AutoMix: Mixup Networks for Sample Interpolation via Cooperative Barycenter Learning

Jianchao Zhu[1][0000−0003−0742−2102], Liangliang Shi[2][0000−0001−7033−4207], Junchi Yan[2][0000−0001−9639−7679]⋆, and Hongyuan Zha[3]

[1] School of Software Engineering, East China Normal University
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[3] The Chinese University of Hong Kong, Shenzhen & Shenzhen Research Institute of Big Data
{1486013862, 851636947}@qq.com, yanjunchi@sjtu.edu.cn, zha@cc.gatech.edu

Abstract. This paper proposes new ways of sample mixing by thinking of the process as generation of barycenter in a metric space for data augmentation. First, we present an optimal-transport-based mixup technique to generate Wasserstein barycenter which works well on images with clean background and is empirically shown complementary to existing mixup methods. Then we generalize mixup to an AutoMix technique by using a learnable network to fit barycenter in a cooperative way between the classifier (a.k.a. discriminator) and generator networks. Experimental results on both multi-class and multi-label prediction tasks show the efficacy of our approach, which is also verified in the presence of unseen categories (open set) and noise.

Keywords: Image Mixing, Generative Model, Image Classification.

1 Introduction and Related Work

Deep networks have achieved unprecedented performance in various tasks, such as image classification [18], speech recognition [13], natural language processing [30], etc. The researches on improving network performance mainly focus on backbone design [26, 31], regularization [28] and data augmentation [36], etc.

Orthogonal to improvement on network backbone side, data augmentation [25, 36, 42] has been widely used for improving (neural network) model training. If a model is trained with a tiny dataset, over-fitting problem tends to occur. The model overly fits to the training data domain and results in poor generalization performance on data out of that domain. As a consequence, a large number of

⋆ Corresponding author is Junchi Yan. Part of the work was done when Junchi Yan was with Tencent AI Lab as visiting scholar. Hongyuan Zha is on leave from Georgia Institute of Technology. Work was partly supported by National Key Research and Development Program of China 2018AAA0100704, NSFC (61672231, U1609220), and SJTU Global Strategic Partnership Fund (2020 SJTU-CORNELL).

methods of data augmentation have been proposed to solve over-fitting. For images, typical operations include rotation, scaling and cropping, as well as adding noise and performing affine and other geometrical transformation. However, such augmentation methods inherently involve human knowledge to define the vicinity or neighborhood around each sample in the training dataset [8], and other potential techniques have not been fully explored.

There is emerging trend on data augmentation using the so-called mixup strategy [42]. It basically involves interpolating raw samples into a synthetic one with a new label, resulting in an augmented dataset. SamplePairing [15] randomly samples two images and mixes them with ratio $\lambda = 0.5$. The first image's label is used to train the network. Mixup [42] uses a randomly selected mixing ratio λ from β distribution to weigh the two images and their corresponding labels respectively. The Between-Class (BC) model [32] randomly selects two images from two distinctive classes and mixes them by a method in [33].

There are other methods do the mixing in Latent space (a.k.a. feature space). The MixFeat [38] and Manifold Mixup [35] seeks to interpolate the feature maps extracted by a convolutional network. While AdaMixUp [11] aims to determine the sample's ratio for interpolation via network learning. However, the interpolation of the methods mentioned above is still a predefined one, e.g., bilinear interpolation. The ACAI [7] uses an auto-encoder to learn mixed encoding of two images and a discriminator network to predict the mixing ratio from the decoded output of mixed encoding. These two parts are trained adversarially. Different from ACAI, the AMR [5] uses the discriminator to predict whether the mix is real or fake and a channel-wise binary mask $m \in \{0, 1\}$ sampled from Bernoulli distribution to randomly activate the feature maps.

In the following, we discuss some basic background related to our approach.

Barycenter Learning. Recently image mixing methods [11,15,32,33,35,38, 42] are emerging, which simply mix two images or two feature maps to achieve the purpose of data augmentation. In this paper, we view such methods as instances of barycenter learning. While such a perspective in fact has not been adopted in literature to our knowledge. We can find data points' mean value, i.e., barycenter via unsupervised learning as studied in optimal transport [1, 2]. As a variational problem, a weighted barycenter $\hat{x}$ can be defined for some points $\{x_i\}_{i=1}^n$ in a metric space $(X, d(\cdot, \cdot))$ with weights $\{\lambda_i\}_{i=1}^n$:

$$\min_x \sum_{i=1}^{n} \lambda_i d(x_i, x)^p, \tag{1}$$

here $d(\cdot, \cdot)$ is a distance function or more generally divergence, p is a constant, $\lambda_i > 0$ and $\sum_{i=1}^n \lambda_i = 1$. Different metric space according to $d(\cdot, \cdot)$ can lead to different results of barycenter. For instance, by setting $p = 2$, $X = \mathbb{R}^n$ and $d(x, y) = \|x - y\|_2$ in Euclidean space, we get its solution directly with $\hat{x} = \sum_{i=1}^n \lambda_i x_i$. Recent data augmentation methods by mixing two images (i.e., $n = 2$, $\hat{x} = \lambda x_1 + (1 - \lambda)x_2$) can be viewed as finding barycenters in this way.

Wasserstein Barycenter. Optimal transport [1,2] has gained its popularity in learning and graphics. Regarding pixels as samples and viewing the images

Table 1. Comparison of mixup methods for classification. Note that the mixing space refers to the space in which the relevant distance is calculated during the mixing phase. Therefore, Manifold Mixup and MixFeat are still bilinear interpolation methods that mix feature maps in Euclidean space. Our OptTransMix is based on optimal transport theory which works in Wasserstein space, and AutoMix directly adopts a network to learn the barycenter as interpolation for input samples in Latent space. Both of them are different from the recent mixup methods.

method	mixing way	mixing phase	mixing space	sample ratio	label ratio
OptTransMix (ours)	optimal transport	raw image	Wasserstein	fixed ratio	fixed ratio (1:0/0:1)
AutoMix (ours)	deep neural networks	feature map	Latent	random ratio	same as sample's
AMR [5]	combine by mask	feature map	Euclidean	random ratio	same as sample's
ACAI [7]	bilinear interpolation	feature map	Euclidean	generated by NN	same as sample's
AdaMixup [11]	bilinear interpolation	raw image	Euclidean	generated by NN	same as sample's
SamplePairing [15]	bilinear interpolation	raw image	Euclidean	half-half (1:1)	fixed ratio (1:0)
Between-Class [32]	bilinear interpolation	raw image	Euclidean	random ratio	same as sample's
Manifold Mixup [35]	bilinear interpolation	feature map	Euclidean	random ratio	same as sample's
MixFeat [38]	bilinear interpolation	feature map	Euclidean	random ratio	fixed ratio (1:0/0:1)
Mixup [42]	bilinear interpolation	raw image	Euclidean	random ratio	same as sample's

as histograms (i.e., probability measure), we can get the Wasserstein distance between images. So by replacing Euclidean space with Wasserstein space and setting $d(x, y) = W(x, y)$, $p = 1$, $n = 2$, we can find Wasserstein barycenters $\hat{x} = \arg\min_x [\lambda W(x_1, x) + (1 - \lambda)W(x_2, x)]$, in which the Wasserstein distance between two images $W(x, y)$ can be calculated iteratively in some way. One technique for calculating barycenter in Wasserstein space can be found in [9].

KL_{D^*} Barycenter. As a way of data augmentation, all these methods do have some effect on improving robustness and accuracy, compared with Empirical Risk Minimization (ERM). However, the way of mixing is limited and requires human interference on the specific design of mixup mechanism. On the basis of Euclidean barycenter and Wasserstein barycenter, we choose to use a neural network to automatically fit barycenters in latent space, instead of manual calculation. Recall that the goal of data augmentation is to improve the classifier's accuracy and in fact we can view the cross-entropy loss in classification as a divergence. Specifically, it equals to KL-divergence between the mixed (as generated by mixup techniques e.g. [32]) and true labels. Hence we propose a barycenter generator network to fit the KL_{D^*} barycenter and a discriminator network (or called classifier) to control the quality of the generator network. We train both of the networks cooperatively such that we can get them better simultaneously. Note that different from the generator and discriminator in GANs [3, 10, 20], their relation is not adversarial but cooperative to make both better. Similar to the Euclidean (Wasserstein) distance in Euclidean (Wasserstein) space, we define a KL_{D^*} divergence to calculate KL_{D^*} barycenter in Sec. 2.3.

Contributions. Table 1 shows the comparison of different methods, w.r.t. the way of mixing, the mixing phase (mix with raw images or feature maps), the different mixing space and mixing ratios for sample and label. AutoMix is the only one using the network to find the barycenter in Latent space, and more

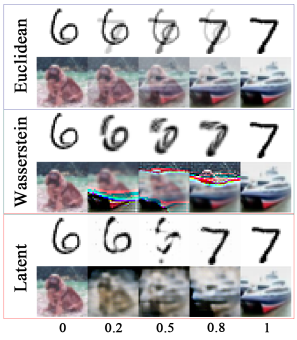


Fig. 1. Different barycenters on MNIST and CIFAR-10.

Table 2. The main observation of this paper: viewing mixup methods from the distance induced space perspective. The proposed method AutoMix enjoys the capability of learning a barycenter from training data tailored to a specific prediction task. In contrast, the Euclidean-distance-based models and Wasserstein-distance-based methods are all learning-free.

Space	Distance and equations	Methods
Euclidean	Euclidean distance, Eq. 3	AdaMixup [11] etc.
Wasserstein	Wasserstein distance, Eq 5	OptTransMix (ours)
Latent	KL_{D^*} divergence, Eq. 7	AutoMix (ours)

broadly speaking, sample mixup by end-to-end network learning. We use Fig. 1 and Table 2 to better summarize the position and results of our approaches: **OptTransMix**, **AutoMix**. The main highlights of our work are:

i) We apply the barycenter theory to view data augmentation and find that many previous studies on image mixing can be viewed as finding Euclidean barycenters to make the discriminator better.

ii) Under this perspective, we propose OptTransMix, whereby Wasserstein distance is used to find the barycenters in contrast to linear interpolation in Euclidean space, as implemented by the optimal-transport-based transformation technique [27]. Though the tool is originally for graphics, while we show it is complementary to bilinear interpolation for mixup-based data augmentation, especially given images with relatively clean backgrounds.

iii) One step further, the cross-entropy loss for image classifier can be viewed as KL divergence between label distribution based on the classifier given two images as input. By this divergence, we devise an AutoMix technique to generate barycenter images with a deep network. It is more general compared with parameters' learning-free mixing methods. Specifically, we have tried to implement such a barycenter generator network with pixel-level processing model U-Net [22] based on SE [14] attention mechanism.

iv) Experimental results on multi-class and multi-label prediction show our method's effectiveness. It also performs robustly against noise, and on the openset problem whereby unseen categories emerge. The source code will be released.

In this paper, we mainly discuss two-image-based barycenter generation for efficiency, while it is straightforward to generalize to multiple images with a larger batch-size. We leave for future work for its necessity and advantage.

2 Proposed Barycenter Learning Method

We first present the main idea of our method, followed by two embodiments named OptTransMix and AutoMix, which is based on fixed Wasserstein distance and learnable network induced space based on training data, respectively.

2.1 Main Idea and Approach Overview

The barycenter learning, especially Wasserstein barycenter learning has been studied in recent years with its common definition in Eq. 1. For data augmentation, we use the barycenter $\hat{x}$ calculated in Eq. 1 as the augmented training set. So we can get the objective for training:

$$\min_{D} \mathbb{E}_{x_i \sim p(x)} \sum_{i=1}^{n} \lambda_i KL\big(y_i || D(\hat{x})\big), \tag{2}$$

where $\hat{x} = \arg\min_x \sum_{i=1}^{n} \lambda_i d(x_i, x)^p$ if given n samples as $x_1, x_2, ..., x_n$ as referred in Eq. 1, $D(\cdot)$ is the target label distribution output by discriminator D, $p(x)$ is the distribution of real images, y_i is the real-label distribution, i.e., one-hot encoding, and $\mathbb{E}[\cdot]$ denotes the expectation. The above object is actually equal to weighted cross-entropy loss. For $p = 2$ and $n = 2$, the barycenter $\hat{x}$ in Euclidean space (i.e., with Euclidean distance) is simplified as:

$$\hat{x} = \arg\min_{x} \lambda ||x_1 - x||_2^2 + (1 - \lambda)||x_2 - x||_2^2, \tag{3}$$

and the solution is $\hat{x} = \lambda x_1 + (1 - \lambda)x_2$, which is exactly the linear mixup methods [15, 32, 42]. For simplicity, barycenter learning is done with different distances $d(\cdot, \cdot)$ and the number of images $n = 2$ in Eq. 1.

Our main approach is based on barycenter learning which relates to optimal transport. We first propose our baseline method called OptTransMix in Sec. 2.2. We extend the optimal transport based barycenter computing to a neural network and propose our main approach AutoMix in Sec. 2.3, which enables learning of barycenter instead of fixed computing.

2.2 OptTransMix: Barycenter Computing by Wasserstein Distance

Although previous studies have partly explained the effectiveness of mixup-like models and achieve good results, while to some extent these methods are basically linear models (either mixing with raw images or feature maps) which is restrictive and equivalent to finding barycenters in Euclidean space. Here we provide a non-linear alternative that leverages the fast optimal-transport-based barycenter computation technique as developed in [27]. The Wasserstein distance in optimal transport can evaluate two probability measures, as given by [1]:

$$W(a_1, a_2) = \min_{\pi \in U(a_1, a_2)} \mathbb{E}_{(x,y)\sim\pi} ||x - y||, \tag{4}$$

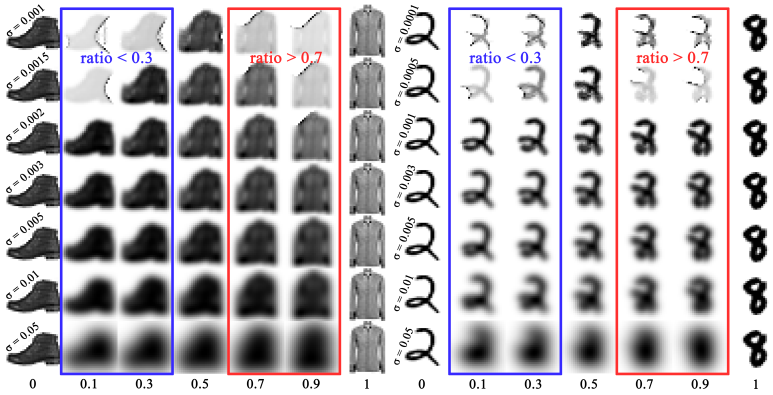


Fig. 2. Results by OptTransMix under smoothing parameter value (y-axis) and ratios (x-axis). We choose the sample whose mixing ratio $[0.1, 0.3]$ or $[0.7, 0.9]$ for training.

where a_1 and a_2 are two probability measures, $U(a_1, a_2)$ is the set of all joint distributions π whose marginals are a_1 and a_2. By viewing a_1 and a_2 as two images, x and y are their pixels, we can calculate the distance between images.

With Wasserstein distance used for Wasserstein barycenter, one can find image barycenters in Wasserstein space as new inputs instead of linear barycenters in Euclidean space for training a classifier (namely discriminator). By setting $p = 1$ in Eq. 1, we can get the Wasserstein barycenter:

$$\min_{a} \sum_{i=1}^{n} \lambda_i W_\sigma(a, a_i), \tag{5}$$

where a_i is the discrete probability measure (i.e., input images), a is the Wasserstein barycenter. For high-resolution input images, we use entropic approximation of barycenters [9] to make it a smooth convex minimization problem which can be calculated by gradient descent. To put it in a same setting with existing mixup methods, we specify $n = 2$ as a nonlinear mixup method.

We call such a geometrical image mixing up method OptTransMix. It considers the global layout of the image content and aims to warp an intermediate one between the two inputs, in a sense of more realistic interpolation in appearance. For synthetic image's label for OptTransMix, we resort to a simple strategy that if the ratio between two input images is under 0.5, then the label is set to the first image's label, otherwise we use the second's label. The samples in the middle (i.e., ratio around 0.5) are not generated for training due to vagueness.

The results generated by OptTransMix are shown in Fig. 2 from MNIST and FASHION-MNIST. Given a set of weights $\lambda = \{\lambda\}_{i=1}^{k} \in \mathbb{R}_{+}^{k}$, we can compute any synthetic one between the two images by re-weighting. Moreover, the regularization strength scalar value σ in the formula, i.e., the smoothing parameter, controls the smoothness of optimal-transport-based interpolation.

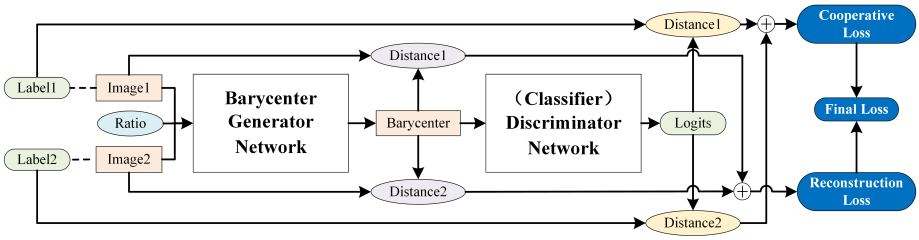


Fig. 3. Our barycenter-based network architecture. Unlike GAN's adversarial training which is time consuming, the two networks in AutoMix reinforce each other and are trained simultaneously thus being much more efficient.

In short, OptTransMix aims to obtain the intermediate states, as a new way of data augmentation. Compared with mixup-like methods' synthetic images (refer to their papers), the ones generated by OptTransMix are more visually meaningful for human and the experimental results also prove the advantages for computer image recognition. However, we find this method has limitations in that it is only applicable to images with clean backgrounds, i.e., the images need to highlight the foreground and ignore the background such that the shape of the image content can be meaningfully interpolated. The failure case of OptTransMix on CIFAR-10 can be seen in the middle of Fig. 1. In fact, the technique is originally applied for graphics with little background noise [27]. Meanwhile, we also note the recent advance on generative adversarial networks (GAN) [10] and Wasserstein GAN [3] related to OT techniques have pushed forward the realistic interpolation of image for specific objects e.g., face [16].

2.3 AutoMix: Barycenter Learning with Learnable Deep Networks

The pipeline of AutoMix is shown in Fig. 3, which is mainly composed of a barycenter generator network and a discriminator network. We input two raw images and the related mixing ratio to get the weighted barycenter, and then feed it into the discriminator to output logits. We use the reconstruction loss of generation process and the cooperative loss of classification process to optimize both of the networks simultaneously. Here we define KL_{D^*} divergence and KL_{D^*} barycenter analogous to the measure rules in Euclidean and Wasserstein space.

Barycenter Generator Network. We start with the barycenter generator network according to Fig. 3 to introduce our AutoMix technique. Specifically, we adopt U-Net [22] as the baseline generative model. This network is known to be state-of-the-art architecture for image segmentation, which we believe is suitable for our pixel-level transformation task. Fig. 4 illustrates the improved U-Net architecture for our barycenter generation task. SE(Squeeze-and-Excitation) [14] module is embedded as an attention mechanism before each downsampling layer in the feature extraction phase based on the original U-Net structure. SE module is divided into three stages, that is, the global pooling layer to generate the

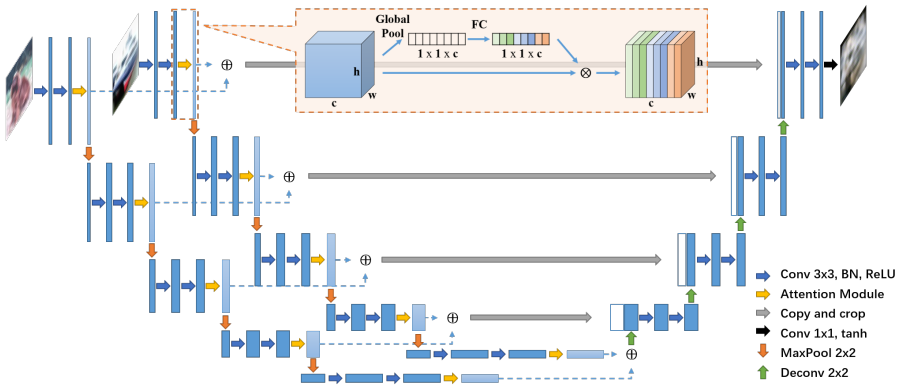


Fig. 4. The improved U-Net with SE modules. Weighted feature maps at various scales of each image are activated by λ and then concatenated with up-sampled feature maps to fuse multi-scale information for the final generation of high-quality barycenters.

embedding of channel-wise feature responses, the fully connected layers to produce a collection of per-channel modulation weights and finally the channel-wise multiplication operation to obtain the weighted feature maps.

We input two raw images and extract their weighted feature maps at multiple scales after passing through the SE modules as shown by the blue dotted arrows in the figure. For feature maps of each scale, the $\bigoplus$ denotes the channel-wise concatenation of two activated feature maps according to the mixing ratio λ. The operation can be summarized as $\left[f^{(l)}(x_1) \times \lambda\right] \bigoplus \left[f^{(l)}(x_2) \times (1-\lambda)\right]$, where x_i denotes the input image, $f^{(l)}(x_i)$ represents the output of SE module at l_{th} scale in U-Net and $l \in [1, 5]$. So we attribute the outstanding performance of AutoMix to our U-Net's excellent feature extraction and feature screening capabilities. With an attention-like module in U-Net, AutoMix can better retain feature maps of two images and mix them appropriately rather than mixup's fixed mixing way.

Although we are not the first to integrate the attention mechanism into U-Net [21, 41], it is worth noting that we may be the first to apply the integration of U-Net and SE module to the field of image generation and achieve good performance according to the experimental results. Besides, our model can accept more than two inputs to generate barycenters of multiple images after expanding.

$\mathbf{KL}_{D^*}$ divergence. Similar to Euclidean and Wasserstein barycenters based on their respective distances, the divergence in Latent space for barycenter learning needs to be defined and we call it KL_{D^*} divergence. In fact, the corresponding barycenter can be derived from the cross-entropy, i.e., KL-divergence for the classifier. For an image x, let $D(x)$ denotes the prediction probability with a discriminator D, which is exactly the distribution of x in Latent space. If x comes from the original data domain, we set $D(x) = y$, otherwise $D(x)$. Here we define an optimal discriminator D^* to combine these two cases, and based on

this observation, we then define the KL_{D^*} divergence given two images x_1, x_2:

$$D^*(x) = \begin{cases} y, & \text{if } x \text{ is a raw image} \\ D(x), & \text{if } x = G(x_1, x_2, \lambda) \end{cases}, KL_{D^*}(x_1||x_2) \stackrel{\text{def}}{=} KL\big(D^*(x_1)||D^*(x_2)\big). \tag{6}$$

KL_{D^*} Barycenter. Different from Euclidean barycenter and Wasserstein barycenter, the barycenter based on KL_{D^*} divergence can not be calculated directly by iterations but gradient descent as it is based on deep-net discriminator. So we use barycenter generator network G, specifically a U-Net [22] architecture to generate the barycenter given two images x_1, x_2 and their ratio λ, as expressed by $\hat{x} = G(x_1, x_2, \lambda)$. So we can get the KL_{D^*} barycenter by optimizing

$$\min_G \lambda KL_{D^*}(x_1||\hat{x}) + (1-\lambda) KL_{D^*}(x_2||\hat{x})\,. \tag{7}$$

Cooperative loss. The generated barycenter $\hat{x}$ is input into discriminator D for training as the augmented sample. Akin to Euclidean (Wasserstein) distance for Euclidean (Wasserstein) barycenter, KL_{D^*} divergence is used for KL_{D^*} barycenter, which is also the loss for discriminator (recall it is image classifier). Hence our optimization for barycenter generator network and discriminator network refers to minimizing the KL divergence between real label distribution and Latent distribution output by D:

$$L_{G,D}(x_1, x_2) = \lambda KL\big(y_1||D(\hat{x})\big) + (1-\lambda) KL\big(y_2||D(\hat{x})\big), \tag{8}$$

where $D(\hat{x})$ is the target distribution in view of discriminator D with barycenter $\hat{x}$, and $\lambda \sim U(0,1)$ is the mixing ratio. Different from GANs [3,10], here the loss for G and D is cooperative. When D reaches optimum, the optimization goal changes from Eq. 8 to Eq. 7. While given optimized G, we just need to optimize Eq. 2, to get discriminator $D \to D^*$ and the generated barycenter closer to the theoretical optimum.

Reconstruction loss. In line with CGAN [20], in order to prevent the generated image from being too far from the original ones, L1 regularization is imposed by a reconstruction loss to train the barycenter generator network:

$$L_G(x_1, x_2) = \lambda||x_1 - \hat{x}||_1 + (1-\lambda)||x_2 - \hat{x}||_1, \tag{9}$$

where x_1 and x_2 are two input images, and $\hat{x} = G(x_1, x_2, \lambda)$ is the barycenter generated by G. The reason for using L1 instead of L2 is that L1 produces images with relatively higher definition.

Final loss. Unlike GAN's adversarial training process, the two networks in AutoMix reinforce each other and are trained jointly with a factor α controlling the weight of two loss terms (not the difference to min max in GAN):

$$\min_D \min_G \mathbb{E}_{x_1, x_2 \sim p(x)} L_{G,D}(x_1, x_2) + \alpha L_G(x_1, x_2). \tag{10}$$

where $p(x)$ is the distribution of real images. We find the performance by setting different α from 0.5 to 2 with a step 0.1 is very small. We finally set $\alpha = 1.5$. The barycenters generated by OptTransMix and AutoMix are compared in Fig. 1.

Table 3. Sampling and training protocols. We use 10-fold cross-validation for MIML because of its small size and we train for 10 trials for other datasets. We train models on both sampling and full set of some of the datasets and test models on their official testing set respectively to explore how these methods perform on small scale datasets. CIFAR-100 and Tiny-ImageNet are only trained with full dataset.

label	dataset	#cls.	sampling	sub/full training set	testing set	training strategy
single	MNIST [19]	10	50 per class	500 / 60,000	official 10,000	10 trials
	F-MNIST [37]	10	50 per class	500 / 60,000	official 10,000	
	CIFAR-10 [17]	10	500 per class	5,000 / 50,000	official 10,000	
	CIFAR-100 [17]	100	500 per class	- / 50,000	official 10,000	
	GTSRB [29]	43	50 per class	2,150 / 39,209	official 12,630	
	T-IMAGENET [39]	200	500 per class	- / 100,000	official 10,000	
multi	MIML [44]	5	train:test=1:1	1000	1000	10-fold CV

We also have tried alternating optimization strategy for two networks while the convergence become slower with worse final performance. Besides the standard classification setting, we also explore the possibility of applying our OptTransMix and AutoMix to other common tasks. The first one is multi-label prediction [34, 43], the other is the so-called openset problem [6, 24] whereby there exist new categories in the testing set which are unseen in the training set. Especially for the second problem, this is still considered as an open problem and a simple but effective treatment is to adopt a threshold-based strategy to decide if the test sample shall be regarded as an unseen one that does not belong to any of the existing categories in training set. Precisely, if the highest inference score (or the top 2 or so) is under a given threshold, this sample will be classified as an unseen sample. Note the threshold can be determined in more diverse and adaptive ways e.g., depending on the specific label of samples. In this paper, for simplicity and generality, we use a fixed threshold for all labels in experiments.

3 Experiment and Discussion

Experiments are conducted on a desktop with Nvidia GeForce GTX1080Ti GPU and 32G memory. We verify our model on public datasets including MNIST [19], FASHION-MNIST [37], the traffic sign benchmark GTSRB [29], CIFAR-10 [17], CIFAR-100 [17] for image classification. The reason for using these datasets is that they have a relatively small size such that the effects of different mixup strategies can be more pronounced as a way of data augmentation. This is also practically common in front of small data for network learning. Besides, for larger dataset, we choose Tiny-ImageNet [39] instead of ImageNet [23] due to our hardware limitation. To further evaluate the behavior of our methods on different tasks, we also adopt the MIML [44] dataset for image multi-label prediction. Note for our OptTransMix, it is only tested on classification benchmark MNIST and FASHION-MNIST as we find it requires clean background. The dataset sampling and training protocols are detailed in Table 3.

Table 4. Top-1 error rate (mean ± standard deviation) on MNIST/FASHION-MNIST by 10 trials. Underline denotes best results.

Method	MNIST(full)	MNIST(sub)	F-MNIST(full)	F-MNIST(sub)
Baseline	0.271 ± 0.012	10.83 ± 0.61	5.202 ± 0.087	20.95 ± 0.97
BC [32]	0.257 ± 0.032	5.93 ± 0.59	4.905 ± 0.071	16.69 ± 0.31
Mixup [42]	0.268 ± 0.015	6.17 ± 0.85	4.788 ± 0.066	17.38 ± 0.59
Manifold Mixup [35]	0.258 ± 0.021	4.98 ± 0.34	4.910 ± 0.079	17.47 ± 0.38
OptTransMix	0.260 ± 0.031	4.86 ± 0.48	5.021 ± 0.068	16.97 ± 0.65
OptTransMix+Mixup	0.265 ± 0.018	8.12 ± 0.48	4.869 ± 0.078	17.10 ± 0.76
OptTransMix+BC	0.257 ± 0.028	$\underline{4.37} \pm \underline{0.44}$	4.872 ± 0.059	$\underline{16.67} \pm \underline{0.53}$
AutoMix	$\underline{0.256} \pm \underline{0.036}$	4.58 ± 0.57	$\underline{4.769} \pm \underline{0.064}$	16.79 ± 0.75

For discriminator backbone, we use a simple 11-layer CNN (2conv + pool + 2conv + pool + 4conv + pool + 3fc) according to [32] and ResNet-18 [12]. The 11-layer CNN is used for MNIST and FASHION-MNIST, while ResNet-18 is used for the other datasets.

We compare the baseline (namely without mixing method) and three mixup methods including Between-Class (BC) learning [32], Mixup [42] and Manifold Mixup [35]. Regarding the details of training, we apply data pre-processing including random cropping, horizontal flipping and normalization on the raw data. We choose Stochastic Gradient Descent with $momentum = 0.9$ as our optimizer and the learning rate is set to 0.1 at the beginning and decays as the training process continues. For classification (single-label prediction), we adopt the log-softmax cross-entropy as loss. For multi-label prediction, as it can be arguably treated as a binary classification problem by selecting one label as positive and the others negative [34], we use sigmoid activation function as the last layer of discriminator network and the sigmoid cross-entropy as the loss function, for simplicity. Note the effectiveness of such a one vs. rest binary classification treatment has been also verified [4, 40].

Single-label classification. We first apply OptTransMix and AutoMix on two simple image classification datasets: MNIST and FASHION-MNIST. In addition, since OptTransMix is orthogonal to Between-Class learning [32] and Mixup [42], we also apply it to generate additional samples for these two methods. We averagely sample 50 images per class to form a training set with a total of 500 images, which is relatively small and thus calls for data augmentation. We also train with full data to figure out how these methods actually work. For testing, we use the official 10,000-image testing set for both MNIST and FASHION-MNIST. To eliminate the impact of random initialization, we train each model for 10 trials. An 11-layer CNN is used as the backbone and is trained for 100 epochs. The learning rate is divided by 10 at the epoch in {50, 75}. Top-1 accuracy is used as the evaluation metric for single-label classification tasks.

We show the results of 10 trials on MNIST and FASHION-MNIST in Table 4. Improvement is made by using OptTransMix and AutoMix. Besides, using Opt-

Table 5. Top-1 error rate (mean ± standard deviation), and mAP for MIML using ResNet-18 trained by 200 epochs for 10 trials. Underline denotes best results.

Dataset	baseline	BC [32]	Mixup [42]	M-Mixup [35]	**AutoMix**
CIFAR10 (full)	4.59 ± 0.09	4.03 ± 0.34	3.96 ± 0.10	3.99 ± 0.06	$\underline{3.72} \pm \underline{0.10}$
CIFAR10 (sub)	18.79 ± 0.33	15.97 ± 0.18	15.76 ± 0.33	15.38 ± 0.22	$\underline{15.17} \pm \underline{0.41}$
CIFAR100 (full)	22.39 ± 0.17	21.19 ± 0.24	21.05 ± 0.24	20.25 ± 0.17	$\underline{20.04} \pm \underline{0.14}$
GTSRB (full)	0.44 ± 0.06	0.41 ± 0.04	0.54 ± 0.05	$\underline{0.38} \pm \underline{0.11}$	0.41 ± 0.08
GTSRB (sub)	4.52 ± 0.69	2.38 ± 0.22	2.98 ± 0.27	2.38 ± 0.21	$\underline{2.34} \pm \underline{0.18}$
T-ImageNet (full)	40.35 ± 0.25	38.71 ± 0.27	38.77 ± 0.19	37.75 ± 0.46	$\underline{37.15} \pm \underline{0.26}$
MIML (full)	72.14 ± 1.08	72.92 ± 0.85	73.02 ± 0.80	74.56 ± 1.02	$\underline{74.80} \pm \underline{0.92}$

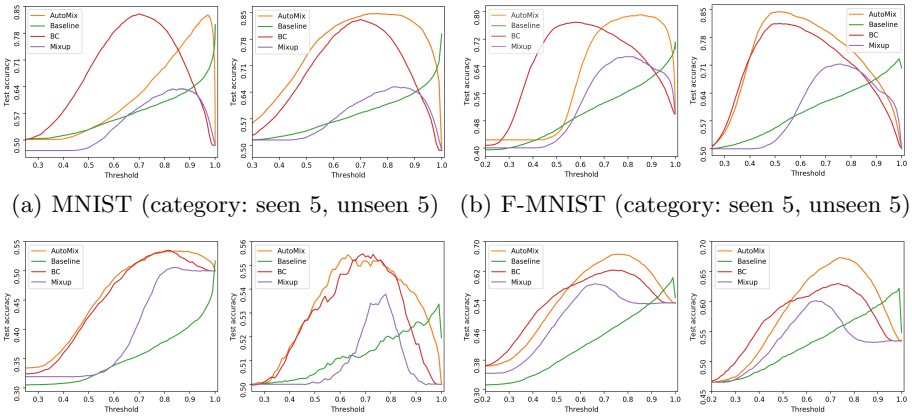
TransMix to generate more diverse samples can further improve the performance of Mixup and BC. Moreover, OptTransMix performs better on small datasets, while AutoMix outperforms on relatively bigger ones.

We then apply AutoMix to some of the bigger and more difficult datasets. Similarly, we still averagely sample training images, namely 500 images per class for CIFAR-10 and 50 images per class for GTSRB, so the total amount of training set is 5,000 images and 2,150 images respectively. Full data training is also conducted. Since CIFAR-100 and Tiny-ImageNet only have 500 images per class, we only train with full data on them. ResNet-18 is used to train 100 epochs for GTSRB and 200 epochs for CIFAR-10/100 and Tiny-ImageNet. The learning rate is divided by 10 at the epoch in {50, 75}, {100, 150} and {100, 150} respectively. We test the model on the official 10,000-image testing set for CIFAR-10/100 and Tiny-ImageNet, and the official 12,630-image testing set for GTSRB.

The average of 10 trials are reported in Table 5, showing that AutoMix outperforms baseline and other mixup-like methods on all datasets except GTSRB.

Multi-label classification. MIML [44] is a multi-instance multi-label dataset, which has a total of 2,000 landscape images. Due to its small amount of data, we divide it into a training set and a testing set with the ratio of 1:1. We use 10-fold cross-validation to evaluate whether our methods perform well on multi-label image classification tasks compared with baseline and other image mixing methods. ResNet-18 is used to train 100 epochs and the learning rate is divided by 10 at the epoch in {50, 75}. Mean average precision (mAP) is used for multi-label classification tasks, which is calculated in a similar way to mAP in object detection. As shown in Table 5, AutoMix outperforms as well.

Openset problem. The openset problem can be defined as there exist unseen categories in testing set but not in training set. In this part, we train the model by a subset of categories and test by whole categories. In testing phase, 'threshold' is used to compare with the confidence to control whether a sample will be classified as a known or unknown category. We conduct results in two settings here: **a) normal openset setting (n seen categories + 1 'unknown' category)** and **b) binary classification setting (1 'known' category + 1 'unknown' category).** The results are reported in Fig. 5, illustrating the



(a) MNIST (category: seen 5, unseen 5) (b) F-MNIST (category: seen 5, unseen 5)

(c) CIFAR-10 (category: seen 5, unseen 5) (d) GTSRB (category: seen 12, unseen 31)

Fig. 5. Top-1 accuracy as threshold grows on the openset problem. Left in each pair: setting a). Right in each pair: setting b). See main test for details of the two settings.

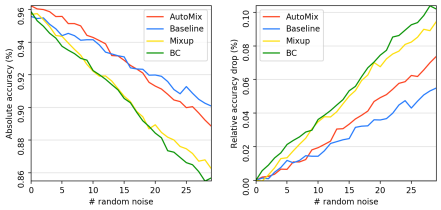


Fig. 6. Absolute accuracy (left) and relative accuracy drop (right) as noise increases. The higher (lower) the better.

Table 6. Ablation study on AutoMix using ResNet-18 on CIFAR-10 by top-1 error rate (mean ± standard deviation of 10 trials). Underline denotes best results.

Comparison	Setting	CIFAR10 (full)	CIFAR10 (sub)
label strategy for synthetic sample	(a)	5.38 ± 0.11	16.32 ± 0.43
	(b)	3.91 ± 0.17	15.44 ± 0.34
	(c)	3.86 ± 0.11	15.35 ± 0.31
	(d)	$\underline{3.72} \pm \underline{0.10}$	$\underline{15.17} \pm \underline{0.41}$
label conflict	same	4.21 ± 0.18	16.10 ± 0.39
	different	3.98 ± 0.15	15.67 ± 0.32
	random	$\underline{3.72} \pm \underline{0.10}$	$\underline{15.17} \pm \underline{0.41}$

relationship between accuracy and threshold. The left figures in each subplot correspond to setting a) and the right figures correspond to b). Intuitively, the bigger the area under the curve is, the more robust the model is, which means we can get high performance in a wide range of threshold values. By a closer study, one can find our method not only achieves a higher peak of accuracy (with optimal threshold value) but also can be less sensitive to threshold, which proves the effectiveness of AutoMix in dealing with openset problems.

Performance against noise. We evaluate the robustness on CIFAR-10 by ResNet-18, by adding random salt-and-pepper noise with level from 0 to 30. The absolute and relative accuracy drop as the noise increases is shown in Fig. 6. Note that the baseline model (blue) has the best robustness against noise, followed by AutoMix (red), and Mixup (yellow) and BC (green) are the worst. The results show that AutoMix can improve the classification accuracy while maintaining the robustness against the adversarial samples with simple noise added.

Ablation study on AutoMix. We are interested in how AutoMix behaves with different mixed labels and different sample constraints fed to the network. Hence an ablation analysis is conducted and the results are shown in Table 3.

We investigate the relationship between the performance and the label settings. We attempt to treat it as a multi-class classification task, either **taking the first image's label (a)** or the **half-half mix of two labels (b)** as the final label. According to the experimental results, treating it as a **multi-label classification task (c)**, i.e., simply add the two labels and constrain it between 0 and 1, is more effective. Furthermore, our proposed AutoMix using **random ratios for label weighting (d)** achieves a more considerable result.

We also study how the categories of the two input images to be mixed affects performance. We find that the performance degrades if the given images have the same label. The best results come from two randomly selected samples, which means AutoMix can learn both within-class and Between-Class features by the barycenter generated in Latent space.

Discussion. The reason why OptTransMix excels on small datasets (Table 4) may be that the optimal-transport-based Wasserstein barycenter can be calculated with any amount of data. Compared to the U-Net in AutoMix, OptTransMix does not need a large amount of data for training to achieve good performance.

Regarding the computational complexity of the models, although the Wasserstein barycenter is computationally intensive (especially for high-dimensional data), we can separate this generation process before the training stage. Thus to some extent, the complexity has not increased for OptTransMix. For AutoMix, the U-Net introduces ~ 1.96 million additional parameters beyond the ~ 11.27 million parameters required by ResNet18, corresponding to $\sim 17\%$ increase.

4 Conclusion

This paper aims to learn the interpolation model for data augmentation which is a general and fundamental building block for practical learning systems. Differing from most existing mixup methods that interpolate the raw images or their feature maps in the Euclidean space by linear or bilinear interpolation, we first explore a more advanced interpolation technique called OptTransMix, which seeks the Wasserstein barycenter between two images, and show their usefulness on small datasets. Then we generalize the fixed Wasserstein-distance-based model to the new approach called AutoMix, which stands out on large datasets. It trains an attention-based barycenter generator network and a discriminator network concurrently with the cooperative loss.

Experiments have shown the efficacy of both our proposed techniques on traditional classification, multi-label prediction, openset problems, and the robustness test against noise. Limited by our hardware device, we leave the experiments on large scale datasets for our future work.

References

1. Agueh, M., Carlier, G.: Barycenters in the wasserstein space. In: SIAM J. on Mathematical Analysis. pp. 904–921 (2011)
2. Anderes, E., Borgwardt, S., Miller, J.: Discrete wasserstein barycenters: Optimal transport for discrete data. Mathematical Methods of Operations Research pp. 389–409 (2016)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning (2017)
4. Babbar, R., Schölkopf, B.: Dismec: distributed sparse machines for extreme multi-label classification. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 721–729 (2017)
5. Beckham, C., Honari, S., Verma, V., Lamb, A.M., Ghadiri, F., Hjelm, R.D., Bengio, Y., Pal, C.: On adversarial mixup resynthesis. In: Advances in Neural Information Processing Systems. pp. 4348–4359 (2019)
6. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1563–1572 (2016)
7. Berthelot, D., Raffel, C., Roy, A., Goodfellow, I.: Understanding and improving interpolation in autoencoders via an adversarial regularizer. In: International Conference on Learning Representations (2018)
8. Chapelle, O., Weston, J., Bottou, L., Vapnik, V.: Vicinal risk minimization. In: Advances in Neural Information Processing Systems. pp. 416–422 (2001)
9. Cuturi, M., Doucet, A.: Fast computation of wasserstein barycenters. In: International Conference on Machine Learning. pp. 685–693 (2014)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
11. Guo, H., Mao, Y., Zhang, R.: Mixup as locally linear out-of-manifold regularization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3714–3722 (2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
13. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine pp. 82–97 (2012)
14. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141 (2018)
15. Inoue, H.: Data augmentation by pairing samples for images classification. arXiv preprint arXiv:1801.02929 (2018)
16. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
17. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech Report (2009)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105 (2012)

19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE pp. 2278–2324 (1998)
20. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
21. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 (2018)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision pp. 211–252 (2015)
24. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE transactions on Pattern Analysis and Machine Intelligence pp. 1757–1772 (2013)
25. Simard, P.Y., LeCun, Y.A., Denker, J.S., Victorri, B.: Transformation invariance in pattern recognition-tangent distance and tangent propagation. Lecture Notes in Computer Science pp. 239–239 (1998)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (2015)
27. Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. ACM Transactions on Graphics (TOG) pp. 1–11 (2015)
28. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research pp. 1929–1958 (2014)
29. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In: IEEE International Joint Conference on Neural Networks (2011)
30. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems. pp. 3104–3112 (2014)
31. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2015)
32. Tokozume, Y., Ushiku, Y., Harada, T.: Between-class learning for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5486–5494 (2018)
33. Tokozume, Y., Ushiku, Y., Harada, T.: Learning from between-class examples for deep sound recognition. International Conference on Learning Representations (2018)
34. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. Data Mining and Knowledge Discovery Handbook pp. 667–685 (2010)
35. Verma, V., Lamb, A., Beckham, C., Najafi, A., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: International Conference on Machine Learning, 6438-6447 (2019)
36. Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: when to warp? In: International Conference on Digital Image Computing: Techniques and Applications. pp. 1–6 (2016)

37. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
38. Yaguchi, Y., Shiratani, F., Iwaki, H.: Mixfeat: Mix feature in latent space learns discriminative space. Submission at International Conference on Learning Representations (2019)
39. Yao, L., Miller, J.: Tiny imagenet classification with convolutional neural networks. CS 231N p. 8 (2015)
40. Yen, I.E.H., Huang, X., Ravikumar, P., Zhong, K., Dhillon, I.: Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In: International Conference on Machine Learning. pp. 3069–3077 (2016)
41. Yue, D., Hua-jun, F., Zhi-hai, X., Yue-ting, C., Qi, L.: Attention res-unet: an efficient shadow detection algorithm. Journal of Zhejiang University(Engineering Science) p. 373 (2019)
42. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. International Conference on Learning Representations (2018)
43. Zhang, W., Yan, J., Wang, X., Zha, H.: Deep extreme multi-label learning. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval. pp. 100–107 (2018)
44. Zhou, Z.H., Zhang, M.L.: Multi-instance multi-label learning with application to scene classification. In: Advances in Neural Information Processing Systems. pp. 1609–1616 (2007)