Supplementary Material

BMVC 2024 Submission # 579

A Equidistant Points Generation Algorithm

As underlined in our main paper, we use a set of equidistant points $[\mathbf{p}_1, \dots, \mathbf{p}_L] \in \mathbb{R}^{d \times L}$ as vertices of a simplex. Creating such points involves positioning them in a way that the distance between any two is the same. Equation (7) describes how to achieve this for a d-dimensional simplex.

$$\mathbf{p}_j = \begin{cases} k^{-1/2}\mathbf{1} & \text{if } j = 0 \\ a\mathbf{1} + be_j & \text{if } 1 \leq j \leq d \end{cases} \tag{7}$$

where $a = -\frac{1+\sqrt{d+1}}{d^{3/2}}, \quad b = \sqrt{\frac{d+1}{d}}$, $\mathbf{1}$ is the vector containing one everywhere in $\mathbb{R}^d$ and e_j is the standard unit vector with 1 at indice j and 0 everywhere else.

To put it plainly, in d-dimensional space, we start by defining the first vertex $\mathbf{p}_0$ at $k^{-1/2}$, ensuring it lies at a specific position along the chosen axis. For the subsequent vertices, indexed by j where $1 \leq j \leq d$, the coordinates are given by $y_j = a\mathbf{1} + be_j$. The constants a and b are computed to ensure equidistance: $a = -\frac{1+\sqrt{k+1}}{k^{3/2}}$ adjusts the position along each axis, and $b = \sqrt{\frac{k+1}{k}}$ scales the points to maintain the same distance between each pair of vertices. This formulation ensures that the vertices of the simplex are evenly spread out, creating a geometrically balanced figure where all points are equidistant from each other. It is worth noting that the equidistant properties of such a structure are invariant under any rotation, reflection, or translation.

B Prototype Gradient Analysis

Here, we will make some remarks on our $\mathcal{L}_{NCC}$ based on its derivation with respect to a sample $\mathbf{z}_i$. For clarity, let's first recall that Eq. (4) represents a weighted sum over all positive samples $\mathbf{z}_i$ of the loss $\ell_{NCC}(\mathbf{z}_i)$, as defined in Eq. (8). Without loss of generality, we will consider the case where only one positive pair is formed with the sample $\mathbf{z}_p$. Furthermore, we will simplify the sum indices, irrespective of iteration over $2N$ samples or L prototypes.

$$\ell_{NCC}(\mathbf{z}_i) = \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p/\tau)}{\sum_{j \neq i} \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)} + \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{p}_{y_i}/\tau)}{\sum_{j \neq i} \exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)} \tag{8}$$

where y_i is sample i's label. We derive from this expression and demonstrate in **Proposition 1.**

Proposition 1. The gradient of $\ell_{NCC}(\mathbf{z}_i)$ shares similar structure for sample-to-sample and sample-to-prototype parts as in Eq. (9).

$$\nabla_{\mathbf{z}_i}\ell_{NCC}(\mathbf{z}_i) = \frac{1}{\tau}[P_i(\mathbf{z}_p) - N_i(\mathbf{z}_j) + P_i(\mathbf{p}_{y_i}) - N_i(\mathbf{p}_{y_j})] \tag{9}$$

where P and N represent the positive and negative contributions to SGD, respectively.

Proof. Let denote by $\cdot$ the scalar product between two vectors. We additionally assume that all input vectors $\mathbf{z}_i$, $\mathbf{z}_p$, $\mathbf{z}_j$ and $\mathbf{p}_{y_i}$ lie in the unit sphere of $\mathbb{R}^d$. Hence, the gradient is:

$$\begin{aligned}
\nabla_{\mathbf{z}_i}\ell_{NCC}(\mathbf{z}_i) &= \nabla_{\mathbf{z}_i}(\mathbf{z}_i \cdot \mathbf{z}_p/\tau) - \nabla_{\mathbf{z}_i}\left(\log\sum_j \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)\right) \\
&\quad + \nabla_{\mathbf{z}_i}(\mathbf{z}_i \cdot \mathbf{p}_{y_i}/\tau) - \nabla_{\mathbf{z}_i}\left(\log\sum_j \exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)\right) \\
&= \frac{1}{\tau}\mathbf{z}_p - \frac{1}{\sum_j \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}\sum_j \frac{\mathbf{z}_j}{\tau}\exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau) \\
&\quad + \frac{1}{\tau}\mathbf{p}_{y_i} - \frac{1}{\sum_j \exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)}\sum_j \frac{\mathbf{z}_j}{\tau}\exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau) \\
&= \frac{1}{\tau}\left(\mathbf{z}_p - \frac{\sum_j \mathbf{z}_j \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}{\sum_j \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}\right) + \frac{1}{\tau}\left(\mathbf{p}_{y_i} - \frac{\sum_j \mathbf{p}_{y_j}\exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)}{\sum_j \exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)}\right) \\
&= \frac{1}{\tau}\left[\mathbf{z}_p - \frac{\sum_j \mathbf{z}_j \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)}{\sum_j \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)} + \mathbf{p}_{y_i} - \frac{\sum_j \mathbf{p}_{y_j}\exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)}{\sum_j \exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)}\right] \\
&= \frac{1}{\tau}\left[P_i(\mathbf{z}_p) - N_i(\mathbf{z}_j) + P_i(\mathbf{p}_{y_i}) - N_i(\mathbf{p}_{y_j})\right].
\end{aligned}$$

□

The roles of the terms P_i and N_i can be summarized as follows: Both P_i terms act as pulling forces; however, $P_i(\mathbf{p}y_i)$ remains static. Additionally, it is important to note that both N_i terms represent weighted means of pushing forces, where the weighting coefficients are either samples or prototypes themselves. Here again, $N_i(\mathbf{p}_{y_j})$ serves as a static pushing term with respect to each $\mathbf{z}_i$. In conclusion, prototypes exhibit an accumulating and stabilizing effect on $\mathcal{L}_{scl}$.