Improving Self-Supervised Contrastive Learning with Additional Distance Metric

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Self-supervised learning (SSL) has overcome the barrier of labelled supervision by learning representations contrastively or using clustering approaches or with redundancy reduction mechanisms and not limiting to distillation approaches. In the SSL framework, the major contributors are loss functions, augmentations or memory banks. In the SSL Regime, there is quite less work emphasizing the importance of distance metrics or the similarity function and how it impacts the quality of representations acquired from SSL training protocol.

In this work, we study how an additional Euclidean metric can contribute to the learning of the SSL model. Our experiments suggest that adding an additional Euclidean metric to the contrastive SSL loss function aids in learning better representations and provides improvements in classification and robustness tasks. Also, we have seen some interpretable results out from our SSL loss. Although this work is currently confined to comparing with one of the standard works by Chen *et al.* [1], we believe it has a much broader scope in addressing this problem by approaching it with the theoretical motivation.

1 Introduction

Recently, contrastive self-supervised learning (SSL) has shown closer performance with supervised approaches. The underlying reason for competent performance can be due to appropriate augmentations [2, 3, 4], contrastive loss functions [1, 5], and the use of memory banks[6, 7].

The literature is evident that contrastive representation learning (CRL) has shown its recent success in vast domains [8]. In one of the well-established approaches of contrastive learning, [1], the samples are learned by the deep neural networks with an intuition of increasing the positive pair similarity and decreasing the negative pair similarity simultaneously. Here, the representations acquired by deep neural networks are operated on the unit hypersphere. The loss function aids learning by pushing dissimilar representations away from similar ones. Constricting the final representational space to a unit sphere can provide greater performance for both supervised [9] and unsupervised learning tasks[10]. Finally, maximizing the representational similarity between the positive pairs on the hypersphere significantly affected the learning of the neural networks.

In a contrastive loss function, the similarity distance metric plays a significant role in calculating similarities among the acquired representations. The spherical distance metrics such as *cosine similarity* have shown tremendous performance compared to Euclidean, or Manhattan Distances [11, 12]. Recent works emphasized the influence of temperature scaling parameters and uniformity of embedding space on performance. In comparison, this work aims to provide the importance of distance metrics operating on the hyperspherical manifold.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Hyperspherical Learning It is well established that, Hyperspherical learning provides generalization in pattern recognition as it constricts the representational space to an n-dimensional sphere [13]. The learning of representations on hyperspherical manifold acquired attention as it leveraged performance by proposing various angular margin objective functions for tackling Face Recognition task[14, 15, 16, 17]. Also, some applications of hyperspherical learning can be seen in few-shot recognition [18, 19, 20, 21]. When uniformly sampled distributions are mapped onto the unit hypersphere, the representations tend to be refined [10]. In variational autoencoders, the latent representations acquired on the Hypersphere are better and more stable compared to that of Euclidean space [11, 12]. Hence, the literature shows hyperspherical learning provides better representations and performance than Euclidean space.

Contrastive Learning The contrastive self-supervised learning has provided tremendous throughput with appropriate augmentations [2, 3, 4], contrastive loss functions [1, 5], and the use of memory banks[6, 7]. Some works provided good performance without negative samples[22] and others without the use of projection head [23].

Contrastive Losses Operating on Hypersphere Each component that contributes to the contrastive learning framework is studied extensively. Here, we constrict to objective functions which operate on the Hypersphere. We consider these works are closely related to our work. First, Chen *et al.* [1] have provided a standard framework and have utilized an objective function that operates on hypersphere by scaling the radius ($1/\tau$ times). Wang *et al.* [5] have proposed two salient properties of contrastive losses, which are *alignment* and *uniformity*. Also, Chen *et al.* [24] have factorized the NT-Xent loss [1] into two proportions where one is responsible for the alignment and the other for distribution. This contrastive loss is assessed with various distribution criteria. Recently, Wang *et al.* [25] have provided some substantial analysis by varying temperature scaling parameters for the NT-Xent loss [1] and clearly detailed tolerance-uniformity dilemma.

Our Contributions

1. This work theoretically motivates that the Euclidean and spherical metrics are *equivalent metrics* and share the same topological regime on *hypersphere*. Thus an additional distance metric could provide reliable improvements by not only learning representations by discriminating them across the spherical curvature but also on the plane.
2. The empirical results are competitive in various tasks such as classification and robustness.

2 Method

The section follows by providing an introduction to the contrastive learning framework. Then, the theoretical motivation to operate the Euclidean distance metric on the hyperspherical manifold and provided. Finally, we provide our loss functions and perspectives to analyze the significance of proposed loss function.

2.1 Contrastive Framework

We chose the work by Chen et al. [1] for the contrastive representational framework. A detailed description of this framework is provided in the Appendix B. The loss function of this framework is,

$$\mathcal{L}_{NT-Xent} = -\log\left(\frac{e^{\tilde{u}_i^T \tilde{v}_i/\tau}}{\sum_{j=1}^{2N} \mathbb{1}_{[i\neq j]} e^{\tilde{u}_i^T \tilde{v}_j/\tau}}\right) \tag{1}$$

2.2 Additional Euclidean Distance Metrics for Contrastive SSL

As the existing loss $\mathcal{L}_{NT-XNT}$ operates on a hypersphere with the spherical similarity metric (refer Appendix B). But, in order to embed the Euclidean distance in the existing loss function, the Euclidean and spherical metrics should be topologically equivalent i.e. they have to share the same metric topology on Hypersphere. As our Theorem 1 guarantees the topological equivalence we embed this Euclidean metric directly into contrastive loss function $\mathcal{L}_{NT-XNT}$.

Theorem 1. *The metric topology of $\mathbb{S}^n$ determined by the Euclidean distance metric d_{euclid} is equivalent to metric topology of $\mathbb{S}^n$ determined by the spherical distance metric d_{sphere} (Proof is detailed in the appendix).*

As the Euclidean distance and spherical distances both can operate on the same metric topological space (unit hypersphere) we embed these metrics in the contrastive loss function to understand their behaviour when operated simultaneously on Euclidean and spherical Metrics. Thus we term them double metrics (DM) and use this throughout the study.

$$\mathcal{L}_{DM\,ij} = -\alpha \log\left(\frac{e^{\tilde{u}_i^T\tilde{v}_i/\tau}}{\sum_{j=1}^{2N}\mathbb{1}_{[i\neq j]}e^{\tilde{u}_i^T\tilde{v}_j/\tau}}\right) - \beta \log\left(\frac{e^{|\tilde{u}_i-\tilde{v}_i|_2}}{\sum_{j=1}^{2N}\mathbb{1}_{[i\neq j]}e^{|\tilde{u}_i-\tilde{v}_j|_2}}\right) \tag{2}$$

The α and β parameters are weighting functions (hyperparameters) that are to be tuned for optimal loss landscape. In this work, we evaluate four different settings for α and β parameters and provide our detailed implementations for these choices of parameters which are detailed in Table 1.

Loss	Parameters
$\mathcal{L}_{DM_1}$	$\alpha = 0.75, \beta = 0.25$
$\mathcal{L}_{DM_2}$	$\alpha = 0.50, \beta = 0.50$
$\mathcal{L}_{DM_3}$	$\alpha = 0.25, \beta = 0.75$
$\mathcal{L}_{DM_4}$	$\alpha = 1.00, \beta = 1.00$

Table 1: Variants of DM Losses

Geometric Intuition Now we comprehend the role of similarity metrics in the contrastive loss function geometrically. For this, let us consider two feature representations acquired from a neural network contrastively as r_{f_1}, and r_{f_2}. These feature representations are l_2-normalised ($\tilde{r}_{f_1}$, and $\tilde{r}_{f_2}$) and now they lie on unit hypersphere. Next, cosine similarity is calculated between $\tilde{r}_{f_1}$, and $\tilde{r}_{f_2}$ and temperature scaling(τ) is applied. The immediate result of τ can be seen as an extension of the radius by a scale of $\frac{1}{\tau}$ i.e. unit hypersphere extends its radius from one to $\frac{1}{\tau}$. Now, this temperature-scaled similarity metric is used in the $\mathcal{L}_{NT-XNT}$ loss[1]. But, in this work we also calculate the Euclidean distance for normalised features $\tilde{r}_{f_1}$, and $\tilde{r}_{f_2}$ and aggregate them with $\mathcal{L}_{NT-XNT}$ loss by appropriate weighting coefficients (α, β). This helps to analyze and discriminate the representations from both the unit-hypersphere and the planar respectively (For example refer to Figure. 2). We are not aware of the exact embedding space of neural networks but are trying to map these representations on a unit-sphere with l_2-norm. So, our intuition helps to discriminate the representation space by the presence of both planar and curvature information with the help of Euclidean and spherical metrics.

3 Experiments

3.1 Performance

As mentioned, to evaluate the performance of DM losses we have utilized standard classification data CIFAR-10, CIFAR-100, and ImageNet-200. From the results illustrated in Table 2, one can infer that without any additional increment in computational expense the DM losses perform superior for most of the scenarios. Specifically, the third version of DM loss $\mathcal{L}_{DM_3}$ has shown greater performance in most of the scenarios.

Loss Function	**Variants**	**Test Accuracy** CIFAR-10	CIFAR-100	ImageNet-200
SimCLR [1]	$\mathcal{L}_{NT-Xent}$	80.87	59.08	44.03
Ours	$\mathcal{L}_{DM_1}$	80.82	**59.63**	<u>44.48</u>
	$\mathcal{L}_{DM_2}$	<u>80.91</u>	58.69	43.98
	$\mathcal{L}_{DM_3}$	**81.85**	<u>59.22</u>	**44.58**
	$\mathcal{L}_{DM_4}$	80.38	58.36	44.30

Table 2: The table below provides the empirical performance of the individual objective functions for standard classification data.

[1]This increment in radius will extend the representational space and thus the samples will have the more cross-sectional volume to occupy.

3.2 Robustness

The Neural Networks are tested for their robustness for safety-critical applications; thus, we assess whether the proposed models are robust. In this work, we assess our models by considering two types of robustness, i.e., Corruptions, Distributional Shifts, and Data Biases. The significance of each robustness task and their evaluation strategy for our study are detailed in Appendix D.

Corruptions To evaluate the robustness to corruptions, we consider the ImageNet-C dataset [27]. From results illustrated in Table 3, with standard augmentations $\mathcal{L}_{DM_3}$ has less error compared to SimCLR. Also, when AugMix is used, $\mathcal{L}_{DM_3}$ has very low mCE, but almost all the losses have similar rel. mCE[2]. Cumulating these results, we say $\mathcal{L}_{DM_3}$ is robust to corruption.

Loss	Augmentation	mCE (%)	rel. mCE (%)
$\mathcal{L}_{NT-Xent}$	Standard	100	100
$\mathcal{L}_{DM_1}$		100.02	100.04
$\mathcal{L}_{DM_2}$		100.21	100.29
$\mathcal{L}_{DM_3}$		**99.73**	**99.53**
$\mathcal{L}_{NT-Xent}$	AugMix [26]	96.93	86.14
$\mathcal{L}_{DM_1}$		96.66	87.19
$\mathcal{L}_{DM_2}$		96.71	**86.12**
$\mathcal{L}_{DM_3}$		**96.13**	86.15

Table 3: Robustness assessment for corruptions.

Biases To assess the robustness of models to biases we consider two synthetic datasets Colored MNIST, Corrupted CIFAR and one real-world dataset– Biased FFHQ [28].As self-supervised contrastive learning does not rely on labels, it is crucial to understand the representations acquired from biased data.So from Table 4 it can be seen that DM contrastive loss outperforms every biased dataset. Also, $\mathcal{L}_{DM_3}$ has provided significant performance in most of the scenarios. Also with our analysis, we say that DM losses provide better performance with *conflicting* samples.

4 Future Directions and Conclusion

These DM contrastive losses were interpreted from their underlying geometrical significance but, have shown their leveraging performance on standard image classification data, biased data and data with corruptions. The key contribution of additional Euclidean Metric to the loss function is that they do not need any additional computational resource and provides better performance under various scenarios. The experiments prove that DM losses do not fluctuate in their performance with altering temperature (refer Table 5) and they provide a significant performance of standard classification and robustness tasks.

Dataset	Ratio (%)	$\mathcal{L}_{NT-Xent}$ [1]	**Ours**			
			$\mathcal{L}_{DM_1}$	$\mathcal{L}_{DM_2}$	$\mathcal{L}_{DM_3}$	$\mathcal{L}_{DM_4}$
Colored MNIST	0.5	87.35	86.03	86.38	85.14	**87.91**
	1.0	90.36	90.49	90.33	**91.05**	90.71
	2.0	92.83	92.48	92.81	**92.84**	92.04
	5.0	94.81	94.42	95.09	**95.14**	95.05
Corrupted CIFAR	0.5	25.50	26.04	25.55	25.70	**26.31**
	1.0	28.58	28.47	28.32	28.51	**28.98**
	2.0	33.33	32.56	33.71	**33.71**	33.38
	5.0	40.39	40.66	39.44	**41.09**	40.07

Table 4: Robustness assessment for data biases.

There are a couple of limitations which we tend to address in future studies. First, these loss functions are restricted to contrastive-based approaches and thus can only work for certain set of loss functions [1, 5, 24, 29]. In the future, we are willing to analyse the impact of various distance metrics on SSL framework and can be applied to various methods [30, 31, 32] to attain a unified perspective. It should be noted that DM losses are sensitive to α, β as they control the distance metrics operating on the hypersphere. To have significant performance, a lower weight is given for loss operating on spherical distance and a higher weight for the loss operating with Euclidean distance ($\mathcal{L}_{DM_3}$). Thus, a better hyperparameter refinement is needed to reach the optima in the loss landscape.

[2]The corruption error for each corruption sub-category is tabulated in the Appendix Section

References

[1] 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. PMLR, 2020.

[2] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[3] Jiangmeng Li, Wenwen Qiang, Changwen Zheng, Bing Su, and Hui Xiong. Metaug: Contrastive learning via meta feature augmentation. 2022.

[4] Junbo Zhang and Kaisheng Ma. Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16650–16659, 2022.

[5] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[8] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[9] 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.

[10] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pages 517–526. PMLR, 2017.

[11] Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[12] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

[13] Paul W Cooper. The hypersphere in pattern recognition. *Information and control*, 5(4):324–346, 1962.

[14] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[15] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[16] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[17] Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh. Sphereface2: Binary classification is all you need for deep face recognition. *ICLR*, 2022.

[18] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[19] Pascal Mettes, Elise van der Pol, and Cees Snoek. Hyperspherical prototype networks. *Advances in neural information processing systems*, 32, 2019.

[20] Weiyang Liu, Zhen Liu, James M Rehg, and Le Song. Neural similarity learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[21] Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. Learning with hyperspherical uniformity. In *International Conference On Artificial Intelligence and Statistics*, pages 1180–1188. PMLR, 2021.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[23] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *ICLR*, 2022.

[24] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 34:11834–11845, 2021.

[25] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[26] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[27] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[28] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021.

[29] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *European Conference on Computer Vision*, pages 668–684. Springer, 2022.

[30] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[31] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[32] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[33] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[34] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[35] Victor Bryant. *Metric spaces: iteration and application*. Cambridge University Press, 1985.

[36] Somaskandan Kumaresan. *Topology of metric spaces*. Alpha Science Int'l Ltd., 2005.

[37] John G Ratcliffe, S Axler, and KA Ribet. *Foundations of hyperbolic manifolds*, volume 149. Springer, 1994.

Part

Appendix

Table of Contents

A Broader Impact **7**

B Contrastive Learning Framework **7**

C Ablation Study **8**

C.1 Experimental Setup . . . 8

C.2 Temperature Alteration . . . 9

C.3 Hyperspherical Distribution . . . 10

D Robustness Results **10**

E Extended Discussion **11**

F Theoretical Study and Prerequisites **12**

G Gradient Analysis of losses **15**

A Broader Impact

A simple geometric distance function can enhance the performance for the considered downstream tasks. The scope for DM losses is also into upstream tasks applications but not limited to image denoising, segmentation, reconstruction and generation.

Our method does not extensively increase computational resources to excel in the performance but, provides a simple geometric trick and improves baselines. Also, the authors have firmly decided to contribute to *safe AI* and thus strive to reduce AI biases. The self-supervised learning methodically relies on the intrinsic characteristics of the given data. Hence, to provide safe AI to the community, the authors ensure that the model is robust to some of the safety-critical aspects. A stronger motivation arises when Deep Learning continues to be applied in various technologies and social domains.

B Contrastive Learning Framework

As it is clearly evident that augmentations are one of the key contributors to contrastive learning, we augment the data into two views using various augmentation techniques such as Gaussian blur, random resize crop and color jitters, etc. So, for a given N mini-batch of samples, we generate $2N$ samples, and of these, $2N-2$ samples are considered negative samples. Hence, we have $N-1$ negative pairs to feed the neural network for one positive pair.

The pair of samples (both positive and negative) are fed to the standard neural network (encoder) ResNet50 to acquire the refined representations. The feature representations acquired by the ResNet50 are vectors in $\mathbb{R}^{2048}$ space. The features in $\mathbb{R}^{2048}$ space are represented as f_{u_i}, f_{v_i} and where, $i \in \{1, 2, ..., N\}$. As $\mathbb{R}^{2048}$ space is computationally expressive to operate, the representations are mapped (projected) to a lower dimensional space of $\mathbb{R}^{128}$ using a 2-layered multilayer perception. These representations are represented as re paired as (u_i, v_i) and where, $i \in \{1, 2, ..., N\}$. The pair (u_i, v_j) is said to be positive pair if $i = j$ and else, is said to be a negative pair.

Finally, the mini-batch of features extracted from the two views is now l_2-normalized and these pairs are represented $(\tilde{u}_i, \tilde{v}_i)$ and where, $i \in \{1, 2, ..., N\}$. After normalization, these features are meant to be on a hypersphere of 128 dimensions i.e. $\mathbb{S}^{127}$. Now, these normalized features are contrastively learned using the following objective function,

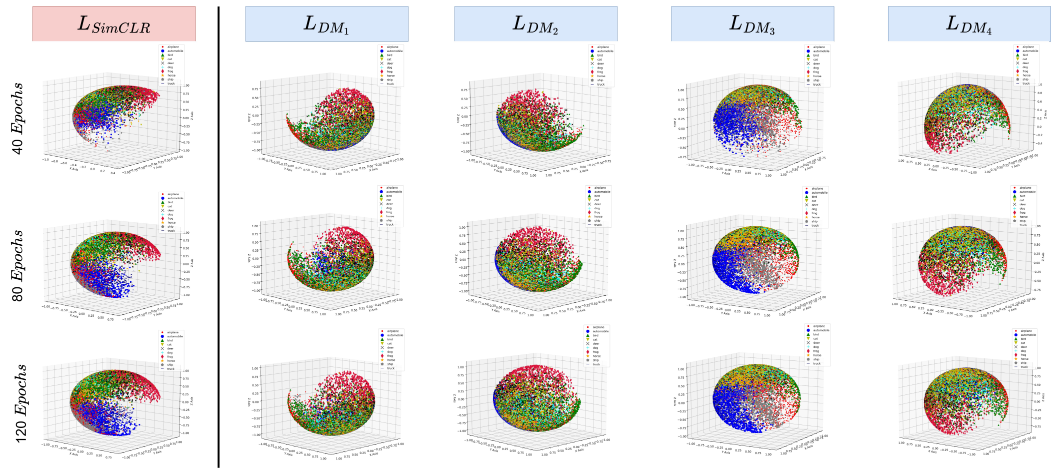

Figure 1: The above figure is a visual description of CIFAR-10 test data mapped to a there dimensional embedding feature vector on 2-sphere, i.e., in $\mathbb{S}^2$. Here it can be observed that samples of SimCLR are not well distributed onto the sphere after training the first 40 epochs but, they spread onto the sphere slowly with an increase in the number of epochs. Whereas, $\mathcal{L}_{DM_3}$ readily occupies the sphere. The results obtained at each of these epochs are detailed in Table 6

$$\mathcal{L}_{NT-Xent} = -\log\left(\frac{e^{\tilde{u}_i^T\tilde{v}_i/\tau}}{\sum_{j=1}^{2N}\mathbb{1}_{[i\neq j]}e^{\tilde{u}_i^T\tilde{v}_j/\tau}}\right) \tag{3}$$

The loss function in eq (3) (Which is same as (1)) is the same as mentioned by Chen *et al.* [1]. Where $\mathbb{1}_{[i\neq j]} \in 0, 1$ is the indicator function which works opposite to that of Kronecker delta i.e. $\mathbb{1}_{[i\neq j]} = 1$ if $[i \neq j]$ and 0 when $[i = j]$.

Linear Evaluation While Linear evaluation, only encoder-learned representations are extracted, and discard the projections. The encoder weights are frozen and the features f_{u_i}, f_{v_i} are now attached to 2 Layered MLP for classification.

Distance Metrics

First, let us consider the well-established spherical and Euclidean distances. Let u, v are the vectors in the Euclidean space of d dimension ($u, v \in \mathbb{R}^d$) and the $\tilde{u}, \tilde{v}$ are the unit vectors in d dimensional hypersphere ($\mathbb{S}^{d-1}$).

$$d_{sphere}(u, v) = \theta(u, v) = \arccos(\tilde{u}^T\tilde{v}) \tag{4}$$

$$d_{euclid}(u, v) = |u - v|_2 = \left(\sum_{i=1}^{d}(u_i - v_i)^2\right)^{\frac{1}{2}} \tag{5}$$

C Ablation Study

C.1 Experimental Setup

Data, Network Architecture, and Parameters For evaluating the performance of DM losses we have utilized standard classification data CIFAR-10, CIFAR-100, and Tiny-ImageNet (ImageNet-200). In the contrastive framework, the augmentations, encoder and projection head, and the hyperparameters for SimCLR and DM losses are kept identical for a fair evaluation. For contrastive training, the augmentations such as random resize crop, random horizontal flip, random grayscale,

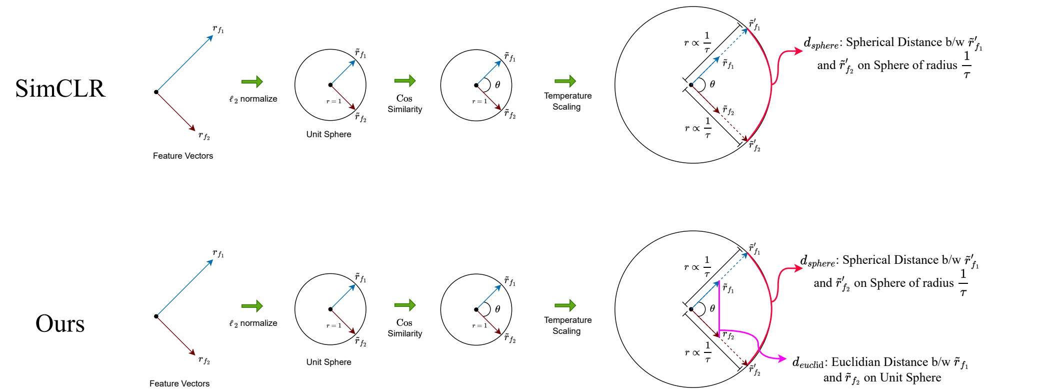

Figure 2: The above figure is a 2-D geometrical intuition of considering additional Euclidean metric. First, we compare SimCLR, and then we differentiate how the proposed method is unique. Till the temperature scaling step, both of them follow the same sequence of steps. A pair of feature vectors (dissimilar) is first l_2-normalised and thus they lie on the unit hypersphere. Next, we calculate the cosine similarity between these two feature vectors. The next successive step is temperature scaling for SimCLR i.e. when we scale the feature vectors with τ then the radius of the sphere is extended $\times\frac{1}{\tau}$ and this can be perceived as *temperature-scaled cosine similarity*. Whereas, we do just rely on the temperature-scaled cosine similarity but calculate the Euclidean distance between the two feature vectors on the unit sphere.

Temp (τ)	$\mathcal{L}_{NT-XNT}$	$\mathcal{L}_{DM_1}$	$\mathcal{L}_{DM_2}$	$\mathcal{L}_{DM_3}$	$\mathcal{L}_{DM_4}$
0.01	<u>78.51</u>	77.14	78.56	**79.32**	77.83
0.05	78.50	78.51	<u>78.74</u>	**79.88**	77.96
0.07	77.96	<u>80.09</u>	**80.33**	79.67	79.73
0.1	<u>81.39</u>	81.32	**82.10**	81.17	80.83
1	78.98	79.12	<u>79.26</u>	**79.42**	77.97

(a) Results on CIFAR-10

Temp (τ)	$\mathcal{L}_{NT-XNT}$	$\mathcal{L}_{DM_1}$	$\mathcal{L}_{DM_2}$	$\mathcal{L}_{DM_3}$	$\mathcal{L}_{DM_4}$
0.01	54.07	53.79	53.85	**55.54**	<u>54.78</u>
0.05	55.24	55.09	<u>54.75</u>	**56.20**	55.09
0.07	56.73	<u>57.23</u>	56.82	**58.17**	55.96
0.1	<u>57.56</u>	57.11	57.42	**58.89**	56.81
1	45.54	42.71	42.71	**46.92**	<u>46.69</u>

(b) Results on CIFAR-100

Table 5: Altering temperature parameter and evaluating the results for the mentioned loss functions on CIFAR-10, 100 Datasets.

and color jitter are applied. The ResNet-50 is used as the encoder for the base encoder and for the projection head, 2-Layered MLP with 2048 to 128 neurons is utilized. LARS optimizer is applied with a learning rate of $0.3\times\frac{\text{batch}}{256}$ applied. For CIFAR-10 and 100 datasets, 1028 samples are trained as a batch. For training ImageNet-200 contrastively we've used a batch size of 256. Also, we've applied linear warmup for the initial 10 epochs and used a cosine scheduler for decaying learning rates without any restarts. If not mentioned specifically, we have used the temperature scaling of $\tau = 0.07$. If not mentioned particularly, the models are trained contrastively for 120 epochs.

After contrastive training, we perform the linear evaluation and for this, we consider augmentations such as random resize crop, random horizontal flip, and normalization. Then freeze the trained encoder which is trained contrastively and attach a 2-Layered MLP (2048 to the number of classes) for classifying the representations using softmax activation. A dropout is used between 2-Layered MLP with a drop rate of 40%. For the linear evaluation, SGD is used to optimize the network with a momentum of 0.9 and with a learning rate of $0.1\times\frac{\text{batch}}{256}$. The learning rate is scheduled at multiple steps i.e. for every 40 epochs by a scale of 0.1. If not mentioned particularly, the models undergo linear evaluation for 90 epochs.

C.2 Temperature Alteration

But, these experiments are performed under optimal temperatures ($\tau = 0.07$). It is necessary to assess the performance for a broad spectrum of temperatures to judge the model's stability. Thus, we fluctuate the temperature τ from 0.01 to 1 and observe the stability of the DM losses. From the results demonstrated in Table 5, it can be understood that even with varying temperatures most of the DM losses have stable learning and again $\mathcal{L}_{DM_3}$ has provided significant performance for both

Loss	ACC @ 40^{th}	ACC @ 80^{th}	ACC @ 120^{th}
$\mathcal{L}_{NT-Xent}$	61.19	68.94	70.98
$\mathcal{L}_{DM_1}$	61.40	68.30	69.85
$\mathcal{L}_{DM_2}$	63.10	69.46	70.97
$\mathcal{L}_{DM_3}$	**63.15**	**70.94**	**72.23**
$\mathcal{L}_{DM_4}$	59.84	67.25	69.32

Table 6: Linear evaluation Accuracy scores for the losses visualised on $\mathbb{S}^2$.

CIFAR-10 and CIFAR-100 datasets respectively. Thus with these evaluations, we infer that DM losses have learning stability even with altering temperatures.

C.3 Hyperspherical Distribution

It should be noted that $\mathcal{L}_{DM_1}$ has comparatively poor performance and does not illustrate the surge of learning. The reason for the superior performance of $\mathcal{L}_{DM_3}$ and substandard performance of $\mathcal{L}_{DM_1}$ can be comprehended by visualizing the sample distribution on the hypersphere.

To understand the behaviour of representations, we provide using hyperspherical Spread (Sample distribution on $\mathbb{S}^2$).

Hyperspherical Spread In the first visualization, the data samples which are fed into a neural network are mapped onto $\mathbb{S}^2$ depicts the visual representations that are distributed on unit 2-sphere i.e. $\mathbb{S}^2$. To visualize these representations on $\mathbb{S}^2$, the contrastive framework is modified accordingly for the CIFAR-10 dataset[3].

First, we use ResNet18 as the base encoder and used 3-Layered MLP ($512 \rightarrow 64 \rightarrow 3$). Hence, while training contrastively we get a pair of feature representations at the terminal layer in 3 dimensions $\mathbb{R}^3$ (Refer Figure **??**. After l_2-normalisation the feature vectors occupy the $\mathbb{S}^2$ space. These feature vectors are directly visualized after training the neural network contrastively for 40, 80, and 120 epochs respectively. Now, in Figure 4 we visualize the test samples which are unseen by the model. One can observe that each of the samples *spreads* on $\mathbb{S}^2$ i.e., the samples have the tendency to occupy the $\mathbb{S}^2$. This gives a vivid picture of DM losses and illustrates that they tend to spread across the $\mathbb{S}^2$ space with the right choice of the parameters α, and β. Specifically, the $\mathcal{L}_{DM_3}$ have a high tendency to spread uniformly across $\mathbb{S}^2$ with increasing epochs.

Rather than just relying on visual facets illustrated in Figure 4, we examine the performance of each loss function for all the mentioned epochs. Uniformly distributed samples on $\mathbb{S}^2$ seek to have greater performance. The linear evaluation accuracy scores are detailed in Table 6 justifies that samples that spread over the hypersphere perform better. This is obliged by integrating a good loss function that provides well-discriminative decision boundaries. As $\mathcal{L}_{DM_3}$ improves the uniformity of distributed samples over the hypersphere with well-discriminative decision boundaries leading to better empirical performance.

D Robustness Results

Corruptions To evaluate the robustness to corruptions, we consider the ImageNet-C dataset [27]. This data has four major categories of corruptions (Noise, Blur, Weather, and Digital), and each category is again divided into sub-categories. Also, each sub-category has five severity levels from 1 to 5 (1 resembles the minimum, and 5 is the maximum severity).

[3]CIFAR-10 is chosen as it would be intuitive to understand the learned representations for 10 classes

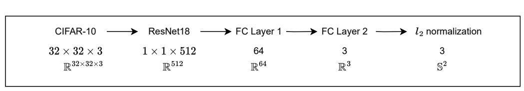

Figure 3: The dimensionality reduction of CIFAR-10 data from $\mathbb{R}^{32\times32\times3} \rightarrow \mathbb{S}^2$ using neural network.

Now, we consider the SimCLR model as the baseline model and evaluate our losses accordingly. Also, we evaluate with two augmentations. The first augmentations are the same as the previous, and the second is AugMix[26]. These augmentations are performed during linear evaluation, and the encoder weights are frozen (The encoder is trained on ImageNet-200). The results are evaluated on the two metrics: mean Corruption Error (mCE) and relative mean Corruption Error (rel. mCE).

Biases The robustness of neural networks to *data biases* when trained on self-supervised contrastive losses is one of the challenges to providing a safe AI [33]. Hence to assess the robustness of models to biases we consider two synthetic datasets Colored MNIST, Corrupted CIFAR and one real-world dataset– Biased FFHQ [28]. The diversity ratio in each of these datasets ranges from 0.5% to 5% (Except for Biased FFHQ). Increasing diversity among the samples has proven significant performance and eventually provided better *de-biased* representations.

As self-supervised contrastive learning does not rely on labels, it is crucial to understand the representations acquired from biased data (A comprehensive evaluation of various *alignment* and *conflict* samples are detailed in the appendix). So from Table 4 it can be seen that DM contrastive loss outperforms every biased dataset. Also, $\mathcal{L}_{DM_3}$ has provided significant performance in most of the scenarios. Also with our analysis, we say that DM losses provide better performance with *conflicting* samples.

From these results, neural networks trained on DM contrastive losses provide incremental robustness to Corruptions and Data Biases. Hence we justify that, adding an additional euclidean distance metric (operating on $\mathbb{S}^{d-1}$) can provide finer performance not just on standard image recognition, but also enhance the robustness of the model.

E Extended Discussion

Gutmann *et al.* [34] has proposed an objective function that learns the distribution of data (in the absence of labels) by discriminating the data distribution with artificially generated noise. This work motivated to develop many objective (loss) functions that are relatively on par with supervised models.

The self-supervised contrastive learning has been viewed from many perspectives, and each of these perspectives has an intuitive conception to understand the representations. Wang *et al.* [5] has proposed two properties of contrastive loss, which are *alignment* and *uniformity*. Likewise, considering DM losses, they are embedded with alignment parameter, i.e., alignment is obtained by factorizing the equation (4).

$$\mathcal{L}_{DM\,ij} = \underbrace{-(\alpha(\tilde{u}_i^T\tilde{v}_i/\tau) + \beta|\tilde{u}_i - \tilde{v}_i|_2)}_{\text{weighted alignment}} + \alpha\log\left(\sum_{j=1}^{2N}\mathbb{1}_{[i\neq j]}e^{\tilde{u}_i^T\tilde{v}_j/\tau}\right) + \beta\log\left(\sum_{j=1}^{2N}\mathbb{1}_{[i\neq j]}e^{|\tilde{u}_i - \tilde{v}_j|_2}\right) \tag{6}$$

Specifically, in equation (5), it is clear that DM losses justify the alignment property of contrastive losses. Specifically, the term *weighted alignment* has distinct parameter values i.e., α, β. Although the weight α is operating on spherical distance (i.e. -$\alpha(\tilde{u}_i^T\tilde{v}_i/\tau)$), according to Lemma **1** those distances satisfy *isometry* in $\mathbb{S}^{d-1}$. Hence, the weighted alignment property is a key contributor to the performance.

Now, other parts of the loss function, other than the weighted alignment, can be closely related to *distribution* property [24]. Hence, we believe that hyperspherical *spread* can be a decisive component to not just provide better performance but also better robustness. As we do not attempt to guarantee the theoretical formulations [21] but rather envision an intuition from the vivid visualizations. Hence, the work by Chen *et al.,* [24] is also considered to be closely related to our work.

The temperature scaling parameter, τ is another important factor in the contrastive loss that proved to have tremendous performance [1]. This τ parameter should be chosen appropriately to determine

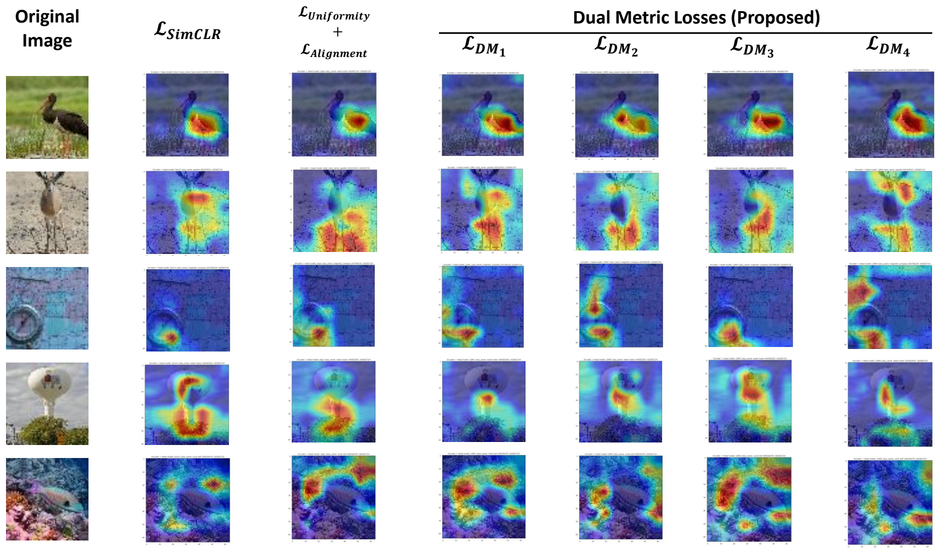

Figure 4: This figure describes the GradCAM Visualizations produced for the test samples of ImageNet200 for 5 distinct classes chosen at random. One can observe that, the class activation maps provided by $\mathcal{L}_{DM_3}$ are quite interpretable compared to the others. Thus, compared to [1] and [5] our methods provide better visually interpretable class activation maps.

the optimal performance. But, Wang *et al.,* [25] conducted a study that details the importance of temperature parameters. First, by providing the theoretical analysis at both extremities τ (i.e. $\tau \in (0, +\infty)$. Second, the authors have experimentally proved that an effective temperature parameter would be $\tau = 0.3$. Also, they contributed a *tolerance* property which gives an intuition to choose the apt temperature. Hence we conduct experiments with this optimal temperature of 0.3 and assess whether the proposed loss can still sustain random temperature fluctuations.

Now, Table 7 compares the above-mentioned loss functions with the DM losses. All the evaluations are fairly evaluated without any alterations in the temperature parameter (τ), augmentations and also other influencing hyperparameters. The parameters related to the loss functions are considered and evaluated at their optimal setting except for the parameters that are chosen optimal as per authors' claims. For instance, Wang *et al.* [25] has illustrated clearly that, the loss function performs optimally at $\tau = 0.3$ and we considered it accordingly Also, the weights of $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ are chosen according to their optimal performance.

So in most cases, $\mathcal{L}_{DM_3}$ competes and provides significant performance. From the results obtained from Tables 2, 5b, 6, 3, 4, 7 the $\mathcal{L}_{DM_3}$ loss do not compromise on performance and robustness at any level. Also, the weighting parameters α, β are notable components as they determine the success rate of DM losses. When there is a higher weight for the loss function, which operates using spherical distance (eg. $\mathcal{L}_{DM_1}$) then the performance fluctuates, and the samples do not quickly spread onto the hypersphere to distribute themselves. Also, when there are non-homogeneous weights ($\alpha + \beta \neq 1$ and $\alpha, \beta \geq 1$) the DM losses tend to perform poorly[4]. Hence, $\mathcal{L}_{DM_3}$ is an apt contrastive loss for most of the downstream self-supervised tasks.

F Theoretical Study and Prerequisites

Some of the fundamental definitions, of theorems, are adapted from the relevant sources. To have a fair understanding of metric spaces refer [35] and to have a topological perspective of the metric spaces refer [36]. The metrical properties of Euclidean and spherical manifolds can be extracted from the work by Ratcliffe *et al.*[37]. The reader can follow the appendix progressively as sufficient fundamentals for the current work are detailed precisely.

[4]A detailed evaluation of various non-homogeneous partitions of weights is provided in Appendix

Loss Functions	CIFAR-10 Test (%)	CIFAR-100 Test (%)	ImageNet-200 Test (%)
Tongzhou *et al.*,[5]	80.86	55.65	42.57
$\mathcal{L}_{NT-Xent}$ [1]	80.87	59.08	44.03
$\mathcal{L}_{DM_1}$	80.82	**59.63**	<u>44.48</u>
$\mathcal{L}_{DM_2}$	80.91	58.69	43.98
$\mathcal{L}_{DM_3}$	**81.85**	<u>59.22</u>	**44.58**
$\mathcal{L}_{DM_4}$	80.83	58.38	44.30
Wang *et al.*, [25] ($\tau = 0.3$)	81.89	55.01	42.82
$\mathcal{L}_{DM_1}$ ($\tau = 0.3$)	**83.85**	55.09	43.27
$\mathcal{L}_{DM_2}$ ($\tau = 0.3$)	82.79	<u>55.68</u>	<u>43.47</u>
$\mathcal{L}_{DM_3}$ ($\tau = 0.3$)	<u>83.64</u>	**57.09**	**44.09**
$\mathcal{L}_{DM_4}$ ($\tau = 0.3$)	82.02	53.56	42.64

Table 7: Comparison of various loss functions with proposed DM losses. All the evaluations are fairly evaluated without any alterations in the temperature parameter (τ), augmentations, and other influencing hyperparameters. But, the parameters specifically related to the loss functions (significant contributions) are considered and evaluated at their optimal setting. The best-performed model is provided and highlighted with **bold** and the second best is highlighted by <u>underline</u>.

Definition 1 *Suppose a non-empty set $X \notin \phi$ and for each $x_1, x_2 \in X$ let $d(x_1, x_2)$ be a real number,*

1. *Non-degenerate $d(x_1, x_2) = 0$ iff $x_1 = x_2$;*
2. *Non-negative $d(x_1, x_2) \geq 0$;*
3. *Symmetric $d(x_1, x_2) = d(x_2, x_1)$ $\forall x_1, x_2 \in X$.*
4. *Triangle Inequality $d(x_1, x_3) \leq d(x_1, x_2) + d(x_1, x_2)$ $\forall x_1, x_2, x_3 \in X$.*

Then 'd' is said to be a metric (distance) on space X and (X, d) is called a metric space.

Definition 2 *Suppose (X, d) is called a metric space and let $\mathcal{T}_d$ be the collection of subsets of U of X such that, for each $x \in U$ $\exists$ $r > 0$ with a open ball $\mathcal{B}(x; r) \subset U$. Then $(X, \mathcal{T}_d)$ is called the topological space defined under the metric (distance) d.*

Definition 3 *Suppose d_A, d_B be distance metrics on X with topologies $\mathcal{T}_A, \mathcal{T}_B$ respectively . Then $d_A$and d_B* metrics are ***equivalent*** iff they have the same topology i.e. $\mathcal{T}_A \equiv \mathcal{T}_B$.

Proposition 2 *Let the distance metrics d_A, d_B on X are such that for some ϵ we have,*

$$\frac{1}{\epsilon} d_A(x_1, x_2) \leq d_B(x_1, x_2) \leq d_A(x_1, x_2) \tag{7}$$

*Where, $\forall x_1, x_2 \in X$. Then these metrics d_A, d_B are **equivalent metrics**.*

Proof. Let $\mathcal{T}_A, \mathcal{T}_B$ be the topologies defined by metrics d_A, d_B respectively. We must show that a subset of U of $X \in \mathcal{T}_A$ iff it $\in \mathcal{T}_B$.

Let, $U \in \mathcal{T}_A$ and$u \in U$. There exist some $r_1 > 0$ such that $\mathcal{B}_{d_A}(u; r_1) \subset U$ i.e.

$$\{u \in X | d_A(u, v) < r_1\} \subset U.$$

Similarly consider $\mathcal{B}_{d_B}(u; r_2)$ where $r_2 = r_1/\epsilon$. If $v \in \mathcal{B}_{d_B}(u; r_1/\epsilon)$ then $d_B(u, v) < r_1/\epsilon$. But, $\frac{1}{\epsilon} d_A(u, v) \leq d_B(u, v)$ and so, for $v \in \mathcal{B}_{d_B}(u; r_1/\epsilon)$ we have

$$d_A(u, v) \leq d_B(u, v) \leq \epsilon \times \frac{r_1}{\epsilon} = r_1$$

Hence, $v \in \mathcal{B}_{d_A}(u; r_1)$ whenever $v \in \mathcal{B}_{d_B}(u; r_1/\epsilon)$ but,

$$\mathcal{B}_{d_A}(u; r_1) \subset U \text{ and so, } \mathcal{B}_{d_B}(u; r_1/\epsilon) \subset \mathcal{B}_{d_A}(u; r_1) \subset U$$

Thus, for $u \in U$, there exist some $r_2 > 0$ ($r_2 = r_1/\epsilon$ such that, $\mathcal{B}_{d_B}(u; r_2) \subset U$. Thus, U is open in the topology determined by metric d_B i.e. $U \in \mathcal{T}_A$ and $U \in \mathcal{T}_B$. As $U \in \mathcal{T}_B$ for $uinU$ $\exists r_1 > 0$ with $\mathcal{B}_{d_B}(u; r_1) \subset U$. If $d_A(u, v) < r_1/\epsilon$ we have $d_B(u, v) \leq \epsilon d_A(u, v) < \epsilon \times \frac{r_1}{\epsilon}$. So, $\mathcal{B}_{d_A}(u; r_1/\epsilon) \subset \mathcal{B}_{d_B}(u; r_1) \subset U$.

$$\text{Thus, } U \in \mathcal{T}_A \text{ iff } U \in \mathcal{T}_B \text{ and } \therefore\ \mathcal{T}_A \equiv \mathcal{T}_B$$

□

Definition 4 *The spherical distance function d_{sphere} is a metric on hypersphere of d dimensions ($\mathbb{S}^{d-1}$).*

Proof. Let u, v are the vectors in the Euclidean space of d dimension ($u, v \in \mathbb{R}^d$) and the $\tilde{u}, \tilde{v}$ are the unit vectors in d dimensional hypersphere ($\mathbb{S}^{d-1}$).

The spherical distance d_{sphere} is written as,

$$d_{sphere}(u, v) = \theta(u, v) = \arccos(\tilde{u}^T \tilde{v})$$

The spherical metric d_{sphere} is non-negative, non-degenerate, and also symmetric. Now we prove the triangle inequality to justify that, ($\mathbb{S}^{d-1}$,d_{sphere}) forms a metric space. It should be noted that the orthogonal transformations of $\mathbb{R}^d \to \mathbb{S}^{d-1}$ preserves spherical distances. So, we transform u, v, w by an orthogonal transformation and $u, v, w \in \mathbb{R}^d$. Here to prove this inequality let us consider $d = 3$ then we have,

$$\begin{aligned}
\cos(\theta(u, v) + \theta(v, w)) &= \cos\theta(u, v)\cos\theta(v, w) - \sin\theta(u, v)\sin\theta(v, w) \\
&= (u.v)(v.w) - |u \times v||v \times w| \\
&\leq (u.v)(v.w) - (u \times v).(v \times w)
\end{aligned}$$

$$\left[(a \times b).(c \times d) = \begin{vmatrix} a.c & a.d \\ b.c & b.d \end{vmatrix}\right]$$

$$\begin{aligned}
\cos(\theta(u, v) + \theta(v, w)) &\leq (u.v)(v.w) - (u \times v).(v \times w) \\
&= (u.v)(v.w) - (u \times v).(v \times w) \\
&= (u.v)(v.w) - ((u.v)(v.w) - (u.w)(vv)) \\
&= (u.w) \\
&= \cos\theta(u, w)
\end{aligned}$$

Thus we obtain $\theta(u, w) \leq \theta(u, v) + \theta(v, w)$. □

Lemma 1 *A function $f : \mathbb{S}^{d-1} \to \mathbb{S}^{d-1}$ is an* isometry *iff it is an isometric w.r.t d_{euclid} on $\mathbb{S}^{d-1}$ because $|u - v|_2^2 \equiv 2(1 - u.v)$.*

Proof. As, $u, v \in \mathbb{S}^{d-1}$ it should be clear that, $|u| = |v| = 1$.

$$\begin{aligned}
|u - v|_2 &= \sqrt{\sum_{i=1}^{d-1}(u_i - v_i)^2} = \sqrt{\sum_{i=1}^{d-1}(u_i^2 + v_i^2 - 2u_i.v_i)} \\
&= \sqrt{\sum_{i=1}^{d-1} u_i^2 + \sum_{i=1}^{d-1} v_i^2 - 2\sum_{i=1}^{d-1} u_i.v_i)} = \sqrt{1 + 1 - 2 \times (u.v)} = \sqrt{2(1 - u.v)}
\end{aligned}$$

So,

$$|u - v|_2^2 \equiv 2(1 - u.v)$$

□

Theorem 1. *The metric topology of $\mathbb{S}^n$ determined by the Euclidean distance metric d_{euclid} is equivalent to metric topology of $\mathbb{S}^n$ determined by the spherical distance metric d_{sphere} (Proof is detailed in the appendix).*

Proof. Suppose, $u, v \in \mathbb{S}^{n-1}$ and $\theta(u, v)$ is already mentioned in the equation B. Where, $d_{sphere}(u, v) \in [0, \pi]$.

It can be verified that,

$$d_{euclid}(u, v) = |u - v|_2 = 2 \sin \left(\frac{\theta(u, v)}{2} \right).$$

Specifically, $\theta(u, v)$ is a strictly increasing function of the euclidean distance $|u - v|_2$.

It is clear from Theorem 1 (appendix) that $\theta(u, v)$ is i) *non-degenerate* ii) *non-negative* and iii) *symmetric*. In order to prove the fourth postulate i.e. *triangle inequality* let us consider $a, b, c \in \mathbb{S}^2$.

$$\text{If } \theta(a, b) + \theta(b, c) \geq \pi \quad \text{we obtain,}$$

$$d_{sphere}(a, c) \leq \pi \leq d_{sphere}(a, b) + d_{sphere}(b, c).$$

Therefore, assume $d_{sphere}(a, b) + d_{sphere}(b, c) \leq \pi$. Now consider b as the north pole and rotate the axis-c to c^* such that, c^* and a are on opposite meridians.

$$|a - c|_2 \leq |a - c^*|_2$$

$$d_{euclid}(a, c) \leq d_{euclid}(a, c^*)$$

$$\therefore d_{sphere}(a, c) \leq d_{sphere}(a, c^*)$$

$$\begin{aligned} d_{sphere}(a, c^*) &= d_{sphere}(a, b) + d_{sphere}(b, c^*) \\ &= d_{sphere}(a, b) + d_{sphere}(b, c) \end{aligned}$$

So, the metric space $(\mathbb{S}^{n-1}, d_{sphere})$ is complete and

$$\frac{2}{\pi}\alpha \leq \sin \alpha \leq \alpha \quad (\text{Where, } 0 \leq \alpha \leq \frac{\pi}{2})$$

From Definition **3** and Proposition **2**[5] it can be concluded that,

$$d_{euclid}(u, v) \leq d_{sphere}(u, v) \leq \frac{\pi}{2} d_{euclid}(u, v).$$

Hence Proposition 1 implies that Euclidean distance metric and spherical distance metric are equivalent and share the same topological space. □

G Gradient Analysis of losses

The gradients are calculated and analyzed w.r.t positive pairs to comprehend the target distribution similar to Chen *et al.* [1]. First, we calculate the gradients w.r.t +ve samples for $\mathcal{L}_{NT-Xent}$ and then we'll further proceed for DM loss $\mathcal{L}_{DM}$.

[5] Refer Appendix section for detailed proofs.

Gradients w.r.t +ve samples for $\mathcal{L}_{NT-Xent}$

$$\mathcal{L}_{NT-Xent} = u^T v^+/\tau - \log\left(\sum_{v \in \{v^+, v^-\}} \exp(u^T v/\tau)\right)$$

$$\frac{\partial}{\partial u}(\mathcal{L}_{NT-Xent}) = \frac{\partial}{\partial u}(u^T v^+/\tau) - \frac{\partial}{\partial u}\left(\log \sum_{v \in \{v^+, v^-\}} \exp(u^T v/\tau)\right)$$

$$\left[\frac{\partial}{\partial x}(a^T x) = \frac{\partial}{\partial x}(a x^T) = a\right]$$

$$\begin{aligned}\frac{\partial}{\partial u}(\mathcal{L}_{NT-Xent}) &= v^+/\tau - \frac{1}{\sum_{v \in \{v^+, v^-\}} \exp(u^T v/\tau)} \times \frac{\partial}{\partial u}\left(\sum_{v \in \{v^+, v^-\}} \exp(u^T v)\right) \\ &= \frac{v^+}{\tau} - \left(\frac{\frac{\partial}{\partial u}\left(\sum_{v^+} \exp(u^T v^+/\tau) + \sum_{v^-} \exp(u^T v^-/\tau)\right)}{\sum_{v \in \{v^+, v^-\}} \exp(u^T v/\tau)}\right)\end{aligned}$$

$$\left[\text{Suppose, } Z(u, v) = \sum_{v \in \{v^+, v^-\}} \exp(u^T v/\tau)\right]$$

$$\therefore \nabla \mathcal{L}_{NT-Xent} = \left(1 - \frac{\sum_{v^+} \exp(u^T v^+/\tau)}{Z(u, v)}\right) . \frac{v^+}{\tau} - \left(\frac{\sum_{v^-} \exp(u^T v^-/\tau)}{Z(u, v)}\right) . \frac{v^-}{\tau}$$

Here, $Z(u, v)$ can be assumed as the partition function [34] for the contrastive loss.

Gradients w.r.t +ve samples for $\mathcal{L}_{DM}$

$$\mathcal{L}_{DM} = \alpha \mathcal{L}_{NT-Xent} + \beta ||u - v^+||_2 - \beta \log\left(\sum_{v \in \{v^+, v^-\}} \exp(||u - v||_2)\right)$$

$$\frac{\partial}{\partial u}(\mathcal{L}_{DM}) = \alpha \frac{\partial}{\partial u}(\mathcal{L}_{SimCLR}) + \beta \frac{\partial}{\partial u}(||u - v^+||_2) - \beta \frac{\partial}{\partial u}\left(\log \sum_{v \in \{v^+, v^-\}} \exp(||u - v||_2)\right)$$

$$\left[\frac{\partial}{\partial x}(||x - y||) = \frac{x - y}{||x - y||_2}\right]$$

$$= \alpha \nabla \mathcal{L}_{NT-Xent} + \beta \frac{u - v^+}{||u - v||_2} - \beta \frac{1}{\sum_{v \in \{v^+, v^-\}} \exp(||u - v||_2)} \times \frac{\partial}{\partial u}\left(\sum_{v \in \{v^+, v^-\}} \exp(||u - v||_2)\right)$$

$$= \alpha \nabla \mathcal{L}_{NT-Xent} + \beta \frac{u - v^+}{||u - v||_2} - \beta \left(\frac{\frac{\partial}{\partial u} \left(\sum_{v^+} \exp(||u - v^+||_2) + \sum_{v^-} \exp(||u - v^-||_2) \right)}{\sum_{v \in \{v^+, v^-\}} \exp(||u - v||_2)} \right)$$

$$\left[\text{Suppose, } Z_2(u, v) = \sum_{v \in \{v^+, v^-\}} \exp(||u - v||_2) \right]$$

$$= \alpha \nabla \mathcal{L}_{NT-Xent} + \beta \frac{u - v^+}{||u - v||_2} - \beta \left(\frac{\sum_{v^+} \exp(||u - v^+||_2) . \frac{u - v^+}{||u - v^+||_2}}{Z_2(u, v)} + \frac{\sum_{v^-} \exp(||u - v^-||_2) . \frac{u - v^-}{||u - v^-||_2}}{Z_2(u, v)} \right)$$

$$\therefore \nabla \mathcal{L}_{DM} = \alpha \nabla \mathcal{L}_{NT-Xent} + \beta \frac{u - v^+}{||u - v||_2} - \beta \left(\frac{\sum_{v^+} \exp(||u - v^+||_2) . \frac{u - v^+}{||u - v^+||_2}}{Z_2(u, v)} + \frac{\sum_{v^-} \exp(||u - v^-||_2) . \frac{u - v^-}{||u - v^-||_2}}{Z_2(u, v)} \right)$$