LABEL SIMILARITY AWARE CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive learning dramatically improves the performance in self-supervised learning by maximizing the alignment between two representations obtained from the same sample while distributing all representations uniformly. Extended supervised contrastive learning boosts downstream performance by pulling embedding vectors belonging to the same class together, even though vectors are obtained from different samples. In this work, we generalize the supervised contrastive learning approach to the universal framework allowing us to fully utilize the ground truth similarities between samples. All pairs of representations are relatively pulled together in proportion to the label similarity, not equally pulling representations having the same class label. To quantitatively interpret the feature space after contrastive learning, we propose a label similarity aware alignment and uniformity, which measures how genuinely similar samples are aligned and how feature distribution preserves the maximal information. We prove asymptotically and empirically that our proposed contrastive loss optimizes two properties, and optimized properties positively affect task performance. Comprehensive experiments on NLP, Vision, Graph, and Multimodal benchmark datasets using BERT, ResNet, GIN, and LSTM encoders consistently showed that our loss outperforms the previous self-supervised and supervised contrastive losses upon a wide range of data types and corresponding encoder architectures. Introducing a task-specific label similarity function further facilitates downstream performance.

1 INTRODUCTION

Contrastive learning is one of the widely used methods in self-supervised learning. Since it is hard to define similarities between samples in unsupervised learning lacking annotations, SimCLR (Chen et al., 2020) regard an augmented sample as a positive representation and maximize mutual information between the original and the positive representation compared to remaining pairs with other samples in the same mini-batch using NT-Xent loss function. This approach is simple but powerful, so that self-supervised learning using derivatives of SimCLR framework with NT-Xent loss (He et al., 2020; Grill et al., 2020; Caron et al., 2020) becomes a powerful contrastive pre-training method over wide-range of research fields. (Radford et al., 2021; Gao et al., 2021; You et al., 2020)

Often neglected due to record-breaking success in self-supervised learning, contrastive learning using triplet loss (Weinberger et al., 2005) or N-pair loss (Sohn, 2016) had recorded outstanding performances in supervised classification. More recently, SupCon (Khosla et al., 2020) generalized SimCLR as a supervised contrastive framework. The underlying logic of SupCon is to maximize the mutual information between original *and* positive representations having the same class label, in addition to their augmented feature vectors. SupCon framework shows an extraordinary performance on image classification and other tasks (Gunel et al., 2020; Li et al., 2022; Mai et al., 2022).

One remaining challenge for contrastive learning is to address real-world problems that are insufficient to define every classification inference as *perfectly correct* or *perfectly wrong*. For example, consider the problem of counting the number of people in an image. Yet the label above is discrete ($y \in \mathbb{N}$) as classic classification, reducing this problem into a multiclass classification neglects relative correctness between the label and answers; for ground truth 3, answering 4 is *relatively more correct* than answering 6. Since real-world labels are often continuous ($y \in \mathbb{R}$, i.e. regression) or multidimensional ($y \in \mathbb{R}^d$), considering subtle differences between labels becomes more important.

In this work, we propose a new contrastive objective function that further generalizes the SupCon loss to reflect relative similarity between samples. Label-similarity Aware Contrastive Loss (LASCon) measures the ground truth label similarity and carefully contrast representations in proportion to the similarity - representation pairs having similar labels are pulled stronger than pairs having distinct labels. LASCon loss maintains useful properties of SimCLR and SupCon, while learns continuous relations between samples unlike SimCLR or SupCon which learn discrete relationships.

We interpret the enhanced quality of continuous decision boundaries by assessing a generalized alignment and uniformity, which measures how well representations of similar samples are aligned and how uniformly representations are distributed to available feature space. Our new circular visualization method also effectively captures subtle distributional changes allowing us to interpret the quality of embedding spaces intuitively. While the concept of label similarity aware contrastive loss were recently proposed independently by several papers throughout multiple domains, to the best of our knowledge this is the first study quantitatively investigating topologies of representation space and find their connection to downstream performances.

Our resulting loss showed an extraordinary performances on four downstream tasks in NLP, Vision, Graph, and Multimodal tasks. Further development using non-linear label similarity shows that our loss can further boost supervised representation learning by focusing on strong-positive samples.

2 PRELIMINARY

Before we discuss about diverse contrastive losses and their properties, we would first revisit two core concepts: ***alignment*** and ***uniformity***. Then we would compare previous self-supervised and supervised contrastive loss functions by summarizing ***different views on positive samples***.

2.1 ALIGNMENT AND UNIFORMITY

Contrastive learning has shown remarkable success in a wide range of tasks. While exact architecture and training methodologies differ, recent research on contrastive learning tends to share a loss function based on NT-Xent (Negative Cross Entropy) loss function suggested by Chen et al. (2020). Wang & Isola (2020) suggested understanding of NT-Xent loss function as simultaneously optimizing two criteria, **alignment** and **uniformity**. Interpretations of two properties are as follows:

- **Alignment**: How embedding vectors of positive input pairs are close to each other.
- **Uniformity**: How input representations are uniformly distributed in embedding space.

The simultaneous optimization of alignment and uniformity is crucial in contrastive learning. A perfectly aligned embedding space can map similar samples to the same point, but this representation space is useless since the encoder cannot distinguish different samples. Similarly, an embedding space with perfect uniformity is impractical because the encoder loses all applicable information among training data while distributing all data points uniformly. Chen et al. (2021) further generalize the standard self-supervised contrastive loss by multiplying scaling constant to the uniformity term. Wang & Liu (2021) interpret effects of temperature using alignment/uniformity

2.2 CONTRASTIVE LOSS FUNCTIONS

2.2.1 SELF-SUPERVISED CONTRASTIVE LOSS: NT-XENT

Definition Chen et al. (2020) proposes Normalized Temperature-scaled Cross Entropy (NT-Xent) loss, which is maximizing *softmax-like relative similarity* defined as:

$$u(\mathbf{x}_i, \mathbf{x}_j) = \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j / \tau\right)}{\sum_{a \in I \setminus \{i\}} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)}$$

within the minibatch I and the normalized representation vectors $\{\mathbf{z}_i\}_{i \in I}$. An adjustable *temperature* $\tau > 0$ is used to control the penalties for hard negative samples. (Wang & Liu, 2021)

Positive Pair *Two augmented representations from the same sample.* Since self-supervised learning condition does not consider any label information, each data is augmented twice and these augmented representations coming from the same raw data are considered as positive pairs.

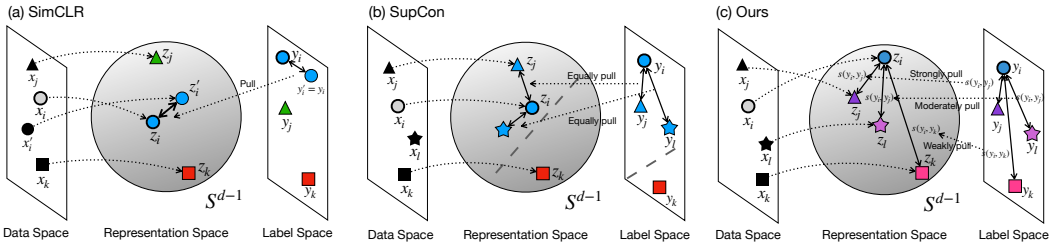

Figure 1: Visualization of contrastive learning frameworks.

For an index of positive sample $j(i) \in I$ for $i \in I$, the NT-Xent loss is then defined as

$$\mathcal{L}_i^{\text{SELF}} = -\log u(\mathbf{x}_i, \mathbf{x}_{j(i)}), \qquad \mathcal{L}^{\text{SELF}} = \sum_{i \in I} \mathcal{L}_i^{\text{SELF}} \tag{1}$$

The quality of contrastive learning depends on the definition of *semantic resemblance*. Hence existing works often focus on developing new augmentation (You et al., 2020; Wei & Zou, 2019) or generating hard/adversarial negatives (Kalantidis et al., 2020; Ho & Nvasconcelos, 2020). Von Kügelgen et al. (2021); Tian et al. (2020) studied the effects of augmentation on representation level.

2.2.2 Supervised Contrastive Loss

Definition Khosla et al. (2020) suggested applying contrastive learning to the supervised classification of images, where the goal is to predict the discrete class label y for given inputs. One can naturally consider pair of items belonging to the same class as positive pairs, and others as negatives.

For the positive indices $P(i) = \{j \in I \setminus \{i\} \mid y_j = y_i\}$, the supervised contrastive loss is defined as

$$\mathcal{L}_i^{\text{SUP}} = \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log u(\mathbf{x}_i, \mathbf{x}_p), \qquad \mathcal{L}^{\text{SUP}} = \sum_{i \in I} \mathcal{L}_i^{\text{SUP}} \tag{2}$$

$\mathcal{L}^{\text{SUP}}$ is a generalization of $\mathcal{L}^{\text{SELF}}$, as Equation 2 reduces to the Equation 1 for the case where every input data belongs to distinct class, and the only positive pair comes from the augmentation.

Positive Pair *Representations having the same class label.* Detailed definition of class label differs depending on fields and tasks. Gunel et al. (2020); Li et al. (2022); Mai et al. (2022) applied SupCon on NLP (GLUE), Graph (Drug-Protein Interaction), and Multimodal (CMU-MOSI) tasks.

2.3 Generalized Contrastive Loss

Previous Approach Generalizing contrastive loss has been suggested very recently in some application domains. Wang et al. (2022) derived a contrastive loss by assuming that predicting distribution is proportional to the similarity between label distributions and assessed on gaze estimation. Anonymous (2022) showed that generalization of SupCon by smoothing the binary nature of SupCon can lead to outstanding performance in predicting sentiment from speech input. Dufumier et al. (2021b) suggested the usage of proxy labels to incorporate prior information into the self-supervised contrastive learning. Dufumier et al. (2021a) formulated above approach using conditional alignment and uniformity, approaching the similar mathematical formulation of loss function as ours.

Limitation and Motivation Previous works have extended contrastive losses and showed empirical effectiveness. However, for representation learning, we found that previous works lack a quantitative assessment of embedding space topologies and their correlation with performances. We also inquired the universality for diverse fields and tasks utilizing various architectures and data formats. Therefore we combine and extend previous contrastive frameworks by formulating label-similarity aware supervised contrastive loss, carefully investigating the transformation of embedding spaces using generalized alignment/uniformity, and conducting experiments on various domains.

3 LABEL SIMILARITY AWARE CONTRASTIVE LEARNING

3.1 PROBLEM DEFINITION

Our target problem aims to predict label $y_i \in \mathcal{Y}$ from its input data $\mathbf{x}_i \in \mathcal{X}$. We consider the underlying true labeling function $\mathcal{F}$, whose value is known only at the training data points. For multimodal setting, the input data is a collection of multiple data formats $\mathbf{x}_i = \{\mathbf{x}_i^A, \mathbf{x}_i^B, \mathbf{x}_i^C\} \in \mathcal{X}$.

We would annotate the underlying distribution of data as p_{data}, where the minibatch $\{\mathbf{x}_i : i \in I\}$ are sampled from. For the indices of minibatch $I = \{1, 2, \dots |I|\}$, we define $A(i) = \{j \in I : j \neq i\}$.

On the space of all possible labels $\mathcal{Y}$, we introduce **label similarity function** $s : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ which measures how similar two labels are. For implementation purposes, we consider the mini-batch - normalized relative label similarity $\tilde{s}$ to be close to 1 for the similar pair as:

$$\tilde{s}(y_i, y_j) = \frac{s(y_i, y_j)}{\sum_{k \in A(i)} s(y_i, y_k)}$$

Our goal is to boost the encoder network to build a more effective representation space, which can ***better reflects the genuine similarities between samples using their labels***.

3.2 REPRESENTATION LEARNING FRAMEWORK

Our formulation of the problem and supervised contrastive learning condition does not rely on augmentation as a method to generate positive pairs. However, to make a fair comparison, we structured a similar framework to that used in SimCLR and SupCon including the augmentation module.

Here, we would reformulate the main components of our framework and their notations. Section D.1 and Figure 10 provides more detailed explanation.

- **Augmentation Module**, $Aug(\cdot)$. For each input $\mathbf{x}$, we generate one random augmentation, $\mathbf{x}' = Aug(\mathbf{x})$. The augmented sample represents a different view of the data and contains some modified subset of the information in the original sample. Augmentation strategy varies depending on the specific task and data format. The augmentation module is omitted for multimodal tasks since multiple encoders already provide different views.
- **Encoder Module**, $Enc(\cdot)$. For each input $\mathbf{x}$, encoder maps $\mathbf{x}$ to a representation vector $\mathbf{r} = Enc(\mathbf{x})$. Both original and augmented samples separately pass the same encoder, resulting in a pair of representation vectors. r is normalized to the unit hypersphere.
- **Projection Module** $Proj(\cdot)$. For each representation $\mathbf{r}$, projector transforms $\mathbf{r}$ to a vector $\mathbf{z} = Proj(\mathbf{r})$. We set $Proj(\cdot)$ as multi-layer perceptron with a single hidden layer. Resulting vector $\mathbf{z}$ is normalized to a unit hypersphere, which enables using an inner product to measure the cosine similarity between samples. The projection layer is replaced to the prediction layer at the inference time.
- **Prediction Module** $Pred(\cdot)$. For each representation $\mathbf{r}$, predictor maps $\mathbf{r}$ to a prediction $p = Pred(\mathbf{r}) \in \mathbb{R}^d$. We set $Pred(\cdot)$ as multi-layer perceptron with one hidden layer.

3.3 ALIGNMENT AND UNIFORMITY: GENERALIZATION

3.3.1 GENERALIZATION OF ALIGNMENT

Wang & Isola (2020) considers a metric to measure alignment of representation as average distance between representations of positive pair as following:

$$\mathcal{L}_{\text{align}}^{\text{SELF}}(f; \alpha) = - \mathop{\mathbb{E}}_{(\mathbf{x}_i, \mathbf{x}_j) \sim p_{\text{pos}}} [\|\mathbf{z}_i - \mathbf{z}_j\|_2^\alpha], \quad \alpha > 0$$

Where p_{pos} denotes the distribution of positive pairs of data. Since we consider all data pairs to have some degree of similarity, our generalized notion of alignment can be then stated analogously as:

$$\mathcal{L}_{\text{align}}(f; \alpha) = - \mathop{\mathbb{E}}_{(\mathbf{x}_i, \mathbf{x}_j) \sim p_{\text{data}}} \left[s(\mathcal{F}(\mathbf{x}_i), \mathcal{F}(\mathbf{x}_j)) \|\mathbf{z}_i - \mathbf{z}_j\|_2^\alpha \right] \quad \alpha > 0 \tag{3}$$

where in our case $\mathbf{x}_i, \mathbf{x}_j$ are i.i.d samples from p_{data}. Note that minimizing this metric can be understood as pulling the representation vectors, with higher weight on closely-labeled data pairs.

3.3.2 UNIFORMITY: UNIVERSALLY GENERALIZED

Wang & Isola (2020) also considers uniformity as logarithm of average Gaussian potential;

$$\mathcal{L}_{\text{uniform}}(f;t) = \log \mathop{\mathbb{E}}_{(\mathbf{x}_i,\mathbf{x}_j)\sim p_{\text{data}}} \left[\exp\left(-t\|\mathbf{z}_i - \mathbf{z}_j\|_2^2\right)\right], \quad t > 0 \tag{4}$$

Since the definition does not require label information, uniformity metric can be used as is.

3.4 LASCON: LABEL-SIMILARITY AWARE SUPERVISED CONTRASTIVE LOSS

3.4.1 DERIVATION OF LASCON

Now, we are ready to build LASCon by generalizing NT-Xent loss

Recall the definition of softmax-like relative similarity :

$$u(\mathbf{x}_i, \mathbf{x}_j) = \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j/\tau\right)}{\sum_{a\in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a/\tau)}$$

Both NT-Xent loss (Equation 1) and SupCon loss (Equation 2) are defined as an expectation of log relative similarity over positive pairs. Analogously, we define LASCon as weighted summation with label similarity as weights over all samples in the minibatch.

$$\mathcal{L}^{\text{LAS}} = \mathop{\mathbb{E}}_{(\mathbf{x}_i,\mathbf{x}_j)\sim p_{\text{data}}} [-s(\mathcal{F}(\mathbf{x}_i), \mathcal{F}(\mathbf{x}_j)) \log u(\mathbf{x}_i, \mathbf{x}_j)]$$

Again, for implementation purposes, we rescale the loss function and exclude similarity with itself to match the NT-Xent and SupCon. For the minibatch, $\mathcal{L}^{\text{LAS}}$ is defined as

$$\mathcal{L}_i^{\text{LAS}} = \sum_{j\in A(i)} \tilde{s}(y_i, y_j) \log u(\mathbf{x}_i, \mathbf{x}_j), \qquad \mathcal{L}^{\text{LAS}} = -\frac{1}{|I|}\sum_{i\in I} \mathcal{L}_i^{\text{LAS}}$$

3.4.2 LASCON OPTIMIZES ALIGNMENT AND UNIFORMITY

We connect the LASCon loss function to optimization of alignment and uniformity by adapting theoretical results for NT-Xent loss (Wang & Isola, 2020) to the $\mathcal{L}^{\text{LAS}}$.

Theorem 1 (Asymptotics of $\mathcal{L}^{\text{LAS}}$)**.** *For fixed temperature $\tau > 0$, the LASCon loss converges to*

$$\begin{aligned}\lim_{|I|\to\infty} \mathcal{L}^{\text{LAS}} - \log|I| &= \mathop{\mathbb{E}}_{(\mathbf{x}_i,\mathbf{x}_j)\sim p_{\text{data}}} [-s(\mathcal{F}(\mathbf{x}_i), \mathcal{F}(\mathbf{x}_j)) \log u(\mathbf{x}_i, \mathbf{x}_j)] - \log|I| \\ &= -\frac{1}{\tau} \mathop{\mathbb{E}}_{(\mathbf{x}_i,\mathbf{x}_j)\sim p_{\text{data}}} [s(\mathcal{F}(\mathbf{x}_i), \mathcal{F}(\mathbf{x}_j))(\mathbf{z}_i \cdot \mathbf{z}_j)] + \mathop{\mathbb{E}}_{\mathbf{x}_i\sim p_{\text{data}}} \left[\log \mathop{\mathbb{E}}_{\mathbf{x}_a\sim p_{\text{data}}} [\exp(\mathbf{z}_i \cdot \mathbf{z}_a/\tau)]\right]\end{aligned}$$

as minibatch size $|I| \to \infty$. Also, the followings holds:

1. *The first term is minimized if and only if the encoder shows perfect alignment.*
2. *If distribution of $f(\mathbf{x})$ for $\mathbf{x} \sim p_{data}$ is the uniform distribution on the hypersphere, f minimizes the second term.*

We provide the complete proof of Theorem 1 in Section A.1.

3.4.3 LASCON IS A GENERALIZATION OF SUPCON AND NT-XENT

After we made the LASCon minimizing generalized definition of alignment and uniformity, LASCon becomes a generalization of SupCon and NT-Xent.

LASCon is a generalization of SupCon For classification tasks, $\mathbb{1}_{y_i=y_j}$ can be used as similarity function s. SupCon then can be induced from LASCon loss where $P(i) = \{j \in A(i) | y_i = y_j\}$:

$$\begin{aligned}\mathcal{L}_i^{\text{LAS}} &= -\sum_{j\in A(i)} \frac{s(y_i, y_j)}{\sum_{a\in A(i)} s(y_i, y_a)} \log u(\mathbf{x}_i, \mathbf{x}_j) \\ &= -\sum_{j\in A(i)} \frac{\mathbb{1}_{y_i=y_j}}{\sum_{a\in A(i)} \mathbb{1}_{y_i=y_a}} \log u(\mathbf{x}_i, \mathbf{x}_j) = -\frac{1}{|P(i)|}\sum_{p\in P(i)} \log u(\mathbf{x}_i, \mathbf{x}_p) = \mathcal{L}_i^{\text{SUP}}\end{aligned}$$

Table 1: Summary of downstream tasks.

Field	Dataset	Label Range	Threshold	Meaning of Labels	Encoder
NLP	IMDB	[1, 10]	7	Movie Review Rating	BERT-base
Vision	FreiHAND	-	-	3D Coordinate of Hand Landmark	ResNet-152
Graph	Davis	[5, 11]	7	Binding Affinity of Drug Molecule	GIN + 1DCNN
Multimodal	CMU-MOSI	[-3, 3]	0	Sentiment of Speech Video	2 LSTM + BERT-base

LASCon is a generalization of NT-Xent In self-supervised setting, we only know that augmented sample has same label to original label; similarity function $s(y_i, y_k)$ is given $\mathbb{1}_{k=j(i)}$.

$$\begin{aligned}\mathcal{L}_i^{\text{LAS}} &= = -\sum_{k\in A(i)} \frac{s(y_i,y_k)}{\sum_{a\in A(i)} s(y_i,y_a)} \log u(\mathbf{x}_i,\mathbf{x}_k) \\ &= -\sum_{k\in A(i)} \frac{\mathbb{1}_{k=j(i)}}{\sum_{a\in A(i)} \mathbb{1}_{a=j(i)}} \log u(\mathbf{x}_i,\mathbf{x}_k) = -\log u(\mathbf{x}_i,\mathbf{x}_{j(i)}) = \mathcal{L}_i^{\text{SELF}}\end{aligned}$$

Here, $\mathbb{1}_{\mathbf{x}_i=\mathbf{x}_{j(i)}}$ can be regarded as discretization of similarity function.

3.4.4 Another Generalization of SupCon, In and Out Version of LASCon

As in Khosla et al. (2020) and Anonymous (2022), generalization can be done in multiple ways; analogously to SupCon, one can consider two versions of LASCon depending on whether label similarity is weighted before or after taking logarithm.

$$\mathcal{L}_{in}^{\text{LAS}} = -\frac{1}{|I|}\sum_{i\in I} \log\left\{\sum_{j\in A(i)} \tilde{s}(y_i,y_j)u(\mathbf{x}_i,\mathbf{x}_j)\right\}$$

$$\mathcal{L}_{out}^{\text{LAS}} = -\frac{1}{|I|}\sum_{i\in I}\sum_{j\in A(i)} \tilde{s}(y_i,y_j)\log u(\mathbf{x}_i,\mathbf{x}_j).$$

We mainly discussed *out* version due to mainly two reasons. First, in SupCon loss function, authors have shown the superiority of *out* version both empirically and by gradient computation. This analysis is also done for regression tasks in Anonymous (2022). Another reason is that *out* version is amenable to adaptation of rich theoretical results and proofs done for NT-Xent loss, which we discuss in Theorem 1 and the Section A.1.
Our experimental results are consistent with such previous considerations, *out* version showing better performance in various tasks spanning multiple domains. (Table 2) We also provide the computation of gradient and comparison in Section A.2.

4 Method

4.1 Task Definition

We evaluate the efficacy of LASCon Loss on four downstream tasks each representing four pillars of machine learning - NLP, Vision, Graph, and Multimodal. We provide a brief summary about datasets in Table 1. Additional data statistics and implementation details are provided in Section C, and sections specified at the end of the descriptions.

NLP Task - *IMDB* Semantic analysis is one of the most important challenge in NLP. As a representative task addressing text semantics, IMDB aims to predict the numeric review score using the comment. Maas et al. (2011) collected 50,000 movie reviews and scores ranging [1,10] and 0.5 as a minimum unit. Following Yang et al. (2019), we use the even split having 25,000 labeled samples for training and test and set 7 as a threshold for SupCon. We test four augmentations following Wei & Zou (2019), and text input is encoded via BERT-base (Devlin et al., 2018). (Section D.2)

Vision Task - *FreiHAND* 3D pose estimation is crucial for solving the 3D semantic segmentation problem in Computer Vision. FreiHAND focuses on the hand, which is challenging since the hand is small but complicated. Zimmermann et al. (2019) annotated 224×224 RGB-formatted hand images to designate coordinates of 21 keypoints correspond joints and tips of fingers and wrist. Therefore

label in FreiHAND is vector $y \in \mathbb{R}^{21\times 3}$; a special similarity function is required (Equation 6).We could not test SupCon since no standard classification criterion is proposed. Total 134,200 images are splitted into 130,240 training and 3,960 evaluation set. We apply five augmentations together following Spurr et al. (2021), and use ResNet-152 (He et al., 2016) for encoder (Section D.3).

Graph Task - *Davis* Drug efficacy prediction is the urgent problem in health/clinical machine learning. Davis is a widely-used dataset containing interaction intensity of drugs with specific protein family, kinase. Davis et al. (2011) consists 25,772 pairs of drug-protein interaction intensity, with the range [5, 11] after log-scaling; smaller means stronger binding. Following Huang et al. (2021), we set 7 as a threshold for SupCon. We adopt the GraphDTA framework (Nguyen et al., 2021) using Graph Isomorphism Network (Xu et al., 2019) for drug molecules and 1DCNN for protein sequences. Three graph augmentation methods (You et al., 2020) are assessed. (Section D.4)

Multimodal Task - *CMU-MOSI* Understanding human communication is a representative multidisciplinary task since a model gets a collection of text (NLP), video (Vision), and audio input data. CMU-MOSI (CMU-Multimodal Corpus of Sentiment Intensity) aims to predict the sentiment of speaker from the speech video. Zadeh et al. (2016) collected 2,199 video clips and manually annotated the sentiment of each utternace video within the range [-3, 3]. Following Han et al. (2021); Yu et al. (2021) we use BERT (Devlin et al., 2018) for text and two unidirectional LSTMs (Hochreiter & Schmidhuber, 1997) for audio and visual input (Section D.5).

4.2 Exploring Representation Space

Quantitative Analysis We implement alignment (3) and uniformity (4) in minibatch as:

$$\mathcal{L}_{\text{align}} = \frac{1}{|I|}\sum_{i\in I}\sum_{j\in A(i)} \tilde{s}(y_i, y_j)\|\mathbf{z}_i - \mathbf{z}_j\|_2^2$$

$$\mathcal{L}_{\text{uniform}} = \frac{1}{2|I|(|I|-1)}\sum_{i\in I}\sum_{j\in A(i)} \exp\Big(-2\|\mathbf{z}_i - \mathbf{z}_j\|_2^2\Big)$$

Where the normalization of similarity and rescaling for implementation purposes were made. Note that we flip the sign of resulting alignment and uniformity to compare in positive range.

Qualitative Analysis Circular graphs approximately describe distributions of representation spaces. The hue of a point indicates the label of corresponding datapoint, and the saturation means the estimated density at the point. The dimension of representation is reduced by principal component analysis (PCA), and density is estimated via Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$.

4.3 Label Similarity Function

One significant advantage of our loss function is optimizability on diverse tasks. Users have freedom to choose arbitrary label similarity function α which can better capture properties of specific data types or further fit to their purpose. Since our NLP, Graph, Multimodal tasks have one dimensional real-valued labels, we define similarity function $s(y_i, y_j)$ as follows. First, we are normalize $\tilde{y_i} = \frac{y_i}{M}$ for $M = \max_{i\in I}|y_i|$. Then, for $y_i, y_j \in \mathcal{Y}$ and an activation function $\alpha \in \{id, \tanh\}$,

$$s(y_i, y_j) = 1 - \alpha\left(c \cdot \frac{|\tilde{y_i} - \tilde{y_j}|}{2}\right) \qquad c \in \mathbb{R} \tag{5}$$

However, for vision task FreiHAND, since labels are in $\mathbb{R}^{21\times 3}$ we used L_1 distance between labels to define similarity. For $M = \max_{i,j\in I}\|y_i - y_j\|_1$ and $m = \min_{i,j\in I}\|y_i - y_j\|_1$,

$$s(y_i, y_j) = 1 - \alpha\left(c \cdot \frac{\|y_i - y_j\|_1 - m}{M - m}\right) \qquad c \in \mathbb{R} \tag{6}$$

Here, we extend linear implementation($\alpha = id(\cdot)$) by applying $\alpha = \tanh(\cdot)$ as a representative non-linear label similarity function. Furthermore, we propose a hyperparameter c in Equation 5 and Equation 6 that controls steepness of label similarity function near zero (Figure 13). Underlying logic on this setting is that larger gradient around zero may encourage model to become more sensitive on more hard samples having similar labels. Unlike classification tasks, Extensive experiments were conducted with varying α and c.

Table 2: Results with linear label similarity.

Tasks		NLP (IMDB)		Vision (FreiHAND)			Graph (Davis)		Multimodal (CMU-MOSI)		
Models	in/out	MAE(↓)	Corr(↑)	3D EPE(↓)	2D EPE(↓)	AUC(↑)	MSE(↓)	Corr(↑)	MAE(↓)	Corr(↑)	Acc-7(↑)
SimCLR		1.282	0.868	0.1012	25.9564	0.7955	0.299	0.795	0.707	0.800	46.550
SupCon	in	1.144	0.888	-	-	-	0.307	0.780	0.707	0.805	45.627
	out	1.134	0.889	-	-	-	0.287	0.777	0.704	0.804	46.696
Ours	in	1.191	0.878	0.0784	18.6674	0.8416	0.308	0.786	0.708	0.801	46.113
	out	1.110	0.895	0.0726	18.4676	0.8533	0.276	0.811	0.703	0.802	46.939

Table 3: Quantitative analysis on generalized alignment and uniformity.

Tasks		NLP (IMDB)		Vision (FreiHAND)		Graph (Davis)		Multimodal (CMU-MOSI)	
Models	in/out	Align(↓)	Uniform(↓)	Align(↓)	Uniform(↓)	Align(↓)	Uniform(↓)	Align(↓)	Uniform(↓)
SimCLR		1.457	2.406	0.522	10.227	4.398	2.742	1.598	1.403
SupCon	in	0.444	0.703	-	-	2.000	1.050	1.235	1.286
	out	0.460	0.811	-	-	1.701	1.170	0.133	0.275
Ours	in	0.851	1.484	0.136	4.984	3.008	1.492	1.577	0.863
	out	0.312	0.575	0.132	2.618	2.639	1.425	0.104	0.221

5 RESULTS

5.1 RESULTS WITH NAIVE LINEAR LABEL SIMILARITY

Label-similarity aware learning achieves SOTA performances. We provide performances using linear label similarity, which is $\alpha(x) = x$ and $c = 1$, in Table 2. The model trained with LASCon showed significant performance improvement in all datasets. Specifically, LASCon achieved 13.42 %, 7.27 %, and 7.69 % performance improvement for NLP, Vision, and Graph tasks using previous SimCLR approach. Even comparing with SupCon, LASCon recorded 2.16 % and 3.99 % enhancement in NLP and Graph tasks. We would also highlight that Vision/Graph/Multimodal tasks which have previous SOTA method on the same dataset, our naive implementation using linear label similarity showed better performance. This result is noteworthy since our framework is *universal*, which can be further optimized for specific tasks and data formats.

Great Alignment/Uniformity make Great Downstream Performances Table 3 and Figure 2 empirically and visually show that LASCon transforms an embedding space to better reflects genuine similarity between labels and it leads performance improvement. Embedding space generated via LASCon showed the lowest alignment and uniformity, which means that samples with similar labels are nearly located, while all embedding vectors are distributed. Correlation graphs in Figure 3 (a,b) implies that measured alignment and uniformity directly affects downstream performances; ***better the embedding space constructed, better the downstream performance***. One exception is Graph-Davis; we hypothesized that highly biased label distribution (60 % samples belongs to the same label 5.0; Figure 8) made extremely low alignment and uniformity after SupCon. Comparing circular graphs of SimCLR with others in Figure 2(a), SupCon and LASCon showed much clear decision boundary while SimCLR locate some positive vectors *in between* negative samples. LASCon, however, shows much smooth distribution of embedding distributions than SupCon; SupCon indeed locate some extreme representations (colored with red or blue) near the decision boundary. Figure 2(c) shows biased distribution; again, highly biased label distribution seems to cause locality.

5.2 STRONG-POSITIVE SAMPLING VIA NON-LINEAR LABEL SIMILARITY

Table 4 empirically proves that introducing a non-linear label similarity function can further boost downstream task performances. Introducing non-linear *not always* enhance performance; using $c = 0.5$ rather hampers model performance. However, the non-linear similarity function with $c = 2.0$ tends to boost the model to learn beneficial relationships between samples. We interpret these results by introducing the concept of *strong-positive sampling*. Recall that real-world problems often require continuous decision boundaries. Unlike classification representation learning which is sufficient to regard only the macro relationship between the distinct label, our target problem necessitates focusing on micro relationships between similar samples. To this end, exaggerating the gradient near perfect similarity could encourage the model to become more sensitive to strong positives having similar labels. Empirically, better performance was observed with large gradients. Comprehensive and task-specific optimization of c and its relationship with temperature will be an interesting future research topic.

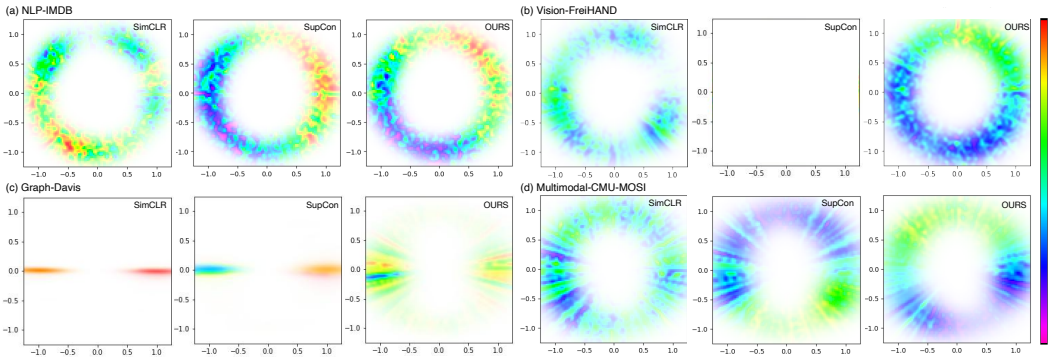

Figure 2: Circular distribution of representation spaces, depending on contrastive loss function. Hue indicates label, and saturation means estimated density.

Table 4: Results with non-linear label similarity.

Tasks		NLP (IMDB)		Vision (FreiHAND)			Graph (Davis)		Multimodal (CMU-MOSI)		
Sim	Coef	MAE(↓)	Corr(↑)	3D EPE(↓)	2D EPE(↓)	AUC(↑)	MSE(↓)	Corr(↑)	MAE(↓)	Corr(↑)	Acc-7(↑)
Linear	-	1.110	0.895	0.0726	18.4676	0.8533	0.276	0.811	0.703	0.802	46.550
Tanh	0.5	1.319	0.857	0.0769	19.9356	0.8618	0.278	0.811	0.718	0.798	45.384
	1.0	1.183	0.887	0.0690	17.5392	0.8606	0.274	0.812	0.709	0.803	45.384
	2.0	1.097	0.893	0.0684	17.4586	0.8618	0.277	0.811 2	0.705	0.805	45.870

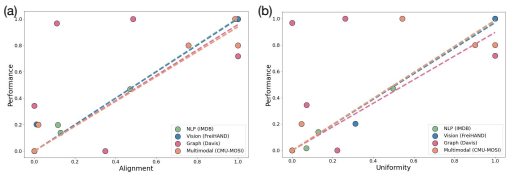

Figure 3: Correlation with performances

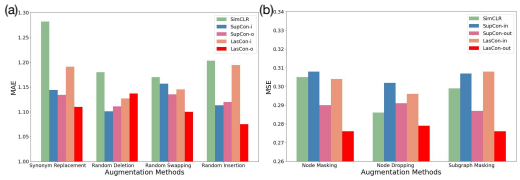

Figure 4: Ablation study on augmentations

5.3 Ablation Study: Robustness on Augmentation

We show the performance change depending on augmentation strategies; Figure 4(a) for NLP-IMDB and 4(b) for Graph-Davis. LASCon loss with *out* configuration shows the best performance invariant to augmentation strategy. One exception is a random deletion in NLP task; we hypothesized that random deletion may introduce critical modification significantly changes the meaning significantly. Investigating and optimizing the effects of augmentation would also be a interesting future work.

6 Conclusion

In this study, we analyze LASCon, a label similarity aware contrastive learning framework, in order to address real-world problems having *relative* similarity between samples. We combine previous approaches and re-interpret LASCon using label similarity-aware alignment and uniformity. Our quantitative analysis of representation space shows a general correlation that great alignment/uniformity makes great performance; we also found that this relationship may be disturbed by extremely biased label distribution. Through comprehensive analysis of four main fields in machine learning utilizing diverse architectures and data formats, LASCon is proved suitable for real-world problems. Our non-linear label similarity results show the potential of LASCon to be optimized for all types of tasks the user wants. Extending our results to the semi-supervised setting where unlabeled data can be used together with labeled data to build richer representation space seems to be an important direction with growing applications. Also, rigorous theoretical analysis of the convergence rate of our generalized contrastive loss and the usage of various model architectures remains an interesting future work.

ETHICS STATEMENT

We strongly believe that building better representation space by contrastive learning and more specifically LASCon can be used to improve various domains and applications. Such applications should be carefully monitored to ensure it's ethical and fair usage.

Our experiments were conducted on some publically available datasets which were collected from human subjects. FreiHAND dataset consists of images of human hand postures, and CMU-MOSI dataset consists of speech video. While these are widely used and thus are routinely checked by the community, it is possible that datasets of human origin contains unintentional biases and stereotypes. Consistent and extensive monitoring, following in depth discussion toward ethical usage of large datasets and representation learning methodologies by the community is indispensable.

REPRODUCIBILITY STATEMENT

We believe that our community can become healthier and more productive when all researchers comply with *Code of Ethics*. To this end, we provide transparent and comprehensive results of our results, including the ones which were not thoroughly discussed in main manuscript due to unsatisfactory performance. We also attached the code used for the experiment on the FreiHAND dataset as the supplementary material, for the clear demonstration of concept and reproducibility. The collection of full code for all experiments will also be made publically available shortly.

REFERENCES

Anonymous. Supervised Contrastive Learning for Multimodal Sentiment Analysis with Continuous Labels. In *Omitted due to ongoing review process*, 2022.

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.

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

Ting Chen, Calvin Luo, and Lala Li. Intriguing Properties of Contrastive Losses. *Advances in Neural Information Processing Systems*, 34:11834–11845, 2021.

Mindy I Davis, Jeremy P Hunt, Sanna Herrgard, Pietro Ciceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Daniel K Treiber, and Patrick P Zarrinkar. Comprehensive Analysis of Kinase Inhibitor Selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Benoit Dufumier, Pietro Gori, Antoine Grigis, and Edouard Duchesnay. Conditional Alignment and Uniformity for Contrastive Learning with Continuous Proxy Labels. In *Medical Imaging Meets NeurIPS Workshop*, 2021a.

Benoit Dufumier, Pietro Gori, Julie Victor, Antoine Grigis, Michele Wessa, Paolo Brambilla, Pauline Favre, Mircea Polosan, Colm McDonald, Camille Marie Piguet, Mary Phillips, Lisa Eyler, and Edouard Duchesnay. Contrastive Learning with Continuous Proxy Meta-data for 3D MRI Classification. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, 2021b.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

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.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *International Conference on Learning Representations*, 2020.

Wei Han, Hui Chen, and Soujanya Poria. Improving Multimodal Fusion with Hierarchical Mutual Information Maximization for Multimodal Sentiment Analysis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9180–9192. Association for Computational Linguistics, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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*, pp. 9729–9738, 2020.

Chih-Hui Ho and Nuno Nvasconcelos. Contrastive Learning with Adversarial Examples. *Advances in Neural Information Processing Systems*, 33:17081–17093, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf H Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard Negative Mixing for Contrastive Learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

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

Yang Li, Guanyu Qiao, Xin Gao, and Guohua Wang. Supervised graph co-contrastive learning for drug–target interaction prediction. *Bioinformatics*, 38(10):2847–2854, 2022.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training Molecular Graph Representation with 3D Geometry. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Sijie Mai, Ying Zeng, Shuangjia Zheng, and Haifeng Hu. Hybrid Contrastive Learning of Tri-modal Representation for Multimodal Sentiment Analysis. *IEEE Transactions on Affective Computing*, 2022.

Leland McInnes, John Healy, and James Melville. Umap: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. GraphDTA: Predicting Drug-target Binding Affinity with Graph Neural Networks. *Bioinformatics*, 37(8):1140–1147, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. *Advances in neural information processing systems*, 29, 2016.

Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11210–11219. IEEE, 2021.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to Fine-tune BERT for Text Classification? In *China national conference on Chinese computational linguistics*, pp. 194–206. Springer, 2019.

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.

Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised Learning with Data Augmentations Provably Isolates Content from Style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

Feng Wang and Huaping Liu. Understanding the Behaviour of Contrastive Loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2495–2504, 2021.

Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Yaoming Wang, Yangzhou Jiang, Jin Li, Bingbing Ni, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Teng Li. Contrastive Regression for Domain Adaptation on Gaze Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19376–19385, 2022.

Jason Wei and Kai Zou. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, 2019.

Kilian Q Weinberger, John Blitzer, and Lawrence Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Advances in neural information processing systems*, 18, 2005.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful Are Graph Neural Networks? In *International Conference on Machine Learning*. PMLR, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNET: Generalized Autoregressive Pretraining for Language Understanding. *Advances in neural information processing systems*, 32, 2019.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph Contrastive Learning with Augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

Wenmeng Yu, Hua Xu, Ziqi Yuan, and Jiele Wu. Learning Modality-Specific Representations with Self-Supervised Multi-Task Learning for Multimodal Sentiment Analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10790–10797, 2021.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. MOSI: Multimodal Corpus of Sentiment Intensity and Subjectivity Analysis in Online Opinion Videos. *arXiv preprint arXiv:1606.06259*, 2016.

Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 813–822, 2019.

A MORE ON LOSS FUNCTION

Throughout the section, we consider $\mathbb{1}$ as the indicator function, which takes the value of 1 or 0 depending on the given condition.

A.1 PROOF OF THEOREM 1

Proof. Since the results are mostly adaptation of the results proven on NT-Xent loss to our $\mathcal{L}^{\text{LAS}}$, most of the proofs can be also applied without much changes; Interested readers can refer to the Wang & Isola (2020) for more detailed proof.

Recall that our loss function is defined as

$$\begin{aligned}
\mathcal{L}^{\text{LAS}} &= -\frac{1}{|I|}\sum_{i\in I}\sum_{j\in A(i)}\tilde{s}(y_i,y_j)\log u(\mathbf{x}_i,\mathbf{x}_j) \\
&= -\frac{1}{|I|}\sum_{i\in I}\sum_{j\in A(i)}\tilde{s}(y_i,y_j)\log\frac{\exp(\mathbf{z}_i\cdot\mathbf{z}_j/\tau)}{\sum_{a\in A(i)}\exp(\mathbf{z}_i\cdot\mathbf{z}_a/\tau)} \\
&= -\frac{1}{|I|}\sum_{i\in I}\sum_{j\in A(i)}\tilde{s}(y_i,y_j)(\mathbf{z}_i\cdot\mathbf{z}_j/\tau) + \frac{1}{|I|}\sum_{i\in I}\sum_{j\in A(i)}\tilde{s}(y_i,y_j)\log\sum_{a\in A(i)}\exp(\mathbf{z}_i\cdot\mathbf{z}_a/\tau) \\
&= -\frac{1}{|I|}\sum_{i\in I}\sum_{j\in A(i)}\tilde{s}(y_i,y_j)(\mathbf{z}_i\cdot\mathbf{z}_j/\tau) + \frac{1}{|I|}\sum_{i\in I}\log\sum_{a\in A(i)}\exp(\mathbf{z}_i\cdot\mathbf{z}_a/\tau)
\end{aligned}$$

Since sum of $\tilde{s}$ over $A(i)$ is 1 for each $i \in I$.

The asymptotics for the first term can be obtained as:

$$\lim_{|I|\to\infty} -\frac{1}{\tau|I|}\sum_{i\in I}\sum_{j\in A(i)}\tilde{s}(y_i,y_j)(\mathbf{z}_i\cdot\mathbf{z}_j) \to -\frac{1}{\tau}\mathop{\mathbb{E}}_{(\mathbf{x}_i,\mathbf{x}_j)\sim p_{\text{data}}}[s(\mathcal{F}(\mathbf{x}_i),\mathcal{F}(\mathbf{x}_j))(\mathbf{z}_i\cdot\mathbf{z}_j)]$$

by the strong law of large numbers. Note that excluding the term of z_i with itself does not change the asymptotic result as batch size grows to infinity.

For the second term, we note that for each $i \in I$, we have

$$\lim_{|I|\to\infty}\log\left(\frac{1}{|I|}\sum_{a\in A(i)}\exp(\mathbf{z}_i\cdot\mathbf{z}_a/\tau)\right) = \log\mathop{\mathbb{E}}_{\mathbf{x}_a\sim p_{\text{data}}}[\exp(\mathbf{z}_i\cdot\mathbf{z}_a/\tau)] \qquad \text{a. s.}$$

by the Strong Law of Large Numbers and the Continuous Mapping Theorem. Again, while the exact number of sample $|A(i)| = |I| - 1$, the estimator remains asymptotically unbiased. By the boundedness of $e^{z_i\cdot z_a/\tau}$ on the hypersphere and the Lebesgue's Dominated Convergence Theorem, we obtain the same result as in NT-Xent loss.

$$\lim_{|I|\to\infty}\frac{1}{|I|}\sum_{i\in I}\log\sum_{a\in A(i)}\exp(\mathbf{z}_i\cdot\mathbf{z}_a/\tau) \to \mathop{\mathbb{E}}_{\mathbf{x}_i\sim p_{\text{data}}}\left[\log\mathop{\mathbb{E}}_{\mathbf{x}_a\sim p_{\text{data}}}\exp(\mathbf{z}_i\cdot\mathbf{z}_a/\tau)\right] \qquad \text{a. s.}$$

First term is minimized by and only by the perfectly aligning encoder : For $\mathbf{z}_i, \mathbf{z}_j \in \mathcal{S}^d$,

$$-s(\mathcal{F}(\mathbf{x}_i),\mathcal{F}(\mathbf{x}_j))(\mathbf{z}_i\cdot\mathbf{z}_j) = s(\mathcal{F}(\mathbf{x}_i),\mathcal{F}(\mathbf{x}_j))(\|\mathbf{z}_i-\mathbf{z}_j\|_2^2/2-1)$$

Hence a perfectly aligning encoder, which gives $\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 = 0$ almost surely, minimizes the first term. Existence of such encoder gives the converse.

Second term is minimized by perfectly uniform encoder : Since the second term remained unmodified, the proof from Wang & Isola (2020) (Appendix A.2) can be directly applied.

□

A.2 GRADIENTS OF LOSS

Here, as in Anonymous (2022), we derive the gradients and prove that arguments about $\mathcal{L}_{out}^{\text{SUP}}$ and $\mathcal{L}_{in}^{\text{SUP}}$ remains within our generalization. For simplicity, let $C = \sum_{a \in A(i)} s(y_i, y_a) \in \mathbb{R}$.

A.2.1 LASCON: IN

We first compute the gradient of $\mathcal{L}_{in}^{\text{LAS}}$.

$$\begin{aligned}
\frac{\partial \mathcal{L}_{in}^{\text{LAS}}}{\partial \mathbf{z}_i} &= -\frac{\partial}{\partial \mathbf{z}_i} \log \left(\frac{1}{C} \sum_{a \in A(i)} s(y_i, y_a) u(\mathbf{x}_i, \mathbf{x}_a) \right) \\
&= -\frac{\partial}{\partial \mathbf{z}_i} \log \sum_{a \in A(i)} s(y_i, y_a) u(\mathbf{x}_i, \mathbf{x}_a) \\
&= -\frac{\partial}{\partial \mathbf{z}_i} \log \frac{\sum_{a \in A(i)} s(y_i, y_a) \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \\
&= -\frac{\partial}{\partial \mathbf{z}_i} \log \sum_{a \in A(i)} s(y_i, y_a) \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau) + \frac{\partial}{\partial \mathbf{z}_i} \log \sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau) \\
&= -\frac{1}{\tau} \frac{\sum_{a \in A(i)} \mathbf{z}_a s(y_i, y_a) \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)}{\sum_{a \in A(i)} s(y_i, y_a) \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} + \frac{1}{\tau} \frac{\sum_{a \in A(i)} \mathbf{z}_a \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \\
&= \frac{1}{\tau} \left(\sum_{a \in A(i)} \mathbf{z}_a \left(u(\mathbf{x}_i, \mathbf{x}_a) - X^{in}(\mathbf{x}_i, \mathbf{x}_a) \right) \right)
\end{aligned}$$

where $X^{in}(\mathbf{x}_i, \mathbf{x}_a)$ is defined as

$$X^{in}(\mathbf{x}_i, \mathbf{x}_a) = \frac{s(y_i, y_a) \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)}{\sum_{j \in A(i)} s(y_i, y_j) \exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}$$

A.2.2 LASCON: OUT

We then compute the gradient of $\mathcal{L}_{out}^{\text{LAS}}$.

$$\begin{aligned}
\frac{\partial \mathcal{L}_{out}^{\text{LAS}}}{\partial \mathbf{z}_i} &= -\frac{1}{C} \frac{\partial}{\partial \mathbf{z}_i} \sum_{a \in A(i)} s(y_i, y_a) \log u(\mathbf{x}_i, \mathbf{x}_a) \\
&= \frac{1}{C} \frac{\partial}{\partial \mathbf{z}_i} \left(- \sum_{a \in A(i)} s(y_i, y_a) \frac{\mathbf{z}_i \cdot \mathbf{z}_a}{\tau} + \sum_{j \in A(i)} s(y_i, y_j) \log \sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau) \right) \\
&= \frac{1}{C\tau} \left(- \sum_{a \in A(i)} s(y_i, y_a) \mathbf{z}_a + \sum_{j \in A(i)} s(y_i, y_j) \frac{\sum_{a \in A(i)} \mathbf{z}_a \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \right) \\
&= \frac{1}{\tau} \left(\sum_{a \in A(i)} \mathbf{z}_a \left(u(\mathbf{x}_i, \mathbf{x}_a) - X^{out}(\mathbf{x}_i, \mathbf{x}_a) \right) \right)
\end{aligned}$$

where $X^{out}(\mathbf{x}_i, \mathbf{x}_a)$ is defined as

$$X^{out}(\mathbf{x}_i, \mathbf{x}_a) = \frac{s(y_i, y_a)}{\sum_{j \in A(i)} s(y_i, y_j)}$$

We notice that as in Khosla et al. (2020), taking $\overline{z}$ as **weighted** average of samples reduces $\mathcal{L}_{in}^{\text{LAS}}$ to $\mathcal{L}_{out}^{\text{LAS}}$, thus stabilization argument is applicable.

B ADDITIONAL RESULTS

In this section, we provide additional u-map visualization (McInnes et al., 2018) of embedding spaces and summarized results.

B.1 UMAP VISUALIZATION OF EMBEDDING SPACE

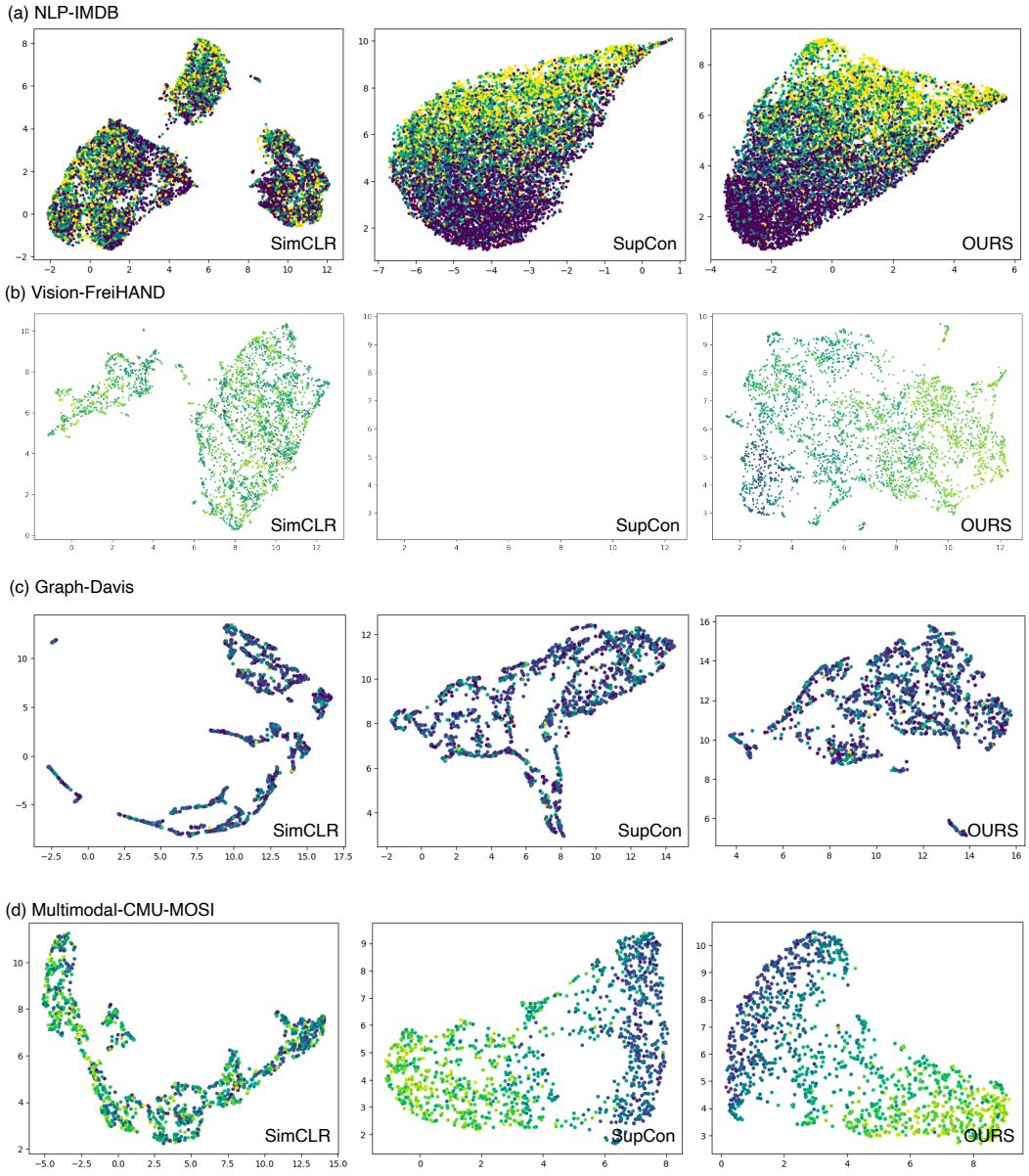

Figure 5: Umap visualization of embedding space.

B.2 SUMMARIZED RESULTS OVER TEMPERATURE.

Table 5: NLP contrastive learning on CMU-MOSI with diverse loss functions. We use the best performances on each experimental setting over temperature.

	Synonym Replacement				Random Deletion			
Models	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
SimCLR	1.282	0.868	92.496	92.494	1.180	0.882	90.903	90.873
SupCon-i	1.144	0.888	91.773	91.757	1.101	0.893	91.343	91.307
LASCon-iT2	1.145	0.888	90.784	90.722	1.109	0.890	91.231	91.199
LASCon-iT1	1.131	0.891	91.987	91.970	1.127	0.885	92.501	92.487
LASCon-iT0.5	1.147	0.836	91.936	91.918	1.175	0.888	92.683	92.676
LASCon-iL	1.191	0.878	91.375	91.345	1.127	0.892	90.904	90.853
SupCon-o	1.134	0.889	91.871	91.853	1.111	0.893	90.939	90.895
LASCon-oT2	1.097	0.893	90.239	90.173	1.100	0.895	91.964	91.939
LASCon-oT1	1.183	0.887	90.196	90.121	1.147	0.887	90.892	90.847
LASCon-oT0.5	1.319	0.857	89.849	89.808	1.297	0.860	91.132	91.119
LASCon-oL	1.110	0.895	92.976	92.973	1.137	0.892	92.049	92.025
	Random Swapping				Random Insertion			
Models	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
SimCLR	1.170	0.879	91.421	91.403	1.203	0.875	90.455	90.411
SupCon-i	1.157	0.890	91.893	91.874	1.113	0.890	92.853	92.848
LASCon-iT2	1.127	0.894	92.275	92.259	1.108	0.893	91.239	91.209
LASCon-iT1	1.155	0.886	92.388	92.380	1.130	0.890	91.876	91.854
LASCon-iT0.5	1.115	0.893	91.931	91.910	1.149	0.890	90.647	90.588
LASCon-iL	1.145	0.884	91.612	91.582	1.194	0.887	91.531	91.499
SupCon-o	1.135	0.890	92.423	92.412	1.120	0.891	92.035	92.011
LASCon-oT2	1.108	0.892	92.028	92.006	1.155	0.886	90.561	90.518
LASCon-oT1	1.139	0.887	90.480	90.430	1.178	0.883	90.476	90.412
LASCon-oT0.5	1.244	0.870	91.644	91.636	1.260	0.870	92.075	92.071
LASCon-oL	1.100	0.896	92.548	92.539	1.075	0.893	90.601	90.553

Table 6: Graph contrastive learning on Davis with diverse loss functions. We use the best performances on each experimental setting over temperature.

	Node Masking			Node Dropping			Subgraph Masking		
Models	MSE(↓)	Corr(↑)	Acc-2(↑)	MSE(↓)	Corr(↑)	Acc-2(↑)	MSE(↓)	Corr(↑)	Acc-2(↑)
SimCLR	0.305	0.789	0.942	0.286	0.806	0.943	0.299	0.795	0.942
SupCon-i	0.308	0.787	0.940	0.302	0.793	0.941	0.307	0.788	0.942
LASCon-iT2	0.298	0.794	0.941	0.305	0.790	0.942	0.305	0.791	0.941
LASCon-iT1	0.302	0.790	0.943	0.296	0.796	0.941	0.300	0.793	0.943
LASCon-iT0.5	0.303	0.791	0.942	0.297	0.797	0.943	0.299	0.794	0.941
LASCon-iL	0.304	0.791	0.942	0.296	0.796	0.942	0.308	0.786	0.941
SupCon-o	0.290	0.801	0.944	0.291	0.801	0.944	0.287	0.802	0.943
LASCon-oT2	0.279	0.809	0.944	0.280	0.810	0.944	0.277	0.811	0.944
LASCon-oT1	0.275	0.813	0.945	0.276	0.811	0.945	0.274	0.812	0.944
LASCon-oT0.5	0.277	0.811	0.945	0.275	0.813	0.943	0.278	0.811	0.944
LASCon-oL	0.276	0.812	0.944	0.279	0.811	0.944	0.276	0.811	0.943

Table 7: Vision contrastive learning on FreiHAND with diverse loss functions. We use the best performances on each experimental setting over temperature.

	FreiHAND				
Models	3D EPE(↓)	2D EPE(↓)	AUC(↑)	Alignment(↓)	Uniformity(↓)
SimCLR	0.1012	25.9564	0.7955	0.5217	10.2271
PeCLR	0.0842	19.6219	0.8299	0.1664	9.4734
LASCon-iT2	0.0754	18.3845	0.8477	0.1248	5.8775
LASCon-iT1	0.0809	19.3998	0.8366	0.1255	4.8791
LASCon-iT0.5	0.0871	20.5561	0.8240	0.1752	5.0987
LASCon-iL	0.0784	18.6674	0.8416	0.1365	4.9839
LASCon-oT3	0.0659	16.4900	0.8668	0.1521	7.6807
LASCon-oT2	0.0684	17.4586	0.8618	0.1902	4.1051
LASCon-oT2 $\tau = 0.1$	0.0748	24.5997	0.8488	0.1044	10.6777
LASCon-oT2 $\tau = 1.0$	0.0683	17.7334	0.8619	0.1985	3.0022
LASCon-oT1	0.0690	17.5392	0.8606	0.0768	2.0115
LASCon-oT0.5	0.0769	19.9356	0.8446	0.0192	1.4438
LASCon-oL	0.0726	18.4676	0.8533	0.1321	2.6182

Table 8: Multimodal contrastive learning on CMU-MOSI with diverse loss functions. We use the best performances on each experimental setting over temperature.

	CMU-MOSI							
Models	MAE(↓)	Corr(↑)	Acc-7(↑)	Acc-5(↑)	Acc-2(↑)	Acc-2(↑)	F1(↑)	F1(↑)
baseline	0.722	0.797	43.246	49.223	82.556	84.654	82.479	84.644
NTXent	0.707	0.800	46.550	52.964	82.410	84.197	82.321	84.167
SupCon-i	0.707	0.805	45.627	51.603	83.625	84.909	83.622	84.948
LASCon-iT2	0.703	0.805	45.287	51.603	82.896	84.604	82.814	84.576
LASCon-iT1	0.709	0.803	44.801	50.777	82.507	84.146	82.431	84.125
LASCon-iT0.5	0.706	0.804	45.287	51.555	83.139	85.163	83.080	85.165
LASCon-iL	0.708	0.801	46.113	52.527	83.188	84.909	83.131	84.905
SupCon-o	0.704	0.804	46.696	52.915	82.556	84.451	82.476	84.433
LASCon-oT2	0.705	0.805	45.870	51.798	82.556	84.197	82.505	84.200
LASCon-oT1	0.709	0.803	45.384	51.361	82.896	84.400	82.856	84.409
LASCon-oT0.5	0.718	0.798	45.384	51.215	82.313	83.740	82.266	83.742
LASCon-oL	0.703	0.802	46.939	52.575	82.604	84.248	82.572	84.270

C DATASETS

C.1 NLP-IMDB

The label of IMDB ranges from [1, 10], 0.5 as a minum unit. We provide the label distribution in Figure 6.

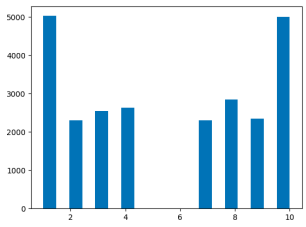

Figure 6: Label distribution of IMDB dataset.

C.2 VISION-FREIHAND

Since FreiHAND comprises 3D image, we provide the label distribution on three respective cartesian axises in Figure 7.

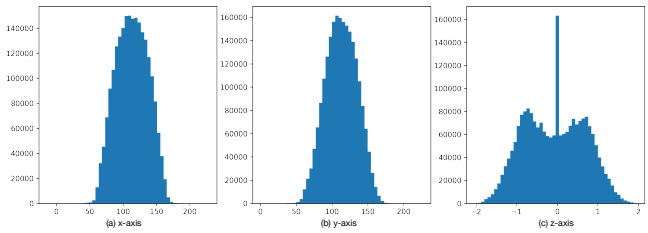

Figure 7: Label distribution of FreiHAND dataset.

C.3 GRAPH-DAVIS

We illustrate the label distribution of Davis dataset in Figure 8. Davis is extremely biased dataset; about 60 % of samples are having the same label 5.0. Therefore we also provide a close distribution ignoring samples belonging to the label 5.0 at Figure 8 (b).

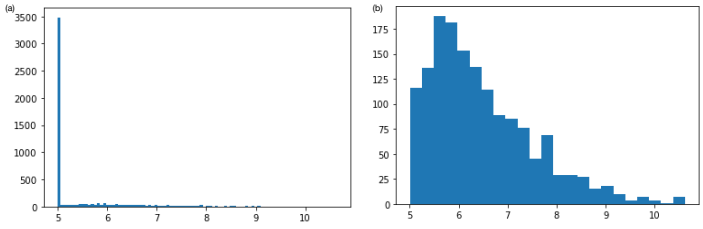

Figure 8: Label distribution of Davis dataset.

C.4 MULTIMODAL-CMU-MOSI

The label of CMU-MOSI ranges from [-3, 3]. We visualize the label distribution in Figure 9.

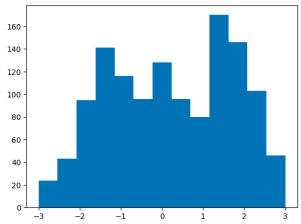

Figure 9: Label distribution of CMU-MOSI dataset.

D EXPERIMENTAL DETAILS

We provide experimental details and hyperparameter settings for each downstream task. We adopt the most widely used augmentation strategy and encoder architecture in each field to minimize deviation according to model configuration. We aim to maintain the same optimization method with the state-of-the-art work. One exception is the NLP task since the IMDB dataset was originally a classification task using different loss functions with regression.

Except for the vision task, we report the best performance over the temperature range [0.1, 1.0]. Among five experimental results using $\{0,1,2,3,4\}$ as random seeds, we report the average performance among three after excluding results from seeds with the highest and lowest error.

We also state practical details about device specification, memory usage, and time consumption which are needed to reproduce or develop from our work.

D.1 FRAMEWORK

Given an input batch of data, the augmentation module modifies the input data to generate an additional derivative of the input data. Forward propagation through the encoder module converts two batches of data to a fixed size of representation vectors, which will lie on the unit hypersphere by normalization. During training, the projection module consisting of two fully-connected layers transforms unit representations into another unit hypersphere. The supervised contrastive loss is computed on the outputs of the projection layers. At inference regression, a regression module comprising a linear classifier replaces the projection module to generate a prediction.

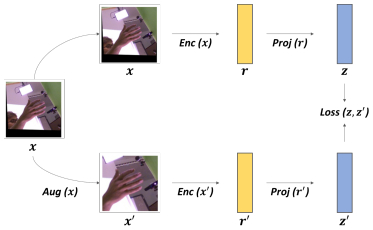

Figure 10: Illustration on LASCon framework.

D.2 NLP-IMDB

Encoder We mainly adopt the pretrained BERT (Bidirectional Encoder Representations from Transformers) encoder, which is one of the most widely-used pre-trained language model. Similar to previous works done by Yang et al. (2019); Sun et al. (2019), we download the BERT-base model provided by the HuggingFace Transformers (Wolf et al., 2019) and finetune using IMDB dataset.

Augmentation Wei & Zou (2019) provides four simple but powerful augmentation techniques for language tasks. We obtain augmented sentence via the default setting of synonym replacement (SR), random insertion (RI), random swap (RS), and random deletion (RD). Table 9 gives specific examples of sentence augmentation.

Table 9: Sentence augmentation example.

None	The storyline of the movie is great, and the acting is perfect.
Synonym Replacement	The **plot line** of the movie is great, and the acting is perfect.
Random Insertion	The storyline of **performing** the movie is great, and the acting is perfect.
Random Swap	The storyline of the movie is great, and **perfect acting is the**.
Random Deletion	The storyline of the movie is great, and the acting is.

Optimization We use Adam optimizer for optimizing parameters in each module in the framework. Initial learning rate is 1e-5 for BERT encoder and 1e-3 for other modules. Supervised contrastive

loss is applied for 1 epoch, then supervised L1 loss is utilized for 1 epoch of prediction. We test the wide range of temperature in the range [0.1, 1.0].

Practical We use NVIDIA GeForce RTX 3090 for experiment. 7GB of GPU memory and 20 minute of time is required to reproduce our result, for one stage of contrastive learning and five stage of fine-tuning with five independent seeds.

D.3 VISION-FREIHAND

We adopt many settings from PeCLR (Spurr et al., 2021).

Encoder We use ResNet-152 He et al. (2016) for encoder model. For pretraining process, two layer MLP is used as projection head. Input dimension of projection head is 2048, hidden dimension is 512, output dimension is 128.

Augmentation We use color jitter, resize, rotate, color drop, and crop for augmentation. Rotation angle is in $r \in [-45°, 45°]$. Color jitter is applied the aspect of hue, saturation, and brightness. Hue and saturation is scaled in $s \in [0.01, 1.0]$. Meanwhile, brightness is factored in $s \in [0.5, 1.0]$. Cropping is applied to crop box which all of keypoints exist in. Please refer Figure 11 for examples of augmentation.

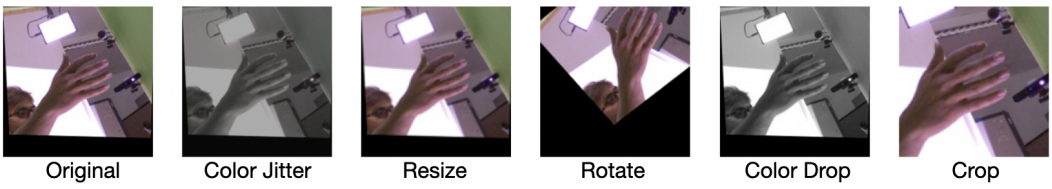

Figure 11: Examples of augmentation in FreiHAND

Optimization Supervised pretraining is performed for 100 epoch, which is adopted in PeCLR. For optimizer, ADAM wrapped with LARS is used with 256 batch size. Learning rate is set to $lr = \sqrt{|I|} * 1e-4$ for minibatch I. In addition, linear warmup is applied for first 10 epoch. Temperature of loss is $\tau = 0.5$. Fine tuning is performed for 100 epoch. In fine tuning, ADAM optimizer is used with 128 batch size, and learning rate $lr = 5e-4$.

Practical We use NVIDIA GeForce RTX 3090 for experiment. 20GB of GPU memory and 10 hours of time are needed for pretraining. For fine tuning, 4 GB of GPU memory and 7 hours of time are needed.

D.4 GRAPH-DAVIS

Encoder We inherit most of the encoder settings from Liu et al. (2022). The main graph encoder for the graph task is five layers of Graph Isomorphism Network (GIN) proposed by Xu et al. (2019). Since the drug-protein interaction prediction task requires additional protein sequence input, we use GraphDTA (Nguyen et al., 2021) framework as a wrapping encoder. Three layers of 1D CNN independently map 1D protein sequence input as a fixed length of a vector. The concatenation of the molecular representation vector from GIN and the protein representation vector from CNN roles as an embedding vector.

Augmentation We apply node masking (NM), node dropping (ND), and subgraph masking (SM) graph augmentation. Specifically, 20 % of random node attributes are masked to zero vector for node masking. For node dropping, 20 % of random nodes are erased from the graph, and corresponding edges are removed simultaneously. Subgraph masking is an extension of node masking so that consecutive 20 % of node attributes and connected edge attributes are masked to zero vectors. Figure 12 graphically explains three augmentation strategies.

Optimization The experiment settings are the same as GraphDTA. Both pretraining and finetuning is performed for 200 epochs with batch size of 256 with Learning rate equals to $lr = 5e-4$.

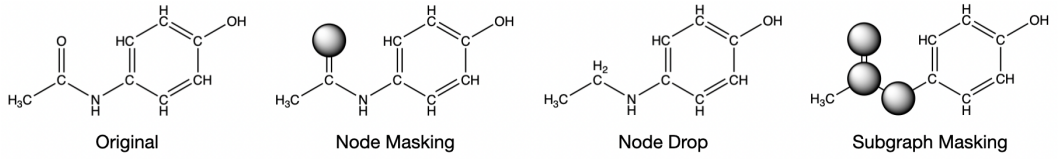

Figure 12: Examples of augmentation in FreiHAND

Practical We use NVIDIA GeForce RTX 3090 for experiment. 4GB of GPU memory and 3 hours of time are needed for pretraining. For fine tuning, 3 GB of GPU memory and 2 hours of time are needed.

D.5 MULTIMODAL-CMU-MOSI

Encoder We use three independent encoders for representing text, acoustic and visual input. Following previous studies (Yu et al., 2021; Han et al., 2021), 12-layers BERT Devlin et al. (2018) encodes text input and two unidirectional LSTMs Hochreiter & Schmidhuber (1997) encode acoustic and visual data.

Augmentation Since multiple encoders provide diverse view of input data, we do not use augmentation method for multimodal task.

Optimization We use Adam optimizer for optimizing parameters in each module in the framework. Initial learning rate is 1e-5 for BERT encoder and 1e-3 for other modules. Supervised contrastive loss is applied for 100 epochs, then supervised L1 loss is utilized for another 100 epochs of prediction. We test the wide range of temperature in the range [0.1, 1.0].

Practical We use NVIDIA GeForce RTX 3090 for experiment. For batch size 32, 16 GB of GPU memory and 20 minutes of time are needed for pretraining. For fine tuning, 16 GB of GPU memory and 1 hour of time are needed.

D.6 LABEL SIMILARITY FUNCTION

We provide a brief graph of label similarity functions to explain gradients near zero.

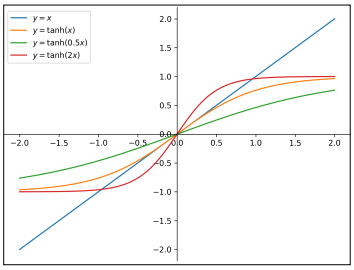

Figure 13: Label similarity functions

E FULL RESULTS

E.1 IMDB

Table 10: NLP contrastive learning on IMDB dataset with diverse loss functions and temperatures. Augmentation = Synonym Replacement.

			IN				OUT	
T	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
SimCLR								
1.0	1.415	0.856	86.843	86.527				
0.9	1.482	0.846	84.599	84.005				
0.8	1.306	0.865	91.556	91.547				
0.7	1.547	0.862	87.499	87.249				
0.6	1.268	0.869	89.564	89.507				
0.5	1.282	0.868	92.496	92.494				
0.4	1.372	0.858	89.837	89.756				
0.3	1.595	0.871	92.236	92.230				
0.2	1.367	0.865	89.317	89.169				
0.1	1.444	0.858	90.233	90.189				
SupCon								
1.0	1.539	0.840	82.387	80.900	1.846	0.748	78.464	77.419
0.9	1.156	0.886	92.657	92.651	1.134	0.889	91.871	91.853
0.8	1.144	0.888	91.773	91.757	1.297	0.884	89.908	89.834
0.7	1.163	0.881	92.216	92.200	1.994	0.772	81.449	80.607
0.6	1.317	0.879	91.559	91.527	1.951	0.745	80.197	78.434
0.5	1.151	0.886	90.337	90.282	2.162	0.739	82.204	81.947
0.4	1.746	0.876	73.967	68.097	1.150	0.881	91.693	91.672
0.3	1.271	0.875	92.344	92.333	1.174	0.892	92.281	92.268
0.2	1.160	0.886	92.833	92.830	1.275	0.889	92.639	92.624
0.1	1.280	0.872	89.253	89.168	1.921	0.772	80.031	79.133
LASCon	Linear							
1.0	1.237	0.879	92.605	92.599	1.246	0.884	91.017	90.979
0.9	1.675	0.846	75.924	72.610	2.737	nan	62.217	51.047
0.8	1.192	0.887	93.124	93.119	1.994	nan	78.581	73.024
0.7	1.293	0.856	91.465	91.438	1.245	0.878	90.225	90.151
0.6	1.248	0.880	91.293	91.259	1.280	0.869	89.348	89.237
0.5	1.178	0.887	93.227	93.224	1.110	0.895	92.976	92.973
0.4	1.147	0.891	92.396	92.379	1.213	0.884	87.509	87.290
0.3	1.401	0.876	86.661	86.005	1.176	0.885	90.959	90.922
0.2	1.191	0.878	91.375	91.345	1.519	0.830	88.351	88.154
0.1	1.202	0.879	88.711	88.601	1.419	0.860	86.009	85.597
LASCon	Tanh	0.5						
1.0	2.316	nan	71.549	65.317	1.692	0.820	87.784	87.697
0.9	1.347	0.876	86.748	86.502	1.333	0.867	91.651	91.638
0.8	1.227	0.879	92.849	92.845	1.490	0.857	88.451	88.332
0.7	1.240	0.878	92.471	92.462	1.319	0.857	89.849	89.808
0.6	1.283	0.872	89.913	89.787	1.322	0.864	88.792	88.714
0.5	1.208	0.887	92.459	92.445	1.358	0.857	87.895	87.748
0.4	1.147	0.836	91.936	91.918	2.032	0.745	80.676	78.928
0.3	1.150	0.889	89.819	89.739	1.352	0.854	89.077	88.991
0.2	1.397	0.870	91.996	91.971	2.055	0.842	90.405	90.375
0.1	1.213	0.882	92.440	92.437	2.048	0.838	73.705	67.912
LASCon	Tanh	1.0						
1.0	1.568	0.816	86.145	85.429	1.210	0.879	92.592	92.584
0.9	1.274	0.878	90.264	90.176	1.479	0.878	77.971	75.345
0.8	1.131	0.891	91.987	91.970	1.153	0.888	92.492	92.486
0.7	1.148	0.888	90.720	90.645	1.186	0.880	90.425	90.384
0.6	1.229	0.885	91.712	91.685	1.183	0.887	90.196	90.121
0.5	1.189	0.883	91.527	91.495	1.407	0.863	87.815	87.647
0.4	1.180	0.888	91.653	91.621	2.022	0.735	78.092	76.037
0.3	1.162	0.889	92.203	92.186	1.225	0.879	92.440	92.428
0.2	1.204	0.879	91.507	91.481	1.193	0.881	90.819	90.774
0.1	1.589	0.880	91.732	91.713	1.225	0.878	90.304	90.244
LASCon	Tanh	2.0						
1.0	1.233	0.879	91.695	91.668	1.184	0.886	91.431	91.388
0.9	1.322	0.872	90.341	90.251	1.126	0.894	91.521	91.493
0.8	1.232	0.878	92.292	92.276	1.112	0.890	90.063	89.998
0.7	1.228	0.879	92.049	92.025	1.260	0.884	90.461	90.401
0.6	1.160	0.891	92.759	92.749	1.559	0.827	84.599	83.987
0.5	1.226	0.877	90.615	90.543	1.405	0.842	83.301	82.452
0.4	1.243	0.880	91.813	91.793	1.097	0.893	90.239	90.173
0.3	1.317	0.889	92.039	92.014	1.211	0.885	91.357	91.299
0.2	1.145	0.888	90.784	90.722	1.134	0.890	91.588	91.564
0.1	1.460	0.867	86.739	86.501	1.101	0.889	89.327	89.244

Table 11: NLP contrastive learning on IMDB dataset with diverse loss functions and temperatures. Augmentation = Random Deletion.

			IN				OUT	
T	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
SimCLR								
1.0	1.260	0.869	92.033	92.022				
0.9	1.288	0.871	91.268	91.241				
0.8	1.225	0.871	90.409	90.371				
0.7	1.279	0.871	90.131	90.080				
0.6	1.260	0.876	91.744	91.729				
0.5	1.215	0.875	92.317	92.314				
0.4	1.511	0.858	90.876	90.847				
0.3	1.324	0.866	89.172	89.080				
0.2	1.180	0.882	90.903	90.873				
0.1	1.342	0.872	88.691	88.598				
SupCon								
1.0	1.666	0.840	90.976	90.951	1.245	0.893	93.572	93.571
0.9	1.273	0.884	90.017	89.939	1.247	0.871	89.865	89.748
0.8	1.638	0.870	92.455	92.443	1.437	0.889	78.731	73.171
0.7	1.319	0.877	88.865	88.698	1.111	0.893	90.939	90.895
0.6	1.799	0.793	85.385	84.516	1.757	0.868	77.273	71.690
0.5	2.214	0.578	71.045	64.936	1.138	0.891	91.765	91.742
0.4	1.308	0.876	88.876	88.637	1.513	0.879	78.664	74.493
0.3	1.101	0.893	91.343	91.307	1.314	0.881	89.853	89.709
0.2	1.187	0.881	91.565	91.536	1.169	0.885	91.083	91.047
0.1	1.440	0.844	85.321	84.391	1.788	0.831	78.963	73.405
LASCon	Linear							
1.0	1.166	0.887	92.831	92.824	1.141	0.890	91.456	91.422
0.9	1.236	0.882	91.036	90.981	1.182	0.883	91.375	91.342
0.8	1.233	0.880	91.751	91.733	1.196	0.881	89.167	89.050
0.7	1.199	0.886	91.633	91.607	1.332	0.883	85.571	84.949
0.6	1.508	0.882	77.888	72.314	1.137	0.892	92.049	92.025
0.5	1.269	0.890	92.503	92.488	1.161	0.888	91.500	91.475
0.4	1.680	0.886	77.005	71.418	1.179	0.885	92.136	92.124
0.3	1.181	0.879	93.087	93.084	1.462	0.883	77.847	72.271
0.2	1.127	0.892	90.904	90.853	1.168	0.888	93.244	93.241
0.1	1.363	0.865	90.009	89.937	1.240	0.876	89.340	89.254
LASCon	Tanh	0.5						
1.0	1.175	0.888	92.683	92.676	1.297	0.860	91.132	91.119
0.9	1.230	0.872	89.761	89.621	1.701	0.835	88.579	88.466
0.8	1.756	0.811	83.720	83.266	1.523	0.851	88.917	88.811
0.7	1.313	0.870	91.427	91.393	1.492	0.841	89.253	89.185
0.6	1.176	0.890	92.608	92.597	1.502	0.831	82.768	82.116
0.5	1.182	0.886	92.351	92.328	1.327	0.864	89.513	89.446
0.4	1.320	0.883	92.864	92.854	1.308	0.859	89.059	88.987
0.3	1.221	0.890	92.039	92.007	2.287	0.845	90.504	90.481
0.2	1.366	0.885	93.137	93.133	1.907	0.791	70.812	66.227
0.1	1.463	0.849	81.195	80.571	1.332	0.859	90.981	90.960
LASCon	Tanh	1.0						
1.0	1.366	0.874	88.245	87.962	1.147	0.887	90.892	90.847
0.9	1.220	0.872	91.260	91.229	1.189	0.887	91.599	91.564
0.8	1.225	0.882	92.112	92.097	1.193	0.882	89.277	89.183
0.7	1.435	0.876	81.083	79.027	1.309	0.879	91.676	91.639
0.6	1.582	0.868	83.145	82.482	1.207	0.885	88.943	88.781
0.5	2.177	0.597	78.429	72.867	1.373	0.875	87.927	87.663
0.4	1.246	0.883	87.841	87.586	1.298	0.862	91.052	91.000
0.3	1.142	0.890	92.937	92.931	1.298	0.878	92.080	92.066
0.2	1.127	0.885	92.501	92.487	1.201	0.885	90.855	90.814
0.1	1.432	0.880	80.553	77.139	1.301	0.876	90.055	89.986
LASCon	Tanh	2.0						
1.0	1.267	0.884	89.507	89.418	1.339	0.880	91.157	91.080
0.9	1.190	0.886	92.277	92.261	1.176	0.885	91.363	91.322
0.8	1.273	0.879	90.147	90.082	1.195	0.889	93.080	93.076
0.7	1.145	0.889	92.577	92.563	1.100	0.895	91.964	91.939
0.6	1.205	0.884	90.113	90.025	1.740	0.780	80.916	80.275
0.5	1.337	0.865	90.021	89.975	1.165	0.890	90.196	90.131
0.4	1.109	0.890	91.231	91.199	1.300	0.875	92.347	92.335
0.3	1.119	0.893	91.301	91.255	1.227	0.876	91.691	91.668
0.2	1.235	0.885	89.985	89.922	1.272	0.883	89.299	89.199
0.1	1.398	0.863	85.845	85.302	1.169	0.890	92.172	92.158

Table 12: NLP contrastive learning on IMDB dataset with diverse loss functions and temperatures. Augmentation = Random Swapping.

			IN				OUT	
T	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
SimCLR								
1.0	1.674	0.860	72.701	68.668				
0.9	1.780	0.838	86.265	85.915				
0.8	1.267	0.874	91.903	91.894				
0.7	1.528	0.838	89.239	89.160				
0.6	1.299	0.865	89.932	89.864				
0.5	1.170	0.879	91.421	91.403				
0.4	1.304	0.865	89.677	89.615				
0.3	1.298	0.862	90.904	90.871				
0.2	1.232	0.877	91.599	91.583				
0.1	1.208	0.877	92.491	92.489				
SupCon								
1.0	1.292	0.884	86.903	86.483	1.286	0.864	88.660	88.415
0.9	1.253	0.883	92.477	92.468	1.196	0.885	92.791	92.783
0.8	1.167	0.888	91.665	91.631	1.271	0.882	90.531	90.467
0.7	1.157	0.890	91.893	91.874	1.135	0.890	92.423	92.412
0.6	1.228	0.886	90.863	90.812	1.427	0.858	90.033	89.937
0.5	1.233	0.879	91.081	91.020	1.220	0.883	90.831	90.778
0.4	1.331	0.878	92.941	92.936	1.514	0.878	80.568	77.142
0.3	1.498	0.850	82.479	81.023	1.275	0.888	93.057	93.052
0.2	1.784	0.856	90.145	90.077	1.174	0.884	92.425	92.413
0.1	2.254	0.783	85.789	85.010	1.146	0.890	92.233	92.220
LASCon	Linear							
1.0	1.193	0.877	90.963	90.911	1.264	0.873	88.236	88.105
0.9	1.992	nan	76.856	71.264	1.298	0.878	90.517	90.457
0.8	1.243	0.884	90.536	90.468	1.524	0.816	85.661	84.977
0.7	1.789	0.841	78.095	72.532	1.178	0.879	91.761	91.725
0.6	1.389	0.871	85.640	85.262	1.585	0.879	82.025	80.478
0.5	1.239	0.883	91.080	91.025	1.260	0.881	92.563	92.547
0.4	1.302	0.884	89.808	89.732	1.100	0.896	92.548	92.539
0.3	1.145	0.884	91.612	91.582	1.447	0.846	86.663	86.397
0.2	1.171	0.887	92.404	92.390	1.177	0.889	89.993	89.918
0.1	1.714	0.874	92.480	92.475	1.953	0.864	77.676	72.108
LASCon	Tanh	0.5						
1.0	1.476	0.882	84.237	83.693	1.271	0.865	91.873	91.868
0.9	1.137	0.888	92.459	92.447	1.300	0.861	91.280	91.271
0.8	1.193	0.890	92.248	92.234	1.393	0.846	90.183	90.158
0.7	2.177	nan	77.305	71.727	1.381	0.851	87.735	87.602
0.6	1.258	0.884	92.759	92.751	1.808	0.835	87.241	87.076
0.5	1.200	0.884	91.433	91.405	1.436	0.834	87.936	87.758
0.4	1.115	0.893	91.931	91.910	2.081	0.822	75.159	70.270
0.3	1.118	0.891	91.579	91.556	1.390	0.857	91.268	91.261
0.2	1.198	0.882	92.207	92.193	1.244	0.870	91.644	91.636
0.1	1.250	0.879	91.571	91.554	1.249	0.874	90.793	90.765
LASCon	Tanh	1.0						
1.0	1.176	0.884	92.971	92.969	1.821	0.841	72.972	67.199
0.9	1.155	0.886	92.388	92.380	1.248	0.879	90.581	90.528
0.8	2.084	nan	77.251	71.661	1.356	0.875	85.553	85.066
0.7	1.241	0.874	89.645	89.566	1.424	0.883	90.100	89.982
0.6	1.193	0.884	92.663	92.654	1.534	0.818	85.084	84.647
0.5	1.424	0.881	79.069	73.563	1.139	0.887	90.480	90.430
0.4	1.766	0.832	86.469	86.021	1.291	0.880	92.540	92.532
0.3	1.158	0.886	92.172	92.161	1.322	0.859	87.395	87.095
0.2	1.209	0.884	91.063	91.012	1.448	0.875	83.339	82.680
0.1	1.277	0.878	92.112	92.104	1.328	0.869	90.108	90.030
LASCon	Tanh	2.0						
1.0	1.286	0.879	88.351	88.251	1.161	0.886	90.764	90.717
0.9	1.172	0.893	92.628	92.617	1.350	0.858	89.029	88.905
0.8	1.521	0.874	83.321	82.803	1.412	0.846	88.139	87.916
0.7	1.654	0.864	73.896	68.188	1.318	0.856	90.831	90.755
0.6	1.127	0.894	92.275	92.259	1.196	0.885	91.051	90.990
0.5	1.341	0.855	87.425	87.171	1.298	0.882	87.327	87.001
0.4	1.383	0.876	85.895	85.302	2.408	0.840	77.656	72.076
0.3	1.438	0.861	89.276	89.098	1.154	0.889	91.205	91.178
0.2	1.135	0.890	92.857	92.851	1.174	0.891	90.601	90.552
0.1	1.211	0.883	92.416	92.411	1.108	0.892	92.028	92.006

Table 13: NLP contrastive learning on IMDB dataset with diverse loss functions and temperatures. Augmentation = Random Insertion.

			IN				OUT	
T	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
SimCLR								
1.0	1.463	0.862	90.032	89.995				
0.9	1.465	0.868	90.417	90.372				
0.8	1.218	0.872	92.093	92.087				
0.7	1.917	0.868	90.936	90.896				
0.6	1.549	0.868	75.593	70.215				
0.5	2.041	nan	76.099	70.494				
0.4	1.203	0.875	90.455	90.411				
0.3	1.272	0.876	91.207	91.171				
0.2	1.244	0.874	91.172	91.139				
0.1	1.507	0.860	76.559	70.975				
SupCon								
1.0	1.189	0.882	92.284	92.273	1.120	0.891	92.035	92.011
0.9	1.113	0.890	92.853	92.848	1.546	0.885	90.687	90.608
0.8	1.145	0.887	92.307	92.296	1.209	0.891	91.841	91.804
0.7	1.249	0.890	87.583	87.232	1.820	0.876	64.035	52.939
0.6	1.793	0.843	83.515	82.936	1.248	0.879	91.399	91.354
0.5	1.216	0.882	90.660	90.578	1.471	0.826	80.016	77.548
0.4	2.048	0.832	90.927	90.882	1.443	0.872	82.393	80.562
0.3	1.371	0.880	86.053	85.511	1.800	0.889	77.788	72.216
0.2	1.164	0.887	92.707	92.698	1.170	0.889	90.953	90.909
0.1	1.673	0.845	81.364	78.420	1.316	0.895	87.427	86.973
LASCon	Linear							
1.0	1.715	0.807	83.617	82.567	1.296	0.888	86.821	86.590
0.9	2.606	0.295	78.307	72.745	1.257	0.884	89.964	89.866
0.8	1.210	0.883	92.839	92.832	1.205	0.879	91.003	90.950
0.7	1.193	0.876	92.120	92.106	1.279	0.887	88.617	88.394
0.6	1.477	0.882	91.652	91.615	1.181	0.886	90.429	90.354
0.5	1.194	0.884	93.043	93.041	1.237	0.881	91.513	91.476
0.4	1.219	0.879	92.599	92.592	1.075	0.893	90.601	90.553
0.3	1.258	0.863	88.392	88.203	1.180	0.881	91.157	91.117
0.2	1.194	0.887	91.531	91.499	1.122	0.891	92.411	92.398
0.1	1.371	0.872	89.316	89.213	1.206	0.886	89.225	89.126
LASCon	Tanh	0.5						
1.0	2.086	nan	78.088	72.526	1.382	0.853	86.533	86.290
0.9	2.360	0.574	74.153	68.431	1.778	0.816	77.725	72.165
0.8	1.154	0.886	92.163	92.147	1.355	0.855	90.408	90.381
0.7	1.239	0.870	89.497	89.382	1.409	0.847	88.897	88.811
0.6	1.149	0.890	90.647	90.588	1.289	0.859	90.271	90.235
0.5	1.288	0.881	89.816	89.750	1.390	0.851	89.145	89.069
0.4	1.194	0.888	92.217	92.193	1.713	0.860	77.915	72.357
0.3	2.492	0.531	70.451	64.061	1.392	0.870	86.791	86.492
0.2	2.512	0.833	90.831	90.795	1.326	0.858	90.923	90.907
0.1	1.305	0.879	92.645	92.640	1.260	0.870	92.075	92.071
LASCon	Tanh	1.0						
1.0	2.251	0.865	88.419	88.241	1.190	0.887	89.648	89.579
0.9	1.157	0.888	91.697	91.661	1.324	0.883	89.925	89.847
0.8	2.766	0.298	76.475	70.876	1.227	0.880	93.172	93.167
0.7	1.130	0.890	91.876	91.854	1.305	0.883	86.340	85.832
0.6	1.189	0.882	91.505	91.479	1.259	0.878	89.395	89.281
0.5	1.141	0.885	89.741	89.636	1.178	0.883	90.476	90.412
0.4	1.153	0.884	91.071	91.019	1.853	0.871	92.148	92.134
0.3	1.294	0.885	89.451	89.358	1.302	0.873	88.016	87.810
0.2	1.389	0.873	86.877	86.327	1.312	0.881	86.801	86.616
0.1	1.630	0.813	82.805	82.057	1.226	0.878	90.068	89.997
LASCon	Tanh	2.0						
1.0	2.571	0.779	82.712	82.113	1.385	0.882	85.485	85.212
0.9	1.290	0.868	89.975	89.875	1.224	0.880	91.149	91.114
0.8	1.142	0.888	90.860	90.817	1.446	0.876	83.041	82.042
0.7	1.130	0.890	92.396	92.383	1.188	0.895	92.136	92.119
0.6	1.382	0.874	77.967	72.436	1.155	0.886	90.561	90.518
0.5	1.616	0.811	85.071	84.208	1.436	0.868	81.593	80.300
0.4	1.108	0.893	91.239	91.209	1.156	0.891	91.329	91.276
0.3	1.121	0.891	90.245	90.190	1.162	0.886	91.877	91.851
0.2	1.731	0.852	75.837	70.183	1.370	0.887	82.000	79.845
0.1	1.304	0.864	88.183	88.026	1.212	0.886	91.267	91.221

E.2 DAVIS

Table 14: Graph contrastive learning on Davis dataset with diverse loss functions and temperatures. Augmentation = Subgraph Masking.

	IN						OUT					
T	MSE(↓)	RMSE(↓)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MSE(↓)	RMSE(↓)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
NT-Xent												
1.0	0.301	0.549	0.324	0.793	0.942	0.937						
0.9	0.307	0.554	0.326	0.790	0.940	0.936						
0.8	0.308	0.555	0.322	0.788	0.940	0.936						
0.7	0.307	0.554	0.327	0.788	0.942	0.936						
0.6	0.310	0.557	0.330	0.787	0.940	0.936						
0.5	0.305	0.552	0.323	0.790	0.942	0.937						
0.4	0.313	0.559	0.326	0.783	0.941	0.936						
0.3	0.318	0.564	0.332	0.779	0.940	0.935						
0.2	0.299	0.547	0.324	0.795	0.942	0.938						
0.1	0.304	0.551	0.328	0.792	0.943	0.938						
SupCon												
1.0	0.318	0.564	0.328	0.780	0.941	0.936	0.306	0.553	0.318	0.789	0.942	0.938
0.9	0.331	0.576	0.343	0.768	0.939	0.933	0.292	0.540	0.312	0.800	0.943	0.938
0.8	0.329	0.574	0.336	0.771	0.938	0.932	0.298	0.546	0.316	0.796	0.942	0.937
0.7	0.323	0.568	0.329	0.775	0.940	0.935	0.299	0.547	0.320	0.794	0.941	0.937
0.5	0.329	0.573	0.328	0.772	0.940	0.934	0.296	0.544	0.315	0.798	0.944	0.939
0.4	0.319	0.564	0.325	0.779	0.940	0.935	0.296	0.544	0.320	0.798	0.943	0.939
0.3	0.320	0.566	0.330	0.776	0.940	0.935	0.295	0.543	0.319	0.800	0.943	0.939
0.2	0.311	0.558	0.326	0.785	0.941	0.935	0.287	0.536	0.313	0.802	0.943	0.938
0.1	0.307	0.554	0.323	0.788	0.942	0.938	0.297	0.545	0.317	0.796	0.943	0.938
LASCon	Linear											
1.0	0.308	0.555	0.320	0.786	0.941	0.936	0.278	0.528	0.310	0.810	0.944	0.940
0.9	0.320	0.566	0.328	0.777	0.941	0.935	0.277	0.526	0.309	0.811	0.945	0.941
0.8	0.336	0.580	0.338	0.764	0.937	0.931	0.280	0.529	0.313	0.809	0.943	0.939
0.7	0.355	0.596	0.351	0.749	0.939	0.932	0.277	0.526	0.308	0.812	0.944	0.940
0.6	0.325	0.570	0.331	0.773	0.941	0.935	0.282	0.530	0.313	0.807	0.944	0.940
0.5	0.331	0.575	0.336	0.770	0.940	0.935	0.276	0.525	0.305	0.811	0.943	0.938
0.4	0.358	0.598	0.356	0.748	0.940	0.933	0.277	0.526	0.308	0.812	0.944	0.939
0.3	0.311	0.557	0.327	0.787	0.940	0.936	0.282	0.531	0.306	0.807	0.943	0.938
0.2	0.308	0.555	0.322	0.788	0.942	0.938	0.280	0.529	0.309	0.808	0.944	0.938
0.1	0.310	0.557	0.330	0.785	0.941	0.937	0.298	0.546	0.320	0.795	0.942	0.937
LASCon	Tanh	0.5										
1.0	0.332	0.576	0.337	0.768	0.939	0.933	0.283	0.532	0.311	0.807	0.943	0.939
0.9	0.321	0.567	0.332	0.778	0.941	0.936	0.279	0.528	0.312	0.808	0.944	0.940
0.8	0.327	0.572	0.328	0.772	0.940	0.934	0.278	0.527	0.310	0.811	0.944	0.940
0.7	0.332	0.576	0.331	0.769	0.940	0.933	0.279	0.528	0.306	0.808	0.943	0.939
0.6	0.330	0.574	0.336	0.770	0.939	0.934	0.280	0.529	0.306	0.807	0.942	0.938
0.5	0.365	0.604	0.364	0.741	0.937	0.929	0.285	0.534	0.309	0.806	0.941	0.936
0.4	0.361	0.601	0.363	0.743	0.940	0.931	0.281	0.530	0.310	0.807	0.943	0.939
0.3	0.307	0.554	0.325	0.788	0.941	0.937	0.281	0.531	0.311	0.808	0.945	0.941
0.2	0.307	0.554	0.322	0.789	0.939	0.935	0.297	0.545	0.317	0.796	0.942	0.937
0.1	0.299	0.547	0.317	0.794	0.941	0.936	0.339	0.582	0.342	0.761	0.939	0.932
LASCon	Tanh	1.0										
1.0	0.329	0.573	0.333	0.770	0.939	0.933	0.279	0.528	0.308	0.809	0.944	0.939
0.9	0.336	0.580	0.334	0.764	0.939	0.933	0.277	0.526	0.308	0.811	0.944	0.939
0.8	0.336	0.579	0.339	0.764	0.939	0.934	0.278	0.528	0.309	0.810	0.944	0.940
0.7	0.335	0.578	0.339	0.765	0.939	0.933	0.277	0.526	0.311	0.811	0.945	0.941
0.6	0.339	0.582	0.334	0.761	0.940	0.934	0.279	0.529	0.309	0.809	0.945	0.941
0.5	0.336	0.579	0.335	0.765	0.939	0.933	0.274	0.524	0.306	0.812	0.944	0.941
0.4	0.358	0.599	0.357	0.746	0.937	0.930	0.277	0.526	0.304	0.812	0.944	0.940
0.3	0.310	0.557	0.330	0.785	0.941	0.937	0.281	0.530	0.310	0.807	0.945	0.941
0.2	0.322	0.567	0.326	0.775	0.940	0.934	0.281	0.530	0.310	0.808	0.943	0.939
0.1	0.300	0.547	0.321	0.793	0.943	0.939	0.290	0.539	0.315	0.801	0.943	0.939
LASCon	Tanh	2.0										
1.0	0.319	0.565	0.322	0.779	0.941	0.936	0.280	0.529	0.308	0.809	0.944	0.939
0.9	0.329	0.574	0.333	0.770	0.939	0.934	0.280	0.529	0.310	0.809	0.943	0.939
0.8	0.322	0.567	0.328	0.774	0.939	0.933	0.284	0.533	0.312	0.805	0.944	0.940
0.7	0.333	0.576	0.330	0.767	0.940	0.933	0.280	0.529	0.310	0.808	0.944	0.940
0.6	0.340	0.583	0.339	0.761	0.939	0.933	0.286	0.535	0.312	0.804	0.943	0.939
0.5	0.346	0.588	0.342	0.755	0.938	0.931	0.280	0.529	0.307	0.809	0.945	0.941
0.4	0.346	0.589	0.348	0.756	0.940	0.933	0.277	0.526	0.307	0.811	0.944	0.939
0.3	0.311	0.558	0.324	0.785	0.940	0.935	0.278	0.527	0.310	0.809	0.944	0.940
0.2	0.305	0.552	0.324	0.790	0.941	0.935	0.282	0.531	0.308	0.808	0.944	0.940
0.1	0.305	0.552	0.326	0.791	0.941	0.937	0.293	0.541	0.320	0.799	0.942	0.937

Table 15: Graph contrastive learning on Davis dataset with diverse loss functions and temperatures. Augmentation = Node Masking.

			IN						OUT			
T	MSE(↓)	RMSE(↓)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MSE(↓)	RMSE(↓)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
NT-Xent												
1.0	0.354	0.595	0.361	0.751	0.939	0.932						
0.9	0.335	0.578	0.357	0.765	0.939	0.932						
0.8	0.310	0.557	0.330	0.786	0.942	0.938						
0.7	0.308	0.555	0.327	0.787	0.942	0.937						
0.6	0.305	0.552	0.325	0.789	0.942	0.937						
0.5	0.305	0.552	0.324	0.791	0.942	0.937						
0.4	0.346	0.588	0.347	0.757	0.940	0.932						
0.3	0.321	0.567	0.337	0.776	0.941	0.935						
0.2	0.320	0.566	0.337	0.779	0.940	0.935						
0.1	0.313	0.560	0.335	0.787	0.941	0.936						
SupCon												
1.0	0.326	0.571	0.339	0.772	0.939	0.932	0.301	0.549	0.318	0.794	0.942	0.938
0.9	0.309	0.556	0.318	0.787	0.939	0.934	0.303	0.550	0.322	0.790	0.942	0.938
0.8	0.325	0.570	0.327	0.774	0.940	0.934	0.299	0.546	0.323	0.794	0.942	0.938
0.5	0.327	0.572	0.331	0.772	0.941	0.935	0.299	0.546	0.315	0.795	0.943	0.938
0.4	0.312	0.559	0.323	0.785	0.942	0.937	0.298	0.545	0.320	0.796	0.943	0.937
0.3	0.320	0.566	0.332	0.778	0.940	0.935	0.299	0.547	0.317	0.794	0.944	0.939
0.2	0.310	0.557	0.320	0.787	0.941	0.936	0.290	0.539	0.318	0.801	0.944	0.939
0.1	0.308	0.555	0.323	0.787	0.940	0.935	0.294	0.542	0.317	0.798	0.942	0.937
LASCon	Linear											
1.0	0.326	0.571	0.332	0.773	0.940	0.935	0.278	0.528	0.306	0.810	0.944	0.940
0.9	0.325	0.570	0.331	0.775	0.940	0.934	0.282	0.531	0.308	0.807	0.943	0.938
0.8	0.323	0.568	0.330	0.775	0.940	0.935	0.279	0.528	0.308	0.809	0.944	0.940
0.7	0.332	0.576	0.329	0.767	0.939	0.933	0.280	0.529	0.313	0.811	0.942	0.938
0.6	0.321	0.566	0.332	0.777	0.940	0.933	0.276	0.525	0.307	0.812	0.944	0.940
0.5	0.326	0.571	0.330	0.772	0.940	0.933	0.279	0.528	0.312	0.809	0.945	0.941
0.4	0.332	0.576	0.343	0.767	0.940	0.934	0.285	0.534	0.309	0.806	0.945	0.941
0.3	0.309	0.556	0.325	0.790	0.941	0.937	0.280	0.529	0.307	0.808	0.944	0.940
0.2	0.304	0.551	0.322	0.791	0.942	0.938	0.282	0.531	0.310	0.808	0.943	0.939
0.1	0.310	0.556	0.325	0.785	0.941	0.936	0.287	0.536	0.318	0.804	0.943	0.938
LASCon	Tanh	0.5										
1.0	0.335	0.579	0.336	0.764	0.939	0.933	0.279	0.529	0.310	0.809	0.944	0.939
0.9	0.324	0.569	0.326	0.773	0.941	0.935	0.278	0.527	0.308	0.810	0.943	0.939
0.8	0.341	0.584	0.339	0.760	0.939	0.933	0.281	0.530	0.311	0.808	0.944	0.940
0.7	0.330	0.575	0.330	0.768	0.940	0.933	0.278	0.527	0.308	0.809	0.945	0.941
0.6	0.323	0.568	0.321	0.775	0.941	0.935	0.277	0.527	0.307	0.811	0.945	0.940
0.5	0.346	0.588	0.344	0.756	0.938	0.932	0.278	0.527	0.307	0.810	0.945	0.941
0.4	0.318	0.564	0.332	0.778	0.940	0.934	0.286	0.535	0.311	0.804	0.944	0.940
0.3	0.303	0.551	0.324	0.791	0.942	0.936	0.285	0.534	0.312	0.806	0.943	0.939
0.2	0.307	0.554	0.325	0.788	0.943	0.938	0.288	0.537	0.316	0.802	0.943	0.938
0.1	0.307	0.554	0.322	0.789	0.941	0.937	0.335	0.578	0.333	0.764	0.940	0.932
LASCon	Tanh	1.0										
1.0	0.329	0.573	0.330	0.772	0.938	0.931	0.281	0.530	0.311	0.808	0.944	0.940
0.9	0.334	0.578	0.332	0.766	0.938	0.932	0.273	0.523	0.308	0.814	0.944	0.940
0.8	0.345	0.588	0.345	0.756	0.937	0.931	0.278	0.528	0.308	0.810	0.944	0.941
0.7	0.331	0.575	0.318	0.768	0.938	0.932	0.276	0.525	0.305	0.811	0.944	0.940
0.6	0.321	0.566	0.327	0.776	0.940	0.933	0.279	0.528	0.308	0.809	0.945	0.939
0.5	0.355	0.596	0.353	0.750	0.936	0.930	0.276	0.526	0.307	0.810	0.945	0.941
0.4	0.357	0.598	0.348	0.747	0.939	0.931	0.279	0.528	0.308	0.811	0.944	0.940
0.3	0.303	0.550	0.332	0.791	0.940	0.936	0.275	0.525	0.309	0.813	0.945	0.942
0.2	0.302	0.550	0.323	0.790	0.943	0.937	0.278	0.528	0.312	0.811	0.944	0.940
0.1	0.311	0.558	0.325	0.785	0.941	0.936	0.293	0.542	0.320	0.799	0.944	0.939
LASCon	Tanh	2.0										
1.0	0.318	0.564	0.329	0.780	0.939	0.933	0.285	0.534	0.310	0.804	0.944	0.940
0.9	0.335	0.579	0.334	0.767	0.939	0.933	0.285	0.534	0.315	0.805	0.943	0.939
0.8	0.318	0.563	0.324	0.780	0.941	0.936	0.283	0.533	0.311	0.807	0.944	0.939
0.7	0.341	0.584	0.339	0.760	0.940	0.934	0.286	0.535	0.320	0.804	0.943	0.940
0.6	0.338	0.581	0.336	0.762	0.938	0.930	0.283	0.532	0.313	0.806	0.945	0.941
0.5	0.352	0.593	0.345	0.752	0.938	0.931	0.284	0.533	0.312	0.806	0.942	0.938
0.4	0.327	0.572	0.334	0.774	0.939	0.934	0.283	0.531	0.311	0.806	0.944	0.940
0.3	0.311	0.558	0.321	0.785	0.941	0.934	0.279	0.528	0.305	0.809	0.944	0.940
0.2	0.298	0.546	0.316	0.794	0.941	0.936	0.279	0.528	0.310	0.809	0.944	0.940
0.1	0.317	0.563	0.328	0.780	0.941	0.936	0.283	0.532	0.311	0.805	0.945	0.940

Table 16: Graph contrastive learning on Davis dataset with diverse loss functions and temperatures. Augmentation = Node Dropping.

			IN						OUT			
T	MSE(↓)	RMSE(↓)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)	MSE(↓)	RMSE(↓)	MAE(↓)	Corr(↑)	Acc-2(↑)	F1(↑)
NT-Xent												
1.0	0.306	0.554	0.319	0.788	0.940	0.935						
0.9	0.314	0.560	0.333	0.786	0.941	0.937						
0.8	0.321	0.567	0.332	0.776	0.942	0.937						
0.7	0.308	0.555	0.327	0.787	0.941	0.936						
0.6	0.307	0.554	0.327	0.787	0.943	0.937						
0.5	0.320	0.566	0.327	0.776	0.941	0.935						
0.4	0.310	0.557	0.325	0.786	0.941	0.935						
0.3	0.306	0.553	0.329	0.788	0.942	0.936						
0.2	0.290	0.538	0.325	0.802	0.943	0.938						
0.1	0.286	0.535	0.315	0.806	0.943	0.938						
SupCon												
1.0	0.355	0.595	0.343	0.751	0.937	0.930	0.297	0.545	0.315	0.796	0.943	0.938
0.9	0.353	0.594	0.351	0.751	0.938	0.932	0.297	0.545	0.318	0.796	0.942	0.938
0.8	0.338	0.581	0.342	0.764	0.939	0.934	0.291	0.539	0.315	0.801	0.944	0.940
0.7	0.320	0.566	0.330	0.777	0.941	0.935	0.295	0.543	0.317	0.798	0.944	0.940
0.5	0.326	0.571	0.330	0.774	0.942	0.935	0.291	0.539	0.312	0.800	0.943	0.938
0.4	0.323	0.568	0.328	0.775	0.940	0.933	0.292	0.540	0.318	0.800	0.943	0.938
0.3	0.313	0.559	0.327	0.784	0.940	0.935	0.293	0.541	0.314	0.798	0.942	0.937
0.2	0.309	0.556	0.322	0.787	0.941	0.936	0.286	0.535	0.310	0.804	0.944	0.939
0.1	0.302	0.549	0.313	0.793	0.941	0.937	0.294	0.542	0.318	0.798	0.943	0.938
LASCon	Linear											
1.0	0.329	0.574	0.341	0.770	0.940	0.934	0.282	0.531	0.314	0.807	0.945	0.941
0.9	0.332	0.576	0.328	0.768	0.941	0.935	0.283	0.533	0.313	0.806	0.944	0.940
0.8	0.344	0.587	0.338	0.760	0.941	0.935	0.281	0.530	0.310	0.808	0.945	0.940
0.7	0.331	0.575	0.336	0.769	0.940	0.933	0.281	0.530	0.312	0.808	0.944	0.939
0.6	0.336	0.580	0.337	0.763	0.940	0.933	0.279	0.528	0.310	0.811	0.944	0.939
0.5	0.327	0.572	0.331	0.772	0.940	0.934	0.282	0.531	0.312	0.808	0.945	0.940
0.4	0.320	0.566	0.333	0.778	0.941	0.935	0.283	0.532	0.311	0.807	0.944	0.940
0.3	0.322	0.568	0.334	0.778	0.940	0.935	0.285	0.534	0.315	0.805	0.945	0.940
0.2	0.307	0.554	0.320	0.788	0.942	0.937	0.288	0.537	0.316	0.803	0.944	0.940
0.1	0.296	0.544	0.317	0.796	0.942	0.938	0.288	0.537	0.320	0.802	0.942	0.938
LASCon	Tanh	0.5										
1.0	0.334	0.578	0.332	0.765	0.940	0.933	0.277	0.526	0.311	0.811	0.943	0.939
0.9	0.342	0.585	0.340	0.760	0.939	0.933	0.278	0.527	0.308	0.810	0.945	0.942
0.8	0.315	0.561	0.327	0.780	0.940	0.934	0.276	0.526	0.307	0.812	0.945	0.941
0.7	0.324	0.569	0.334	0.776	0.942	0.937	0.283	0.532	0.310	0.807	0.945	0.941
0.6	0.335	0.579	0.336	0.766	0.939	0.932	0.281	0.530	0.310	0.809	0.944	0.939
0.5	0.332	0.576	0.339	0.767	0.941	0.935	0.278	0.527	0.311	0.809	0.944	0.940
0.4	0.317	0.564	0.331	0.779	0.942	0.936	0.282	0.531	0.309	0.808	0.943	0.939
0.3	0.312	0.558	0.324	0.785	0.940	0.935	0.275	0.524	0.308	0.813	0.943	0.939
0.2	0.307	0.554	0.323	0.788	0.941	0.936	0.280	0.529	0.310	0.809	0.944	0.940
0.1	0.297	0.546	0.316	0.797	0.943	0.938	0.282	0.531	0.321	0.807	0.943	0.939
LASCon	Tanh	1.0										
1.0	0.322	0.567	0.330	0.777	0.942	0.936	0.283	0.532	0.312	0.808	0.945	0.941
0.9	0.324	0.569	0.321	0.774	0.940	0.934	0.279	0.528	0.311	0.811	0.944	0.940
0.8	0.309	0.556	0.318	0.786	0.942	0.937	0.278	0.527	0.307	0.810	0.945	0.941
0.7	0.349	0.591	0.344	0.753	0.937	0.930	0.276	0.526	0.309	0.811	0.945	0.941
0.6	0.332	0.576	0.337	0.769	0.940	0.935	0.279	0.527	0.309	0.810	0.945	0.940
0.5	0.318	0.564	0.325	0.779	0.941	0.935	0.280	0.529	0.305	0.809	0.944	0.940
0.4	0.307	0.554	0.322	0.788	0.941	0.937	0.287	0.536	0.314	0.803	0.942	0.938
0.3	0.315	0.561	0.333	0.781	0.939	0.933	0.279	0.529	0.313	0.809	0.944	0.939
0.2	0.312	0.558	0.324	0.785	0.940	0.935	0.282	0.531	0.309	0.807	0.944	0.940
0.1	0.296	0.544	0.314	0.796	0.941	0.936	0.288	0.537	0.315	0.804	0.944	0.940
LASCon	Tanh	2.0										
1.0	0.318	0.564	0.323	0.778	0.941	0.936	0.286	0.534	0.306	0.804	0.945	0.940
0.9	0.321	0.567	0.329	0.776	0.940	0.935	0.280	0.529	0.313	0.810	0.944	0.940
0.8	0.343	0.586	0.340	0.758	0.938	0.932	0.285	0.534	0.311	0.806	0.944	0.939
0.7	0.315	0.562	0.330	0.781	0.941	0.935	0.282	0.531	0.311	0.806	0.945	0.940
0.6	0.333	0.577	0.334	0.766	0.939	0.933	0.287	0.535	0.313	0.802	0.946	0.942
0.5	0.359	0.599	0.353	0.745	0.937	0.930	0.280	0.529	0.308	0.809	0.944	0.940
0.4	0.318	0.563	0.327	0.778	0.941	0.934	0.281	0.530	0.309	0.807	0.943	0.939
0.3	0.321	0.566	0.334	0.776	0.941	0.935	0.282	0.531	0.313	0.807	0.945	0.941
0.2	0.305	0.552	0.323	0.790	0.942	0.936	0.281	0.531	0.311	0.807	0.944	0.940
0.1	0.306	0.554	0.323	0.788	0.941	0.936	0.288	0.536	0.312	0.803	0.944	0.940

E.3 CMU-MOSI

Table 17: Multimodal contrastive learning on CMU-MOSI with diverse loss functions and temperatures.

				IN					OUT							
T	MAE(↓)	Corr(↑)	Acc-7(↑)	Acc-5(↑)	Acc-2(↑)		F1(↑)		MAE(↓)	Corr(↑)	Acc-7(↑)	Acc-5(↑)	Acc-2(↑)		F1(↑)	
NT-Xent																
1.0	0.718	0.792	45.675	51.312	82.750	84.400	82.717	84.420								
0.9	0.746	0.785	44.509	49.903	82.362	84.299	82.311	84.310								
0.8	0.718	0.801	45.530	51.166	82.410	84.248	82.328	84.226								
0.7	0.707	0.800	46.550	52.964	82.410	84.197	82.321	84.167								
0.6	0.771	0.778	43.052	48.348	81.730	83.435	81.711	83.479								
0.5	0.777	0.778	42.711	47.619	81.924	83.537	81.937	83.603								
0.4	0.810	0.764	41.448	46.842	80.466	81.911	80.478	81.987								
0.3	0.710	0.801	46.161	52.381	82.410	84.248	82.311	84.211								
0.2	0.718	0.794	45.675	51.749	83.771	85.467	83.669	85.417								
0.1	0.712	0.799	46.307	52.235	82.750	84.502	82.716	84.522								
SupCon																
1.0	0.712	0.798	45.675	51.701	82.459	84.146	82.428	84.170	0.709	0.801	46.793	52.527	82.702	84.248	82.664	84.261
0.9	0.719	0.795	45.773	51.361	82.993	84.604	82.953	84.614	0.707	0.803	45.821	52.138	82.993	84.909	82.939	84.912
0.8	0.712	0.798	46.259	52.284	82.750	84.451	82.710	84.464	0.708	0.802	45.294	51.562	83.195	84.996	83.119	84.975
0.7	0.718	0.798	46.550	51.944	82.507	84.146	82.466	84.158	0.709	0.802	46.987	52.478	83.382	84.858	83.374	84.896
0.6	0.718	0.794	46.453	52.478	82.945	84.654	82.871	84.635	0.711	0.801	45.481	50.729	82.604	84.350	82.552	84.354
0.5	0.720	0.794	45.724	51.895	82.702	84.654	82.637	84.651	0.705	0.801	46.842	52.381	82.702	84.502	82.625	84.484
0.4	0.721	0.795	44.752	50.729	82.459	84.146	82.437	84.179	0.704	0.804	46.696	52.915	82.556	84.451	82.476	84.433
0.3	0.720	0.798	45.578	51.118	82.604	84.096	82.565	84.106	0.705	0.803	45.287	51.312	82.702	84.502	82.646	84.502
0.2	0.713	0.800	45.092	50.826	83.382	85.112	83.300	85.084	0.711	0.799	46.064	51.846	82.653	84.248	82.590	84.237
0.1	0.707	0.805	45.627	51.603	83.625	84.909	83.622	84.948	0.708	0.801	46.647	52.332	83.236	84.807	83.200	84.820
LASCon	Linear															
1.0	0.708	0.801	46.113	52.527	83.188	84.909	83.131	84.905	0.734	0.799	43.732	49.660	81.778	83.079	81.774	83.125
0.9	0.717	0.800	45.578	51.992	83.139	84.909	83.090	84.913	0.729	0.797	45.287	51.312	81.438	82.927	81.427	82.971
0.8	0.714	0.799	45.335	51.652	83.042	84.451	82.996	84.453	0.712	0.799	46.696	53.013	82.362	84.096	82.292	84.083
0.7	0.716	0.796	46.307	52.624	82.119	83.740	82.076	83.752	0.703	0.802	46.939	52.575	82.604	84.248	82.572	84.270
0.6	0.728	0.792	43.975	50.486	81.876	83.486	81.841	83.506	0.714	0.797	46.259	52.332	82.216	83.994	82.171	84.006
0.5	0.713	0.799	45.675	51.798	83.042	84.654	82.984	84.648	0.721	0.796	45.432	51.409	82.021	83.689	81.940	83.665
0.4	0.722	0.795	44.801	50.923	82.653	83.841	82.641	83.872	0.723	0.797	44.898	50.729	81.924	83.689	81.848	83.673
0.3	0.711	0.801	45.384	51.701	82.507	84.197	82.445	84.191	0.727	0.799	44.849	51.020	82.167	83.638	82.172	83.694
0.2	0.719	0.792	45.578	52.041	82.847	84.248	82.805	84.252	0.734	0.794	44.655	50.680	81.633	83.283	81.611	83.319
0.1	0.728	0.796	44.534	50.802	82.362	83.613	82.400	83.701	0.738	0.791	44.558	50.632	81.633	83.232	81.608	83.262
LASCon	Tanh	0.5														
1.0	0.706	0.804	45.287	51.555	83.139	85.163	83.080	85.165	0.718	0.798	45.384	51.215	82.313	83.740	82.266	83.742
0.9	0.715	0.798	45.821	52.235	82.702	83.841	82.683	83.868	0.744	0.789	43.537	49.174	81.098	82.419	81.075	82.447
0.8	0.709	0.798	47.182	53.401	82.264	83.943	82.213	83.947	0.732	0.791	44.849	50.632	81.438	82.876	81.396	82.887
0.7	0.729	0.793	44.169	50.292	82.070	83.740	82.045	83.770	0.728	0.791	43.197	49.271	81.195	82.927	81.122	82.915
0.6	0.708	0.800	46.064	52.041	82.993	84.705	82.935	84.701	0.738	0.794	43.246	48.737	81.584	82.927	81.558	82.952
0.5	0.710	0.800	46.404	52.575	83.382	85.010	83.365	85.043	0.733	0.792	43.878	49.903	81.244	82.774	81.184	82.770
0.4	0.737	0.791	43.732	49.927	82.289	83.537	82.291	83.588	0.735	0.794	43.732	49.271	81.851	83.232	81.851	83.283
0.3	0.722	0.794	44.655	51.603	82.896	84.553	82.860	84.571	0.753	0.783	43.246	49.368	80.807	82.317	80.805	82.373
0.2	0.706	0.802	46.016	52.235	82.556	84.096	82.511	84.101	0.723	0.794	44.218	50.049	81.924	83.587	81.827	83.548
0.1	0.718	0.800	44.995	51.263	82.653	84.096	82.640	84.132	0.720	0.792	44.606	51.069	82.313	83.892	82.282	83.913
LASCon	Tanh	1.0														
1.0	0.719	0.792	45.287	51.652	81.827	83.181	81.804	83.207	0.709	0.803	45.384	51.361	82.896	84.400	82.856	84.409
0.9	0.711	0.801	45.578	51.555	82.847	84.400	82.826	84.431	0.712	0.798	46.307	52.721	82.945	84.705	82.895	84.710
0.8	0.711	0.801	45.190	51.020	82.896	84.400	82.846	84.398	0.715	0.799	45.918	52.041	82.459	83.943	82.412	83.946
0.7	0.709	0.803	44.801	50.777	82.507	84.146	82.431	84.125	0.718	0.797	44.947	50.389	82.604	84.146	82.575	84.168
0.6	0.726	0.794	44.558	50.923	81.876	83.587	81.833	83.602	0.714	0.804	45.627	51.409	82.556	84.350	82.495	84.346
0.5	0.711	0.795	45.773	52.964	82.945	84.604	82.905	84.616	0.719	0.799	45.190	51.020	82.021	83.689	81.988	83.712
0.4	0.721	0.796	43.367	49.854	83.163	85.061	83.066	85.024	0.729	0.797	45.238	50.826	81.681	83.283	81.615	83.273
0.3	0.736	0.784	44.169	50.486	81.584	82.978	81.591	83.036	0.732	0.796	43.635	49.903	81.973	83.537	81.936	83.553
0.2	0.737	0.792	44.606	51.312	82.070	83.486	82.060	83.528	0.728	0.792	44.558	50.194	81.147	82.825	81.111	82.849
0.1	0.711	0.793	46.501	52.672	82.459	84.096	82.395	84.086	0.728	0.795	43.440	49.417	81.438	83.079	81.395	83.093
LASCon	Tanh	2.0														
1.0	0.703	0.805	45.287	51.603	82.896	84.604	82.814	84.576	0.706	0.803	44.995	51.312	83.333	85.112	83.276	85.109
0.9	0.712	0.799	45.821	51.992	81.924	83.841	81.856	83.837	0.705	0.805	45.870	51.798	82.556	84.197	82.505	84.200
0.8	0.717	0.798	45.432	51.312	83.090	84.451	83.036	84.440	0.711	0.800	45.675	51.701	82.750	84.451	82.724	84.478
0.7	0.711	0.802	45.675	52.187	83.382	85.112	83.324	85.108	0.713	0.797	46.939	52.964	82.604	84.350	82.548	84.350
0.6	0.728	0.787	46.599	53.158	82.216	83.638	82.186	83.658	0.710	0.804	45.530	51.603	82.702	84.299	82.676	84.325
0.5	0.730	0.794	44.704	51.069	82.313	83.740	82.276	83.753	0.712	0.798	45.870	51.798	82.604	84.299	82.557	84.305
0.4	0.729	0.787	45.724	52.041	82.362	83.841	82.331	83.862	0.718	0.799	45.700	51.676	82.289	83.765	82.269	83.796
0.3	0.721	0.799	45.044	51.215	83.431	85.315	83.386	85.326	0.725	0.797	44.801	50.923	82.459	83.943	82.440	83.975
0.2	0.711	0.801	46.307	52.478	82.896	84.451	82.841	84.445	0.725	0.795	45.287	51.458	82.216	83.689	82.140	83.665
0.1	0.730	0.790	44.849	50.534	82.945	84.350	82.919	84.371	0.726	0.792	45.141	51.458	81.778	83.232	81.777	83.282