CAME: Contrastive Automated Model Evaluation

Ru Peng[1]*, Qiuyang Duan[1]*, Haobo Wang[1], Jiachen Ma[1,2], Yanbo Jiang[1], Yongjun Tu[3], Xiu Jiang[3], Junbo Zhao[1]†

[1]Zhejiang University [2]DI-Lab, Zhejiang University [3]OPPO

{rupeng,duanqiuyang,wanghaobo,mjc_zjdx,yanbojiang,j.zhao}@zju.edu.cn
{tuyongjun1,jiangxiu}@oppo.com

Abstract

*The Automated Model Evaluation (AutoEval) framework entertains the possibility of evaluating a trained machine learning model without resorting to a labeled testing set. Despite the promise and some decent results, the existing AutoEval methods heavily rely on computing distribution shifts between the unlabelled testing set and the training set. We believe this reliance on the training set becomes another obstacle in shipping this technology to real-world ML development. In this work, we propose **C**ontrastive **A**utomatic **M**odel **E**valuation (**CAME**), a novel AutoEval framework that is rid of involving training set in the loop. The core idea of CAME bases on a theoretical analysis which bonds the model performance with a contrastive loss. Further, with extensive empirical validation, we manage to set up a predictable relationship between the two, simply by deducing on the unlabeled/unseen testing set. The resulting framework CAME establishes a new SOTA results for AutoEval by surpassing prior work significantly.* [1]

1. Introduction

During the last decade, the technological advancement of artificial intelligence and machine learning has attained unprecedented achievements, affecting a variety of domains or verticals. Ubiquitously, off these milestones, to properly evaluate, assess and benchmark the trained models is undoubtedly pivotal, particularly when considering the deployment towards the production in real-world scenarios. To do that, the traditional means often relies on a pre-split and static testing set for model evaluation, which is principally left out of the sight during training or validation phase. However, several recent works has pointed out the drawback of this standardized scheme due to its requirement of careful sample selection, randomization due to the sample set split, the OOD gap between the deployment environment, and (somewhat) expensive label annotation [14, 13, 5], etc. Most recently, we see that *Automated Model Evaluation* (AutoEval) has emerged to tackle these problems [14].

In particular, the vanilla prototype of the Automated Model Evaluation approaches aim at estimating a provided model's performance on an unlabeled testing set. Notably, these approaches first generate a few *meta-sets* by adopting pre-selected data augmentations on the training set. In what follows, one can estimate a certain distance — for instance, the Frechet distance [16] — averaged between the meta-sets with the testing set. As a result, the prior work has proactively shown that this averaged distance measurement is related to the final model performance on the testing set. Indeed, we believe this setup of AutoEval on the testing set possesses positive prospects because it manifests a high similarity towards real production — where the testing set is acquired on the fly in the real-world, leaving no time/space for these samples to be annotated or persist. A graphical illustration of the AutoEval against the conventional static testing set evaluation is depicted in Figure 1.

Despite its promise and prospect, we realize that the current paradigm of AutoEval may still fail in its real-world deployment, under certain conditions. On one hand, it is widely acknowledged that the prior works are dedicated to avoiding annotating the testing samples and to amortizing the vexing randomness through the massive generation of meta-sets offline. On the other hand, however, these techniques still demand the full presence of the sample input from the training set, which in many — if not most — of the occasions probably imply expensive storage and computation cost. Hence, we argue that this requirement cannot be easily ensured in many scenarios, most notably on limited-capacity, limited-storage, or low-power platforms such as edge devises for IOT or autonomous driving. Hereby, we

*Both authors contributed equally to this research.
†Corresponding author.
[1]Our code is publicly available at: https://github.com/pengr/Contrastive_AutoEval

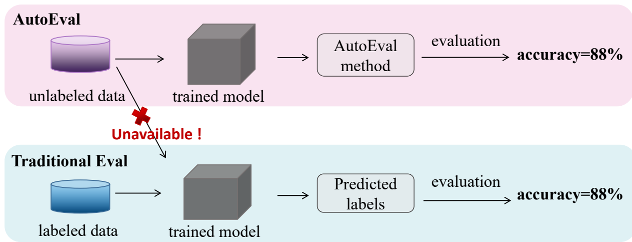


Figure 1: Illustration of workflow differences between AutoEval and traditional model evaluation.

pose the core motivation of the design of this work: can we establish an AutoEval framework without keeping the training set in the loop?

To reach this target is not trivial, and it cannot be achieved by incrementally changing the prior method. This is mostly due to the heavy bond of the final model performance regressor with the meta-sets induced from the training data. In this work, we hope to break this paradigm commonly used in prior work, and propose a novel paradigm — **C**ontrastive **A**utomatic **M**odel **E**valuation, dubbed **(CAME)**. Unlike the previous approaches, CAME aims to regress to the final model performance that assumes the absence of the training data. In particular, CAME is very much motivated by the following series of the theories:

Theorem 1.1 *[59] Given a model f with an optimal functional minimizer $f^* = \arg\min \mathcal{L}_{NCE}(f)$, its classification risk can be upper- and lower-bounded by its contrastive learning risk as*

$$\mathcal{L}_{NCE}(f^*) - \mathcal{O}(M^{-1/2}) \leq \mathcal{L}^{\mu}_{CE}(f^*) + \log(M/K) \leq \mathcal{L}_{NCE}(f^*) + \mathcal{O}(M^{-1/2}) \tag{1}$$

where M is the number of negative samples in contrastive learning, K is the number of classes, $\mathcal{L}_{NCE}$ [50] is the InfoNCE loss for contrastive learning, and $\mathcal{L}^{\mu}_{CE}$ [59] is the mean CE Loss used to indicate the downstream classification risk (definitions in section 3).

Theorem 1.1 indicate that under mild assumptions, the contrastive loss constantly bounded the cross-entropy loss and thus, can reflect the overall trends of generalization. Moreover, analogous theoretical guarantees of bounding CE loss through CL risk are also evident in [53] and [2]. Notably, in the AutoEval problem, with distribution shifts and the absence of ground-truth labels on the test sets, the cross-entropy loss $\mathcal{L}_{CE}$ is inaccessible. Fortunately however, $\mathcal{L}_{NCE}$ is self-supervised and can be inferred purely from testing inputs. Based on the theoretical analysis, we cast a hypothesis as follows. The contrastive loss — calculated from the testing set alone — is informative towards predicting the performance of the provided model.

To this regard, we briefly introduce our framework, CAME. It is prerequisite composed of two conditions: (i)-the model is trained jointed of a normal task loss together with a contrastive loss and (ii)-the model performance is not affected by jointly contrastive learning. Based on the model yielded this way, we conduct a rigorous empirical study — guided by the theories we pose above — that we prove the correlation between the contrastive loss on the testing set with its performance truly exists. The AutoEval established this way enjoys the following two major merits: (i)-it shreds the need for the training set during the evaluation phase which further extends the AutoEval technique towards production in real-world; (ii)-CAME sets a new record of testing performance estimation by exceeding the prior work by significant margins.

2. Related Works

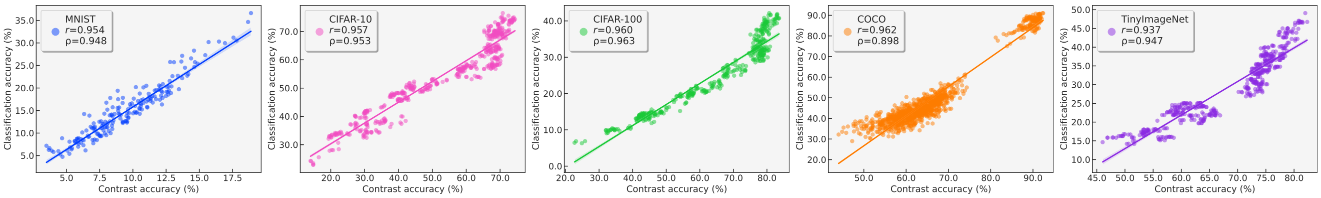
Automated Model Evaluation. [14, 13, 56, 60] build regression models on many test sets with distribution shifts to predict model's accuracy on an unlabeled test set. [23, 20] use confidence based methods, . Our work does not use distribution shift measurements. Instead, we directly use contrastive accuracy to regress classification accuracy.

Model Generalization Prediction. This problem is to estimate the generalization gap and predict the generalization error. From model perspective, [10, 34, 35] predict generalization error by leveraging model parameters. From data perspective, [9, 62] predict the generalization gap under distribution shifts via data representations. Different from these works, our work aims to directly predict a model's accuracy on unseen unlabeled test sets.

Out-of-Distribution Detection. The OOD Detection task [28, 44, 46, 26] is to detect test samples subject to a distribution different from the training data distribution. It focuses on the data distribution of training data and testing data. Different from this, our work focuses on estimating model's accuracy on unlabeled OOD test set.

Contrastive Learning. Contrastive Learning (CL) [6, 24, 7, 22, 3, 8] is a typical self-supervised learning paradigm to learn effective representation of input samples. Used as a pre-training task, it can significantly enhance the downstream semantic classification task, which means CL learns information-rich features for object classification. Inspired by these pioneering efforts, we choose contrastive learning as the auxiliary learning task, and take the classical CL framework (SimCLR) to validate the feasibility of CL in this work.

Unsupervised Domain Adaptation. Unsupervised domain adaptation is also an active research field, which aims to use labeled source data and unlabeled target data to learn a model generalizing well from the source domain to the target domain. In recent years, many researches such as [55, 47, 57, 43, 31] have proposed different measures and



(a) MNIST (b) CIFAR-10 (c) CIFAR-100 (d) COCO (e) TinyImageNet

Figure 2: Experimental results about the correlation study. On the above different datasets, we show that there exists strong linear correlation between contrastive accuracy (x-axis) and classification accuracy (y-axis). The symbols r represents Pearson's correlation coefficient and ρ indicates Spearman's rank correlation coefficient.

frameworks to promote the development of this field. In this work, given a model trained on the source dataset, we focus on obtaining a precise estimation of its accuracy on unlabeled target sets.

3. Contrastive Automated Model Evaluation

In this section, we describe our proposed method, CAME, in detail. The core idea of CAME is simple. We first formulate a multi-task learning framework by integrating a normal task loss with a contrastive learning loss objective. After attaining the model through regular optimization process, on an unseen and unlabeled testing set, we build a simple and separate neural network to regress from the contrastive loss to a proximal model performance. A pseudo code is provided in algorithm 1.

3.1. Problem Definition

Consider a image classification task, we aim to estimate the classification performance of a trained classifier on different unseen and unlabeled test sets automatically. We denote the training set as $\mathcal{D}_o = \{(x_i, y_i)\}_{i=1}^{I}$, where $y_i \in \{1, 2, \cdots, K\}$ denotes the label of the image x_i. These unseen and unlabeled test set are denoted as $D_t = \{\{x_j\}_{j=1}^{J_1}, \ldots, \{x_j\}_{j=1}^{J_M}\}$, where $\{x_j\}_{j=1}^{J_M}$ represent the m-th unseen test set. Based on the theoretical anaylsis in Theorem 1.1 — "*For the contrastive learning model, its classification risk can be upper and lower bounded by its contrastive risk in any unseen test data distribution*", which means that it is feasible to predict classification accuracy with contrastive accuracy. Motivated by this exhilarating finding, in unseen test set $D_t^m = \{x_j\}_{j=1}^{J_m}$, we fit a linear regressors [32] $R : (f, g, D_t^m) \rightarrow \hat{Acc}$ to predict a classifier's accuracy $\hat{Acc}_{cla}$ by its contrastive accuracy:

$$\hat{Acc}_{cla} = R(f, g, D_t^m) = R(Acc_{con}), \tag{2}$$

where f, g are the shared encoder and contrastive projection head in Section 3.3.

Algorithm 1 Contrastive Automated Model Evaluation

Input: Training set $\mathcal{D}_o$, seed set $\mathcal{D}_e$, unseen test set $\mathcal{D}_t$; shared encoder f, projection heads h, g; contrastive loss weight λ; data transformations $t \sim \mathcal{T}_s$

// Multi-task learning

for $epoch = 1, 2, ...T$ **do**
  sample the mini-batch $\mathcal{D}_l$, $\mathcal{D}_u$ from $\mathcal{D}_o$
  for $x \in \mathcal{D}_l$ **do**
    $z = g(f(x)) \in \mathbb{S}^{2N-1}$
    $Wf(x) = h(f(x)) \in \mathbb{R}^K$
    $\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{NCE}$, $\mathcal{L}_{CE}$ and $\mathcal{L}_{NCE}$ in Sec. 3.3
  end for
end for

// Synthesizing Sample Sets

for $i = 1, 2, ..., a$ **do**
  $\mathcal{D}_s^i = t_i(\mathcal{D}_e)$
  $Acc_{con}^i = \sum_{i=0}^{N-1} \mathbb{I}\left[i = \underset{j\in[0,N-1]}{\arg\max}\, g_j(f(D_s^i))\right] / N$
  $Acc_{cla}^i = \sum_{i=0}^{N-1} \mathbb{I}\left[y = \underset{i\in\{1,2,\cdots,K\}}{\arg\max}\, h_i(f(D_s^i))\right] / N$
end for
Correlation Coefficients: r, ρ statistic from $\mathcal{D}_s = \{(Acc_{con}^1, Acc_{cla}^1), \ldots, (Acc_{con}^a, Acc_{cla}^a)\}$
Linear regressor $R : Acc_{cla} = W^\top(Acc_{con}) + b$

// Automated Model Evaluation via Regression

In $\mathcal{D}_t$: $\hat{Acc}_{cla} = R(Acc_{con})$; Acc_{con}, $\hat{Acc}_{cla}$ in Eq. 8
Mean Absolute Error: $\varepsilon = |Acc_{cla} - \hat{Acc}_{cla}|$

Output: Pearson's correlation r, Spearman's rank correlation ρ and Mean Absolute Error ε.

3.2. Correlation Analysis

To further corroborate the feasibility of predicting classification accuracy from contrastive learning accuracy, we analyze the correlation between the two sources of accuracies among various data setup in Figure 2. For each data setup, we train CAME on the training set, then evaluate its accuracy pairs on each synthetic sample set (mentioned in

Section 3.4). Finally, we report the scatter plot and both Pearson's correlation coefficient r and Spearman's rank correlation coefficient ρ. Here, each data point in the scatter plot corresponds to the accuracy pair of a synthetic sample set. From Figure 2, we can see that among various data environments, the contrastive learning accuracy and classification accuracy exhibit strong linear correlation ($r > 0.937$ and $\rho > 0.898$). This finding prompts us to train a linear regressor to predict classification accuracy on unlabeled unseen test distribution, rather than building a non-linear MLP like previous work.

3.3. Multi-task Learning

In this section, we begin by introducing how CAME trains the model via pivotal multi-task learning, to reveal the inherent strong correlation between classification accuracy and contrastive learning accuracy. In summary, the co-training paradigm consists of the supervised classification and the self-supervised contrastive learning. Specifically, given a mini-batch of N labeled samples $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^N$ where labels $y_i \in \{1, \ldots, K\}$ and N unlabeled samples $\mathcal{D}_u = \{x_i\}_{i=1}^N$. We adopt a network-unrestricted encoder $f \in \mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^d$ to extract image representations from input samples. Then, we apply two projection heads $g : \mathbb{R}^d \rightarrow \mathbb{S}^{2N-1}$ and $h : \mathbb{R}^d \rightarrow \mathbb{R}^K$ to map the shared representations into two spaces where contrastive loss and classification loss are applied, respectively. The illustration of our model is shown in Figure 3.

Contrastive Learning. Taken a training sample $x \in \mathcal{D}_u$, we apply a set of random data augmentation $t \sim \mathcal{T}$ to generate its positive sample $x^+ = t(x)$, and treat the other augmented samples $\{x_i^-\}_{i=1}^{2(N-1)}$ within a mini-batch as its negative samples. Then, the shared encoder f can be learned by the InfoNCE contrastive loss [50] which mapping from the d-dimensonal image representations to a unit hypersphere:

$$\mathcal{L}_{\mathrm{NCE}}(f, g) = \mathop{\mathbb{E}}_{p(x,x^+)} \mathop{\mathbb{E}}_{\{p(x_i^-)\}} \left[-\log \frac{\exp\left(z^\top z^+\right)}{\sum_{i=1}^{2N-1} \exp\left(z^\top z_i^-\right)} \right], \quad (3)$$

where $z = g(f(x))$, $p(x)$ is the data distribution, and $p(x, x^+)$ is the joint distribution of positive data pairs.

Classification Learning. Given a labeled data pairs $(x, y) \in \mathcal{D}_l$, we usually use the cross-entropy (CE) loss to train the shared encoder f through classification learning:

$$\mathcal{L}_{\mathrm{CE}}(f, h) = \mathbb{E}_{p(x,y)} \left[-\log \frac{\exp\left(f(x)^\top w_y\right)}{\sum_{i=1}^K \exp\left(f(x)^\top w_i\right)} \right], \quad (4)$$

where $h = [w_1, w_2, \ldots, w_K]$ is the classification head.

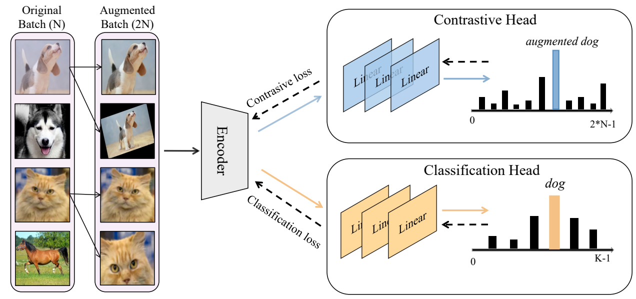


Figure 3: The model architecture of our multi-task learning. Here, we adopt SimCLR [6] as the contrastive learning framework. Note that we only consider the original batch when calculating the classification accuracy.

Hereafter, we train our classifier based on the aforementioned multi-task learning paradigm, by minimizing the following loss:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{NCE}, \quad (5)$$

where λ is a weighing coefficient.

3.4. Synthesizing Sample Sets

To fit the linear regressor (Eq. 2) for performance prediction, we need to collect the contrastive accuracy and the counterpart classification accuracy in various test environments by the multi-task-learning based model. For this, we require many test sets which should include: *i)-various distributions, ii)-the same classes as the training set but without image overlap,* and *iii)-sufficient samples.* However, one obstacle still is to collect such testing sets from natural distributions. As a surrogate, we synthesize these "sample sets". Overall, we synthesize these sample sets by applying a combination of transformations $t \sim \mathcal{T}_s$ on a seed dataset. Note that all possibilities of the transformation sequences are not calculated by permutation and combination, so there are many random states when applying these transformations. Therefore, we generate image sets of various distributions. For all the data transformation setup, we inherit the labels from the seed dataset directly because the transformations do not alter the profound semantics. Meanwhile, We generate 400 synthetic sample sets to calculate the accuracy pairs $\{(Acc_{con}^1, Acc_{cla}^1), \ldots, (Acc_{con}^a, Acc_{cla}^a)\}$, which is applied to fit a linear regression model and statistic the correlation among these accuracy pairs. a is the amount of synthetic sample sets, the default value is set as 400. Specific setups are elaborated as follow.

- **MNIST setup.** We synthesize sample sets by applying various transformations on MNIST test dataset. MNIST as simple gray-scale images, we first consider change their black background to color ones. Specifically, for each sample of MNIST, we randomly select an image from COCO dataset and randomly crop a patch, which is then used as the background of the hand-written digit sample. After that, we randomly apply three out of the six transformations on the background-replaced images: $\{$*autoContrast, rotation, color, brightness, sharpness, translation*$\}$.

- **CIFAR-10 setup.** We synthesize sample sets based on CIFAR-10 test set by adopting three image transformations on it. The three transformations are randomly selected from the six transformations of MNIST Setup.

- **CIFAR-100 setup.** We study on CIFAR-100 to explore if a strong correlation still exists between contrastive learning and classification accuracy in more semantic classes case. The applied transformations are followed as CIFAR-10 setup. Here, we use the training split of CIFAR-100 dataset to train our model and the testing split to synthesize the various sample sets.

- **COCO setup.** Following the practice in [13] and [51], we select 12 common classes: aeroplane, bike, bird, boat, bottle, bus, car, dog, horse, monitor, motorbike and person. Resort to the annotation boxes, we crop out the objects of images belonging to these categories in the COCO training set and validation set to build training and testing set for the classification task. Based on the testing set, we use 15 common corruptions from CIFAR-10-C to generate 60 sample sets, and randomly use three transformations in MNIST setup to generate the rest of sample sets.

- **TinyImageNet setup.** The synthetic sample sets of TinyImageNet are synthesized from its validation split. As in the MNIST setup mentioned above, we randomly use three kinds of transformations to synthesize sample sets for linear regression and correlation study.

3.5. Automated Model Evaluation via Regression

As the final step, we propose to use contrastive learning accuracy to regress to the model's classification accuracy on unseen testing set. Notably, for the input and output side of training the regressor, we adopt the soft version attained from contrastive learning and classification tasks. On the input side, we define it as the probability of an augmented sample being positive-sample. For the output, we use the confidence value of the sample being corrected predicted.

Consequently we write down the forms as follow:

$$Acc_{con} = \sum_{i=0}^{N-1} \mathbb{I}\left[i = \underset{j\in[0,N-1]}{\arg\max}\, g_j(f(x))\right]/N, \quad (6)$$

$$Acc_{cla} = \sum_{i=0}^{N-1} \mathbb{I}\left[y = \underset{i\in\{1,2,\cdots,K\}}{\arg\max}\, h_i(f(x))\right]/N \quad (7)$$

$$\varepsilon = |Acc_{cla} - \hat{Acc}_{cla}|, \quad (8)$$

Table 1: Pearson's correlation (r) and Spearman's rank correlation (ρ) on the different data setup (higher is better). "-" indicates that the results are not reported in original paper.

Method	MNIST		CIFAR-10		CIFAR-100		COCO	
	r	ρ	r	ρ	r	ρ	r	ρ
Frechet [14]	0.912	-	-		-	-	0.908	-
Rotation [13]	-	0.960	-	0.981	-	0.950	-	0.881
Jigsaw [49]	-	-	-	0.958	-	-	-	-
CAME (ours)	0.948	0.954	0.953	0.957	0.963	0.960	0.898	0.962

where $\mathbb{I}[\cdot]$ is an indicator function, N is the mini-batch size, i is the index of an image in original batch, y is the ground-truth class label. ε is the mean absolute error (MAE) for classification accuracy estimation.

4. Experiments

4.1. Experimental Setup

We conduct extensive experiments on various datasets. Similar to prior work, the final assessment of CAME is based on a homogeneous but different unseen test environment. For instance, in hand-written datasets, we train our model and regressor on MNIST but test them in SVHN [48] and USPS [33]. The differing distribution of the training and testing set is significant but the tasks are indeed homogeneous. This protocol effectively validates the generalizability of AutoEval approaches.

For natural images, we train DenseNet-40-12 [30] on CIFAR-10 [37], then conducting test on CIFAR-10.1 [52] and CIFAR-10-C [27]. For CIFAR-100 setup [37], we keep the setttings on CIFAR-10, and test it on CIFAR100-C. For COCO setup [45], following [13], we choose 12 object classes (aeroplane, bicycle, bird, boat, bottle, bus, car, dog, horse, tv-monitor, motorcycle, person). To build the training set, object annotation boxes among these 12 classes are cropped from the COCO training images containing them. These images are used to train a ResNet-50 backbone [25]. Similarly, we build unseen testing sets for classification accuracy from Caltech-256 [21], PASCAL VOC 2007 [18], ImageNet [12], which carrying the same 12 categories. For TinyImageNet setup [39], we train ResNet-50 and the testing is conducted on TinyImageNet-C.

4.2. Main Results

In Table 2, we report the mean absolute error (MAE) results of estimating classifier accuracy on unseen test sets. From this table, among all data setup, we conclude that our method reduces the accuracy estimation error by about **47.2%** on average upon prior SOTA method. Further, CAME shows strong performance regardless of the image domains or the granularities of the classification categories.

Table 2: Mean absolute error (MAE) results for evaluating the classifier accuracy on the unseen test sets. The training set of each group is MNIST, CIFAR-10, CIFAR-100, COCO and TinyImageNet, respectively. In MNIST group, the merged cells represent the average MAE value among the SVHN and USPS. "-" indicates that the results are not reported in original paper. The time cost for different algorithms we count at here[2].

Method	MNIST		CIFAR-10		CIFAR-100	COCO			TinyImageNet
	SVHN	USPS	CIFAR-10.1	CIFAR10-C	CIFAR100-C	Caltech	Pascal	ImageNet	TinyImageNet-C
Pred (τ = 0.8) [28, 44]	10.58	21.18	3.00	1.82	-	3.25	2.45	2.66	6.21
Pred (τ = 0.9) [28, 44]	0.99	35.13	1.30	1.26	-	8.31	8.43	8.00	8.54
Entropy (τ = 0.2) [36]	3.57	32.29	1.05	1.80	-	5.81	6.29	5.33	8.48
Ens. AC [38]	80.05	12.84	-	23.7	-	-	-	-	-
Proxy Risk [9]	13.20	1.21	-	5.3	-	-	-	-	-
Ens. RI [4]	79.56	8.01	-	14.9	-	-	-	-	-
Ens. RM [4]	3.88	0.65	-	2.2	-	-	-	-	-
Frechet [14]	0.82	13.94	0.96	1.94	-	13.63	2.26	5.64	8.13
Frechet + μ + σ [14]	2.06	0.03	0.83	1.83	-	4.01	1.63	2.99	7.96
Rotation [13]	1.78	12.42	3.74	1.99	-	1.91	2.86	3.15	8.21
SSDR [56]	0.76	-	0.74	1.28	-	-	-	-	5.95
AC [28, 17]	21.17		9.88	16.50	23.61	-	-	-	32.44
IM [5]	18.48		6.60	12.33	13.69	-	-	-	19.86
DOC [23]	20.19		7.25	13.87	14.60	-	-	-	25.02
GDE [35]	24.42		4.77	6.55	9.85	-	-	-	5.41
ATC-MC [20]	5.02		3.21	4.65	5.50	-	-	-	5.93
ATC-NE [20]	3.14		2.99	4.21	4.72	-	-	-	5.00
CAME (ours)	**0.52**	**0.24**	**0.49**	**0.84**	**2.34**	**0.80**	**0.85**	**0.81**	**2.50**

Additionally, in Table 1 and 3, we report some more detailed results, regarding statistical correlation scores with other intermediate results.

Validity of Multi-task Co-training Setup. To guarantee a fair-minded assessment of model performance, we must ensure that auxiliary contrastive learning task satisfies the following criteria: i)-no extra learning complexity for the main task, ii)-minimal network changes and iii)-does not degrade classification accuracy. To substantiate the soundness of our co-training buildup, we report the ground-truth accuracies of the co-trained classifiers in Table 3. And below, we give an apple-to-apple comparison with classification-only model – FD [14]: SVHN (23.42 vs. 25.46), USPS (88.64 vs. 64.08), Pascal (85.04 vs. 86.13), Caltech (96.08 vs. 93.40), ImageNet (82.66 vs. 88.83). Drawing from these results we can know that co-training as a feasible strategy will not degrade the performance of the model to be evaluated. This signifies our adherence to the principle of fairly evaluating a model's performance that deserves testing.

4.3. Ablation Studies

4.3.1 Contrastive Learning Parameters

As we adopt SimCLR in our framework, we want to know if our overall performance is consistent to its basic

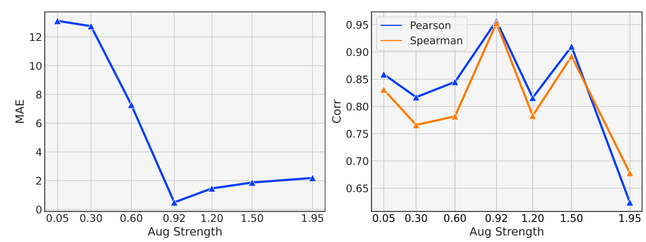


Figure 4: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different augmentation strength on the RandomResizedCrop operation under CIFAR-10 setup.

settings (i.e. the linear correlation coefficient achieves its maximum and the estimation error achieves its minimum under the best training parameters of SimCLR). According to [59], among the data augmentations adopted in SimCLR, RandomResizedCrop is the most important augmentation, and ColorJitter is the second. So we study the impact of these two kinds of augmentations on CIFAR-10 in our work.

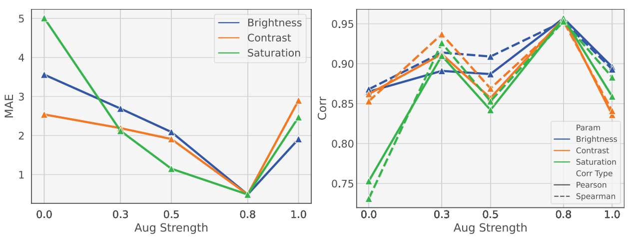
For **RandomResizedCrop**, to quantify its influence, we use the augmentation strength defined in [59]. For a RandomResizedCrop operator with scale range $[a, b]$, its aug-strength can be defined as $r = (1 - b) + (1 - a)$. In Figure 4, we show that under different augmentation strengths, the accuracy estimation error achieves its minimum at the default strength value 0.92.

For **ColorJitter**, we study its parameters: brightness, contrast, saturation and hue, where the augmentation strength is corresponding to the parameter value. Note that

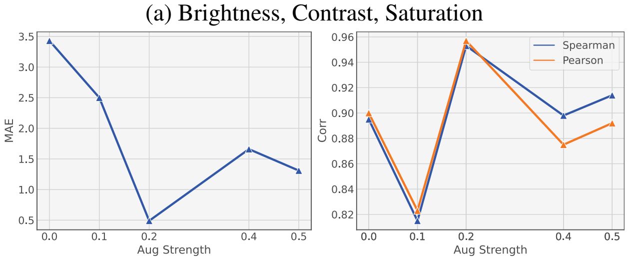
[2]Since the pipeline of each algorithms to came up with the accuracy evaluation is vastly different, we simplify compare their time complexity at the algorithm category level: *confidence-based* $<$ *regression-based* ***(we are here)*** $<$ *agreement-based* $<$ *distribution statistics* $<$ *self-training*.

Table 3: Results of ground-truth classification accuracy, predicted classification accuracy, contrastive learning accuracy and MAE from CAME (ours) in different data setups.

Dataset	MNIST		CIFAR-10		CIFAR-100	COCO			TinyImageNet
	SVHN	USPS	CIFAR-10.1	CIFAR10-C	CIFAR100-C	Caltech	Pascal	ImageNet	TinyImageNet-C
Ground-truth Cla. Acc.	23.42	88.64	80.80	73.71	48.04	96.08	85.04	82.66	40.41
Predicted Cla. Acc.	23.94	88.40	80.31	74.55	45.70	96.88	85.89	83.47	42.51
Con. Acc.	14.19	42.98	88.47	80.60	98.74	98.43	91.74	90.02	82.83
MAE	0.52	0.24	0.49	0.84	2.34	0.80	0.85	0.81	2.50



(a) Brightness, Contrast, Saturation



(b) Hue

Figure 5: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different augmentation strength on color jittering operations under CIFAR-10 setup.

all other augmentations in SimCLR are kept default. In Figure 5, for each of these parameters, we plot the changes of the accuracy estimation error under the CIFAR-10 setup. Here we have similar observation that the default parameter values ($Brightness = Contrast = Saturation = 0.8, Hue = 0.2$) in SimCLR yield best performance.

Also, **temperature scaling** is an important factor during the training process of SimCLR. We study the temperature parameter τ on CIFAR-10. As Figure 6 shows, when using default temperature value $\tau = 0.07$, we can obtain best performance for both MAE and correlation coefficient.

Finally, we investigate how to assign appropriate **task weights** in the proposed multi-task training. Here we fix the classification weight to 1.0, and change different CL task weights: 1.0, 0.1, 0.01, 0.001. In Table 4, we report the linear correlation score and estimation error. The insight is that our framework is more likely to perform well when assigning small weight (< 0.01) for the CL task.

In summary, these empirical results demonstrate that when adopting CL frameworks, keeping default optimal settings is most likely to build strong linear correlation between the CL accuracy and classification accuracy, as well as obtain lowest accuracy estimation error on the final unlabeled test set.

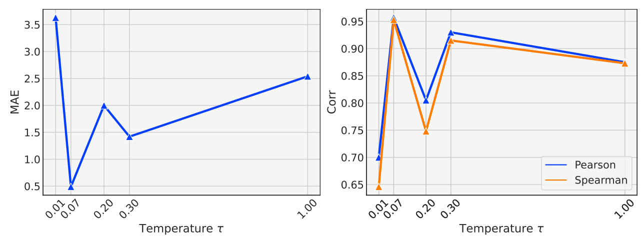


Figure 6: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different temperature scaling parameters under CIFAR-10 setup.

Table 4: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different contrastive learning task weight λ under CIFAR-10 setup.

CL task weight	1.0	0.1	0.01	0.001
r	0.953	0.946	0.872	0.953
ρ	0.957	0.933	0.860	0.957
MAE	0.71	2.87	0.91	0.49

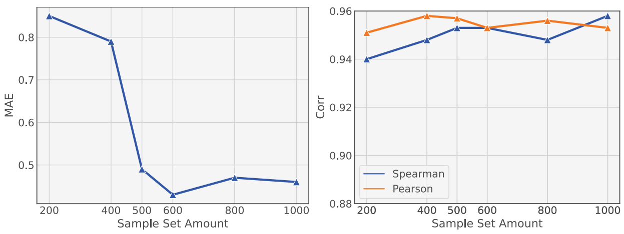
Table 5: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different CL data augmentation under CIFAR-10 setup.

CL Data Aug	SimCLR	MoCo-v1	MoCo-v2	BYOL
r	0.957	0.868	0.893	0.922
ρ	0.953	0.864	0.897	0.902
MAE	0.49	1.20	0.88	0.59

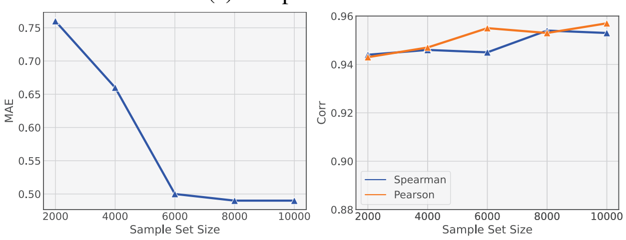
4.3.2 Different Training Settings

Different contrastive learning augmentation groups. In this paper, we adopt SimCLR, to study if other frameworks can fit in well, we change SimCLR to MoCo-v1 [24], MoCo-v2 [7], and BYOL [22]. From Table 5, we can observe that on CIFAR-10, the linear correlations are all strong across different CL frameworks ($r > 0.86$).

Different amounts and sizes of synthetic datasets. In this paper, we synthesize many sample sets to build a re-



(a) Sample set amount



(b) Sample set size

Figure 7: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different meta-set size and sample-set size under CIFAR-10 setup.

Table 6: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different CNN backbones under CIFAR-10 setup.

Backbone	ResNet-18	ResNet-34	VGG-11	VGG-19	DenseNet-40-12
r	0.853	0.854	0.816	0.863	0.957
ρ	0.858	0.853	0.787	0.849	0.953
MAE	0.92	0.63	1.17	0.77	0.49

gression model for accuracy estimation, so we study the influence of the amount of synthetic test sets and the size of each test set. Here, we refer to them as *sample set amount* and *sample set size* respectively. As Figure 7 shows, the linear correlation is quite robust and the estimation error is also robust when there are enough test sets.

Different backbones. In experimental setup, we use DenseNet-40-12 for CIFAR-10 setup in default. Here we change it to other model structures (ResNet-18, ResNet-34, VGG-11, VGG-19 [54]) to study if the choice of backbone significantly influences the performance. As Table 6 shows, our framework is robust against different backbones (strong linear correlation and precise accuracy estimation).

Different random seeds. To check if the experimental results are robust to the initial random state, we choose different random seeds for training (use 0 as default seed). As Table 7 shows, the performance of our framework is robust to randomness.

5. Method Interpretability

Regressor Robustness on transformed test sets. To further check the regressor robustness in the unobserved testing environment. We rigorously shift three natural datasets (ImageNet, Pascal and Caltech) with new transformations that do not overlap with the various transformations constructed when fitting regression curves. Specifically, we use Cutout [15], Shear, Equalize and ColorTemperature [11] to generate ImageNet-A/B, Pascal-A/B, Caltach-A/B. We note the following observations from Figure 8. First, the classifier accuracies will fluctuate after these test sets are shifted. Second, even in these unseen test sets cases undergoing new transformations, our method consistently achieves superior results. We conjecture that this phenomenon stems from contrastive learning performing well across a broad spectrum of underlying distributions.

Table 7: Pearson's correlation (r), Spearman's rank correlation (ρ) and MAE with different random seeds under CIFAR-10 setup.

Random Seed	0	21	42
r	0.957	0.892	0.869
ρ	0.953	0.894	0.845
MAE	0.49	0.65	0.51

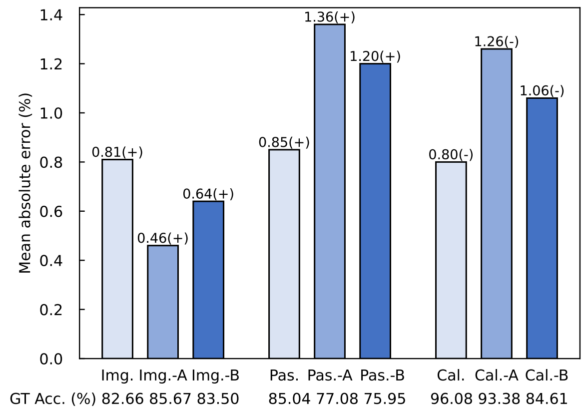
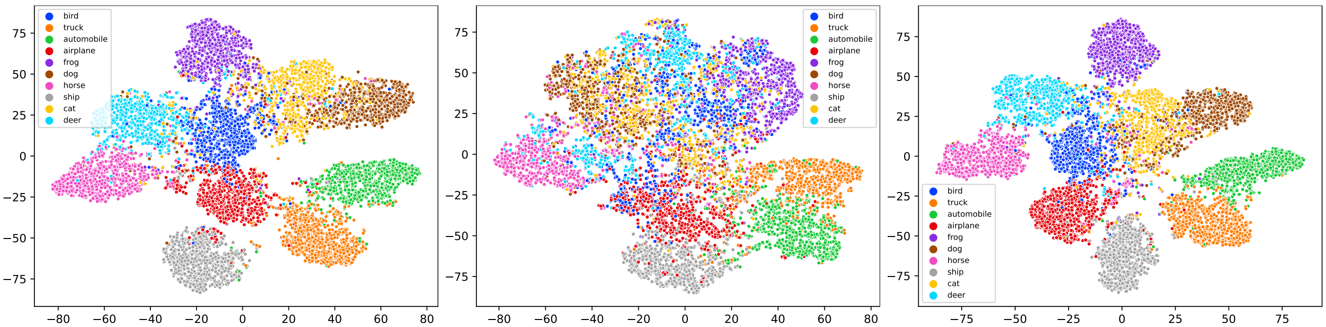


Figure 8: The MAE of linear regressor on transformed test sets (ImageNet, Pascal, and Caltech). The transformed datasets are denoted by "-A" and "-B" with new transformations such Cutout [15], Shear, Equalize and ColorTemperature [11]. (-) / (+) denotes the estimated accuracy is lower and higher than the ground-truth accuracy, respectively.

Underlying relationship between contrastive accuracy and classification accuracy. In common practice, contrastive learning is often used as a pre-training task, which has been proved effective for learning visual representations for downstream classification task. In this work, we adopt SimCLR in multi-task training. Here we compare the two ways in Table 8. We can observe that both of them can yield strong linear correlation and precise accuracy estimation, while the multi-task way is better. This finding further reveals the underlying relationship between contrastive accuracy and classification accuracy, regardless

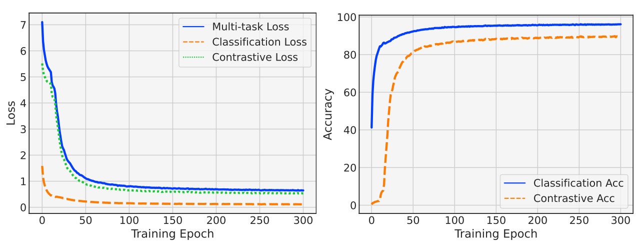


(a) Only Classification (b) Pre-train + Fine-tune (c) Multi-task

Figure 9: T-SNE visualization of the classification head features on CIFAR-10 validation set. Different colors correspond to different classes.

Table 8: Comparison between the pre-train+fine-tune (Pre+Fine) and multi-task training on CIFAR-10.

Paradigm	r	ρ	Acc_{con}	Acc_{cla}	$\widehat{Acc}_{cls}$	MAE
Pre+Fine	0.900	0.928	94.87	61.20	60.20	1.00
Multi-task	0.957	0.953	88.47	80.80	80.41	0.49



(a) Loss (b) Accuracy

Figure 10: Learning curve plot of accuracy and loss in training phase.

of the training way.

Multi-task training learns better features for AutoEval. Intuitively, strong linear correlation bewteen contrastive accuracy and classification accuracy can be built because they both learn class-wise visual representations. What kind of image features will be learned in multi-task training? In Figure 9 and 10, we plot some intermediate results during the training process. Notably, compared to pre-train + fine-tune, multi-task yields better feature clusters, which well corresponds to the results in Table 8. This further justifies the usage of contrastive learning in a multi-task paradigm is indeed feasible.

6. Conclusion

In this paper, we propose a novel framework CAME for estimating classifier accuracy on invisible test sets without ground-truth labels. We find that there is a strong linear correlation between contrastive accuracy and classification accuracy and give the theoretical analysis to support this discovery. Thus, our work indicates that it is feasible to estimate classifier accuracy by self-supervised contrastive accuracy using linear regression. Extensive experimental results show that our method achieves new SOTA results for AutoEval by outperforming previous works. In future, we will explore the feasibility of other self-supervised learning tasks in AutoEval, and extend CAME to other fields such as natural language processing and graph learning tasks.

Acknowledgement

This work is majorly supported by the Fundamental Research Funds for the Central Universities (No. 226-2022-00028), and in part by the NSFC Grants (No. 62206247). RP, HW and JZ also thank the sponsorship by the CAAI-Huawei Open Fund.

Limitation

Our method is grounded on an assumption that approximates the image distribution of unknown data environments via image transformations applied to the proposed synthetic sample sets. However, this might encounter certain intricate real-world cases where not be able to work. For example, there could be non-negligible samples in testing sets whose classes have never appeared in the training label space. Consequently, even though we will still predict an estimated accuracy, it is essentially unavailable. Despite the seemingly extreme situation, this issue could be well alleviated by employing out-of-distribution detection techniques [29, 46, 42, 44] to help detect and reject such samples. Furthermore, our method is not a plug-and-play solution for model evaluation due to the co-training strategy. Nonetheless, our method serves as a general technique with the potential for extension across various fields, supported by the widespread deployment of contrastive learning work.

References

[1] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

[2] Han Bao. On the surrogate gap between contrastive and supervised losses. *ICML*, 2022.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[4] Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992, 2021.

[5] Mayee Chen, Karan Goel, Nimit S Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Ré. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*, pages 1617–1629. PMLR, 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[9] Ching-Yao Chuang, Antonio Torralba, and Stefanie Jegelka. Estimating generalization under distribution shifts via domain-invariant representations. *arXiv preprint arXiv:2007.03511*, 2020.

[10] Ciprian A Corneanu, Sergio Escalera, and Aleix M Martinez. Computing the testing error without a testing set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2677–2685, 2020.

[11] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Weijian Deng, Stephen Gould, and Liang Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *International Conference on Machine Learning*, pages 2579–2589. PMLR, 2021.

[14] Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proc. CVPR*, 2021.

[15] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[16] DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[17] Hady Elsahar and Matthias Gallé. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2163–2173, 2019.

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[19] Alex Gain and Hava Siegelmann. Abstraction mechanisms predict generalization in deep neural networks. In *International Conference on Machine Learning*, pages 3357–3366. PMLR, 2020.

[20] Saurabh Garg, Sivaraman Balakrishnan, Zachary C Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234*, 2022.

[21] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[23] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1134–1144, 2021.

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

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

[26] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.

[27] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[28] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[29] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10951–10960, 2020.

[30] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[31] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. *Advances in Neural Information Processing Systems*, 34:3635–3649, 2021.

[32] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[33] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[34] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*, 2018.

[35] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*, 2021.

[36] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

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

[38] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[39] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[41] Jason D Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021.

[42] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[43] Mao Li, Kaiqi Jiang, and Xinhua Zhang. Implicit task-driven probability discrepancy measure for unsupervised domain adaptation. *Advances in neural information processing systems*, 34:25824–25838, 2021.

[44] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[46] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

[47] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

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

[49] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, pages 69–84. Springer, 2016.

[50] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[51] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[52] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[53] Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *ICML*, 2019.

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[55] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[56] Xiaoxiao Sun, Yunzhong Hou, Hongdong Li, and Liang Zheng. Label-free model evaluation with semi-structured dataset representations. *arXiv preprint arXiv:2112.00694*, 2021.

[57] Korawat Tanwisuth, Xinjie Fan, Huangjie Zheng, Shujian Zhang, Hao Zhang, Bo Chen, and Mingyuan Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:17194–17208, 2021.

[58] Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya Tolstikhin. Predicting neural network accuracy from weights. *arXiv preprint arXiv:2002.11448*, 2020.

[59] Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022.

[60] Yaodong Yu, Zitong Yang, Alexander Wei, Yi Ma, and Jacob Steinhardt. Predicting out-of-distribution error with the projection norm. *arXiv preprint arXiv:2202.05834*, 2022.

[61] Yi Zhang, Arushi Gupta, Nikunj Saunshi, and Sanjeev Arora. On predicting generalization using gans. *arXiv preprint arXiv:2111.14212*, 2021.

[62] Julian Zilly, Hannes Zilly, Oliver Richter, Roger Wattenhofer, Andrea Censi, and Emilio Frazzoli. The frechet distance of training and test distribution predicts the generalization gap. 2019.