A. Details of Dataset Setup

In our work, we consider both natural and synthetic distribution shifts in empirical evaluation. The details of the dataset settings are shown in Table 9.

B. Baseline Methods

Prediction Score (Pred) [28] It is defined as the maximum softmax output of the model. If the prediction score of a sample is greater than a given threshold τ, it is regarded correct.

Entropy Score (Entropy) [36] It is defined as the normalized entropy of softmax outputs (normalized by $\log K$, where K is the number of classes. If the entropy score of a sample is less than a given threshold τ, it is regarded correct.

Proxy Risk [9] This work propose a set of domain-invariant predictors as a proxy for the unknown, true target labels. They train the check model and fine-tune it to maximize the disagreement using a separate target training dataset sampled from the distribution of the test dataset. If the check model gives a different prediction, it is regarded as an error prediction of the given model.

Ensemble Average Confidence (Ens. AC) [38] This method involves training multiple neural networks with different initializations and architectures, and then combining their predictions to obtain a probabilistic estimate of the target variable. The model's accuracy is estimated by the average confidence calculated from the model ensemble.

Ensemble Method $\mathcal{T}_{RI}$ (Ens. RI) [4] This method uses the same training algorithm as for the given model to train a model ensemble from different random initialization (RI). If the model gives a different prediction, it is regarded as an error prediction of the given model.

Ensemble Method $\mathcal{T}_{RM}$ (Ens. RM) [4] This method is also based on model ensemble similar to the Ens RI mentioned above, but designed with the representation matching (RM) technique for domain adaptation, which can potentially improve the accuracy of the ensemble on some test inputs related to the training data.

Frechet [14] This method first synthesizes many test sets. And then, it computes the frechet Distance (FD) between the training set and each of the test set. Using the (FD, acc) value pairs, it can build a regression model to estimate the model's accuracy on an unlabeled test set.

Frechet + μ + σ [14] Similar to frechet mentioned above, but adds the mean and variance values to the Frechet Distance and train a neural network regression to estimate the testing performance of unlabeled set.

Semi-Structured Dataset Representations (SSDR) [56] Similar to the frechet Distance based method mentioned above, it uses a semi-structured dataset feature to regress the model's accuracy.

Average Confidence (AC) [28] It uses the average of the model's confidence (maximum softmax output) as the model's accuracy on the test set.

Difference of Confidence (DoC) [23] This method uses the difference of confidences on source and target distribution to regress the model's accuracy.

Importance-re-weighting (IM) [5] This method estimates the model's accuracy on target data by the importance ratios, by using this, the trained model's accuracy can be converted to the accuracy on the unlabeled target test set.

Generalization Disagreement Equality (GDE) [35] This method first trains two models, which are trained on the same training set but with different initialization or different data ordering. Then, it simultaneously uses the two models to predict, if their predictions disagree, it's considered as an error prediction.

Average Thresholded Confidence (ATC-MC and ATC-NE) [20] This method proposes average thresholded confidence, which learns a threshold on a score of model confidence on validation source data and predicts target domain accuracy as the fraction of unlabeled target points that receive a score above that threshold. ATC-MC uses the mean confidence as the score, while ATC-NE uses the negative entropy.

C. Training Details

Overall, we train classification along with SimCLR in a multi-task way. Here are some detailed training parameters under different setups. **MNIST** We train LeNet-5 on MNIST. We choose the Adam optimizer, with learning rate $3e^{-4}$, and train 700 epochs with batch size 2048.

CIFAR-10 and CIFAR-100 For CIFAR-10 and CIFAR-100, the model architecture we use is DenseNet-40-12 (40 layers with growth rate 12). In the training phase, we use the SGD optimizer with momentum 0.9, and train 300 epochs with batch size 128. The initial learning rate is 0.1, and decay by multiplying 0.1 at epoch 150 and epoch 225.

COCO We use pre-trained ResNet-50, and train 50 epochs with batch size 128. For training, we use the SGD optimizer with momentum 0.9. The initial learning rate is $1e^{-3}$, decayed by multiplying 0.1 at epoch 20 and epoch 30.

TinyImageNet We use pre-trained ResNet-50, and train 50 epochs with batch size 128. For training, we use the SGD optimizer with momentum 0.9. The initial learning rate is $5e^{-3}$, decayed by multiplying 0.1 at epoch 20 and epoch 30.

D. Sample Visualization of Synthetic Sets

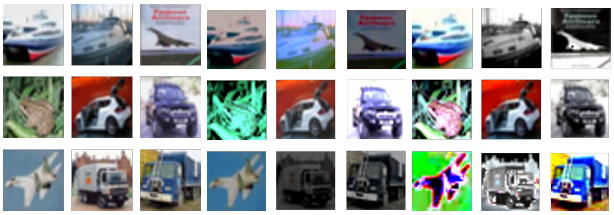
In section 3.4, we describe how we synthesize sample sets by applying various transformations on the original

Table 9: Details of the datasets considered in our paper, where the validation set that has not undergone data transformation is used as the seed set.

Train set (source)	Valid set (source)	Unseen test set (target)
MNIST (train)	MNIST (valid)	USPS, SVHN
CIFAR-10 (train)	CIFAR-10 (valid)	CIFAR-10.1, 95 CIFAR10-C (Fog and Motion blur, etc.)
CIFAR-100 (train)	CIFAR-100 (valid)	95 CIFAR-100-C (Fog and Motion blur, etc.)
COCO 2014 (train)	COCO 2014 (valid)	Caltech256 (test), PASCAL VOC 2007 (test), ImageNet (test)
TinyImageNet (train)	TinyImageNet (valid)	95 TinyImageNet-C (Fog and Motion blur, etc.)

(a) original set (b) sample set 1 (c) sample set 2

Figure 11: MNIST sample



(a) original set (b) sample set 1 (c) sample set 2

Figure 12: CIFAR-10 sample

seed set. Here we provide some visualizations for the generated sample sets (see Figure 11, 12).

E. Additional Theoretical Discussion

Recalling to Theorem 1.1, we provide some more detailed discussions of this theorem at here, including its basic assumptions and a extended theorems under weaker conditions [59].

Assumption E.1 $\forall x, x^+ \sim p(x, x^+)$, *the labels are deterministic(one-hot) and consistent:* $p(y|x) = p(y|x^+)$.

Under the premise of satisfying the natural and minimum assumption — label consistency, we can extend Theorem 1.1 to the situation of any model:

Theorem E.1 *For any model* $f \in \mathcal{F}$*, its downstream classification risk* $\mathcal{L}^{\mu}_{CE}(f)$ *can be bounded by the contrastive learning risk* $\mathcal{L}_{NCE}(f)$

$$\mathcal{L}_{NCE}(f) - \sqrt{Var(f(x)|y)} - \frac{1}{2}\sum_{j=1}^{m}\sqrt{Var(f_j(x)|y)} - \mathcal{O}(M^{-1/2}) \leq \mathcal{L}^{\mu}_{CE}(f) + \log(M/K) \leq \mathcal{L}_{NCE}(f) + \sqrt{Var(f(x)|y)} + \mathcal{O}(M^{-1/2}) \tag{9}$$

where $\mathcal{L}^{\mu}_{CE}(f) = \mathbb{E}_{p(x,y)}[-\log \frac{\exp(f(x)^T \mu_y)}{\sum_{i=1}^{K}\exp(f(x)^T \mu_i)}]$, $\log(M/K)$ *is a constant,* $\mathcal{O}(M^{-1/2})$ *denotes the order of the approximation error by using* M *negative samples,* $f_j(x)$ *denotes the* j*-th coordinate of* $f(x)$*, and* $Var(f(x)|y) = \mathbb{E}_{p(y)}[\mathbb{E}_{p(x|y)}||f(x) - \mathbb{E}_{p(x|y)}f(x)||^2]$ *denotes the conditional variance.*

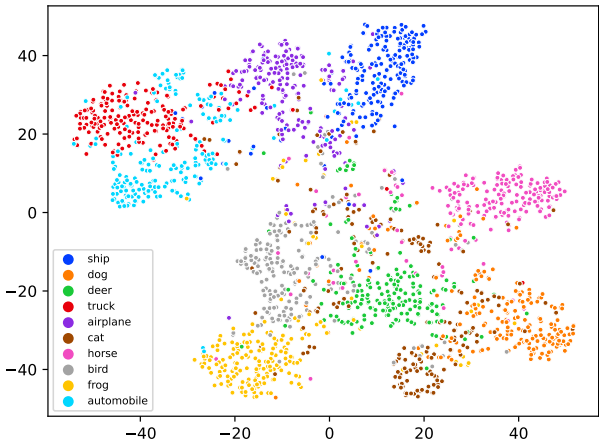
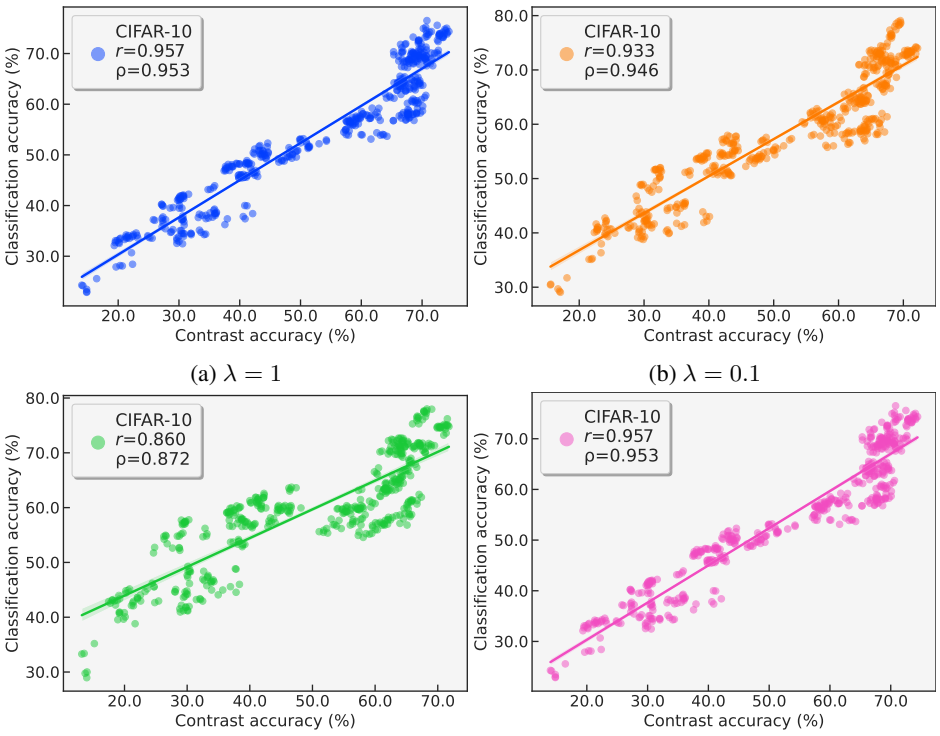


Figure 13: T-SNE clustering visualization of the classification features on CIFAR-10.1 test set. Different colors correspond to different classes.

F. Tightness Analysis of Bounded Theorem

In this section, we aim to delve into the tightness of the bounds defined by Theorem E.1, specifically examining whether the variance term under domain shift without fine-tuning on the test set can be neglected. Revisiting the literature [59], eliminating the troublesome variance term must be based on the concurrent fulfillment of the *Intra-class Connectivity* and *Perfect alignment* assumptions. The



(a) $\lambda = 1$ (b) $\lambda = 0.1$

(c) $\lambda = 0.01$ (d) $\lambda = 0.001$

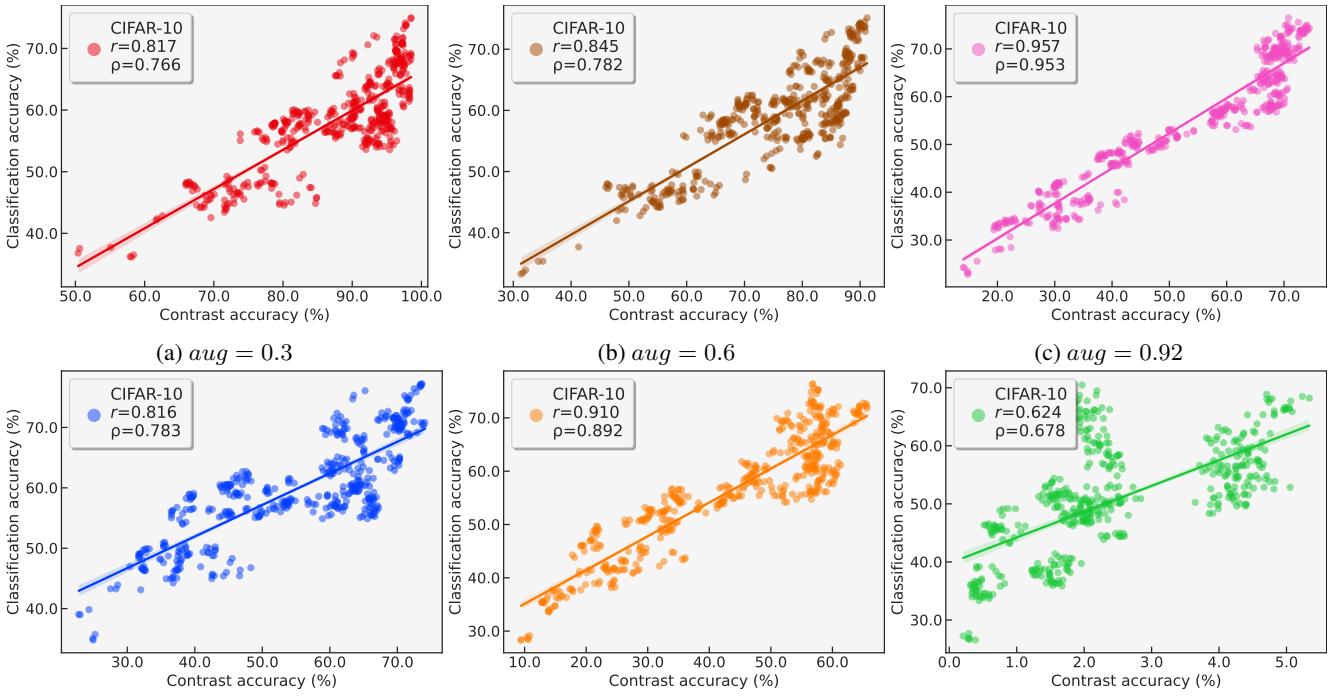
Figure 14: Scatter plots of the linear correlation with different contrastive learning task weights (λ).

Intra-class Connectivity is defined as follows: for a given data set $\mathcal{D}$, there exists a appropriate augmentation set $\mathcal{T}$ in which different intra-class samples can overlap with an aggressive augmentation from $\mathcal{T}$. The *Perfect alignment* means that the classifier has a minimum InfoNCE Loss. To this end, we did an interesting visualization experiment on the CIFAR-10.1 test set.

In Figure 13, we present an intriguing visualization of the CIFAR-10.1 test set. Our method concurrently attains high contrastive accuracy (88.47%) and well intra-class clustering effect in CIFAR-10.1. This implies that we are able to fulfill the above-mentioned assumptions of negligible variance term under the shifted test set. These results further guaranteed that our CL accuracy can being a good indicator of classification accuracy in widely spread unseen test distributions.

G. Linear correlation with different training settings.

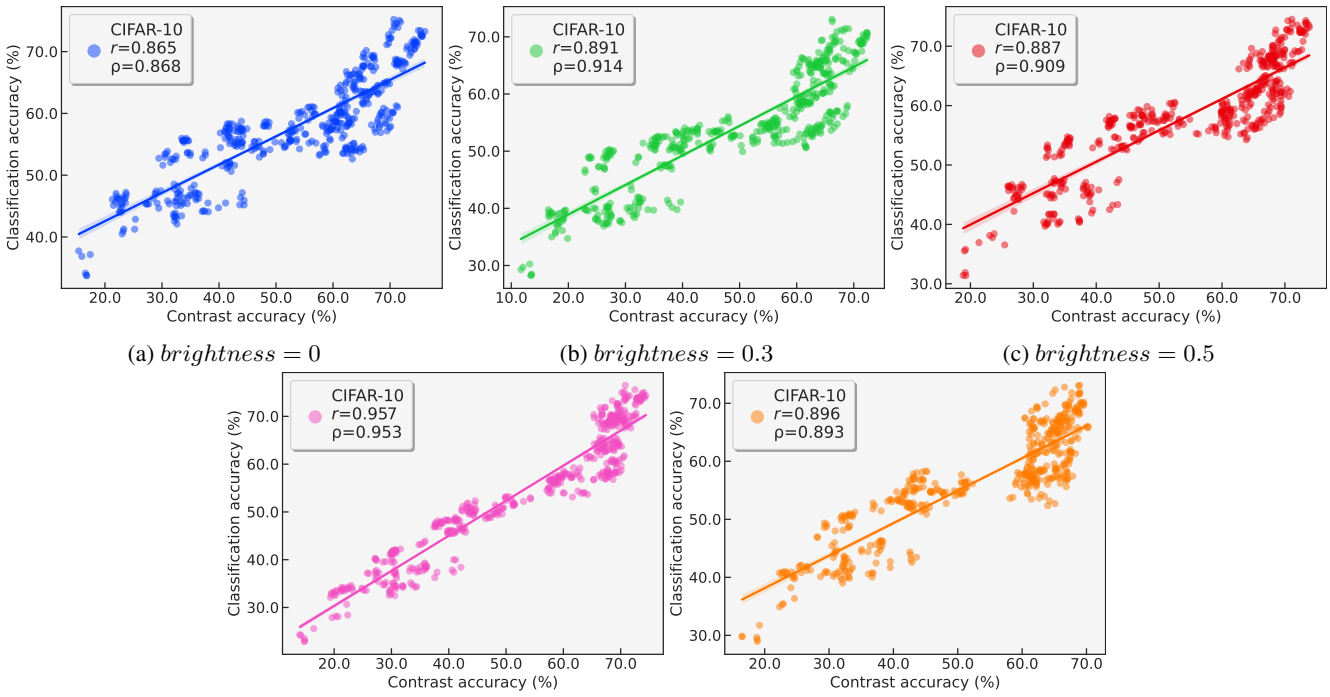
In this section, we display the scatter plots of linear correlations under different training settings, as follows in Figure 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24.



(a) $aug = 0.3$ (b) $aug = 0.6$ (c) $aug = 0.92$

(d) $aug = 1.2$ (e) $aug = 1.5$ (f) $aug = 1.95$

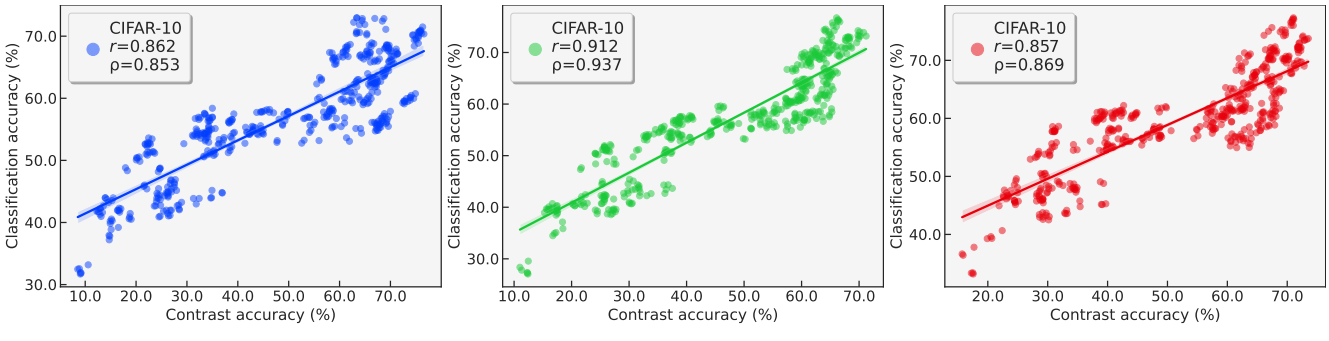
Figure 15: Scatter plots of the linear correlation with different RandomResizedCrop augmentation strengths (aug).



(a) $brightness = 0$ (b) $brightness = 0.3$ (c) $brightness = 0.5$

(d) $brightness = 0.8$ (e) $brightness = 1.0$

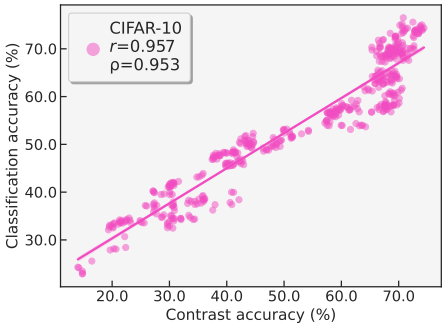
Figure 16: Scatter plots of the linear correlation with different color jittering parameter $brightness$.



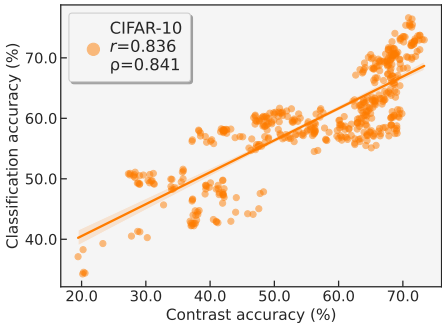
(a) $contrast = 0$

(b) $contrast = 0.3$

(c) $contrast = 0.5$

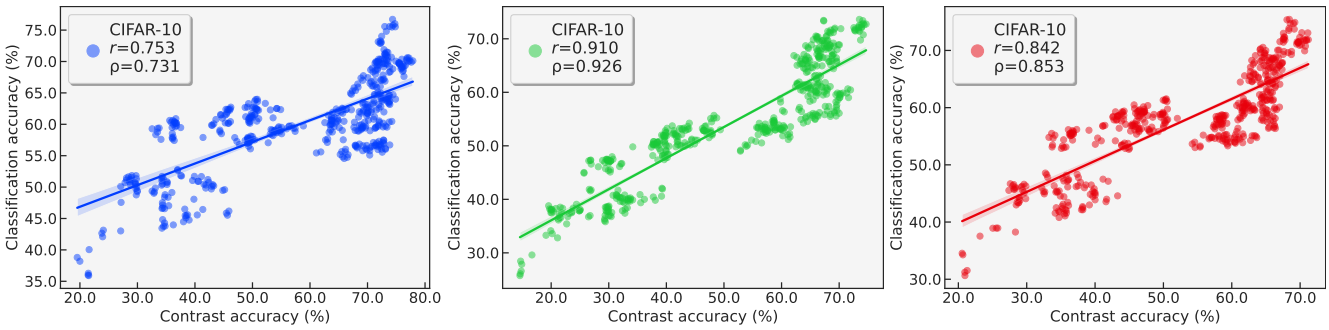


(d) $contrast = 0.8$



(e) $contrast = 1.0$

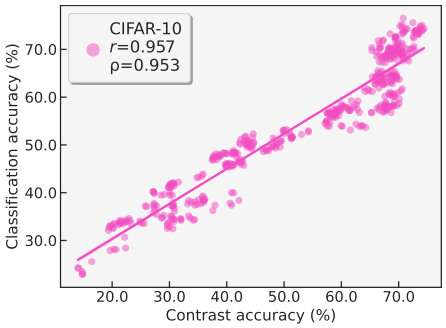
Figure 17: Scatter plots of the linear correlation with different color jittering parameter $contrast$.



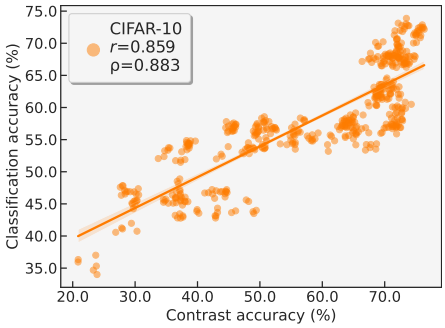
(a) $saturation = 0$

(b) $saturation = 0.3$

(c) $saturation = 0.5$

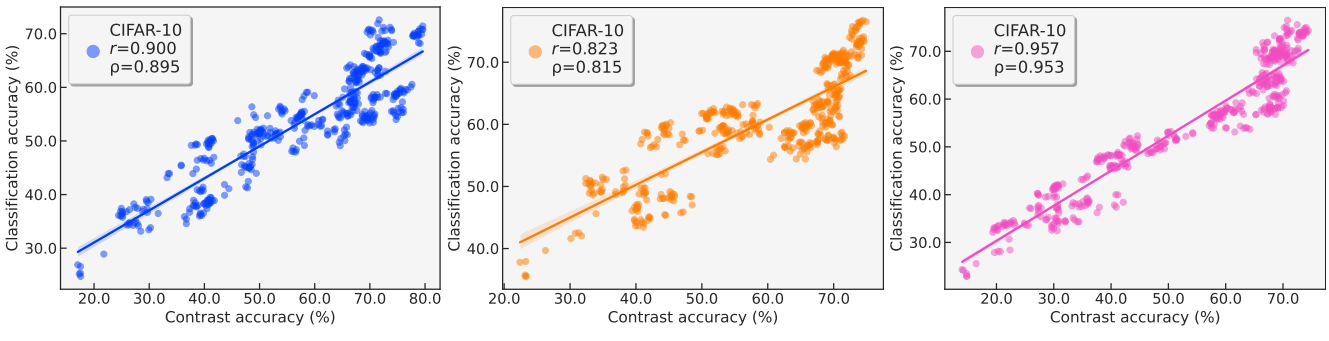


(d) $saturation = 0.8$



(e) $saturation = 1.0$

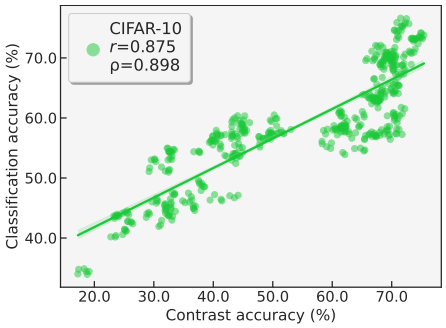
Figure 18: Scatter plots of the linear correlation with different color jittering parameter $saturation$.



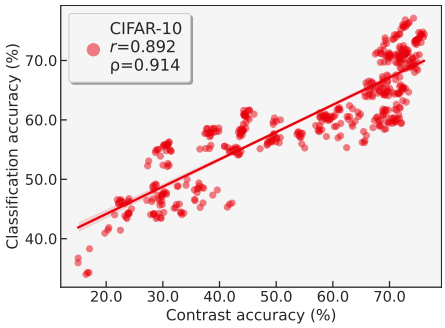
(a) $hue = 0$

(b) $hue = 0.1$

(c) $hue = 0.2$

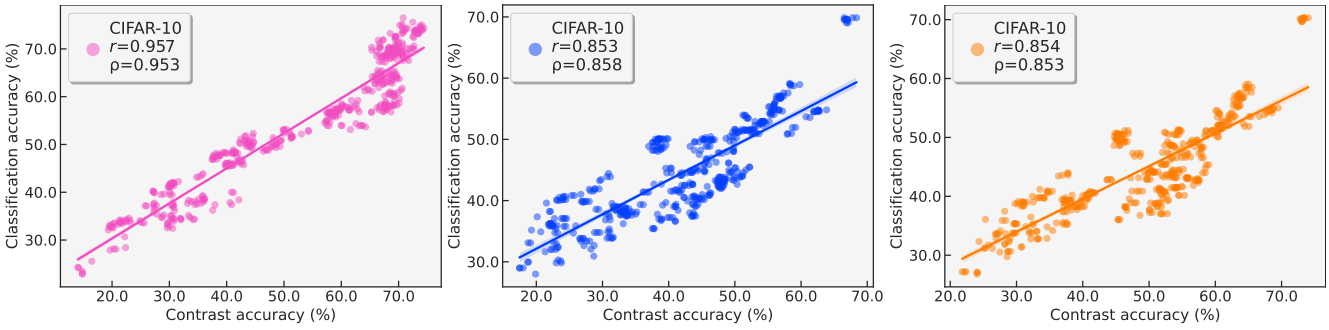


(d) $hue = 0.4$



(e) $hue = 0.5$

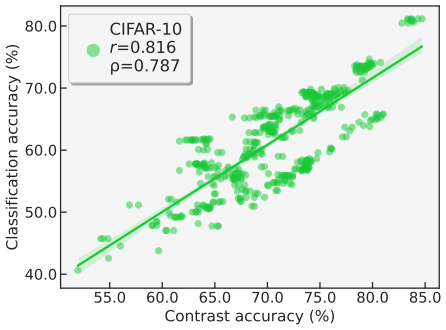
Figure 19: Scatter plots of the linear correlation with different color jittering parameter hue.



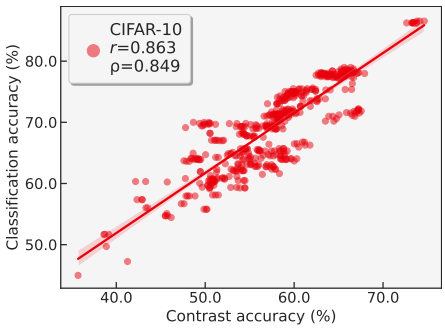
(a) DenseNet-40-12

(b) ResNet-18

(c) ResNet-34

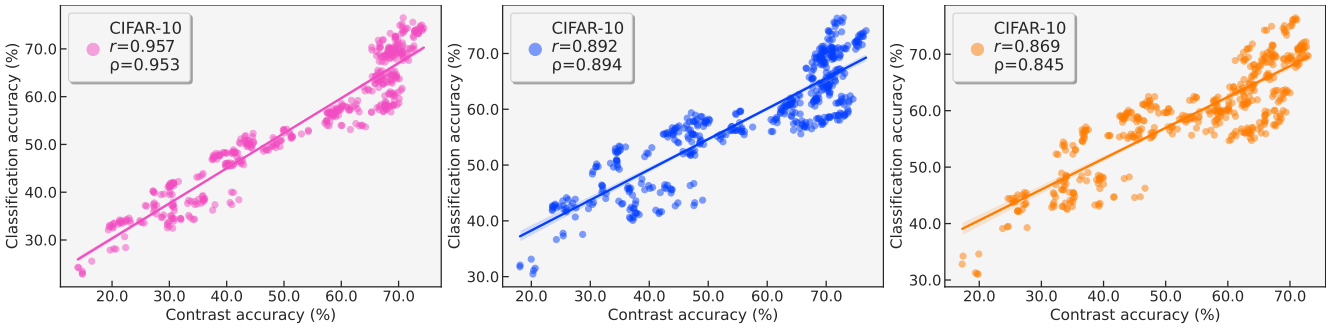


(d) VGG-11



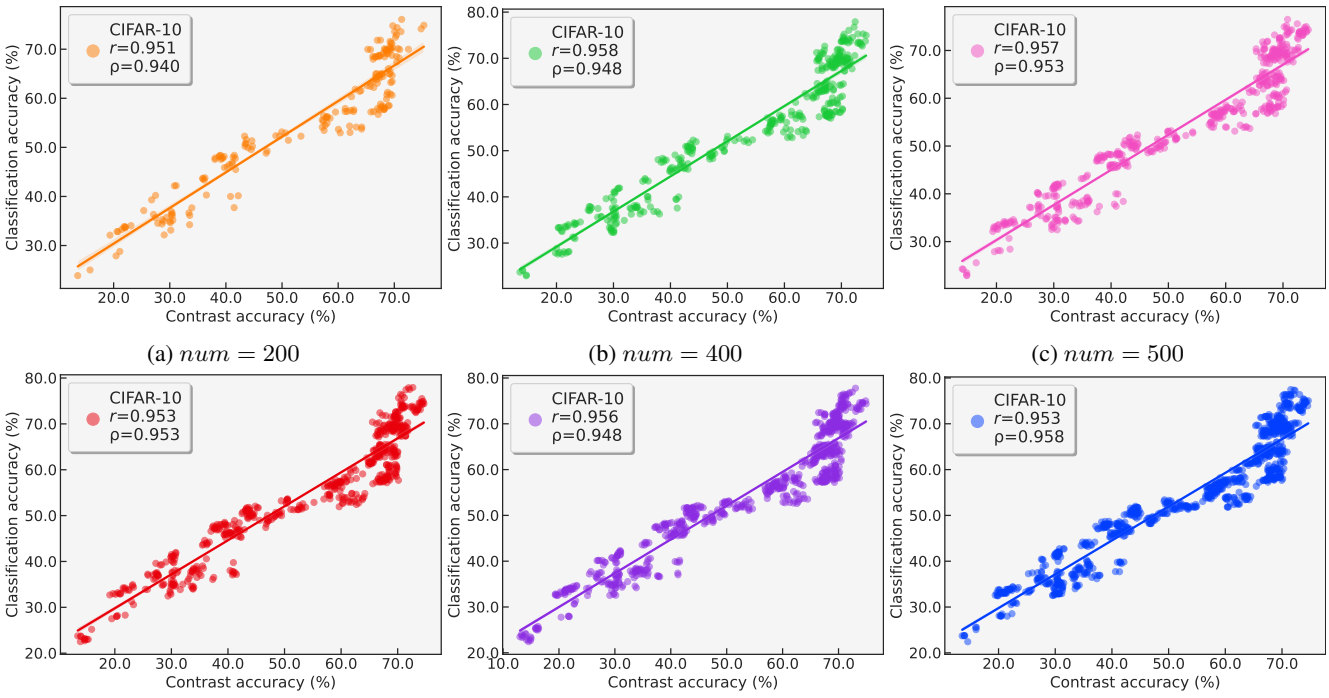
(e) VGG-19

Figure 20: Scatter plots of the linear correlation with different backbones.



(a) $seed = 0$ (b) $seed = 21$ (c) $seed = 42$

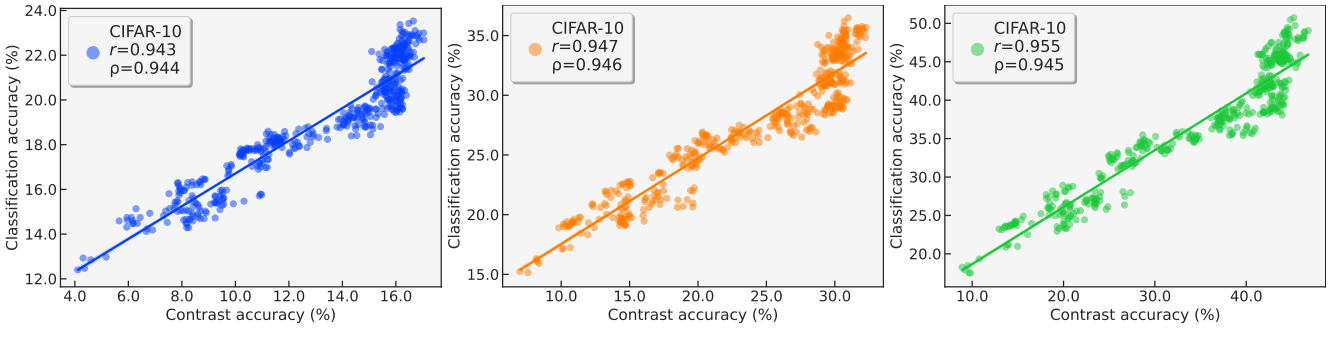
Figure 21: Scatter plots of the linear correlation with different random seeds.



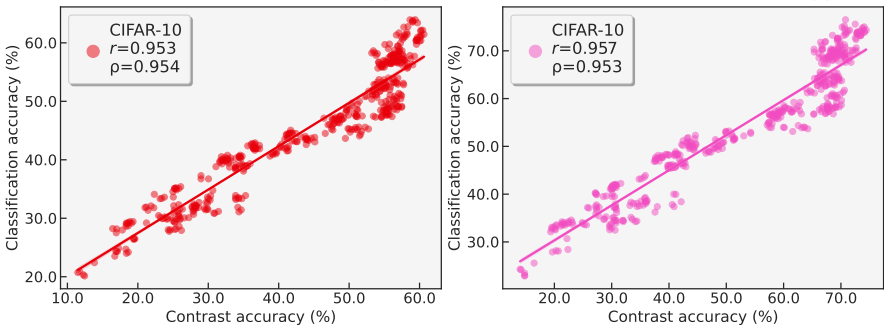
(a) $num = 200$ (b) $num = 400$ (c) $num = 500$

(d) $num = 600$ (e) $num = 800$ (f) $num = 1000$

Figure 22: Scatter plots of the linear correlation with different sample set amounts num (each sample set contains 10000 images).

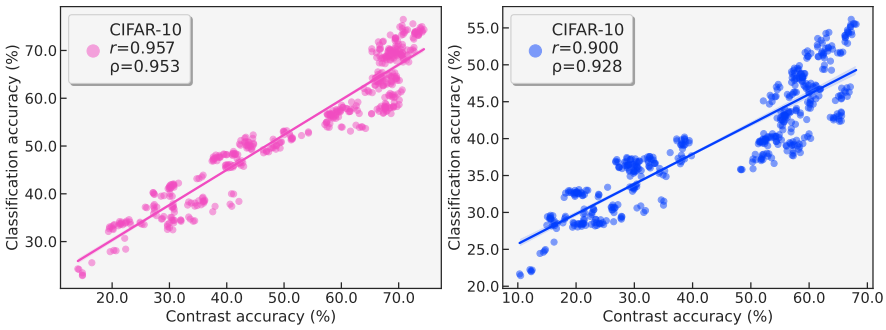


(a) $size = 2000$ (b) $size = 4000$ (c) $size = 6000$



(d) $size = 8000$ (e) $size = 10000$

Figure 23: Scatter plots of the linear correlation with different sample set sizes $size$ (using 500 sample sets).



(a) Multi-task (b) Pretain + fine-tune

Figure 24: Scatter plots of the linear correlation with training ways.