

# A Contrastive Learning Algorithm Based on Bayesian Random Semantic Feature Augmentation

Xiaofang Gao

gxfhttp@sxu.edu.cn

Shanxi University

Xianlei Meng

Shanxi University

---

## Research Article

**Keywords:** Contrastive learning, Bayesian inference, Feature enhancement, Frequency domain feature fusion

**Posted Date:** September 19th, 2025

**DOI:** <https://doi.org/10.21203/rs.3.rs-7441449/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

**Additional Declarations:** No competing interests reported.

---

# A Contrastive Learning Algorithm Based on Bayesian Random Semantic Feature Augmentation

Xiaofang Gao<sup>1\*</sup> and Xianlei Meng<sup>1†</sup>

<sup>1\*</sup>School of Computer and Information Technology, Shanxi University,  
Taiyuan, 030006, China.

\*Corresponding author(s). E-mail(s): [gxftp@sxu.edu.cn](mailto:gxftp@sxu.edu.cn);

Contributing authors: [mengxltt@qq.com](mailto:mengxltt@qq.com);

<sup>†</sup>These authors contributed equally to this work.

## Abstract

The Simple Contrastive Learning (SimCLR) algorithm is a self-supervised method that projects high-dimensional features into a low-dimensional space to enhance the similarity between positive samples and the difference between negative samples. However, this projection may cause the loss of feature information, significantly impacting the performance of downstream tasks. Therefore, we propose A Bayesian Random Semantic Data Augmentation Algorithm (SimKFA), an improved contrastive learning algorithm. A learnable linear transformation layer, KLP, replaces the multilayer perceptron (MLP) to preserve feature information. A feature enhancement module (BFAB), which combines Bayesian inference and frequency-domain feature analysis, is proposed. This module adaptively captures key information, maintains the continuity of the information flow, and reduces feature loss during deep learning. The experimental results show that the proposed algorithm prevents loss of feature information caused by the projection head and achieves excellent performance on multiple benchmark datasets.

**Keywords:** Contrastive learning, Bayesian inference, Feature enhancement, Frequency domain feature fusion

## 1 Introduction

Self-supervised learning has been widely applied in multiple fields, such as computer vision[1][2], natural language processing [3][4], and graph neural networks[5][6], and has achieved excellent results. Contrastive learning is a self-supervised method for

instance-level discrimination. It maximizes the similarity of positive sample pairs (two augmented versions of the same instance) and minimizes the similarity of negative sample pairs (other instances), enabling the model to learn features and perform well in downstream tasks. However, it is impossible to express complex feature relationships accurately when data samples are mapped from a high-dimensional to a low-dimensional feature space. Important information may be lost, reducing model performance in downstream tasks. Therefore, retaining key information in the low-dimensional space and performing feature-level data augmentation are critical challenges in contrastive learning.

Recent studies have shown that feature-level data augmentation can significantly improve the performance of networks[7][8][9][10] and promote contrastive learning in complex tasks. The deep feature space contains various semantic directions. New feature samples with the same class label but different semantic contents can be generated by representing features in these directions [11]. The implicit semantic data augmentation (ISDA)[12] algorithm generates augmented data in the feature space through random interference, interpolation, or extrapolation and minimizes the upper bound of the cross-entropy loss of the augmented dataset. Reference [13] proposed a Bayesian data augmentation method using the generalized Monte Carlo method and an adversarial network. It learns the feature distribution from the training set and performs data augmentation to increase the number of samples. InfoMin Aug[14] is an augmentation strategy to maximize the mutual information between samples and minimize information redundancy and inconsistency. ContrastiveCrop[15] is an augmentation strategy that utilizes local information cropping, providing good results for complex backgrounds. The SimFRA[16] is a feature reconstruction amplifier (FRA) that provides additional information and low-dimensional abstract features to increase the model's discriminative ability. DCL[17] proposed the debiased contrastive loss model to correct the sampling bias and minimize the impact of false negatives. HCL[18] increases the number of semantic features using hard negative samples. SimCLR-SSCL[19] synthesizes hard negative samples to boost model performance but suffers from false negatives. TNCC[20] framework mitigates this by imposing neighbor consistency constraints.

Inspired by the Kolmogorov-Arnold network (KAN)[21] and the Bayesian random semantic data augmentation (BSDA)[22], this paper proposes a new contrastive learning framework, SimKFA. It uses the semantic feature augmentation module BFAB, Bayesian inference, and the reconstruction of frequency-domain features to capture information adaptively from different feature dimensions. It maintains the continuity of information flow and reduces feature loss during deep learning. The advantage of the KAN network in the feature representation is the B-spline activation function in the multilayer perceptron (MLP), which significantly improves model performance, exceeding or matching that of KAN[21]. The SimKFA contrastive learning framework has a modified activation function. The linear weights are replaced with parameterized splines to improve model performance. The contributions of this paper are as follows:

- A semantic feature enhancement module (BFAB) is proposed. It combines Bayesian inference with feature reconstruction in the frequency domain to capture information from different feature dimensions adaptively in different semantic directions. This

strategy improves the network's learning and generalization ability, maintains the continuity of information flow, and reduces feature loss during deep learning.

- We propose a universal module (KLP) that uses a B-spline activation function in the MLP network, improving the representation of characteristics from the high-dimensional to low-dimensional space and optimizing the structure of the model.
- The contrastive learning loss function of SimCLR[1] algorithm is improved by adding a reconstruction loss and Kullback-Leibler (KL) divergence loss to the BFAB and an error loss function after enhancement.

## 2 Related Work

### 2.1 SimCLR

The SimCLR[1] method ensures that positive features are close to the anchor point and negative features are far from the anchor point in the contrastive space. Suppose that there are  $n$  instances  $X\{X_1, X_2, \dots, X_n\}$  in each mini-batch, where  $x_i$  represents the  $i$ -th instance. We establish a set  $A_v\{A_{v1}, A_{v2}, \dots, A_{vm}\}$  that contains  $m$  augmentation methods, where  $A_{vi}$  represents the  $i$  augmentation method in the  $v$ -th view. We generate two views ( $X_1$  and  $X_2$ ) of an instance by randomly selecting different augmentation methods  $A_1$  and  $A_2$ . We use the encoder network  $f(\cdot)$  to map views  $X_1$  and  $X_2$  to their representations  $H_1$  and  $H_2$  respectively:

$$H_v = f(X_v) = \{h_{vi}\}_{i=1}^n (v \in (1, 2)), \quad (1)$$

where  $h_{vi}$  represents the representation of the  $i$ -th sample in the  $v$ -th view. We project the high-dimensional representations  $H_v$  into the contrastive learning embedding space using the projection of the head  $g_p$  and obtain the embedded representations  $Z_1$  and  $Z_2$

$$Z_v = g_p(H_v) = \{z_{vi}\}_{i=1}^n, \quad (2)$$

where  $z_{vi}$  is the contrastive embedding of the  $i$ -th sample in the  $v$ -th view. The embeddings of pairs of samples  $z_{v1}$  and  $z_{v2}$  represent positive pairs, whereas the other samples in the same mini-batch are negative pairs.

The definition of the InfoNCE loss is as follows:

$$L_{infoNCE} = -\log \frac{\exp(\text{sim}(z_{i1} \cdot z_{i2}) / \tau)}{\sum_{j=1}^{2n} \exp(\text{sim}(z_{i1} \cdot z_j) / \tau)}, \quad (3)$$

where  $\tau$  is the temperature parameter, and  $\text{sim}(z_{i1} \cdot z_{i2})$  measures the similarity between the pairs of samples (usually the similarity of the cosine).

### 2.2 BSDA

The BSDA[22] uses the original features  $z$  and a semantic magnitude  $m$  in the randomly selected semantic directions  $d_\lambda$ . The augmented feature  $\tilde{z}$  is calculated as follows:

$$\tilde{z} = z + d_\lambda \cdot m, \quad (4)$$

where  $d \in \{0, 1\}^k$  is a binary vector that determines the semantic directions in which the augmentation is performed. It is initially set to 1 and then to 0 with a probability of  $\lambda$ .  $m \in R^k$  represents the magnitude of semantic augmentation. The feature remains unchanged when the augmentation magnitude is zero. The model changes the feature in specific directions when the augmentation magnitude is sampled from  $p(m|z)$ .

An indicator function is used to ensure that the semantic augmentation is applied in an appropriate direction:

$$\tilde{z} = z + \prod_{z \neq 0} d_\lambda \cdot m, \quad (5)$$

where  $\prod$  is the indicator function, its value is 1 when  $z \neq 0$ . We need to estimate  $p(m|z)$  to ensure that the augmented features have the correct distribution. We utilize an approximate distribution  $q_{\Phi_m}(m|z)$  and optimize it by minimizing the KL divergence to ensure the distribution matches the real one  $p(m|z)$ . The optimization objective function is:

$$\Phi_m = \operatorname{argmax} D_{KL}(q_{\Phi_m}(m|z) || p(m|z)), \quad (6)$$

We remove entries unrelated to parameter  $\Phi_m$  in  $D_{KL}$ :

$$D_{KL}(q_{\Phi_m}(m|z) || p(m|z)) - E_{m \sim q_{\Phi_m}(m|z)}(\log p(z|m)), \quad (7)$$

where the first term indicates how well the distribution of the enhanced features matches the true distribution, and the second term, given the features of , measures the enhanced features. We utilize a variational autoencoder (VAE) [23] and rewrite the second term as  $p_{\Phi_z}(z|m)$  to differentiate between the generator and the decoder. The loss function of the BSDA[22] is obtained as follows:

$$L_B(\Phi_m, \Phi_z; z) = -KL(q_{\Phi_m}(m|z) || p(m)) + E_{m \sim q_{\Phi_m}(m|z)}(\log p_{\Phi_z}(z|m)), \quad (8)$$

We assume that the marginal distribution  $p(m)$  follows the normal distribution  $N(0, 1)$  and  $q_{\Phi_m}(m|z)$  follows the normal distribution. We set the mean to 0. Learning the offset with respect to the original features yields the following loss function for BSDA:

$$L_B(\Phi_m, \Phi_z; z) = \frac{1}{2} \sum_{i=0}^N (1 + \log(\sigma^2) - \sigma^2) + \frac{1}{2N} \sum_{i=1}^N (\hat{z} - z)^2, \quad (9)$$

where  $\sigma^2$  is the estimated variance of BSDA, and  $\hat{z}$  is the reconstructed feature using  $m$ .

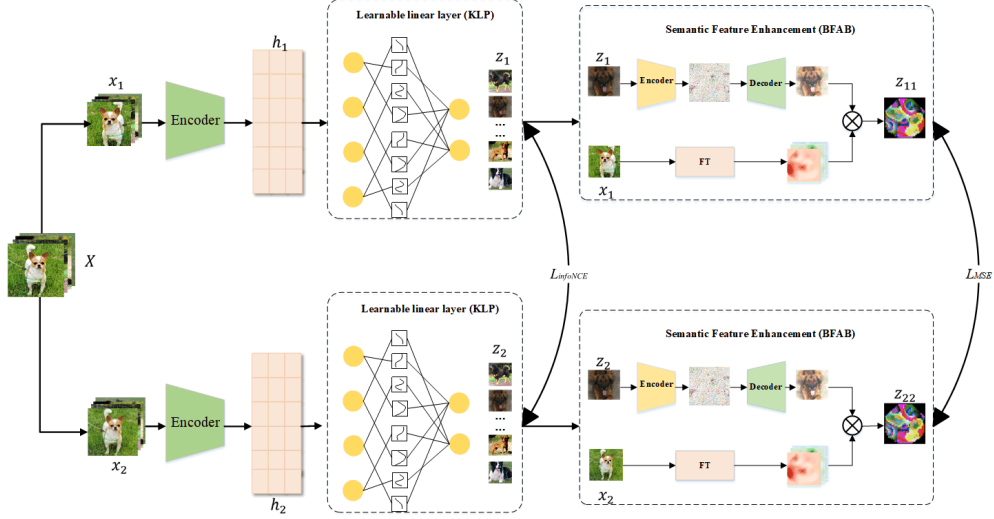


Fig. 1: Framework of SimKFA

### 3 SimKFA algorithm based on Bayesian random semantic feature augmentation

#### 3.1 SimKFA

The proposed SimKFA algorithm is based on the classical SimCLR framework. It uses a learnable linear transformation layer (KLP) instead of a multilayer perceptron MLP and a semantic feature enhancement module (BFAB) that combines Bayesian inference and frequency-domain feature fusion to improve feature representation. An improved contrastive loss function is incorporated to improve the model's contrastive learning capacity. The algorithm framework is shown in fig. 1.

Data augmentation is performed on the original image  $X$  to generate different views ( $x_1$  and  $x_2$ ), and features  $h_1$  and  $h_2$  are extracted by an encoder with shared weights. These features are transformed in the KLP layer using a modified projection header to provide more flexible feature mapping capabilities and generate feature representations in the contrastive learning space  $z_1$ ,  $z_2$ . The BFAB feature fusion enhancement block augments feature representation  $z_1$  and  $z_2$  the data enhancement view  $x_1$  and  $x_2$ . The feature representation is augmented by the BSDA[22], and the enhanced view is augmented using a Fourier transform (FT) in the frequency domain. Feature fusion is performed by utilizing frequency-domain information to optimize feature representation, and augmented features are used to generate  $z_{11}$  and  $z_{22}$  to improve feature robustness.

#### 3.2 Linear Transformation Layer KLP

The MLP has excellent feature extraction and classification performance[24][25]. It has been widely used for image classification and is a crucial component of the SimCLR

algorithm. As shown in the fig. 2a The output of the MLP's linear layer is a weighted sum :

$$y = W \cdot x + b, \quad (10)$$

where  $W$  is the weight matrix,  $x$  is the input, and  $b$  is the bias vector. The output of the linear layer is generally passed to a nonlinear activation function (such as ReLU or Sigmoid) for processing.

However, the MLP is limited to linear weight mapping and has shortcomings in nonlinear feature modeling and interpretability. The KAN[21] network is an alternative to the MLP and has demonstrated excellent feature transformation in multiple fields. It is suitable for image classification and contrastive learning. It uses a learnable activation function and parameterized splines instead of linear weights[21], significantly improving the model's accuracy and interpretability in image classification. The learnable linear transformation layer (KLP layer) and the B-splines [26][27] capture nonlinear relationships. This strategy characterizes local changes in input features using a polynomial approach, enabling independent scaling, capturing feature relationships, and retaining key information . As shown in the fig. 2b. It maps the input data using piecewise polynomials and enhances the feature expression capacity by linearly combining the outputs of the basis functions:

$$y_{klp} = W_{base} \cdot \sigma(x) + W_{spline} \cdot B-splines(x), \quad (11)$$

where  $W_{base}$  is the basic linear weight,  $W_{spline}$  is the weight of the B-spline,  $\sigma$  is the nonlinear activation function (such as SiLU), and the B-splines are piecewise polynomials.

### 3.3 Feature Augmentation Layer BFAB

The enhancement block BFAB consists of the BSDA[22] and the feature frequency-domain fusion. BSDA utilizes a variational Bayesian approach to estimate the magnitude of data augmentation. It extracts samples from the data and adds them to the

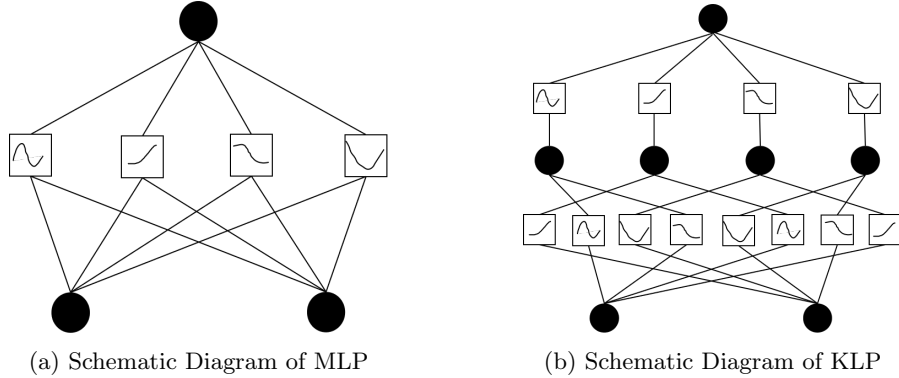


**Fig. 2:** Schematic Diagram of MLP and KLP

original features. BSDA performs feature enhancement on the data features  $z_1$  and  $z_2$ . The KLP layer significantly influences the results.

Frequency-domain feature fusion aims to capture different types of information by transforming the image from the null domain to the frequency domain, improving the performance of the image classification model. The randomly enhanced feature  $x_1$  and  $x_2$  are Fourier transformed to obtain the frequency-domain features of  $x_1$  and  $x_2$ . The BSDA is used to fuse the frequency-domain features with the reconstructed  $\hat{z}$  to generate the new  $\bar{z}$ , which maintains the feature information of the data. The loss function of the augmented data layer is:

$$L_B(\Phi_m, \Phi_z; z) = \frac{1}{2} \sum_{i=0}^N (1 + \log(\sigma^2) - \sigma^2) + \frac{1}{2N} \sum_{i=1}^N (\bar{z} - z)^2, \quad (12)$$

Re-parameterization is used to simplify the computation of the loss function while ensuring backpropagation of the gradient flow. The random variable  $m$  is a deterministic variable  $m = g_\phi(\lambda, \alpha)$ , where  $\lambda \sim N(0, 1)$  is an auxiliary variable with independent marginal distribution  $p(\varepsilon)$ , and is a vector-valued function parameterized by  $\Phi_m$ , which is obtained by BFAB augmentation:

$$\tilde{Z} = b_p(Z_v), \quad (13)$$

The two sample features  $\tilde{Z}_1$  and  $\tilde{Z}_2$  obtained after BFAB enhancement are highly similar to the original features because they originate from the same sample. A constrained loss function is obtained:

$$L_{MSE} = \frac{1}{2N} \sum_{i=1}^{2N} (\tilde{Z}_i - \tilde{Z}_j)^2, \quad (14)$$

### 3.4 Constructing the objective loss function

The loss function of the SimKFA algorithm is:

$$L = L_{infoNCE} + L_B + L_{MSE}, \quad (15)$$

where  $L_{infoNCE}$  is the instance-level contrast loss [Equation 3](#).  $L_B$  is the loss function of the data-enhanced BFAB [Equation 12](#).  $L_{MSE}$  is the contrast loss of the sample features after data enhancement [Equation 14](#).

### 3.5 Description of the SimKFA algorithm

The SimKFA algorithm introduces diverse semantic perturbations through a "sample-feature" dual-level semantic augmentation module. It combines Fourier frequency-domain information fusion and a multi-loss joint optimization strategy to guide the backbone neural network in learning feature representations with higher discriminability and generalization ability. Its overall framework and execution process are shown in [table 1](#).

**Table 1:** Description of the SimKFA Algorithm Framework

---

|                                                                                                      |
|------------------------------------------------------------------------------------------------------|
| <b>Algorithm:</b> Contrastive learning algorithm based on Bayesian random semantic data augmentation |
|------------------------------------------------------------------------------------------------------|

---

|                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Inputs:</b><br>Batch samples $X$ , Augmentation method $A_v$ , Backbone neural network $f(\cdot)$ , Feature projection head $g_p(\cdot)$ , Fourier transform $Ft(\cdot)$ , Feature augmentation module $b_p(\cdot)$ . |
| <b>Output:</b> Trained backbone neural network                                                                                                                                                                           |
| <b>Parameters:</b><br>- Temperature hyperparameters $\tau$<br>- Number of training rounds $n$                                                                                                                            |

---

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Procedure:</b><br><b>for</b> $i = 1$ <b>to</b> $n$ :<br>$X_v = A_v(X)$<br>$H_v = f(X_v)$<br>$Z_v = g_p(H_v)$<br>Calculate contrastive loss using Equation 3<br>$\tilde{Z}_v = b_p(Z_v) + Ft(X_v)$<br>Calculate the reconstruction loss and the KL scatter loss using Equation 12<br>Calculate the error loss using Equation 14<br>Calculate the overall loss using Equation 15 and optimize the network<br><b>end for</b><br><b>Return:</b> Backbone network |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## 4 Experiments

We compared the SimKFA contrastive learning framework and multiple contrastive learning methods on several standard datasets. The performance of the SimKFA contrastive learning framework was evaluated using comparison of classification performance, transfer experiment, ablation studies, adjusting the temperature coefficient and visualization experiment.

### 4.1 Dataset

The dataset adopts a common dataset. The various parameters are shown in table 2 as follows:

**Table 2:** Statistics of datasets used in experiments.

| Dataset      | Samples | Pixel Size | Classes |
|--------------|---------|------------|---------|
| CIFAR-10     | 60,000  | 32×32      | 10      |
| STL-10       | 13,000  | 96×96      | 10      |
| CIFAR-100    | 60,000  | 32×32      | 100     |
| ImageNet-100 | 135,000 | 64×64      | 100     |
| TinyImageNet | 120,000 | 64×64      | 200     |

## 4.2 Comparison of Classification Performance

We compared SimKFA with several comparative learning algorithms. The results of 500 epochs are listed in table 3. SimKFA achieves the optimal performance on all datasets. It performs especially well on datasets with many categories. It exhibits improvements in the Top-1 accuracy of 2.26 % and 4.66% compared to CIFAR-100[28] and TinyImageNet[29], respectively. The proposed SimKFA algorithm has better performance than the SimCLR [1] algorithm due to the KLP linear layer, which is superior to MLP linear mapping, enabling the model to learn complex feature distributions. Unlike MOCO[2] and BYOL[30], SimKFA does not require momentum updates or negative samples but uses FT transform to improve the global feature expression capability, resulting in more stable model training. DCL[17] is prone to feature collapse due to the lack of a feature enhancement mechanism. HCL[18] has a low generalization ability for small sample sizes due to the high computational complexity of calculating the hierarchical contrast loss. SimKFA does not exhibit feature space collapse, performs stronger on datasets with more classes, and has a higher generalization ability.

SimKFA exhibits a 1.53% improvement over ImageNet-100[29] due to its effectiveness on large datasets (table 4). The results indicate that SimKFA outperforms classical methods on a wide range of datasets.

**Table 3:** Comparative Experiments between SimKFA and Various Contrastive Learning Algorithms

| Method               | CIFAR-10     | STL-10       | CIFAR-100    | Tiny<br>Imagenet |
|----------------------|--------------|--------------|--------------|------------------|
| SimCLR               | 89.94        | 90.90        | 65.40        | 47.06            |
| MOCO                 | 77.02        | 80.97        | 52.01        | -                |
| MOCO v2              | 84.39        | 85.63        | 60.90        | 22.57            |
| DCL                  | 87.03        | 82.98        | 57.27        | 25.56            |
| HCL                  | 85.91        | 83.82        | 58.71        | 37.15            |
| BYOL                 | 86.15        | 87.36        | 55.36        | 13.34            |
| SimCLR-SSCL          | 86.92        | 82.08        | 61.18        | 37.31            |
| TNCC                 | 90.64        | 90.11        | 64.37        | 48.23            |
| SimFRA               | 90.72        | 91.07        | 66.72        | 49.47            |
| <b>SimKFA (ours)</b> | <b>90.99</b> | <b>91.16</b> | <b>67.66</b> | <b>51.72</b>     |

**Table 4:** Top-1 and Top-5 accuracy (%) comparison of different methods.

| Method               | Top-1        | Top-5        |
|----------------------|--------------|--------------|
| MoCo                 | 55.02        | 80.96        |
| SimCLR               | 69.81        | 91.12        |
| <b>SimKFA (ours)</b> | <b>71.34</b> | <b>91.81</b> |

### 4.3 Transfer learning

One goal of self-supervised learning is to learn transferable features. We evaluated SimKFA’s generalization and transfer learning ability on different datasets. We used a source dataset to train the pre-trained ResNet-50[31] and another target dataset to train a linear classifier to assess the model’s classification accuracy.

The results in table 5 indicate that SimKFA outperforms SimCLR[1] in transfer learning on all 6 datasets, demonstrating high feature generalization capability. Notably, when the source dataset is CIFAR-10 or STL-10—both with fewer categories compared to CIFAR-100—the performance of traditional pre-trained models often deteriorates on fine-grained classification tasks due to limited feature expressiveness. In contrast, SimKFA, equipped with the KLP and BFAB modules, effectively alleviates the issue of insufficient information and enhances the discriminative power and robustness of the learned representations.

Moreover, when CIFAR-100 is used as the source dataset, SimKFA exhibits even larger performance gains. This suggests that the complexity of the source domain does not hinder the transferability of SimKFA. On the contrary, the model demonstrates stronger discriminative ability and generalization, indicating its effectiveness in adapting to diverse feature distributions across domains.

**Table 5:** Accuracy comparison between SimCLR and SimKFA on different dataset transfer tasks.

| Source    | Target    | Method               | Acc          |
|-----------|-----------|----------------------|--------------|
| CIFAR-10  | CIFAR-100 | SimCLR               | 61.91        |
|           |           | <b>SimKFA (ours)</b> | <b>63.17</b> |
|           | STL-10    | SimCLR               | 75.73        |
|           |           | <b>SimKFA (ours)</b> | <b>76.34</b> |
| CIFAR-100 | CIFAR-10  | SimCLR               | 84.39        |
|           |           | <b>SimKFA (ours)</b> | <b>86.18</b> |
|           | STL-10    | SimCLR               | 72.96        |
|           |           | <b>SimKFA (ours)</b> | <b>73.69</b> |
| STL-10    | CIFAR-10  | SimCLR               | 86.10        |
|           |           | <b>SimKFA (ours)</b> | <b>86.83</b> |
|           | CIFAR-100 | SimCLR               | 61.35        |
|           |           | <b>SimKFA (ours)</b> | <b>62.23</b> |

**Table 6:** Results of the ablation study for the SimKFA model.

| Method |     |      |      | Dataset      |              |              |              |
|--------|-----|------|------|--------------|--------------|--------------|--------------|
| SimCLR | KLP | BSDA | BFAB | CIFAR-10     | STL-10       | CIFAR-100    | TinyImagenet |
| ✓      |     |      |      | 84.60        | 84.48        | 57.72        | 43.28        |
| ✓      | ✓   |      |      | 85.59        | 86.04        | 59.02        | 45.16        |
| ✓      |     | ✓    |      | 85.93        | 84.09        | 61.62        | 45.45        |
| ✓      |     |      | ✓    | 86.44        | 84.41        | 61.51        | 45.71        |
| ✓      | ✓   |      | ✓    | <b>87.19</b> | <b>86.66</b> | <b>62.92</b> | <b>47.55</b> |

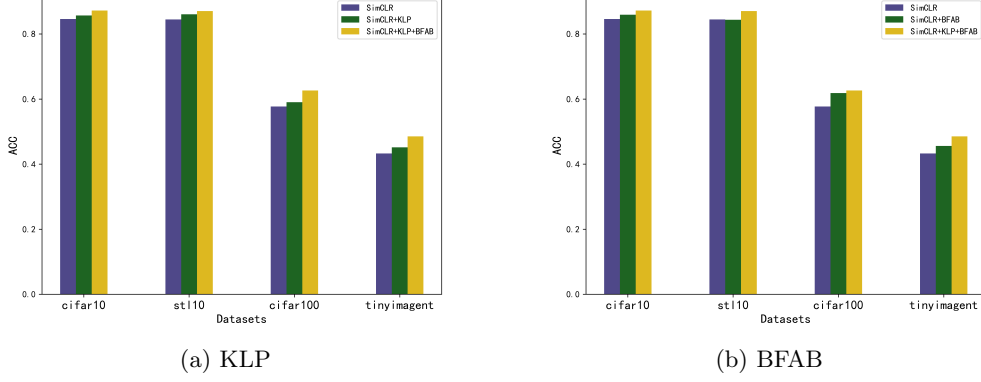


**Fig. 3:** Top-1 of each dataset on KLP and BFAB

#### 4.4 Ablation Studies

An ablation study were conducted by running 100 epoch each on the CIFAR-10[28], STL-10[32], CIFAR-100[28], and TinyImageNet[29] datasets to verify the influences of the KLP and BFAB layers. The KLP layer fig. 3a improves the quality of the learned representations for all datasets, especially for the STL-10[32] dataset, although the sample size of the training set is small. The BFAB data enhancement module fig. 3b significantly improves model performance on the CIFAR-100[28] dataset, but the performance improvement is lower on the STL-10[32].

The results of the ablation study show that the KLP module performs better for a small training sample size, whereas the BFAB module performs better for a larger sample size. However, good experimental results are obtained when the two modules are combined, regardless of the sample size table 6.

#### 4.5 Adjusting the Temperature Coefficient

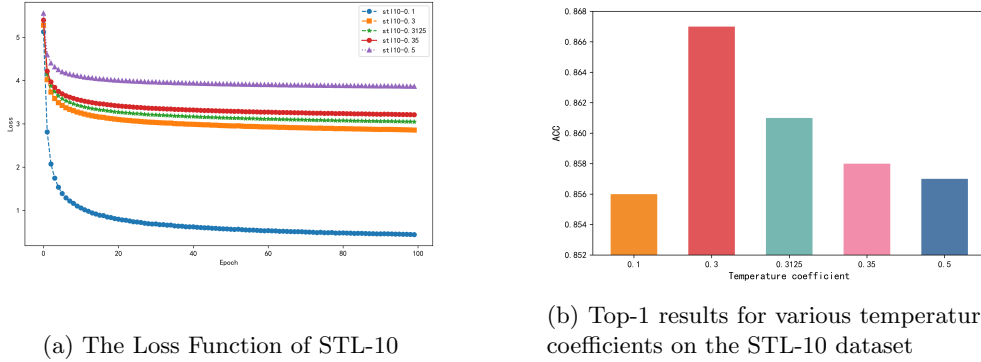
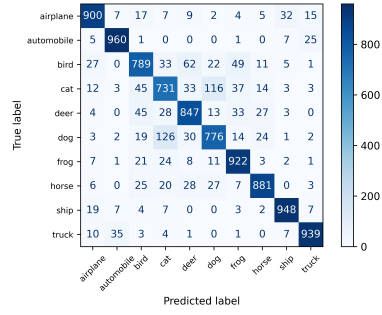


**Fig. 4:** The Influence of the Temperature Coefficient on the SimKFA

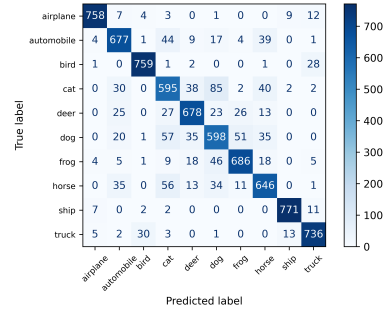
The temperature coefficient can be used to adjust the sensitivity of the contrastive loss. An appropriate temperature coefficient is required to distinguish the samples. The results of adjusting the temperature coefficients are shown in fig. 4a. The model converges too quickly when the temperature coefficient is 0.1, and the parameter space is insufficient, reducing model performance on the test set. Although the convergence speed is nearly the same at 0.3, the loss is relatively large, which may cause insufficient learning capacity. The convergence speed and the model’s generalization ability are appropriate at a temperature coefficient of 0.3. Therefore, we evaluated different temperature coefficients on the validation set. The model achieves the highest accuracy at a temperature coefficient of 0.3. as shown in the fig. 4b

## 4.6 Visualization Experiment

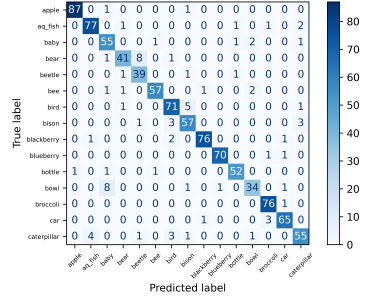
The confusion matrices of SimKFA’s classification results on the CIFAR-10[28], STL-10[32], CIFAR-100[28], and ImageNet-100[29] datasets are shown in fig. 5. The results indicate excellent classification performance. The SimKFA model can accurately classify image data with different semantics.



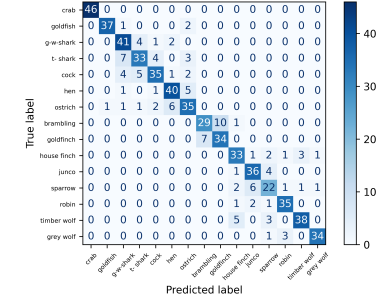
(a) CIFAR-10



(b) STL-10



(c) CIFAR100



(d) ImageNet-100

**Fig. 5:** Confusion matrices for different datasets

## 5 Conclusion

We proposed the contrastive learning algorithm SimKFA . It uses the KLP layer to improve model’s feature representation ability using polynomials and B-splines to retain important features during downscaling from high-dimensional to low-dimensional data. The feature enhancement block utilizes Bayesian inference and frequency-domain feature reconstruction to preserve the features during the fusion of high-dimensional and low-dimensional data. The experimental results verify the high performance of SimKFA.

However, the SimKFA algorithm performs better on large datasets with more categories and slightly worse on small datasets with fewer categories. The reason is that the BFAB layer is not suitable for small sample sizes. We plan to improve the SimKFA to increase its performance.

**Acknowledgements.** This research is funded by Project of Basic Research Program of Shanxi Province(202203021221001).

## References

- [1] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020). PmLR
- [2] He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
- [3] Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821 (2021)
- [4] Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., Xu, W.: Consert: A contrastive framework for self-supervised sentence representation transfer. arXiv preprint arXiv:2105.11741 (2021)
- [5] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. *Advances in neural information processing systems* **33**, 5812–5823 (2020)
- [6] Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021, pp. 2069–2080 (2021)
- [7] Ahn, E., Kumar, A., Fulham, M., Feng, D., Kim, J.: Unsupervised domain adaptation to classify medical images using zero-bias convolutional auto-encoders and context-based feature augmentation. *IEEE transactions on medical imaging* **39**(7), 2385–2394 (2020)

- [8] Kang, Y., Zhao, X., Zhang, Y., Li, H., Wang, G., Cui, L., Xing, Y., Feng, J., Yang, L.: Improving domain generalization performance for medical image segmentation via random feature augmentation. *Methods* **218**, 149–157 (2023)
- [9] Li, P., Li, D., Li, W., Gong, S., Fu, Y., Hospedales, T.M.: A simple feature augmentation for domain generalization. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8886–8895 (2021)
- [10] Wang, M., Yuan, J., Qian, Q., Wang, Z., Li, H.: Semantic data augmentation based distance metric learning for domain generalization. In: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3214–3223 (2022)
- [11] Wang, Y., Huang, G., Song, S., Pan, X., Xia, Y., Wu, C.: Regularizing deep networks with semantic data augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(7), 3733–3748 (2021)
- [12] DeVries, T., Taylor, G.W.: Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538* (2017)
- [13] Tran, T., Pham, T., Carneiro, G., Palmer, L., Reid, I.: A bayesian data augmentation approach for learning deep models. *Advances in neural information processing systems* **30** (2017)
- [14] Poole, B., Sun, C., Schmid, C., Krishnan, D., Isola, P., Tian, Y.: What makes for good views for contrastive representation learning? *Proc. NeurIPS*, 13 (2020)
- [15] Peng, X., Wang, K., Zhu, Z., Wang, M., You, Y.: Crafting better contrastive views for siamese representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16031–16040 (2022)
- [16] Cui, W., Bai, L.: Contrastive learning with the feature reconstruction amplifier. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 7279–7287 (2023)
- [17] Zhou, K., Zhang, B., Zhao, W.X., Wen, J.-R.: Debaised contrastive learning of unsupervised sentence representations. *arXiv preprint arXiv:2205.00656* (2022)
- [18] Robinson, J., Chuang, C.-Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020)
- [19] Dong, H., Long, X., Li, Y.: Synthetic hard negative samples for contrastive learning. *Neural Processing Letters* **56**(1), 33 (2024)
- [20] Cui, W., Bai, L., Yang, X., Liang, J.: A new contrastive learning framework for reducing the effect of hard negatives. *Knowledge-Based Systems* **260**, 110121 (2023)
- [21] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M.,

- Hou, T.Y., Tegmark, M.: Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2024)
- [22] Zhu, Y., Cai, X., Wang, X., Yao, Y.: Bayesian random semantic data augmentation for medical image classification. arXiv e-prints, 2403 (2024)
- [23] Kingma, D.P., Welling, M., et al.: Auto-encoding variational bayes. Banff, Canada (2013)
- [24] Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* **2**(4), 303–314 (1989)
- [25] Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural networks* **2**(5), 359–366 (1989)
- [26] Cox, M.G.: The numerical evaluation of b-splines. *IMA Journal of Applied mathematics* **10**(2), 134–149 (1972)
- [27] De Boor, C.: On calculating with b-splines. *Journal of Approximation theory* **6**(1), 50–62 (1972)
- [28] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
- [29] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). Ieee
- [30] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., *et al.*: Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems* **33**, 21271–21284 (2020)
- [31] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
- [32] Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011). JMLR Workshop and Conference Proceedings