix one and tune the other.



| Temperature $\tau$ \ $\lambda$ | 0.001 | 0.002 | 0.005 | 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.5 | 1.0 | 2.0 | 5.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.01 | 31.1 | 23.6 | 12.2 | 14.2 | 9.0 | 12.7 | 31.8 | 38.5 | 34.3 | 31.8 | 31.2 | 30.3 |
| 0.02 | 46.5 | 60.6 | 30.5 | 60.0 | 59.3 | 42.3 | 58.6 | 38.4 | 57.0 | 50.8 | 42.7 | 34.0 |
| 0.05 | 10.0 | 77.3 | 75.1 | 75.9 | 77.8 | 78.0 | 77.0 | 74.0 | 68.7 | 63.5 | 47.9 | 27.5 |
| 0.1 | 10.0 | 87.1 | 86.7 | 82.5 | 79.5 | 74.0 | 69.3 | 64.7 | 45.3 | 22.8 | 16.5 | 15.8 |
| 0.2 | 10.0 | 88.6 | 88.5 | 88.1 | 87.0 | 82.2 | 74.2 | 65.9 | 38.2 | 18.3 | 17.3 | 16.4 |
| 0.5 | 9.5 | 10.4 | 86.8 | 88.4 | 89.1 | 89.1 | 88.7 | 87.9 | 84.2 | 77.4 | 62.7 | 29.0 |
| 1.0 | 10.1 | 14.0 | 10.7 | 84.9 | 86.9 | 88.4 | 88.9 | 89.0 | 88.7 | 87.9 | 86.0 | 77.7 |
| 2.0 | 10.0 | 14.3 | 10.9 | 11.6 | 81.9 | 85.5 | 87.3 | 88.3 | 89.0 | 89.0 | 88.8 | 87.5 |

Figure A.2: Linear evaluation of ResNet-18 trained on CIFAR-10 (200 epochs) using decoupled NT-Xent loss (Eq. 4). The temperature $\tau$ and the weighting $\lambda$ are mostly inverse correlated.

### A.3 Linear evaluation of generalized contrastive losses on CIFAR-10 and ImageNet

Table A.1, A.2 and A.3 show linear evaluation performance of ResNet-50 trained with different losses (numerical results of Figure 1). Similar to [13, 14], a square root learning rate is used. In addition, results of different batch sizes are also compared, and we find the differences are small with reasonable sizes (e.g. 128 for CIFAR-10 and 1024 for ImageNet).

Table A.1: Linear evaluation accuracy (top-1) of ResNet-50 trained with different losses on CIFAR-10.

|  | Epoch | 100 | 200 | 400 | 800 |
|---|---|---|---|---|---|
| Loss | Batch size |  |  |  |  |
| NT-Xent | 128 | 87.4 | 91.0 | 93.0 | 93.9 |
|  | 256 | 88.0 | 91.3 | 93.0 | 93.6 |
|  | 512 | 87.9 | 91.3 | 92.9 | 93.7 |
|  | 1024 | 88.2 | 91.2 | 92.7 | 93.3 |
| Decoupled NT-Xent | 128 | 87.8 | 91.0 | 93.0 | 94.0 |
|  | 256 | 87.7 | 91.1 | 92.8 | 93.6 |
|  | 512 | 87.5 | 91.3 | 92.7 | 93.6 |
|  | 1024 | 87.5 | 91.0 | 92.6 | 93.7 |
| SWD (normal) | 128 | 86.3 | 90.5 | 92.8 | 93.8 |
|  | 256 | 86.2 | 90.8 | 93.1 | 94.1 |
|  | 512 | 85.0 | 90.7 | 92.9 | 94.1 |
|  | 1024 | 83.3 | 89.9 | 93.0 | 93.9 |
| SWD (uniform hypercube) | 128 | 85.1 | 90.1 | 92.6 | 93.4 |
|  | 256 | 84.6 | 89.9 | 92.9 | 93.8 |
|  | 512 | 83.1 | 89.8 | 92.8 | 93.8 |
|  | 1024 | 81.3 | 88.3 | 92.2 | 93.6 |
| SWD (uniform hypersphere) | 128 | 87.0 | 90.9 | 92.9 | 93.8 |
|  | 256 | 87.1 | 90.9 | 92.5 | 93.7 |
|  | 512 | 86.6 | 90.8 | 92.9 | 93.4 |
|  | 1024 | 86.0 | 90.3 | 92.5 | 93.2 |

## B More on feature suppression

### B.1 Extra results on CIFAR-10 and ImageNet with random bits added

Figure B.1 shows linear evaluation on CIFAR-10 with different random bits added trained with a wider range of batch sizes. It is worth noting that the bits (in the x-axis) are calculated based on the total size of uniform integer distribution. However, this is an overestimation of actual bits as due to collision in generated integers. We observe that the linear evaluation accuracy decreases

Table A.2: Linear evaluation accuracy (top-1) of ResNet-50 trained with different losses on ImageNet (with 2-layer projection head).

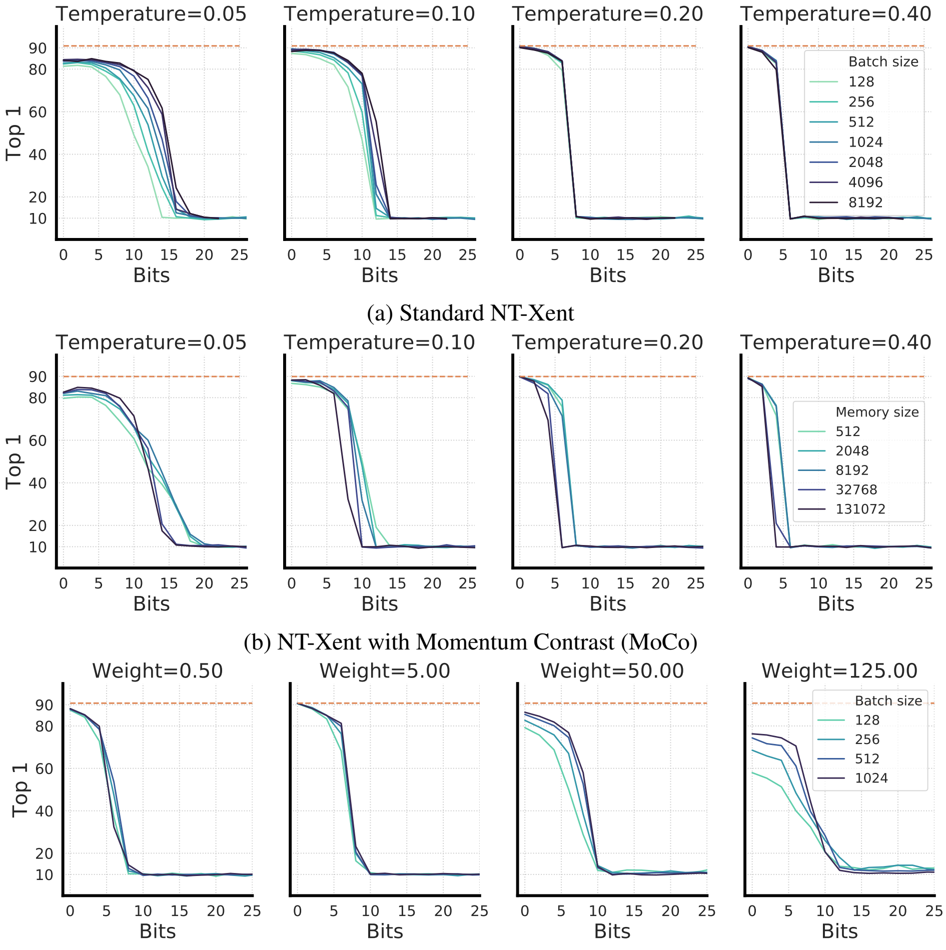
| Loss | Epoch<br>Batch size | 100 | 200 | 400 | 800 |
|---|---|---|---|---|---|
| NT-Xent | 512 | 65.4 | 67.3 | 68.7 | 69.3 |
| | 1024 | 65.6 | 67.6 | 68.8 | 69.8 |
| | 2048 | 65.3 | 67.6 | 69.0 | 70.1 |
| Decoupled NT-Xent | 512 | 65.8 | 67.6 | 68.9 | 69.5 |
| | 1024 | 66.0 | 67.9 | 69.0 | 70.1 |
| | 2048 | 65.8 | 67.9 | 69.3 | 70.2 |
| SWD (normal) | 512 | 64.9 | 66.8 | 68.0 | 69.0 |
| | 1024 | 65.0 | 67.1 | 68.2 | 69.3 |
| | 2048 | 65.0 | 66.9 | 68.4 | 69.7 |
| SWD (uniform hypercube) | 512 | 64.3 | 66.4 | 67.8 | 68.7 |
| | 1024 | 64.2 | 66.5 | 67.9 | 68.9 |
| | 2048 | 63.9 | 66.6 | 67.9 | 69.0 |
| SWD (uniform hypersphere) | 512 | 65.6 | 67.7 | 69.0 | 70.0 |
| | 1024 | 65.8 | 67.9 | 69.0 | 69.6 |
| | 2048 | 65.6 | 67.8 | 69.2 | 69.8 |

Table A.3: Linear evaluation accuracy (top-1) of ResNet-50 trained with different losses on ImageNet (with 3-layer projection head).

| Loss | Epoch<br>Batch size | 100 | 200 | 400 | 800 |
|---|---|---|---|---|---|
| NT-Xent | 512 | 66.6 | 68.4 | 70.0 | 71.0 |
| | 1024 | 66.8 | 68.9 | 70.1 | 70.9 |
| | 2048 | 66.8 | 69.1 | 70.4 | 71.3 |
| Decoupled NT-Xent | 512 | 66.8 | 68.4 | 69.6 | 70.6 |
| | 1024 | 66.6 | 68.9 | 69.9 | 70.8 |
| | 2048 | 66.6 | 69.0 | 70.1 | 70.8 |
| SWD (normal) | 512 | 66.5 | 68.4 | 69.8 | 70.8 |
| | 1024 | 66.6 | 68.8 | 70.1 | 71.1 |
| | 2048 | 66.7 | 69.1 | 70.2 | 71.1 |
| SWD (uniform hypercube) | 512 | 66.1 | 68.3 | 69.7 | 70.7 |
| | 1024 | 66.3 | 68.5 | 70.0 | 71.3 |
| | 2048 | 65.8 | 68.2 | 70.1 | 71.1 |
| SWD (uniform hypersphere) | 512 | 66.5 | 68.3 | 69.5 | 70.5 |
| | 1024 | 66.6 | 68.6 | 69.8 | 70.8 |
| | 2048 | 66.5 | 68.7 | 70.2 | 70.9 |

quickly with a few bits of the extra channel competing feature added. And this detrimental effect on the representation quality cannot be avoided by different contrastive loss functions, batch sizes, or memory mechanism in momentum contrast [10]. Although a smaller temperature ($\tau$) or larger weighting ($\lambda$) slightly mitigate the degeneration effect, its baseline performance when no extra bits are added is also worse. With less than 15 bits of competing features added, the representation quality degenerates to the level where RGB channels are completely ignored.

Similar results are shown for ImageNet as shown in Figure B.2.

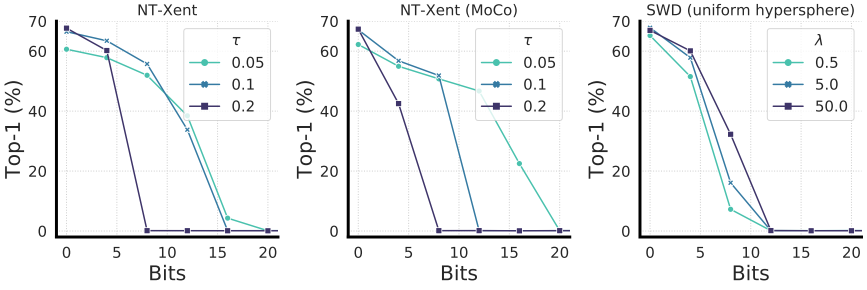


(a) Standard NT-Xent

(b) NT-Xent with Momentum Contrast (MoCo)

(c) SWD (uniform hypersphere)

Figure B.1: Linear evaluation accuracy on CIFAR-10 of ResNet-18 (400 epochs) when different random bits are added. Different contrastive losses and batch sizes are compared.



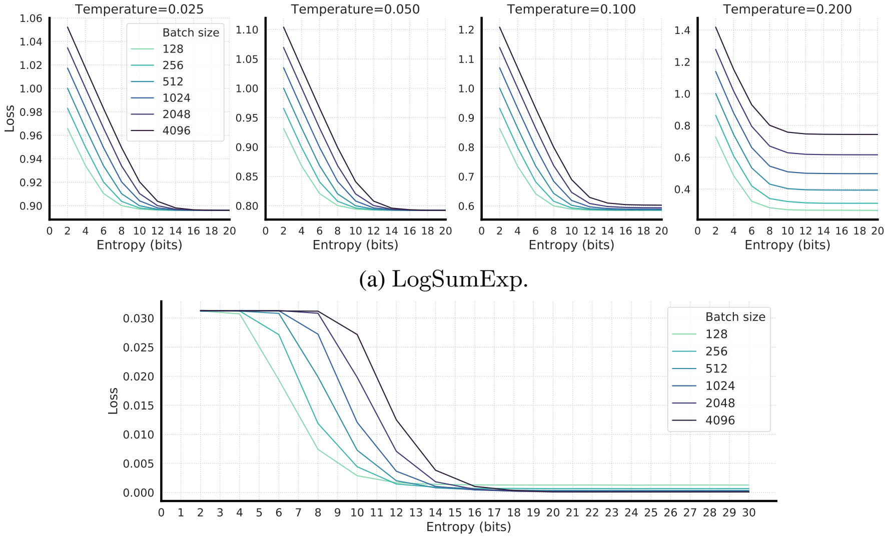
(a) ImageNet.

Figure B.2: Linear evaluation of learned features when a few bits of competing features added on ImageNet. A few bits added completely disable contrastive learning (across various batch size or losses).

### B.2 Distribution matching loss, $\mathrm{LogSumExp}$ or SWD, saturates with a few bits of entropy

Here we study the saturation of distribution matching loss (based on $\mathrm{LogSumExp}$ or SWD), without presence of the alignment term. To do so, we create square images with $k$ binary channels (instead of RGB channels), and all pixels at different locations of a $32 \times 32$ image share the same value, this allows us to use the same architecture as one for CIFAR-10 (i.e. ResNet-18 and 2-layer projection head with output dimensionality of 64). We note that this experiment can also be conducted on images of $1 \times 1$ size with other architecture. It is not difficult to see the entropy of this dataset is $k$ bits. A mini-batch of data points (without augmentations) are first encoded via the network, and then the distribution matching loss is defined on the network's outputs. The network is trained for 400 epochs, and longer training epochs makes little difference.

Figure B.3 shows that distribution loss saturates quickly with a few bits of entropy in the dataset (same or less bits in representations), and both temperature and batch sizes have effects on the saturation behavior. It also shows that linear increase of bits in representation requires exponentially increase of batch size, which is not sustainable as the required batch size can quickly go beyond the size of the dataset (e.g., 30 bits would require more than 1 billion batch size, which is larger than most of the existing datasets). This is one of the main reasons why data augmentation is critical for contrastive learning - that the network can learn a few bits that give rise to useful representations.



(a) $\mathrm{LogSumExp}$.

(b) SWD (uniform hypersphere).

Figure B.3: Distribution matching loss saturates quickly with a few bits of entropy. The saturation varies slightly across batch sizes.