ReSmooth: Detecting and Utilizing OOD Samples When Training With Data Augmentation

Chenyang Wang®, Junjun Jiang®, Senior Member, IEEE, Xiong Zhou®, and Xianming Liu®, Member, IEEE

Abstract—Data augmentation (DA) is a widely used technique for enhancing the training of deep neural networks. Recent DA techniques which achieve state-of-the-art performance always meet the need for diversity in augmented training samples. However, an augmentation strategy that has a high diversity usually introduces out-of-distribution (OOD) augmented samples and these samples consequently impair the performance. To alleviate this issue, we propose ReSmooth, a framework that first detects OOD samples in augmented samples and then leverages them. To be specific, we first use a Gaussian mixture model (GMM) to fit the loss distribution of both the original and augmented samples and accordingly split these samples into in-distribution (ID) samples and OOD samples. Then we start a new training where ID and OOD samples are incorporated with different smooth labels. By treating ID samples and OOD samples unequally, we can make better use of the diverse augmented data. Furthermore, we incorporate our ReSmooth framework with negative DA (NDA) strategies. By properly handling their intentionally created OOD samples, the classification performance of NDAs is largely ameliorated. Experiments on several classification benchmarks show that ReSmooth can be easily extended to the existing augmentation strategies [such as RandAugment (RA), rotate, and jigsaw] and improve on them. Our code is available at https://github.com/Chenyang4/ReSmooth.

Index Terms—Data augmentation (DA), out-of-distribution (OOD) detection, sample selection, visual recognition.

NOMENCLATURE

| DA | Data augmentation. |
| NDA | Negative data augmentation. |
| OOD samples | Out-of-distribution samples. |
| ID samples | In-distribution samples. |
| DAOOD samples | DA-incurred OOD samples. |
| DAID samples | In-distribution data augmented samples. |
| LS | Label smoothing. |
| LSR | Label smoothing regularization. |

I. INTRODUCTION

DA is an essential and effective technique for the learning problem. By virtue of it, many deep learning methods can substantially boost their performance in various vision tasks, including object recognition [1], [2], [3], defect detection [4], clustering [5], [6], [7], object detection [8], and semantic segmentation [9]. There are two main intuitions in the literature to design a good DA strategy, namely, minimal data distribution shift and maximal diversity of augmented samples. Typical examples for the former include image transformations such as rotation, flipping, and cropping [10], while the latter mainly includes methods considering the combination of multiple transforms [2], [3], [11]. According to the empirical results in [12], a better augmentation policy should have both bigger affinity (between the augmented data distribution and the original data distribution) and bigger diversity (of augmented samples).

However, the recently proposed DA strategies that achieve state-of-the-art performance usually ignore the former and mainly focus on the diversity property, which we call diverse DA. As a result, some samples are inevitably out of the original data distribution, leading to a decline in affinity. For these samples, one feasible solution is to discard them during training, but this will attenuate the diversity of the training samples and consequently limit the model performance (see Section IV-F1 for details). Therefore, we wonder whether there is a strategy to maintain the diversity of the augmented samples while not detrimentally affecting the learning of samples lying in the original data distribution.

To that effect, we dig into these OOD samples and note an interesting phenomenon: a big part of these OOD samples, unlike the announcements in previous works [13], [14], are not noisy or ambiguous but semantic-unchanging for the target task, which means these samples do not change their task-related attributes (e.g., labels in the context of the classification task) from the view of us. We call these samples DAOOD samples, and some examples on ImageNet are shown in Fig. 1. This phenomenon widely exists in natural image datasets, but treating DAOOD samples and ID samples (including both
Fig. 1. Some examples of DAOOD samples in the ImageNet training set. For each sample, we show the image before and after augmentation in the top left and top right, and the bottom left and right are their predictions by the model pretrained on unaugmented data.

the original samples and DAID samples) naively equal will impair the fit of ID samples and thus result in suboptimal performance. In this article, we mainly focus on leveraging these samples with our proposed ReSmooth framework.

ReSmooth adopts a detecting–utilizing two-step scheme to make use of DAOOD samples. In the first step, we try to divide the training samples into ID samples and DAOOD samples. In the second step, we differently treat the two groups of the samples where, ideally, the introduction of DAOOD samples will benefit the learning of ID samples. To be specific, we first train a network with unaugmented samples and then use the trained network to estimate a mixture model by loss distribution modeling. Then, the mixture model is used to distinguish OOD samples from ID samples. With this separation, ID samples are trained by the original training scheme with standard cross-entropy loss, whereas OOD samples are trained with LS. The per-sample smoothness parameter is acquired according to the posterior probability predicted by the estimated mixture model. With this divide-and-conquer strategy, our method can make full use of both the original data and the newly obtained augmented data.

Unlike diverse augmentation, some known transformations in the literature are rarely used in regular training pipelines for the reason of poor performance or unsatisfying interpretability, e.g., rotation with large angle as in [15] and jigsaw transformation as in [16]. These transformations are also known as NDA [17] which pursues intentionally creating OOD samples. Though not lying on the support of the data distribution, samples created in this way are informative and can be semantic-unchanging for the target task. They can be viewed as DAOOD samples from another source and be leveraged by the proposed ReSmooth.

The main contributions of this work are summarized as follows.

1) We reveal an interesting phenomenon that lots of OOD samples caused by diverse DAs are semantic-unchanging and can be exploited for training.
2) We propose a novel and simple framework for learning with diverse and NDA. The framework can be flexibly adopted by combining different DA strategies, data split methods, and OOD utilizing designs.
3) By combining with different DA strategies, we empirically show that our method can further improve on the existing SOTA DA strategies.

The rest of this article is unfolded as follows. Section II introduces previous works related to ReSmooth, including related DA strategies, data split methods, and learning strategies. Section III introduces the adopted DA techniques in our work, gives the definition of DAOOD samples, and presents the proposed framework, and Section IV shows the comparison results of our proposed method with others, ablation analysis, and some potential concerns. Section V concludes this study. Nomenclature is provided for convenience of reading the rest of this article.

II. RELATED WORK

Our work aims at detecting and utilizing OOD samples when training with DA. In other word, the key contribution of our work is the data separating strategy and the following learning strategy under the background of DA. In this section, we retrospect the related works of DA, data separating methods, and learning strategies with separated data.

A. Data Augmentation

In this article, we mainly focus on the DA which pursues the diversity property. AA [2] is a typical diverse DA method for searching the optimal combination of basic transformations by reinforcement learning. Following the idea of auto augmenting, some efforts are made to alleviate the prohibitive computation cost of AA. To name some of many, population-based augmentation [18] applied evolution algorithm to generate the augmentation policy, RA [3] utilized a reduced search space for cost-free search, and AutoDo [19] modeled the parameters in augmentation as a source of hyperparameters and optimize them with scalable probabilistic implicit differentiation. However, these augmentation strategies may introduce low-quality samples and harm the performance. To that
effect, AugMix [20] tried to output natural-looking samples, Gong et al. [14] proposed to keep the fidelity of the augmented samples by pasting the patches back from the original samples or avoiding cutting the important areas, and Suzuki [21] introduced teacher knowledge to constrain adversarial DA strategies. What is more, Fast AA [22] and Faster AA [23] are two follow-up researches of AA which explicitly match the density between augmented and unaugmented data.

The work most closely related to ours is [13]. In [13], like ReSmooth, a network is first trained with unaugmented data and then the augmented data are inputted to the network to get the predictions. The key differences lie in how to train a new better network given the predictions. Wei et al. [13] builds upon the observations that when heavy DA is added to the training image, and it is probable that part of its semantic information is removed. As a result, insisting on the ground-truth label is no longer the best choice and the predictions may represent the true semantics left in the data. Therefore, Wei et al. [13] propose to distill the new network with the predictions. In ReSmooth, however, we just use the predictions to estimate a loss distribution and softly split the data accordingly for a new training, because we argue that DAOOD samples are the most of cases in augmented data and distillation will harm the learning of them. We provide more analysis of the augmented data in Section IV-F2 and give the possible reason why we perform better than [13] there.

B. Heterogeneous Data Separating

Real-world data are collected from multiple sources and therefore is always heterogeneous. A branch of methods handling this situation is to separate the heterogeneous data into homogeneous splits based on some statistics during training or from extra prior knowledge. For the conventional OOD detection setting [24], a pretrained model is provided by solving tasks on the ID samples and a score is needed for the detection split. Based on the pretrained model, Hendrycks and Gimpel [25] used maximum Softmax score, Lee et al. [26] adopted the maximum Mahalanobis distance of the penultimate layer representations, and Liu et al. [27] proposed the energy score to measure the sample energy. For the task of noisy label learning, [28] and [29] proposed to model the data noise according to the loss statistics during training. They suggest that the loss distributions are separable during the early stage of training. Different from them, [30], [31], and [32] proposed to identify the samples with noisy labels directly from the loss values of two separated models. In addition to loss statistics, some methods explore to use more complicated and effective statistics, such as area under the margin ranking [33]. In unsupervised representation learning task with imbalanced data, Hooker et al. [34] and Jiang et al. [35] identified hard-to-memory samples from tail classes by different outputs between the network and its pruned version. Some works used a threshold of the loss to distinguish atypical data [36] or not well-learned data [37] in voice–face mapping and semisupervised learning, respectively. Here, we mainly focus on the heterogeneous data created by DA and follow the idea of noise distribution modeling to separate the ID and DAOOD samples.

C. Learning Strategy With Separated Data

Different tasks treat separated samples differently. For example, in the noisy label learning task, identified noisy samples can be dropped or treated as unlabeled samples. In the field of DA, augmented data can be viewed as naturally separated data compared with the original data. Based on this assumption, [38] proposed to use split BatchNorms to capture different statistics. Yi et al. [39] used a much more fine-grained strategy considering per-sample per-augmentation reweighting with a min–max optimization scheme. Different from these methods, in this work, we consider the relationships among samples augmented by the same strategy. We follow the assumption that the property of an augmented sample is a function of both the sample and the augmentation strategy. So even given a specific DA, the learning strategy should vary across different samples, especially for diverse augmentation strategies such as RA [3].

III. Method

In this section, we first introduce DA techniques considered in this work including diverse DA and NDA. Then we give the definition of the so-called DAOOD samples. Finally, we introduce our ReSmooth framework and specific detection and utilizing strategies.

A. Preliminaries: Data Augmentation

We focus on two main kinds of DA: diverse DA and NDA. Diverse DA implies the augmentation that creates a much more diverse training set throughout the training phase, while NDA refers to the augmentation that aims to create OOD samples during training. In this work, we propose to estimate data distributions in the loss space (see Section III-B), and thus diverse DA and NDA can be further depicted with the help of loss statistics. In this sense, diverse DA refers to the augmentation where augmented samples are of a big variance/diversity in the loss space. Specifically, a part of augmented samples of the diverse DA lie in the original data distribution, while another part of augmented samples are not. For augmented samples of the NDA, the situation changes, that most of the augmented samples are heavy OOD and lie far away from the original data distribution. To be noticed, both diverse DA and NDA considered in this work are ideally semantic-unchanging for the classification task.

We choose RA [3] as the representative of diverse DA and choose jigsaw and rotation as the representatives of NDA in our experiments. The loss statistics of them can be found in Fig. 2.

RA considered the combination of multiple known transformations for image data, such as contrast, sharpness, and solarize. To make the transformed images meaningful for DA, RA designed a largely reduced search space to find the optimal parameters to combine the transformations. Because of its probabilistic augmentation property, the augmented samples are of a big diversity.

Jigsaw transformation is a transformation that rearranges the spatial order of patches. It is a representative transformation that destroys the spatial correlation of images but...
preserves the local information. Though it is OOD, viewing it as the positive DA can help the model to distinguish object parts and/or resemble parts to recognize it.

Rotation transformation is also semantic-unchanging except for the rotation angle. It can be viewed as a kind of NDA for the reason that samples after rotation with big angles (90°, 180°, 270°) are heavy OOD. It may relate to the inductive bias of the ConvNet which does not have the rotation-invariant property. Whatever, rotated images follow the definition of OOD, viewing with DAOOD is not like learning with noisy data that harm performance. The key lies in how to generalize to the introduced knowledge without confusing the model or impairing the learning of the existing knowledge in the original dataset.

B. Data Augmentation-Incurred OOD

We begin with the formulation of the learning problem with DA. Let \( \mathcal{X} \) denote the data space and \( \mathcal{Y} = \{1, \ldots, K\} \) denote the label set. We are supposed to learn a labeling function \( f : \mathcal{X} \rightarrow \mathcal{Y} \) on the dataset \( D = \{(x_i, y_i)\}_{i=1}^N \) drawn from a distribution \( P \) over \( \mathcal{X} \times \mathcal{Y} \). Given a loss function \( \ell \), we usually minimize the empirical risk of \( \ell \) over the observed dataset \( D \)

\[
R_{\text{emp}}(f) = \frac{1}{|D|} \sum_{(x_i, y_i) \in D} \ell(f(x_i), y_i). \quad (1)
\]

Let \( \mathcal{A}(\cdot|x) \) denote the distribution of the transformations in a DA strategy and \( D' \) denote the augmented dataset. The sample in \( D' = \{(x'_i, y'_i)\}_{i=1}^N \) follows: \( (x'_i, y'_i) = (f(x_i), y_i) \), \( t \sim \mathcal{A}(\cdot|x) \). Though minimizing (1) on \( D' \) is tenable and most of the existing methods are based on the objective, it is hard to optimize in practice. We find the main reason behind this phenomenon is the introduction of DAOOD samples. Here, we refer to DAOOD samples as the augmented samples with semantics unchanged but model-related metric changed. In other words, DAOOD samples are augmented samples where semantic information, especially task-related information, is unchanged from the view of human being but changed from the view of the model.

To further illustrate that point, we compare it with noisy label learning. Data with noisy labels always introduce harmful knowledge for learning but data augmented mainly introduces new knowledge for better generalization. As a result, learning with DAOOD is not like learning with noisy data that harm the original data distribution implicitly. By it, we can draw the mixture of the original distribution and OOD distribution in the log \( \ell \) space and estimate a GMM model to fit it. Then in the ReSmooth training stage, we can predict posterior probability \( w_i \) of each input and define the smooth parameter linearly by \( a_i = a \cdot w_i \). Using these sample-wise smooth parameters, we can train the network according to (5).

C. ReSmooth: DAOOD Detection and Utilization

Our proposed ReSmooth framework first detects DAOOD samples by leveraging a model pretrained on unaugmented data and an estimated GMM in the loss space of the pretrained model. Then we use the detected OOD samples to train a new model by samplewise LS. An overview of our framework is in Fig. 3.

1) DAOOD Detection: The first part of the ReSmooth framework is to identify DAOOD samples from the augmented training set. Formally, we assume DAID samples are lying in the original data distribution \( P \), while DAOOD samples are not. In our framework, we choose to approximately estimate \( P \) in the loss space because DAOOD samples are samples recognized wrongly by a model learned on \( P \) and thus have large loss values. Notably, a trained model is often sensitive to the distribution of the training data [12]. Therefore, we first train a model \( f_D \) on unaugmented dataset \( D \) and then use \( f_D \) to acquire the loss distribution of \( D \cup D' \).

At the first glance, it is hard to distinguish what \( P \) is like for the reason that the loss distribution exhibits high skew to zero and is of a long tail. When we view loss values in the logarithm domain, however, the loss distribution is mostly like a Gaussian mixture distribution [40] (as shown in Fig. 4)

\[
\log \mathcal{L} \sim \pi_0 \mathcal{N}(\mu_0, \sigma_0) + \pi_1 \mathcal{N}(\mu_1, \sigma_1) \quad (2)
\]

where \( \mu_1, \sigma_1 \) is the mean and variance of the i-th Gaussian component \( \mathcal{N}(\mu_i, \sigma_i) \) and \( \pi_i \) is its mixing coefficient, satisfying \( \sum_i \pi_i = 1 \). Notably, \( \mathcal{N}(\mu_0, \sigma_0) \) corresponds to the original data distribution \( P \) in the loss space, which satisfies

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\( \mu_0 = \min_i(\mu_i) \). We estimate \( \mu, \sigma, \pi \) by the EM algorithm. Then, we can estimate the posterior probability \( w_i \) of a sample \( x_i \) belonging to \( N(\mu_0, \sigma_0) \) given its loss value \( \ell_i \):

\[
w_i \triangleq p(x_i \in \mathcal{P}|\ell_i) = \frac{\pi_0 p_N(\mu_0, \sigma_0)(\log \ell_i)}{\sum_{k=0}^{K} \pi_k p_N(\mu_k, \sigma_k)(\log \ell_i)}.
\]

We finish the detection stage by getting the posterior probabilities, where we can softly split samples (or explicitly split samples into \( D_{\text{ID}} \) and \( D_{\text{OOD}} \) by setting a hard threshold \( \tau \)). Note that the detection procedure is in an online way because the DA is applied randomly during training and even the effect of the same augmentation function differs for different samples.

2) DAOOD Utilization: In the ReSmooth stage, we adopt the original objective function for ID samples and slightly adjust the objective function for DAOOD samples. Specifically, we use LS [41] strategy for DAOOD samples. In this way, we do not change the marginal distribution \( p(x) \) while changing the joint distribution \( p(x, y) \). It has two properties: 1) keeping the diversity of DA and 2) not changing the separability in training.

Formally, we refer to \( q \) as the ground-truth one-hot label defined in \( D' \) and \( p \) as the model prediction. Given the smoothness parameter \( \alpha \) and the uniform distribution \( u \) over labels (satisfying \( u_k = 1/K \), where \( K \) is the number of classes), the target label \( q' \) meets: \( q' = (1-\alpha)q + \alpha u \). So the cross-entropy loss with LS is as follows:

\[
H(q', p) = (1-\alpha)H(q, p) + \alpha H(u, p)
\]

where \( H \) denotes the cross-entropy.

With posterior probabilities \( \{w_i\} \) derived from the detection stage, we can get the smooth parameter for each sample as \( a_i = a(1 - w_i) \). The soft split will give the sample not likely from the original data distribution a higher \( a_i \) and the sample in original data distribution a lower \( a_i \). Our final loss for training on \( D' \) by adopting samplewise LS strategy comes as follows:

\[
\mathcal{L}_{\text{div}} = \frac{1}{|D'|} \sum_{x_i \in D'} (1-a_i)H(q_i, p_i) + a_i H(u, p_i).
\]

The above loss is adopted for training with diverse DA. As for NDA, due to its OOD property at the beginning of design, we can ignore the detection procedure and simply utilize them in the ReSmooth framework. To be specific, we assume all the augmented samples are OOD and use a constant hyperparameter \( \alpha \) for all the augmented samples for efficiency as follows:

\[
\mathcal{L}_{\text{neg}} = \frac{1}{|D'_{\text{ID}}|} \sum_{x_i \in D'_{\text{ID}}} H(q_i, p_i)
+ \frac{1}{|D'_{\text{OOD}}|} \sum_{x_i \in D'_{\text{OOD}}} (1-\alpha)H(q_i, p_i) + \alpha H(u, p_i).
\]

The overall algorithm can be found in Algorithm 1.

IV. EXPERIMENTS

To experimentally evaluate our proposed ReSmooth strategy, we first test it on several public vision datasets. Next, we validate the effectiveness of some designs in our framework by ablation study. Finally, we conduct additional experiments to further address several potential concerns about DAOOD and DAID samples.

A. Experimental Settings

We evaluate our proposed method on four vision classification datasets: CIFAR-10 [42], CIFAR-100 [42], SVHN [43], and ImageNet datasets [44]. The CIFAR-10 dataset consists of 60K \( 32 \times 32 \) RGB images (6K images per class). The CIFAR-100 dataset is similar to CIFAR-10, except that it has 100 classes and 600 images per class. SVHN is a dataset of house numbers, which has a core training set of 73K images and 531K additional training images. ImageNet is a large visual dataset consisting of natural images in high resolution. We use the ImageNet-1000 subset of it, which has 1K classes and 1.3K training images and 50 validation images per class. In our experiments, we use the standard train split for the training but use the standard validation set in company with the DAOOD and DAID validation sets derived from it for testing in some experiments. The detailed description of DAOOD and DAID validation sets can be found in Section IV-B.
We report the results for both the jigsaw transformation and the rotation transformation. We compare our work with several baselines to demonstrate the effectiveness of our proposed LS strategy. The comparison baselines are briefly introduced as follows.

1) SA: A weak baseline. It includes random crop and random horizontal flip for CIFAR and SVHN, while for ImageNet it includes random resized crop, random horizontal flip and color jittering. “Unaugmented” data in this article refer to the data with SA.

2) AA [2]: A strong baseline with RandAugment.

3) RA [3]: A strong baseline with RandAugment.

4) RA With Label Smooth [41] (LSR+RA): A strong baseline with RA and adequate LS strength.

5) KDforAA [13]: A state-of-the-art method for handling diverse DA by virtue of the model pretrained on un-/weak-augmented data.

6) KDforRA [13]: Our reimplementation of [13] with RA (using the same hyperparameters).

B. DAOOD Dataset

Although the main purpose of this work is to improve the accuracy on the original test set when training with DA, we also collect the DAOOD dataset for potential further research like OOD generalization [47]. DAOOD samples are semantic-invariant under the meaning of recognition and should be generalized in robust model. Recall that DAOOD samples. We repeat this process until getting enough samples. The overall algorithm can be seen in Algorithm 2.

To be noticed, we do not use this algorithm for online detecting DAOOD samples in Section III-C mainly because of its higher computation and memory cost. We collect...
three datasets, named CIFAR-10-RA, CIFAR-100-RA, and ImageNet-RA, from CIFAR-10 [42], CIFAR-100 [42], and ImageNet [44], respectively. The algorithm can be used to collect from any dataset combined with any DA strategy. The DAOOD dataset also have train and test split collected from its counterpart in the referenced dataset. It is to be noticed that there is no guarantee that any sample in the referenced dataset has a counterpart in the DAOOD dataset. What is more, even if the Algorithm 2 can select DAOOD samples more accurately than Algorithm 1, the collected dataset is still noisy.

C. Classification Performance

1) Diverse DA: From Table I, we can see that our method consistently improves on the basic DA strategy RA w or w/o LSR. To be specific, in CIFAR-10 our method achieves approximately 0.2% improvement in comparison to the second-best, which is not trivial considering the baseline is relatively high. For CIFAR-100, the absolute improvement is 1.8% compared with RA baseline and 1.0% compared with the second-best method.

On the other hand, however, the improvement in SVHN and ImageNet is relatively limited and the reasons behind them are slightly different. For SVHN, digits recognition is a comparatively easy task and mainly solved by the recognition of shape. In that sense, DAOOD samples are rarely detected in SVHN and most of the detected samples are semantics missing samples. A similar phenomenon can also be observed in ImageNet, but there still exists a large proportion of DAOOD samples. We will discuss the detected OOD samples later in detail in Section IV-F2.

2) Negative DA: We also augment the training data with NDA and train the model with different baselines. From Table II, we can draw that using NDA in the usual way (w/ or w/o LSR) may not improve the generalization, especially for a small model. A smaller model has a poorer representative ability and may be hard to represent the data from two different distributions. With the ReSmooth strategy, however, the model tends to learn better in the original data distribution while benefiting from the NDA, and the performance is consistently boosted. In addition, the KD-based method is not proposed to handle DAOOD samples and its performance is inconsistent and not as good as ReSmooth in the NDA setting.

3) More Data Augmentation Strategies: We carry out some quick experiments on more DA strategies in this section to verify the consistent improvement of our proposed ReSmooth framework. We consider two well-known DA methods AA [2] and Cutout [1] and a recently proposed generation-based DA method TeachAugment [21]. Cutout augmentation randomly cut a patch out from the original image and the residual parts can be viewed as semantic-unchanging. We consider two cut sizes: 8 × 8 and 16 × 16. TeachAugment trained an augmentation network and a recognition network in an adversarial manner, where the training of augmentation network is constrained by a teacher model. We have experiments on CIFAR-100 with Resnet18, and the results can be found in Table IV.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Table III. We can see that ReSmooth can be easily extended to more augmentations and improve on them.

### D. Ablation Study

The only difference between our method and LSR+RA baseline in training is the LS parameters. In LSR+RA baseline, one LS parameter $\alpha$ is used for all the training samples, whereas in our method, different samples with different transformations are given different LS parameters based on the posterior probabilities estimated in the OOD detection stage: $a_i = \alpha(1 - w_i)$. To validate the effectiveness of the ReSmooth framework, we ablate the choice of the LS parameters. Specifically, we compare our detection-based LS parameters with the following methods.

1. **Uni_given**: Unified smoothness where the parameter is the same as the (max) smooth parameter of the ReSmooth framework.
2. **Uni_avg**: Unified smoothness where the parameter equals the average of the smooth parameters in the ReSmooth framework.
3. **Uni_optimal**: Unified smoothness where the parameter is optimal in a grid search step.
4. **Random Sampling**: Give every sample a random smooth parameter sampling from a uniform distribution $U(0, \alpha)$.
5. **Random Split**: Shuffle the smooth parameters derived from the OOD detection step.
6. **Reverse**: Give every sample a reversed smooth parameter: $a_i = \alpha w_i$.

On the other hand, we ablate the loss distribution estimation by replacing the log loss distribution with the normalized loss distribution. In this situation, the loss distribution is unimodal and the estimated GMM is arguably unimodal (because one Gaussian component is centered while the other Gaussian component is not and comparably negligible). Based on the posterior probability distribution of the centered Gaussian component, we follow the ReSmooth framework to train the new network. The results can be found in “RS_norm” column and “RS_log” column.

The results in Table V support that though smoothed labels can always benefit the performance, our methods are the best and improve on them with a large margin. Noticeably, samplewise strategies are not of consistent performance across different datasets except for ours. It suggests that samplewise smooth heavily relies on the smooth parameters and our parameters derived from OOD detection are helpful and appropriate for all datasets. What is more, the performance of ReSmooth combined with the normalized loss distribution is similar but inferior to the best one. It is because the estimated Gaussian distribution is centered, and thus the sample deviated from the center tends to be assigned a bigger smooth parameter. It has two disadvantages: 1) more ID-like samples are viewed as OOD samples and they are less distinguishable and 2) the training is attenuated because of the bigger LS parameters. As a result, the performance of RS_norm is suboptimal and the gap becomes even larger for bigger model.

### E. Parameter Analysis

Given a specific DA strategy, the only hyperparameter to tune in the ReSmooth framework is the smooth parameter $\alpha$. What is more, we observe that sometimes our ReSmooth framework performs better with larger augmentation probability $p$ compared with baselines. As a result, we perform two separate search of the optimal hyperparameters: $\alpha$ and $p$. At the beginning, we fixed the smooth parameter $\alpha$ and search $p$, and then we fix $p$ and search $\alpha$. Because the optimal smooth parameter and the optimal augmentation probability are influenced by both the dataset [48] and the augmentation strategy, we perform the search process independently per augmentation strategy per dataset as shown in Fig. 5. The results suggest that for diverse DA, bigger $p$ usually corresponds to better performance, while for NDA bigger $p$ causes worse performance in standard training. Compared with it, the ReSmooth framework can always ameliorate the performance and benefit from the larger augmentation probability.

Once $p$ is determined, we turn to find the optimal $\alpha$. In this section, we provide the results of both ReSmooth and the standard training scheme with LSR to illustrate the different appearances of them. $p$ for LSR is set the same as the optimal $p$ for standard training, because we find it performs empirically better under this setting. The results in Fig. 6 show that the ReSmooth framework is superior to the standard training scheme with LSR in a large range of $\alpha$. What is more, ReSmooth usually benefits from larger $\alpha$ compared with the LSR baseline. The above results and observations suggest that setting $\alpha \in [0.3, 0.5]$ is suitable for most of the cases across different datasets and augmentations. For the ImageNet results, we only search $\alpha$ with the same augmentation strategy as [3] and we tried 0.3, 0.4, and 0.5. The accuracies are 77.98, 78.09, and 78.27, respectively. For the WideResnet28-10 results, we experimented with the searched $p$ of Resnet18 and $\alpha$s (if exist) that result in similar performance with Resnet18. Almost all the optimal hyperparameters of Resnet18 are also optimal for WideResnet28-10 except that the optimal $\alpha$ of

| Dataset | Model | Uni_given | Uni_avg | Uni_optimal | R_Sampling | R_Split | Reverse | RS_norm | RS_log |
|---------|-------|-----------|---------|-------------|------------|---------|---------|---------|--------|
| CIFAR-10 | Res18 | 96.24±0.11 | 96.35±0.14 | 96.38±0.22 | 96.35±0.10 | 96.41±0.06 | 96.34±0.07 | 96.65±0.18 | 96.65±0.07 |
| Wide28-10 | 97.31±0.09 | 97.61±0.04 | 97.66±0.07 | 97.60±0.06 | 97.71±0.07 | 97.56±0.07 | 97.73±0.04 | 97.84±0.05 |
| CIFAR-100 | Res18 | 79.71±0.01 | 80.02±0.04 | 80.15±0.23 | 79.82±0.10 | 80.02±0.11 | 79.89±0.15 | 80.77±0.09 | 80.98±0.18 |
| Wide28-10 | 83.12±0.14 | 83.53±0.18 | 83.71±0.21 | 83.57±0.11 | 83.71±0.18 | 83.92±0.09 | 84.41±0.13 | 84.84±0.21 |
| SVHN | Res18 | 97.69±0.08 | 97.72±0.01 | 97.74±0.02 | 97.74±0.07 | 97.70±0.07 | 97.71±0.01 | 97.77±0.01 | 97.75±0.04 |
| Wide28-10 | 98.14±0.08 | 98.17±0.03 | 98.20±0.03 | 98.20±0.02 | 98.13±0.03 | 98.12±0.03 | 98.18±0.04 | 98.21±0.03 |
Fig. 5. Influence of the augmentation probability $p$ on different datasets with Resnet18. Left: Influence of $p$ on CIFAR-10 with NDA and diverse DA. Middle: Influence of $p$ on CIFAR-100 with NDA and diverse DA. Right: Influence of $p$ on SVHN with diverse DA.

Fig. 6. Influence of the smoothing parameter $\alpha$ on different datasets with Resnet18. Left: Influence of $\alpha$ on CIFAR-10 with NDA and diverse DA. Middle: Influence of $\alpha$ on CIFAR-100 with NDA and diverse DA. Right: Influence of $\alpha$ on SVHN with diverse DA.

Algorithm 3 Fair Comparison Training

Input: $f_D$, GMM, $D$, FLAG, maximal samples $n$
Output: $f_{D'}$
1: Randomly initialize $f_{D'}$
2: for epoch=1,2,\ldots,N do
3: Sample a mini-batch data $X_j$ from $D$
4: Split $X_j$ into $X_{ID}$ and $X_{OOD}$ given $f_D$ and GMM
5: $m = \min(n, |X_{ID}|, |X_{OOD}|)$
6: if FLAG is “DAOOD” then
7: Randomly get $m$ samples ($S_j$) from $X_{OOD}$
8: else if FLAG is “DAID” then
9: Randomly get $m$ samples ($S_j$) from $X_{ID}$
10: else if FLAG is “mixture” then
11: Randomly get $m$ samples ($S_j$) from $X_j$
12: end if
13: Compute loss $\mathcal{L}$ on $S_j$ with $f_D$ as in (1)
14: Update $f_{D'}$ with $\mathcal{L}$
15: end for
16: return $f_{D'}$

WideResnet28-10 for jigsaw on CIFAR-10 is 0.3, while $\alpha$ is 0.2 for Resnet18.

F. Potential Concerns

In this section, we will figure out some potential concerns about DAID and DAOOD samples empirically.

1) Are DAOOD Samples Needed for Training?: On one hand, DAOOD samples are samples out of the original data distribution, and learning from it may confuse the model, change the data manifold, and consequently divert the representation learning toward the DAOOD samples. On the other hand, either from the perspective of DA diversity or based on the intuition that DAOOD samples are informative and semantic-preserving, DAOOD samples may help train a model of better generalization. To verify the effectiveness of DAOOD during training, we propose to fairly compare training with DAID, training with DAOOD, and training with the mixture of them as usual. The main principle we are concerned with is to use the same number of samples of DAID, DAOOD and mix to train the models, respectively, as shown in Algorithm 3. The results can be found in Table IV.

From it, we can find that training with DAID samples is the easiest, and the trained model performs well in the DAID test set and original test set. It is expected because DAID samples are supposed to lie in the same distribution as the original data. On contrary, the model trained on DAOOD performs worse (but not too bad) in both the original test set and the DAID test set. The model trained on the mixture, though not performing best in DAID or DAOOD test set, achieves best test set accuracy. On top of that, our method ameliorates the training with the mixture and has a better performance in both the DAID and the original test set while incurring a negligible decrease in the DAOOD test set. Therefore, we conclude that DAOOD samples help the model generalization and our method can make better use of these DAOOD samples.

2) What Samples Are Detected?: To verify the effectiveness of the OOD detection module, we try to figure out what kinds of samples are detected (distinguished by a threshold $\tau$) and their corresponding proportions in this section. We show some examples of detected samples along with their predicted labels.
Fig. 7. Different types of detected OOD samples on (a) ImageNet, (b) CIFAR-100, and (c) SVHN training sets. For each pair, we show the image before RA (left) and the image after RA (right). Under each image is their corresponding label predicted by the pretrained model.

by the pretrained model and the samples before DA to better identify different types of them. These examples can be found in Fig. 7. Specifically, we have divided samples into seven categories: 1) DAOOD sample, as described in Section III-B; 2) low confidence sample, which is semantic-unchanging and is rightly predicted after augmentation but with a low confidence; 3) hard sample, which is semantic-unchanging but is wrongly predicted after augmentation; 4) semantics changing sample, where the object to be recognized remains in the image but semantic label changes after augmentation (e.g., a sea changes to a cloud in Fig. 7); 5) semantics missing sample, where the object misses after augmentation; 6) ambiguity, which is not recognizable for us; and 7) multi-object sample, where multiple objects exist and multiple labels are reasonable. In addition, we estimate their proportions by random sampling 100 samples. The results are shown in Fig. 8. We find that the situation differs between the CIFAR-100 dataset and ImageNet. In CIFAR-100, most of the OOD samples detected are DAOOD samples, while most of the remaining are predicted correctly with low confidence. In ImageNet, the sources of data are heterogeneous, including DAOOD samples, semantics changing samples, semantics missing samples, multiobject scene, hard samples, and ambiguity samples.

Intuition suggests that samples with low confidence, including both the low confidence samples (with right prediction) and hard samples (with wrong prediction), can be viewed as the DAOOD samples or the OOD samples in the original datasets (if samples before DA are with low confidences) to some extent. In this sense, we argue that our method is
suitable to handle both the DAOOD samples and samples with low confidence, but cannot do well in other situations such as missing semantics (e.g., noisy label or not cropping informative region) or multijob scene recognition. We think this is the main reason why the improvement is relatively limited in the SVHN and ImageNet datasets. On the other hand, the proportions may uncover the underlying reason why distillation-based methods can improve performance like [13]. The distillation-based methods can ameliorate training when the given label is noisy or insufficient. But on the contrary, the distillation-based methods are not good at handling DAOOD samples and samples with low confidence for the teacher model will provide wrong supervision on these samples. As a result, it is reasonable to find that our method performs better than [13] when DAOOD samples, low confidence samples, and hard samples are of a large proportion. In other word, though the claimed phenomenon in [13] can be observed by us, it is not the most case. Naturally, a fine-grained split of detected OOD is preferred to better facilitate the learning problem with diverse DA. Combining the predictions before and after augmentation may be helpful because DAOOD samples are predicted correctly before augmentation and wrongly after augmentation. But this method is time- and memory-consuming, so we leave the fine-grained split problem into future research.

V. CONCLUSION

In this article, we propose a ReSmooth framework to ameliorate the training with diverse DA. The key observation behind it is that a big part of the augmented samples are out of the original data distribution, denoted as DAOOD samples. We empirically show that the DAOOD samples are crucial for training a model of better generalization and our ReSmooth framework can better utilize DAOOD samples and further improve on regular training strategy. In addition, we also include the learning with NDA as part of work that only involves OOD utilizing. The classification results show that in most of the cases, our proposed framework can benefit the learning of both diverse DA and NDA.

REFERENCES

[1] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” 2017, arXiv:1708.04552.

[2] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning augmentation policies from data,” 2018, arXiv:1805.09501.

[3] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “RandAugment: Practical automated data augmentation with a reduced search space,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 702–707.

[4] X. Ren, W. Lin, X. Yang, X. Yu, and H. Gao, “Data augmentation in defect detection of sanitary ceramics in small and non-i.I.D datasets,” IEEE Trans. Neural Netw. Learn. Syst., early access, Mar. 9, 2022, doi: 10.1109/TNNLS.2022.3152245.

[5] X. Ji, A. Vedaldi, and J. Henriques, “Invariant information clustering for unsupervised image classification and segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9865–9874.

[6] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Prosmans, and L. Van Gool, “SCAN: Learning to classify images without labels,” in Proc. ECCV. Cham, Switzerland: Springer, 2020, pp. 268–285.

[7] Q. Ji, Y. Sun, J. Gao, Y. Hu, and B. Yin, “A decoder-free variational deep embedding for unsupervised clustering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 10, pp. 5681–5693, Oct. 2021.

[8] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, “Detectron,” 2018. [Online]. Available: https://github.com/facebookresearch/detectron

[9] H.-S. Fang, J. Sun, R. Wang, M. Gou, Y.-L. Li, and C. Lu, “InstaBoost: Boosting instance segmentation via probability map guided copy-pasting,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 682–691.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 25, Dec. 2012, pp. 1097–1105.

[11] A. J. Ratner, H. R. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, “Learning to compose domain-specific transformations for data augmentation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, p. 3239.

[12] R. Gontijo-Lopes, S. J. Smulkin, E. D. Cubuk, and E. Dyer, “Affinity and diversity: Quantifying mechanisms of data augmentation,” 2020, arXiv:2002.08973.

[13] L. Wei, A. Xiao, L. Xie, X. Zhang, X. Chen, and Q. Tian, “Circumventing outliers of autoaugment with knowledge distillation,” in Proc. ECCV. Cham, Switzerland: Springer, 2020, pp. 608–625.

[14] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, “KeepAugment: A simple information-preserving data augmentation approach,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 1055–1064.

[15] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” 2018, arXiv:1803.07728.

[16] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in Proc. ECCV. Cham, Switzerland: Springer, 2016, pp. 69–84.

[17] A. Sinha, K. Ayush, J. Song, B. Uzentk, H. Jin, and S. Ermon, “Negative data augmentation,” 2021, arXiv:2102.05113.

[18] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, “Population based augmentation: Efficient learning of augmentation policy schedules,” in Proc. ICML, 2019, pp. 2731–2741.

[19] D. Gadovskiy, L. Rizgazoe, S. Ishizaka, K. Kozuka, and S. Tsukizawa, “AutoDO: Robust AutoAugment for biased data with label noise via scalable probabilistic implicit differentiation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 16601–16610.
[20] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “AugMix: A simple data processing method to improve robustness and uncertainty,” 2019, arXiv:1912.02781.

[21] T. Suzuki, “TeachAugment: Data augmentation optimization using teacher knowledge,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 10904–10914.

[22] J. Lim, I. Kim, T. Kim, C. Kim, and S. Kim, “Fast AutoAugment,” in Proc. NIPS, vol. 32, 2019, pp. 6665–6675.

[23] R. Hataya, J. Zdenek, K. Yoshizoe, and H. Nakayama, “Faster AutoAugment: Learning augmentation strategies using backpropagation,” in Proc. ECCV. Cham, Switzerland: Springer, 2020, pp. 1–16.

[24] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized out-of-distribution detection,” in Proc. AAAI, vol. 34, 2020, pp. 12334–12341.

[25] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” 2016, arXiv:1610.02136.

[26] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in Proc. NeurIPS, vol. 31, 2018, pp. 1–11.

[27] W. Liu, X. Wang, J. Owens, and Y. Li, “Energy-based out-of-distribution detection,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 21464–21475.

[28] J. Li, R. Socher, and S. C. H. Hoi, “DivideMix: Learning with noisy labels as semi-supervised learning,” 2020, arXiv:2002.07394.

[29] F. Ariño, D. Ortego, P. Albert, N. O’Connor, and K. McGuinness, “Unsupervised label noise modeling and loss correction,” in Proc. ICML, 2019, pp. 312–321.

[30] H. Wei, L. Feng, X. Chen, and B. An, “Combating noisy labels by agreement: A joint training method with co-regularization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 13726–13735.

[31] B. Han et al., “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” 2018, arXiv:1804.06872.

[32] E. Malach and S. Shalev-Shwartz, “Decoupling ‘when to update’ from ‘how to update,’” 2017, arXiv:1706.02613.

[33] G. Pleiss, T. Zhang, E. R. Elsenberg, and K. Q. Weinberger, “Identifying mislabeled data using the area under the margin ranking,” 2020, arXiv:2001.10528.

[34] S. Hooker, A. Courville, G. Clark, Y. Dauphin, and A. Frome, “What do compressed deep neural networks forget?” 2019, arXiv:1911.05248.

[35] Z. Jiang, T. Chen, B. Mortazavi, and Z. Wang, “Self-damaging contrastive learning,” 2021, arXiv:2106.02990.

[36] P. Wen, Q. Xu, Y. Jiang, Z. Yang, Y. He, and Q. Huang, “Seeking the shape of sound: An adaptive framework for learning voice-face association,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 16347–16356.

[37] K. Sohn et al., “FixMatch: Simplifying semi-supervised learning with consistency and confidence,” 2020, arXiv:2001.07685.

[38] A. Merchant, B. Zoph, and E. D. Cubuk, “Does data augmentation benefit from split BatchNorms?” 2020, arXiv:2010.07810.

[39] M. Yi, L. Hou, L. Shang, X. Jiang, Q. Liu, and Z.-M. Ma, “Reweighting augmented samples by minimizing the maximal expected loss,” 2021, arXiv:2103.08933.

[40] H. Permuter, J. Francos, and I. Jermyn, “A study of Gaussian mixture models of color and texture features for image classification and segmentation,” Pattern Recognit., vol. 39, no. 4, pp. 695–706, 2006.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

[42] A. Kirzhnitsky, “Learning multiple layers of features from tiny images,” Univ. Toronto, Toronto, ON, Canada, 2009.

[43] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” in Proc. NeurIPS Workshop Deep Learn. Unsupervised Feature Learn., 2011.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–777.

[46] S. Zagoruyko and N. Komodakis, “Wide residual networks,” 2016, arXiv:1605.07144.

[47] J. Djolonga et al., “On robustness and transferability of convolutional neural networks,” in Proc. CVPR, 2021, pp. 16458–16468.

[48] R. Müller, S. Kornblith, and G. E. Hinton, “When does label smoothing help?” in Proc. NeurIPS, vol. 32, 2019, pp. 1–10.

Chenyang Wang received the B.Eng. degree from the School of Computer Science and Technology, Dalian University of Technology, Dalian, China, in 2015, and the M.Eng. degree from the Faculty of Computing, Harbin Institute of Technology, Harbin, China, in 2021, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and machine learning.

Junnj Jang (Senior Member, IEEE) received the B.S. degree from the Department of Mathematics, Huaqiao University, Quanzhou, China, in 2009, and the Ph.D. degree from the School of Computer, Wuhan University, Wuhan, China, in 2014. From 2015 to 2018, he was an Associate Professor with the China University of Geosciences, Wuhan. Since 2016, he has been a Project Researcher with the National Institute of Informatics, Tokyo, Japan. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His research interests include image processing and computer vision.

Dr. Jiang received the Finalist of the World’s FIRST 10K Best Paper Award at ICMC 2017, and the Best Student Paper Runner-up Award at MMM 2015. He also received the 2016 China Computer Federation (CCF) Outstanding Doctoral Dissertation Award and the 2015 ACM Wuhan Doctoral Dissertation Award.

Xiong Zhou received the B.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2019, where he is currently pursuing the Ph.D. degree in computer science.

His research interests include computer vision, image processing, and machine learning, especially learning with imperfect data. His research works have been published in International Conference on Machine Learning (ICML), International Conference on Learning Representations (ICLR), and IEEE/CVF International Conference on Computer Vision (ICCV).

Xianming Liu (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 2006, 2008, and 2012, respectively.

In 2011, he spent half a year at the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, as a Visiting Student, where he was a Post-Doctoral Fellow from 2012 to 2013. He was a Project Researcher with the National Institute of Informatics (NII), Tokyo, Japan, from 2014 to 2017. He is currently a Professor with the School of Computer Science and Technology, HIT. He has authored over 50 international conference and journal publications, including top IEEE journals, such as the IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (T-CSVT), IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (T-IFS), and IEEE TRANSACTIONS ON MULTIMEDIA (T-MM), and top conferences, such as International Conference on Machine Learning (ICML), International Conference on Learning Representations (ICLR), IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR), and IEEE/CVF International Conference on Computer Vision (ICCV).

Dr. Liu was a recipient of the IEEE ICME 2016 Best Student Paper Award.