IMPROVING DISCRIMINATIVE VISUAL REPRESENTATION LEARNING VIA AUTOMATIC MIXUP

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixup, a convex interpolation technique for data augmentation, has achieved great success in deep neural networks. However, the community usually confines it to supervised scenarios or applies it as a predefined augmentation strategy in various fields, grossly underestimating its capacity for modeling relationships between two classes or instances. In this paper, we decompose mixup into two sub-tasks of mixup generation and classification and formulate it for discriminative representations as class- and instance-level mixup. We first analyze and summarize the properties of instance-level mixup as local smoothness and global discrimination. Then, we improve mixup generation with these properties from two aspects: we enhance modeling non-linear mixup relationships between two samples and discuss learning objectives for mixup generation. Eventually, we propose a general mixup training method called AMix to improve discriminative representations on various scenarios. Extensive experiments on supervised and self-supervised scenarios show that AMix consistently outperforms leading methods by a large margin.

1 INTRODUCTION

One of the fundamental problems in machine learning is to learn a good low-dimensional representation efficiently that captures the intrinsic structures of data and facilitates downstream tasks such as classification or clustering (Bengio et al., 2013). Supervised learning (SL) and self-supervised learning (SSL) are two commonly-used settings for discriminative representation learning and have demonstrated their successes in several domains. Recently, SSL has shown more exciting achievements than SL on image recognition (He et al., 2016), natural language processing (Devlin et al., 2018), and video understanding (Korbar et al., 2018).

Since class information is invisible to SSL, pretext tasks are proposed as the supervision, such as context prediction (Doersch et al., 2015), inpainting (Pathak et al., 2016), and recently proposed contrastive learning (CL) (Grill et al., 2020; He et al., 2020). Among them, CL has shown state-of-the-art performance for learning discriminative representations on large-scale datasets. Based on specific domain knowledge, like anchors and positive samples constructed from the same data, CL optimizes a neighborhood system of each instance by maximizing the similarity between positive pairs while minimizing that of negative ones. Despite their effectiveness, there is a serious limitation of these two approaches: *SL and SSL mainly focus on aligning each sample to a discrete class centroid or anchor and discriminating from other centroids or instances but without considering the transitional space between centroids.* For example, SL methods suffer an over-confidence problem (Thulasidasan et al., 2019); CL establishes the neighborhood system from a single instance and push-away all other instances even if they should be considered in the same cluster (Robinson et al., 2021).

MixUp (Zhang et al., 2017), proposed for SL, bridges the decision boundary between two class distributions by generating virtual samples via convex interpolation. There are three types of variants in terms of the way of mixed sample generation: (1) linear approaches (Yun et al., 2019; Qin et al., 2020; Harris et al., 2020) combine two samples by cutting or the same weighted pixel-wise mask; (2) saliency-based methods (Kim et al., 2020; 2021) significantly improve performance by generating mixed samples that maximizing saliency information; (3) end-to-end fashion, AutoMix (Liu et al., 2021) proposes a sub-network (Mix Block) to learn inter-class relationships online via feature maps. However, they are all limited to the class-level. Some recently proposed works try to combine mixup and CL (Lee et al., 2021; Shen et al., 2021), but they focus on the mixup classification problem, *i.e.*,

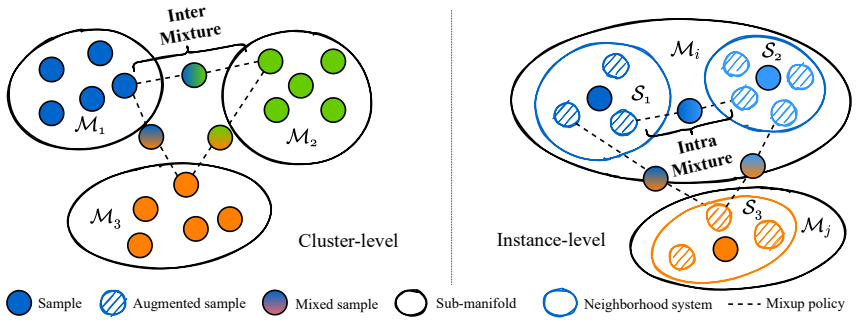

Figure 1: In cluster-level (supervised), the high-dimensional data is supported on its classes sub-manifold $\mathcal{M}_i$, while a neighborhood systems $\{\mathcal{S}_j\}$ is defined by their augmented views. We hope that the mixed samples can prompt the representation to be learned more discriminative.

modifying the contrastive loss to utilize linear mixup samples. In this paper, we discuss mixup from a general view in discriminative representation, including class- and instance-level mixup in SL and SSL settings. Compared to CL, which achieves the compactness of a neighborhood system for each instance by discriminating augmented positive and negative samples, mixup smoothly optimizes the relationship between two classes (or instances) by discriminating mixed samples from corresponding classes (or instances). Meanwhile, mixup is more flexible than CL since the mixed samples can be generated according to the data. Naturally, we decompose mixup into two sub-tasks of mixed sample generation and mixup classification, and **automaticly learn the smooth relationships in both class- and instance-level** to improve the discriminative representation learned by SL and SSL. Specifically, we propose and verify properties of the mixed sample as follows:

- *Local relationships*: The inter-class mixed data should smooth the gap between the two clusters while the intra-class one should be correlated to its neighborhood system.
- *Global relationships*: an inter-class mixed data should be discriminative to other clusters.

Based on the properties above, we propose AMix, a general class- and instance-level mixup training framework for discriminative representations. Taking AutoMix (Liu et al., 2021) as a baseline, we first improve Mix Block, a sub-network used to generate mixed samples, as Mix Block+, which strengthens the learning ability of non-linear relationship between two samples by content modeling and global attention with adaptive λ encoding. We then discuss the learning objective for mixup generation by analyzing local and global properties of mixup CE and infoNCE loss and propose binary cross-entropy (BCE) loss which delivers local relationships by regressing two related views. As for instance-level mixup, combining with BCE, we propose AMix-I using infoNCE loss while AMix-C using CE loss which adopts class information from pseudo labels. Extensive experiments in both SL and SSL tasks show that AMix has strong generalization and outperforms existing methods.

2 Problem Definition

Given a finite set of i.i.d samples, $X = [x_i]_{i=1}^n \in \mathbb{R}^{D\times n}$, each data $x_i \in \mathbb{R}^D$ is drawn from a mixture of, say K, distributions $\mathcal{D} = \{\mathcal{D}_c\}_{c=1}^K$. Our basic assumption for discriminative representations is that the each component distribution $\mathcal{D}_c$ has relatively low-dimensional intrinsic structures. Thus, we may assume that the distribution $\mathcal{D}_c$ is constrained on a sub-manifold, say $\mathcal{M}_c$ with dimension $d_c \ll D$, and the distribution $\mathcal{D}$ of X is consisted of these sub-manifolds, $\mathcal{M} = \cup_{c=1}^K \mathcal{M}_c$. Consider a discriminative problem like classification or clustering, we seek a low-dimensional representation $z_i \in \mathcal{M}$ of x_i by learning a continuous mapping modeled by a deep network, $f_\theta(x) : x \longmapsto z$ with the parameter $\theta \in \Theta$, which captures intrinsic structures of $\mathcal{M}$ and facilitates discriminative tasks.

2.1 Discriminative Representation Learning

We may divide most methods for the discriminative problem into two categories in terms of *class supervision is available or not*, and define their learning objective of f_θ respectively.

Parametric training with class supervision To ease the discriminative tasks in practical scenarios, *some supervised class information is available*, such as the class labels in supervised learning (SL)

or the cluster number K in clustering (C). Here, we assume that a one-hot label $y_i \in \mathbb{R}^K$ of each sample x_i can be somehow obtained, $Y = [y_1, y_2, ..., y_n] \in \mathbb{R}^{K \times n}$. We denote the labels generated or adjusted during training as pseudo labels (PL), while the fixed as ground truth labels (L). Notice that each component D_c (or sub-manifold $\mathcal{M}_c$) is considered *separated* in this scenario. Then, a *parametric* classifier can be learned to map the representation z_i of each sample to its class label y_i by predicting the probability of z_i being assigned to the c-th class using the softmax criterion,

$$P(c|z_i) = \frac{\exp(w_c^T z_i)}{\sum_{j=1}^{K} \exp(w_{j=1}^T z_i)}, \tag{1}$$

where w_c is a weight vector for the class c, and $w_c^T z_i$ measures how similar between z_i and the class c. The learning objective is to minimize the negative log-likelihood, say cross-entropy loss (CE).

Non-parametric training as instance discrimination Complementary to the above parametric settings, *non-parametric* approaches are usually adopted in unsupervised scenarios (label-free). Due to the *lack of class information*, an instance discriminative task can be designed instead of using the parametric class centroids in Eq.1. Based on an assumption of local compactness, the low-dimensional neighborhood systems $\mathcal{S}_i \in \mathbb{R}^{d_i}$ of the data x_i is invariant to a set of predefined augmentations $\mathcal{T}$, i.e., $x_i \in \mathcal{S}_i$ iff $\tau(x_i) \in \mathcal{S}_i$ for all $\tau \in \mathcal{T}$. We mainly discuss contrastive learning (CL) and taking MoCo (He et al., 2020) as an example. Consider a pair of augmented image (x_q, x_{k_+}) from the same instance $x \in \mathbb{R}^{C \times H \times W}$, the local compactness is introduced by alignment of the encoded representation pair (z_q, z_{k_+}) from $f_{\theta,q}$ and $f_{\theta,k}$, while constrained to the global uniformity by contrasting z_q to a dictionary of encoded keys from other images, $\{z_{k,j}\}_{j=1}^{K}$, where K denotes the length of the dictionary. It can be achieved by the popular non-parametric CL loss, called infoNCE:

$$\mathcal{L}_{q,k_+} = -\log \frac{\exp(z_q z_{k,+}/t)}{\sum_{j=1}^{K} \exp(z_q z_{k,j}/t)} \tag{2}$$

where t is a temperature hyper-parameter. Comparing to Eq.1, the infoNCE loss is a $(K+1)$-way non-parametric classification problem that attempts to classify z_q as $z_{k,+}$.

2.2 MIXUP TRAINING FOR DISCRIMINATIVE REPRESENTATION

Recall two sub-tasks in mixup training: (a) mixed data generation and (b) mixup classification. As for the sub-task (a), two mixup functions are defined, $h(\cdot)$ and $g(\cdot)$, to generate *mixed samples* and corresponding *mixed labels* with a mixing ratio λ sampled from $Beta(\alpha, \alpha)$ Given the mixed data, (b) defines a mixup training objective to optimize the inter-class discriminative relationships.

Mixup classification as the main task We first define two types of mixup classification objectives, *cluster-level* and *instance-level* mixup, for parametric and non-parametric training. As for parametric training, given two randomly selected data pairs, (x_i, y_i) and (x_j, y_j), the mixed data is generated as $x_{mix} = h(x_i, x_j, \lambda)$ and $y_{mix} = g(y_i, y_j, \lambda)$. The cluster-level mixup can be formally write as:

$$\min_{\theta,w} \ell_{CE}(f_\theta(h(x_i, x_j, \lambda)), g(y_i, y_j, \lambda)). \tag{3}$$

Notice that we fix the label mixup as the linear interpolation in our discussions, *i.e.*, $g(y_i, y_j, \lambda) = \lambda y_i + (1-\lambda) y_j$. Similarly to Eq.3, we can generate x_{mix} with a sample pair randomly selected from different instances (x_i, x_j) and generalize mixup to the infoNCE loss as instance-level mixup:

$$\mathcal{L}_{q_m,k_i,k_j}(\lambda) = -\lambda \log \frac{\exp(z_{q,m} z_{k,i}/t)}{\sum_{c=1}^{K} \exp(z_{q,m} z_{k,c}/t)} - (1-\lambda) \log \frac{\exp(z_{q,m} z_{k,j}/t)}{\sum_{c=1}^{K} \exp(z_{q,m} z_{k,c}/t)} \tag{4}$$

where $z_{q,m}$, $z_{k,i}$ and $z_{k,j}$ denote the representation of x_{mix} and corresponding instances.

Mixup generation as auxiliary task Different from the learning object on the *fixed* data X in Sec.2.1, the performance of classification is depending on the sub-task (a) because the mixup policies h and g reflect a certain relationship between the two sub-manifolds. Following AutoMix (Liu et al., 2021), we regard (b) as an auxiliary task to (a) and model $h(\cdot)$ as a sub-network $\mathcal{M}_\phi$ with another set of parameters $\phi \in \Phi$, called Mix Block (MB). $\mathcal{M}_\phi$ generates a pixel-wise mixup mask $s \in \mathbb{R}^{H \times W}$, where $s_{w,h} \in [0, 1]$. Since the mixup mask is directly related to representations of (x_i, x_j) and the mixing ratio λ, $\mathcal{M}_\phi$ takes l-th layer feature maps $z^l \in \mathbb{R}^{C_l \times H_l \times W_l}$ and λ as the

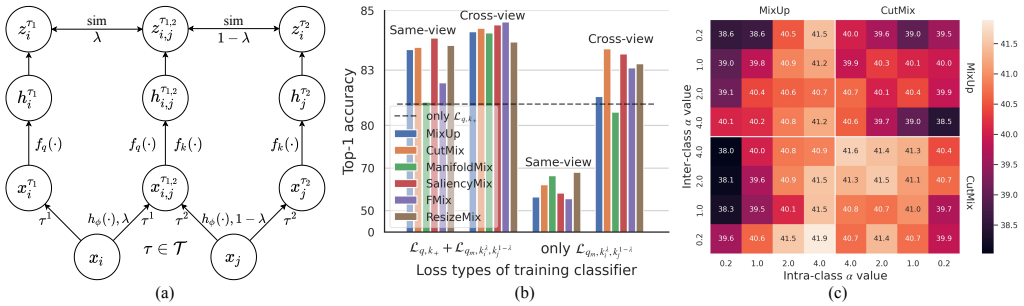

Figure 2: (a) Illustration of the cross-view pipeline for instance-level mixup training. (b) Analysis of mixup in CL, whether to use the cross-view pipeline and combining the mixup infoNCE loss. (c) A heat map that represents the effects of using MixUp and CutMix as the inter- and intra-class mixup.

input, $s_i = \mathcal{M}_\phi(z_i^l, z_j^l, \lambda)$ and $s_j = 1 - s_i$. $\mathcal{M}_\phi$ can be supervised by a mixup classification loss, denoted as $\mathcal{L}_\phi^{cls}$, and the loss for generated mask s_i, denoted as $\mathcal{L}_\phi^{mask}$. Both $\mathcal{L}_\theta$ and $\mathcal{L}_\phi$ can be optimized alternatively in a unified framework, *e.g.*, the momentum pipeline proposed in AutoMix.

3 AMix for Discriminative Representations

We first propose an efficient pipeline for instance-level mixup training and discuss the properties of two sub-tasks in instance-level mixup in Sec.3.1. Then, we discuss MB in two aspects: improving relationship modeling in Sec.3.2 and discussing the objective of mixup generation $\mathcal{L}_\phi$ in Sec.3.3. Moreover, we analyze mixed samples generated by Mix Block+ to reflect mixup properties.

3.1 Instance-level Mixup

Cross-view training pipeline We first analyze the instance-level mixup classification with fixed mixup samples. As shown in Fig.2 (b), we design an experiment of instance-level mixup methods with $\alpha = 1$ on STL-10 based on ResNet-18 (more experiment settings are detailed in Sec.4.2) to answer following two questions: (i) Properties of mixup classification. When only use Eq.4, degenerated solutions occur when use same augmented views for both mixup generation and classification, *i.e.*, $x_{mix} = h(x_i^{\tau_q}, x_j^{\tau_q}, \lambda)$, $z_{k,i} = f_{\theta,k}(x_i^{\tau_q})$, $z_{k,j} = f_{\theta,k}(x_j^{\tau_q})$, where τ_q denotes the augmentation for $f_{\theta,q}$. We hypothesize that it is caused by degenerated mixed samples which contain parts of the same view of two source images. Therefore, we propose a cross-view pipeline to address this issue, which is shown in Fig.2 (a), where $z_{k,i}$ and $z_{k,j}$ in Eq.4 are representations of $x_i^{\tau_k}$ and $x_j^{\tau_k}$. Notice that we use this framework in all subsequent experiments. (ii) Relationship between the CL task (Eq.2) and mixup (Eq.4). We find that combining both the Eq.2 and Eq.4 surpasses only using one of them, which indicate that mixup enables f_θ to learn relationship between each local neighborhood system.

Properties of mixup generation We then discuss the properties of instance-level mixup generation from two aspects: *inter-class* and *intra-class* mixup. In the case of class-level mixup, as we already know, it is important to consider class information to model the inter-class mixup relationship while there is no need to model the intra-class since each class sub-manifold is considered compact. At the instance level, we argue that the difference of inter- and intra-class relationships should be considered. To verify our assumption, we conducted an experiment that tries MixUp or CutMix as inter- and intra-class mixup policies on Tiny-ImageNet, as shown in Fig.2. This experiment uses different α values to approximate the intensity of MixUp and CutMix within or between classes, based on their global and local properties. Notice that the best result is achieved by using MixUp with large α as the intra-class mixup while CutMix with small α as the inter-class. Obviously, the class information is vital to mixup generation to distinguish the relationships between *global* and *local*, *inter-* and *intra-class*. Formally, following properties are expected to be held by the sample mixup function $h^*(\cdot)$, suppose the generated x_{mix}^* is the optimal from $x_i \in \mathcal{M}_a$ and $x_j \in \mathcal{M}_b$ by the given λ:

- *Local λ smoothness inter two classes*: If $\mathcal{M}_a \neq \mathcal{M}_b$, x_{mix}^* is generated as intermediate sample for filling the gap between $\mathcal{M}_a$ and $\mathcal{M}_b$.
- *Local compactness intra-class*: If $\mathcal{M}_a = \mathcal{M}_b$, then $x_{mix}^* \in \mathcal{M}_a$.
- *Global discriminative inter-class*: x_{mix}^* is all orthogonal to other sub-manifolds, $x_{mix}^* \neq \mathcal{M}_c$ for $c \neq a$ and $c \neq b$.

3.2 MIX BLOCK+

We improve MB in AutoMix from three aspects: (a) how to encode the ratio λ, (b) how to model the non-linear relationship between two samples, and (c) how to encode the prior knowledge of mixup.

Adaptive λ encoding According to the mixup labels, the predicted mask should be proportional to the randomly sampled λ. In original MB, λ is combined to the feature z_i by concatenating, $z^l_{i,\lambda} = concat(z^l_i, \lambda)$. However, this λ concatenating method only provides static additive impact to z^l_i, which might be ignored or cause trivial solutions during training. Therefore, we propose an *adaptive λ encoding*, $z^l_{i,\lambda} = (1 + \gamma\lambda)z^l_i$, where γ is a learnable scalar that constrained to $[0, 1]$. Symmetrically, we have $z^l_{j,1-\lambda} = (1 + \gamma(1 - \lambda))z^l_j$. Notice that γ is initialized to 0 during training.

Non-linear content modeling and global attention Intuitively, the mask s_i should take the content of both x_i and x_j into account. Based on $(z^l_{i,\lambda}, z^l_{j,1-\lambda})$, MB in AutoMix predicts s_i by two steps: Firstly, it models the content of x_i as a linear projection, $C_i = W_z z^l_{i,\lambda}$, where W_Z is a 1×1 convolution and $C_i \in \mathbb{R}^{H_l \times W_l}$ is a 2D tensor like the final mask s_i. Meanwhile, it computes the relationship between two samples as a cross-attention weight, $P_{i,j} = Softmax(\frac{W_P z^l_{i,\lambda} \otimes W_P z^l_{j,1-\lambda}}{N(z^l_{i,\lambda}, z^l_{j,1-\lambda})})$, where $N(z^l_{i,\lambda}, z^l_{j,1-\lambda})$ denotes a normalization factor and $\otimes$ is matrix multiplication. Then, it predicts the probability of each coordinate belonging to x_i as, $s_i = U(Sigmoid(P_{i,j} \otimes C_i))$, where U is an upsampling function. However, we find that MB in AutoMix is unstable in the early training period and sometimes trapped in trivial solutions, such as all coordinates on s_i predicted as a constant. We visualize C_i and $P_{i,j}$ and compare some trivial results with the non-trivial ones, as shown in Appendix. We find that the constant s_i is usually caused by a constant C_i. We hypothesize that trivial solutions happen earlier in C_i, which might be caused by the linear projection from C_l-dim to 1-dim. Thus, we design a *non-linear content modeling* module C_{NCL} that contains two 1×1 convolution layers with a batch normalization layer and a ReLU layer in between, as shown in Fig.3. To increase the robustness and randomness of mixup training, we add a Dropout layer with the dropout ratio of 0.1 to C_{NCL}. Moreover, we replace the original cross-attention with a self-attention mechanism by concatenating $(z^l_{i,\lambda}, z^l_{j,1-\lambda})$ as the input, $z^l_{concat} = concat(z^l_{i,\lambda}, z^l_{j,1-\lambda})$, which provides a *global attention* of cross samples and each sample itself. Formally, the improved MB can be written as,

$$s^\lambda_i = U\Big(Sigmoid\Big(Softmax(\frac{W_P z^l_{concat} \otimes W_P z^l_{concat}}{N(z^l_{concat}, z^l_{concat})}) \otimes C_{NCL}(z^l_{i,\lambda})\Big)\Big). \tag{5}$$

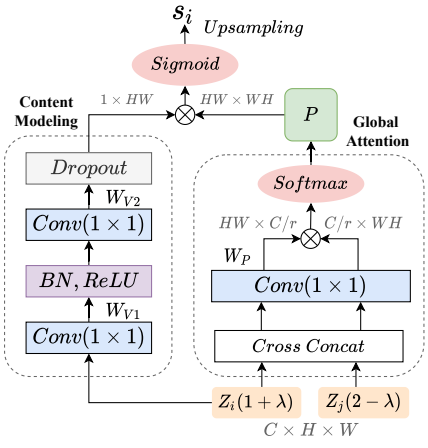

Figure 3: Illustration of Mix Block.

Loss function	STL-10 SL	STL-10 CL	Tiny ImageNet SL	Tiny ImageNet CL
Mixup	78.94	84.51	63.39	41.24
CE (L)	**82.14**	**85.95**	**68.56**	**43.11**
CE (PL), K=100	$81.47^{-0.67}$	$85.56^{-0.39}$	$67.12^{-1.44}$	$41.97^{-1.14}$
CE (PL), K=200	$81.65^{-0.49}$	$85.64^{-0.31}$	$67.39^{-1.17}$	$42.36^{-0.75}$
pBCE (L)	$81.96^{-0.18}$	$85.78^{-0.17}$	$67.84^{-0.72}$	$42.70^{-0.41}$
BCE	80.50	85.25	66.02	41.28
infoNCE	$80.67^{+0.17}$	$85.37^{+0.12}$	$66.36^{+0.34}$	$41.93^{+0.65}$
infoNCE (PL)	$81.04^{+0.54}$	$85.53^{+0.28}$	$66.79^{+0.77}$	$42.10^{+0.82}$
infoNCE (L)	$\mathbf{81.28}^{+0.78}$	$\mathbf{85.64}^{+0.39}$	$\mathbf{67.07}^{+1.05}$	$\mathbf{42.31}^{+1.03}$

Table 1: Analysis of the loss functions for training MixBlock. L and PL denote ground-truth labels and pseudo labels respectively. K is the number cluster assigned to the clustering algorithm (ODC). The upper part of the table demonstrates the analysis for parametric losses, while the lower part is for non-parametric losses.

Prior knowledge of mixup Taking the input space pixel-wise mixup mask $s_i \in \mathbb{R}^{H \times W}$ as an example, we recall some prior knowledge of mixup: (i) the mean of s_i should be linearly correlated with λ, (ii) mixed image should be discriminative to both classes (cluster-level mixup), (iii) the local patch of mixed images is smooth. We summarize these prior properties into two aspects: (a) adjusting s_i with λ, and (b) balancing the local smoothness and discrimination. Since MB in AutoMix has a mask loss to align the mean of s_i to λ, $\mathcal{L}_{mean} = \beta \max(|\lambda - \mu_i| - \epsilon, 0)$, where $\mu_i = \frac{1}{HW}\sum_{h,w} s_{i,h,w}$ is the mean and $\epsilon = 0.1$ as a margin, we improve the first aspect by a test time λ *adjusting* method. As we discussed in Sec.3.1, we regrad MixUp as mixing global (image

level) information of two samples, while patch-wise or pixel-wise mixup policies as mixing local discriminative regions. Assuming $\mu_i < \lambda$, we adjust each coordinate on s_i as $\hat{s}_i = \frac{\mu_i}{\lambda} s_i$, and $\hat{s}_j = 1 - \hat{s}_i$. As for the second aspect, we adopt a bilinear upsampling as U for smoother masks and propose a variance loss to encourage the sparsity of learned masks, $\mathcal{L}_{var} = \frac{1}{WH} \sum w, h(\mu_i - s_{w,h})^2$. Following the original MB, we summarize the mask loss $\mathcal{L}_\theta^{mask} = \beta(\mathcal{L}_{mean} + \mathcal{L}_{var})$.

3.3 How to train Mix Block

Based on the improved Mix Block+ (MB+) in Sec.3.2, we discuss the learning objective $\mathcal{L}_\theta^{cls}$ for MB in both class-level and instance-level mixup. Referring to the properties of an optimal mixup sample in Sec.3.1, the learning object $\mathcal{L}_\theta^{cls}$ demands balancing between the global and local relationships, like the mixup CE loss in Eq.3. Since the mixup CE has achieved good performance in optimizing both the mixup classification and generation, we regard it as a template and design an experiment of testing various losses in both SL and CL tasks, as shown in Tab.1, to answer the question: **how much does local structure influence the optimization of mix block?** We first decompose the mixup CE and mixup infoNCE in Eq.4 into the global and local parts. Both mixup CE and infoNCE provide a global view, *i.e.,* the denominator contains all the class centroids or negative instances. The local part in mixup CE can be defined as parametric binary cross-entropy loss (pBCE), which can be formulated as: $\mathcal{L}_{pBCE} = \lambda P(q|z_{mix}) + (1-\lambda) P(k|z_{mix})$. pBCE only considers the local relationship between two sub-manifold. Symmetrically, we have non-parametric binary cross-entropy (BCE) as:

$$\mathcal{L}_{BCE} = \lambda \frac{z_q z_{mix}}{z_q z_{mix} + z_k z_{mix}} + (1-\lambda) \frac{z_k z_{mix}}{z_q z_{mix} + z_k z_{mix}} \tag{6}$$

Notice that the main difference between CE and infoNCE is whether to adopt class centroids. Then we try to weaken the effect of centroids in mixup CE by using a pseudo label (PL) instead of labels (L), where PL is generated online by ODC (Zhan et al., 2020). Meanwhile, the class supervision to mixup infoNCE is added by PL and L for filtering out views that belong to the same sub-manifold, *i.e.,* approximating the centroid in SL, denoted as infoNCE (L) and infoNCE (PL).

As shown in the upper part of Tab.1, we find that destroying the structure of sub-manifolds has a negative impact on MB optimization. Surprisingly, using pBCE achieves better performance than CE (PL), which proves the importance of optimizing local relationships for mixup generation. As shown in the latter part of Tab.1 of the non-parametric loss, infoNCE is more accurate for global modeling of sub-manifold after introducing the label. In the end, the experimental results are consistent with our expectations. We conclude that modeling local relationships *determines* the performance of mixup generation while global discrimination serves as a *constraint*. Thus, We propose two versions of $\mathcal{L}_\phi^{cls}$ to train MB for instance-level mixup, $\mathcal{L}_{CE} + \mathcal{L}_{BCE}$ as the clustering version (AMix-C) using ODC to generate PL, $\mathcal{L}_{q_m,k_i,k_j} + \mathcal{L}_{BCE}$ as the instance version (AMix-I).

3.4 Discussion and analysis of AMix

From supervised (fine-grained to coarse-grained) to unsupervised (clustering and infoNCE) training, the richness of predefined cluster centroids decreases in order to learn more general representations. Since the learning objective for mixup generation in AMix influences global and local properties of mixup by cluster characteristics. We visualize mixed samples generated by AMix, as presented in Fig.4, to provide some insights for understanding the essence of automatic mixup training.

Class-level mixup in parametric learning From the left to right, the first four mixed samples are generated under the parametric loss. We see that as the granularity of clusters becomes larger, the discriminative information shifts from the attributes, the head of the bird and plane, to the whole object. Notice that the continuity of objects is influenced by the environmental textures (Wu et al., 2021). This is caused by the inaccurate pseudo-labeling provided from the clustering algorithm, the detailed analysis please refer to Tab.1. Although these different scenarios have their own characteristics, MB is still able to generate beneficial virtual samples in a specific task.

Instance-level mixup in non-parametric learning From the perspective of mixed features, we see that fine-grained SL and CL result in similar fine-grained features. Hence we hypothesize that both fine-grained SL and CL adopt fine-grained clusters than standard SL, $i.e.$, fine-grained SL further differentiating within a coarse class and CL discriminating each instance. In a sense, CL can be regarded as unsupervised finer-grained classification. Nevertheless, the former has fine-grained

Figure 4: Visualization of mixed samples generated by AMix in various learning scenarios. Note that λ=0.5 and Image B is set as the value. CL(C) and CL denote AMix-C and AMix-I separately.

class supervision while the latter does not. Therefore, the mixed sample of CL should have specific and attributed features. We further generated mixed samples by only using BCE or infoNCE loss in Appendix. We find that mixed samples generated with BCE contain continuous and certain parts of objects while samples generated with infoNCE prefer more specific and fragmented features. Thus, to take both the local and global characteristics into account, BCE is the *key* to the mixup generation unsupervised and the class/cluster information is also important for mixup to be discriminative.

4 Experiments

We first evaluate AMix in two popular discriminative representation learning, supervised learning (SL) in Sec.4.1 and self-supervised learning (SSL) in Sec.4.2, and then perform ablation studies in Sec.4.3. Six benchmarks are used for evaluating AMix: CIFAR100 (Krizhevsky et al., 2009), Tiny-ImageNet (Tiny) (Chrabaszcz et al., 2017), ImageNet-1k (IN-1k) (Russakovsky et al., 2015), STL-10 (Coates et al., 2011), CUB-200 (Wah et al., 2011), and FGVC-Aircraft (Maji et al., 2013). All experiments are conducted with PyTorch and Tesla V100 GPU and reported the *mean of 3 trials*.

Method	CIFAR-100		CUB-200		FGVC-Aircraft	
	R-18	RX-50	R-18	RX-50	R-18	RX-50
Vanilla	78.04	81.09	77.68	83.01	80.23	85.10
MixUp	79.12	82.10	78.39	84.58	79.52	85.18
CutMix	78.17	78.32	78.40	85.68	78.84	84.55
ManifoldMix	80.35	82.88	79.76	86.38	80.68	86.60
SaliencyMix	79.12	78.77	77.95	83.29	80.02	84.31
FMix	79.69	79.02	77.28	84.06	79.36	84.85
PuzzleMix	80.43	82.57	78.63	84.51	80.76	86.23
ResizeMix	80.01	79.73	78.50	84.77	78.10	84.08
AutoMix	82.04	83.64	79.87	86.56	81.37	86.69
AMix (ours)	**82.20**	**84.07**	**81.11**	**86.83**	**82.15**	**86.80**
Gain	+0.16	+0.37	+1.24	+0.27	+0.78	+0.11

Table 2: Top-1 accuracy (%) on CIFAR-100, CUB-200 and FGVC-Aircraft using R-18 and RX-50.

Method	Tiny ImageNet		ImageNet-1k			
	R-18	RX-50	R-18	R-34	R-50	R-101
Vanilla	61.68	65.04	71.83	75.29	77.35	78.91
MixUp	63.39	66.36	71.72	75.73	78.44	80.60
CutMix	64.40	66.47	70.03	75.16	78.69	80.59
ManifoldMix	62.76	67.30	71.73	75.44	78.21	80.64
SaliencyMix	64.95	66.55	70.21	75.01	78.46	80.45
FMix	62.28	65.08	70.30	75.12	78.51	80.20
PuzzleMix	65.63	66.92	71.64	75.84	78.87	80.67
Co-Mixup	66.29	67.31	71.73	75.89	78.92	80.69
ResizeMix	64.50	65.87	71.32	75.64	78.91	80.52
AutoMix	67.33	70.72	72.05	76.10	79.25	80.98
AMix (Ours)	**68.56**	**72.12**	**72.32**	**76.31**	**79.35**	**81.03**
Gain	+1.23	+1.40	+0.27	+0.21	+0.10	+0.05

Table 3: Top-1 accuracy (%) on Tiny ImageNet and ImageNet-1k using various backbones.

4.1 Evaluation on Supervised Image Classification

This subsection evaluates the performance gain of AMix for fine-grained and large-scale image classification tasks. To verify the generalizability of AMix for various scale network architectures, we adopt the residual neural networks ResNet (R) and ResNeXt (32x4d) (RX) (Xie et al., 2017) as backbone networks. Also, all experiments used the cosine scheduler (Loshchilov & Hutter, 2016) to adjust the learning rate with the SGD optimizer. For a fair comparison, grid search is performed for hyper-parameters of all mixup algorithms, $\alpha \in \{0.2, 0.5, 1.0, 2.0, 4.0\}$. The default setting is $\alpha = 1$ and other hyper-parameters follow the original paper. Following AutoMix, the momentum coefficient in AMix is gradually increased from $m = 0.999$ to 1 in a cosine curve and the feature layer $l = 3$ by default. The **median** of test performances in the last 10 training epochs is recorded for each trial.

Setups For a fair comparison, the basic training settings are set identically for all algorithms as following. We adopt *RandomFlip* and *RandomCrop* with 4 pixels reflect padding as basic data

augmentations for CIFAR-100 while *RandomFlip* and *RandomResizedCrop* for the rest datasets. For CIFAR-100, the SGD weight decay $wd = 0.0001$, momentum is 0.9, initial learning rate $lr = 0.1$, and train 800 epochs with the batch size of 100. For Tiny ImageNet, the total epochs are 400 with the initial learning rate $lr = 0.2$ and the batch size of 100. For ImageNet-1k, the total epochs are 300 with the initial learning rate $lr = 0.1$ and the batch size of 200. For fine-grained classification tasks, we use ImageNet pre-trained models provided by Pytorch as initialization, and set the initial learning rate $lr = 0.001$, the weight decay $wd = 0.0005$, the batch size is 16, and total epochs are 200.

Comparison and discussion On the small scale and fine-grained classification tasks, as shown in Tab.2, AMix consistently improves the classification performance over the previous best algorithm AutoMix on CIFAR-100, CUB-200, and FGVC-Aircraft by improving the design of Mix Block. Notice that AMix significantly improved the performance of CUB-200 and FGVC-Aircraft by 1.24% and 0.78% based on ResNet-18, and continued to expand its dominance on Tiny ImageNet by bringing 1.23% and 1.40% improvement on ResNet-18 and ResNeXt-50. Meanwhile, on the large-scale classification task, AMix also outperforms all existing methods. It is noteworthy to point out that AMix and AutoMix are the only two algorithms that surpass Vanilla based on ResNet-18.

CL baseline	mixup method	R18 400ep	R18 800ep	R50 400ep	R50 800ep
MoCo.V2	-	81.50	85.64	84.89	89.68
	MixUp	84.51	87.93	88.24	92.20
	ManifoldMix	84.17	87.70	88.06	91.65
	CutMix	84.28	87.60	87.51	90.81
MoCo.V2	SaliencyMix	84.33	87.27	87.35	90.77
	FMix	84.43	87.68	88.14	91.56
	ResizeMix	83.88	87.25	86.88	90.83
	AMix-I (Ours)	**85.37**	**88.58**	**88.87**	**92.41**
SwAV* (C)	-	81.10	85.56	84.35	88.79
MoCo.V2 (C)	Inter-Intra	84.89	87.85	88.33	92.24
MoCo.V2 (C)	PuzzleMix	84.98	88.07	88.40	91.98
MoCo.V2 (C)	**AMix-C (Ours)**	**85.56**	**88.63**	**88.91**	**92.45**

Table 4: Top-1 accuracy (%) of linear classification on STL-10 pre-training 400 and 800-epoch.

CL method	mixup method	Tiny ImageNet R18	Tiny ImageNet R50	ImageNet-1k R18	ImageNet-1k R50
MoCo.V2	-	38.29	42.08	52.85	67.66
MoCo.V2	MixUp	41.24	46.61	53.03	68.07
MoCo.V2	CutMix	41.62	46.24	52.98	68.28
MoCo.V2	FMix	41.09	46.30	53.10	68.42
MoCo.V2 (C)	PuzzleMix	41.86	46.72	53.28	68.48
MoCHi*	*input+latent*	41.78	46.55	53.12	68.01
i-Mix*	*input+latent*	41.61	46.57	53.09	68.10
UnMix†	*input+latent*	-	-	-	68.60
WBSIM†	*input*	-	-	-	68.40
MoCo.V2	**AMix-I (Ours)**	41.97	47.23	53.47	68.79
MoCo.V2 (C)	**AMix-C (Ours)**	**42.53**	**47.41**	**53.62**	**68.86**

Table 5: Top-1 accuracy (%) of linear classification on Tiny ImageNet and ImageNet-1k.

CL Method	mixup method	AP	AP_{50}	AP_{75}
MoCo.V2	-	56.9	82.2	63.4
MoCo.V2	Mixup	57.2	82.6	64.1
MoCo.V2	CutMix	57.3	82.7	64.0
MoCHi*	*input+latent*	57.1	82.7	64.1
i-Mix*	*input+latent*	57.5	82.7	64.2
UnMix†	*input+latent*	57.7	83.0	**64.3**
WBSIM†	*input*	57.4	82.8	64.2
MoCo.V2	**AMix-I (Ours)**	**57.8**	**83.1**	**64.3**

Table 6: Transferring to detection with Faster R-CNN on VOC07+12.

method	STL-10	Tiny
BCE	85.25	41.28
infoNCE	85.37	41.90
infoNCE+BCE	**85.44**	**41.97**
pBCE (C)	85.45	42.18
CE (C)	85.56	42.36
CE (C)+infoNCE	85.41	42.12
CE (C)+BCE	**85.60**	**42.53**

Table 7: Ablation of proposed losses on STL-10 and Tiny.

module	SL	CL
MixBlocl (baseline)	67.33	41.25
+Adaptive λ	67.85	41.62
+Non-linear content	68.08	41.96
+Global attention	68.34	42.10
+$\mathcal{L}_{mask}$	68.42	42.17
+Bilinear	**68.56**	42.28
+λ adjusting	68.37	**42.53**

Table 8: Ablation of proposed modules in MB+ on Tiny.

4.2 Evaluation on Self-supervised Learning

In this subsection, we evaluate AMix on SSL tasks on STL-10, Tiny ImageNet, and ImageNet-1k. Excepts SwAV (Caron et al., 2020), all comparing methods are based on MoCo.V2. We adopt all the training and hyper-parameter configurations from MoCo.V2 for pre-training on these datasets unless otherwise stated. We compared AMix in two dimensions in CL: (i) compare with other mixup variants, based on our proposed cross-view pipeline, and the predefined cluster information is given (denotes by C) or not, as shown in Fig.4. (ii) we then provide a longitudinal comparison with CL methods that utilize *input space* (*i.e.,* Mixup and CutMix) and *latent space* mixup strategies, including MoCHi (Kalantidis et al., 2020), i-Mix (Lee et al., 2021), Un-Mix (Shen et al., 2021) and WBSIM (Chu et al., 2020), as presented in Fig.5. Notice that * denotes results reproduced with the official source code and † denotes results reported in the original paper.

Linear Classification Following the linear classification protocol proposed in MoCo (He et al., 2020), we train a linear classifier on the top of frozen backbone features with the supervised train set. We train 100 epochs using SGD with a batch size of 256. The initialized learning rate is set to 0.1 for Tiny ImageNet and STL-10 while 30 for ImageNet, and decay by 0.1 at epoch 30 and 60. As shown in Tab.4, AMix outperforms all the linear mixup methods by a large margin and surpasses the saliency-based PuzzleMix when the cluster information is available. And AMix-I has both global and local properties through infoNCE and BCE losses. Meanwhile, Tab.5 demonstrates

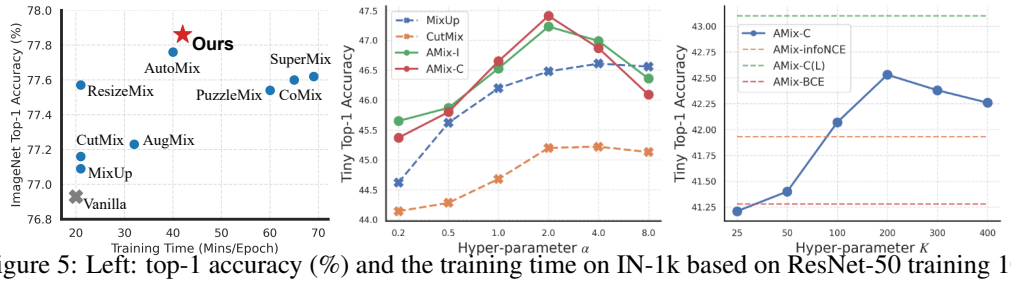

Figure 5: Left: top-1 accuracy (%) and the training time on IN-1k based on ResNet-50 training 100 epochs. Middle and Right: hyper-parameter ablation on α and the cluster number K on Tiny.

that AMix surpasses the other CL methods combined with the mixup, while AMix-C which using class information (PL) provided by ODC indeed helps MB to generate more accurate virtual samples.

Downstream Tasks Following the evaluation protocol in MoCo, we evaluate the transferable ability of the learned representation on the object detection task, in Fig.6. We benchmark the comparing methods on PASCAL VOC using Faster R-CNN (Ren et al., 2015) implemented in Detectron2 (Wu et al., 2019) with ResNet50-C4 backbone (i.e., using extracted features of the 4-th stage). We fine-tune the pre-trained models on the split of `trainval07+12` and evaluate on the VOC `test2007`.

4.3 Ablation Study

We conduct ablation studies in four aspects: (a) **MixBlock+**: The effectiveness of each proposed module to improve MB is verified in parametric (SL) and non-parametric (CL) tasks. As shown in Tab.8, the first three key modules strengthen the ability of mixup relationship modeling. Notice that the prior knowledge λ adjusting obtains a large margin gain in CL. (b) **Loss functions**: Based on this improved MB, we analyze the effectiveness of proposed losses. As shown in Tab.7, adding BCE loss to the mixup CE and infoNCE consistently improves the performance both in STL-10 and Tiny, which shows that BCE loss plays an important role in the mixed sample generation. (c) **Time complexity analysis**: As shown in the left of Fig.5, computational analysis is conducted on the SL task on IN-1k following the 100 epochs training protocol Wong et al. (2020). We can find that the overall efficiency of AMix is superior in contrast to other methods in this time-accuracy scatter plot. (d) **Hyper-parameter**: Fig.5 (middle and left) shows the ablation of the hyper-parameter α and the clustering number K in ODC for AMix-C. We empirically choose α=2.0 and $K = 200$ as default.

5 Related Work

Mixup in class level There are three types of class-level mixup: linear (Zhang et al., 2017; Yun et al., 2019; Qin et al., 2020; Verma et al., 2019; Hendrycks et al., 2019), saliency-based (Uddin et al., 2020; Kim et al., 2020; 2021), and end-to-end training mixup generation and classification (Liu et al., 2021; Dabouei et al., 2021). In this paper, AMix belongs to the third class, which also can model instance-level relationships.

Mixup in instance level A complementary method for better instance-level representation learning is to apply mixup on CL. Most approaches are limited to linear mixup meth attempts to use MixUp in the input space for self-supervised learning without a ground-truth label. In contrast, the developers of MoChi (Kalantidis et al., 2020) propose mixing the negative sample in the embedding space to increase the number of hard negatives but at the expense of classification accuracy. i-Mix (Lee et al., 2021) and BSIM (Chu et al., 2020) demonstrated how to regularize contrastive learning by mixing instances in the input/latent and virtual label spaces. We introduce AMix for SSL, which adaptively learns the instance relationship based on inter- and intra-cluster properties online.

6 Conclusion

Learning the intrinsic structure of data is a challenge in the deep learning community. However, as a method of modeling relationships between samples, mixup has been underestimated and is only used as a means of data augmentation. AMix provides a landscape for learning adaptive mixup policy at both class and instance levels, which is guided by the properties of mixup training. Specifically, the improved MB+ and cross-view training pipeline empower mixup beyond just "data augmentation".

REFERENCES

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Xiangxiang Chu, Xiaohang Zhan, and Xiaolin Wei. Beyond single instance multi-view unsupervised representation learning. *arXiv preprint arXiv:2011.13356*, 2020.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M Nasrabadi. Supermix: Supervising the mixing data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13794–13803, 2021.

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.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

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.

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*, 2(3):4, 2020.

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*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

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. *arXiv preprint arXiv:1912.02781*, 2019.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pp. 5275–5285. PMLR, 2020.

Jang-Hyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. *arXiv preprint arXiv:2102.03065*, 2021.

Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230*, 2018.

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

Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. I-mix: A domain-agnostic strategy for contrastive representation learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Zicheng Liu, Siyuan Li, Di Wu, Zhiyuan Chen, Lirong Wu, Jianzhu Guo, and Stan Z Li. Automix: Unveiling the power of mixup. *arXiv preprint arXiv:2103.13027*, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Jie Qin, Jiemin Fang, Qian Zhang, Wenyu Liu, Xingang Wang, and Xinggang Wang. Resizemix: Mixing data with preserved object information and true labels. *arXiv preprint arXiv:2012.11101*, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. In *International journal of computer vision*, pp. 211–252. Springer, 2015.

Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Un-mix: Rethinking image mixtures for unsupervised visual representation learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001*, 2019.

AFM Uddin, Mst Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. *arXiv preprint arXiv:2006.01791*, 2020.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. California Institute of Technology, 2011.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. 2020.

Di Wu, Siyuan Li, Zelin Zang, Kai Wang, Lei Shang, Baigui Sun, Hao Li, and Stan Z Li. Align yourself: Self-supervised pre-training for fine-grained recognition via saliency alignment. *arXiv preprint arXiv:2106.15788*, 2021.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.