Contrastive Learning with the Feature Reconstruction Amplifier

Wentao Cui[1], **Liang Bai**[1,2*]

[1] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi, China
[2] Institute of Intelligent Information Processing, Shanxi University, Taiyuan, 030006, Shanxi, China
cuiwentao.sxu@qq.com, bailiang@sxu.edu.cn

Abstract

Contrastive learning has emerged as one of the most promising self-supervised methods. It can efficiently learn the transferable representations of samples through the instance-level discrimination task. In general, the performance of the contrastive learning method can be further improved by projecting the transferable high-dimensional representations into the low-dimensional feature space. This is because the model can learn more abstract discriminative information. However, when low-dimensional features cannot provide sufficient discriminative information to the model (e.g., the samples are very similar to each other), the existing contrastive learning method will be limited to a great extent. Therefore, in this paper, we propose a general module called the Feature Reconstruction Amplifier (FRA) for adding additional high-dimensional feature information to the model. Specifically, FRA reconstructs the low-dimensional feature embeddings with Gaussian noise vectors and projects them to a high-dimensional reconstruction space. In this reconstruction space, we can add additional feature information through the designed loss. We have verified the effectiveness of the module itself through exhaustive ablation experiments. In addition, we perform linear evaluation and transfer learning on five common visual datasets, the experimental results demonstrate that our method is superior to recent advanced contrastive learning methods.

Introduction

Today, contrastive learning (CL) has achieved great success as a kind of self-supervised learning in the fields of computer vision (He et al. 2020; Chen et al. 2020a), natural language processing (Gao, Yao, and Chen 2021; Yan et al. 2021), graph neural network (You et al. 2020; Zhu et al. 2021), etc. The core idea of contrastive learning is pulling positive feature embeddings close to the anchor but pushing negative feature embeddings far away. Note that to make the model learn more abstract feature information, contrastive learning methods usually use low-dimensional feature embeddings rather than high-dimensional representations. That is, we tend to use a nonlinear multi-layer perceptron (i.e., a projection head) to project representations to a low-dimensional space. By discriminating these abstract feature embeddings, the performance of contrastive learning methods is greatly improved. However, are that representation information, i.e., more essential feature information, really useless?

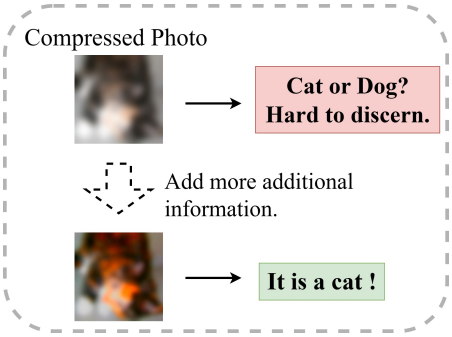


Figure 1: Contrastive learning models learn discriminative information of samples in a low-dimensional feature space. This process is similar to identifying the original content from a compressed image. When it is hard to discern, there is no doubt that adding additional information is a good method.

Here, we give an example as shown in Figure 1. Let's compress one photo and then identify its original content in the compressed photo. If we can discern the original content, the information in this compressed photo is sufficient. But what if the content in the compressed photo is simply illegible? We believe it is necessary to add some additional information at this point. Therefore, we design the Feature Reconstruction Amplifier (**FRA**) supplemented with additional information. Specifically, the FRA module reconstructs low-dimensional feature embeddings and projects them into a high-dimensional space, called the reconstruction space. In this reconstruction space, we train the entire framework by an additional loss, using (high-dimensional) reconstruction embeddings. In this way, the model can learn richer feature information that satisfies more conditions through FRA. In other words, we want these reconstruction embeddings to contain more correct discriminative information. Then, low-dimensional feature embeddings will be also more discriminative, because the reconstruction embeddings are generated through these feature embeddings. As in the previous example, we continuously adjust the compressed photo until it is

*Liang Bai is the corresponding author: bailiang@sxu.edu.cn

informative enough for us to identify its original content.

We call our overall framework a simple contrastive learning framework with the feature reconstruction amplifier (**SimFRA**). It should be emphasized that the Feature Reconstruction Amplifier is a general module. So, theoretically, the loss in the reconstruction space can be any loss function that obeys our assumptions. We experimentally verify the information gain brought by different losses based on FRA and the effectiveness of the SimFRA framework. In summary, our contributions are threefold:

- We propose a general module, i.e., FRA, to add extra feature information, forcing the original low-dimensional abstract features to be more discriminative.
- We verify the effectiveness of the module itself and the information gain brought by different losses through exhaustive ablation experiments.
- We verify the effectiveness of the proposed method by comparing it with state-of-the-art contrastive learning methods on several common vision datasets.

Related Work

In this section, we first review the development of contrastive learning. Then, we summarize some methods that involve adding additional information to enhance contrastive learning. Additional information mainly includes augmented image information, feature information, and text information.

Contrastive learning. Contrastive learning falls in the area of self-supervised learning (SSL). The key of a typical SSL method is to set pretext tasks, such as context prediction (Doersch, Gupta, and Efros 2015), colorization (Zhang, Isola, and Efros 2016), inpainting (Pathak et al. 2016), rotation (Komodakis and Gidaris 2018). Through these tasks, the model can obtain useful feature information from a large amount of unlabeled data. Among them, the pretext task of contrastive learning is an instance discrimination task. Specifically, InstDisc (Wu et al. 2018) treated each instance as a separate class, proposed the non-parametric classification problem at the instance level, and used the Noise Contrastive Estimation (**NCE**) loss (Gutmann and Hyvärinen 2010) to simplify the computation process and a memory bank to store a large number of instance-level class feature embeddings (Dosovitskiy et al. 2014). Due to the inconsistency of feature embeddings in the static memory bank, MoCo (He et al. 2020) set up a queue to dynamically update feature embeddings, and used the **InfoNCE** loss (Oord, Li, and Vinyals 2018) and a momentum update method to train the Siamese network. SimCLR (Chen et al. 2020a) built a simple weight-sharing Siamese network framework, which utilized sufficient data augmentations, large batch sizes, and a new projection space to greatly improve the CL model's performance.

In addition, there are some contrastive learning methods that only use positives. These methods can effectively learn visual representation information without the "collapse" problem. SwAV (Caron et al. 2020) obtained the cluster centers (i.e., prototypes) of instance feature embeddings in an online clustering manner and used the instance and its prototype for contrastive learning. BYOL (Grill et al. 2020) built an asymmetric Siamese network to predict the output of one view from another view with the mean square error loss (**MSE**), where one branch of the network is a momentum encoder. Based on BYOL, SimSiam (Chen and He 2021) removed the momentum encoder to further analyze the reason why the CL method using only positives is effective without the "collapse" problem. W-MSE (Ermolov et al. 2021) restricted feature embeddings to the spherical distribution by using a whitening transform and demonstrated that multiple positive pairs extracted from a single image can improve performance.

Methods to add additional information. Researchers have explored in several directions how to make feature embeddings more discriminative within the existing contrastive learning framework, i.e., how to add more correct and useful information. CLIP (Radford et al. 2021) used massive images and corresponding raw texts to construct (image, text) pairs for contrastive learning, and achieved results that surpassed supervised learning methods in the zero-shot transfer test way. InfoMin Aug (Poole et al. 2020) and ContrastiveCrop (Peng et al. 2022) respectively proposed new data augmentation methods. InfoMin Aug made the augmented view retain task-relevant information while minimizing irrelevant noise. ContrastiveCrop took semantic information into account when augmenting a sample. In the image-to-image translation task, Cut (Park et al. 2020) and NEGCUT (Wang et al. 2021) extracted feature embeddings in different layers of the encoder network to increase feature information of different levels, that is, multi-layer contrastive methods. As for negatives, MoCHi (Kalantidis et al. 2020) used two methods to generate new hard negatives at the feature embedding level. DCL (Chuang et al. 2020) proposed a biased contrastive loss to correct the sampling bias, hoping to reduce the impact of false negatives as much as possible. HCL (Robinson et al. 2021) proposed a new sampling method to obtain harder negatives and avoid false negatives through the defined "hardness".

Method

In this section, we first review the typical contrastive learning method. Then, we propose the Feature Reconstruction Amplifier (FRA) to add additional information to further force low-dimensional feature embeddings in the contrastive space to be more discriminative.

Preliminary

We take SimCLR (Chen et al. 2020a) as our baseline. The core idea of this method is to make the positive feature embedding close to the anchor but keep negative feature embeddings far away in the contrastive space. Suppose there are n instances in each mini-batch $X = \{x_1, x_2, ..., x_n\}$, where x_i means the i-th instance. We set up a family of m augmentation methods $A^v = \{a_1^v, a_2^v, ..., a_m^v\}$, where a_i^v denotes the i-th augmentation method used in the v-th view. Then, we generate two views X^1, X^2 of X by randomly different augmentation methods A^1 and A^2. By using an encoder network $f(\cdot)$, we can obtain representations

$H^v = f(X^v) = \{h_i^v\}_{i=1}^n$, where h_i^v denotes the i-th representation in the v-th view. With a nonlinear projection head $g_p(\cdot)$, H^v is projected into the contrastive space to get $Z^v = g_p(H^v) = \{z_i^v\}_{i=1}^n$. For convenience, we denote the positive pair by (z_i, z_j), where i is the index of z_i^1 in the mini-batch and j is the index of z_i^2. At last, the InfoNCE loss for the instance discrimination is defined as:

$$\mathcal{L}_{InfoNCE} = -\frac{1}{2n}\sum_{i=1}^{2n}\log\frac{\exp(z_i \cdot z_j/\tau)}{\sum_{i\neq k}\exp(z_i \cdot z_k/\tau)}, \quad (1)$$

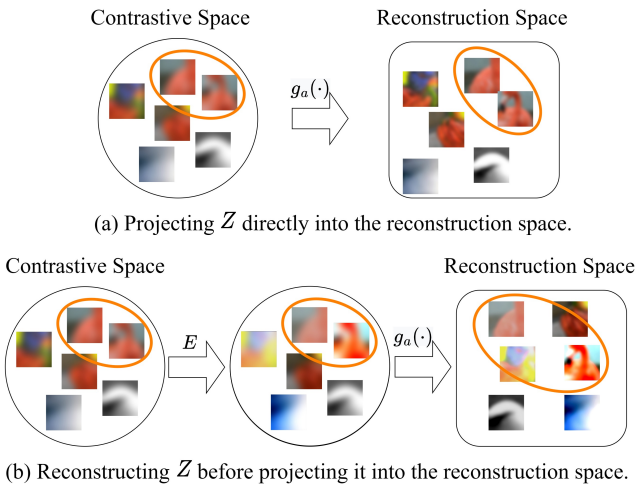
where τ is a temperature hyperparameter and z is distributed on the hypersphere through ℓ_2 normalization.

A Simple Contrastive Learning Framework with the Feature Reconstruction Amplifier

In contrastive learning, h_i is in fact a high-dimensional feature embedding, which contains more essential feature information of the instance. SimCLR empirically showed that projecting H into a low-dimensional space, which we call the contrastive space, can enable the model to obtain better representations. We think this is due to the fact that low-dimensional feature embeddings have higher-level abstract information, which greatly increases the difficulty of the instance discrimination task. However, we believe that removing instance-specific feature information completely will lead to a lack of discriminative information for the model.

Based on the above considerations, we propose a general module, called the Feature Reconstruction Amplifier (FRA). FRA reconstructs the embeddings Z^v to generate reconstruction feature embeddings R^v and uses R^v to add additional feature information. Specifically, we first recombine z_i^1, z_i^2 with a Gaussian noise vector e_i, respectively. In fact, we spliced the first half of z_i^1 with the second half of e_i to get the contrastive reconstruction feature $\widetilde{z}_i^1$. Similarly, we spliced the first half of e_i with the second half of z_i^2 to get $\widetilde{z}_i^2$. Then, we design a nonlinear MLP network as our amplifier, denoted as the amplifier head $g_a(\cdot)$. The linear layer dimension setting in $g_a(\cdot)$ is the exact opposite of $g_p(\cdot)$. Because the goal of $g_p(\cdot)$ is to obtain low-dimensional abstract feature embeddings, while the goal of $g_a(\cdot)$ is to obtain high-dimensional concrete feature embeddings. We feed $\widetilde{Z}^v$ into the amplifier head $g_a(\cdot)$ to get reconstruction feature embeddings $R^v = g_a(\widetilde{Z}^v) = \{r_i^v\}_{i=1}^n$, where the embedding dimension of R and H are the same.

The reason we didn't directly feed the amplifier head with Z^1, Z^2 is that Z^1 and Z^2 may be very similar. Despite using a nonlinear projection head, similar Z can easily lead to similar R as shown in Figure 2 (a). In the example, we still use images to vividly represent the feature vectors and the images in the red circle represent positives. In this case, it is difficult for the model to learn any additional discriminative feature information. We believe that complex tasks are more effective in improving the performance of the model, such as image augmentation before pre-training the model. Therefore, we set the above reconstruction method. In this way, FRA is trained to learn new information so that the positive samples in R can be close to each other, rather than being initially close like Figure 2 (b).



(a) Projecting Z directly into the reconstruction space.

(b) Reconstructing Z before projecting it into the reconstruction space.

Figure 2: Differences in the initial states of R under the direct projection method and the reconstruction projection method.

For the loss used in FRA, we think it can be any loss function that obeys our assumptions because the FRA module is a simple and general-purpose component. We set up three common losses to verify the effectiveness of FRA, i.e., the MSE loss ($\mathcal{L}_{MSE\_A}$), the InfoNCE loss ($\mathcal{L}_{InfoNCE\_A}$) and the Student-t loss ($\mathcal{L}_{ST\_A}$). Specifically, the MSE loss considers only the positive pairs in R. $\mathcal{L}_{MSE\_A}$ makes (r_i^1, r_i^2) consistent by simply reducing the distance between each positive pair. In contrast, the InfoNCE loss and the Student-t loss will take into account the positive and negative pairs in R. The goal of $\mathcal{L}_{InfoNCE\_A}$ and $\mathcal{L}_{ST\_A}$ is to make the reconstructed representations between positive pairs close but the reconstructed representations between negative pairs far away. The difference between $\mathcal{L}_{InfoNCE\_A}$ and $\mathcal{L}_{ST\_A}$ lies in the different ways of measuring similarity. The above losses are defined as:

$$\mathcal{L}_{MSE\_A} = \frac{1}{2n}\sum_{i=1}^{2n}(r_i - r_j)^2, \quad (2)$$

$$\mathcal{L}_{InfoNCE\_A} = -\frac{1}{2n}\sum_{i=1}^{2n}\log\frac{\exp(r_i \cdot r_j/\tau)}{\sum_{i\neq k}\exp(r_i \cdot r_k/\tau)}, \quad (3)$$

$$\mathcal{L}_{\mathrm{ST\_A}} = -\frac{1}{2n}\sum_{i=1}^{2n}\log\frac{(1+\|r_i - r_j\|^2)^{-1}}{\sum_{i\neq k}(1+\|r_i - r_k\|^2)^{-1}}, \quad (4)$$

where τ in Eq. (2) is a temperature hyperparameter and r in Eq. (4) does not perform ℓ_2 normalization. The overall objective function can be expressed as:

$$\mathcal{L}_{\mathrm{SimFRA}} = (1-w)\mathcal{L}_{InfoNCE} + w\mathcal{L}_A, \quad (5)$$

where w is a coefficient that increases linearly to $\frac{1}{2}$ in the first 100 training epochs. After 100 epochs, w becomes a

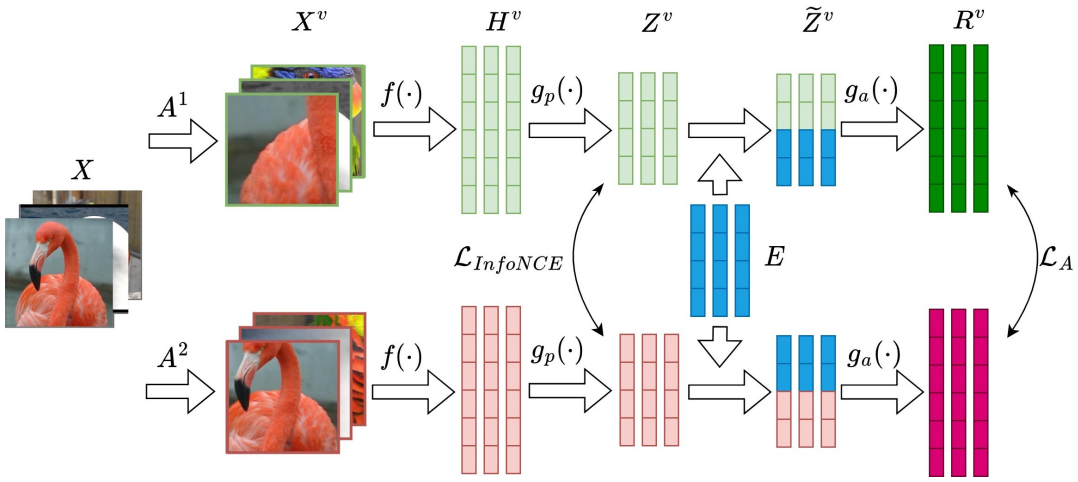


Figure 3: The SimFRA framework.

constant value (i.e., $\frac{1}{2}$). This is because, in the early stages of training, R contains task-unrelated or invalid information, we need to constrain R through $\mathcal{L}_{InfoNCE}$ so that they can gradually generate the right concrete feature information we need. $\mathcal{L}_A$ is one of $\mathcal{L}_{MSE\_A}$, $\mathcal{L}_{InfoNCE\_A}$ and $\mathcal{L}_{ST\_A}$. We study the information gain caused by the three losses.

Framework and Algorithm

As shown in Figure 3, the SimFRA framework is a symmetric Siamese network following SimCLR (Chen et al. 2020a). Assuming that the data in mini-batch is X, we augment X to two related view X^1 and X^2 through A^1 and A^2, and denoted as X^v. After obtaining representations H^v through the encoder network $f(\cdot)$, we use the projection head $g_p(\cdot)$ to project H^v into the contrastive space to get Z^v. Then we calculate the $\mathcal{L}_{InfoNCE}$ loss with Eq. (1). Next, we reconstruct Z^v with Gaussian noises E for the contrastive reconstruction feature $\widetilde{\mathrm{Z}}^v$. Then, R^v is obtained by the amplifier head $g_a(\cdot)$, and the $\mathcal{L}_A$ loss is calculated with Eq. (2) to (4). Finally, we calculate the overall SimFRA loss with Eq. (5). The overall algorithm flow is shown in Algorithm 1.

Algorithm 1: The SimFRA algorithm

Input: Instances X; augmentation methods A^v; the encoder network $f(\cdot)$; the projection head $g_p(\cdot)$; the amplifier head $g_a(\cdot)$
Parameter: Temperature hyperparameter τ; number of training epochs n
Output: The encoder network $f(\cdot)$

for $i = 1$ to n **do**
  $X^v = A^v(X)$
  $H^v = f(X^v)$
  $Z^v = g_p(H^v)$
  Generate random Gaussian noises E and get $\widetilde{\mathrm{Z}}^v$ by reconstructing Z^v with E
  $R^v = g_a(\widetilde{\mathrm{Z}}^v)$
  calculate the $\mathcal{L}_{\mathrm{InfoNCE}}$ loss by Eq. (1)
  calculate the $\mathcal{L}_{\mathrm{A}}$ loss by Eq. (2) to (4)
  optimize the SimFRA network by Eq. (5)
end for
return the encoder network $f(\cdot)$

Experiments

In this section, we first introduce implementation details in our experiments. Then we conduct detailed ablation experiments of the FRA module, including three different losses and the network structure. Lastly, we compare the SimFRA framework with several recent contrastive learning methods (our reproduced version), including the linear evaluation and transfer learning.

Implementation Details

We introduce the implementation details from four aspects: datasets, the experimental setup, augmentation methods, and the evaluation protocol. The specific content is as follows:

Datasets. We investigate contrastive learning using some common image datasets, such as CIFAR-10, CIFAR-100, STL-10, ImageNet-100, and Voc2007. Among them, **CIFAR-10** and **CIFAR-100** (Krizhevsky and Hinton 2009) each contains 50,000 training images and 10,000 test images. The size of each color image is 32×32. The difference is that CIFAR-10 contains ten classes while CIFAR-100 contains one hundred classes. Both **STL-10** (Coates, Ng, and Lee 2011) and **ImageNet-100**, i.e., IN-100, are derived from the ImageNet-1k dataset (Deng et al. 2009). STL-10 contains 10 classes, each with 500 training images and 800 test images. In addition, STL-10 has 100,000 unlabeled images for training. The image size in STL-10 is set to 96×96. IN-100 contains 100 random classes from IN-1k. Each class contains 1,300 training images and 50 test images. **Voc2007** is a standard small dataset with 9,963 images, specifically, 5,011 training images and 4,952 test images. It contains a total of 20 classes, and the number of images in each class is inconsistent. The size of each image is inconsistent, roughly 500×375 (the horizontal image) or 375×500 (the vertical

image).

Experimental setup. We reproduce several contrastive learning methods based on the code provided in previous work. All the data in the experiments are the test results of our reproduction methods. We generally set two **batch sizes**. On the IN-100 dataset, we set the batch size to 64. On other datasets, the batch size is 32. As for the **backbone network**, we mainly use the standard ResNet-18 and ResNet-50 (He et al. 2016). In ablation experiments, we use ResNet-18 as the backbone network. In comparison with other methods, we uniformly use the standard ResNet-50 as the backbone network, except for DCL and HCL. In DCL and HCL, the first convolutional layer in the ResNet-50 network is modified to be more suitable for images of small size according to the paper. Although this will cause some differences between the backbone network and augmentation methods, it is the only way to reproduce the results presented in the paper.

As for the **optimizer**, most methods use the Adam optimizer (Kingma and Ba 2014), but MoCo and MoCo v2 (Chen et al. 2020b) use the SGD optimizer. In MoCo and MoCo v2, the initial learning rate is set to 0.03, the SGD weight decay is 10^{-4} and the SGD momentum is 0.9. In DCL and HCL, the learning rate is 0.001 and the weight decay is 10^{-6}. In SimCLR and our SimFRA, the learning rate is 3×10^{-4}. In BYOL, the learning rate is 2×10^{-4}. As for the specific **hyperparameters** of each method, we set the temperature $\tau = 0.07$, the memory bank size $k = 65536$, and the momentum $m = 0.999$ in MoCo and MoCo v2. In SimCLR and our SimFRA, the temperature τ is set to 0.5. In DCL, the temperature $\tau = 0.5$, and the positive class prior $\tau^+ = 0.1$. In HCL, the temperature τ is set to 0.5. The positive class prior τ^+ and the concentration parameter β are set following in the paper. In BYOL, the exponential moving average parameter τ is set to 0.99. At last, we train models on the Voc2007 dataset for 500 **epochs**. On other datasets, we train the model for 400 epochs.

Augmentation methods. In most of the methods, we use the same augmentation methods. We first extract crops with a random size from 0.2 to 1.0 of the original image and then scale these crops to the size of 224×224. Next, we apply horizontal flip with probability 0.5, the color jittering with configuration (0.8, 0.8, 0.8, 0.2) with probability 0.8 and grayscaling with probability 0.2. When testing the model, we only resize the image to 224×224. The difference is that, in DCL and HCL, crops are scaled to the size of 32×32. DCL, HCL and MoCo v2 also add the GaussianBlur augmentation method.

Evaluation protocol. Following the widely adopted linear evaluation protocol, we use the well-trained frozen ResNet network to extract fixed representations. Note that we only use unlabeled data during training this ResNet network, strictly following the self-supervised setting. And throughout the testing process, the parameters of this ResNet network are fixed. For testing the representation quality, we train a supervised linear classifier for 500 epochs with these fixed feature embeddings. At last, we test the classification accuracy on the test set. For the optimizer used in the training of the classifier, most methods use the Adam optimizer. However, MoCo and MoCo v2 use the SGD optimizer following the paper setting.

Ablation Studies

In the ablation experiments, we only change the FRA module, i.e., the $\mathcal{L}_A$ loss and the amplifier head $g_a(\cdot)$. We set up the backbone network and the projection head with reference to SimCLR (Chen et al. 2020a). To demonstrate the effectiveness of the FRA module itself, we test three losses with Eq. (2) to (4) on the CIFAR-10 dataset. On the basis of FRA, they all effectively improve the quality of the final learned representations, and the results are shown in Table 1. When $\mathcal{L}_{ST\_A}$ is used as the FRA loss, the linear evaluation of the SimFRA model is the best. We then analyze the structure of $g_a(\cdot)$ and test the linear evaluation of SimFRA at different training epochs.

Different losses have different preferences for the structure of the amplifier head $g_a(\cdot)$. We tested the effect of different BN layers on FRA in our experiments. The results are shown in Figure 4. The amplifier head $g_a(\cdot)$ is similar to the projection head $g_p(\cdot)$, i.e., a nonlinear MLP. The differences are the dimension settings and the BN layer settings in the two headers. Specifically, the projection head $g_p(\cdot)$ projects representations to a low-dimensional space so that the model can learn abstract image information from the low-dimensional feature embeddings. In $g_p(\cdot)$, we set the dimension to $[2048, 2048, 128]$, add the BN layer after each linear layer, and add a nonlinear activation (ReLU) layer after the first BN layer. Instead, the amplifier head $g_a(\cdot)$ aims to provide additional information to the model from more specific feature embeddings in the higher dimension. We set the dimension in $g_a(\cdot)$ to $[128, 2048, 2048]$ and add a ReLU layer after the first linear layer. For the BN layer, differences in structural preferences between different losses are evident. Therefore, according to Figure 4, the FRA module with each loss takes the best performing amplifier structure.

Under the above settings, the linear evaluation of SimCLR and SimFRA with different training epochs are shown in Figure 5. It can be seen that by the time the SimFRA model is trained for 100 epochs, performances of SimFRA_MSE and SimFRA_ST are already better than or comparable to the performance of SimCLR. With the process of training, the weight of $\mathcal{L}_A$ in the FRA module increases and SimFRA performs better than SimCLR overall. This proves the effectiveness of our proposed method itself.

At the same time, we found a puzzling but interesting phenomenon during the experiment. When using MSE as the loss of FRA, we observe that the $\mathcal{L}_{MSE\_A}$ loss rapidly

$\mathcal{L}_{InfoNCE}$	$\mathcal{L}_A$			ACC
	$\mathcal{L}_{MSE}$	$\mathcal{L}_{InfoNCE}$	$\mathcal{L}_{ST}$	
✓				87.98
✓	✓			88.56
✓		✓		88.31
✓			✓	89.27

Table 1: Effectiveness of our framework based on the ResNet-18 network on CIFAR-10.

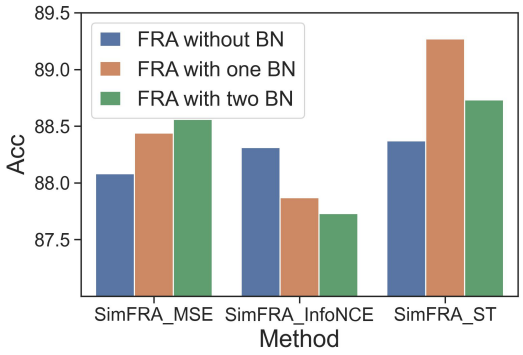


Figure 4: Linear evaluation of SimFRA with different amplifier heads $g_a(\cdot)$ on CIFAR-10.

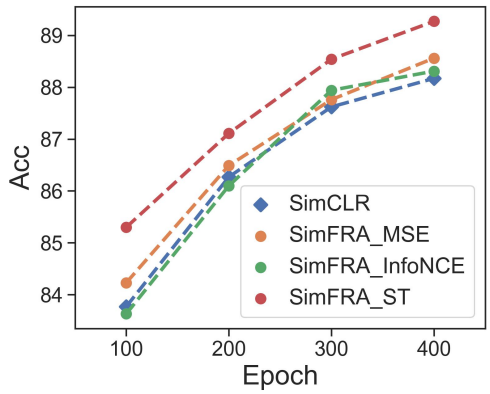


Figure 5: Comparison of linear evaluation between SimCLR and SimFRA with different training epochs on CIFAR-10.

converges to zero and the feature embeddings of the FRA module output also appear to the collapsed solution. This is caused by the MSE loss simply bringing the positives closer through the symmetric Siamese network. However, as shown in Figure 4, different structures of the FRA module with $\mathcal{L}_{MSE\_A}$ are indeed affecting the performance of the model. And except for $g_a(\cdot)$, the backbone network and the projection head in this experiment are all set up the same. This indicates that even though $\mathcal{L}_{MSE\_A}$ is a very small value, it is still acting on the model.

Comparison with State-of-the-Art

We compare SimFRA with advanced contrastive learning methods in linear evaluation and transfer learning. The experimental results show that SimFRA performs best in both assessment methods.

Linear evaluation. In these experiments, we use $\mathcal{L}_{ST\_A}$ as the loss of the FRA module, since this combination obtained the highest linear classification accuracy. Table 2 shows the results of experiments on the small and medium-sized datasets. SimFRA performs the best among all the four datasets, especially on the CIFAR-100 and STL-10 datasets. This is a good demonstration of the effectiveness of our

Method	CIFAR-10 Acc	CIFAR-100 Acc	STL-10 Acc	Voc2007 mAP
MoCo	77.02	52.01	80.97	-
MoCo v2	84.39	60.90	85.63	-
SimCLR	89.16	62.65	87.40	61.13
DCL	87.03	57.27	82.98	53.67
HCL	87.51	58.80	83.82	55.45
BYOL	88.70	64.23	87.36	56.89
SimFRA	90.72	66.72	91.07	62.61

Table 2: Classification accuracy (Acc) under linear evaluation on CIFAR-10, CIFAR-100 and STL-10 datasets. Mean average precision (mAP) on the Voc2007 dataset.

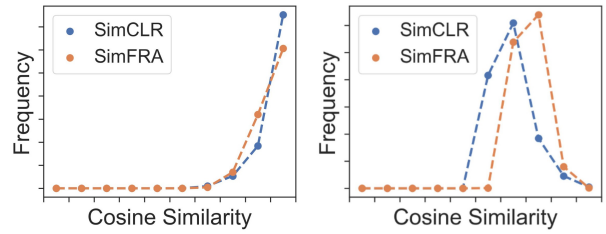
Method	Top-1	Top-5
MoCo (He et al. 2020)	55.02	80.96
SimCLR (Chen et al. 2020a)	69.03	90.21
SimFRA (ours)	69.84	90.60

Table 3: Classification accuracy on the ImageNet-100 dataset. Top-1 and top-5 correspond to the accuracy of a linear classifier.

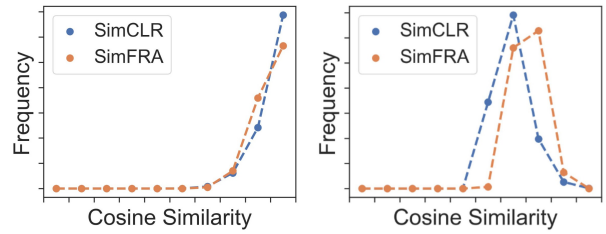
method. CIFAR-100 contains 100 classes, while each class has only 500 samples, and the 100,000 unlabeled samples in STL-10 contain a part of noisy samples (i.e., other types of animals and vehicles in addition to the ones in the labeled set). Due to the complex data, existing contrastive learning methods cannot learn the true distribution of the data well. However, SimFRA can effectively enhance the learning capability of the model by using additional feature information.

Then, we compare the top-1 and top-5 classification accuracy with MoCo and SimCLR on the IN-100 dataset, and the results are shown in Table 3. Compared to SimCLR, SimFRA provides 0.81% top-1 accuracy gains. In addition, we found that using the amplified head of a single BN layer leads to an unstable training process when training SimFRA on STL-10 and IN-100. It seems that the amplifier head is sensitive to the distribution of the inputs to each layer, especially on large datasets.

Furthermore, to verify the effect of additional feature information on the learned distribution, we measure and visualize the similarities of representations within each positive pair and negative pair. In this experiment, we respectively feed the images in CIFAR-10 and CIFAR-100 into the SimCLR pre-trained and SimFRA pre-trained ResNet-50. Then representations are extracted from each ResNet-50 network. After calculating the similarity of each positive pair and negative pair, we convert the value of cosine similarity to $[0, 1]$ by $(cos + 1)/2$. As we set the batch size to 64, there will be a total of 12,600,000 negative pairs ($50,000 \times 2{\times}126$). Considering that the similarity of (h_i^1, h_i^2) is the same as that of (h_i^2, h_i^1), we only count the similarity of (h_i^1, h_i^2), i.e., total of 50,000 positive pairs. Figure 6 shows each histogram of cosine similarities. The similarity of the positive pairs in both SimCLR and SimFRA is mainly concentrated in $[0.8, 1.0]$. However, the frequency of the similarity inter-



(a) Positive pairs on CIFAR-10. (b) Negative pairs on CIFAR-10.



(c) Positive pairs on CIFAR-100. (d) Negative pairs on CIFAR-100.

Figure 6: Cosine similarity of representations within each positive pair and negative pair on CIFAR-10 and CIFAR-100.

val $[0.8, 0.9]$ in SimFRA is significantly higher than that in SimCLR. For the negative pairs, the similarity distribution has a large difference. In SimFRA, the similarity scores between the anchor and its negatives are higher, concentrated in $[0.6, 0.9]$. This is due to the loss used in the FRA module. Theoretically, we can add any loss of reasonably assumed data distribution to make the model perform better.

Transfer learning. One of the main goals of self-supervised learning is to learn transferable features. To investigate the generalization ability of SimFRA on different datasets, we evaluate the transfer learning performance on CIFAR-10, CIFAR-100 and STL-10. Unlike linear evaluation, the data distribution used in training the pre-trained model is different from the data distribution used in the downstream classification task. Specifically, we use one dataset (the source dataset) to train the pre-trained ResNet-50, and use another dataset (the target dataset) to train the linear classifier during evaluation. Finally, the linear classifier is used to test the classification accuracy of the target dataset. The results are shown in Table 4. In all six sets of experimental results, SimFRA outperformed SimCLR in terms of transfer performance.

We can see that the improvement of SimFRA is most obvious when the target dataset is CIFAR-100. The source datasets are CIFAR-10 and STL-10, where the number of classes is much smaller than that in CIFAR-100. The pre-trained model trained with only low-dimensional features is insufficient to handle finer-grained classification tasks. SimFRA, which adds more feature information, effectively solves the problem of insufficient information. Besides, in experiments with CIFAR-100 as the source dataset, SimFRA has the smallest improvement. This shows that the more complex the data in the source dataset, the more discriminative information can be generated by the low-dimensional features. In this case, the information gain from FRA will be relatively less than in the first case.

Source	Target	Method	Acc
CIFAR-10	CIFAR-100	SimCLR	59.81
		SimFRA (ours)	63.86
	STL-10	SimCLR	72.59
		SimFRA (ours)	75.14
CIFAR-100	CIFAR-10	SimCLR	83.16
		SimFRA (ours)	84.99
	STL-10	SimCLR	70.21
		SimFRA (ours)	72.58
STL-10	CIFAR-10	SimCLR	84.15
		SimFRA (ours)	86.78
	CIFAR-100	SimCLR	57.24
		SimFRA (ours)	61.68

Table 4: Results of transfer learning across CIFAR-10, CIFAR-100 and STL-10 datasets with ResNet50. The source dataset is used to train the model. The target dataset is used to train the liner classifier and test the classification accuracy (Acc).

Conclusions

In this paper, we propose a general module called the Feature Reconstruction Amplifier (FRA) that applies to the contrastive learning method. When low-dimensional features cannot provide sufficient discriminative information to the model, FRA can effectively improve the performance of the model by supplementing with additional feature information. Using the SimCLR method as a baseline, we perform detailed ablation experiments on FRA and demonstrate the effectiveness of the FRA module itself in combination with different losses. In addition, we compare linear evaluation and transfer learning on common visual datasets with recent contrastive learning methods. The experimental results show that SimFRA achieves the best results.

We think there are two directions for future investigation. (a) We can design a loss function that is more suitable for the FRA module, although the existing loss function can already significantly improve the performance of the model. (b) In the transfer learning experiments, when the number of classes in the source dataset is more than that in the target dataset, the improvement to the model is relatively small. This means that FRA needs to add more discriminative feature information. In the future, we can further improve the structure and loss function of FRA.

Acknowledgements

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work is supported by National Key Research and Development Program of China (No. 2021ZD0113303), the National Natural Science Foundation of China (Nos. 62022052, 62276159).

References

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Neural Information Processing Systems*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 1597–1607. PMLR.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. arXiv:2003.04297.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750–15758.

Chuang, C.-Y.; Robinson, J.; Lin, Y.-C.; Torralba, A.; and Jegelka, S. 2020. Debiased Contrastive Learning. In *Neural Information Processing Systems*.

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 215–223. JMLR Workshop and Conference Proceedings.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. IEEE.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 1422–1430.

Dosovitskiy, A.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2014. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in Neural Information Processing Systems*, 27: 766–774.

Ermolov, A.; Siarohin, A.; Sangineto, E.; and Sebe, N. 2021. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, 3015–3024. PMLR.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Pires, B.; Guo, Z.; Azar, M.; et al. 2020. Bootstrap Your Own Latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 297–304. JMLR Workshop and Conference Proceedings.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Kalantidis, Y.; Sariyildiz, M. B.; Pion, N.; Weinzaepfel, P.; and Larlus, D. 2020. Hard Negative Mixing for Contrastive Learning. In *Neural Information Processing Systems*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.

Komodakis, N.; and Gidaris, S. 2018. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4).

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748.

Park, T.; Efros, A. A.; Zhang, R.; and Zhu, J.-Y. 2020. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 319–345. Springer.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2536–2544.

Peng, X.; Wang, K.; Zhu, Z.; Wang, M.; and You, Y. 2022. Crafting better contrastive views for siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16031–16040.

Poole, B.; Sun, C.; Schmid, C.; Krishnan, D.; Isola, P.; and Tian, Y. 2020. What makes for good views for contrastive representation learning? In *Neural Information Processing Systems*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763. PMLR.

Robinson, J. D.; Chuang, C.-Y.; Sra, S.; and Jegelka, S. 2021. Contrastive Learning with Hard Negative Samples. In *International Conference on Learning Representations*.

Wang, W.; Zhou, W.; Bao, J.; Chen, D.; and Li, H. 2021. Instance-wise hard negative example generation for contrastive learning in unpaired image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14020–14029.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3733–3742.

Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; and Xu, W. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In *Annual Meeting of the Association for Computational Linguistics*, 5065–5075.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *European Conference on Computer Vision*, 649–666. Springer.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, 2069–2080.