A Proofs

A.1 Lemma 1

Proof. Throughout the following proofs, we will use the fact that classes are assumed to be balanced in order to consider $\mathcal{Z}_k$, for any class k, as a constant $|\mathcal{Z}_k| = \frac{n}{K}$. We will also use the feature normalization assumption to connect cosine and Euclidean distances. On the unit-hypersphere, we will use that: $D_{i,j}^{\cos} = 1 - \frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2}{2}$.

Tightness terms: Let us start by linking center loss to contrastive loss. For any specific class k, let $\boldsymbol{c}_k = \frac{1}{|\mathcal{Z}_k|} \sum\limits_{z \in \mathcal{Z}_k} z$ denotes the hard mean. We can write:

$$
\begin{aligned}
\sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \|\boldsymbol{z}_i - \boldsymbol{c}_k\|^2 &= \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} [\|\boldsymbol{z}_i\|^2 - 2\boldsymbol{z}_i^\top \boldsymbol{c}_k] + |\mathcal{Z}_k| \, \|\boldsymbol{c}_k\|^2 \\
&= \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \|\boldsymbol{z}_i\|^2 - 2\frac{1}{|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} \boldsymbol{z}_i^\top \boldsymbol{z}_j + \frac{1}{|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} \boldsymbol{z}_i^\top \boldsymbol{z}_j \\
&= \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \|\boldsymbol{z}_i\|^2 - \frac{1}{|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} \boldsymbol{z}_i^\top \boldsymbol{z}_j \\
&= \frac{1}{2}[\sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \|\boldsymbol{z}_i\|^2 + \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} \|\boldsymbol{z}_j\|^2] - \frac{1}{|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} \boldsymbol{z}_i^\top \boldsymbol{z}_j \\
&= \frac{1}{2|\mathcal{Z}_k|}[\sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} \|\boldsymbol{z}_i\|^2 + \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} \|\boldsymbol{z}_j\|^2] \\
&\quad - \frac{1}{2|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} 2\boldsymbol{z}_i^\top \boldsymbol{z}_j \\
&= \frac{1}{2|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{Z}_k} \|\boldsymbol{z}_i\|^2 - 2\boldsymbol{z}_i^\top \boldsymbol{z}_j + \|\boldsymbol{z}_j\|^2 \\
&= \frac{1}{2|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{Z}_k} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 \\
&\overset{\text{C}}{=} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{Z}_k} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2
\end{aligned}
$$

Summing over all classes k, we get the desired equivalence. Note that, in the context of K-means clustering, where the setting is different[‡], a technically similar

[‡]In clustering, the optimization is performed over assignment variables, as opposed to DML, where assignments are already known and optimization is carried out over the embedding.

result could be established [32], linking K-means to pairwise graph clusteirng objectives.

Now we link contrastive loss to SNCA loss. For any class k, we can write:

$$
\begin{aligned}
-\sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \log \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k \setminus \{i\}} e^{\frac{D^{\cos}_{i,j}}{\sigma}} &\overset{\text{c}}{=} -\sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \log \left(\frac{1}{|\mathcal{Z}_k| - 1} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k \setminus \{i\}} e^{\frac{D^{\cos}_{i,j}}{\sigma}} \right) \\
&\leq -\sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k \setminus \{i\}} \frac{D^{\cos}_{i,j}}{(|\mathcal{Z}_k| - 1)\sigma} \\
&\overset{\text{c}}{=} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k \setminus \{i\}} \frac{\|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2}{2\sigma(|\mathcal{Z}_k| - 1)} \\
&\overset{\text{c}}{=} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k \setminus \{i\}} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2
\end{aligned}
$$

where we used the convexity of $x \to -\log(x)$ and Jenson's inequality. The proof can be finished by summing over all classes k.

Finally, we link MS loss [40] to contrastive loss:

$$
\begin{aligned}
\sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \frac{1}{\alpha} \log \left(1 + \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k \setminus \{i\}} e^{-\alpha(D^{\cos}_{i,j} - 1)} \right) &= \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \frac{1}{\alpha} \log \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} e^{-\alpha(D^{\cos}_{i,j} - 1)} \\
&\overset{\text{c}}{=} \sum_{\boldsymbol{z}_i \in \mathcal{Z}_k} \frac{1}{\alpha} \log \left(\frac{1}{|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_j \in \mathcal{Z}_k} e^{-\alpha(D^{\cos}_{i,j} - 1)} \right) \\
&\geq \frac{1}{|\mathcal{Z}_k|} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{Z}_k} -(D^{\cos}_{i,j} - 1) \\
&\overset{\text{c}}{=} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{Z}_k} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 ,
\end{aligned}
$$

where we used the concavity of $x \to \log(x)$ and Jenson's inequality.

Contrastive terms: In this part, we first show that the contrastive terms C_{SNCA} and C_{MS} represent upper bounds on $C = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j \neq y_i} D_{ij}^2$:

$$\begin{aligned} C_{MS} = \frac{1}{\beta n}\sum_{i=1}^{n}\log\left(1+\sum_{j:y_j\neq y_i} e^{\beta(D_{ij}^{\cos}-1)}\right) &\geq \frac{1}{\beta n}\sum_{i=1}^{n}\log\left(\sum_{j:y_j\neq y_i} e^{\beta(D_{ij}^{\cos}-1)}\right) \\ &\overset{c}{\geq} \frac{1}{\beta n}\sum_{i=1}^{n}\sum_{j:y_j\neq y_i}\beta(D_{ij}^{\cos}-1) \\ &\overset{c}{=} -\frac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j\neq y_i} D_{ij}^2 \\ &= C \end{aligned}$$

where, again, we used Jenson's inequality in the second line above. The link between SNCA and contrastive loss can be established quite similarly:

$$C_{SNCA} = \frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{j\neq i} e^{\frac{D_{ij}^{\cos}}{\sigma}}\right) = \frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{j\neq i: y_i=y_j} e^{\frac{D_{ij}^{\cos}}{\sigma}} + \sum_{j:y_j\neq y_i} e^{\frac{D_{ij}^{\cos}}{\sigma}}\right) \tag{16}$$

$$\geq \frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{j:y_j\neq y_i} e^{\frac{D_{ij}^{\cos}}{\sigma}}\right) \tag{17}$$

$$\overset{c}{\geq} \frac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j\neq y_i}\frac{D_{ij}^{\cos}}{\sigma} \tag{18}$$

$$\overset{c}{=} -\frac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j\neq y_i} D_{ij}^2 \tag{19}$$

$$= C \tag{20}$$

Now, similarly to the reasoning carried out in Section 3.1, we can write:

$$C = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j\neq y_i} D_{ij}^2 = \underbrace{-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}^2}_{\text{contrast } \propto\ \mathcal{H}(\hat{Z})} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{j:y_j=y_i} D_{ij}^2}_{\text{tightness subterm } \propto\ \mathcal{H}(\hat{Z}|Y)}$$

Where the redundant tightness term is very similar to the tightness term in contrastive loss $T_{contrast}$ treated in details in Section 3.1. As for the truly contrastive part of C, it can also be related to the differential entropy estimator used in [38]:

$$\widehat{\mathcal{H}}(\widehat{Z}) = \frac{d}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}\log D_{ij}^2 \overset{c}{=} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\log D_{ij}^2 \tag{21}$$

In summary, we just proved that the contrastive parts of MS and SNCA losses are upper bounds on the contrastive term C. The latter term is composed of a proxy for the entropy of features $\mathcal{H}(\hat{Z})$, as well as a tightness sub-term. □

A.2 Proposition 1

Proof. First, let us show that $\mathcal{L}_{CE} \geq \mathcal{L}_{PCE}$. Consider the usual softmax parametrization of point i belonging to class k: $p_{ik} = (f_{\boldsymbol{\theta}}(z_i))_k = \frac{\exp \boldsymbol{\theta}_k^\intercal \boldsymbol{z}_i}{\sum_j \exp \boldsymbol{\theta}_j^\intercal \boldsymbol{z}_i}$, where $z = \phi_{\mathcal{W}}(x)$. We can explicitly write the cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{CE} &= -\frac{1}{n}\sum_{i=1}^{n} \log f_{\boldsymbol{\theta}}(z_i) \\ &= \underbrace{-\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\theta}_{y_i}^\intercal \boldsymbol{z}_i + \frac{\lambda}{2}\sum_{k=1}^{K} \boldsymbol{\theta}_k^\intercal \boldsymbol{\theta}_k}_{f_1(\boldsymbol{\theta})} + \underbrace{\frac{1}{n}\sum_{i=1}^{n} \log \sum_{j=1}^{K} e^{\boldsymbol{\theta}_j^\intercal \boldsymbol{z}_i} - \frac{\lambda}{2}\sum_{k=1}^{K} \boldsymbol{\theta}_k^\intercal \boldsymbol{\theta}_k}_{f_2(\boldsymbol{\theta})}. \end{aligned} \tag{22}$$

Where we introduced $\lambda \in \mathbb{R}$. How to specifically set λ will soon become clear. Let us now write the gradients of f_1 and f_2 in Eq. 22 with respect to $\boldsymbol{\theta}_k$:

$$\frac{\partial f_1}{\partial \boldsymbol{\theta}_k} = -\frac{1}{n}\sum_{i:y_i=k} \boldsymbol{z}_i + \lambda \boldsymbol{\theta}_k \tag{23}$$

$$\frac{\partial f_2}{\partial \boldsymbol{\theta}_k} = \frac{1}{n}\sum_{i} \underbrace{\frac{\exp(\boldsymbol{\theta}_k^\intercal \boldsymbol{z}_i)}{\sum_{j=1}^{K} \exp(\boldsymbol{\theta}_j^\intercal \boldsymbol{z}_i)}}_{p_{ik}} \boldsymbol{z}_i - \lambda \boldsymbol{\theta}_k \tag{24}$$

Notice that f_1 is a convex function of $\boldsymbol{\theta}$, regardless of λ. As for f_2, we set λ such that f_2 becomes a convex function of $\boldsymbol{\theta}$. Specifically, by setting:

$$\lambda = \min_{k,l} \sigma_l(A_k) \tag{25}$$

where $A_k = \frac{1}{n}\sum_{i=1}^{n}(p_{ik} - p_{ik}^2)\boldsymbol{z}_i \boldsymbol{z}_i^\intercal$ and $\sigma_l(A)$ represents the l^{th} eigenvalue of A, we make sure that the hessian of f_2 is semi-definite positive. Therefore, we can look for the minima of f_1 and f_2.

Setting gradients in Eq. 23 and Eq. 24 to 0, we obtain that for all $k \in [1, K]$, the optimal $\boldsymbol{\theta}_k$ for f_1 is, up to a multiplicative constant, the hard mean of features from class k: $\boldsymbol{\theta}_k^{f_1 *} = \frac{1}{\lambda n}\sum_{i:y_i=k} \boldsymbol{z}_i \propto \boldsymbol{c}_k$, while the optimal $\boldsymbol{\theta}_k$ for f_2 is, up to a multiplicative constant, the soft mean of features: $\boldsymbol{\theta}_k^{f_2 *} = \frac{1}{\lambda n}\sum_{i=1}^{n} p_{ik}\boldsymbol{z}_i = \boldsymbol{c}_k^s/\lambda$. Therefore, we can write:

$$f_1(\boldsymbol{\theta}) \geq f_1(\boldsymbol{\theta}^{f_1 *}) = -\frac{1}{\lambda n^2}\sum_{i=1}^{n}\sum_{j:y_j=y_i} \boldsymbol{z}_i^\intercal \boldsymbol{z}_j + \frac{\lambda}{2\lambda^2}\sum_{i=1}^{n}\sum_{j:y_j=y_i} \boldsymbol{z}_i^\intercal \boldsymbol{z}_j \tag{26}$$

$$= -\frac{1}{2\lambda n^2}\sum_{i=1}^{n}\sum_{j:y_j=y_i} \boldsymbol{z}_i^\intercal \boldsymbol{z}_j \tag{27}$$

And

$$f_2(\boldsymbol{\theta}) \geq f_2(\boldsymbol{\theta}^{f_2*}) \tag{28}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\log\sum_{k=1}^{K}\exp\left(\frac{1}{\lambda n}\sum_{j=1}^{n}p_{jk}\boldsymbol{z}_i^\top\boldsymbol{z}_j\right) - \frac{1}{2\lambda}\sum_{k=1}^{K}\|\boldsymbol{c}_k^s\|^2 \tag{29}$$

Putting it all together, we can obtain the desired result:

$$\mathcal{L}_{CE} \geq -\frac{1}{2\lambda n^2}\sum_{i=1}^{n}\sum_{j:y_j=y_i}\boldsymbol{z}_i^\top\boldsymbol{z}_j + \frac{1}{n}\sum_{i=1}^{n}\log\sum_{k=1}^{K}e^{\frac{1}{\lambda n}\sum_j p_{jk}\boldsymbol{z}_i^\top\boldsymbol{z}_j} - \frac{1}{2\lambda}\sum_{k=1}^{K}\|\boldsymbol{c}_k^s\|^2 \tag{30}$$

$$= \mathcal{L}_{PCE} \tag{31}$$

where $\boldsymbol{c}_k^s = \frac{1}{n}\sum_{i=1}^{n}p_{ik}\boldsymbol{z}_i$ represents the soft mean of class k.

Let us now justify that minimizing cross-entropy can be seen as an approximate bound optimization on $\mathcal{L}_{PCE}$. At every iteration t of the training, cross-entropy represents an upper bound on Pairwise Cross-entropy.

$$\mathcal{L}_{CE}(\mathcal{W}(t), \boldsymbol{\theta}(t)) \geq \mathcal{L}_{PCE}(\mathcal{W}(t), \boldsymbol{\theta}(t)) \tag{32}$$

When optimizing w.r.t θ, the bound almost becomes tight. The approximation comes from the fact that $\boldsymbol{\theta}_k^{f_1*}$ and $\boldsymbol{\theta}_k^{f_2*}$ are quite dissimilar in early training, but become very similar as training progresses and the model's softmax probabilities align with the labels. Therefore, using the notation:

$$\boldsymbol{\theta}(t+1) = \min_{\boldsymbol{\theta}} \mathcal{L}_{CE}(\mathcal{W}(t), \boldsymbol{\theta}) \tag{33}$$

We can write:

$$\mathcal{L}_{CE}(\mathcal{W}(t), \boldsymbol{\theta}(t+1)) \approx \mathcal{L}_{PCE}(\mathcal{W}(t), \boldsymbol{\theta}(t+1)) \tag{34}$$

Then, minimizing $\mathcal{L}_{CE}$ and $\mathcal{L}_{PCE}$ w.r.t $\mathcal{W}$ becomes approximately equivalent. □

A.3 Lemma 2

Proof. Using the discriminative view of MI, we can write:

$$\mathcal{I}(\widehat{Z}; Y) = \mathcal{H}(Y) - \mathcal{H}(Y|\widehat{Z}) \tag{35}$$

The entropy of labels $\mathcal{H}(Y)$ is a constant and, therefore, can be ignored. From this view of MI, maximization of $\mathcal{I}(\widehat{Z}; Y)$ can only be achieved through a minimization of $\mathcal{H}(Y|\widehat{Z})$, which depends on our embeddings $\widehat{Z} = \phi_{\mathcal{W}}(X)$. We can relate this term to our cross-entropy loss using the following relation:

$$\mathcal{H}(Y; \widehat{Y}|\widehat{Z}) = \mathcal{H}(Y|\widehat{Z}) + \mathcal{D}_{KL}(Y\|\widehat{Y}|\widehat{Z}) \tag{36}$$

Therefore, while minimizing cross-entropy, we are implicitly both minimizing $\mathcal{H}(Y|\widehat{Z})$ as well as $\mathcal{D}_{KL}(Y\|\widehat{Y}|\widehat{Z})$. In fact, following Eq. 36, optimization could naturally be decoupled in 2 steps, in a *Maximize-Minimize* fashion. One step would consist in fixing the encoder's weights $\mathcal{W}$ and only minimizing Eq. 36 w.r.t to the classifier's weights $\boldsymbol{\theta}$. At this step, $\mathcal{H}(Y|\widehat{Z})$ would be fixed while $\widehat{Y}$ would be adjusted to minimize $\mathcal{D}_{KL}(Y||\widehat{Y}|\widehat{Z})$. Ideally, the KL term would vanish at the end of this step. In the following step, we would minimize Eq. 36 w.r.t to the encoder's weights $\mathcal{W}$, while keeping the classifier fixed. □

B Preliminary results with SPCE

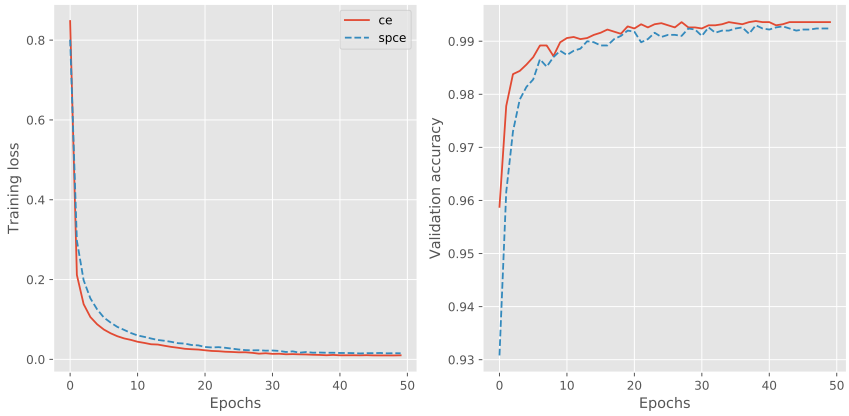

Fig. 1. Evolution of the cross-entropy loss (CE) and the simplified pairwise cross-entropy (SPCE) during training on MNIST, as well as the validation accuracy for both losses.

In Fig. 1, we track the evolution of both loss functions and validation accuracy when training with $\mathcal{L}_{CE}$ and $\mathcal{L}_{SPCE}$ on MNIST dataset. We use a small CNN composed of four convolutional layers. The optimizer used is Adam. Batch size is set to 128, learning rate to $1e^{-4}$ with cosine annealing, weight decay to $1e^{-4}$ and feature dimension to $d = 100$. Fig. 1 supports the theoretical links that were drawn between Cross-Entropy and its simplied pairwise version SPCE. Particularly, this preliminary result demonstrates that SPCE is indeed employable as a loss, and exhibits a very similar behavior to the original cross-entropy. Both losses remain very close to each other throughout the training, and so remain the validation accuracies.

C Analysis of ranking losses for Deep Metric Learning

Some recent works [1, 24, 39] tackle the problem of deep metric learning using a rank-based approach. In other words, given a point in feature space $\boldsymbol{z}_i$, the pairwise losses studied throughout this work try to impose manual margins m, so that the distance between $\boldsymbol{z}_i$ and any negative point $\boldsymbol{z}_j^-$ is at least m. Rank-based losses rather encourage that all points are well ranked, distance-wise, such that $d(\boldsymbol{z}_i, \boldsymbol{z}_j^+) \leq d(\boldsymbol{z}_i, \boldsymbol{z}_j^-)$ for any positive and negative points $\boldsymbol{z}_j^+$ and $\boldsymbol{z}_j^-$. We show that our tightness/contrastive analysis also holds for such ranking losses. In particular, we analyse the loss proposed in [1]. For any given query embedded point $\boldsymbol{z}_i$, let us call D the random variable associated to the distance between z_i and all other points in the embedded space, defined over all possible (discretized) distances $\mathcal{D}$. Furthermore, let us call R the binary random variable that describes the relation to the current query point (R^+ and R^- describe respectively a positive and negative relationship to $\boldsymbol{z}_i$). The loss maximized in [1] reads:

$$\text{FastAP} = \sum_{d \in \mathcal{D}} \frac{P(D < d|R^+)P(R^+)}{P(D < d)} P(D = d|R^+) \tag{37}$$

Taking the logarithm, and using Jensen's inequality, we can lower bound this loss:

$$\begin{aligned} \log(\text{FastAP}) &\geq \sum_{d \in \mathcal{D}} P(D = d, R^+) \log\left(\frac{P(D < d|R^+)}{P(D < d)}\right) \\ &= \underbrace{\mathop{\mathbb{E}}_{d \sim P(.,R^+)} \log P(D < d|R^+)}_{T_{AP}=\text{TIGHTNESS}} - \underbrace{\mathop{\mathbb{E}}_{d \sim P(.,R^+)} \log P(D < d)}_{C_{AP}=\text{CONTRASTIVE}} \end{aligned} \tag{38}$$

To intuitively understand what those two terms are doing, let us imagine we approximate each of the expectations with a single point Monte-Carlo approximation. In other words, we sample a positive point $\boldsymbol{z}_j^+$, take its associated distance to $\boldsymbol{z}_i$, which we call d^+, then we approximate the tightness term as:

$$T_{AP} \approx \log P(D < d^+|R^+) \tag{39}$$

Maximizing T_{AP} has a clear interpretation: it encourages all positive points to lie inside the hypersphere of radius d^+ around query point $\boldsymbol{z}_i$. Similarly:

$$C_{AP} \approx -\log P(D < d^+) \tag{40}$$

Maximizing C_{AP} also has a clear interpretation: it encourages all points (both positive and negative ones) to lie outside the hypersphere of radius d^+ around query point $\boldsymbol{z}_i$. Now, Eq. 38 is nothing more than an expectation over all positive distance d^+ one could sample. Therefore, such loss can be analyzed through the same lens as other DML losses, i.e., one tightness term that encourages all points from the same class as $\boldsymbol{z}_i$ to lie close to it in the embedded space, and one contrastive term that oppositely refrains all points from approaching $\boldsymbol{z}_i$ closer than its current positive points.

D On the limitations of cross-entropy

While we demonstrated that the cross-entropy loss could be competitive in comparison to pairwise losses, while being easier to optimize, there still exist scenarios for which a straightforward use of the CE loss becomes prohibitive. Hereafter, we describe two such scenarios.

Case of relative labels: The current setting assumes that absolute labels are given for each sample, *i.e.*, each sample $\boldsymbol{x}_i$ belongs to a single absolute class y_i. However, DML can be applied to more general problems where the absolute class labels are not available. Instead, one has access to relative labels that only describe the relationships between points (*e.g.*, a pair is similar or dissimilar). From these relative labels, one could still define absolute classes as sets of samples inside which every pair has a positive relationship. Note that with this definition, each sample may belong to multiple classes simultaneously, which makes the use of standard cross-entropy difficult. However, with such re-formulation, our Simplified Pairwise Cross-Entropy (SPCE), which we hereby remind:

$$\mathcal{L}_{SPCE} = \underbrace{-\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j:y_j=y_i}\boldsymbol{z}_i^\mathsf{T}\boldsymbol{z}_j}_{\text{TIGHTNESS}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\log\sum_{k=1}^{K}\exp\Big(\frac{1}{n}\sum_{j:y_j=k}\boldsymbol{z}_i^\mathsf{T}\boldsymbol{z}_j\Big)}_{\text{CONTRASTIVE}} \tag{15}$$

can handle such problems, just like any other pairwise loss.

Case of large number of classes: In some problems, the total number of classes K can grow to several millions. In such cases, even simply storing the weight matrix $\boldsymbol{\theta} \in \mathbb{R}^{K\times d}$ of the final classifier required by cross-entropy becomes prohibitive. Note that there exist heuristics to handle such problems with standard cross-entropy, such as sampling subsets of classes and solving those sub-problems instead, as was done in [49]. However, we would be introducing new training heuristics (e.g., class sampling), which defeats the initial objective of using the cross-entropy loss. Again, the SPCE loss underlying the unary cross-entropy could again handle such cases, similarly to other pairwise losses, given that it doesn't require storing such weight matrix.