arXiv:2308.01140v1 [cs.LG] 2 Aug 2023

DySTreSS: Dynamically Scaled Temperature in Self-Supervised Contrastive Learning

Siladittya Manna
CVPR Unit[1], ISI, Kolkata[2]
siladittya_r@isical.ac.in

Soumitri Chattopadhyay
UNC Chapel Hill[3], USA
soumitri.chattopadhyay@gmail.com

Rakesh Dey
CVPR Unit, ISI, Kolkata
id.rakeshdey@gmail.com

Saumik Bhattacharya
Department of EECE[4], IIT Kharagpur[5]
saumik@ece.iitkgp.ac.in

Umapada Pal
CVPR Unit, ISI, Kolkata
umapada@isical.ac.in

Abstract

In contemporary self-supervised contrastive algorithms like SimCLR, MoCo, etc., the task of balancing attraction between two semantically similar samples and repulsion between two samples from different classes is primarily affected by the presence of hard negative samples. While the InfoNCE loss has been shown to impose penalties based on hardness, the temperature hyper-parameter is the key to regulating the penalties and the trade-off between uniformity and tolerance. In this work, we focus our attention to improve the performance of InfoNCE loss in SSL by studying the effect of temperature hyper-parameter values. We propose a cosine similarity-dependent temperature scaling function to effectively optimize the distribution of the samples in the feature space. We further analyze the uniformity and tolerance metrics to investigate the optimal regions in the cosine similarity space for better optimization. Additionally, we offer a comprehensive examination of the behavior of local and global structures in the feature space throughout the pre-training phase, as the temperature varies. Experimental evidence shows that the proposed framework outperforms or is at par with the contrastive loss-based SSL algorithms. We believe our work (DySTreSS) on temperature scaling in SSL provides a foundation for future research in contrastive learning.

1 Introduction

With its prowess at learning high-quality representations from large-scale unlabeled data, self-supervised learning (SSL) has revolutionized the field of machine learning. Classical approaches in SSL involved designing a suitable pretext task, such as solving jigsaw puzzles [1], image inpainting [2], colorization [3], etc., that could aid representation learning. However, the problem with these methods was that there exists a significant difference between the nature of the pretext task and the desired

[1] Computer Vision and Pattern Recognition Unit
[2] Indian Statistical Institute, Kolkata
[3] University of North Carolina at Chapel Hill, USA
[4] Electronics and Electrical Communication Engineering
[5] Indian Institute of Technology Kharagpur

Preprint. Under review.

downstream task (classification/segmentation). Instead, pieces of more recent works have been on the lines of contrastive learning [4–6] – wherein the model learns an embedding space such that the features of augmented versions of the same sample lie close to each other, pushing the embeddings of other samples and their augmented versions farther apart. Unsupervised contrastive models, aided by heavy augmentations and robust abstraction capabilities are capable of learning certain levels of representational structures. Empirically, contrastive learning-based algorithms have been found to perform better at downstream tasks than the former classical SSL methods.

The most commonly used loss function for self-supervised contrastive learning is the InfoNCE objective, used in several works such as CPC [7], MoCo [4], SimCLR [5], etc. Although the temperature hyper-parameter is an integral part of the InfoNCE function, it has mostly been trivialized in different works as a mere scaling coefficient. Recently, in [8] the authors have proposed a temperature hyper-parameter (τ) scheduling for SSL. However, the method proposed in [8] is mainly focused on task switching based on the temperature hyper-parameter τ rather than focusing on the effect of τ on false negative samples. In this work, however, we argue that there's more to this seemingly redundant factor. Our theoretical analyses bring forth an intuitive yet vital aspect related to the presence of constructively false negative yet inherently positive pairs – samples that do not originate from the same instance yet show a high degree of semantic representational similarity as they belong to the same underlying class. As the main objective of the contrastive loss function is to maximize the similarity of the different augmentations of the same instance while minimizing the same for different instances, the aforementioned constructively false negative pairs are repelled away. This action implies that semantic information is not an integral part of contrastive loss. Pushing the samples in semantically similar pairs away creates an adverse effect on representation learning. Large penalties on these samples along with true negative samples may increase the uniformity but it adversely affects the alignment of the local structure constituted by samples with similar semantic information. Hence, arises the uniformity-tolerance (alignment) dilemma as addressed in [9]. In this work, we intend to utilize the temperature hyper-parameter to effectively modulate the repelling effect in these false negative pairs without disrupting the local and global structures of the feature space to improve representation learning.

The primary contributions of the proposed DySTreSS can be summarized as:

- We systematically study the role of temperature hyper-parameter and its effect on local and global structure in the feature space during optimization of the InfoNCE loss, both intuitively and theoretically, to establish the motivation for our proposed method.
- We introduce the philosophy behind the source of the uniformity-tolerance dilemma in InfoNCE loss-based contrastive learning and the effect of temperature variation on the same.
- With the established groundwork, we propose a temperature-scaled (DySTreSS) contrastive learning framework that dynamically modulates the temperature hyper-parameter.
- We show the effectiveness of our approach by conducting experimentation across several benchmark vision datasets, with empirical results showing that our method performs at par or better than the state-of-the-art SSL algorithms in the literature.

The rest of the paper is organized as follows. Sec. 2 briefs contemporary works in self-supervised learning, along with an account of the works where the temperature hyper-parameter is the focus point. Sec. 3 gives a detailed account of the theoretical background, leading to our motivation for the proposed framework. Sec. 4 introduces the proposed framework discussing the motivation behind the proposed temperature scaling function and analysing the same. Next, we present the implementation details in Sec. 5. The experimental evidence along with various ablation studies are presented with illustrations in Sec. 6. Finally, we conclude our work in Sec. 7.

2 Related Work

Self-supervised Learning. SSL approaches [5, 6, 10–16] have become the de facto standard in unsupervised representation learning with the aim to learn powerful features from unlabelled data that can be effectively transferred to downstream tasks. Several pre-training strategies have been proposed, which can be categorized as generative or reconstruction-based [3, 17–20], clustering-based [21–24], and contrastive learning approaches [4–6, 14, 15, 25]. Other popular methods include similarity

learning [16], redundancy reduction within embeddings [12], and an information maximization-based algorithm [11].

Contrastive Learning. The majority of recent SSL algorithms have leveraged contrastive learning, a powerful feature learning paradigm that causes distorted versions of the same sample to attract and different samples to repel. SimCLR [5] simply used a contrastive loss function for representation learning, while MoCo [4] leveraged momentum encoding and a dictionary-based feature bank for negative samples. These works were further enhanced in [14] and [15] respectively. More recent additions to this literature include DCL [6] where the authors decoupled the positive and negative pairing components of the InfoNCE [26] loss function. Furthermore, the work by [27] investigated several intriguing properties of contrastive learning.

Temperature in Contrastive Learning. Recently, there have been a few works that have focused on the temperature hyper-parameter in the InfoNCE loss function in Contrastive Learning. In [28], the authors present a temperature hyper-parameter as a function of the input representations thereby incorporating uncertainty in the form of temperature. [9] explores the hardness-aware property of contrastive loss and the role of temperature in it by measuring the uniformity and tolerance of representations. On the other hand, MACL [29] assumed the temperature hyperparameter as the function of alignment to address the uniformity-tolerance dilemma existing in the InfoNCE loss design. Motivated by the study shown in [9], the authors in [8] proposed a continuous task switching between instance discrimination and group-wise discrimination by using simple cosine scheduling.

3 Theoretical Background

3.1 InfoNCE Loss

In self-supervised contrastive learning [4, 5, 7, 14, 15, 30, 31], the probability of any pair of samples being predicted as a positive pair is given by $p_{ij} = \frac{exp(\frac{s_{ij}}{\tau})}{\sum_k exp(\frac{s_{ik}}{\tau})}$, where s_{ij} is the cosine similarity between the latent vectors of the samples x_i and x_j. The probabilities p_{ij} follow a Boltzmann distribution. Consequently, the InfoNCE loss used in contrastive learning is given by Eqn. 1.

$$\mathcal{L} = \sum_i \mathcal{L}_i = -\sum_i ln\left(p_{ii+}\right) = -\sum_i ln\left(\frac{exp(\frac{s_{ii}}{\tau})}{\sum_j exp(\frac{s_{ij}}{\tau})}\right) \tag{1}$$

3.2 Nomenclature of Sample Pairs

In self-supervised contrastive learning frameworks like SimCLR [5], MoCo [4], etc., we assume that each sample is a class on its own. This results in the pairing of any two samples that may belong to the same class, resulting in the formation of false negative pairs. The similarity between these samples in these types of pairs can assume high cosine similarity values. On the contrary, pairs consisting of two samples belonging to two different classes comprise true negative pairs. However, depending on the mapping of the corresponding features to the feature space, true negative pairs can also have high cosine similarity between the constituent samples, and are called hard negative pairs. False negative pairs by construction can also act as hard negative pairs.

3.3 Role of Temperature in Contrastive Learning

InfoNCE loss concentrates on hard negative optimization by penalizing the hard negative pairs according to their hardness [9]. The gradient of $\mathcal{L}_i$ w.r.t. s_{ii} and s_{ij} is given by Eqn. 2 and 3. A simple relative penalty term is defined in [9] and as given in Eqn. 4. This penalty term also follows the Boltzmann distribution.

$$\frac{\partial \mathcal{L}_i}{\partial s_{ii}} = -\frac{1}{\tau}\sum_{k\neq i} p_{ik} \tag{2}$$

$$\frac{\partial \mathcal{L}_i}{\partial s_{ij}} = \frac{1}{\tau} p_{ij} \tag{3}$$

$$r(s_{ij}) = \frac{|\frac{\partial \mathcal{L}_i}{\partial s_{ij}}|}{|\frac{\partial \mathcal{L}_i}{\partial s_{ii}}|} = \frac{\exp(\frac{s_{ij}}{\tau})}{\sum_{k\neq i}\exp(\frac{s_{ik}}{\tau})} \tag{4}$$

The role of temperature in contrastive loss is to control the penalty for the hard negative samples. As a result, low temperature values tend to penalize more without semantic similarity awareness and

create a more uniformly distributed feature space. It is evident that the temperature hyper-parameter acts as the control knob for the uniformity of the samples in the feature space.

3.4 Effect of Temperature on Local and Global Structures

In an ideal scenario, every sample of any particular class would converge to a closed set of points in the feature space. However, ideal convergence is not achieved in any self-supervised pre-training. In such a scenario, the term local structure of any sample refers to the arrangement of the other samples in the close local neighbourhood of that sample and can be denoted by the samples included in a ball of radius r_j around sample x_j. Likewise, the term global structure takes the arrangement of all the samples in the feature space into account. Ideally, the global structure should consist of N closed sets, imitating a multi-modal Gaussian distribution, each Gaussian component having low variance. However, the ideal global structure diverges towards a uniform distribution. Although the uniformity measure indicates how close the distribution of the samples is to a uniform distribution on an instance level, it is also capable of indicating the closeness of the distribution of the samples to the ideal scenario on a group level. An increase in uniformity will indicate a greater divergence from the ideal global structure and vice-versa.

As already mentioned in the previous sub-section, decreasing the temperature tends to penalize the hard negative pairs more. This is because hard negative pairs tend to have high cosine similarity (say, s_{ij}). With small temperature, the quantity $\frac{s_{ij}}{\tau}$ is further amplified, consequently resulting in a larger penalty (from Eqn 4). This causes samples constituting false negative pairs included in hard negative pairs to drift apart. Consequently, the local structure consisting of samples of any particular class is disturbed. This phenomenon gives rise to the "uniformity-tolerance dilemma" as studied in [9].

The effect of temperature can be better understood if we take the gradient of the loss with respect to any latent vector z_j. Taking the expression of the derivative of the loss $\mathcal{L}$, given by Eqn. 1, with respect to z_j, we get,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial z_j} &= \left[-\frac{z_{j^+}}{\tau} + \frac{\frac{z_{j^+}}{\tau} \cdot e^{C_{jj^+}} + \sum_{\substack{i=1 \\ i\neq j}}^{N} \frac{z_i}{\tau} \cdot e^{C_{ji}}}{e^{C_{jj^+}} + \sum_{\substack{i=1 \\ i\neq j}}^{N} e^{C_{ji}}} + \sum_{\substack{i=1 \\ i\neq j}}^{N} \frac{\frac{z_i}{\tau} \cdot e^{C_{ij}}}{e^{C_{ii^+}} + \sum_{\substack{k=1 \\ k\neq i}}^{N} e^{C_{ik}}} \right] \\ &= -\left[\frac{z_{j^+}}{\tau} \left(1 - p^{jj^+}\right) - \sum_{\substack{i=1 \\ i\neq j}}^{N} \frac{z_i}{\tau} \left(p^{j\Downarrow i} + p^{i\Downarrow j}\right) \right]\end{aligned} \tag{5}$$

where $C_{ji} = \frac{s_{ji}}{\tau}$, denotes the cosine similarity between the feature vectors z_j and z_i, scaled by temperature τ, and (z_j, z_{j+}) forms the positive pair. The quantity $p^{j\Downarrow i}$ is the probability of the pair (x_j, x_i) being predicted as a positive pair with the sample x_j as the anchor. Hence, from the expression of the displacement vector $\frac{\partial \mathcal{L}}{\partial z_j}$, we can arrive at the same conclusion as [9], that at a low-temperature value the sample z_j moves away from any sample z_i if they are mapped close to each other in the feature space. In other words, contrastive loss penalizes hard negative pairs. The effect of the reduction of temperature in different scenarios is discussed as follows.

Reducing Temperature for False Negatives: Going by the rule for gradient descent, ($z_j^{t+1} = z_j^t - \frac{\partial \mathcal{L}}{\partial z_j^t}$), the contribution of false negative pairs will be negative to the gradient of $\mathcal{L}$ with respect to the feature (latent) vector z_j. For false negative pairs, the value of cosine similarity between the two elements in the pair can be positive or negative depending on where the samples are mapped. Adjusting the temperature hyper-parameter allows us to control the contribution of the false negative pairs in the loss optimization process, by scaling the weights of the latent vectors. For two closely placed false negative samples, the sum of the corresponding probabilities will be high. If the temperature is decreased, the contribution of the sample z_i in the gradient increases further, resulting in the sample z_j drifting opposite to the direction of z_i. Conversely, if we take the derivative of $\mathcal{L}$ with respect to z_i, we will get a term involving z_j, which will enforce a similar effect on z_i. This results in the disruption of the local cluster structure in the feature space.

Reducing Temperature for Hard Negatives: For hard negative pairs, we can expect the two constituent samples to drift apart from each other if a low enough temperature is applied. However,

in the absence of a ground truth label, it is not possible to apply selective temperature moderation to all the pairs. If we decrease the temperature for all pairs whose cosine similarity is above a certain threshold (say, C_α), then the closely spaced false negative pairs will also be affected, resulting again in disruption of the local cluster structure.

What if there were no False Negatives? Now, if we assume that all the negative pairs are true negative pairs, then the problem becomes easier. In such an ideal scenario, (like in supervised contrastive learning [32]), we may make the mistake of assuming that we can safely decrease the temperature. Decreasing the temperature for true negative pairs will certainly improve performance up to a certain level, below which the performance degrades due to numerical instability [32], as the gradients become too large. This degradation in performance is due to the disruption in the global structure of the feature space. Disruption in local structure causes degradation of alignment in the feature space, whereas disruption in the global structure will cause an increase in uniformity [9, 33].

Increasing Global Temperature: On the other hand, increasing the temperature for all the samples has the opposite effect. As the temperature is increased, the drift in the false negative pairs is reduced, thereby helping in maintaining proper alignment. However, the uniformity may be affected as the repulsion between samples constituting true negative pairs including the hard true negatives, will also be reduced. Hence, increasing temperature causes an increase in alignment but affects uniformity.

4 Methodology

4.1 Motivation of Proposed Temperature Scaling Function

In self-supervised contrastive learning, we cannot know for certain the boundary between true and false negatives. However, we have described the effect of temperature on the feature space in the subsection before and can list some criteria we need to follow to design a proper temperature scaling function. The criteria are as follows: (1) A very low temperature in the highly negative cosine similarity region will disrupt the global structure, (2) A very low temperature in the highly positive cosine similarity region will disrupt the local structure, (3) A very low temperature for false negative pairs, which we can assume to lie in the range $[-s_{fn}, +1.0]$ can affect hard true negatives and true positives pairs, where $-s_{fn}$ denotes a cosine similarity score, (4) A high temperature will affect the uniformity of the feature space and delay convergence.

Let us assume $\tau(\cdot)$ is the temperature function, which takes the cosine similarity of a pair as input and outputs a temperature value for the same. For the subsequent parts of this literature, we will consider $\tau_{ij} = \tau(s_{ij})$.

$$\frac{\partial \mathcal{L}_i}{\partial s_{ii+}} = -\frac{\tau_{ii} - s_{ii+}\frac{\partial \tau_{ii}}{\partial s_{ii+}}}{\tau_{ii}^2} \cdot (1 - p_{ii+}) \quad (6) \qquad \frac{\partial \mathcal{L}_i}{\partial s_{ij}} = \frac{\tau_{ij} - s_{ij}\frac{\partial \tau_{ij}}{\partial s_{ij}}}{\tau_{ij}^2} \cdot p_{ij} \quad (7)$$

where s_{ii+} and s_{ij} denote the cosine similarity of positive and negative pairs, respectively.

For negative pairs, $\frac{\partial \mathcal{L}}{\partial s_{ij}} > 0 = \delta$, where δ is a non-negative number. From Eqn. 7, we get,

$$\frac{\tau_{ij} - s_{ij}\frac{\partial \tau_{ij}}{\partial s_{ij}}}{\tau_{ij}^2} \cdot p_{ij} \geq 0 \implies \frac{\tau_{ij} - s_{ij}\frac{\partial \tau_{ij}}{\partial s_{ij}}}{\tau_{ij}^2} \geq 0 \ [\because \ p_{ij} > 0] \implies \frac{\partial \tau_{ij}}{\partial s_{ij}} \leq \frac{\tau_{ij}}{s_{ij}} \quad (8)$$

Without loss of generality, we can always assume $\tau_{ij} > 0$. As the temperature parameter cannot be negative, the temperature value would be 0.0 or some positive constant. For $s_{ij} < 0$, to satisfy our criteria (1) and (3), we should have $\frac{\partial \tau_{ij}}{s_{ij}}$ is always less than some negative number. Hence, the slope of the temperature function is negative in the negative half of the cosine similarity space. Going by the same logic, the slope of the temperature function is less than some positive number in the positive half of the cosine similarity space. Thus, the slope can be negative, zero or positive. However, a negative slope in the positive half would mean that temperature would decrease at high cosine similarity, violating our criteria (2) and (3). A low temperature at high cosine similarity will affect the hard negative pairs and degrade the local structure. Therefore, we adopt a cosine similarity function

for the temperature function, such that the temperature does not violate criteria (4) at high cosine similarity values.

4.2 Proposed Framework

Combining all the above philosophies together we describe the framework proposed in this work. In this work, we use SimCLR [5] as the baseline framework. For satisfying the conditions derived in the section above, we adopt a cosine function of the cosine similarity as the temperature function, as shown in Alg. 1. The cosine function was chosen for the following reasons: (1) High temperature at high cosine similarity values prevents fluctuations in closely spaced false negative pairs, thereby preserving the local structure, (2) High temperature at low Cosine similarity prevents disruption of global structure, (3) Low temperature in the middle provides higher gradients for pairs that are true negatives and need to be pushed further apart. However, false negatives in this cosine similarity range will also be affected. Whether the improvement in uniformity due to the repulsion of true negative pairs compensates for the decrease in tolerance due to increased repulsion between false negative pairs is evident from the uniformity and tolerance plots for different configurations of the proposed temperature scaling function as shown in Sec. 6.

Algorithm 1: Temperature Scaling Function

Data: τ_{max} and τ_{min}
Input: $s_{ij} \rightarrow$ Cosine Similarity of the pair (x_i, x_j)
$\tau_{ij} = \tau_{min} + 0.5 \times (\tau_{max} - \tau_{min}) \times (1 + \cos(\pi(1 + s_{ij})))$

For different τ_{max} and τ_{min} values, we obtain different temperature scaling functions as shown in Fig. 1. We will analyze how different temperature scaling scheme affects performance in Sec. 6.2.

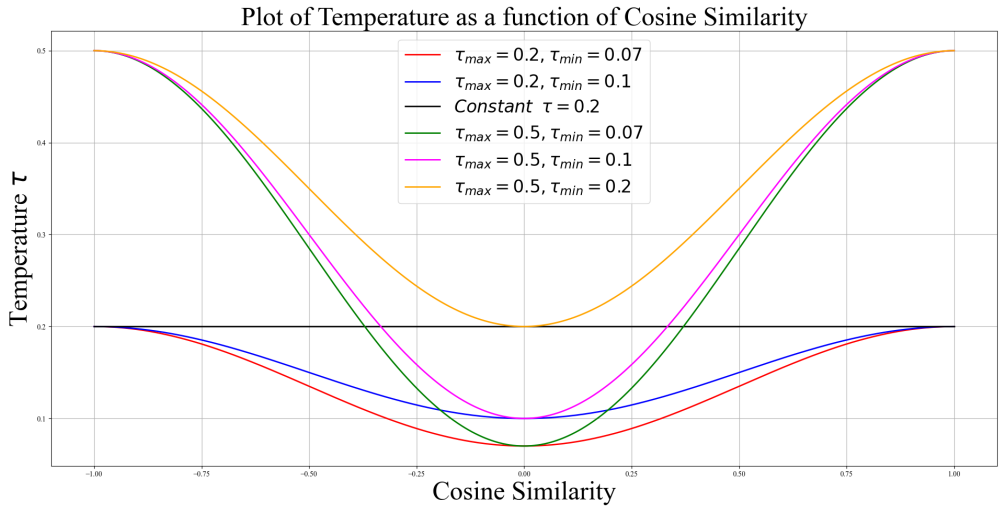

Figure 1: Plot of Temperature functions for different τ_{max} and τ_{min}. The figure is best visible at 200%.

4.3 Analysis of the Temperature Scaling Function

In the above discussions, we have assumed that the temperature scaling function attains minima at $s_{ij} = 0$. However, in the feature space, there is no such hard distinction between false negatives and true negatives in case of SSL. Cosine similarity of False Negative pairs can range from $-s_{fn}$ to $+1.0$. Decreasing the temperature around $s_{ij} = 0$ as in Alg. 1, can assign lower temperatures to false negative pairs, thereby pushing the constituent samples far apart. To reduce this phenomenon and decrease the possibility of pushing false negatives away, we can shift the minimum of the temperature function into the negative half-plane of cosine similarity, that is, towards the cosine similarity values

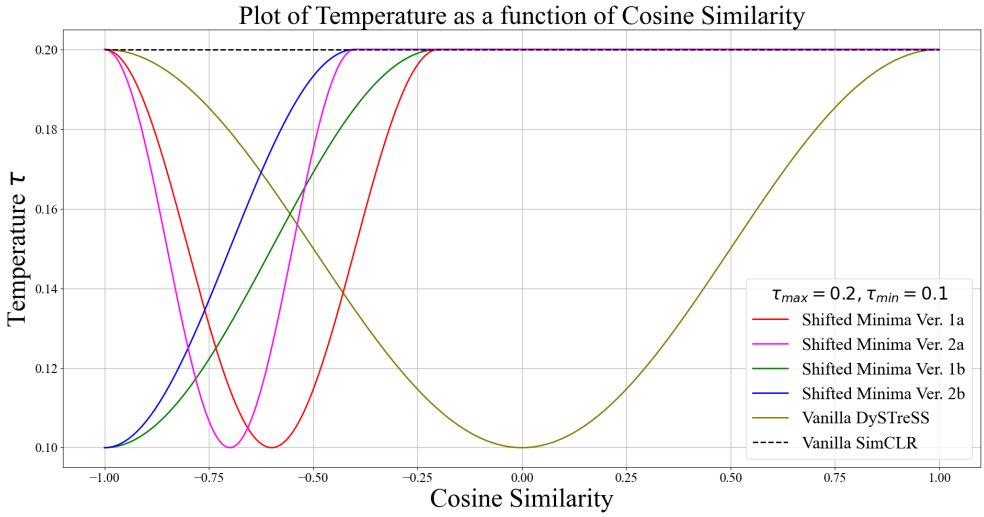

Figure 2: Plot of Different Versions of Temperature functions used in our experiments. The figure is best visible at 200%.

where true negatives lie. The effect of such modification can be seen in Fig. 4 in Sec. 6.2.2. As the displacement gradient (Eqn. 5) contribution corresponding to the false negative pairs reduces, the performance also improves.

Going a step further, increasing τ_{min} gradually with each epoch to approach τ_{max} as the training progresses. We intuit that it stabilizes the gradients from the true negative (or fringe false negative) pairs as the pre-training approaches its end. We can observe the effect of the above in Sec. 6.2.4.

5 Implementation Details

Datasets To study the effects of temperature in the self-supervised contrastive learning framework, we used 2 different datasets, namely, CIFAR10 [34], CIFAR100 [34] and Tiny-ImageNet [35]. We also study the effect of our proposed framework on the Long-tailed versions of the aforementioned datasets, which we term CIFAR10-LT and CIFAR100-LT.

Pre-training Model Architecture The encoder used in the pre-training model for experiments on CIFAR10 and CIFAR100 is ResNet18 [36]. The first convolutional layer in ResNet18 is replaced by a convolutional layer with a kernel of dimension 3×3 and the subsequent Max-pooling layer is removed. Similarly, for CIFAR10-LT and CIFAR100-LT, we used the aforementioned ResNet18, as well. However, for the experiments on Tiny-ImageNet, we use the original ResNet50 [36]. The last fully-connected layer from the ResNet network is removed for all experiments, and the output obtained from the ResNet encoder is fed into a 2-layer multi-layered perceptron (MLP) network called Projector. For the projector architecture, we follow the SimCLR [5], where the Linear layers are followed by Batch Normalization (BN) [37] layers, with a ReLU [38] activation function in between the first BN layer and the second Linear layer.

Training Procedure For the pre-training procedure, we use SGD optimizer (momentum$= 0.9$, weight decay factor$= 5e-4$) with a learning rate of 0.1 and batch size of 128 for all the datasets. For the balanced CIFAR and Tiny-ImageNet datasets, we run the optimization procedure for 200 epochs. For the Long-tailed versions of the CIFAR datasets, we adopt 500 epochs of training [8].

For the inference stage, we adopt a kNN classifier. For the balanced CIFAR datasets, we used a kNN classifier with $k = 200$, with cosine similarity-based weights. For the Long-tailed versions of CIFAR datasets, we used a k value of 1 and 10 with $L2$-distance-based weights. All the training and inference were run on a 16GB NVIDIA P100 GPU. Since the proposed framework is based on InfoNCE, the computation overhead is the same as contemporary frameworks such as SimCLR [5], MoCov2 [15], etc.

Augmentations During the pre-training stage, two augmented versions of each input image are generated and used as positive pairs. As augmentation, we randomly cropped each image and subsequently resized it to the original resolution of the image, followed by random horizontal flipping, color jittering, and grayscale conversion. The augmentations are followed by normalization by channel-wise mean and standard deviation values obtained from ImageNet [39] dataset.

6 Experimental Results and Ablations

6.1 Experimental Results

From the comparative study presented in Tab. 1, we observe that the best-performing configurations of the proposed framework outperform the contemporary SSL frameworks, both with and without stabilization. For the CIFAR-100 dataset (Tab. 2), the vanilla DySTreSS framework outperforms all the SSL frameworks.On the long-tailed datasets (Tab. 3 and 4), the DySTreSS outperforms the baseline vanilla SimCLR [5] on both 10-NN and 1-NN accuracy. All results presented in the Tables 1,2, 3 and 4 have been reproduced on the aforesaid system.

Table 1: Comparison with contemporary state-of-the-art Self-supervised Contrastive Learning frameworks on CIFAR10 dataset.

Framework	τ_{max}	τ_{min}	Accuracy 200-NN
SimCLR [5]	0.2	–	83.65
MoCoV2 [15]	0.07	–	83.9
MACL [29]	0.1	–	82.58
DySTreSS	0.2	0.07	**84.47**
DySTreSS + Stabilization	0.2	0.1	**84.67**

Table 2: Comparison with contemporary state-of-the-art Self-supervised Contrastive Learning frameworks on CIFAR100 dataset.

Framework	τ_{max}	τ_{min}	Accuracy 200-NN
SimCLR [5]	0.2	-	51.77
MoCoV2 [15]	0.07	-	52.64
MACL [29]	0.1	-	52.15
DySTreSS	0.2	0.07	**53.54**
DySTreSS + Stabilization	0.2	0.07	**53.81**

On the CIFAR-10 dataset, the proposed framework outperforms the recent state-of-the-art (SoTA) framework MACL [29] by 1.89%. On applying the stabilization mechanism on DySTreSS, the accuracy increases by 0.2%. On the CIFAR-100 dataset, we again see an improvement in performance by 0.27% on the application of the stabilization. It outperforms MACL by 1.39% and 1.66%, with and without stabilization, respectively. On the long-tailed versions of the CIFAR datasets, the proposed framework improves upon the SoTA SimCLR by more than 1%, as seen in tables 3 and 4.

Table 3: Comparison with contemporary state-of-the-art Self-supervised Contrastive Learning frameworks on CIFAR10-**LT** dataset.

Framework	τ_{max}	τ_{min}	Accuracy 1-NN	10-NN
SimCLR [5]	0.2	-	57.12	55.29
DySTreSS	0.2	0.07	**58.36**	56.40
DySTreSS	0.2	0.1	58.34	**56.54**

Table 4: Comparison with contemporary state-of-the-art Self-supervised Contrastive Learning frameworks on CIFAR100-**LT** dataset.

Framework	τ_{max}	τ_{min}	Accuracy 1-NN	10-NN
SimCLR [5]	0.2	-	28.27	26.18
DySTreSS	0.2	0.07	**29.43**	27.10
DySTreSS	0.2	0.1	28.82	**27.32**

On TinyImageNet [35], the proposed framework achieved a 200-NN accuracy of 40.09% for $\tau_{min} = 0.1$ and $\tau_{max} = 0.2$, outperforming vanilla SimCLR (39.75%) under the same conditions.

6.2 Ablation Studies

6.2.1 Effect of Different Temperature Ranges

The performance of the proposed framework also varies as the values of τ_{max} and τ_{min} are varied. We experimented with different temperature ranges as given in Fig. 3 and observed that the proposed framework achieves the best accuracy in the temperature range $\tau_{min} = 0.07$ to $\tau_{max} = 0.2$ for both the CIFAR datasets. The different temperature range affects different types of samples differently. It is evident from Fig. 3, an increase in temperature causes a decrease in uniformity. For example, for temperature ranges((τ_{min}, τ_{max})), $(0.2, 0.5)$ and $(0.2, 0.2)$, we see a decrease in uniformity and an increase in tolerance. Whereas, for temperature ranges with lower τ_{min} but the same τ_{max}, the general trend shows that the uniformity is greater and tolerance is lower. This conforms with our theory in Sec. 3.4. On the contrary, for CIFAR100, due to the presence of *more true negative pairs* than CIFAR10, increasing temperature inhibits the uniformity (tolerance) from increasing (decreasing) sufficiently to improve performance. Hence, for the same τ_{min}, the performance was better for a lower τ_{max}.

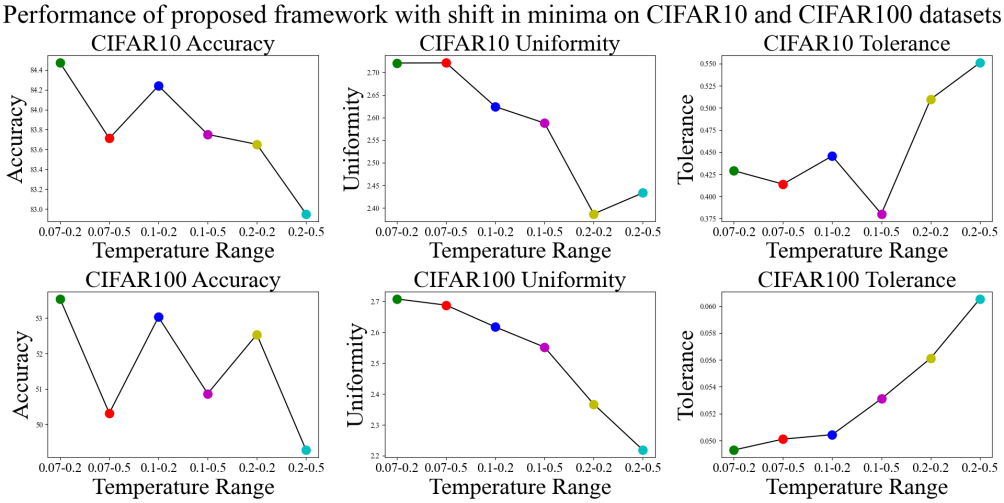

Figure 3: Plot of Accuracy, Uniformity and Tolerance on CIFAR10 (top) and CIFAR100 (bottom) datasets for different temperature ranges. The figure is best visible at 500%.

6.2.2 Effect of Shift in the Minimum

In Fig. 4, we present the accuracy, uniformity, and tolerance values for different temperature functions given in Fig. 2. We observe that shifting the minimum of the temperature function influences different samples and consequently changes the structure of the feature space and performance accordingly. Along the x-axis in Fig. 4, 'SMvx' denotes 'Shifted Minima Version x'. 'Ver. xa' and 'Ver. xb' denotes two shifts of -0.2 and -0.4 from the origin. For the CIFAR10 dataset, we can observe that a shift of -0.2 is better than a shift of -0.4, while the reverse is true for the CIFAR100 dataset. However, none of the configurations yields better results than the vanilla version with no shift. A lower temperature towards $s_{ij} = -1$ increases uniformity, as evident from the difference in uniformity between SMv1b and SMv2b or SMV1a and SMv2a, while the reverse is true for tolerance. The drop in performance is primarily due to the fact that a constant temperature in the range $[\tau_{shift}, 1.0] \mid (\tau_{shift} \in \{-0.2, -0.4\})$ caused by the shift results in decreased repulsion of the hard true negative samples.

6.2.3 Effect of Learning Rate

As the temperature is decreased, the displacement gradients (Eqn. 5) increase, resulting in an increase in the magnitude of fluctuations from false negative pairs. A low learning rate plays a crucial role in this scenario in smoothening out the fluctuations. On the contrary, at a high learning rate, the fluctuations are amplified and should degrade the performance. However, from Fig. 5, we observe

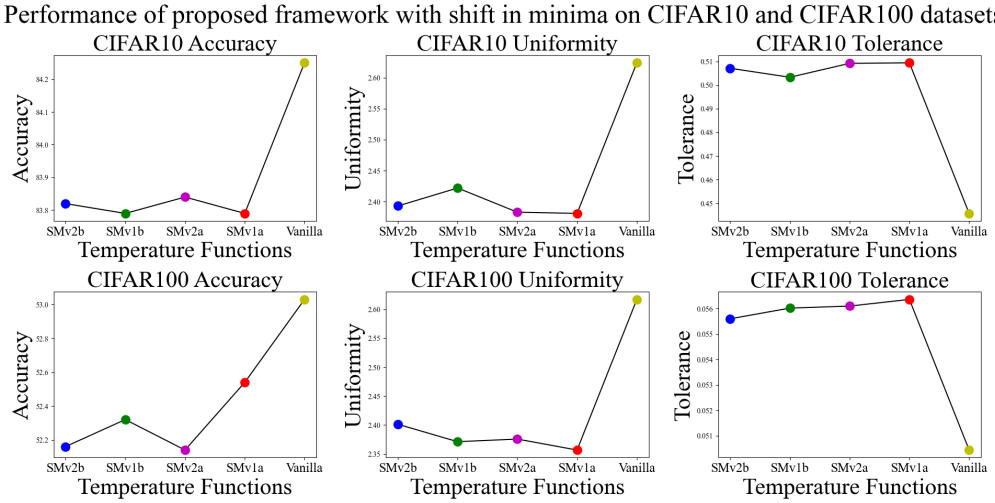

Figure 4: Plot of Accuracy, Uniformity, and Tolerance with a shift in minima for CIFAR10 (top) and CIFAR100 (bottom) dataset. The colour codes are matched to the curves in Fig. 2. The figure is best visible at 500%.

this effect for CIFAR10 only, as the number of false negative pairs in a batch is greater than that in CIFAR100.

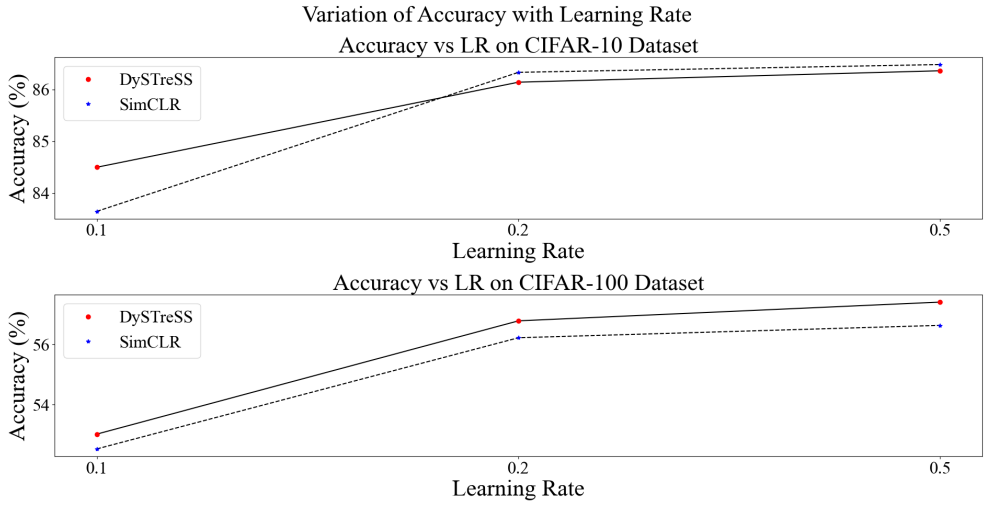

Figure 5: Plot of Accuracy with change in Learning Rate for the datasets CIFAR10 (top) and CIFAR100 (bottom). The figure is best visible at 500%.

6.2.4 Effect of Stabilization

As the pre-training phase approaches its end, we tried to reduce the fluctuations for false negative samples due to low temperature by linearly increasing τ_{min} till it becomes equal to τ_{max}. We apply this procedure to all the temperature functions shown in Fig. 2. As intuited, from Fig. 6, we can observe that this procedure introduces a stabilization effect as convergence is achieved and improves performance for most temperature functions on the CIFAR10 dataset. However, for the CIFAR100 dataset, stabilization does not improve performance. This conforms with the results in Sec. 6.2.1. The temperature functions along the x-axis denote the same as described in Sec. 6.2.2. We also present the t-SNE plots for the test samples of the CIFAR10 dataset with the proposed framework DySTreSS,

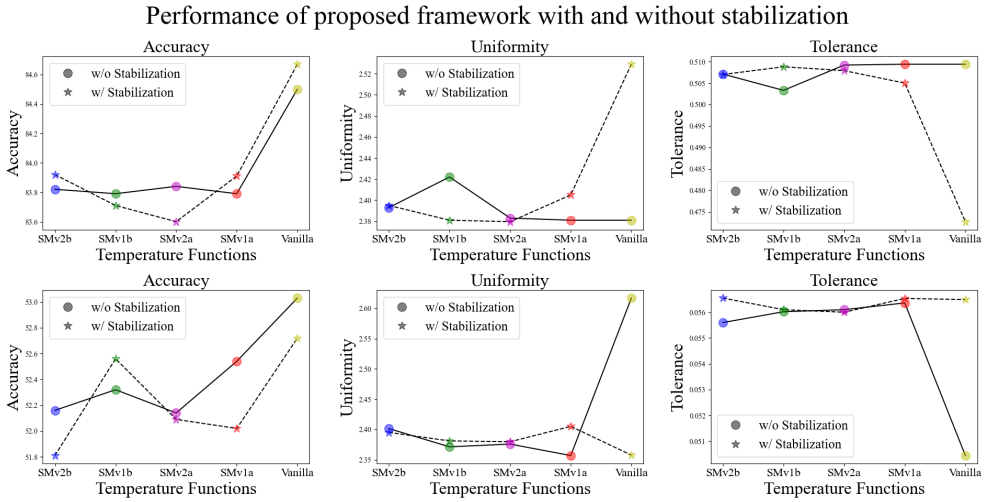

Figure 6: Plot of Accuracy, Uniformity, and Tolerance with and without stabilization on CIFAR10 (top) and CIFAR100 (bottom) datasets. The figure is best visible at 500%.

the stabilized version of DySTreSS, and vanilla SimCLR [5] in Fig. 7, 8 and 9, respectively. It is quite evident from the t-SNE plots itself, that the separability is better for DySTreSS than SimCLR.

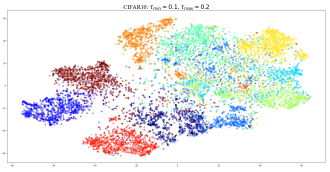

Figure 7: t-SNE plot of test samples for Vanilla DySTreSS

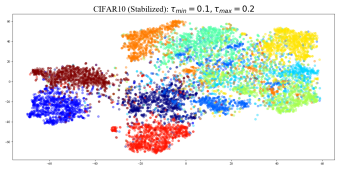

Figure 8: t-SNE plot of test samples for Stabilized DySTreSS

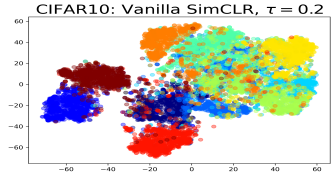

Figure 9: t-SNE plot of test samples for Vanilla SimCLR

7 Conclusion

In this work, we identified a specific category of pairs in self-supervised contrastive learning and analyzed the effect of temperature on such pairs in the optimization of the InfoNCE loss. We observed that by varying the temperature as a function of the cosine similarity values of the feature vectors of all pairs, we can control the dynamics of the optimization process and improve the performance of the baseline method, SimCLR. Through extensive experiments, we show that the proposed framework improves performance over the baseline and state-of-the-art algorithms. Finally, this work lay the foundation for further research into the working principle and dynamics of the InfoNCE loss function.

References

[1] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 1

[2] Deepak Pathak, Philipp. Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 1

[3] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 649–666, Cham, 2016. Springer International Publishing. 1, 2

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3, 6, 7, 8, 11

[6] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *Computer Vision – ECCV 2022*, pages 668–684, Cham, 2022. Springer Nature Switzerland. 2, 3

[7] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[8] Anna Kukleva, Moritz Böhle, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. Temperature schedules for self-supervised contrastive methods on long-tail data. *arXiv preprint arXiv:2303.13664*, 2023. 2, 3, 7

[9] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, 2021. 2, 3, 4, 5

[10] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE TPAMI*, 2020. 2

[11] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. 2021. 2, 3

[12] 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. 2, 3

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, et al. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, 2020. 2

[14] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 2, 3

[15] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. 2, 3, 7, 8

[16] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2, 3

[17] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 2014. 2

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2

[21] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2

[22] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2

[23] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *ICML*, 2019. 2

[24] Miguel A Bautista, Artsiom Sanakoyeu, Ekaterina Tikhoncheva, and Bjorn Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *NeurIPS*, 2016. 2

[25] Siladittya Manna, Umapada Pal, and Saumik Bhattacharya. Mio: Mutual information optimization using self-supervised binary contrastive learning. *arXiv preprint arXiv:2111.12664*, 2021. 2

[26] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 3

[27] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. volume 34, pages 11834–11845, 2021. 3

[28] Oliver Zhang, Mike Wu, Jasmine Bayrooti, and Noah D. Goodman. Temperature as uncertainty in contrastive learning. *arXiv*, abs/2110.04403, 2021. 3

[29] Huang Zizheng, Chen Haoxing, Wen Ziqi, Zhang Chao, Li Huaxiong, Wang Bo, and Chen Chunlin. Model-aware contrastive learning: Towards escaping the dilemmas. In *ICML*, 2023. 3, 8

[30] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 3

[31] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 3

[32] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 5

[33] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020. 5

[34] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. 7

[35] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 7, 8

[36] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 7

[38] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 7

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 8

A Etiology of the Temperature Function

The primary objective of our proposed temperature function is to modulate the temperature for sample pairs to improve representation learning. In SSL, the primary hurdle is the repulsion between the Hard False Negative pairs. Hard True negative pairs also hinder the learning process. In this section, we will discuss the mathematical motivation behind adopting a cosine similarity function as the temperature-modulating function in our work. Ideally, for negative pairs, the gradient of the InfoNCE loss with respect to s_{ij} will be non-negative, because, if loss decreases, then the cosine similarity of negative pairs should decrease. We assume that the value of this gradient is δ, as shown in Eqn. 1.

$$\frac{\partial \mathcal{L}}{\partial s_{ij}} = \frac{\tau_{ij} - s_{ij}\frac{\partial \tau_{ij}}{\partial s_{ij}}}{\tau_{ij}^2} p_{ij} = \delta \quad \text{where } \delta < \epsilon \quad \text{and } \delta, \epsilon > 0 \tag{1}$$

where ϵ is a small non-negative number. Expanding the Eqn. 1, we get,

$$\begin{aligned}
&\frac{\tau_{ij} - s_{ij}\frac{\partial \tau_{ij}}{\partial s_{ij}}}{\tau_{ij}^2} p_{ij} = \delta \\
\implies &\frac{\tau_{ij} - s_{ij}\frac{\partial \tau_{ij}}{\partial s_{ij}}}{\tau_{ij}^2} = \frac{\delta}{p_{ij}} \\
\implies &\tau_{ij} - s_{ij}\frac{\partial \tau_{ij}}{\partial s_{ij}} = \tau_{ij}^2 \frac{\delta}{p_{ij}} \\
\implies &\frac{\partial \tau_{ij}}{\partial s_{ij}} = \frac{1}{s_{ij}}\left[\tau_{ij} - \tau_{ij}^2 \frac{\delta}{p_{ij}}\right] \\
\implies &\frac{\partial \tau_{ij}}{\partial s_{ij}} = \frac{\tau_{ij}}{s_{ij}}\left[1 - \frac{\tau_{ij}\delta}{p_{ij}}\right]
\end{aligned} \tag{2}$$

We can assume that $\tau_{ij} > 0$ without loss of generality.

In self-supervised contrastive learning, the temperature should be high for false negatives to prevent too much repulsion. We have discussed the criteria and the motivation behind our temperature function in Sec. 4.1 of the main manuscript. Also, the temperature should not be very small in the regions with highly negative cosine similarity. We assume that the number of false negatives decreases as we move towards the point $s_{ij} = 0.0$. Hence, for the vanilla case, we will consider two regions, (1) $s_{ij} > 0$ and (2) $s_{ij} \leq 0$.

Expanding the expression for p_{ij} in Eqn. 2, we get,

$$
\begin{aligned}
\frac{\partial \tau_{ij}}{\partial s_{ij}} &= \frac{\tau_{ij}}{s_{ij}} \left[1 - \frac{\tau_{ij}\delta}{\frac{exp(s_{ij}/\tau_{ij})}{\sum_{k=1}^{N} exp(s_{ik}/\tau_{ik})}} \right] \\
\implies \frac{\partial \tau_{ij}}{\partial s_{ij}} &= \frac{\tau_{ij}}{s_{ij}} \left[1 - \frac{\tau_{ij}\delta \sum_{k=1}^{N} exp(s_{ik}/\tau_{ik})}{exp(s_{ij}/\tau_{ij})} \right] \\
\implies \frac{\partial \tau_{ij}}{\partial s_{ij}} &= \frac{\tau_{ij}}{s_{ij}} \left[1 - \tau_{ij}\delta \frac{exp(s_{ij}/\tau_{ij}) + \sum_{\substack{k=1 \\ k\neq j}}^{N} exp(s_{ik}/\tau_{ik})}{exp(s_{ij}/\tau_{ij})} \right] \\
\implies \frac{\partial \tau_{ij}}{\partial s_{ij}} &= \frac{\tau_{ij}}{s_{ij}} \left[1 - \tau_{ij}\delta \frac{exp(s_{ij}/\tau_{ij}) + \sum_{\substack{k=1 \\ k\neq j}}^{N} exp(s_{ik}/\tau_{ik})}{exp(s_{ij}/\tau_{ij})} \right] \\
\implies \frac{\partial \tau_{ij}}{\partial s_{ij}} &= \frac{\tau_{ij}}{s_{ij}} \left[1 - \tau_{ij}\delta \left(1 + \frac{\sum_{\substack{k=1 \\ k\neq j}}^{N} exp(s_{ik}/\tau_{ik})}{exp(s_{ij}/\tau_{ij})} \right) \right] \\
\implies \frac{\partial \tau_{ij}}{\partial s_{ij}} &= \frac{\tau_{ij}}{s_{ij}} \left[1 - \tau_{ij}\delta \left(1 + K \cdot exp(-\frac{s_{ij}}{\tau_{ij}}) \right) \right]
\end{aligned}
\tag{3}
$$

where $K = \sum_{\substack{k=1 \\ k\neq j}} exp(\frac{s_{ik}}{\tau_{ik}})$ is taken as a constant with respect to s_{ij}, that is, $\frac{\partial K}{\partial s_{ij}} = 0$.

If $N \to \infty$ or for very large N, we can safely assume

$$
\frac{exp(s_{ij}/\tau_{ij}) + \sum_{\substack{k=1 \\ k\neq j}}^{N} exp(s_{ik}/\tau_{ik})}{exp(s_{ij}/\tau_{ij})} \simeq \frac{\sum_{\substack{k=1 \\ k\neq j}}^{N} exp(s_{ik}/\tau_{ik})}{exp(s_{ij}/\tau_{ij})} = K \cdot exp(-\frac{s_{ij}}{\tau_{ij}}) \tag{4}
$$

Hence, Eqn. 3 reduces to,

$$
\frac{\partial \tau_{ij}}{\partial s_{ij}} = \frac{\tau_{ij}}{s_{ij}} \left[1 - \tau_{ij}\delta \left(K \cdot exp(-\frac{s_{ij}}{\tau_{ij}}) \right) \right] \tag{5}
$$

Solving the first-order nonlinear ordinary differential equation given by Eqn. 5, we get,

$$
\tau_{ij} = \frac{s_{ij}}{\log(\delta \cdot K \cdot s_{ij} - c)} \tag{6}
$$

where c is the integral constant.

To find the value of c, we have to solve for the value of τ_{ij} at the endpoints of the cosine similarity space. It is to be remembered, τ_{ij} takes the value τ_{max} at $s_{ij} = -1$ and $s_{ij} = +1$ (Please refer to Sec. 4.1 in the main manuscript).

Solving, the above equation for the two above-mentioned cases, we get,

$$
\begin{aligned}
c^{-} &= -\delta \cdot K - exp(-1/\tau_{max}) \\
c^{+} &= -\delta \cdot K - exp(1/\tau_{max})
\end{aligned}
\tag{7}
$$

Varying the value of the constant in the range $[c^{-}, c^{+}]$, we get different curves with different slopes for different values of δ and K, as shown in the Fig. 1

We can observe, that the plotted curves in Fig. 1 do in fact show positive and negative gradients on the positive and negative half plane of the cosine similarity space, respectively, as stated in the Sec. 4.1 of the main manuscript. This establishes our theoretically derived condition for the slopes of the temperature-modulating function in the negative and positive halves of the cosine similarity space.

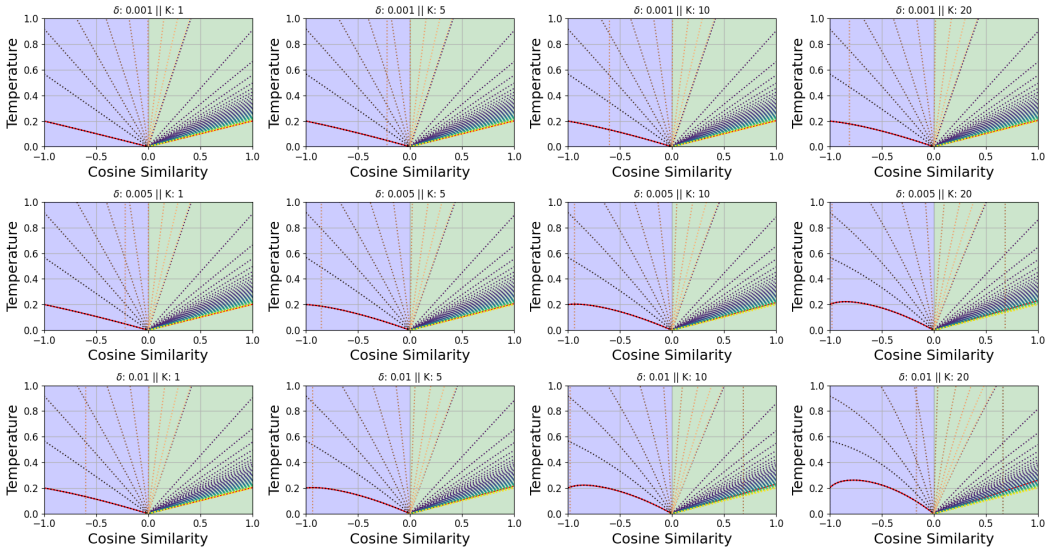

Figure 1: Plots of the solution of ODE in Eqn. 6 for different values of the integral constant, over different values of δ and K.

B Pseudocode for Vanilla DySTreSS

Algorithm 1: PyTorch-style pseudocode for Vanilla DySTreSS

```
# f:  Encoder Network, N: Batch Size, XEnt:  Cross Entropy Loss
# tmin, tmax:  Minimum Temperature, Maximum Temperature
for x in loader:
   # Augment to generate positive pairs
   x_1, x_2 = augment(x)
   # Pass through the encoder to get feature vectors
   z_1, z_2 = f(x_1), f(x_2)
   # L2-Normalize the feature vectors
   z_1, z_2 = F.normalize(z_1, dim = -1), F.normalize(z_2, dim = -1)
   # Generate Cosine Similarity matrix
   sim = z_1 @ z_2.T
   # Generate Similarity matrix mask
   sim_mask = get_sim_mask(N) # From Algorithm 2
   # Segregate the positive pair cosine similarity values
   pos_x = torch.cat([sim.diag(N), sim.diag(-N)], dim = 0).reshape(2N,1)
   pos_temp = tmin + 0.5(tmax - tmin)(1 + cos(π(1 + pos_x.detach())))
   pos_x = pos_x / pos_temp
   # Segregate the negative pair cosine similarity values
   neg_x = sim[sim_mask].reshape(2N, -1)
   neg_temp = tmin + 0.5(tmax - tmin)(1 + cos(π(1 + neg_x.detach())))
   neg_x = neg_x / neg_temp
   # Concatenate with respect to anchors
   logits = torch.cat([pos_x, neg_x], dim = -1)
   labels = torch.zeros(2N)
   # Compute Cross Entropy loss
   loss = XEnt(logits, labels)
   # Optimize Loss
   loss.backward()
   optimizer.step()
```

Algorithm 2: PyTorch-style pseudocode for Generating Similarity Matrix Mask

```
def get_sim_mask(N):
    # Generate Similarity matrix mask
    sim_mask = torch.ones(2N).fill_diagonal(0)
    sim_mask = sim_mask.fill_nth_diagonal(0, offset = N)
    sim_mask = sim_mask.fill_nth_diagonal(0, offset = -N)
    return sim_mask
```