i-MIX: A STRATEGY FOR REGULARIZING CONTRASTIVE REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Contrastive learning has shown state-of-the-art performance in self-supervised representation learning. However, a majority of the progress is limited to vision domains, given data augmentation techniques carefully designed based on domain knowledge. In this work, we propose i-*Mix*, a simple yet effective regularization strategy for improving contrastive representation learning schemes in both vision and non-vision domains. In particular, we cast them as training a non-parametric classifier by assigning a unique virtual class to each data in a batch. Then, i-Mix provides more augmented data by mixing given inputs in the data space and their virtual class labels in the label space. Our experiments demonstrate that i-Mix consistently improves the quality of learned representations across different domains, resulting in a performance gain for downstream tasks, which is often on par with end-to-end supervised learning. Also, we investigate under what conditions i-Mix is effective. The code will be released.

1 INTRODUCTION

Representation learning (Bengio et al., 2013) is a fundamental task in machine learning since the success of machine learning relies on the quality of representation. Self-supervised representation learning (SSL) has been successfully applied in several domains, including image recognition (He et al., 2020; Chen et al., 2020a), natural language processing (Mikolov et al., 2013; Devlin et al., 2018), robotics (Sermanet et al., 2018; Lee et al., 2019), speech recognition (Ravanelli et al., 2020), and video understanding (Korbar et al., 2018; Owens & Efros, 2018). Since no label is available in the unsupervised setting, pretext tasks are proposed to provide self-supervision: for example, context prediction (Doersch et al., 2015), inpainting (Pathak et al., 2016), and contrastive learning (Wu et al., 2018b; Hjelm et al., 2019; He et al., 2020; Chen et al., 2020a). SSL has also been used as an auxiliary task to improve the performance on the main task, such as generative model learning (Chen et al., 2019), semi-supervised learning (Zhai et al., 2019), and improving robustness and uncertainty (Hendrycks et al., 2019).

Recently, contrastive representation learning has gained increasing attention by showing state-of-the-art performance in SSL for large-scale image recognition (He et al., 2020; Chen et al., 2020a), which outperforms its supervised pre-training counterpart (He et al., 2016) on downstream tasks. However, while the concept of contrastive learning is applicable to any domains, the quality of learned representations rely on the domain-specific inductive bias: as anchors and positive samples are obtained from the same data instance, data augmentation introduces semantically meaningful variance for better generalization. To achieve a strong, yet semantically meaningful data augmentation, domain knowledge is required, e.g., color jittering in 2D images or structural information in video understanding. Hence, contrastive learning in different domains requires an effort to develop effective data augmentation methods. Furthermore, while recent works have focused on large-scale settings where millions of unlabeled data is available, it would not be a practical setting in real-world applications; for example, in lithography, acquiring data is very expensive in terms of both time and cost due to the complexity of manufacturing process (Lin et al., 2018; Sim et al., 2019).

Meanwhile, MixUp (Zhang et al., 2018) has shown to be a successful data augmentation method for supervised learning in various domains and tasks, including image classification (Zhang et al., 2018), generative model learning (Lucas et al., 2018), and natural language processing (Guo et al., 2019; Guo, 2020). In this paper, we explore the following natural, yet important question: is the idea of MixUp useful for unsupervised, self-supervised, or contrastive learning across different domains?

To this end, we propose *instance Mix (i-Mix)*, a data-driven augmentation method for contrastive representation learning effective across different domains. The key idea of i-Mix is to introduce virtual labels in a batch and mix instances and their corresponding virtual labels in the input and label spaces, respectively. We first introduce the general formulation of i-Mix, and then we show the applicability of i-Mix to the state-of-the-art contrastive learning methods, SimCLR (Chen et al., 2020a) and MoCo (He et al., 2020), and a variant without negative pairs, BYOL (Grill et al., 2020).

Through the experiments, we demonstrate the efficacy of i-Mix in a variety of settings: first, we show the effectiveness of i-Mix by evaluating the discriminative performance of learned representations in multiple domains. Specifically, we adapt i-Mix to the state-of-the-art contrastive representation learning methods, advancing the state-of-the-art performance across different domains, including image (Krizhevsky & Hinton, 2009; Deng et al., 2009), speech (Warden, 2018), and tabular (Asuncion & Newman, 2007) datasets. Then, we study i-Mix in various conditions, including when 1) the model and training dataset is small or large, 2) domain knowledge is limited, and 3) transfer learning.

Contribution. In summary, our contribution is three-fold:

- We propose i-Mix, a method for regularizing contrastive representation learning, motivated by MixUp (Zhang et al., 2018). We show how to apply i-Mix to state-of-the-art contrastive learning method and its variant (Chen et al., 2020a; He et al., 2020; Grill et al., 2020).
- Our results show that i-Mix consistently improves the performance of contrastive representation learning in both vision and non-vision domains. In particular, the discriminative performance of representations learned with i-Mix is on par with fully supervised learning on CIFAR-10/100 (Krizhevsky & Hinton, 2009), and Speech Commands (Warden, 2018).
- We study in what settings i-Mix is effective. We empirically observed that i-Mix significantly improves contrastive representation learning when 1) the training dataset size is small, or 2) domain knowledge for data augmentation is not enough.

2 Related Work

Self-supervised representation learning (SSL) aims at learning representations from unlabeled data by solving a pretext task that is derived from self-supervision. Early works on SSL proposed pretext tasks based on data reconstruction by autoencoding (Bengio et al., 2007), such as context prediction (Doersch et al., 2015) and inpainting (Pathak et al., 2016). Decoder-free SSL has made a huge progress in recent years. Exemplar CNN (Dosovitskiy et al., 2014) learns by classifying individual instances with data augmentation. SSL of visual representation, including colorization (Zhang et al., 2016), solving jigsaw puzzles (Noroozi & Favaro, 2016), counting the number of objects (Noroozi et al., 2017), rotation prediction (Gidaris et al., 2018), next pixel prediction (Oord et al., 2018; Hénaff et al., 2019), and combinations of them (Doersch & Zisserman, 2017; Kim et al., 2018; Noroozi et al., 2018) often leverages image-specific properties to design pretext tasks. Meanwhile, though deep clustering (Caron et al., 2018; 2019; Asano et al., 2020) is often distinguished from SSL, it also leverages clustering assignments as self-supervision for representation learning.

Contrastive representation learning has gained lots of attention for SSL (He et al., 2020; Chen et al., 2020a). As opposed to early works on exemplar CNN (Dosovitskiy et al., 2014; 2015), contrastive learning maximizes similarities of positive pairs while minimizes similarities of negative pairs instead of training an instance classifier. As the choice of negative pairs is crucial for the quality of learned representations, recent works carefully designed them. Memory-based approaches (Wu et al., 2018b; Hjelm et al., 2019; Bachman et al., 2019; Misra & van der Maaten, 2020; Tian et al., 2020a) maintain a memory bank of embedding vectors of instances to keep negative samples, where the memory is updated with embedding vectors extracted in previous batches. In addition, MoCo (He et al., 2020) showed that differentiating the model for anchors and positive/negative samples is effective, where the model for positive/negative samples is updated by the exponential moving average of the model for anchors. On the other hand, recent works (Ye et al., 2019; Misra & van der Maaten, 2020; Chen et al., 2020a; Tian et al., 2020a) showed that learning invariance to different views is important in contrastive representation learning. The views can be generated through data augmentation carefully designed based on domain knowledge (Chen et al., 2020a), splitting input channels (Tian et al., 2020a), or borrowing the idea of other pretext tasks, such as creating jigsaw puzzles or rotating inputs (Misra & van der Maaten, 2020). In particular, SimCLR (Chen et al., 2020a) showed that memory-free approaches with a large batch size and strong data augmentation has a comparable performance to memory-based approaches. InfoMin (Tian et al., 2020b) further

studied a way to generate good views for contrastive representation learning and achieved state-of-the-art performance by combining prior works. BYOL (Grill et al., 2020) is a variant of contrastive representation learning method without negative pairs, where the proposed pretext task aims at predicting latent representations of one view from another. While prior works have focused on SSL on large-scale visual recognition tasks, our work focuses on contrastive representation learning in both small- and large-scale settings in different domains.

Data augmentation is a technique to increase the diversity of data, especially when training data are not enough for generalization. Since the augmented data must be understood as the original data, data augmentation methods are carefully designed based on the domain knowledge about image (DeVries & Taylor, 2017b; Cubuk et al., 2019a;b; Zhong et al., 2020), speech (Amodei et al., 2016; Park et al., 2019), or natural language (Zhang et al., 2015; Wei & Zou, 2019).

Some works have studied data augmentation methods with less domain knowledge: DeVries & Taylor (2017a) proposed a domain-agnostic data augmentation method by first encoding the dataset and then applying augmentation in the feature space. MixUp (Zhang et al., 2018) is an effective data augmentation method in supervised learning, which performs vicinal risk minimization instead of empirical risk minimization, by linearly interpolating input data and their labels on the data and label spaces, respectively. On the other hand, MixUp has also shown its effectiveness in other tasks and non-vision domains, including generative adversarial networks (Lucas et al., 2018), improving robustness and uncertainty (Hendrycks et al., 2020), and sentence classification in natural language processing (Guo, 2020; Guo et al., 2019). Other variations have also been investigated by interpolating in the feature space (Verma et al., 2019) or leveraging domain knowledge (Yun et al., 2019). MixUp would not be applicable directly in some domains, e.g., linear interpolation of 3D point clouds does not maintain the form of original shapes, but its adaptation can be effective (Harris et al., 2020). Our proposed i-Mix is a kind of data augmentation method for better generalization in contrastive representation learning, resulting in better performances on downstream tasks.

3 Approach

In this section, we review MixUp (Zhang et al., 2018) in supervised learning and present i-Mix in contrastive learning (He et al., 2020; Chen et al., 2020a; Grill et al., 2020). Throughout this section, let $\mathcal{X}$ be a data space, $\mathbb{R}^D$ be a D-dimensional embedding space, and a model $f : \mathcal{X} \to \mathbb{R}^D$ be a mapping between them. For conciseness, $f_i = f(x_i)$ and $\tilde{f}_i = f(\tilde{x}_i)$ for $x_i, \tilde{x}_i \in \mathcal{X}$, and model parameters are omitted in loss functions.

3.1 MixUp in Supervised Learning

Suppose an one-hot label $y_i \in [0, 1]^C$ is assigned to a data x_i, where C is the number of classes. Let a linear classifier predicting the labels consists of weight vectors $\{w_1, \dots, w_C\}$, where $w_c \in \mathbb{R}^D$.[1] Then, the cross-entropy loss for supervised learning is defined as:

$$\ell_{\text{Sup}}(x_i, y_i) = -\sum_{c=1}^{C} y_{i,c} \log \frac{\exp(w_c^\top f_i)}{\sum_{k=1}^{C} \exp(w_k^\top f_i)}. \tag{1}$$

While the cross-entropy loss is widely used for supervised training of deep neural networks, there are several challenges of training with the cross-entropy loss, such as preventing overfitting or networks being overconfident. Several regularization techniques have been proposed to alleviate these issues, including label smoothing (Szegedy et al., 2016), adversarial training (Miyato et al., 2018), and confidence calibration (Lee et al., 2018).

MixUp (Zhang et al., 2018) is another simple and effective regularization method with a minimal computational overhead. It conducts a linear interpolation of two data instances in both input and label spaces and trains a model by minimizing the cross-entropy loss defined on the interpolated data and labels. Specifically, for two labeled data (x_i, y_i), (x_j, y_j), the MixUp loss is defined as follows:

$$\ell_{\text{Sup}}^{\text{MixUp}}\big((x_i, y_i), (x_j, y_j); \lambda\big) = \ell_{\text{Sup}}(\lambda x_i + (1-\lambda)x_j, \lambda y_i + (1-\lambda)y_j), \tag{2}$$

where $\lambda \sim \beta(\alpha, \alpha)$ is a mixing coefficient. MixUp is a vicinal risk minimization method (Chapelle et al., 2001) that augments data and their labels in a data-driven manner. Not only improving the generalization on the supervised task, it is also known to improve adversarial robustness (Zhang et al., 2018; Pang et al., 2019) and confidence calibration (Thulasidasan et al., 2019).

[1] We omit bias terms for presentation clarity.

Algorithm 1 Loss computation for i-Mix on N-pair contrastive learning in PyTorch-like style.

```
a, b = aug(x), aug(x) # different augmentations of x
lam = Beta(alpha, alpha).sample() # mixing coefficient
randidx = randperm(len(x))
a = lam * a + (1-lam) * a[randidx]
logits = matmul(normalize(model(a)), normalize(model(b)).T) / t
loss = lam * CrossEntropyLoss(logits, arange(len(x))) + \
       (1-lam) * CrossEntropyLoss(logits, randidx)
```

3.2 i-MIX IN CONTRASTIVE LEARNING

We introduce *instance mix (i-Mix)*, a data-driven data augmentation for contrastive representation learning to improve the generalization of learned representations. Intuitively, instead of mixing class labels, i-Mix interpolates their *virtual* labels, which indicates their identity in a batch.

Let $\mathcal{B}=\{(x_i,\tilde{x}_i)\}_{i=1}^N$ be a batch of data pairs, where N is the batch size, $x_i,\tilde{x}_i\in\mathcal{X}$ are two views (e.g., augmentations) of the same data. For each anchor x_i, we call $\tilde{x}_i$ and $\tilde{x}_{j\neq i}$ positive and negative samples, respectively.[2] Then, the model f learns to maximize similarities of positive pairs (instances from the same data) while minimize similarities of negative pairs (instances from different data) in the embedding space. The output of f is L2-normalized, which has shown to be effective (Wu et al., 2018a; He et al., 2020; Chen et al., 2020a). Let $v_i\in[0,1]^N$ be the virtual label of x_i and $\tilde{x}_i$ in a batch $\mathcal{B}$, where $v_{i,i}=1$ and $v_{i,j\neq i}=0$. For a general sample-wise contrastive loss with virtual labels $\ell(x_i,v_i)$, the i-Mix loss is defined as follows:

$$\ell^{i\text{-Mix}}\big((x_i,v_i),(x_j,v_j);\mathcal{B},\lambda\big)=\ell(\text{Mix}(x_i,x_j;\lambda),\lambda v_i+(1-\lambda)v_j;\mathcal{B}), \tag{3}$$

where $\lambda\sim\beta(\alpha,\alpha)$ is a mixing coefficient and Mix is a mixing operator, which can be adapted to depending on target domains: for example, $\text{MixUp}(x_i,x_j;\lambda)=\lambda x_i+(1-\lambda)x_j$ (Zhang et al., 2018) when data values are continuous, and $\text{CutMix}(x_i,x_j;\lambda)=M_\lambda\odot x_i+(1-M_\lambda)\odot x_j$ (Yun et al., 2019) when data values have a spatial correlation, where M_λ is a binary mask filtering out a region whose relative area is $(1-\lambda)$, and $\odot$ is an element-wise multiplication. In the following, we show how to apply i-Mix to contrastive representation learning methods. We use the MixUp operator for i-Mix formulations and experiments, unless otherwise specified.

SimCLR (Chen et al., 2020a) is a simple contrastive representation learning method without a memory bank, where each anchor has one positive sample and $(2N-2)$ negative samples. Let $x_{N+i}=\tilde{x}_i$ for conciseness. Then, the $(2N-1)$-way discrimination loss is written as follows:

$$\ell_{\text{SimCLR}}(x_i;\mathcal{B})=-\log\frac{\exp\big(s(f_i,f_{(N+i)\bmod 2N})/\tau\big)}{\sum_{k=1,k\neq i}^{2N}\exp\big(s(f_i,f_k)/\tau\big)}, \tag{4}$$

where τ is a temperature scaling parameter and $s(f,\tilde{f})=(f^\top\tilde{f})/\|f\|\|\tilde{f}\|$ is the inner product of two L2-normalized vectors. In this formulation, i-Mix is not directly applicable because virtual labels are defined differently for each anchor. To resolve this issue, we simplify the formulation of SimCLR by excluding anchors from negative samples. Then, with virtual labels, the $(N+1)$-way discrimination loss is written as follows:

$$\ell_{\text{N-pair}}(x_i,v_i;\mathcal{B})=-\sum_{n=1}^{N}v_{i,n}\log\frac{\exp\big(s(f_i,\tilde{f}_n)/\tau\big)}{\sum_{k=1}^{N}\exp\big(s(f_i,\tilde{f}_k)/\tau\big)}, \tag{5}$$

where we call it the **N-pair** contrastive loss, as the formulation is similar to the N-pair loss in the context of metric learning (Sohn, 2016).[3] For two data instances (x_i,v_i), (x_j,v_j) and a batch of data pairs $\mathcal{B}=\{(x_i,\tilde{x}_i)\}_{i=1}^N$, the i-Mix loss is defined as follows:

$$\ell_{\text{N-pair}}^{i\text{-Mix}}\big((x_i,v_i),(x_j,v_j);\mathcal{B},\lambda\big)=\ell_{\text{N-pair}}(\lambda x_i+(1-\lambda)x_j,\lambda v_i+(1-\lambda)v_j;\mathcal{B}). \tag{6}$$

Algorithm 1 provides the pseudocode of i-Mix on N-pair contrastive learning for one iteration.[4] We present the application of i-Mix to SimCLR in the supplementary material.

[2] Some literature (He et al., 2020; Chen et al., 2020a) refers to them as query and positive/negative keys.

[3] InfoNCE (Oord et al., 2018) is a similar loss inspired by the idea of noise-contrastive estimation (Gutmann & Hyvärinen, 2010).

[4] For losses linear with respect to labels (e.g., the cross-entropy loss), they are equivalent to $\lambda\ell(\lambda x_i+(1-\lambda)x_j,v_i)+(1-\lambda)\ell(\lambda x_i+(1-\lambda)x_j,v_j)$, i.e., optimization to the mixed label is equivalent to joint optimization to original labels. The proof for losses in contrastive learning methods is provided in the supplementary material.

Pair relations in contrastive loss. To make a sense of the contrastive loss as a representation learning objective, one needs to properly define a pair relation $\{(x_i, \tilde{x}_i)\}_{i=1}^N$. For contrastive representation learning, where semantic class labels are not provided, the pair relation would be defined in that 1) a positive pair, x_i and $\tilde{x}_i$, are different views of the same data and 2) a negative pair, x_i and $\tilde{x}_{j\neq i}$, are different data instances. For supervised representation learning, x_i and $\tilde{x}_i$ are two data instances from the same class, while x_i and $\tilde{x}_{j\neq i}$ are from different classes. Note that two augmented versions of the same data also belong to the same class, so they can also be considered as a positive pair. i-Mix is not limited to self-supervised contrastive representation learning, but it can also be used as a regularization method for supervised contrastive representation learning (Khosla et al., 2020) or deep metric learning (Sohn, 2016; Movshovitz-Attias et al., 2017).

MoCo (He et al., 2020). In contrastive representation learning, the number of negative samples affects the quality of learned representations (Arora et al., 2019). Because SimCLR mines negative samples in the current batch, having a large batch size is crucial, which often requires a lot of computational resources (Chen et al., 2020a). For efficient training, recent works have maintained a memory bank $\mathcal{M} = \{\mu_k\}_{k=1}^K$, which is a queue of previously extracted embedding vectors, where K is the size of the memory bank (Wu et al., 2018b; He et al., 2020; Tian et al., 2020a;b). In addition, MoCo introduces an exponential moving average (EMA) model to extract positive and negative embedding vectors, whose parameters are updated as follows: $\theta_{f^{\text{EMA}}} \leftarrow m\theta_{f^{\text{EMA}}} + (1-m)\theta_f$, where $m \in [0, 1)$ is a momentum coefficient and θ is model parameters. The loss is written as follows:

$$\ell_{\text{MoCo}}(x_i; \mathcal{B}, \mathcal{M}) = -\log \frac{\exp\left(s(f_i, \tilde{f}_i^{\text{EMA}})/\tau\right)}{\exp\left(s(f_i, \tilde{f}_i^{\text{EMA}})/\tau\right) + \sum_{k=1}^K \exp\left(s(f_i, \mu_k)/\tau\right)}. \tag{7}$$

The memory bank $\mathcal{M}$ is then updated with $\{\tilde{f}_i^{\text{EMA}}\}$ in the first-in first-out order. In this $(K+1)$-way discrimination loss, data pairs are independent to each other, such that i-Mix is not directly applicable because virtual labels are defined differently for each anchor. To overcome this issue, we include the positive samples of other anchors as negative samples, similar to the N-pair contrastive loss in Eq. (5). Let $\tilde{v}_i \in [0, 1]^{N+K}$ be a virtual label indicating the positive sample of each anchor, where $\tilde{v}_{i,i} = 1$ and $\tilde{v}_{i,j\neq i} = 0$. Then, the $(N+K)$-way discrimination loss is written as follows:

$$\ell_{\text{MoCo}}(x_i, \tilde{v}_i; \mathcal{B}, \mathcal{M}) = -\sum_{n=1}^N \tilde{v}_{i,n} \log \frac{\exp\left(s(f_i, \tilde{f}_n^{\text{EMA}})/\tau\right)}{\sum_{k=1}^N \exp\left(s(f_i, \tilde{f}_k^{\text{EMA}})/\tau\right) + \sum_{k=1}^K \exp\left(s(f_i, \mu_k)/\tau\right)}. \tag{8}$$

As virtual labels are bounded in the same set in this formulation, i-Mix is directly applicable: for two data $(x_i, \tilde{v}_i)$, $(x_j, \tilde{v}_j)$ and a batch of data pairs $\mathcal{B} = \{(x_i, \tilde{x}_i)\}_{i=1}^N$ and the memory bank $\mathcal{M}$, the i-Mix loss is defined as follows:

$$\ell_{\text{MoCo}}^{i\text{-Mix}}\big((x_i, \tilde{v}_i), (x_j, \tilde{v}_j); \mathcal{B}, \mathcal{M}, \lambda\big) = \ell_{\text{MoCo}}(\lambda x_i + (1-\lambda)x_j, \lambda\tilde{v}_i + (1-\lambda)\tilde{v}_j; \mathcal{B}, \mathcal{M}). \tag{9}$$

BYOL. Different from contrastive learning methods, BYOL is a self-supervised representation learning method without contrasting negative pairs. For two views of the same data $x_i, \tilde{x}_i \in \mathcal{X}$, the model f learns to predict a view embedded with the EMA model $\tilde{f}_i^{\text{EMA}}$ from its embedding f_i. Specifically, an additional prediction layer g is introduced, such that the difference between $g(f_i)$ and $\tilde{f}_i^{\text{EMA}}$ is learned to be minimized. The BYOL loss is written as follows:

$$\ell_{\text{BYOL}}(x_i, \tilde{x}_i) = \sum_{n=1}^N \left\| g(f_i)/\|g(f_i)\| - \tilde{f}_i/\|\tilde{f}_i\| \right\|^2 = 2N - 2\sum_{n=1}^N s(g(f_i), \tilde{f}_i). \tag{10}$$

This formulation can be represented in the form of the general contrastive loss in Eq. (3), as the second view $\tilde{x}_i$ can be accessed from the batch $\mathcal{B}$ with its virtual label v_i. To derive i-Mix in BYOL, let $\tilde{F} = [\tilde{f}_1/\|\tilde{f}_1\|, \ldots, \tilde{f}_N/\|\tilde{f}_N\|] \in \mathbb{R}^{D\times N}$ be the collection of L2-normalized embedding vectors of the second views, such that $\tilde{f}_i/\|\tilde{f}_i\| = \tilde{F}v_i$.

$$\ell_{\text{BYOL}}(x_i, v_i; \mathcal{B}) = \sum_{n=1}^N \left\| g(f_i)/\|g(f_i)\| - \tilde{F}v_i \right\|^2 = 2N - 2\sum_{n=1}^N s(g(f_i), \tilde{F}v_i). \tag{11}$$

For two data instances (x_i, v_i), (x_j, v_j) and a batch of data pairs $\mathcal{B} = \{(x_i, \tilde{x}_i)\}_{i=1}^N$, the i-Mix loss is defined as follows:

$$\ell_{\text{BYOL}}^{i\text{-Mix}}\big((x_i, v_i), (x_j, v_j); \mathcal{B}, \lambda\big) = \ell_{\text{BYOL}}(\lambda x_i + (1-\lambda)x_j, \lambda v_i + (1-\lambda)v_j; \mathcal{B}). \tag{12}$$

3.3 INPUTMIX

The contribution of data augmentation methods to the quality of learned representations is crucial in contrastive representation learning. For the case when the domain knowledge about efficient data augmentation methods is limited, we propose to apply InputMix together with i-Mix, which mixes input data but not their labels. This method can be viewed as introducing structured noises driven by auxiliary data to the principal data with the largest mixing coefficient λ, and the label of the principal data is assigned to the mixed data.

4 EXPERIMENTS

In this section, we demonstrate the effectiveness of i-Mix. In all experiments, we conduct contrastive representation learning on a pretext dataset and evaluate the quality of learned representations via supervised classification on a downstream dataset, and report the average performance of up to five runs. Specifically, in the first stage, a convolutional neural network (CNN) or multilayer perceptron (MLP) followed by the two-layer multilayer perceptron (MLP) projection head is trained on an unlabeled pretext dataset. Then, the projection head is replaced with a linear classifier and only the linear classifier is trained on a labeled downstream dataset. Except for transfer learning, datasets for the pretext and downstream tasks are the same. For i-Mix, we sample a mixing coefficient $\lambda \sim \beta(\alpha, \alpha)$ for each data, where $\alpha = 1$ for image and speech datasets and $\alpha = 2$ for tabular datasets. Additional details for the experimental settings can be found in the supplementary material.

4.1 EXPERIMENTAL SETUP

Compared methods. As baselines, we consider 1) N-pair contrastive learning as a memory-free contrastive learning method,[5] 2) MoCo (He et al., 2020; Chen et al., 2020b)[6] as a memory-based contrastive learning method, and 3) BYOL (Grill et al., 2020) as a variant of contrastive learning method. Then, we apply i-Mix to the methods above and compare their performances. To show the effectiveness of i-Mix in different domains, we evaluate the methods on 1) CIFAR-10 and 100 (Krizhevsky & Hinton, 2009) and ImageNet (Deng et al., 2009) as image datasets, 2) Speech Commands (Warden, 2018) as a speech dataset, and 3) Forest Cover Type (CovType) and Higgs Boson (Higgs) from UCI repository (Asuncion & Newman, 2007) as tabular datasets.

Image. CIFAR-10 and 100 consist of 50k training and 10k test images, and ImageNet has 1.3M training and 50k validation images, where we use them for evaluation. We apply a set of data augmentation methods randomly in sequence including random resized cropping, horizontal flipping, color jittering, gray scaling, and Gaussian blurring for ImageNet, which has shown to be effective in contrastive visual representation learning (Chen et al., 2020a;b). We use ResNet-50 (He et al., 2016) as a backbone network. Models are trained with a batch size of 256 (i.e., 512 augmented data)[7] for up to 4000 epochs on CIFAR-10 and 100, and with a batch size of 512 for 800 epochs on ImageNet. For ImageNet, we use the CutMix (Yun et al., 2019) version of i-Mix.

Speech. The Speech Commands dataset contains 51k training, 7k validation, and 7k test data in 12 classes. We apply a set of data augmentation methods randomly in sequence: changing amplitude, speed, and pitch in time domain, stretching, time shifting, and adding background noise in frequency domain. Data is then transformed to a 32×32 mel spectogram. We use the same architecture with image experiments. Models are trained with a batch size of 256 for 500 epochs.

Tabular. CovType contains 15k training and validation, and 566k test data in 7 classes, and Higgs contains 10.5M training and 0.5M test data for binary classification. Since the domain knowledge for data augmentation on tabular data is limited, only a masking noise with the probability 0.2 is considered as a data augmentation method. We use a 5-layer MLP with batch normalization (Ioffe & Szegedy, 2015) as a backbone network. Models are trained with a batch size of 512 for 1000 epochs.

4.2 i-MIX IN DIFFERENT DOMAINS

Table 1 shows the wide applicability of i-Mix to the state-of-the-art contrastive representation learning methods in different domains. i-Mix results in consistent improvements on the classification accuracy, e.g., up to 6.1% when i-Mix is applied to MoCo on CIFAR-100. Interestingly, we observe that linear

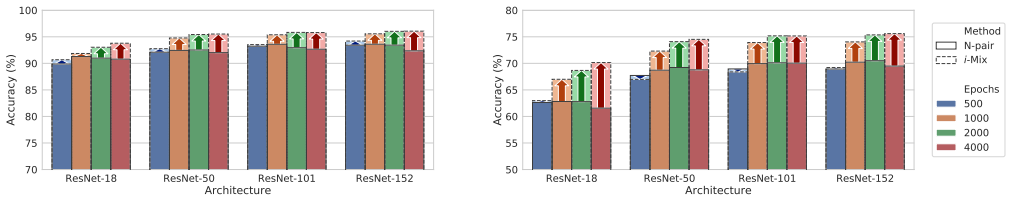
[5] N-pair contrastive learning results in no worse performance than SimCLR in our experiments, as shown in the supplementary material.

[6] We follow the settings in MoCo v2 (Chen et al., 2020b).

[7] A larger batch size does not affect performances on downstream tasks significantly on CIFAR datasets.

Domain	Dataset	N-pair	+ i-Mix	MoCo	+ i-Mix	BYOL	+ i-Mix
Image	CIFAR-10	92.5	**95.5**	93.3	**95.9**	94.1	**96.1**
	CIFAR-100	69.2	**74.5**	71.7	**77.8**	73.3	**78.9**
Speech	Commands	83.7	**91.1**	96.0	**98.2**	96.4	**97.9**
Tabular	CovType	65.8	**69.5**	66.4	**69.1**	65.9	**69.5**

Table 1: Comparison of contrastive representation learning methods and i-Mix in different domains.



(a) CIFAR-10 (b) CIFAR-100

Figure 2: Comparison of performance gains by applying i-Mix to N-pair contrastive learning with different model sizes and number of epochs on CIFAR-10 and 100.

Domain	Dataset	MoCo	+ i-Mix
Image	ImageNet	70.9	**71.3**

Domain	Dataset	N-pair	+ i-Mix
Tabular	Higgs (100k)	73.3	**73.5**
	Higgs (10M)	**76.3**	75.4

Table 2: Comparison of contrastive learning and i-Mix on large-scale datasets.

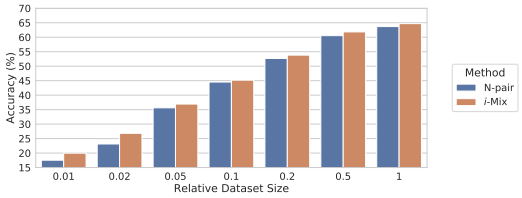


Figure 3: Comparison of MoCo and i-Mix trained on the different size of ImageNet.

classifiers trained on top of representations learned with i-Mix and then frozen while training on downstream tasks is often on par or even better than end-to-end supervised learning from random initialization: e.g., i-Mix vs. end-to-end supervised learning performance is 96.1% vs. 95.5% on CIFAR-10, 78.9% for both on CIFAR-100, and 98.2% vs. 98.0% on Speech Commands.

4.3 Scalability of i-Mix

A better regularization method often benefits from longer training of deeper models, and may as well improve the performance when trained on a smaller dataset. To investigate the regularization effect of i-Mix, we make a comparison between N-pair contrastive learning and i-Mix by training with different model sizes and number of training epochs on the pretext task. First, we test the efficacy of i-Mix with respect to different model sizes and training epochs. We train ResNet-18, 50, 101, and 152 models with the varying number of training epochs from 500 to 4000. Figure 2 shows the performance of N-pair contrastive learning (solid box) and i-Mix (dashed box). When models are trained for 500 epochs, the performance of i-Mix is on par with its baseline. However, i-Mix improves the performance by significant margins when trained longer. This is because i-Mix introduces more variance to the pretext dataset, such that it requires more update to learn them. In addition, it also benefits from deeper models, achieving 96.1% on CIFAR-10 and 75.6% on CIFAR-100 using ResNet-152 after 4000 epochs of training. On the other hand, the models trained without i-Mix starts to show overfitting to the pretext task when trained longer than 1000 epochs. The trend clearly shows that i-Mix results in better representations via improved regularization.

Next, we investigate the effect of the size of pretext datasets. Table 2 shows the effect of i-Mix applied to contrastive representation learning methods on large-scale datasets.[8] Different from the trend in Table 1, performance gains become marginal or even worse when the number of training data is large, similar to MixUp in supervised learning on large-scale datasets (Zhang et al., 2018). This may be due to the fact that i-Mix is a data augmentation method, such that it is not effective

[8] "large-scale" stands for a large number of data in our context.

Aug	CIFAR-10		CIFAR-100		Speech Commands		CovType		Higgs (100k)		Higgs (10M)	
	N-pair	+ i-Mix	N-pair	+ i-Mix	N-pair	+ i-Mix	N-pair	+ i-Mix	N-pair	+ i-Mix	N-pair	+ i-Mix
-	17.0	**79.7**	3.0	**50.7**	62.4	**72.7**	41.6	**68.5**	58.8	**72.6**	56.0	**74.6**
✓	92.5	**95.5**	69.2	**74.5**	83.7	**91.1**	65.8	**69.5**	73.3	**73.5**	**76.3**	75.4

Table 3: Comparison of contrastive learning and i-Mix with and without data augmentation.

Pretext	CIFAR-10		CIFAR-100	
Downstream	N-pair	+ i-Mix	N-pair	+ i-Mix
CIFAR-10	92.5	**95.5**	84.9	**86.9**
CIFAR-100	61.8	**65.1**	69.2	**74.5**

(a) CIFAR-10 and 100 as the pretext dataset

VOC Object Detection	ImageNet	
	MoCo	+ i-Mix
AP	57.3	**57.5**
AP_{50}	82.5	**82.7**
AP_{75}	63.8	**64.2**

(b) ImageNet as the pretext dataset

Table 4: Comparison of contrastive learning and i-Mix in transfer learning.

when the given training data are enough to learn the true data distribution. To verify this, we compare representations learned with different pretext dataset sizes from 1% to 100% of the ImageNet training data in Figure 3. For quick evaluation, only 10% of the ImageNet training data are used as the downstream training dataset. We can observe that the performance gain by i-Mix is significant when the size of the pretext dataset is small, but it becomes marginal as the size increases, i.e., i-Mix may over-regularize the network training.

4.4 Contrastive Learning without Domain-Specific Data Augmentation

Data augmentation plays a key role in contrastive representation learning, and therefore it raises a question when applying it to domains with a limited or no knowledge of such augmentations. In this section, we study the effectiveness of i-Mix as a data-driven data augmentation strategy for contrastive representation learning, which can be adapted to different domains. Table 3 shows the performance of N-pair contrastive learning and i-Mix with and without data augmentation methods. We observe significant improvements by i-Mix when other data augmentation methods are not applied. For example, compared to the accuracy of 92.5% when data augmentation methods are applied, contrastive learning achieves only 17.0% when trained without any data augmentation. This suggests that data augmentation is an essential part for the success of contrastive representation learning (Chen et al., 2020a). However, i-Mix is able to learn meaningful representations without other data augmentation methods and achieves close to the accuracy of 80% on CIFAR-10.

4.5 i-Mix with Contrastive Learning Methods and Transferability

In this section, we show the improved transferability of the representations learned with i-Mix. The results are provided in Table 4. First, we train linear classifiers with downstream datasets different from the pretext dataset used to train backbone networks and evaluate their performance, e.g., CIFAR-10 as pretext and CIFAR-100 as downstream datasets or vice versa. We observe consistent performance gains when learned representations from one dataset are evaluated on classification tasks on another dataset. Next, we transfer representations trained on ImageNet to the PASCAL VOC object detection task (Everingham et al., 2010). We follow the settings in prior works (He et al., 2020; Chen et al., 2020b): the parameters of the pre-trained ResNet-50 is transferred to a Faster R-CNN detector with the ResNet50-C4 backbone (Ren et al., 2015), and it is fine-tuned end-to-end on the VOC 07+12 trainval dataset and evaluated on the VOC 07 test dataset; note that the backbone network is fine-tuned here, different from other downstream tasks. As metrics, we use the average precision (AP) averaged over IoU thresholds between 50% to 95% at a step of 5%, and AP_{50} and AP_{75}, which are AP values when IoU threshold is 50% and 75%, respectively. Similar Table 2, we observe small but consistent performance gains in all metrics. Those results confirm that i-Mix improves the quality of the learned representations, such that performances on downstream tasks are improved.

5 Conclusion

We propose i-Mix, a data-driven data augmentation method for contrastive representation learning. The key idea of i-Mix is to introduce a virtual label to each data instance, and mix both inputs and the corresponding virtual labels. We show that i-Mix is applicable to the state-of-the-art contrastive learning methods, which consistently improves the performance in a variety of settings.

REFERENCES

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML*, 2016.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, 2019.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, 2007.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *PAMI*, 35(8):1798–1828, 2013.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *NIPS*, 2001.

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *CVPR*, 2019.

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

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

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019a.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017a.

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

Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *PAMI*, 38(9):1734–1747, 2015.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Hongyu Guo. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *AAAI*, 2020.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, and Adam Prügel-Bennett Jonathon Hare. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *WACV*, 2018.

Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *ICRA*, 2019.

Yibo Lin, Yuki Watanabe, Taiki Kimura, Tetsuaki Matsunawa, Shigeki Nojima, Meng Li, and David Z Pan. Data efficient lithography modeling with residual neural networks and transfer learning. In *Proceedings of the 2018 International Symposium on Physical Design*, pp. 82–89, 2018.

Thomas Lucas, Corentin Tallec, Jakob Verbeek, and Yann Ollivier. Mixed batches and symmetric discriminators for gan training. In *ICML*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *PAMI*, 41(8):1979–1993, 2018.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.

Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018.

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

Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.

Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. *arXiv preprint arXiv:1909.11515*, 2019.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. Multi-task self-supervised learning for robust speech recognition. In *ICASSP*, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018.

Woojoo Sim, Kibok Lee, Dingdong Yang, Jaeseung Jeong, Ji-Suk Hong, Sooryong Lee, and Honglak Lee. Automatic correction of lithography hotspots with a deep generative model. In *Optical Microlithography XXXII*, volume 10961, pp. 1096105. International Society for Optics and Photonics, 2019.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *NeurIPS*, 2019.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020b.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 2019.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Jason W Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Zhengyou Zhang, and Yun Fu. Incremental classifier learning with generative adversarial networks. *arXiv preprint arXiv:1802.00853*, 2018a.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018b.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, 2015.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.