Learning representations on Lp hyperspheres: The equivalence of loss functions in a MAP approach

Nicolas Michel NICOLAS@CVM.T.U-TOKYO.AC.JP
The University of Tokyo

Jean-François Bercher JF.BERCHER@ESIEE.FR
Univ Gustave Eiffel, LIGM, ESIEE-Paris, France

Toshihiko Yamasaki YAMASAKI@CVM.T.U-TOKYO.AC.JP
The University of Tokyo

Editors: List of editors' names

Abstract

A common practice when training Deep Neural Networks is to force the learned representations to lie on the standard unit hypersphere, with respect to the L_2 norms. Such practice has been shown to improve both the stability and final performances of DNNs in many applications. In this paper, we derive a unified theoretical framework for learning representation on any L_p hyperspheres for classification tasks, based on Maximum A Posteriori (MAP) modeling. Specifically, we give an expression of the probability distribution of multivariate Gaussians projected on any L_p hypersphere and derive the general associated loss function. Additionally, we show that this framework demonstrates the theoretical equivalence of all projections on L_p hyperspheres through the MAP modeling. It also provides a new interpretation of traditional Softmax Cross Entropy with temperature (SCE-τ) loss functions. Experiments on standard computer vision datasets give an empirical validation of the equivalence of projections on L_p unit hyperspheres when using adequate objectives. It also shows that the SCE-τ on projected representations, with optimally chosen temperature, shows comparable performances.

Keywords: Representation Learning, Hypersphere, Maximum A Posteriori.

1. Introduction

Cross-entropy (CE) is the most commonly used loss function for classification, even though it is often modified Ahn et al. (2021); Caccia et al. (2022); Wang et al. (2017) or coupled with additional loss terms Hinton et al. (2014); Li et al. (2019). On the other hand, many studies in the literature address designing output normalization. Bouchard (2007) introduced upper bounds for improving softmax computation stability. De Brebisson and Vincent (2016) introduced a family of functions behaving as normalizing functions and gave experimental justifications for softmax alternatives. Other sparse alternatives have similarly been developed Martins and Astudillo (2016); Laha et al. (2018); Liu et al. (2017). Further studies considered the probabilistic modeling of the trained feature space explicitly. Wan et al. (2018) have leveraged a Gaussian mixture model coupled with a CE. Additional studies also consider a similar setting, opposing the obtained loss function to the traditional Softmax Cross Entropy (SCE) Yan et al. (2020). It is known that the softmax operation can be interpreted as resulting from the formulation of the a posteriori distribution of the class given the data and that the search for the a posteriori maximum leads, with a Gaussian assumption, to the standard cross-entropy criterion; see for example (Bishop, 2006, Section 4.2, pages 197-199). A standard practice when training Deep Neural Networks is to force the learned representations to lie on the standard unit hypersphere, with respect to the L_2 norms. Such practice has been shown to improve both the stability and final performances of DNNs in many applications, see e.g. Wang et al. (2017); Tian et al. (2020); Zimmermann et al. (2021); Chen et al. (2020). However, this is usually not directly accounted for when deriving a loss function for the whole classification process, including the projection step.

In this paper, we first recall the MAP approach for the DNN classification problem and give an explicit connection to SCE and its variant Softmax Cross-Entropy with temperature (SCE-τ) and bias, see Zhang et al. (2018); Agarwala et al. (2020). Indeed, we show that SCE can be interpreted as a MAP with a class-conditional isotropic Gaussian hypothesis on the standard *scaled*-simplex (the standard simplex scaled by a factor r). This corresponds to assuming that the within-class residuals around the class mean, located on one of the simplex axes, can be approximated by a Gaussian distribution. Similarly, we demonstrate that the temperature parameter used for re-scaling the network outputs in SCE-τ can be expressed as the ratio between the scaling factor r and the Gaussian distributions variance v. The insights given by the MAP approach allow us to give a meaningful interpretation of the SCE and, more than that, to consider more general scenarios. Specifically, we investigate the impact of a particular family of nonlinear output transformations: projections onto L_p hyperspheres, notably to compare performances with SCE and assess the impact of p. While the already mentioned L_2 projections are commonly adopted, the general case of L_p projections is widely unexplored. Different L_p norms change the geometry of hyperspheres, affecting how data is projected and separated. For instance, with $p > 2$, hyperspheres become more flattened, while $p < 2$ makes them more angular, which can enhance class separation in certain directions.

In the MAP approach to learn representations, we show the equivalence of all L_p projections, which all lead to the same loss function. We then focus on the particular case of a Gaussian assumption for the learned features and derive the expression of the probability distribution for Gaussian distributions projected on general L_p hyperspheres. This introduces the Projected Gaussian Distribution (PGD), a generalization of the Angular Gaussian Distribution presented in Michel et al. (2024). Eventually, we experiment with PGD through the MAP framework as well as with SCE-τ on output projected on the L_p unit-sphere. Finally, we conclude that PGD and SCE-τ can lead to **comparable performances, in case of a L_p projection layer**, provided optimal v values are used for any values of p and show that leveraging PGD or projecting on the hypercube can **improve stability** concerning the variance. In summary, we make the following contributions: (1) we highlight a connection between the MAP approach and SCE variants, which give additional insight on the loss function; (2) we propose an expression of PGD, the distribution of a Gaussian distribution on any L_p hypersphere; (3) we show that projecting on the hypercube or leveraging PGD benefits stability with regard to v, while maintaining performance on par with the best SCE-τ strategy.

2. Related work

Softmax-Cross Entropy and its variants. One of the most widely used loss functions for classification tasks is the Cross-Entropy, commonly combined with the softmax function applied to the output layer Goodfellow et al. (2016). Numerous works have been proposed as alternatives to the traditional softmax operator, such as sparse alternative Martins and Astudillo (2016); Liu et al. (2017); Laha et al. (2018) or spherical softmax De Brebisson and Vincent (2016). Similarly, prototype-based alternatives to SCE have been developed Bytyqi et al. (2023); Wei et al. (2023); Mettes et al. (2019). Another variant of SCE introduces a temperature parameter Wang et al. (2017); Pang et al. (2020), with the corresponding loss:

$$\mathcal{L}_{CE}(\boldsymbol{z}) = -\sum_{c=1}^{L} \nVdash(y = c) \log \frac{e^{z_c/\tau}}{\sum_{j=1}^{L} e^{z_j/\tau}} \tag{1}$$

where y is the true label of $\boldsymbol{z}$, L the total number of classes, z_c the c^{th} component of $\boldsymbol{z}$ and $\tau \in \mathbb{R}^{+\star}$ the temperature. The usage of temperature similarly goes beyond SCE and has been studied in contrastive learning Zhang et al. (2021); Khosla et al. (2020); Chen et al. (2020). However, such studies are mostly empirical, and often lack theorical analysis.

MAP for learning representations. Maximum A Posteriori is a fundamental probabilistic method and has been applied to countless problems Gauvain and Lee (1994); Santini and Del Bimbo (1995); in the context of DNNs, Michel et al. (2024) applied a natural MAP framework for learning representations on the unit-hypersphere. Other probabilistic modeling also derived similar loss functions Hasnat et al. (2017). However, to the best of our knowledge, no explicit link to the SCE-τ loss and its implication in terms of interpretation has been developed in earlier research.

Projection on L_p hyperspheres. While projection on the unit-hypersphere (a.k.a. normalization) is a common practice in representation learning Grill et al. (2020); Khosla et al. (2020); Mettes et al. (2019); Michel et al. (2024), it is, to the best of our knowledge, bounded to the L_2 hypersphere. In adversarial training, L_∞ metric is also used for measuring the distance between the original and the attacked sample Mao et al. (2019); Tramer and Boneh (2019). Few studies of the general family of L_p projections for DNNs in the context of image classification exist. An attempt to leverage L_p normalization of the penultimate layer can be found in Trivedi et al. (2022).

3. From MAP to SCE

3.1. Posterior expression from latent representation

As introduced in Section 1, we are interested in expressing the posterior $p(c|\boldsymbol{x})$, the probability distribution that the class random variable takes the value c, given the fact that the random vector $\mathbf{x}$ is equal to the actual input vector $\boldsymbol{x}$. We start from the consideration that Deep Neural Networks (DNN) are fundamentally encoders that can learn a mapping between an input $\mathbf{x} \in \mathbb{R}^D$ to a latent representation $\mathbf{z} \in \mathbb{R}^d$, where D and d are the dimensions of the input and the latent representation respectively and $D \gg d$. From this point, the posterior estimation problem becomes estimating $p(c|\boldsymbol{z})$. By the Bayes rule and expressing $f(\boldsymbol{z})$ by marginalizing across considered classes, $p(c|\boldsymbol{z})$ can be expressed as:

$$p(c|\boldsymbol{z}) = \frac{f_c(\boldsymbol{z})\pi_c}{\sum_{\ell=1}^{L} f_\ell(\boldsymbol{z})\pi_\ell} \tag{2}$$

where π_ℓ are class priors, L the total number of classes and $f_\ell(\boldsymbol{z})$ the conditional p.d.f. of $\mathbf{z}$ given c. The previous expression can similarly be written with latent representations as DNN outputs such that $\mathbf{z} = \Phi_\theta(\mathbf{x})$, with θ the trainable DNN parameters.

3.2. Maximum A Posteriori log-loss

For a set of b of observations $(\boldsymbol{z}_i, y_i)_{1\leq i\leq b}$, where the $y_i \in [\![1, L]\!]$ are the labels of classes and $\boldsymbol{z}_i \in \mathbb{R}^d$, we want to maximize $p(y_1 \cdots y_b|\boldsymbol{z}_1 \cdots \boldsymbol{z}_b)$. Let us consider such observations to be independent. The objective becomes maximizing $\prod_{c=1}^{L}\prod_{i\in I_c} p(c|\boldsymbol{z}_i)$ with $I_c = \{i \in [\![1, b]\!] \mid y_i = c\}$. The posterior distribution can thus be expressed as

$$p(y_1 \cdots y_b|\boldsymbol{z}_1 \cdots \boldsymbol{z}_b) = \prod_{c=1}^{L}\prod_{i\in I_c} \frac{f_c(\boldsymbol{z}_i)\pi_c}{\sum_{\ell=1}^{L} f_\ell(\boldsymbol{z}_i)\pi_\ell}. \tag{3}$$

A more practical log-loss form can obtained from (3) by taking the average of the logarithm:

$$\mathcal{L}^{\log}_{\mathrm{MAP}}(\mathcal{B}, \theta) = -\frac{1}{|\mathcal{B}|}\sum_{c=1}^{L}\sum_{i\in I_c} \log \frac{f_c\big(\Phi_\theta(\boldsymbol{x}_i)\big)\pi_c}{\displaystyle\sum_{\ell=1}^{L} f_l\big(\Phi_\theta(\boldsymbol{x}_i)\big)\pi_\ell} \tag{4}$$

With $|\mathcal{B}|$ the size of batch $\mathcal{B} = (\boldsymbol{x}_i, y_i)_{1\leq i\leq b}$. In the MAP framework, we minimize $\mathcal{L}^{log}_{MAP}$.

3.3. Gaussian hypothesis and equal priors

The MAP framework described above depends on the choice of the class-conditional p.d.f. $f_c(.)$. In classification tasks, we know that the learned features concentrate around the means of each class. The Gaussian hypothesis assumes that within-class variations around the mean can be approximated by a Gaussian distribution. This modeling is supported by e.g. Lee et al. (2018); Asao et al. (2022); Doshi et al. (2023); Gao et al. (2023) who argue for this behavior for infinitely wide deep networks by an application of the multivariate Central Limit. This leads to Proposition 1 with proof in Appendix E.

Proposition 1 *Let $\{r_l\}_{1\leq l\leq L}$ be a basis of $\mathbb{R}^L$ such that $r_l = r \cdot e_l$ with $r \in \mathbb{R}$ and $\{e_l\}_{1\leq l\leq L}$ the standard basis of $\mathbb{R}^L$. Under the following assumptions: (1) the conditional probability density functions $\{f_l(.)\}_{1\leq l\leq L}$ follow an isotropic Gaussian distribution of variance v centered around means $\{r_l\}_{1\leq l\leq L}$; (2) classes priors $\{\pi_l\}_{1\leq l\leq L}$ are equal; the loss $\mathcal{L}^{\log}_{MAP}$ takes the following form:*

$$\mathcal{L}^{log}_{MAP}(\mathcal{B}, \theta) = -\frac{1}{|\mathcal{B}|}\sum_{c=1}^{L}\sum_{i\in I_c} \log \frac{e^{\frac{r}{v}\Phi_\theta(\boldsymbol{x}_i)_c}}{\displaystyle\sum_{\ell=1}^{L} e^{\frac{r}{v}\Phi_\theta(\boldsymbol{x}_i)_l}} \tag{5}$$

with $\Phi_\theta(\boldsymbol{x}_i)_c$ the c-th component of $\Phi_\theta(\boldsymbol{x}_i)$, the output of the model given the input x_i.

3.4. Connection with SCE and its variants

Under simple assumptions, the MAP framework leads to the $\mathcal{L}^{log}_{MAP}$ as defined in (5). When $\frac{r}{v} = 1$, we recover the usual SCE loss. Additionally, if we define $\tau = \frac{v}{r}$, then we recover the SCE-τ loss. Thus, we can interpret SCE-τ as a MAP with a class conditional Gaussian hypothesis on the standard *scaled*-simplex whose scaling ratio r and variance v are conditioned such that $\frac{r}{v} = \tau$. This statement similarly holds for SCE when $\tau = 1$. Furthermore, the Softmax operation appears naturally in this modeling. From this interpretation, two scenarios can be identified. If the learned representation is projected on the unit-hypersphere, $r = 1$, and if we assume that these projections are also Gaussian, then variance v follows by the remodeling as $\tau = v$. Of course, the Gaussian assumption for the projected distribution onto the hypersphere is questionable. In Section 4.3, we discuss the validity of this assumption, and we give the expression of the Projected Gaussian Distribution in Section 4. Moreover, if the learned representations are not constrained, it follows that r and v are learned such that the relation $\tau = \frac{v}{r}$ is respected. Connection with more variants can be found in Appendix.

4. Learning on the L_p hypersphere

We showed that minimizing an SCE-τ objective with representations learned on the unit-sphere gives control over the Gaussian variance, provided that the projection itself is considered Gaussian. In this section, we discuss the impact of invertible and non-invertible transformations on the resulting distribution and on MAP training objective, in the case of projections on L_p hyperspheres.

4.1. MAP with additional transformations

4.1.1. Invertible transformations

In the above, we have modeled the conditional distribution $f_c(.)$ for a class c through the intermediate variable $\mathbf{z} \in \mathbb{R}^L$, the neural network output. Let us now consider that an additional transformation $h : \mathbb{R}^L \to \mathbb{R}^L$ is applied to $\mathbf{z}$. If $h(.)$ is a one-to-one invertible transformation, the conditional probability density function q_c of the resulting variable $\boldsymbol{\zeta} = h(\mathbf{z})$ can be expressed as $q_c(\boldsymbol{\zeta}) = \frac{f_c(\boldsymbol{z})}{|J_h(\boldsymbol{\zeta})|}$, see Murphy (2022). With $J_h(\boldsymbol{\zeta})$ the Jacobian of h and $|J_h(\boldsymbol{\zeta})|$ its determinant evaluated at $\boldsymbol{\zeta}$. Starting from Equation 2, it follows that trying to express the posterior $p(c|\boldsymbol{\zeta})$ with regard to $\boldsymbol{\zeta}$ leads to:

$$p(c|\boldsymbol{\zeta}) = \frac{q_c(\boldsymbol{\zeta})\pi_c}{\sum_{\ell=1}^{L} q_\ell(\boldsymbol{\zeta})\pi_\ell} = \frac{\frac{f_c(\boldsymbol{z})}{|J_h(\boldsymbol{\zeta})|}\pi_c}{\sum_{\ell=1}^{L} \frac{f_\ell(\boldsymbol{z})}{|J_h(\boldsymbol{\zeta})|}\pi_\ell} = p(c|\boldsymbol{z}) \tag{6}$$

Thus, combining invertible transformations with the MAP framework gives strictly identical a posteriori probability distributions. This observation also holds for Cross-Entropy given the equivalence showed in Section 3.3.

4.1.2. Projections on L_p hyperspheres and their equivalence in the MAP setting

This family of transformations reduces the vector's dimensionality, resulting in a non-invertible transformation. We define such transformations as $T_{l_p} : \mathbb{R}^L \to \mathbb{R}^L$ on $\boldsymbol{z} = (z_1, \cdots, z_L) \in \mathbb{R}^L$ such that $T_{l_p}(\boldsymbol{z}) = \boldsymbol{z}_p = \frac{\boldsymbol{z}}{||\boldsymbol{z}||_p}$, with $\boldsymbol{z}$ the output representation of the neural network, $||\boldsymbol{z}||_p = (\sum_{i=1}^{L} |z_i|^p)^{1/p}$ and $|z_i|$ the absolute value of z_i.

Equivalence of projections. There is a one-to-one correspondence between projections for different values of p, e.g. between a projection $\boldsymbol{z}_p$ on the p-hypersphere and a projection $\boldsymbol{z}_q$ on the q-hypersphere. Indeed, $\boldsymbol{z}_q = \frac{\boldsymbol{z}}{||\boldsymbol{z}||_q} = \frac{\boldsymbol{z}_p}{||\boldsymbol{z}_p||_q}$, because the normalizations by $||\boldsymbol{z}||_p$ simplify in the last right term. Therefore, we see by (6) that projections onto different L_p hyperspheres lead to the same posterior, hence the same loss function, no matter the value of p used when projecting. It follows that, in the MAP framework, every projection is equivalent to a projection on the unit hypersphere.

4.2. Projected Gaussian Distribution

Given the previous analysis, we argue that using SCE-τ on projected representation is not theoretically justified. Indeed, the result of a radial projection (equivalently, normalization) of a Gaussian distribution on the L_p unit hypersphere is most likely not Gaussian. Additionally, such transformation being non invertible, the obtained MAP objective should be adapted accordingly. We propose an analytical expression for the projection of a Gaussian distribution on any L_p hypersphere and give the proof of this result in Appendix F.

Proposition 2 *Let $p, d \in \mathbb{N}^{+\star}$. For $\mathbf{z} \in \mathbb{R}^d$ following a d-variate Gaussian of mean $\boldsymbol{\mu} \in \mathcal{S}_p^d$ and covariance matrix $\Sigma = \sigma^2 I$, the distribution of $\mathbf{u}$, the projection of $\mathbf{z}$ on $\mathcal{S}_p^d$ such that $\mathbf{u} = \frac{\mathbf{z}}{||\mathbf{z}||_p}$ is defined by:*

$$g_\kappa^{PGD}(\boldsymbol{u}, \boldsymbol{\mu}) = a_\kappa e^{-\frac{1}{2}\kappa^2} \sum_{n=0}^{\infty} \frac{(\kappa \frac{\boldsymbol{u}^T \cdot \boldsymbol{\mu}}{||\boldsymbol{u}||_2 \cdot ||\boldsymbol{\mu}||_2})^n \, \Gamma\left(\frac{d}{2} + \frac{n}{2}\right)}{n! \; \Gamma\left(\frac{d}{2}\right)} \tag{7}$$

with $\kappa^2 = \frac{||\boldsymbol{\mu}||_2}{\sigma^2}$, $a_\kappa = \frac{\Gamma(\frac{d}{2})(\boldsymbol{u}^T \boldsymbol{u})^{-\frac{d}{2}}}{2\pi^{\frac{d}{2}} w}$ a normalization factor and $w = ||u||_{2(p-1)}^{(p-1)}$

The expression for the lost function $\mathcal{L}_{PGD}^p$ on the standard simplex follows directly by combining PGD from (7) and the MAP log-loss from (4):

$$\mathcal{L}_{PGD}^p(\mathcal{B}, \theta) = -\frac{1}{|\mathcal{B}|} \sum_{c=1}^{L} \sum_{i \in I_c} \log \frac{g_\kappa^{PGD}(T_{l_p}(\Phi_\theta(\boldsymbol{x}_i)), \boldsymbol{e_c})}{\sum_{\ell=1}^{L} g_\kappa^{PGD}(T_{l_p}(\Phi_\theta(\boldsymbol{x}_i)), \boldsymbol{e_\ell})}. \tag{8}$$

Equivalence of projections. In this particular case, we recover the projection equivalence underlined in Section 4.1.2. Indeed, changing the value of p in (7) only impacts the normalization term a_κ. The other term depending on $\boldsymbol{u}$, denoting $\boldsymbol{u_p} = \frac{\boldsymbol{u}}{||\boldsymbol{u}||_p}$, we have for any $p \in \mathbb{N}^{+\star}$:

$$\frac{\boldsymbol{u_p}^T \cdot \boldsymbol{\mu}}{||\boldsymbol{u_p}||_2 \cdot ||\boldsymbol{\mu}||_2} = \frac{\frac{\boldsymbol{u}}{||\boldsymbol{u}||_{\boldsymbol{p}}}^T \cdot \boldsymbol{\mu}}{||\frac{\boldsymbol{u}}{||\boldsymbol{u}||_{\boldsymbol{p}}}||_2 \cdot ||\boldsymbol{\mu}||_2} = \frac{\boldsymbol{u}^T \cdot \boldsymbol{\mu}}{||\boldsymbol{u}||_2 \cdot ||\boldsymbol{\mu}||_2} \tag{9}$$

Therefore, when plugging the expression of PGD from (7) into the MAP log-loss expression from (4), the normalization factors simplify, and the resulting loss is unchanged.

4.3. SCE-τ on L_p hypersphere

SCE-τ can be used with representations projected onto the L_p hypersphere, even though the Gaussian assumption is not fulfilled. Various works have shown that SCE-τ can empirically achieve competitive performances on the L_2 hypersphere De Brebisson and Vincent (2016); Wang et al. (2017). In our setting, a potential justification of such results is the validity of a Gaussian projection approximation for small variance values. Indeed, the projection of a multivariate Gaussian along one of its components is a Gaussian. We refer to this projection as an axial projection. While such a result does not hold for radial projections (or normalizations), we can show that the radial and axial projections tend to result in the same projections when v tends to 0. Let us consider $\boldsymbol{z} = [z_1, \cdots, z_L] \in \mathbb{R}^L$, a vector sampled from a Gaussian centered around $\boldsymbol{e_1} = [1, 0 \cdots, 0] \in \mathbb{R}^L$. It follows that $\boldsymbol{z}_p = \left[\frac{z_1}{||\boldsymbol{z}||_p}, \frac{z_2}{||\boldsymbol{z}||_p}, \cdots, \frac{z_L}{||\boldsymbol{z}||_p}\right]$ and $\boldsymbol{z}_a = [1, z_2, \cdots, z_L]$, with $\boldsymbol{z}_p$ and $\boldsymbol{z}_a$ being the radial L_p and axial projections respectively. Eventually,

$$\begin{cases} \boldsymbol{z}_a & \to \boldsymbol{e_1} \\ v & \to 0 \end{cases} \qquad \begin{cases} \boldsymbol{z}_p & \to \boldsymbol{e_1} \\ v & \to 0 \end{cases} \qquad \begin{cases} ||\boldsymbol{z}_p - \boldsymbol{z}_a||_2 & \to 0 \\ v & \to 0 \end{cases} \tag{10}$$

Hence, the smaller the variance, the more likely axial and radial projections will lead to the same resulting Gaussian distribution. A geometric interpretation is that for small variance values, the hypersphere surface around the mean can be approximated by a plane perpendicular to the mean direction. Of course, such approximation differs for different values of p, in the case of $p = \infty$, the surface is a plane perpendicular to the mean. In the case of $p = 2$, the surface might be considered planar locally. Hence, we expect the optimal value of v when training with SCE-τ to be proportional to the value of p.

5. Experiments

We conducted experiments on standard datasets for image classification. We compare the performances of SCE, SCE-τ, and the loss function derived from MAP with the PGD model and confirm our intuitions based on our insights from the MAP modeling. More details regarding the experimental setup can be found in Appendix A.

5.1. Results

Accuracy. Table 1 shows the obtained accuracy at the end of training for SCE, SCE-τ and PGD losses on considered datasets. The value of p indicates the hypersphere on which representations are projected. For baseline, performances of SCE and SCE-τ without projection are also reported. Following previous studies, it can be observed that projecting representations on the L_2 hypersphere leads to a significant increase in performance, given that the optimal variance (or equivalently temperature) is used. Furthermore, we observe that similar performances can be obtained on all datasets for any projection strategies with SCE-τ. Eventually, the obtained results with PGD are on par with the best SCE-τ results. For PGD, we indicate no values of p since the loss is independent of the projection strategy.

Table 1: Accuracy (%) of different losses and projections strategies on CIFAR10, CIFAR100, and ImageNet. SCE corresponds to Softmax Cross-Entropy and SCE-τ corresponds to SCE with temperature and PGD to the PGD loss defined in (8). The values of p and v used for training are similarly reported. For CIFAR10 and CIFAR100, the average and standard deviation over 5 runs are reported, while only 1 run was realised for ImageNet100.

Loss	p	**CIFAR10**		**CIFAR100**		**ImageNet100**	
		Acc.	v	Acc.	v	Acc.	v
SCE	no proj.	90.44±0.44	N/A	65.44±0.64	N/A	63.38	N/A
SCE-τ	no proj.	90.93±0.31	2.3	66.20±0.69	2.7	64.16	2.7
SCE-τ	$p = 0.5$	92.15±0.19	0.006	68.56±0.33	5e-05	66.52	5e-05
SCE-τ	$p = 1$	92.48±0.13	0.15	68.62±0.38	0.007	65.84	0.007
SCE-τ	$p = 1.5$	92.32±0.30	0.30	68.19±0.45	0.035	67.32	0.025
SCE-τ	$p = 2$	92.14±0.21	0.45	68.67±0.48	0.050	67.34	0.050
SCE-τ	$p = 3$	92.22±0.45	0.50	68.90±0.30	0.09	66.98	0.09
SCE-τ	$p = \infty$	91.91±0.27	0.40	68.69±0.37	0.22	67.16	0.22
PGD	any	92.36±0.26	0.35	68.84±0.18	0.12	66.30	0.21

Impact of v. We study the impact of the variance parameter for SCE-τ and PGD losses on CIFAR100. Figure 1 shows the accuracy at the end of training with SCE-τ on CIFAR100 for various values of p and v. For each value of p, an optimal value of v can be found to obtain the best performances. Notably, a strong performance degradation occurs for large variances rather than for smaller variances. However, a lower variance value might hinder training stability. Additionally, according to the intuition given in Section 4.3 and similar to the results presented in Table 1, the larger the value of p, the greater the resulting optimal variance is. Plus, SCE-τ performances gain in stability with regard to v for larger values of p. We discuss this phenomenon in more detail in Section 5.2. Additional experiments on CIFAR-10 are provided in Appendix B. Additinally, PGD loss is invariant with p, as detailed in Appendix B, PGD exhibits similar performances to SCE-τ with $p = \infty$, not only in terms of maximum performance but also in terms of stability with regard to v. We discuss such similarity in Section 5.2.

5.2. Discussions

From the results above, we can make several observations. First, SCE-τ and PGD losses tend to produce similar results on L_p hyperspheres. This can be explained by the fact that Gaussian projections remain close to Gaussian for small values of v, making SCE-τ theoretically well-grounded since it is equivalent to the MAP log-loss. With appropriate variance, both losses are expected to converge to similar solutions, which explains their comparable accuracies. Second, sensitivity to v is stronger for smaller values of p. This arises from the geometry of the L_p hypersphere: when $p < 1$, the shape resembles an astroid with sharp corners near the basis vectors, while for $p = \infty$ it becomes a hypercube,

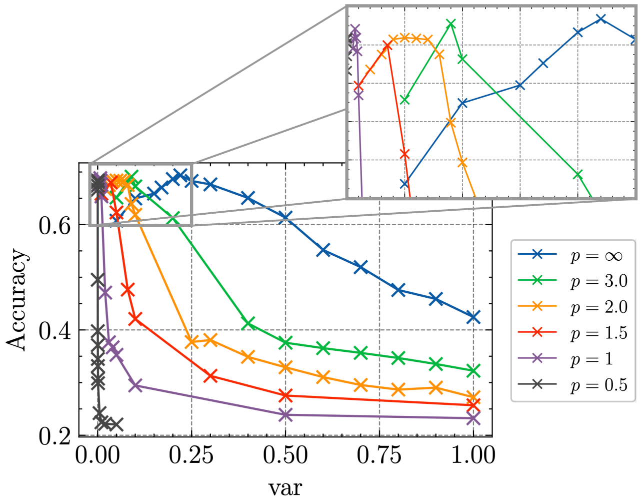


Figure 1: Accuracy at the end of training a ResNet18 on CIFAR100 with a MAP objective (or equivalently SCE-τ) for different (p, v) values. The top left part is zoomed in for better readability.

locally flat around each face. Larger values of p therefore make the Gaussian approximation more valid, and this approximation remains robust even at higher variances. Third, PGD is generally more stable than SCE-τ when $p \neq \infty$, though both perform similarly when $p = \infty$. Since PGD models the true radial projection distribution, it does not depend on the Gaussian approximation and remains reliable for larger v. However, when v grows too large, performance still declines, especially on CIFAR100, because overlapping Gaussians reduce discriminability. Overall, these findings highlight that the infinite norm ($p = \infty$) is particularly appealing: it supports both SCE-τ and PGD as sound modeling choices, provides the most accurate Gaussian approximation, and is computationally simple and stable. As such, L_∞ should be considered a practical alternative to the commonly used L_2 norm in training DNNs.

6. Conclusion

This paper provides a unified perspective on the connection between output normalization and loss functions in classification problems. By extending the Maximum-a-Posteriori (MAP) approach to encompass both the loss function and output normalization, we have established theoretical connections between the Softmax Cross-entropy (SCE) and its variants, notably including SCE-τ. Our results demonstrate that SCE-τ can be interpreted as a MAP with a class-conditional isotropic Gaussian hypothesis on the standard simplex and that the temperature can be expressed as the ratio between, the scaling factor and the variance of Gaussian distributions. However, we indicated that such an objective is not theoretically adapted when projecting on the L_p hypersphere. Therefore, we have introduced the Projected Gaussian Distribution (PGD) to model Gaussian distributions projected on any L_p hypersphere. We showed that in our framework, projections on L_p hyperspheres are equivalent for all values of p. Moreover, that SCE-τ, is a valid approximation for small variance values. Finally, we give evidence that PGD and SCE-τ on the hypercube present several advantages over other values of p, such as greater stability with respect to v and computational simplicity in the case of the hypercube.

Eventually the modeling is based on the assumption that the network outputs can be approximated by a Gaussian distribution; which can be a limitation in some specific cases. Presented performances and comparisons are established with a specific DNN and problem setting (image classification); of course, different figures can be obtained with other settings.

References

Atish Agarwala, Jeffrey Pennington, Yann Dauphin, and Sam Schoenholz. Temperature check: theory and practice for training models with softmax-cross-entropy losses. *arXiv preprint arXiv:2010.07344*, 2020.

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. SS-IL: Separated Softmax for Incremental Learning. In *IEEE/CVF International Conference on Computer Vision*, pages 824–833, 2021.

Yasuhiko Asao, Ryotaro Sakamoto, and Shiro Takagi. Convergence of neural networks to gaussian mixture distribution. *arXiv preprint arXiv:2204.12100*, 2022.

Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, 2006. ISBN 978-0-387-31073-2.

Guillaume Bouchard. Efficient bounds for the softmax function and applications to approximate inference in hybrid models. In *NIPS 2007 workshop for approximate Bayesian inference in continuous/hybrid systems*, volume 6, 2007.

Qendrim Bytyqi, Nicola Wolpert, Elmar Schömer, and Ulrich Schwanecke. Prototype softmax cross entropy: A new perspective on softmax cross entropy. In *Scandinavian Conference on Image Analysis*, pages 16–31. Springer, 2023.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *The Tenth International Conference on Learning Representations*, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020.

Richard Courant. *Differential and Integral Calculus, Volume 2.* Wiley Classics Library. Wiley, 2011. ISBN 9781118031483.

Alexandre De Brebisson and Pascal Vincent. An exploration of softmax alternatives belonging to the spherical loss family. In *The Fourth International Conference on Learning Representations*, 2016.

Matthias De Lange and Tinne Tuytelaars. Continual Prototype Evolution: Learning Online from Non-Stationary Data Streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8250–8259, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 248–255, 2009.

Darshil Doshi, Tianyu He, and Andrey Gromov. Critical initialization of wide and deep neural networks using partial jacobians: General theory and applications. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 40054–40095, 2023.

Tianxiang Gao, Xiaokai Huo, Hailiang Liu, and Hongyang Gao. Wide neural networks as gaussian processes: Lessons from deep equilibrium models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 54918–54951, 2023.

J-L Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2:291–298, 1994.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284, 2020.

Md Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von Mises-Fisher Mixture Model-based Deep learning: Application to Face Verification. *arXiv preprint arXiv:1706.04264*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *presented at NIPS 2014 Deep Learning Workshop*, 2014. arXiv:1503.02531.

Stella Ho, Ming Liu, Lan Du, Longxiang Gao, and Yong Xiang. Prototype-guided memory replay for continual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Quentin Jodelet, Xin Liu, and Tsuyoshi Murata. Balanced softmax cross-entropy for incremental learning. In *International Conference on Artificial Neural Networks*, pages 385–396. Springer, 2021.

P.E. Jupp and K.V. Mardia. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2009. ISBN 9780470317815.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations*, 2015.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Master's thesis, University of Toronto, 2009.

Anirban Laha, Saneem Ahmed Chemmengath, Priyanka Agrawal, Mitesh Khapra, Karthik Sankaranarayanan, and Harish G Ramaswamy. On controllable sparse alternatives to softmax. *Advances in Neural Information Processing Systems*, 31, 2018.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. In *The Sixth International Conference on Learning Representations*, 2018.

Gwen Legate, Lucas Caccia, and Eugene Belilovsky. Re-weighted softmax cross-entropy to control forgetting in federated learning. In *Conference on Lifelong Learning Agents*, pages 764–780, 2023.

Xiaoxu Li, Dongliang Chang, Tao Tian, and Jie Cao. Large-margin regularized softmax cross-entropy loss. *IEEE access*, 7:19572–19578, 2019.

Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. PCR: Proxy-based Contrastive Replay for Online Class-incremental Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24246–24255, 2023.

Weiyang Liu, Yan-Ming Zhang, Xingguo Li, Zhiding Yu, Bo Dai, Tuo Zhao, and Le Song. Deep hyperspherical learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623, 2016.

Pascal Mettes, Elise Van der Pol, and Cees Snoek. Hyperspherical prototype networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Nicolas Michel, Giovanni Chierchia, Romain Negrel, and Jean-François Bercher. Learning Representations on the Unit Sphere: Investigating Angular Gaussian and von Mises-Fisher Distributions for Online Continual Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14350–14358, 2024.

Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. In *The Eigth International Conference on Learning Representations*, 2020.

Tarmo M. Pukkila and C. Radhakrishna Rao. Pattern recognition based on scale invariant discriminant functions. *Information Sciences*, 45(3):379–389, 1988. ISSN 0020-0255.

Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in Neural Information Processing Systems*, 33: 4175–4186, 2020.

Simone Santini and Alberto Del Bimbo. Recurrent neural networks can be trained to be maximum a posteriori probability classifiers. *Neural Networks*, 8(1):25–29, 1995.

John G. Saw. A family of distributions on the m-sphere and some hypothesis tests. *Biometrika*, 65(1):69–73, 1978. ISSN 00063444.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *The Eigth International Conference on Learning Representations*, 2020.

Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in Neural Information Processing Systems*, 32, 2019.

Chintan Trivedi, Konstantinos Makantasis, Antonios Liapis, and Georgios N Yannakakis. Revisiting Lp-constrained softmax loss: A comprehensive study. *arXiv preprint arXiv:2206.09616*, 2022.

Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Eecognition*, pages 9117–9126, 2018.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017.

Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18764–18774, 2023.

Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert YS Lam. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1050–1060, 2020.

Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Oliver Zhang, Mike Wu, Jasmine Bayrooti, and Noah Goodman. Temperature as uncertainty in contrastive learning. In *NeurIPS 2021 Workshop: Self-Supervised Learning - Theory and Practice*, 2021.

Xiao Zhang, Rui Zhao, Yu Qiao, and Hongsheng Li. Rbf-softmax: Learning deep representative prototypes with radial basis function softmax. In *Proceedings of 16th European Conference Computer Vision*, pages 296–311, 2020.

Xu Zhang, Felix Xinnan Yu, Svebor Karaman, Wei Zhang, and Shih-Fu Chang. Heated-up softmax embedding. *arXiv preprint arXiv:1809.04157*, 2018.

Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 12979–12990, 2021.

Daniel Zwillinger, Victor Moll, I.S. Gradshteyn, and I.M. Ryzhik, editors. *Table of Integrals, Series, and Products (Eighth Edition)*. Academic Press, Boston, eighth edition edition, 2014. ISBN 978-0-12-384933-5.

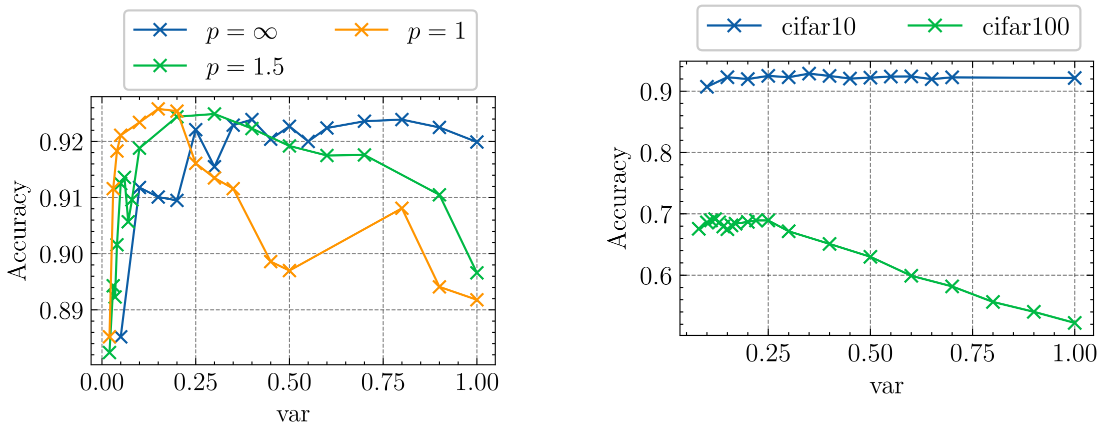
Appendix

Appendix A. Experimental setup

Datasets. To compare the presented losses, we use 3 benchmark datasets. CIFAR10 Krizhevsky (2009) is composed of 50,000 train images and 10,000 test images for 10 classes. All images are of size 32×32. CIFAR100 Krizhevsky (2009) is similarly composed of 50,000 32×32 train images and 10,000 test images but has 100 classes. Imagenet100 is a subset of the ILSVRC-2012 Deng et al. (2009) classification dataset. Different from Tiny-ImagNet, ImageNet100 is composed of the 100 first classes of ILSVRC-2012. This corresponds to a total of 130,000 224x224 train images and 5,000 224x224 test images.

Losses and projections. In these experiments, we compare the performances of the following losses: SCE, SCE-τ and PGD-loss. Additionally, we compare projections on various L_p hyperspheres with $p \in \{0.5, 1, 2, 3, \infty\}$.

Implementation details For each loss, we train a ResNet18 He et al. (2016) from scratch for 300 epochs with an Adam Kingma and Ba (2015) optimizer, learning rate $1e^{-4}$, and a batch size value of 256. We also use data augmentations. Namely, random horizontal flip, random crop and color jitter. The main results showed in Table 1 have been obtained with the best variance values after conducting a hyper-parameter search. More details can be found in Appendix H.



(a) Accuracy on CIFAR10 with SCE-τ

(b) Accuracy with PGD on CIFAR100/CIFAR10

Figure 2: (a) Accuracy at the end of training on CIFAR10 with a SCE-τ objective for different values of p and variance. (b) Accuracy after training a ResNet18 on CIFAR100 and CIFAR10 with PGD for different values of v.

Appendix B. Additional Experiments

We present additional experiment regarding the dependence to the variance on CIFAR-10 and CIFAR-100 in Figure 2. Similar to the results presented in Table 1, the larger the value of p, the greater the resulting optimal variance is. Plus, SCE-τ performances gain in stability with regard to v for larger values of p.

Appendix C. Connection between MAP and more SCR variants

Another popular practice when tackling classification problems is prototype learning Zhang et al. (2020); Lin et al. (2023); Yang et al. (2018); Ho et al. (2023); Wei et al. (2023); De Lange and Tuytelaars (2021). The main idea is to compare the learned representations to a set of prototypes $\mathcal{P} = \{\boldsymbol{p_1}, \cdots, \boldsymbol{p_L}\}$. The probabilities are computed using a modified version of the softmax, such as detailed in Equation 11.

$$\text{ProtoSoftmax}(\mathbf{z}, P)_i = \frac{e^{\boldsymbol{z} \cdot \boldsymbol{p_i}}}{\sum_{j=1}^{L} e^{\boldsymbol{z} \cdot \boldsymbol{p_j}}} \tag{11}$$

Moreover, several studies introduce an additional class-dependent coefficient in the softmax operator, referred to as softmax with bias or re-weighted softmax: Jodelet et al. (2021); Ren et al. (2020); Legate et al. (2023).

Proposition 3 *Starting from the MAP log-loss defined in Equation 4, under the following assumptions:*

- *The prototypes $\mathcal{P}$ lie on a hypersphere.*
- *The conditional probability density functions $\{f_l(.)\}_{1 \leq l \leq L}$ follow an isotropic gaussian distribution of variance v centered around means $\mathcal{P}$*
- *The variance v of the isotropic Gaussians is equal to one.*

Then, the MAP log-loss is equivalent to the SCE with prototype and bias loss.

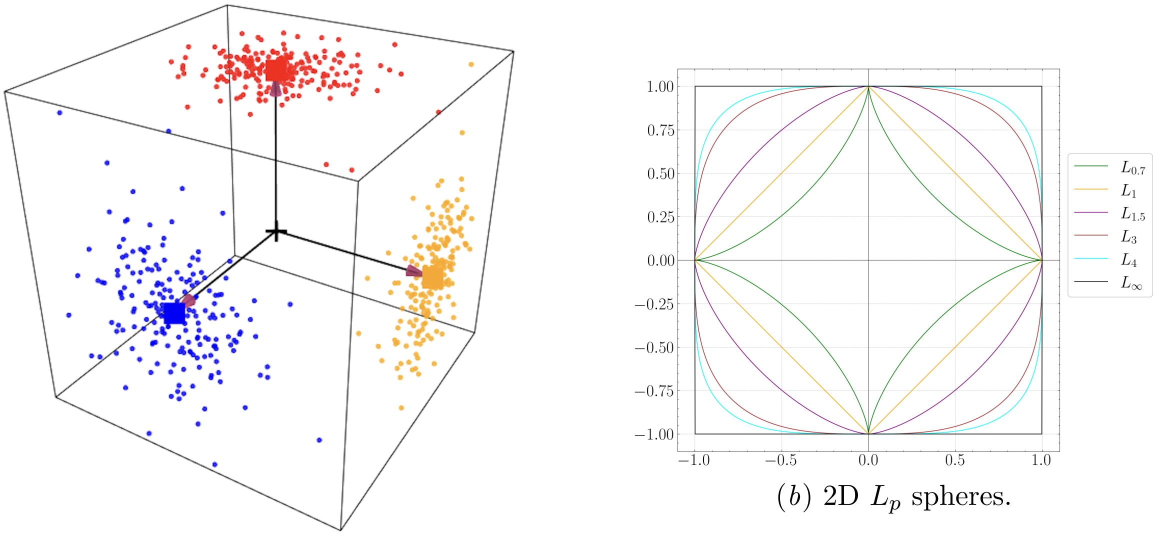
Appendix D. Visualization of L_p norms

We provide visualization of L_p hyperspheres for various values of p as well as visualization of Gaussians projected onto the hypercube in 3D. Such visuzalizations can be found in Figure 3.

Appendix E. Proof of Proposition 1

Starting with Equation 4, the conditional probability distribution of $\mathbf{z}$ given $Y = c$ follows a Gaussian distribution centered around $\boldsymbol{r_c} \in \mathbb{R}^L$, with covariance matrix Σ_c:

$$f_c(\boldsymbol{z}) = (2\pi)^{-L/2} |\Sigma_c|^{-1} e^{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{r_c})^T \Sigma_c^{-1} (\boldsymbol{z}-\boldsymbol{r_c})}, \tag{12}$$



(b) 2D L_p spheres.

(a) 3D Gaussians on the unit cube.

Figure 3: (a) Gaussian-sampled points projected onto a 3D unit cube, with Gaussians centered around the standard basis. (b) Visualization of 2D L_p hyperspheres for various p values.

with T being the superscript for the transpose operator and $|\Sigma_c|$ the determinant of Σ_c. The conditional Gaussian are isotropic if $\Sigma_c = v_c \cdot I$ with I being the identity matrix of size L and v_c the variance for class c. In such situation, $f_c(.)$ becomes

$$f_c(\boldsymbol{z}) = (2\pi v_c)^{-L/2} e^{-\frac{1}{2v_c}||\boldsymbol{z}-\boldsymbol{r_c}||_2^2} \tag{13}$$

Combining Equations (4) and (13) leads to the general form below.

$$\begin{aligned}\mathcal{L}_{Gauss}(\mathcal{B},\theta) &= -\frac{1}{|\mathcal{B}|}\sum_{c=1}^{L}\sum_{i\in I_c}\log\frac{\pi_c\cdot(2\pi v_c)^{-L/2}e^{-\frac{1}{2v_c}||\Phi_\theta(\boldsymbol{x}_i)-\boldsymbol{r_c}||_2^2}}{\sum_{\ell=1}^{L}\pi_l\cdot(2\pi v_l)^{-L/2}e^{-\frac{1}{2v_l}||\Phi_\theta(\boldsymbol{x}_i)-\boldsymbol{r_l}||_2^2}}\\ &= -\frac{1}{|\mathcal{B}|}\sum_{c=1}^{L}\sum_{i\in I_c}\log\frac{\pi_c\cdot v_c^{-L/2}e^{\frac{1}{v_c}\Phi_\theta(\boldsymbol{x}_i)^T\cdot\boldsymbol{r_c}-\frac{1}{2v_c}||\Phi_\theta(\boldsymbol{x}_i)||_2^2-\frac{1}{2v_c}||\boldsymbol{r_c}||_2^2}}{\sum_{\ell=1}^{L}\pi_l\cdot v_l^{-L/2}e^{\frac{1}{v_l}\Phi_\theta(\boldsymbol{x}_i)^T\cdot\boldsymbol{r_l}-\frac{1}{2v_l}||\Phi_\theta(\boldsymbol{x}_i)||_2^2-\frac{1}{2v_l}||\boldsymbol{r_l}||_2^2}}\end{aligned} \tag{14}$$

Now, with equal variances, previous Equation (14) simplifies to:

$$\mathcal{L}_{Gauss}(\mathcal{B},\theta) = -\frac{1}{|\mathcal{B}|}\sum_{c=1}^{L}\sum_{i\in I_c}\log\frac{\pi_c\cdot e^{\frac{1}{v}\Phi_\theta(\boldsymbol{x}_i)^T\cdot\boldsymbol{r_c}}}{\sum_{\ell=1}^{L}\pi_l\cdot e^{\frac{1}{v}\Phi_\theta(\boldsymbol{x}_i)^T\cdot\boldsymbol{r_l}}} \tag{15}$$

The means are assigned to the re-scaled standard basis vectors such that $\boldsymbol{r_c} = r \cdot \boldsymbol{e_c}$ with $e_c = [0, 0, \dots, 1, 0, \dots 0]$ a vector where every component is 0 except the c-th component and $c \in [\![1, L]\!]$. Therefore, the previous equation can be rewritten like in Equation 16 and this ends the proof:

$$\begin{aligned} \mathcal{L}_{Gauss}(\mathcal{B}, \theta) &= -\frac{1}{|\mathcal{B}|} \sum_{c=1}^{L} \sum_{i \in I_c} \log \frac{\pi_c \cdot e^{\frac{r}{v} \Phi_\theta(\boldsymbol{x}_i)^T \cdot \boldsymbol{e_c}}}{\sum\limits_{\ell=1}^{L} \pi_l \cdot e^{\frac{r}{v} \Phi_\theta(\boldsymbol{x}_i)^T \cdot \boldsymbol{e_l}}} \\ &= -\frac{1}{|\mathcal{B}|} \sum_{c=1}^{L} \sum_{i \in I_c} \log \frac{\pi_c \cdot e^{\frac{r}{v} \Phi_\theta(\boldsymbol{x}_i)_c}}{\sum\limits_{\ell=1}^{L} \pi_l \cdot e^{\frac{r}{v} \Phi_\theta(\mathbf{x}_i)_l}} \end{aligned} \tag{16}$$

Appendix F. Proof of Proposition 2

Let $\mathbf{z}$ be a random vector of $\mathbb{R}^d$ with a Gaussian distribution of mean $\boldsymbol{\mu}$ and covariance matrix Σ:

$$f_{\mathbf{z}}(\boldsymbol{z}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\boldsymbol{z} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{z} - \boldsymbol{\mu}) \right) \tag{17}$$

and define

$$\mathbf{u} = \frac{\mathbf{z}}{||\mathbf{z}||_p} = \frac{\mathbf{z}}{||\mathbf{z}||_p} = \frac{\mathbf{z}}{r} \tag{18}$$

the projected vector onto the unit sphere $S_p^d = \{\boldsymbol{x} \in \mathbb{R}^d : ||\boldsymbol{x}||_p = 1\}$. The marginal of $\mathbf{z}$ on S_2^d is called *projected-normal* in Jupp and Mardia (2009).

We present several expressions for the density function $f_{\mathbf{u}}(\boldsymbol{u})$ of the normalized vector $\mathbf{u}$. Building on previous work by Pukkila and Radhakrishna Rao (1988) and extending the result to general cases where $p \neq 2$, we provide a recursively computable integral representation, proving a result which has been stated inSaw (1978) without direct proof. Furthermore, we derive a closed-form expression in terms of a special function. To begin with, we establish a change-of-variable formula for $\mathbf{z} \to (r, \mathbf{u})$, where $\mathbf{u}$ is constrained to live in $\mathcal{S}_p^d$.

Proposition 4 *If $\mathbf{z}$ has a probability density $f_{\mathbf{z}}(\boldsymbol{z})$, with $\boldsymbol{z} \in \mathbb{R}^d$, then the transformation $\mathbf{z} \to (r, \mathbf{u})$, where $\mathbf{u}$ is constrained to live in $\mathcal{S}_p^d$ leads to the density $f_{R,\mathbf{u}}(r, \boldsymbol{u})$:*

$$f_{R,\mathbf{u}}(r, \boldsymbol{u}) = \frac{r^{d-1}}{||\boldsymbol{u}||_{2(p-1)}^{p-1}} f_{\mathbf{z}}(r.\boldsymbol{u}) \tag{19}$$

with respect to d_σ, the element of area of the surface $\mathcal{S}_p^d$.

Proof Let $\xi = \Phi(z_1, \cdots, z_d)$ define a surface element in $\mathbb{R}^d$. A general result in Courant (2011) pages 301-302, states that for any function, we have

$$\int \cdots \int f(z_1, \cdots, z_d) dz_1 \cdots dz_d = \int \cdots \int \frac{f(z_1, \cdots, z_d)}{\sqrt{\Phi_{z_1}^2 + \cdots + \Phi_{z_d}^2}} d_{\sigma_\xi} d_\xi$$

where $\Phi_{z_i} = \frac{\delta\Phi}{\delta z_i}$ and $d_{\sigma_\xi} = \frac{\sqrt{\Phi^2_{z_i}+\cdots+\Phi^2_{z_d}}}{\Phi_{z_d}} dz_1 \cdots dz_{d-1}$ with $\Phi(z_1, \cdots, z_d) = \Sigma_{i=1}^d |z_i|^p = ||\boldsymbol{z}||_p^p$, we have

$$\sqrt{\Phi^2_{z_1} + \cdots + \Phi^2_{z_d}} = \sqrt{\Sigma_{i=1}^d \left(p|z_i|^{p-1} sign(z_i)\right)^2} \tag{20}$$

$$= p\sqrt{||z||_{2(p-1)}^{2(p-1)}} = p||z||_{2(p-1)}^{p-1} \tag{21}$$

with $\xi = r^p$, we have $d\xi = d(r^p) = p\, r^{p-1} dr$.

Now, if we let $\boldsymbol{z} = r\boldsymbol{u}$, it becomes clear that $d_{\sigma_r} = r^{d-1} d_\sigma$, where d_σ is the element of area of $\mathcal{S}_p^d$ and d_{σ_r} is the element of area of the surface $||.||_p = r$. On the other hand, we have $||\boldsymbol{z}||_p^{p-1} = r^{p-1}||\boldsymbol{u}||_p^{p-1}$. Combining these elements, we obtain:

$$f_{\mathbf{z}}(z_1, \cdots, z_d) d_z = f_{R,\mathbf{u}}(r, \boldsymbol{u}) dr d_\sigma = \frac{r^{d-1}}{||\boldsymbol{u}||_{2(p-1)}^{p-1}} f_{\mathbf{z}}(r.\boldsymbol{u}) dr d\sigma \tag{22}$$

which gives the result.

Remark 5 *Observe that with $p = 2$, $||\boldsymbol{u}||_{2(p-1)}^{(p-1)} = ||\boldsymbol{u}||_2^1 = 1$ and $f_{R,\mathbf{u}}(r, \boldsymbol{u}) = r^{d-1} f_{\mathbf{z}}(r\boldsymbol{u})$.*

■

Proposition 6 *The projection of a normal distribution on S_p^d is:*

$$f_{\mathbf{u}}(\boldsymbol{u}) = \frac{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{-\frac{d}{2}}}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}} w} \exp\left(-\frac{1}{2}\lambda^2\right) \int_0^\infty r'^{d-1} \exp\left(-\frac{1}{2}r'^2 + \lambda r'\ \bar{\boldsymbol{u}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}\right) \mathrm{d}r' \tag{23}$$

with $\lambda = (\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}$, $\bar{\boldsymbol{u}} = \frac{\boldsymbol{u}}{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{\frac{1}{2}}}$, $w = ||u||_{2(p-1)}^{(p-1)}$ and $\bar{\boldsymbol{\mu}} = \frac{\boldsymbol{\mu}}{(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}}$

Proof By a direct application, we get the density for a normal distribution:

$$\begin{aligned} f_{R,\mathbf{u}}(r, \boldsymbol{u}) &= \frac{r^{d-1}}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}} w} \exp\left(-\frac{1}{2}(r\boldsymbol{u} - \boldsymbol{\mu})^T\Sigma^{-1}(r\boldsymbol{u} - \boldsymbol{\mu})\right) \\ &= \frac{r^{d-1}}{(2\pi)^{\frac{d}{2}}|\Sigma| w} \exp\left(-\frac{1}{2}\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}\right) \exp\left(-\frac{1}{2}r^2\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u} + r\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{\mu}\right). \end{aligned} \tag{24}$$

with $w = ||\boldsymbol{u}||_{2(p-1)}^{(p-1)}$. The density for $f_{\mathbf{u}}(\boldsymbol{u})$ is obtained by marginalizing $f_{R,\mathbf{u}}(r, \boldsymbol{u})$ over r: $f_{\mathbf{u}}(\boldsymbol{u}) = \int_0^\infty f_{R,\mathbf{u}}(r, \boldsymbol{u}) \mathrm{d}r$. Let $r' = r(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{\frac{1}{2}}$; then

$$f_{\mathbf{u}}(\boldsymbol{u}) = \frac{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{-\frac{d}{2}}}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}} w} \exp\left(-\frac{1}{2}\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}\right) \int_0^\infty r'^{d-1} \exp\left(-\frac{1}{2}r'^2 + r'\frac{\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{\mu}}{\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u}}\right) \mathrm{d}r' \tag{25}$$

Denoting $\lambda = (\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}$, $\bar{\boldsymbol{u}} = \frac{\boldsymbol{u}}{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{\frac{1}{2}}}$ and $\bar{\boldsymbol{\mu}} = \frac{\boldsymbol{\mu}}{(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}}$, which finally gives (25). ■

Remark 7 *With $p = 2, \boldsymbol{\mu} = 0$ and $\Sigma = \sigma^2 1$, which means that x is distributed as a centered isotropic Gaussian, (23) reduces to*

$$f_{\mathbf{u}}(u) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_0^\infty r'^{d-1} \exp\left(-\frac{1}{2}r'^2\right) \mathrm{d}r' = \frac{\Gamma\left(\frac{d}{2}\right)}{2\pi^{\frac{d}{2}}} = \frac{1}{\omega_{d-1}} \tag{26}$$

where we used $\mathbf{u}^T\boldsymbol{u} = 1$ and the known property

$$\int_0^\infty r^{d-1} \exp\left(-\frac{1}{2}r^2\right) \mathrm{d}r = 2^{\frac{d}{2}-1}\Gamma\left(\frac{d}{2}\right). \tag{27}$$

Equation 26 shows that $f_{\mathbf{u}}(u)$ is the uniform distribution on the unit-sphere, where ω_{d-1} is the surface of the unit-sphere.

Starting with (25), we can now state the first result, which is due to Pukkila and Radhakrishna Rao (1988).

Proposition 8 *With $\lambda = (\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}$ and $\alpha = \frac{\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{\mu}}{\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u}}$, the probability density of the normalized Gaussian vector is*

$$f_{\mathbf{u}}(\boldsymbol{u}) = \frac{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{-\frac{d}{2}}}{(2\pi)^{\frac{d}{2}-1}|\Sigma|^{\frac{1}{2}}w} \exp\left(-\frac{1}{2}\left(\lambda^2 - \alpha^2\right)\right) I_d(\alpha) \tag{28}$$

with

$$I_d(\alpha) = \frac{1}{\sqrt{2\pi}} \int_0^\infty r^{d-1} \exp\left(-\frac{1}{2}(r-\alpha)^2\right) \mathrm{d}r \tag{29}$$

and can be computed as

$$I_d(\alpha) = \alpha I_{d-1}(\alpha) + (d-2) I_{d-2}(\alpha),$$

with $I_1 = \Phi(\alpha)$ and $I_2 = \phi(\alpha) + \alpha\Phi(\alpha)$, where $\phi(.)$ and $\Phi(.)$ are respectively the standard normal probability density function and cumulative distribution function.

Proof Completing the square in the argument of the exponential under the integral in (25) gives (28), with the definition of I_d in (29). Integration by part of I_d yields the recurrence equation. Finally, the initial values follow by direct calculation. ■

The main drawback of (28) is that it relies on an integral form, although this integral can be easily evaluated through a recurrence. In contrast, (23) allows us to express the density as a series. We present this result in the general case and recover the result stated in Saw (1978) without proof.

Proposition 9 *With $\lambda = (\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}$, $\bar{\boldsymbol{u}} = \frac{\boldsymbol{u}}{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{\frac{1}{2}}}$, $\bar{\boldsymbol{\mu}} = \frac{\boldsymbol{\mu}}{(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}}$, $w = ||\boldsymbol{u}||_{2(p-1)}^{(p-1)}$ the probability density of the normalized Gaussian vector is*

$$f_{\mathbf{u}}(\boldsymbol{u}) = \frac{\Gamma\left(\frac{d}{2}\right)}{2\pi^{\frac{d}{2}}} \frac{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{-\frac{d}{2}}}{|\Sigma|^{\frac{1}{2}}w} e^{-\frac{1}{2}\lambda^2} \sum_{k=0}^{\infty} \left(\lambda\bar{\boldsymbol{u}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}\right)^k \frac{\Gamma\left(\frac{d+k}{2}\right)}{k!\,\Gamma\left(\frac{d}{2}\right)} \tag{30}$$

Proof In the integral in (23), we can expand the exponential $\exp\left(\lambda r\ \bar{\boldsymbol{u}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}\right)$ in Taylor series, so that

$$\begin{aligned}&\int_0^\infty r^{d-1}\exp\left(-\frac{1}{2}r^2+\lambda r\ \bar{\boldsymbol{u}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}\right)\mathrm{d}r\\ &=\int_0^\infty r^{d-1}\exp\left(-\frac{1}{2}r^2\right)\sum_{k=0}^\infty\frac{1}{k!}\left(\lambda r\ \bar{\boldsymbol{u}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}\right)^k\mathrm{d}r\\ &=\sum_{k=0}^\infty\frac{1}{k!}\left(\lambda\bar{\boldsymbol{u}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}\right)^k\int_0^\infty r^{d-1+k}\exp\left(-\frac{1}{2}r^2\right)\\ &=2^{\frac{d}{2}-1}\sum_{k=0}^\infty\frac{1}{k!}\left(\lambda\bar{\boldsymbol{u}}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}\right)^k\Gamma\left(\frac{d+k}{2}\right)\end{aligned}\tag{31}$$

where the last line follows from the identity (27). Plugging this in (23) and simplifying yield (30). ∎

For $p=2$, we can observe that the first term in (30) is the inverse of the unit-sphere's surface ω_{d-1}. Still for $=2$, in the isotropic case where $\Sigma=\sigma^2\mathbb{1}$, (30) reduces to

$$f_{\mathbf{u}}(\boldsymbol{u})=\frac{\Gamma\left(\frac{d}{2}\right)}{2\pi^{\frac{d}{2}}}e^{-\frac{1}{2}\lambda^2}\sum_{k=0}^\infty\left(\lambda\boldsymbol{u}^T\bar{\boldsymbol{\mu}}\right)^k\frac{\Gamma\left(\frac{d+k}{2}\right)}{k!\,\Gamma\left(\frac{d}{2}\right)}\tag{32}$$

where we used the fact that $\boldsymbol{u}^T\boldsymbol{u}=1$ and where $\bar{\boldsymbol{\mu}}$ is now $\bar{\boldsymbol{\mu}}=\frac{\boldsymbol{\mu}}{(\boldsymbol{\mu}^T\boldsymbol{\mu})^{\frac{1}{2}}}$. This is the formula given in Saw (1978), up to minor notations differences. Finally, for $\boldsymbol{\mu}=0$, (32) reduces to the uniform distribution on the unit-sphere $f_{\mathbf{u}}(\boldsymbol{u})=1/\omega_{d-1}$.

Finally, it is possible to obtain a closed form in terms of a special function.

Proposition 10 *With $\lambda=(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})^{\frac{1}{2}}$ and $\gamma=\frac{\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{\mu}}{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{\frac{1}{2}}}$, the probability density of the normalized Gaussian vector is*

$$f_{\mathbf{u}}(\boldsymbol{u})=\frac{(\boldsymbol{u}^T\Sigma^{-1}\boldsymbol{u})^{-\frac{d}{2}}}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}w}e^{-\frac{1}{2}\lambda^2-\frac{1}{8}\gamma^2}\Gamma(d)D_{-d}\left(\sqrt{2}\gamma\right),\tag{33}$$

where D_{-d} is a Parabolic cylinder function.

Proof A result in the celebrated Tables of integrals, Series and Products of Gradshteyn and Ryzhik states, (Zwillinger et al., 2014, eq. 3.462), that

$$\int_0^\infty x^{\nu-1}e^{-\beta x^2-\gamma x}\mathrm{d}x=(2\beta)^{-\nu/2}\Gamma(\nu)e^{-\frac{\gamma^2}{8\beta}}D_{-\nu}\left(\frac{\gamma}{\sqrt{2\beta}}\right)\text{ for }\beta>0,\nu>0\tag{34}$$

where D_ν is a parabolic cylinder function, (Zwillinger et al., 2014, eq. 9.240). We see that the integral in (23) has precisely this form, with $\nu=d$, $\beta=1/2$, and $\gamma=\lambda\bar{u}^T\Sigma^{-1}\bar{\boldsymbol{\mu}}$. Plugging this in (23) and rearranging yield (33). ∎

Corollary 11 *Let $p, d \in \mathbb{N}^{+\star}$. For $\boldsymbol{z} \in \mathbb{R}^d$ following a d-variate Gaussian of mean $\boldsymbol{\mu} \in \mathcal{S}_p^d$ and covariance matrix $\Sigma = \sigma^2 I$, the distribution of $\boldsymbol{u}$, the projection of $\boldsymbol{z}$ on $\mathcal{S}_p^d$ such that $\boldsymbol{u} = T_{l_p}(\boldsymbol{z})$ is defined by:*

$$g_{\kappa}^{PGD}(\boldsymbol{u}, \boldsymbol{\mu_c}) = a_{\kappa} e^{-\frac{1}{2}\kappa^2} \sum_{n=0}^{\infty} \frac{(\kappa \frac{\boldsymbol{u}^T \cdot \boldsymbol{\mu}}{||\boldsymbol{u}||_2 \cdot ||\boldsymbol{\mu}||_2})^n \, \Gamma\left(\frac{d}{2} + \frac{n}{2}\right)}{n! \; \Gamma\left(\frac{d}{2}\right)} \tag{35}$$

with $\kappa^2 = \frac{||\boldsymbol{\mu}||_2}{\sigma^2}$ and a_{κ} a normalization factor.

Proof Starting from (30) leads to (35) with $a_{\kappa} = \frac{\Gamma\left(\frac{d}{2}\right)\left(\boldsymbol{u}^T \boldsymbol{u}\right)^{-\frac{d}{2}}}{2\pi^{\frac{d}{2}} w}$ ■

Appendix G. Proof of Proposition 3

Trivial starting from Equation (15) and replacing r_c by p_c.

Appendix H. Hyper-parameter search

We conducted a small hyper-parameter for the optimizer and v to obtain the results presented in Table 1. The values tested are presented in Table 2.

H.1. Hardware and computation

For the compared methods, we trained on RTX A5000 for 300 epochs. The training time consumption is 4 hours for CIFAR10 and CIFAR100 and 60 hours for ImageNet100.

Loss	Parameter	Values
CIFAR10		
SCE	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
SCE-τ	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0, 0.5, 1, 1.5, 2, 2.1, 2.2, $\cdots$, 3, 4]
SCE-τ, $p = 0.5$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.01, 0.1, 0.2,$\cdots$, 1.0]
SCE-τ, $p = 1$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.05, 0.1, 0.15, $\cdots$, 0.95, 1]
SCE-τ, $p = 1.5$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.05, 0.1, 0.15, $\cdots$, 0.95, 1]
SCE-τ, $p = 2$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.05, 0.1, 0.15, $\cdots$, 0.95, 1]
SCE-τ, $p = 3$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.05, 0.1, 0.15, $\cdots$, 0.95, 1]
SCE-τ, $p = \infty$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.05, 0.1, 0.15, $\cdots$, 0.95, 1]
CIFAR100		
SCE	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
SCE-τ	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0, 0.5, 1, 1.5, 2, 2.1, 2.2 $\cdots$, 3, 4]
SCE-τ, $p = 0.5$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[$1e^{-5}$, $2e^{-5}$, $\cdots$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, 0.1, 0.2, $\cdots$, 1.0]
SCE-τ, $p = 1$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.001, 0.002, $\cdots$, 0.01, 0.02, $\cdots$, 0.1, 0.2, $\cdots$, 1]
SCE-τ, $p = 1.5$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.005, 0.01, $\cdots$, 0.1, 0.2, $\cdots$, 0.1, 0.2, 1]
SCE-τ, $p = 2$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.01, 0.02, $\cdots$, 0.05, 0.1, 0.15, $\cdots$, 0.95, 1]
SCE-τ, $p = 3$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.01, 0.02, $\cdots$, 0.03 0.1, 0.2, $\cdots$, 1]
SCE-τ, $p = \infty$	optim	[SGD, Adam]
	lr	[0.0001, 0.001, 0.01, 0.1]
	v	[0.05, 0.1, 0.15, 0.16, $\cdots$, 0.3, 0.4, $\cdots$, 1]
ImageNet100		
SCE	optim	[Adam]
	lr	[0.0001]
SCE-τ	optim	[Adam]
	lr	[0.0001]
	v	[2.7]
SCE-τ, $p = 0.5$	optim	[Adam]
	lr	[0.0001]
	v	[$1e^{-5}$, $2e^{-5}$, $\cdots$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, 0.1, 0.2, $\cdots$, 1.0]
SCE-τ, $p = 1$	optim	[Adam]
	lr	[0.0001]
	v	[0.007]
SCE-τ, $p = 1.5$	optim	[Adam]
	lr	[0.0001]
	v	[0.02, 0.025,0.030, 0.035]
SCE-τ, $p = 2$	optim	[Adam]
	lr	[0.0001]
	v	[0.05]
SCE-τ, $p = 3$	optim	[Adam]
	lr	[0.0001]
	v	[0.09]
SCE-τ, $p = \infty$	optim	[Adam]
	lr	[0.0001]
	v	[0.12, 0.19, 0.2, 0.21, 0.22, 0.23]

Table 2: Hyper-parameters for every method on CIFAR10, CIFAR100 and ImageNet100