RETHINKING OF METRIC BASED CONTRASTIVE LEARNING METHOD'S GENERALIZATION CAPABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, semi-supervised/self-supervised methods based on contrastive learning have made great empirical progress in various fields of deep learning, and even outperform supervised methods in some fields (such as NLP and CV). However, there are very few theoretical works that may explain why the model trained using contrastive learning-based methods can outperform the model trained in general supervised methods on supervised tasks. Based on the manifold assumption about the input space, this work proposes three elements of metric-based contrastive learning:(1) Augmented neighborhood defined for every point in the input space (2) Metric-based optimization loss on the output space. (3) Generalization error on the union of the augmented neighborhood. Moreover, we propose an upper bound of (3) named UBGEAN(Upper Bound of Generalization Error on Augmented Neighborhood) which relate to labeled empirical loss and unlabeled metric-based contrastive loss. We also explain the relationship between the existing contrastive semi-supervised/self-supervised methods and our upper bound. Finally, based on it, we propose a supervised consistent contrastive learning method based on this upper bound. we verify the validity of the UBGEAN's generalization capacity against empirical loss by conducting a series of experiments and achieving an 8.2275% improvement on average in 4 tasks. Also, we design another set of experiments to verify the fine-tuning of the self-supervised training model of contrast learning, and it shows that our upper bound can provide a more stable effect to make the self-supervised pre-trained model of contrast learning achieve the effect of supervised pre-training model.

1 INTRODUCTION

Semi-supervised/self-supervised representation learning algorithms based on contrastive learning have developed rapidly in recent years, emerging from NLP(Devlin et al. (2019),Fang et al. (2020),Giorgi et al. (2021),Schick & Schütze (2020)), and then sweeping through computer vision(Liu et al. (2019),Chen et al. (2020),He et al. (2020),Grill et al. (2020),Chen & He (2021),Laine & Aila (2017)), recommendation algorithm(Yang et al. (2022),Xia et al. (2021)) and other related fields. They have attracted wide attention due to their low cost of data learning and high generalization performance. Moreover, combined with transfer learning, the models trained by semi-supervised/self-supervised representation learning methods based on contrastive learning achieve good empirical results in downstream tasks, and even outperform supervised methods in some fields.

However, at the same time, there is a lack of good theoretical work to explain how semi-supervised/self-supervised training methods based on contrastive learning affect the generalization performance of machine learning models in downstream supervised task. While in the existing few theoretical analysis work, we notice that most of the work on self-supervised contrastive learning is constructed on the definition of the feature metric learning problem(Huang et al. (2021),Wang & Isola (2020)). We believe that under this definition, this kind of analysis can only explain the generalization ability of the agent task itself based on metric learning. But it doesn't explain how the models that trained on the contrastive learning methods affect the generalization performance of the downstream supervised task (such as image classification). For example, in the field of CV self-supervised learning (BYOL(Grill et al. (2020)), SimSiam(Chen & He (2021)) et al.), how does the fine-tuning performance of the model trained by their methods catch up with or even surpass the model obtained by supervised learning after it is transferred to the supervised task? In the semi-

supervised domain, how do some excellent semi-supervised contrastive learning algorithms (Sohn et al. (2020))achieve good generalization results? These are all examples where the existing theoretical work cannot explain how the contrastive learning training methods affect the generalization performance of the model.

In order to figure out the impact of metric-based contrastive learning methods on generalization errors, this work takes the methods in CV as an example to analyze most self-supervised/semi-supervised representation learning methods based on contrast learning and extracts three elements that these algorithm designs follow in common: (1). Data augmentations that maintain semantic invariance. (2). Metric-based contrastive loss defined on representation space. (3). The empirical loss of labeled data should be used to participate in the optimization when allocating the model to train some supervised tasks(Self-supervise learning use label during finetune stage). And these elements will be discussed in more detail using mathematical theory. Based on the assumption that the input space is manifold, from the perspective of contrastive learning, we show a rigorous definition of generalization error on sample space with data augmentation. Combined with the proposed theory, we reconstructed the previously extracted corresponding elements as follows: (1). Data augmentation in contrastive learning induces an augmentation neighborhood $\mathcal{O}^*_{x_j}$ on the manifold for each sample point. (2). Metric-based optimization loss L on output space. (3). Generalization error: $\int_{\bigsqcup_{j=1}^{n} \mathcal{O}^*_{x_j}} L(f(x), F(x))p(x)\mathrm{d}x$ that is closer to the real input space and defined on the augmented neighborhood. And finally we prove an upper bound named UBGEAN(UBGEAN (Upper Bound of Generalization Error on Augmented Neighbourhood) of the generalization error on the union of augmentation neighborhoods, which is related to the empirical loss and the contrastive loss.

Our contribution can be summarized in the following three points:

- We define the generalization error $\int_{O^*_{x_j}} L(f(x), F(x))p(x)\mathrm{d}x$ on the data-induced augmentation neighborhood $\mathcal{O}^*_{x_j}$ more precisely from the mathematical point of view.
- We prove an upper bound UBGEAN of the generalization error on the union of augmentation neighborhoods, which is related to the empirical loss and the contrastive loss. And we explain the relationship between the upper bound and existing self-supervised/semi-supervised methods through theoretical analysis.
- Based on UBGEAN, we propose a consistent supervised contrastive learning method with better generalization ability. And then we experimentally verify that for the transfer of the self-supervised contrastive learning algorithm, using UBGEAN to finetune the loss will make the fine-tuning effect have a more stable improvement and exceed the supervised pre-trained model.

2 Related work

Semi Supervised Contrastive Learning A recent line of deep semi-supervised learning algorithms(Laine & Aila (2017)Verma et al. (2022)Zhang et al. (2018)Xie et al. (2020))are designed based on a simple concept that, if a proper augmentation was to be applied to an unlabeled example, the prediction should not change significantly.Therefore, with this concept, loss function can be designed to enforce model to have a consistent prediction on an unlabeled data and any of its perturbed version and to have the same prediction on labeled data and their label at the same time. Our work reveals that the essence of this kind of algorithm is to estimate and optimize two different parts of an upper bound of the generalization error of the model with inconsistent samples, and reveals the reason why the model using this kind of algorithm can achieve good generalization effect according to our theory. We also verified the influence of the sample inconsistency on the generalization ability of the model through experiments.

Self Supervised Contrastive Learning Early works such as MoCo(He et al. (2020)) and SimCLR (Chen et al. (2020)), use loss like InfoNCE to pull the positive samples together while enforcing them away from the negative samples in the embedding space. This need of negative samples requires large memory bank, which is expensive, and properly designed strategies to produce negative samples. Some recent work like BYOL(Grill et al. (2020)) and SimSiam(Chen & He (2021)) has proposed algorithms that do not require negative sample for training, and achieved better results than previous models that require negative samples. In our work, we point out that the normal form of using positive samples to construct contrast loss for pre-training and fine- tuning on downstream

tasks is actually to estimate and decoupled optimize two terms of an upper bound of the model's generalization error with inconsistent data.

3 THEORY

3.1 AUGMENTED NEIGHBOURHOOD

In order to state our result, we'd like to begin by introducing some basic notations and common assumptions. We use $M_n^{i_n}$ to denote the n^{th} compact manifold of dimension n, and let the input space $X = M_1^{i_1} \sqcup M_2^{i_2} \sqcup \cdots \sqcup M_k^{i_k}$ be a disjoint union of k compact manifold. We also set our model a continuous map from the input space to a unit hypersphere in the $m-$dimension space, that is, $f \in \mathcal{H} : X \to \mathbb{S}^{m-1}$. The output space of f is usually called embedding space when it comes to the representation learning. The real map $F : X \to \mathcal{S}^{m-1}$, known as conception, is also continuous. The image of the input space under F is denoted as $\mathcal{X}$, and this continuity of F allows us to explore the potential consistency between X and $\mathcal{X}$.

We consider the classification task as our downstream task and denote it as $\mathcal{T}$. The labeling function of task $\mathcal{T}$ here is a surjective continuous function $g_{\mathcal{T}} : \mathcal{X} \to \mathcal{Y} = \{l_1, \ldots, l_d\}$. $\mathcal{Y}$ is a finite space called label space when given the discrete topology, where every single point set in $\mathcal{Y}$ is both an open set and a closed set. For $\mathcal{T}$, the embedding space we consider is actually a subset of the real embedding space $\mathcal{X}$ and is denoted as $\mathcal{X}_{\mathcal{T}}$. We assume that $\mathcal{X}_{\mathcal{T}}$, the features extracted by F, can be written as a finite collection of closed disjoint sets $\mathcal{X}_1^{\mathcal{T}}, \ldots, \mathcal{X}_d^{\mathcal{T}}$, among which $\mathcal{X}_i^{\mathcal{T}} = g_{\mathcal{T}}^{-1}(l_i) := \{F(x) \in \mathcal{X} | g_{\mathcal{T}}(F(x)) = l_i\}$ is a close set. As a result, $g_{\mathcal{T}}$ induces a partition on $\mathcal{X}_{\mathcal{T}}$ as is shown in 1.

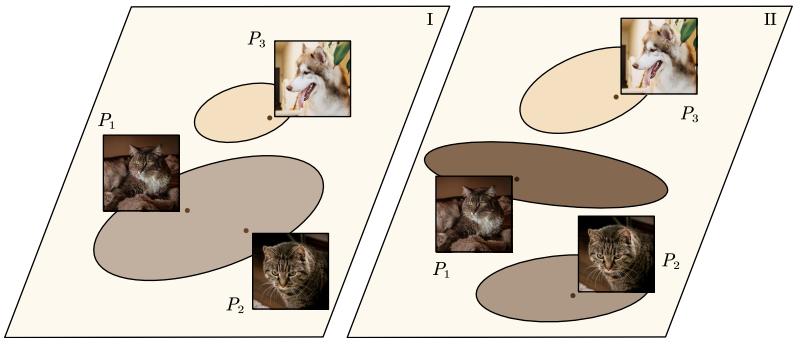


Figure 1: Different tasks induces different partition on embedding space. Task corresponding to the picture on the left is to classify cats and dogs, while the right one require to classify short haired cat, long haired cat and dog

As for the instance discrimination task, the agent task for contrastive learning paradigms also induces a partition on $\mathcal{X}_{\mathcal{T}}$. Since every sample in our training set $\mathcal{D} = \{x_i\}_{i=1}^n$ is the label of its own class, the label space of this task is $\mathcal{Y} = \{l_{x_1}, \ldots, l_{x_n}\}$. We denote by $A_i : X \to X$ to one type of augmentation operation, then the collection of augmentation used during the training is $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$. Therefore, the actual training set is $\mathcal{D}_{\mathcal{A}} = \mathcal{D} \cup \mathcal{A}(\mathcal{D})$, and we denote $g : \mathcal{X} \to \mathcal{Y}$ as the label function of the instance discrimination task. So $\mathcal{X}_{x_j}^{\mathcal{A}} = g^{-1}(l_{x_j})$ actually gives a closed neighborhood of $F(x_j)$, containing all the images of the samples under F which are generated by applying $\mathcal{A}$ to sample points x_j. On the embedding space, for every $F(x_j)$, we define a distribution $p(y|F(x_j))y \in \mathcal{X}$ "around" it, whose value is proportional to the distance between y and $F(x)$(See more detail in Appendix A.1). Then by $p(y|F(x_j))$, for any given $\tau \in [0, 1]$, we have a closed ball $B(F(X_j), \delta_{x_j})$, such that $\forall F(x) \in \overline{\mathcal{B}(F(x_j), \delta_{x_j})} \Rightarrow p_{\mathcal{A}}(y|F(x_j)) \geq \tau$. This closed ball gives an area where our model is "confident" enough to recognize all points in the same class, and it also represents an area with density that is high enough. So now we introduce our first assumption:

Assumption 1 (The Smoothness Assumption)**.** *If two points x_1, x_2 resided in a high-density region are close, then so should be their corresponding outputs y_1, y_2, and vice versa.*

Clearly the close ball mentioned above describes an area satisfying the smoothness assumption. And every two points in it will be classified into the same class under g or other proper classifier. In addition, the points in the closed ball are also considered to be relatively easy to learn for our model, and we often make f only learn these points instead of letting it learn those points after adversarial data augmentation. Because they are so bad that even if the features are perfectly extracted, our classifier's "confidence" in its classification results is often insufficient or even unable to get the correct results.

With $S_{\delta_{x_j}} = \{x \in X | F(x) \in \overline{\mathcal{B}(F(x_j), \delta_{x_j})}\}$, we naturally derive the definition of **augmented neighborhood**:

Definition 1. *For the collection of data augmentation $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$, we name the following set:*

$$\bigcup_{i=1}^{n} \{A_i(X) \bigcap S_{\delta_{x_j}}\} \tag{1}$$

as the augmented neighborhood of x_j, and denoted it as $\mathcal{O}_x^*$. As a visualization, please see 2, A.2 provide an example to help to build intuition.

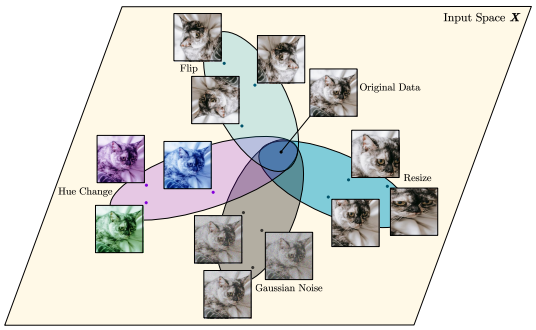


Figure 2: A Visualization for Augmented Neighborhood

3.2 Upper Bound of Generalization Error of Augmented Neighborhood

After defining the basic set (augmented neighborhood) clearly for modeling the contrastive learning instance discrimination task, we now consider the generalization error on this set.

When studying the generalization error of machine learning models, we first need to define the loss function. According to the characteristic that the common loss functions used in current contrastive learning work(MSE, cosine similarity, L2 norm, etc.) are all essentially metric loss, in this work metric loss is denoted as L, which implies positivity,symmetry and triangle inequality.

Now we generalize the output space of the model f mentioned above from the embedding space to any other output space, and F also changes accordingly. Given some arbitrary samples $D = \{x_i\}_{i=1}^n$ drawn from the sample space X_D which is the subset of the real input space X, for each point in the space, if we consider its augmented neighborhood, we get the relation between sample space and the union of augmented neighborhoods:$X_D \subset \bigsqcup_{x \in X_D} \mathcal{O}_x^*$.

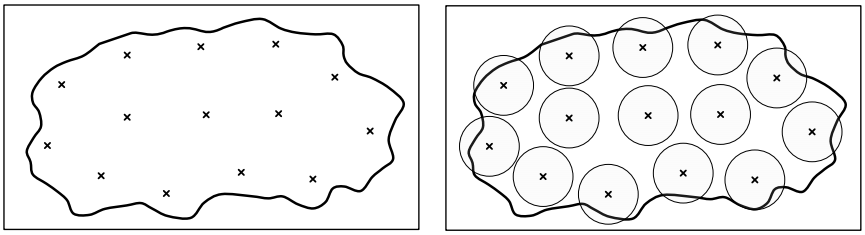
Definition 2. *For the sample space X_D, we call the generalization error on the union of augmented neighborhood of each point GEAN(Generalization error based on the augmented neighborhood), which is defined as the following equation:*

$$\int_{\bigsqcup_{x \in X_D} \mathcal{O}_x^*} L(f(x), F(x)) p(x) \, \mathrm{d}x \tag{2}$$

$p(x)$is a probability density function over X. We assume that the $p_D(x)$ is the probability density function on X_D, it meets $p_D(x) = \frac{p(x)}{\int_{X_D} p(x) \, \mathrm{d}x}$. Combining the relation above:$X_D \subset \bigsqcup_{x \in X_D} \mathcal{O}_x^*$

,we give the following inequality:

$$\int_{X_D} L(f(x), F(x))p_D(x)\,\mathrm{d}x \leqslant \int_{X_D} p(x)\,\mathrm{d}x \cdot \int_{X_D} L(f(x), F(x))p(x)\mathrm{d}x \leqslant \int_{X_D} L(f(x), F(x))p(x)\,\mathrm{d}x \leqslant \int_{\bigsqcup_{x\in X_D} \mathcal{O}_x^*} L(f(x), F(x))p(x)\,\mathrm{d}x \leqslant \int_X L(f(x), F(x))p(x)\mathrm{d}x \tag{3}$$



(a) Sample Space (b) Augmented Neighborhood

Figure 3: Sample Space and Augmented Neighborhood of Samples

Actually,the left term of this inequality is the generalization error on the sample space X_D, which is a traditional form of error. While the middle term is the generalization error on the disjoint union of the augmented neighborhood of each point in the sample space X_D we proposed. And the rightmost term is the generalization error on the entire input space, it contains more than one sample space like X_D. According to Equation (3), if we consider GEAN, it is not only a better approximation of the generalization error on the real input space, compared with the traditional loss, but also enables the model f to learn the property of local continuity of F in the augmented neighborhood as expected.

Instead of considering the generalization error on the whole sample space, we often concentrate on the generalization error defined on the union of the augmented neighborhood of each point in the sample $D = \{x_j\}_{j=1}^n$, which is:

$$\int_{\bigsqcup_{j=1}^n \mathcal{O}_{x_j}^*} L(f(x), F(x))p(x)\,\mathrm{d}x \tag{4}$$

In the following, we default to discuss GEAN as (4). With some necessary assumption(see appendix A.3), it is equvialent to:

$$\sum_{j=1}^{n} \int_{\mathcal{O}_{x_j}^*} L(f(x), F(x))p_{x_j}(x)\,\mathrm{d}x \tag{5}$$

Where $p_{x_j}(x) := \frac{p(x)}{\int_{\mathcal{O}_{x_j}^*} p(x)\,\mathrm{d}x}$.Therefore, we find that GEAN is actually equivalent to a multi-targets learning problem on the augmented neighborhoods of many sample points. That is, GEAN is a multi-objective optimization problem by simultaneously minimizing the generalization error on $\mathcal{O}_{x_j}^*(j = 1, 2, \cdots n)$.

Theorem 1.

$$C + \sum_{j=1}^{n} \left(\int_{\mathcal{O}_{x_j}^*} L(F(x_j), f(x_j))p_{x_j}(x)\,\mathrm{d}x + \int_{\mathcal{O}_{x_j}^*} L(f(x), f(x_j))p_{x_j}(x)\,\mathrm{d}x \right) \tag{6}$$

is an upper bound of the generalization error (4), which is called UBGEAN (upper bound of generalization error on augmented neighborhood) for short.In which $C = \sum_{j=1}^{n} (\int_{\mathcal{O}_{x_j}^*} L(F(x), F(x_j))p_{x_j}(x)\mathrm{d}x$

(see proof in A.5)

This inequality actually gives an upper bound of the generalization error defined on the union of augmented neighborhoods. More importantly, this upper bound reveals the profound nature of semi-supervised and self-supervised learning based on metric-contrast loss. Let's explain what each item in UBGEAN means:

For the optimization problem with related to the parameter of f, the first term $\sum_{j=1}^{n} \int_{\mathcal{O}^*_{x_j}} L(F(x), F(x_j)) p_{x_j}(x)\,\mathrm{d}x$ of UBGEAN does not contain term f, so this term is regarded as a negligible constant.

For the second term $\sum_{j=1}^{n} \int_{\mathcal{O}^*_{x_j}} L(F(x_j), f(x_j)) p_{x_j}(x)\,\mathrm{d}x$ of UBGEAN, according to the definition of the distribution on each augmented neighborhood, it satisfies $\int_{\mathcal{O}^*_{x_j}} p_{x_j}(x)\,\mathrm{d}x = 1$ $(j = 1, 2, \cdots .)$. Therefore, we can deduce the following equation:

$$\begin{aligned}\sum_{j=1}^{n} \int_{\mathcal{O}^*_{x_j}} L(F(x_j), f(x_j)) p_{x_j}(x)\,\mathrm{d}x &= \sum_{j=1}^{n} L(F(x_j), f(x_j)) \int_{\mathcal{O}^*_{x_j}} p_{x_j}(x)\,\mathrm{d}x \\ &= \sum_{j=1}^{n} L(F(x_j), f(x_j))\end{aligned} \tag{7}$$

This is the sum of the metric loss between the output of the model for each data point and the its label (or conception coding). And the estimation of the generalization error on the sample space X_D is $\sum_{j=1}^{n} L(F(x_j), f(x_j))\mathrm{d}x$, so the second term is equivalent to the general empirical loss.

The third term $\sum_{j=1}^{n} \int_{\mathcal{O}^*_{x_j}} L(f(x), f(x_j)) p_{x_j}(x)\,\mathrm{d}x$ represents the sum of expectations of the metric loss between each sample point and its augmented neighborhood, which measures the continuity of the model on augmented neighborhood of each sample. Under the smoothness assumption, all existing metric-based contrastive loss using only positive samples (such as SimSiam and Byol) is equivalent to the third term. In particular, we take SimSiam as an example to explain the reasons. SimSiam takes $\mathbb{E}_{x,\mathcal{T}}\left[\|\mathcal{F}_\theta(\mathcal{T}(x)) - \mathcal{F}_{\theta^t}(\mathcal{T}'(x))\|_2^2\right]$ as the loss function to update parameters, which can be regarded as the third term $\mathbb{E}_x\left[\int_{\mathcal{O}^*_{\mathcal{T}'(x)}} L(f(x), f(\mathcal{T}'(x))) p_{\mathcal{T}'(x)}(x)\,\mathrm{d}x\right]$ of UBGEAN on the augmented neighborhood $\mathcal{O}^*_{\mathcal{T}'(x)}$ induced by $\mathcal{T}'(x)$, and its empirical estimation is a Monte Carlo estimation of $\int_{\mathcal{O}^*_{\mathcal{T}'(x)}} L(f(x), f(\mathcal{T}'(x))) p_{\mathcal{T}'(x)}(x)\,\mathrm{d}x$. In addition, the existing common semi-supervised work using metric based contrastive loss(such as PI-Model, UDA, FixMatch45) all use MSE loss for their unsupervised contrastive loss, which is also equivalent to the third term here.

4 Experiment And Conclusion

In this section, we are going to directly state the connection between UBGEAN we proposed and semi-supervised and self-supervised learning methods using metric based loss.

4.1 Consistent Upper Bound

According to our theory, directly optimizing UBGEAN yields model with better generalization ability, since the first term of UBGEAN is independent of the model, what we are really need in optimization is:

$$\sum_{j=1}^{n} \left(\int_{\mathcal{O}^*_{x_j}} L(F(x_j), f(x_j)) p_{x_j}(x)\,\mathrm{d}x + \int_{\mathcal{O}^*_{x_j}} L(f(x), f(x_j)) p_{x_j}(x)\,\mathrm{d}x \right) \tag{8}$$

As a computable approximation of this objective function, we usually consider its monte carlo estimation:

$$\sum_{j=1}^{n}\left(L(F(x_j),f(x_j))+\frac{1}{N}\sum_{i=1}^{N}L(f(x_j^i),f(x_j))\right) \tag{9}$$

where $x_j^i \sim p_{x_j}(x)$(Sampling from the augmented neighborhood is to randomly augment the data). And for computational efficiency, we set $N = 1$. And the pseudo code is following:

Algorithm 1 pseudo code here

```
# aug: augmentation method

for x, label in loader:
    if UBGEAN != 0: # >=1 1 use UBGEAN loss
    # here UBGEAN is the number of samples.
        aug = aug(x)
        UBGEAN_loss += loss(aug, output)
    exp_loss = loss(model(x), label) # CosineSimilarity loss
    loss = exp_loss + beta * UBGEAN_loss
    loss.backward()
    update(model)
```

Experiment Set: We choose Resnet50 as our baseline models.We use cosine similarity as a metric loss after softmax (this is equivalent to do a metric learning by projecting the probability vector onto the unit hypersphere. Detailed proof is provided in the Appendix (A.4)), and we choose Adam as optimizer. we set the same augmentations as SimSiam(Chen et al. (2020)). See appendix B.1 for more detailed experimental settings.

Random Init	Food101	SVHN	CIFAR10	CIFAR100
Experience loss	57.32%	95.49%	86.45%	61.80%
UBGEAN	**74.82%**	**97.04%**	**93.37%**	**68.74%**

Table 1: Results of Random Initialization on each Dataset with the Standard ResNet-50 Architecture. We train randomly initialized ResNet-50 model with UBGEAN and empirical loss respectively under the same baseline on Food101 Bossard et al. (2014), SVHN Netzer et al. (2011), CIFAR-10 Krizhevsky et al. (2009) and CIFAR-100 Krizhevsky et al. (2009).

Main Result: We can tell from Table 1 that, compared with the general experience loss, our method has greatly improved the performance of the model, which indicates that UBGEAN, as a upper bound of GEAN, effectively controls the generalization error of the model and help it to achieve a better generalization effect. Meanwhile, we can see from figure 1 that, our method has always been ahead of the traditional method based on experience loss in the generalization ability of the whole learning process.Therefore, the theory we propose can offer a better generalization performance,and you can use it in any supervised learning scenario.

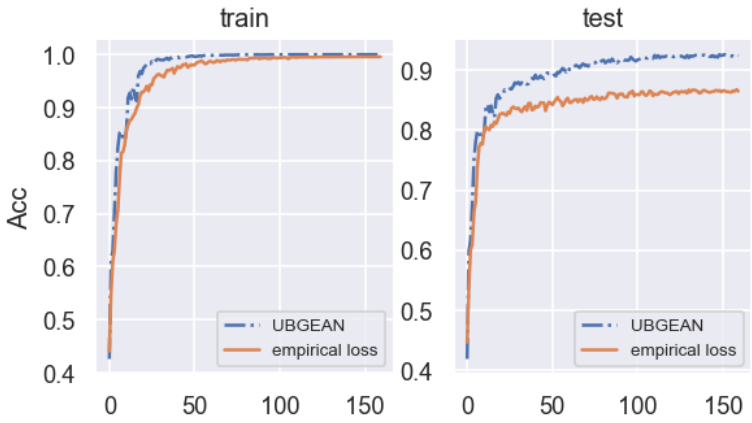


Figure 4: Random Init on CIFAR-10 with different loss

4.2 Inconsistent Estimation

Many representative semi-supervised methods containing contrastive loss(Sohn et al. (2020),Xie et al. (2020),Lee et al. (2013)), the final loss of these methods are usually made up of two parts, the first one is the unsupervised contrastive loss based on metric or cross entropy computed using unlabeled data, and the second one is a supervised loss based on cross entropy computed using labeled data. In order to generalize our theory, we need to find the equivalence between cross entropy(CE) loss and usual metric loss like squared error(SE) loss, it is shown in that under some condition that is good enough, we may assume that CE is equivalent to SE.

As a result, according to our theory, the existing semi-supervised methods is essentially decoupling UBGEAN in the estimation phase. They use labeled data$\mathcal{D}_l$ to estimate (8) or just the first term in it, and use unlabeled data to estimate second term of (8). This decoupling of estimates actually sacrifices the accuracy of estimates to reduce the dependence on labeled data.

The essential difference between these methods and ours is the consistency of data used when estimating (8). For the consistent estimation, which is ours, the data used when updating (8) comes from the same batch, while the inconsistent one use data from different batch or even different amount of data to estimate the two terms in (8) respectively.

4.3 Decoupled Optimization

For the self supervised transfer learning method, we think that it has performed a transfer learning for the third term of UBGEAN. The current measurement based comparative learning pre training method can be divided into two steps: first, learn the third term of UBGEAN on a large number of unlabeled data, and then transfer the model to the downstream task for fine-tuning with empirical loss.

According to our theory, the common self supervised learning paradigm essentially uses data from two different domains to decoupling optimize the second and third terms of UBGEAN(Where $\mathcal{D}_u$ and $\mathcal{D}_l$ represent two different domains respectively). After the pre train stage based on contrastive learning, if we transfer the model to a new domain, the third terms of the UBGEAN on this new domain $\sum_{x' \in \mathcal{D}_l} \int_{\mathcal{O}^*_{x'}} L(f(x'), f(x)) p_{x'}(x)\,\mathrm{d}x$ has a good starting point because of the pre training. After appropriate fine-tuning, the UBGEAN on this new domain will be smaller, which implies the improvement of model generalization performance. This explains why the pre training based on contrastive learning can effectively improve the generalization ability of the model.

Therefore, we believe that after fine-tuning with experience loss, the UBGEAN of the model that uses contrastive learning for pre training will have a smaller value than that of the model that uses supervised pre training.

Experiment set: We use the supervised pre trained model and SimSiam pre trained Resnet50.We use cosine similarity as a metric loss. After the same fine-tuning using experiential loss, we select the best models base on its performance on verification set. Then we use UBGEAN as a comparative indicators, since we use random augmentation for its estimation, we compute 20 times and take the average as an estimation for the real UBGEAN. See Appendix B.3 for detailed experimental settings.

Main Result: According to Table 2 below, the experimental results are consistent with our theoretical analysis. The SimSiam pre training model has a lower UBGEAN value after fine-tuning, which indicates that pre training method based on contrastive learning effectively controls the UBGEAN of the model after transferring into a new domain.

Table 2: Loss of differernt pre-trained models after fine-tuning with empirical loss

	Simsiam	Supervised
CIFAR10		
UBGEAN(2nd+3rd)	$-0.958 \pm 4.99 \times 10^{-6}$	$-0.974 \pm 4.99 \times 10^{-6}$
UBGEAN(3rd)	$-0.00826 \pm 4.98 \times 10^{-6}$	$-0.00594 \pm 3.06 \times 10^{-6}$
CIFAR100		
UBGEAN(2nd+3rd)	$-0.785 \pm 2.70 \times 10^{-6}$	$-0.840 \pm 2.45 \times 10^{-6}$
UBGEAN(3rd)	$-0.00408 \pm 2.70 \times 10^{-6}$	$-0.00259 \pm 2.45 \times 10^{-6}$

After this, We do a further exploration for the fine-tuning stage of the transfer learning problem.

Experiment set: We used Resnet50 fine tuned by SimSiam and supervised methods on ImageNet for comparative experiments. For the two pre trained models, we use empirical loss and UBGEAN to fine tune(i.e. 1, the augmentations are the same as SimSiam(Chen et al. (2020))). See Appendix B.4 for detailed experimental settings.

Table 3: Transfer Learning Results with the Standard ResNet-50 Architecture

Fine-tuned	Food101	SVHN	CIFAR10	CIFAR100	DTD	Pets	Aircraft
SimSiam							
Empirical loss	75.88%	96.50%	94.22%	76.23%	61.60%	78.25%	54.76%
UBGEAN	**81.62%**	**97.49%**	**96.46%**	**81.61%**	**67.77%**	**83.92%**	**70.12%**
Supervised							
Empirical loss	79.57%	96.57%	96.22%	82.52%	**69.95%**	89.81%	52.81%
UBGEAN	**82.96%**	**96.87%**	**96.94%**	**84.15%**	**69.95%**	**92.10%**	**70.30%**

Main results: The results in Table 3 indicate that when we use UBGEAN to fine tune the model pre trained on ImageNet with loss equivalent to the third term of UBGEAN and the model of supervised pre trained on ImageNet as well, we find out that the former has better generalization ability than the latter. That is to say, for all the model using unsupervised pre training methods whose loss equivalent to the third term of UBGEAN, using the same data augmentation to compute the consistent estimatation of UBGEAN in the fine-tuning stage will have a greater impact on the final result than the supervised pre training model.

REPRODUCIBILITY STATEMENT

The implementation code can be found in file submitted with the paper. All datasets and the code platform (PyTorch) we use are public. In addition, we also provide detailed experiment set and mathematical proof in the Appendix.

REFERENCES

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15745–15753, 2021. doi: 10.1109/CVPR46437.2021.01549.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. *ArXiv*, abs/2005.12766, 2020.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 879–895, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.72. URL https://aclanthology.org/2021.acl-long.72.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2020. doi: 10.1109/CVPR42600.2020.00975.

Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning. *ArXiv*, abs/2111.00743, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJ6oOfqge.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10550–10559, 2019. doi: 10.1109/ICCV.2019.01065.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2021.10.008. URL https://www.sciencedirect.com/science/article/pii/S0893608021003993.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *ArXiv*, abs/2005.10242, 2020.

Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. Self-supervised hypergraph convolutional networks for session-based recommendation. 35(5):4503–4511, 2021. doi: 10.1609/aaai.v35i5.16578.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pp. 1434–1443, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3532009. URL https://doi.org/10.1145/3477495.3532009.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

A Proof and Example

A.1 The distribution on embedding space

In piratical we usually consider $p(l_{x_i}|F(x_j))$, the output layer of a model for classification tasks, who gives a multinomial distribution implying the corresponding "confidence" that the model recognize $F(x_j)$ as each class which is the cosine similarity between the sample and the center of each class. Inspired by that we now define a distribution making $\mathcal{X}_{x_j}^{\mathcal{A}}$ the area with high density:

Definition 3. *We denote $p_{\mathcal{A}}(y|F(x_j))$ as the distribution making $\mathcal{X}_{x_j}^{\mathcal{A}}$ the area with high density, whose value is proportional to the cosine similarity between $y = F(x)\,, x \in \mathcal{X}_{x_j}^{\mathcal{A}}$ and $F(x_j)$, since $F(x), F(x_j) \in \mathbb{S}^{m-1}$, then the cosine similarity here is then the usual inner product of vector, and it is defined as followed:*

$$\begin{aligned} p_{\mathcal{A}}(y|F(x_j)) \propto \exp\{y^T F(x_j)\} = \exp\left\{1 - \frac{||y - F(x_j)||^2}{2}\right\} \propto \\ \exp\left\{-\frac{1}{2}(y - F(x_j))^T (y - F(x_j))\right\},\ \forall y \in \mathcal{X} \end{aligned} \tag{10}$$

And this actually actually gives the fact that $y|F(x_j) \sim \mathcal{N}(F(x_j), I)$.

In order to find an area with density that is high enough in $\mathcal{X}_{x_j}^{\mathcal{A}}$, we give the following theorem:

Theorem 2. *For any given $\tau \in [0, 1]$(we name it as "confidence level"), there exists $\delta_{x_j} > 0$ such that*

$$\forall F(x) \in \overline{\mathcal{B}(F(x_j), \delta_{x_j})} \rightrightarrows p_{\mathcal{X}_{x_j}^{\mathcal{A}}}(x) \geqslant \tau$$

Proof: We denote the normalization term of $p(y|F(x_j))$ as N, then the inequality we want is actually

$$\frac{p_{\mathcal{A}}(y|F(x_j))}{N} \geqslant \tau. \tag{11}$$

Now we have the following

$$\begin{aligned} \frac{p_{\mathcal{A}}(y|F(x_j))}{N}(x) \geqslant \tau \Leftrightarrow \exp\left\{1 - \frac{||y - F(x_j)||^2}{2}\right\} \geqslant \tau N \\ \left(2 \ln \frac{e}{\tau N}\right)^{\frac{1}{2}} \geqslant ||y - F(x_j)|| \end{aligned} \tag{12}$$

Then we complete the proof.

A.2 An Example for Augmented Neighborhood

As a way to comprehend this neighborhood, for example, consider a set of uniform noise augmentations $A_{\mathcal{U}}$ and an open neighborhood U_{x_j} of a sample x_j in input space, whose local coordinate representation is $h_{HWC} \cdot h_{\mathbb{M}_{H\times W\times C}}$. $h_{\mathbb{M}_{H\times W\times C}}$ actually refers to a method of storing pictures of certain objects in reality in an $H \times W \times C$ tensor form and using them as the input of the model. The subscript $\mathbb{M}_{H\times W\times C}$ of the reflection represents a space composed of all $H \times W \times C$ tensors, where H and W are adjusted to the needs while C is usually 3 since we're dealing with colored pictures. And h_{HWC} is a tile reflection from $\mathbb{M}_{H\times W\times C}$ to the vector space $\mathbb{R}^{HWC}$ with $H \cdot W \cdot C$ dimensions, which is

$$\begin{aligned} h_{HWC} : \mathbb{M}_{H\times W\times C} \to \mathbb{R}^{HWC} \\ h_{\mathbb{M}_{H\times W\times C}}(x_j) \mapsto \text{flatten}(h_{\mathbb{M}_{H\times W\times C}}(x_j)) \end{aligned} \tag{13}$$

After this mapping, now we're pondering the problem in a metric space $(\mathbb{R}^{HWC},\ \|\cdot\|_{L_2})$ based on the L_2-norm, because we map the open neighborhood U_{x_j} in the input space manifold X into the metric space $(\mathbb{R}^{HWC},\ \|\cdot\|_{L_2})$ under the local coordinate representation is $h_{HWC} \cdot h_{\mathbb{M}_{H\times W\times C}}$. In practical applications, based on the above local coordinate representation, there must be a correspondence like below, which makes every specific uniform noise augmentation $A_{\mathcal{U}}^y \in A_{\mathcal{U}}$ can uniquely correspond to a tensor in $\mathbb{M}_{H\times W\times C}$:

$$\begin{aligned} \forall A_{\mathcal{U}}^y \in A_{\mathcal{U}},\ \exists! M_y \in \mathbb{M}_{H\times W\times C} \\ \text{s.t.} \\ h_{\mathbb{M}_{H\times W\times C}}(A_{\mathcal{U}}^y(x_j)) = h_{\mathbb{M}_{H\times W\times C}}(x) +_{\text{pointwise}} M_y \end{aligned} \tag{14}$$

where the image $h_{HWC}(M_y)$ of M_y in fact is uniformly distributed in $B(O,\ 1) \subset (\mathbb{R}^{HWC},\ \|\cdot\|_{L_2})$. Let $\mathcal{U}_{B(O,\ 1)}$ be the uniform distribution and $p_{\mathcal{U}}$ be its probability density function. When a uniform

noise augmentation acts on the sample x_j, firstly M_y is produced, where $h_{HWC}(M_y) \sim p_{\mathcal{U}}(x)$. Then M_y is pointwise added in $h_{\mathbb{M}_{H\times W\times C}}(x_j)$. After that, we get the needed tensor. If we flatten this tensor, then it's easy to see that it is a sample from $\mathcal{U}_{B(h_{HWC}\cdot h_{\mathbb{M}_{H\times W\times C}}(x_j),\,1)}$. And when we traverse all uniform noise augmentations, or say, consider a set of tensors which are produced by all uniform noise augmentations acting on the sample x_j, we in fact get a hypersphere $B(h_{HWC}\cdot h_{\mathbb{M}_{H\times W\times C}}(x_j),\,1) \subset (\mathbb{R}^{HWC},\,\|\cdot\|_{L_2})$ under the local coordinate representation $h_{HWC}\cdot h_{\mathbb{M}_{H\times W\times C}}(\cdot)$ after flattening. By the homeomorphic property of the local representation map, we actually get an open neighborhood about x_j in the input space X, on which there is also an inherited uniform distribution.

A.3 Assumption for augmented neighborhood

Assumption 2. *For any two samples drawn from the same sample spaceX_D, we consider the integration on any set A:$\int_A p(x)\mathrm{d}x$, if we take A as their augmented neighborhood, we consider these two terms $\int_{\mathcal{O}^*_{x_i}} p(x)\,\mathrm{d}x,\ \int_{\mathcal{O}^*_{x_j}} p(x)\,\mathrm{d}x$ to be the same.*

This assumption is summarized from an intuition that since two samples are drawn from the same sample space, then we would like to expect their "behaviour" under the same augmentation is the same, and since augmented neighborhood can be explained as "the collection of the augmented sample that looks like the original one", then we may consider these two samples have the same "size" of augmented neighborhood.

Definition 4. *The pdf $p_{x_j}(x)$ defined on the augmented neighborhood$\mathcal{O}^*_{x_j}$ of any sample point x_j satisfies:*

$$p_{x_j}(x) = \frac{p(x)}{\int_{\mathcal{O}^*_{x_j}} p(x)\,\mathrm{d}x},\ \forall x \in \mathcal{O}^*_{x_j},\ \overline{\mathrm{supp}(x)} = \mathcal{O}^*_{x_j} \tag{15}$$

Since GEAN is defined on the disjoint union of several closed sets, combined with the probability density function we defined before on the augmented neigborhood of each sample point, GEAN has the following:

$$\int_{\bigsqcup_{j=1}^n \mathcal{O}^*_{x_j}} L(f(x),F(x))p(x)\,\mathrm{d}x = \sum_{j=1}^n \mathbb{P}(\mathcal{O}^*_{x_j}) \int_{\mathcal{O}^*_{x_j}} L(f(x),F(x))p_{x_j}(x)\,\mathrm{d}x \tag{16}$$

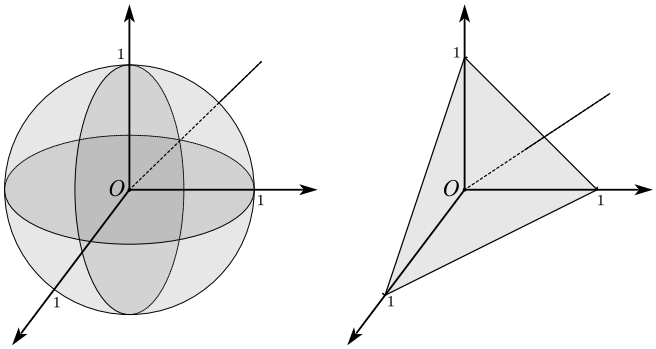
In other words, GEAN is the weighted sum of generalization error on the augmented neighborhood, whose weights are the measure of the corresponding augmented neighborhood of each sample point. With Assumption 2, we can ignore the influence caused by these weights.

A.4 Proof for cosine similarity

Injection proof of projection into hypersphere metric space

Theorem 3. *Cosine similarity loss $Sim(\cdot,\cdot)$ can be regarded as the objective function to project the model output results to the unit hypersphere for metric learning*

Proof: When the output layer of model f is softmax, all the output of the model lies in $\Delta^{n-1} = \left\{ x = (x_1, x_2, \cdots, x_n) \,\middle|\, \sum_{i=1}^n x_i = 1, x_i \geqslant 0 \right\}$, the $n-1$-dimensional simplex as it is shown in Figure 5.



(a) (b)

Figure 5: The Multidimensional Simplex

And normalized mapping $\phi(x) = \frac{x}{\|x\|}$, $\forall x \in R^n$, are mapped into $\mathbb{S}^{n-1} = \left\{(y_1,\ y_2,\ \cdots,\ y_n) \middle| \sum_{i=1}^{n} y_i^2 = 1\right\}$, the unit hypersphere in the n dimension space as it is shown in picture 2, which means that $\forall x \in R^n, \phi(x) \in \mathbb{S}^{n-1}$. Then $\forall x_a,\ x_b \in R^n$, if $\phi(x_a) = \phi(x_b)$, we have that $x_a = \frac{\|x_a\|}{\|x_b\|} x_b \Leftrightarrow x_a,\ x_b$ belongs to the same set $HF_a = \{x | x = \alpha \cdot x_a, \alpha \geqslant 0\}$.

Now let's prove that $\phi(x) : \Delta^{n-1} \to \mathbb{S}^{n-1}$ is an injection: If $x_a \neq x_b \in E, \phi(x_a) = \phi(x_b)$, then $x_a \neq x_b \in HF_a \Rightarrow x_a = \alpha \cdot x_b,\ \alpha \neq 1 \Rightarrow x_a = (x_{a1},\ x_{a2},\ \cdots,\ x_{an}) = \alpha \cdot x_b = (\alpha \cdot x_{b1},\ \alpha \cdot x_{b2},\ \cdots,\ \alpha \cdot x_{bn})$. Then since $x_a,\ x_b \in E \Rightarrow \sum_{i=1}^{n} x_{ai} = 1 \Rightarrow \alpha \cdot \sum_{i=1}^{n} x_{bi} = 1 \Rightarrow \alpha = 1$, then the contradiction arises, we finished the proof. Then $\phi(x)$ now an injection.

A unit hypersphere space with vector inner product $< \mathbb{S}^{n-1}, \cdot >$is a metric space, so cosine similarity loss can be regarded as the objective function of projecting the output of the model results to the unit hypersphere for metric learning.

A.5 Proof for our theorem

Theorem 4.

$$\sum_{j=1}^{n} \left(\int_{\mathcal{O}^*_{x_j}} L(F(x), F(x_j)) p_{x_j}(x)\,\mathrm{d}x + \int_{\mathcal{O}^*_{x_j}} L(F(x_j), f(x_j)) p_{x_j}(x)\,\mathrm{d}x + \int_{\mathcal{O}^*_{x_j}} L(f(x), f(x_j)) p_{x_j}(x)\,\mathrm{d}x \right) \tag{17}$$

is an upper bound of the generalization error (4), which is called UBGEAN (upper bound of generalization error on augmented neighborhood) for short.

Proof: Since L is a metric, according to the triangle inequality, for each sample pointx_j the following inequality holds:

$$L(f(x), F(x)) \leqslant L(F(x), F(x_j)) + L(F(x_j), f(x_j)) + L(f(x), f(x_j)) \tag{18}$$

Calculate the expectation of both sides simultaneously on the augmented neighborhood of sample point x_j, and sum both sides of the inequality concerning the index j, then the following inequality is obtained:

$$\begin{aligned} \sum_{j=1}^{n} \int_{\mathcal{O}^*_{x_j}} L(f(x), F(x)) p_{x_j}(x)\,\mathrm{d}x \leqslant \sum_{j=1}^{n} \Bigg(& \int_{\mathcal{O}^*_{x_j}} L(F(x), F(x_j)) p_{x_j}(x)\,\mathrm{d}x + \\ & \int_{\mathcal{O}^*_{x_j}} L(F(x_j), f(x_j)) p_{x_j}(x)\,\mathrm{d}x + \int_{\mathcal{O}^*_{x_j}} L(f(x), f(x_j)) p_{x_j}(x)\,\mathrm{d}x \Bigg) \end{aligned} \tag{19}$$

And that completes the proof.

B EXPERIMENTAL SETUP

B.1 DATASETS

Table 4: Dataset Partitioning

Dataset	Train examples	Test examples	Valid. examples	Classes	Evaluation criterion
Food101	68175	25250	7575	101	Top-1 accuracy
SVHN	65937	26032	7320	10	Top-1 accuracy
CIFAR-10	45000	10000	5000	10	Top-1 accuracy
CIFAR-100	45000	10000	5000	100	Top-1 accuracy
DTD	1880	1880	1880	47	Top-1 accuracy
Aircraft	3334	3333	3333	100	Top-1 accuracy
Pets	3312	3669	368	37	Top-1 accuracy

DTD Cimpoi et al. (2014) and Aircraft Maji et al. (2013) have test set and validation set on Torchvision, so we directly use their test set and validation set. On Food101 Bossard et al. (2014), SVHN Netzer et al. (2011), CIFAR-10 Krizhevsky et al. (2009), CIFAR-100 Krizhevsky et al. (2009), Pets Parkhi et al. (2012) datasets, we divide the validation set from the original test set by stratified sampling with the proportion shown in the table.

B.2 PART I

B.2.1 RANDOM INITIALIZATION

Task	Backbone	LR	BS	Optimizer	Loss	Decoder	LR-decay	UBGEAN 3rd weight
Food101	resnet50	0.001	225	AdamW	CS	Linear	cosine	0.3
SVHN	resnet50	0.001	256	AdamW	CS	Linear	cosine	0.3
CIFAR-10	resnet50	0.003	200	AdamW	CS	Linear	cosine	0.3
CIFAR-100	resnet50	0.003	200	AdamW	CS	Linear	cosine	0.3

Table 5: Training Hyperparameters and Details of Random Initialization Considered in This Benchmark. CS represents CosineSimilarity loss and BS represents BatchSize. All the tasks use a cosine annealing with a linear warm-up learning rate scheduler Loshchilov & Hutter (2016); Goyal et al. (2017) and a UBGEAN weight of 0.3.

We choose a dataset with a relatively large amount of data for random initialization experiments. We use the AdamW optimizer, which weight decay is 0.0001. The decoder consists of only one linear layer, without connecting the dropout or BatchNorm layers. During the training process, we need to calculate the comparison loss between the original image data and the augmented image data, so we only resized the original image data to 224x224 and normalize the color channels (the average color and the standard deviation is computed on ImageNet). No other methods of data augmentation were used. It's the same to test set and val set. Finally, the test set accuracy of the model with the best performance on the validation set is selected as our final accuracy. The Settings used for both UBGEAN loss and experience loss are the same as those in Table 4.

B.2.2 FINE-TUNING

Task	Backbone	LR	BS	Optimizer	Loss	Decoder	LR-decay	UBGEAN 3rd weight
Food101	resnet50	0.00005	225	AdamW	CS	Linear	cosine	0.3
SVHN	resnet50	0.00005	256	AdamW	CS	Linear	cosine	0.3
CIFAR10	resnet50	0.00005	200	AdamW	CS	Linear	cosine	0.3
CIFAR100	resnet50	0.00005	200	AdamW	CS	Linear	cosine	0.3
DTD	resnet50	0.00005	188	AdamW	CS	Linear	cosine	0.3
Aircraft	resnet50	0.00005	200	AdamW	CS	Linear	cosine	0.3
Pets	resnet50	0.00005	196	AdamW	CS	Linear	cosine	0.3

Table 6: Training Hyperparameters and Details of Fine-tuning Considered in this Benchmark. CS represents CosineSimilarity loss and BS represents BatchSize. We use AdamW with a weight decay value of 1e-4, linear warm-up for the first 5 epochs and decay the learning rate with the cosine decay schedule Loshchilov & Hutter (2016); Goyal et al. (2017). The decoder consists of a simple linear layer and doesn't connect the Dropout and BatchNorm layers

We select all datasets for fine-tuning experiments. We use the AdamW optimizer, which weight decay is 0.0001. The decoder consists of only one linear layer, without connecting the dropout or BatchNorm layers. During the training process, as mentioned in the random initialization experiment, we only resized the original image data to 224x224 and normalize the color channels. It's the same to test set and val set.Finally, the test set accuracy of the model with the best performance on the validation set is selected as our final accuracy. The Settings used for both UBGEAN loss and experience loss are the same as those in Table 5.