IMBALANCED CLASSIFICATION VIA ADVERSARIAL MINORITY OVER-SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

In most real-world scenarios, training datasets are highly class-imbalanced, where deep neural networks suffer from generalizing to a balanced testing criterion. In this paper, we explore a novel yet simple way to alleviate this issue via synthesizing less-frequent classes with *adversarial examples* of other classes. Surprisingly, we found this counter-intuitive method can effectively learn generalizable features of minority classes by transferring and leveraging the diversity of the majority information. Our experimental results on various types of class-imbalanced datasets in image classification and natural language processing show that the proposed method not only improves the generalization of minority classes significantly compared to other re-sampling or re-weighting methods, but also surpasses other methods of state-of-art level for the class-imbalanced classification.

1 INTRODUCTION

Deep neural networks (DNNs) trained by large-scale datasets have enabled many breakthroughs in machine learning, especially in various classification tasks such as image classification (He et al., 2016a), object detection (Redmon & Farhadi, 2017), and speech recognition (Park et al., 2019). Here, a practical issue in this large-scale training regime, however, is at the difficulty in data acquisition process across labels, e.g. some labels are more abundant and easier to collect (Mahajan et al., 2018). This often leads a dataset to have "long-tailed" label distribution, as frequently found in modern real-world large-scale datasets. Such class-imbalanced datasets make the standard training of DNN harder to generalize (Wang et al., 2017; Ren et al., 2018; Dong et al., 2018), particularly if one requires a class-balanced performance metric for a practical reason.

A natural approach in attempt to bypass this *class-imbalance problem* is to re-balance the training objective artificially in class-wise with respect to their numbers of samples. Two of such methods are representative: (a) "re-weighting" the given loss function by a factor inversely proportional to the sample frequency in class-wise (Huang et al., 2016; Khan et al., 2017), and (b) "re-sampling" the given dataset so that the expected sampling distribution during training can be balanced, either by "over-sampling" the minority classes (Japkowicz, 2000; Cui et al., 2018) or "under-sampling" the majority classes (He & Garcia, 2008).

The methods on this line, however, usually result in harsh over-fitting to minority classes, since in essence, they cannot handle the lack of information on minority data. Several attempts have been made to alleviate this over-fitting issue: Cui et al. (2019) proposed the concept of "effective number" of samples as alternative weights in the re-weighting method. In the context of re-sampling, on the other hand, SMOTE (Chawla et al., 2002) is a widely-used variant of the over-sampling method that mitigates the over-fitting via data augmentation, but generally this direction has not been much explored recently. Cao et al. (2019) found that both re-weighting and re-sampling can be much more effective when applied at the later stage of training, in case of neural networks.

Another line of the research attempts to prevent the over-fitting with a new regularization scheme that minority classes are more regularized, where the margin-based approaches generally suit well as a form of data-dependent regularizer (Zhang et al., 2017; Dong et al., 2018; Khan et al., 2019; Cao et al., 2019). There have also been works that view the class-imbalance problem in the framework of active learning (Ertekin et al., 2007; Attenberg & Ertekin, 2013) or meta-learning (Wang et al., 2017; Ren et al., 2018; Shu et al., 2019; Liu et al., 2019).

Contribution. In this paper, we revisit the over-sampling framework and propose a new way of generating minority samples, coined Adversarial Minority Over-sampling (AMO). In contrast to other over-sampling methods, e.g. SMOTE (Chawla et al., 2002) that applies data augmentation to minority samples to mitigate the over-fitting issue, we attempt to generate minority samples in a completely different way: AMO does *not* use the existing minority samples for synthesis, but use *adversarial examples* (Szegedy et al., 2014; Goodfellow et al., 2015) of non-minority samples made from another, baseline classifier (potentially, over-fitted to minority classes) independently trained using the given imbalanced dataset. This motivation leads us to a very counter-intuitive method at a first glance: it results in labeling *minority class* on an adversarial example of a majority class at last. Our key finding is that, this method actually can be very effective on learning generalizable features in the imbalanced learning: it does not overly use the minority samples, and leverages the richer information of the majority samples simultaneously.

Our minority over-sampling method consists of three components to improve the sampling quality. First, we propose an optimization objective for generating synthetic samples, so that a majority input can be translated into a synthetic minority sample via optimizing it, while not affecting the performance of the majority class (even the sample is labeled to the minority class). Second, we design a sample rejection criteria based on the observation that generation from more majority class is more preferable. Third, based on the proposed rejection criteria, we suggest an optimal distribution for sampling the initial seed points of the generation.

We evaluate our method on various imbalanced classification problems, including synthetically imbalanced CIFAR-10/100 (Krizhevsky, 2009), and real-world imbalanced datasets including Twitter dataset (Gimpel et al., 2011) and Reuters dataset (Lewis et al., 2004) in natural language processing. Despite its simplicity, our method of adversarial minority over-sampling significantly improves the balanced test accuracy compared to previous re-sampling or re-weighting methods across all the tested datasets. These results even surpass the results from state-of-the-art margin-based method (LDAM; Cao et al. 2019). We also highlight that our method is fairly orthogonal to the regularization-based methods, by showing that joint training of our method with LDAM could further improve the balanced test accuracy as well.

Despite the great generalization ability of DNNs, they are known to be susceptible to adversarial examples, which makes it difficult to deploy them in real-world safety-critical applications (Szegedy et al., 2014; Goodfellow et al., 2015). The broad existence of adversarial examples in DNNs is still a mysterious phenomenon (Gilmer et al., 2019; Galloway et al., 2019; Ilyas et al., 2019), and we think our results can be of independent interest to shed new insight on understanding their property.

2 Adversarial Minority Over-sampling

We consider a classification problem with K classes from a dataset $D = \{(x_i, y_i)\}_{i=1}^N$, where $x \in \mathbb{R}^d$ and $y \in \{1, \ldots, K\}$ denote an input and the corresponding class label, respectively. Let $f : \mathbb{R}^d \to \mathbb{R}^K$ be a classifier designed to output K logits, which we want to train against the class-imbalanced dataset D. We denote $N := \sum_k N_k$ to be the total sample size of D, where N_k is that of class k. Without loss of generality, we assume $N_1 \geq N_2 \geq \cdots \geq N_K$. In the *class-imbalanced* classification, the class-conditional data distributions $P_k := p(x \mid y = k)$ are assumed to be invariant across training and test time, but they have different prior distributions, say $p_{\texttt{train}}(y)$ and $p_{\texttt{test}}(y)$, respectively: $p_{\texttt{train}}(y)$ is highly imbalanced while $p_{\texttt{test}}(y)$ is usually assumed to be the uniform distribution. The primary goal of the class-imbalanced learning is to train f from $D \sim P_{\texttt{train}}$ that generalizes well under $P_{\texttt{test}}$ compared to the standard training, e.g., empirical risk minimization (ERM) with an appropriate loss function $L(f)$:

$$\min_f \mathbb{E}_{(x,y) \sim D}[L(f; x, y)]. \tag{1}$$

Our method is primarily based on over-sampling technique (Japkowicz, 2000), a traditional and principled way to balance the class-imbalanced training objective via sampling minority classes more frequently. In other words, we assume a "virtually balanced" training dataset $D_{\texttt{bal}}$ made from D such that the class k has $N_1 - N_k$ more samples, and f is trained on $D_{\texttt{bal}}$ instead of D.

A key difficulty in over-sampling is to prevent *over-fitting* on minority classes, as the objective modified is essentially much biased to a few samples of minority classes. In contrast to prior work that focuses on applying data augmentation to minority samples to mitigate this issue (Chawla et al.,

2002; Liu et al., 2019), we attempt to synthesize minority samples in a completely different way: our method does *not* use the minority samples for synthesis, but use *adversarial examples* of non-minority samples made from another classifier $g : \mathsf{R}^d \ ! \ \mathsf{R}^K$ independently trained on D.

2.1 Overview of adversarial minority over-sampling

Consider a scenario of training a neural network f on a class-imbalanced dataset D. The proposed *Adversarial Minority Over-sampling* (AMO) attempts to construct a new balanced dataset $\mathsf{D}_{\mathtt{bal}}$ for training of f, by adding *adversarial examples* (Szegedy et al., 2014) of another classifier g. Here, we assume the classifier g is a pre-trained neural network on D so that performs well (at least) on the training imbalanced dataset, e.g., via standard ERM training. Therefore, g may be over-fitted to minority classes and perform badly under the balanced testing dataset. On the other hand, f is the target network we aim to train to perform well on the balanced testing criterion.

During the training f, AMO utilizes the classifier g to generate new minority samples, and the resulting samples are added to D to construct $\mathsf{D}_{\mathtt{bal}}$ on the fly. To obtain a single synthetic minority point x of class k, our method solves an optimization problem starting from another training sample x_0 of a (relatively) major class $k_0 < k$:

$$x \ = \underset{x:=x_0+\delta}{\arg\min} \ \mathsf{L}_{\mathtt{CE}}(g; x, k) + \lambda \ f_{k_0}(x), \tag{2}$$

where $\mathsf{L}_{\mathtt{CE}}$ denotes the standard cross entropy loss and $\lambda > 0$ is a hyperparameter. In other words, our method "translates" a seed point x_0 into x , so that g confidently classifies it as class k. It is not required for f to classifies x to k as well, but the optimization objective restricts that f to have lower confidence at the original class k_0. The generated sample x is then labeled to class k, and fed into f for training to perform better on $\mathsf{D}_{\mathtt{bal}}$. Here, the regularization term $\lambda \ f_{k_0}(x)$ on logit reduces the risk when x is labeled to k, whereas it may contain significant features of x_0 in the viewpoint of f. Intuitively, one can regard the overall process as teaching the minority classifiers of f to learn new features which g considers it significant, i.e., via extension of the decision boundary from the knowledge g. Figure 1 illustrates the basic idea of our method.

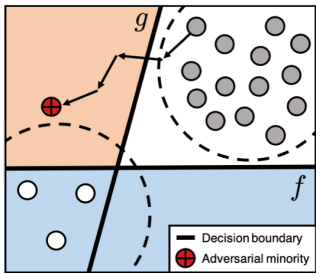


Figure 1: An illustration of AMO via solving (2).

One may understand our method better by considering the case when g is an "oracle" (possibly the Bayes optimal) classifier, e.g., (roughly) humans. Here, solving (2) essentially requires a transition of the original input x_0 of class k_0 with 100% confidence to another class k with respect to g: this would let g "erase and add" the features related to the class k_0 and k, respectively. Hence, in this case, our process corresponds to collecting more in-distribution minority data, which may be argued as the best way one could do to resolve the class-imbalance problem.

An intriguing point here is, however, that neural network models are very far from this ideal behavior, even for that achieves super-human performance. Instead, when f and g are neural networks, (2) often finds x at very close to x_0, i.e., similar to the phenomenon of *adversarial examples* (Szegedy et al., 2014; Goodfellow et al., 2015). Nevertheless, we found our method still effectively improves the generalization of minority classes even in such cases. This observation is, in some sense, aligned to a recent claim that adversarial perturbation is not a "bug" in neural networks, but a "generalizable" feature (Ilyas et al., 2019).

2.2 Detailed components of AMO

Sample rejection criteria. An important factor that affects the quality of the synthetic minority samples in our method is the quality of g, especially for g_{k_0}: a better g_{k_0} would more effectively "erase" important features of x_0 during the generation, thereby making the resulting minority samples more reliable. In practice, however, g is not that perfect so the synthetic samples still contain some discriminative features of the original class k_0, in which it may even harm the performance of f. This risk of "unreliable" generation becomes more harsh when N_{k_0} is small, as we assume that g is also trained on the given imbalanced data D.

Algorithm 1 Adversarial Minority Over-sampling (AMO)

Input: A dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ with $N = \sum_{k=1}^{K} N_k$. A receiving classifier f. A generating classifier g. $\lambda, \gamma > 0$ and $\beta \in [0, 1)$.
Output: A class-balanced dataset D_{bal}

Initialize $D_{\text{bal}} \leftarrow D$
for $k = 2$ **to** K **do**
  $\Delta \leftarrow N_1 - N_k$
  for $i = 1$ **to** Δ **do**
    $k_0 \sim P(k_0|k) \propto 1 - \beta^{(N_{k_0} - N_k)^+}$
    $x_0 \leftarrow$ A randomly-chosen sample of class k_0 in D
    $x \leftarrow x_0 + \delta \cdot \arg\min_{x: x = x_0 + \delta} L(g; x, k) + \lambda \cdot f_{k_0}(x)$
    $R \sim \text{Bernoulli}(\beta^{(N_{k_0} - N_k)^+})$
    if $L(g; x, k) > \gamma$ **or** $R = 1$ **then**
      $x \leftarrow$ A randomly-chosen sample of class k in D
    end if
    $D_{\text{bal}} \leftarrow D_{\text{bal}} \cup \{(x, k)\}$
  end for
end for

To alleviate this risk, we consider a simple criteria for *rejecting* each of the synthetic samples randomly with probability depending on k_0 and k:

$$\mathsf{P}(\text{Reject } x \text{ as } k_0, k) := \beta^{(N_{k_0} - N_k)^+}, \tag{3}$$

where $(\cdot)^+ := \max(\cdot, 0)$, and $\beta \in [0, 1)$ is a hyperparameter which controls the reliability of g: the smaller β, the more reliable g. For example, if $\beta = 0.999$, the synthetic samples are accepted with probability more than 99% if $N_{k_0} - N_k > 4602$. When $\beta = 0.9999$, on the other hand, it requires $N_{k_0} - N_k > 46049$ to achieve the same goal. This exponential modeling of rejection probability is motivated by the *effective number* of samples (Cui et al., 2019), a heuristic recently proposed to model the observation that the impact of adding a single data point exponentially decreases at larger datasets. When a synthetic sample is rejected, we simply replace it with another minority point over-sampled from the original D to maintain the loss balance.

Optimal seed-point sampling. Another design choice of our method is *how to choose* an initial seed point x_0 for each generation in (2). This is important since it also affects the final quality of the generation, as the choice of x_0 corresponds to the sampling distribution of k_0. Based on the rejection policy proposed in (3), we design a sampling distribution for selecting the class of initial point x_0 given target class k, namely $Q(k_0|k)$, considering two aspects: (a) Q maximizes the *acceptance rate* under our rejection policy, and at the same time (b) Q chooses *diverse* classes as much as possible, i.e., the entropy $H(Q)$ is maximized. In our over-sampling scenario, i.e., the marginal sampling distribution is uniform in class-wise, these objectives lead Q to be equal to the distribution $P(k_0|k)$ such that each class is sampled proportional to its *acceptance rate*:

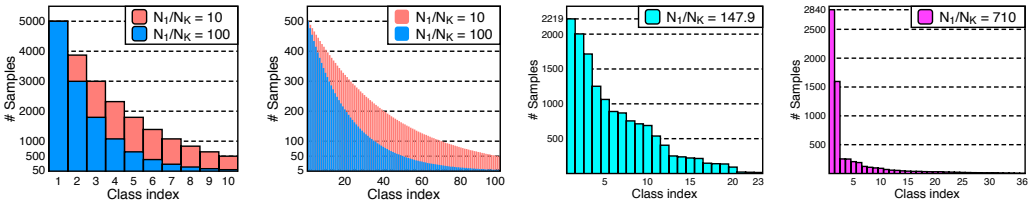
$$P(k_0|k) \propto 1 - \beta^{(N_{k_0} - N_k)^+}, \tag{4}$$

as it maximizes a joint objective of (a) and (b) above, which turns out to be equivalent to the KL-divergence of P and Q when (a) is formulated to $\mathsf{E}_Q[\log P]$, i.e., the expected value of the log-probability of P:

$$\max_Q \Big[\underbrace{\mathsf{E}_Q[\log P]}_{\text{(a)}} + \underbrace{H(Q)}_{\text{(b)}} \Big] = \min_Q \Big[H(Q, P) - H(Q) \Big] = \min_Q D_{\mathsf{KL}}(Q \| P), \tag{5}$$

where $D_{\mathsf{KL}}(\cdot \| \cdot)$ denotes the KL-divergence. Therefore, we use (4) to sample a seed point for each generation, as the sample-wise re-weighting factor with respect to its class and the given target minority class.

Practical implementation via re-sampling. In practice of training a neural network f, e.g., stochastic gradient descent (SGD) with mini-batch sampling, AMO is implemented using batch-wise re-sampling: more precisely, in order to simulate the generation of $N_1 - N_k$ samples for the class k,



(a) CIFAR-10 (b) CIFAR-100 (c) Twitter (d) Reuters

Figure 2: An illustration of histograms on training sample sizes for the datasets used in this paper.

we first obtain a *balanced* mini-batch $B = f(x_i, y_i)g_{i=1}^{m}$ via standard re-sampling, and randomly select the indices i to perform the generation with probability $\frac{N_1}{N_1} \frac{N_{y_i}}{} = 1 \quad N_{y_i}/N_1$. The generation is only performed for the selected indices, where each y_i acts as the target class k. For a single generation, we select a seed image x_0 inside the given mini-batch following (4): we found sampling seed images per each mini-batch does not degrades the effectiveness of AMO. Starting from the selected x_0, we solve the optimization (2) by performing gradient descent for a fixed number of iterations T. We only accept the result sample x only if $L(g; x \ , k)$ is less than $\gamma > 0$ for stability. The overall procedure of AMO is summarized in Algorithm 1.

3 Experiments

We evaluate our method on various class-imbalanced classification tasks in visual recognition and natural language processing: synthetically-imbalanced CIFAR-10/100 (Krizhevsky, 2009), Twitter (Gimpel et al., 2011), and Reuters (Lewis et al., 2004) datasets. Figure 2 illustrates the class-wise sample distributions for each dataset considered in our experiments. In overall, our results clearly demonstrate that minority synthesis via adversarial examples consistently improves the efficiency of over-sampling, in terms of the significant improvement of the generalization in minority classes compared to other re-sampling baselines, across all the tested datasets. We also perform an ablation study to verify the effectiveness of our main ideas. Throughout this section, we divide the classes in a given dataset into "majority" and "minority" classes, so that the majority classes consist of top-k frequent classes with respect to the training sample size where k is the minimum number that $\sum_k N_k$ exceeds 50% of the total. We denote the minority classes as the remaining classes.

3.1 Experimental setup

Baseline methods. We consider a wide range of baseline methods, as listed in what follows: (*a*) empirical risk minimization (ERM): training on standard loss without any re-balancing; (*b*) re-sampling (RS; Japkowicz 2000): balancing the objective from different sampling probability for each sample; (*c*) SMOTE (Chawla et al., 2002): a variant of re-sampling with data augmentation; (*d*) re-weighting (RW; Huang et al. 2016): balancing the objective from different weights on sample-wise loss; (*e*) class-balanced re-weighting (CB-RW; Cui et al. 2019): a variant of re-weighting that uses the inverse of effective number for each class, defined as $(1 \quad \beta^{N_k})/(1 \quad \beta)$. Here, we use $\beta = 0.9999$; (*f*) deferred re-sampling (DRS; Cui et al. 2019): re-sampling is deferred until the later stage of the training; (*g*) focal loss (Focal; Lin et al. 2017): the objective is up-weighted for relatively hard examples to focus more on the minority; (*h*) label-distribution-aware margin (LDAM; Lin et al. 2017): the classifier is trained to impose larger margin to minority classes. Roughly, the considered baselines can be classified into three categories: (i) "re-sampling" based methods - (*b, c, f*), (ii) "re-weighting" based methods - (*d, e*), and (iii) different loss functions - (*a, g, h*).

Training details. We train every model via stochastic gradient descent (SGD) with momentum of weight 0.9. For CIFAR-10/100 datasets, we train ResNet-32 (He et al., 2016b) for 200 epochs with mini-batch size 128, and set a weight decay of $2 \quad 10^{4}$. We follow the learning rate schedule used by Cui et al. (2019) for fair comparison: the initial learning rate is set to 0.1, and we decay it by a factor of 100 at 160-th and 180-th epoch. Although it did not affect much to our method, we also adopt the linear warm-up strategy on the learning rate (Goyal et al., 2017) in the first 5 epochs, as some of the baseline methods, e.g. re-weighting, highly depend on this strategy. For Twitter and Reuters datasets, on the other hand, we train a 2-layer fully connected network for 15 epochs with

Table 1: Comparison of test accuracy on the two long-tailed CIFAR-10 datasets. The number of majority and minority classes are reported in parentheses. All the values and error bars are mean and standard deviation across 3 trials upon randomly chosen seeds, respectively.

CIFAR-LT-10		$N_1/N_K = 100$			$N_1/N_K = 10$		
Loss	Re-balancing	Major (2)	Minor (8)	Average	Major (3)	Minor (7)	Average
ERM	-	95.3 1.34	62.1 1.82	68.7 1.43	93.7 0.65	82.8 0.73	86.0 0.69
ERM	RS	92.8 1.50	64.8 1.18	70.4 1.15	92.3 0.60	84.1 0.28	86.6 0.37
ERM	SMOTE	91.2 1.17	66.6 0.90	71.5 0.57	92.1 0.37	83.0 0.49	85.7 0.25
ERM	RW	91.4 1.66	68.2 0.34	72.8 0.33	91.7 0.48	84.5 0.26	86.6 0.18
ERM	CB-RW	90.2 3.34	66.4 1.77	71.2 1.14	92.1 0.05	84.6 0.72	86.8 0.49
ERM	DRS	97.3 0.35	69.7 0.29	75.2 0.26	92.4 0.36	84.9 0.53	87.1 0.26
ERM	**AMO (ours)**	93.3 0.85	**74.6** 0.34	**78.3** 0.16	92.3 0.36	**86.0** 0.39	**87.9** 0.21
Focal	-	95.4 2.23	61.5 1.64	68.3 1.19	93.2 0.13	82.0 0.70	85.3 0.47
LDAM	-	97.9 0.10	66.6 0.47	72.8 0.37	93.1 0.05	83.2 0.17	86.2 0.12
LDAM	DRW	96.1 0.75	72.4 0.52	77.1 0.49	91.6 0.71	85.2 0.21	87.1 0.28
LDAM	**AMO (ours)**	95.3 0.31	**75.1** 0.17	**79.1** 0.19	91.3 0.44	**86.0** 0.04	**87.5** 0.15

Table 2: Comparison of test accuracy on the two long-tailed CIFAR-100 datasets. The number of majority and minority classes are reported in parentheses. All the values and error bars are mean and standard deviation across 3 trials upon randomly chosen seeds, respectively.

CIFAR-LT-100		$N_1/N_K = 100$			$N_1/N_K = 10$		
Loss	Re-balancing	Major (15)	Minor (85)	Average	Major (26)	Minor (74)	Average
ERM	-	70.8 1.43	31.3 1.11	37.2 1.12	72.3 0.65	50.5 0.73	56.2 0.69
ERM	RS	59.6 2.10	26.7 1.21	31.6 1.26	66.8 0.61	50.6 0.57	54.8 0.47
ERM	SMOTE	61.7 0.09	29.1 0.41	34.0 0.33	66.7 1.25	49.3 0.81	53.8 0.93
ERM	RW	50.2 2.83	27.1 0.45	30.1 0.59	67.8 0.54	51.8 0.30	56.0 0.35
ERM	CB-RW	71.6 1.42	32.8 0.45	38.6 0.46	68.2 0.49	51.6 0.48	55.9 0.24
ERM	DRS	67.6 0.95	36.9 0.40	41.5 0.21	68.6 0.72	53.9 0.30	57.7 0.40
ERM	**AMO (ours)**	65.0 0.24	**39.0** 0.10	**42.9** 0.16	67.4 0.61	**55.0** 0.33	**58.2** 0.08
Focal	-	70.0 2.12	32.0 1.42	37.7 1.38	71.7 0.23	49.5 0.52	55.3 0.42
LDAM	-	73.5 0.71	33.5 0.88	39.5 0.69	73.5 0.30	48.1 0.30	54.7 0.16
LDAM	DRW	70.2 0.73	37.2 0.22	42.1 0.09	70.0 0.39	52.3 0.18	56.9 0.15
LDAM	**AMO (ours)**	67.0 0.93	**39.3** 0.15	**43.5** 0.22	70.3 0.43	**53.2** 0.10	**57.6** 0.14

mini-batch 64, with a weight decay of 5 10 5. The initial learning rate is also set to 0.1, but we decay it by a factor of 10 at 10-th epoch.

Details on AMO. When our method is applied in the experiments, we use another classifier g of the same architecture that is pre-trained on the given (imbalanced) dataset via standard ERM training. Also, in a similar manner to that of Cao et al. (2019), we use the deferred scheduling to our method, i.e., we start to apply our method after the standard ERM training of 160 epochs. We choose hyperparameters in our method from a fixed set of candidates, namely β 2 f0.99, 0.999g, λ 2 f0.1, 0.5g and γ 2 f0.9, 0.99g, based on its validation accuracy.

3.2 Long-tailed CIFAR datasets

CIFAR-10/100 datasets (Krizhevsky, 2009) consist of 60,000 images of size 32 32, 50,000 for training and 10,000 for test. Although the original datasets are balanced across 10 and 100 classes, respectively, we consider some "long-tailed" variants of CIFAR-10/100 (CIFAR-LT-10/100), in order to evaluate our method on various levels of imbalance. To simulate the long-tailed distribution frequently appeared in imbalanced datasets, we control the *imbalance ratio* $\rho > 1$ and artificially reduce the training sample sizes of each class except the first class, so that: (a) N_1/N_K equals to ρ, and (b) N_k in between N_1 and N_K follows an exponential decay across k. We keep the test dataset unchanged during this process, thereby the evaluation can be done in the balanced setting.

Table 3: Comparison of test accuracy on the two naturally imbalanced NLP datasets: Twitter and Reuters. The number of majority and minority classes are reported in parentheses. All the values and error bars are mean and standard deviation across 3 trials upon randomly chosen seeds, respectively.

Real-world		Twitter (N_1/N_K 148)			Reuters ($N_1/N_K = 710$)		
Loss	Re-balancing	Major (5)	Minor (18)	Average	Major (2)	Minor (34)	Average
ERM	-	92.5 0.25	69.7 0.66	74.7 0.46	97.6 0.07	57.6 1.23	59.8 1.17
ERM	RS	87.8 2.19	72.5 0.94	75.8 0.30	97.2 0.41	61.3 0.98	63.3 0.90
ERM	SMOTE	91.2 0.35	65.7 0.89	70.7 0.67	97.2 0.46	60.4 1.36	62.5 1.30
ERM	RW	84.5 1.62	73.9 0.80	76.2 0.95	95.3 1.60	63.2 1.17	65.0 1.08
ERM	CB-RW	88.5 0.53	74.4 0.61	77.5 0.40	97.4 0.24	62.9 0.47	64.8 0.45
ERM	DRS	90.9 0.40	74.1 1.20	77.8 0.85	97.6 0.09	60.3 0.41	62.4 0.39
ERM	**AMO (ours)**	90.8 0.72	**74.7** 0.37	**78.2** 0.35	97.3 0.18	**64.5** 0.45	**66.3** 0.42
Focal	-	82.3 0.63	72.0 2.91	74.2 2.35	97.8 0.11	57.1 0.43	59.4 0.42
LDAM	-	92.1 0.35	69.7 1.53	74.6 0.40	97.7 0.43	61.0 1.46	63.0 1.36
LDAM	DRW	91.0 0.42	74.4 1.08	78.0 0.87	97.2 0.27	62.2 0.32	64.1 0.31
LDAM	**AMO (ours)**	90.5 0.47	**75.6** 0.28	**78.8** 0.21	96.3 0.46	**68.5** 0.71	**70.0** 0.68

We compare the (balanced) test accuracy of various training methods (including ours) on CIFAR-LT-10 and 100, considering two imbalance ratios ρ 2 f100, 10g for each (See Figure 2(a) and 2(b) for an illustration of the sample distribution). For all the tested methods, we also report the test accuracies computed only on major and minor classes, to identify the relative impacts of each method on the major and minor classes, respectively.

Table 1 and 2 summarize the main results. In overall, the results show that our method consistently improves the test accuracy by a large margin, across all the tested baselines. For example, in the case when $N_1/N_K = 100$ on CIFAR-10, our adversarial minority over-sampling method applied on the baseline ERM improves the test accuracy by 14.0% in the relative gain. This result even surpasses the "LDAM+DRW" baseline (Cao et al., 2019), which is known to be a state-of-the-art to the best of our knowledge. Moreover, we point out, in most cases, our method could further improve the overall test accuracy when applied upon the LDAM training scheme (see "LDAM+AMO"): this indicates that the accuracy gain from our method is fairly orthogonal to that of LDAM, i.e., the margin-based approach, which suggests a new promising direction of improving the generalization when a neural network suffers from a problem of small data.

3.3 Real-world imbalanced datasets

Next, we further verify the effectiveness of AMO on *real-world* imbalanced dataset, especially focusing on two natural language processing (NLP) tasks: Twitter (Gimpel et al., 2011) and Reuters (Lewis et al., 2004) datasets. Twitter dataset is for a part-of-speech (POS) tagging task. There are 14,614 training examples with 23 classes, and the imbalance ratio, i.e., N_1/N_k, naturally made is about 150 (see Figure 2(c) for the details). Reuters dataset, on the other hand, is for a text categorization task which is originally composed of 52 classes. For a reliable evaluation, we discarded the classes that have less than 5 test examples, and obtained a subset of the full dataset of 36 classes with 6436 training samples. Nevertheless, the distribution of the resulting dataset is still extremely imbalanced, e.g. $N_1/N_k = 710$ (see Figure 2(d) for the details). Unlike CIFAR-10/100, we found that the two datasets have imbalance issue even in the test samples. Therefore, we report the averaged value of the class-wise accuracy instead of the standard test accuracy.

Table 3 demonstrates the results. Again, AMO performed best amongst other baseline methods, demonstrating a wider applicability of our algorithm beyond image classification. Remarkably, the results on Reuters dataset suggest our method can be even more effective under regime of *extremely* imbalanced datasets, as the Reuters dataset has much larger imbalance ratio than the others.

3.4 Ablation Study

We conduct an ablation study on the proposed method, investigating the detailed analysis on it. All the experiments throughout this study are performed with ResNet-32 models, trained on CIFAR-LT-10 with the imbalance ratio 100.

(a) ERM (b) SMOTE (c) LDAM+DRW (d) AMO (ours)

Figure 3: Visualization of t-SNE embeddings of the penultimate features computed from a balanced subset of training samples in CIFAR-LT-10 with ResNet-32 on different methods.

Methods	Major (2)	Minor (8)	Average
AMO	93.3 0.85	74.6 0.34	78.3 0.16
AMO (= 0)	92.8 0.97	73.0 0.10	76.9 0.15
AMO-Clean	78.4 2.45	72.7 0.60	73.5 0.81

Table 4: Comparison of test accuracy across various types of ablation methods. All the values and error bars are mean and standard deviation across 3 trials upon randomly chosen seeds, respectively.

Figure 4: An illustration of a synthetic sample generated by AMO. The noise image is ampli ed by 10 for better visibility.

The use of adversarial examples. The most intriguing component that consists our method would be the use of "adversarial examples", i.e., to label an adversarial example of majority class to a minority class, e.g. as illustrated in Figure 4. To understand more on how the adversarial perturbations affect our method, we consider a simple ablation, which we call "AMO-Clean": recall that our algorithm synthesizes a minority sample x from a seed image x_0. Instead of using x, this ablation uses the "clean" initial point x_0 as the synthesized minority when accepted. Under the identical training setup, we notice a signi cant reduction in the overall accuracy of AMO-Clean compared to the original AMO (see Table 4). This observation reveals that the adversarial perturbations ablated are extremely crucial to make our algorithm to work, regardless of how the noise is small.

The effect of . In the optimization objective (2) for the synthesis in AMO, we impose a regularization term $f_{k_0}(x)$ to improve the quality of synthetic samples as they might confuse if it still contains important features of the original class in a viewpoint of . To verify the effect of this term, we consider an ablation that is set to 0, and compare the performance to the original method. As reported in Table 4, we found a certain level of degradation in test accuracy at this ablation, which shows the effectiveness of the proposed regularization.

Comparison of t-SNE embeddings. To further validate the effectiveness of our method, we visualize and compare the penultimate features learned from various training methods (including ours) using t-SNE (Maaten & Hinton, 2008). Each embedding is computed from a randomly-chosen subset of training samples in the CIFAR-LT-10 (= 100), so that it consists of 50 samples per each class. Figure 3 illustrates the results, and shows that the embedding from our training method (AMO) is of much separable features compared to other methods: one could successfully distinguish each cluster under the AMO embedding (even though they are from minority classes), while others have some obscure region.

4 Conclusion

We propose a new over-sampling method for imbalanced classi cation, called Advserarial Minority Over-sampling (AMO). The problems we explored in this paper lead us to an essential question that whether an adversarial perturbation could be a good feature. Our ndings suggest that it could be at least to improve imbalanced learning, where the minority classes suffer over- tting due to insuf cient data. We believe our method could open a new direction of research both in imbalanced learning and adversarial examples.

REFERENCES

Josh Attenberg and Seyda Ertekin. Class imbalance and active learning. *Imbalanced Learning: Foundations, Algorithms, and Applications*, 2013.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *JAIR*, 2002.

Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, 2018.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE PAMI*, 2018.

Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *CIKM*, 2007.

Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019.

Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In *International Conference on Machine Learning*, pp. 2280–2289, 2019.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *ACL*, 2011.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *TKDE*, 2008.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016b.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *ICAI*, 2000.

Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *CVPR*, 2019.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *TNNLS*, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *JMLR*, 2004.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *CVPR*, 2017.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 2008.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

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

Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, 2017.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, 2017.

Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *CVPR*, 2017.