Breadcrumbs: Adversarial Class-Balanced Sampling for Long-tailed Recognition

Bo Liu[1], Haoxiang Li[1], Hao Kang[1], Gang Hua[1], and Nuno Vasconcelos[2]

[1] Wormpex AI Research
{richardboliu,lhxustcer,haokheseri,ganghua}@gmail.com
[2] University of California San Diego
nuno@ece.ucsd.edu

Abstract. The problem of long-tailed recognition, where the number of examples per class is highly unbalanced, is considered. While training with class-balanced sampling has been shown effective for this problem, it is known to over-fit to few-shot classes. It is hypothesized that this is due to the repeated sampling of examples and can be addressed by feature space augmentation. A new feature augmentation strategy, EMANATE, based on back-tracking of features across epochs during training, is proposed. It is shown that, unlike class-balanced sampling, this is an adversarial augmentation strategy. A new sampling procedure, Breadcrumb, is then introduced to implement adversarial class-balanced sampling without extra computation. Experiments on three popular long-tailed recognition datasets show that Breadcrumb training produces classifiers that outperform existing solutions to the problem. Code: https://github.com/BoLiu-SVCL/Breadcrumbs

1 Introduction

The availability of large-scale datasets, with many images per class [4], has been a major factor in the success of deep learning for computer vision. However, these datasets are manually curated and artificially balanced. This is unlike most real world applications, where the frequencies of examples from different classes can be highly unbalanced, leading to skewed distributions with long tails. These datasets are composed by a few popular classes and many rare classes. This class imbalance has been observed in image classification [31], face identification [13, 18], object detection [15, 40], and many other applications. Researchers have tackled it from various angles, including zero-shot learning [5, 33, 34], few-shot learning [29, 25, 6], and more recently long-tailed recognition [19].

In this work, we focus on the long-tailed recognition setting, where classes are grouped into three types that differ in training sample cardinality: many-shot (> 100 samples), medium-shot (between 20 and 100 samples), and few-shot (≤ 20 samples). Performance is evaluated over each group independently, in addition to the overall classification accuracy. While training data is highly unbalanced, the test set is kept balanced so that equally good performance on all classes is a requisite for high accuracy. One of the insights from the long-tailed

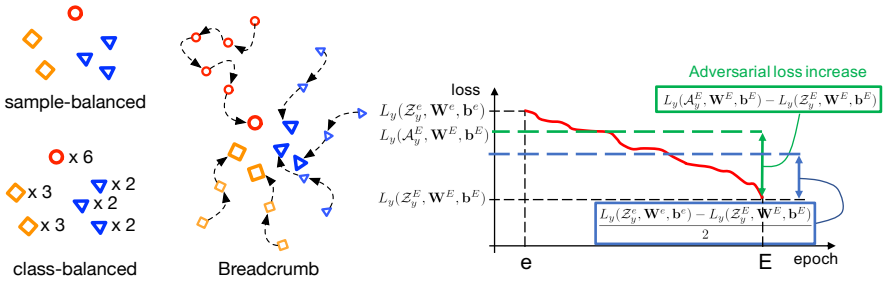

Fig. 1. (a)Upper-Left: Random sampling is sample-balanced. The number of examples per class has a long-tailed distribution. This leads to under-fitting in few-shot classes. Lower-Left: Class-balanced sampling duplicates few-shot samples in feature space and can leads to over-fitting for these classes. Right: Breadcrumb produces trails of features by back-tracking through training epochs. This is shown to be an adversarial augmentation technique, which mitigates the over-fitting problem. (b) Adversarial nature of EMANATE. The loss increase, between two epochs, due to feature augmentation by EMANATE is never smaller than half of the training gain (loss decrease) between them.

recognition literature is that techniques targeting specific dataset limitations, e.g. few-shot learning by data augmentation [2, 30], predicting classifier weights [22], prototype-based non-parametric classifiers [25], and optimization with second derivatives [6], are frequently harmful to classes that do not suffer from those limitations, e.g. many-shot. Hence, it is important to address the problem holistically, considering all types of classes simultaneously.

Since long-tailed recognition datasets have a continuous coverage of the number of samples per class, they are best addressed by training a model on the entire dataset, in a way robust to data imbalance. Standard classifier training follows the *sample-balanced* sampling setting of Figure 1(a). This consists of sampling images uniformly to create batches for network training. In result, as shown in the figure, few-shot classes (red) are under-represented and many-shot classes (blue) are over-represented in each batch. Hence, learning typically *under-fits* less populated classes. This has motivated procedures to fight class imbalance with data re-sampling [37] or cost-sensitive losses [15] that place more training emphasis on examples of lower populated classes. One of the more successful approaches is to decouple the training of feature embedding and classifier [14]. While the embedding is learned with image-balanced training, the classifier is trained with *class-balanced* sampling. As illustrated in Figure 1(a), this consists of sampling classes uniformly and then sampling uniformly within the class. However, for few shot classes, this approach leads to repeated sampling of the same examples. In result, the classifier can easily *over-fit* on few-shot classes.

In this work, we adopt the decoupled training strategy but seek to avoid over-fitting in the classifier training stage. For this, we propose to enrich the training data in the feature space at the output of the embedding, without extra

computation. The idea is to back-track features to access the large diversity of feature vectors that are available per training image in prior epochs. This can be exploited to generate more diverse training data than simply replicating existing features. We refer to this procedure as *feature back-tracking*. As shown in Figure 1(a), it allows the sampling of large numbers of feature vectors from the few-shot classes without duplication. Since the embedding changes across training epochs, an alignment is necessary to simplify network training. We show that a simple alignment of class means suffices to accomplish this goal and propose the *fEature augMentAtioN by bAck-tracking wiTh alignmEnt* (EMANATE) procedure. This consists of augmenting the feature vectors collected at an epoch with aligned replicas of the vectors that emanate from them by back-tracking.

A theoretical analysis shows that, unlike class-balanced sampling, EMANATE is an adversarial feature augmentation technique, in the sense that it is guaranteed to increase the training loss for any convergent training scheme. This places EMANATE in the realm of feature augmentation methods popular in the few-shot literature [2, 30]. However, these require extra computation to generate new examples and sometimes introduce convergence problems. EMANATE requires no extra computation and can be applied differently to each class, according to its number of samples. For classes with enough samples, only features from the last epoch are used, i.e. no resampling is performed. For those without, features are back-tracked over previous epochs, until there are enough features. This results in a new training feature set of higher variance for few-shot classes but forces no change on many-shot classes. In result, it is possible to improve classification accuracy for the former without degrading performance for the latter.

A new sampling scheme, denoted *Breadcrumb Sampling* is then proposed to leverage the feature trails extracted by EMANATE, in the context of the two-stage training of class-balanced sampling. Breadcrumb Sampling relies on EMANATE to collect these feature trails in a first stage, when the embedding is trained with image-balanced sampling. In the second stage, the classifier is then learned with class-balanced training based on these trails. Two sampling variants are considered. Weak Breadcrumb Sampling only uses feature trails collected at the end of stage 1, i.e. once the embedding has converged. Strong Breadcrumb Sampling uses trails collected throughout stage 1 training, i.e. as the embedding evolves. This tends to create an even more adversarial training set

Overall, this work makes several contributions. First, we point out that class-balanced sampling is not an adversarial augmentation technique, which limits its ability to combat over-fitting in few-shot classes. Second, we propose EMANATE, a data augmentation technique that addresses this problem by feature back-tracking with alignment. Third, we show theoretically that, unlike class-balanced sampling, EMANATE is an adversarial technique. Fourth, we propose two variants of a new sampling scheme, Breadcrumbs, which leverage EMANATE to enable long-tailed recognition with state of the art performance. All of this is achieved with no extra computation and no performance degradation for classes with many examples.

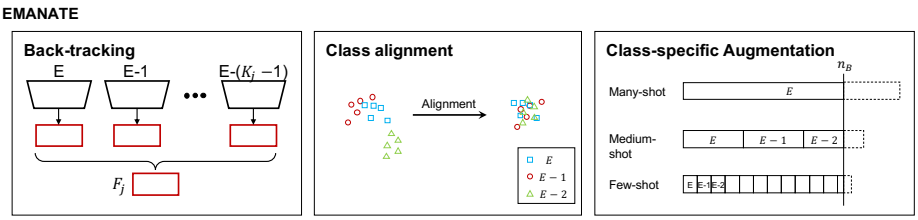

Fig. 2. EMANATE. Left: features from embeddings learned in previous epochs are back-tracked to compose a class-balance training set. Middle: Class alignment aligns the means of features from different epochs. Right: different classes have different back-tracking lengths. Many-shot classes only collect features from the current epoch; medium-shot classes back-track for a few epochs; and few-shot classes from many. When the number of samples exceeds n_B, the earliest epoch is randomly sampled to meet this target.

2 Related Work

Long-tailed recognition has recently received substantial attention [32, 21, 15, 37, 19, 31]. Several approaches have been proposed, including metric learning [21, 37], loss weighting [15], or meta-learning [31]. Some methods propose dedicated loss functions to mitigate the data imbalanced problem. For example, lift loss [21] introduces margins between many training samples. Range loss [37] encourages data from the same class to be close and different classes to be far away in the embedding space. The focal loss [15] dynamically balances weights of positive, hard negative, and easy negative samples. As reported by Liu et al [19], when applied to long-tailed recognition, many of these methods improved accuracy of the few-shot group, but at the cost of lower accuracy for many-shot classes.

Other methods, e.g. class-balanced experts [24] and knowledge distill [35], try to mitigate this problem by artificially dividing the training data into subsets, based on number of examples, and training an expert per subset. However, experts learned from arbitrary data divisions can be sub-optimal, especially for few-shot classes, where training data is insufficient to learn the expert model.

More recent works [14, 39] achieve improved long-tailed recognition by training feature embedding and classifier with separate sampling strategies. The proposed Breadcrumbs approach follows this strategy, learning the embedding in a first stage with sample-balanced (random) sampling and the classifier in a second stage with class-balanced sampling. In fact, Breadcrumbs can be seen as a data augmentation method tailored for this strategy, improving its long-tailed recognition performance over all class groups.

Another related work is LEAP [17], a method mostly tested on person re-identification and face recognition problems, where datasets usually have long-tailed distributions. LEAP augments data samples from tail (few-shot) classes by transferring intra-class variations from head (many-shot) classes. This assumes a shared intra-class variation across classes, which can hold for person re-ID and

face recognition but may not be applicable for general long-tailed recognition tasks. Besides, LEAP is technically orthogonal to Breadcrumbs and the two methods could potentially be combined for further improvement.

Few-shot learning focus solely on the data scarcity problem. A large group of approaches is based on meta-learning, using gradient based methods such as MAML and its variants [6, 7], or LEO [23]. These methods take advantage of second derivatives to optimize the model from few-shot samples. Another group of methods, including matching network [29], prototypical network [25], and relation network [26],aims to learn robust metrics. Since these methods are designed specifically for few-shot classes, they often under-perform for many-shot classes, which makes them ineffective for long-tailed recognition.

Similarly to Breadcrumbs, some few-shot methods have proposed to augmenting training data by combining GANs with meta-learning [30], synthesizing features across object views [16] or using other forms of data hallucination [10]. All these method introduces non-negligible extra computation to generate the new data samples. The application of GAN-based methods to few-shot data without external large-scale datasets can also create convergence problem. In Breadcrumbs, data samples are augmented with saved feature vectors from prior epochs and no extra computation.

3 EMANATE

In this section, we introduce the data augmentation method that underlies Breadcrumbs.

3.1 Data Sampling and Decoupling Training

Consider an image recognition problem with training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i); i = 1, \ldots, N\}$, where x_i is an example and $y_i \in \{1, \ldots, C\}$ its label, where C is the number of classes. A CNN model combines a feature embedding $\mathbf{z} = f(\mathbf{x}; \theta) \in \mathbb{R}^d$, implemented by several convolutional layers of parameters θ, and a classifier $g(\mathbf{z}) \in [0, 1]^C$ that operates on the embedding to produce a class prediction $\hat{y} = \arg\max_i g_i(\mathbf{z})$. Standard (image-balanced) CNN training relies on mini-batch SGD, where each batch is randomly sampled from $\mathcal{D}$. A class j of n_j training example has probability $\frac{n_j}{N}$ of being represented in the batch. Without loss of generality, we assume classes sorted by decreasing cardinality, i.e. $n_i \leq n_j$, $\forall i > j$.

In the long-tail setting, where $n_1 \gg n_C$, the model is not fully trained on classes of large index j (tail classes) and under-fits. This can be avoided with recourse to non-uniform sampling strategies, the most popular of which is class-balanced sampling. This samples each class with probability $\frac{1}{C}$, over-sampling tail classes, and is particularly successful when the training of embedding and classifier are decoupled [14], which is also simple to implement. The embedding is first trained with image-balanced sampling, and different sampling and structures can then be used for the classifier. In this work, we adopt the popular linear

classifier $g(\mathbf{z}) = \nu(\mathbf{W}\mathbf{x}+\mathbf{b})$, where ν is the softmax function, and class-balanced sampling.

3.2 Augmentation by Feature back-tracking

Class-balanced sampling over-samples classes of few examples. For a class j with $n_j < N/C$ the over-sampling factor is $\rho = \frac{N}{Cn_j}$. In the long-tail setting, ρ is usually larger than 10. This heavily resamples the few available samples and can lead to over-fitting, impairing generalization for tail classes. While over-fitting can be combated with data augmentation, traditional image-level methods, such as random cropping, horizontal flipping, or color jittering, make little difference in feature space, because the embedding is trained to be invariant to such transformations. Feature-level augmentations have been investigated in the few-shot setting [30, 16, 10], but typically require training of additional models, which add complexity and sometimes have convergence problems. Ideally, the augmentation technique should be adversarial, i.e. increase training difficulty, and require little extra computation. One possibility is to rely on adversarial examples [9]. However, these require optimization at each training iteration and have large computational cost. In our experience, standard adversarial attacks are also not effective at improving generalization for tail classes, because they are too close to the few available examples.

In this work, we propose a different adversarial feature-level augmentation strategy, based on *feature backtracking*. The idea is that the embedding $f(\mathbf{x};\theta)$, obtained after training converges, is simply the final element in the family of embeddings $f(\mathbf{x};\theta^e)$ learned from epochs $e \in \{1,\dots,E\}$, where E is the number of training epochs. It follows that a particular image $\mathbf{x}_i$ produces a sequence of feature vectors

$$\mathcal{B}_i = \{\mathbf{z}_i^e = f(\mathbf{x}_i;\theta^e) | e \in \{1,\dots,E\}\} \tag{1}$$

during the optimization. We equate $\mathcal{B}_i$ to a trail of *bread crumbs* that can be backtracked, as illustrated in Figure 1(a,right). These bread crumbs can be used to perform data augmentation *without* added computation. It suffices to store, at epoch e the set of features

$$\mathcal{Z}^e = \{\mathbf{z}_i^e = f(\mathbf{x}_i;\theta^e) | \mathbf{x}_i \in \mathcal{D}\} \tag{2}$$

produced by the embedding learned at the end of the epoch. This is denoted as the training set *snapshot* at epoch e.

Since the embedding $f(\mathbf{x};\theta^e)$ changes with e, features from different epochs are usually not aligned in feature space. This may lead to bread crumb trails that are "all over the place," e.g. because the space has been translated or rotated between epochs. Hence, when feature vectors collected at different epochs are to be used together, a *class alignment* is recommended to simplify the training. On the other hand, this alignment cannot be too strong, so as not to defeat the purpose of data-augmentation. In particular, the alignment operation should not jeopardize the adversarial nature of the latter. A simple operation, which is

shown to satisfy this property in the following section, is to align the mean feature vectors synthesized per class during back-tracking. This consists of splitting $\mathcal{Z}^e$ into a set of class snapshots, where

$$\mathcal{Z}_y^e = \{\mathbf{z}_i^e \in \mathcal{Z}^e | y_i = y\} \tag{3}$$

is the snapshot of class y, compute the mean of each class

$$\bar{\mathbf{z}}_y^e = \frac{1}{n_j} \sum_{j=1}^{n_j} \mathbf{z}_{y,j}^e \tag{4}$$

where $\mathbf{z}_{y,j}^e$ is the j^{th} element of $\mathcal{Z}_y^e$, and apply

$$\mathbf{z}_{y,j}^{e' \to e} = \mathbf{z}_{y,i}^{e'} - \bar{\mathbf{z}}_j^{e'} + \bar{\mathbf{z}}_j^e, \tag{5}$$

where $\mathbf{z}_{y,j}^{e' \to e}$ is the alignment, with respect to snapshot e, of the j^{th} feature vector $\mathbf{z}_{y,j}^e$ of class y from epoch e'. This produces a snapshot *transferred from epoch e' to e*

$$\mathcal{Z}_y^{e' \to e} = \{\mathbf{z}_{y,j}^{e' \to e} | \mathbf{z}_{y,j}^{e'} \in \mathcal{Z}_y^{e'}\}. \tag{6}$$

This snapshot can then be combined with $\mathcal{Z}_y^e$ to produce an *augmented snapshot of class y for epoch e*

$$\mathcal{A}_y^e = \mathcal{Z}_y^e \bigcup \mathcal{Z}_y^{e' \to e}. \tag{7}$$

This process is denoted *fEature augMentAtioN by bAcktracking wiTh alignmEnt* (EMANATE), as $\mathcal{A}_y^e$ backtracks the breadcrumb trails that emanate from class y at epoch e.

3.3 Theoretical justification

In this section, we provide theoretical motivation for EMANATE as an adversarial data augmentation technique. Let $\nu(\mathbf{W}^e\mathbf{z} + \mathbf{b}^e)$ be the linear classifier learned at the end of epoch e, i.e. from the snapshots $\mathcal{Z}_y^e = \{\mathbf{z}_{y,i}^e\}$ of (3). The corresponding cross-entropy loss is

$$L(\mathcal{Z}^e, \mathbf{W}^e, \mathbf{b}^e) = \sum_y L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e) \tag{8}$$

where

$$L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e) = -\frac{1}{|\mathcal{Z}_y^e|} \Sigma_i \log \nu_y(\mathbf{W}^e \mathbf{z}_{y,i}^e + \mathbf{b}^e), \tag{9}$$

is the loss of class y and ν_y the y^{th} element of the softmax output. It is assumed that the classifier is optimal for the training data under this loss, i.e.

$$L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e) \leq L_y(\mathcal{Z}_y^e, \mathbf{W}, \mathbf{b}), \quad \forall\, y, \mathbf{W}, \mathbf{b}. \tag{10}$$

A feature augmentation procedure adds new features to $\mathcal{Z}_y^e$. It is denoted adversarial when the augmented training set is more challenging than the original.

Definition 1 *Consider the augmentation $\mathcal{A}_y^e$ of the training set snapshot $\mathcal{Z}_y^e$ from epoch e and class y. The augmentation is adversarial with respect to class y if*

$$L_y(\mathcal{A}_y^e, \mathbf{W}^e, \mathbf{b}^e) > L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e) \tag{11}$$

where $L_y(.)$ the loss of (9).

For low-shot classes y, class-balanced sampling replicates the features of $\mathcal{Z}_y^e$, creating the augmented feature set $\mathcal{A}_y^e = \mathcal{Z}_y^e \cup \mathcal{Z}_y^e$. Since, from (9)

$$\begin{aligned} L_y(\mathcal{A}_y^e, \mathbf{W}^e, \mathbf{b}^e) &= -\frac{1}{2|\mathcal{Z}_y^e|} 2\Sigma_i \log \nu_y(\mathbf{W}^e \mathbf{z}_{y,i}^e + \mathbf{b}^e), \\ &= L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e) \end{aligned} \tag{12}$$

we obtain the following corollary.

Corollary 1 *Class-balanced sampling is not an adversarial feature augmentation strategy.*

We next consider augmentation with EMANATE. The following lemma establishes a lower bound for the increase of the training loss under this augmentation technique.

Lemma 1 *Consider the augmentation of $\mathcal{Z}_y^e$ with the snapshot transferred from epoch $e' < e$ by EMANATE, i.e. $\mathcal{A}_y^e = \mathcal{Z}_y^e \cup \mathcal{Z}_y^{e' \to e}$, where $\mathcal{Z}_y^{e' \to e}$ is as defined in (6). Then*

$$\begin{aligned} &L_y(\mathcal{A}_y^e, \mathbf{W}^e, \mathbf{b}^e) - L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e) \geq \\ &\frac{L_y(\mathcal{Z}_y^{e'}, \mathbf{W}^{e'}, \mathbf{b}^{e'}) - L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e)}{2}, \end{aligned} \tag{13}$$

where $(\mathbf{W}^e, \mathbf{b}^e)$ is the classifier of (10).[3]

The lemma shows that the adversarial increase of the loss due to the augmentation ($L_y(\mathcal{A}_y^e, \mathbf{W}^e, \mathbf{b}^e) - L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e)$) is at least half of decrease in the loss of the trained classifier between epochs e' (loss $L(\mathcal{Z}_y^{e'}, \mathbf{W}^{e'}, \mathbf{b}^{e'})$) and e (loss $L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e)$), i.e. half of what has been gained by training the classifier from epochs e' to e. This is illustrated in Figure 1(b) and leads to the following theorem.

Theorem 1 *EMANATE is an adversarial feature augmentation strategy for any convergent training scheme, i.e. whenever $L_y(\mathcal{Z}_y^{e'}, \mathbf{W}^{e'}, \mathbf{b}^{e'}) > L_y(\mathcal{Z}_y^e, \mathbf{W}^e, \mathbf{b}^e) \forall e' < e$.*

Since successful training requires a convergent training scheme, EMANATE is an adversarial feature augmentation technique for most training procedures of practical interest.

[3] Proof is provided in supplementary material.

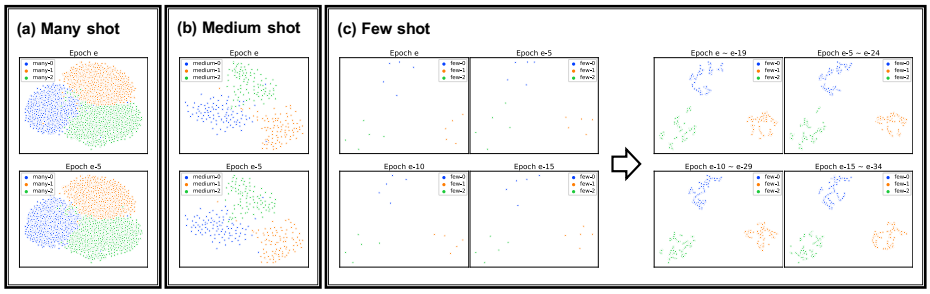

Fig. 3. t-SNE visualizations of feature snapshots at different epochs. Many-shot (a) and medium-shot (b) features compose a well-defined geometry that does not change along epochs. Due to the scarcity of data, few-shot features (c, left) fail to hold a consistent geometry along epochs. After augmentation with EMANATE (c, right), the features have more variety and the geometry changes less among epochs.

3.4 Assembling feature trails

So far, we have considered the augmentation of $\mathcal{Z}_y^e$ with the transferred snapshot $\mathcal{Z}_y^{e'\to e}$. The augmentation can obviously be repeated for several transferred snapshots e', in order to meet any target number n_B of samples per class at epoch e. This is done as follows. Consider a class j with n_j samples. The features in $\mathcal{Z}_j^e$ are first selected. If there is a differential to n_B, the transferred snapshot $\mathcal{Z}_j^{e-1\to e}$ is selected next. The procedure is repeated until number of the feature vectors reaches n_B. If the addition of the final set places the feature cardinality above n_B, the necessary number of feature vectors is sampled randomly. The augmented set of features that emanate from class j at epoch e is then

$$\mathcal{A}_j^e = \bigcup_{k=0}^{K_j-2} \mathcal{Z}_j^{e-k\to e} \bigcup \tilde{\mathcal{Z}}_j^{e-(K_j-1)\to e}, \tag{14}$$

where $K_j = \left\lceil \frac{n_B}{n_j} \right\rceil$, and $\tilde{\mathcal{Z}}_j^{e-K_j-1\to e}$ is a random sample from $\mathcal{Z}_j^{e-K_j-1\to e}$ of size $n_B - K_j n_j$. The complete training set of epoch e is $\mathcal{A}^e = \cup_{j=1}^{C} \mathcal{A}_j^e$.

The number of snapshots in $\mathcal{A}_j^e$ depends heavily on the number of examples n_j of the class. As shown in Figure 2 (a, right), many-shot classes use a single snapshot, medium-shot classes require snapshots from a few epochs, and few-shot classes require many snapshots to assemble enough training features. However, in all cases, because all feature vectors are already computed during the optimization of the embedding, the only computation required is the mean alignment of (5). This is negligible when compared to the back-propagation computations, making EMANATE nearly computation free, if the necessary snapshots are kept in memory. In fact, it only necessary to keep in memory the snapshots of classes with $n_j < n_B$. Furthermore, the number K_j of snapshots to be stored adapts to n_j, as shown in (14). The larger the class, the fewer snapshots are required. In

summary, EMANATE has no computational overhead and adapts the memory requirements to the class cardinalities, never requiring more than n_B examples per class. This is the complexity of class-balanced sampling.

Figure 3 shows t-SNE [20] of training set snapshots collected at different epochs. While the geometry of many- and medium shot classes (Figure 3(a,b)) is fairly stable across epochs, that of few-shot classes (Figure 3(c) left) can change significantly, due to data scarcity. EMANATE produces larger clusters with more stable geometry, enabling a more robust training set for the classifier.

4 Breadcrumbs

In this section, we investigate two sampling mechanisms based on EMANATE, which are denoted as Breadcrumb sampling. The two mechanisms differ in how the sets $\mathcal{A}_j^e$ are collected. In both cases, the two stage training procedure of [14] is adopted. In the first stage, the feature extractor $f(\mathbf{x};\theta)$ and the classifier $\nu(\mathbf{W}\mathbf{x}+b)$ are trained with image balanced sampling. The sets $\mathcal{A}_j^e, e = \{1,\ldots,E\}$ of class snapshots are collected at each epoch of this stage. In the second stage, the feature extractor $f(\mathbf{x};\theta)$ is kept fixed and the classifier $\nu(\mathbf{W}\mathbf{x}+b)$ retrained using these sets. As shown in Figure 4, the two augmentation schemes differ in the classifier update step.

4.1 Weak Beadcrumb Sampling.

In the first approach, EMANATE is only applied *after convergence* of the first stage training. That is, only the sets $\mathcal{A}_j^E$ assembled in the *final* epoch E of the first stage are used to retrain the classifier in the second stage. This is illustrated in the left of Figure 4, for the case where $E = 3$ and augmentation sets span two epochs. We refer to this sampling technique as *Weak Beadcrumb Sampling*, since all snapshots emanate from the feature set produced by the optimal embedding $f(\mathbf{x},\theta^E)$. While this creates some diversity, feature snapshots from neighboring epochs are likely to be similar. This makes the sampling technique less adversarial and therefore "weak".

4.2 Strong Beadcrumb Sampling.

Strong Beadcrumb Sampling aims to increase feature diversity, so as to create a more adversarial data augmentation. Rather than the augmentation $\mathcal{A}_j^E$ of the final epoch E of the first stage training, *all* augmentations $\mathcal{A}_j^e, e = \{1,\ldots,E\}$ are saved in that stage, as illustrated in the right of Figure 4. The classifier retraining of the second stage is then run for E epochs, using the the feature trail sets $\mathcal{A}_j^e$ collected at epoch e of the first stage to train the classifier in epoch e of the second stage. Since each epoch of classifier training contains new data, increasing the difficulty of the classification task, this sampling method is more adversarial and therefore "strong". Since the classifier is trained on an evolving feature set $\mathcal{A}^e$,

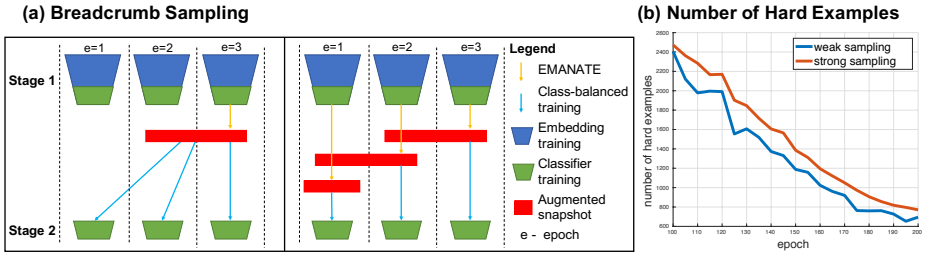

Fig. 4. (a) Breadcrumb Sampling relies on EMANATE to collect augmented snapshots $\mathcal{A}_j^e$ (in red) in a first stage, when the embedding is trained with image-balanced sampling. In a second stage, the classifier is learned with class-balanced training based on these snapshots. In this example $E = 3$ and snapshots have length $K_j = 2$ (a single class is shown for simplicity). Left: Weak Breadcrumb Sampling only uses snapshots collected at the end of stage 1. Right: Strong Breadcrumb Sampling uses snapshots collected throughout stage 1 training. (b) Number of hard examples (loss larger than 5) in few-shot classes during training, for ResNet-10 on ImageNet-LT. Strong Breadcrumb sampling increases the number of hard examples during training, compared to the weak one. The plot starts at epoch 100 because early epochs have too many hard examples and dominate the scale.

This setting yields a natural selection of the target number n_B of samples per class. To keep the pace of classifier training the same as the embedding training, the size of the dataset should be approximately the same, i.e. $n_B = \left\lceil \frac{1}{C} \sum_{j=1}^{C} n_j \right\rceil$. Figure 2(b), shows that Strong Breadcrumb Sampling increases the number of hard examples in few-shot classes per epoch, when compared to Weak Breadcrumb Sampling. This confirms that it is a more adversarial data augmentation strategy.

5 Experiments

5.1 Experimental set-up

Datasets. We consider three long-tailed recognition datasets, ImageNet-LT [19], Places-LT [19] and iNatrualist18 [28]. ImageNet-LT is a long-tailed version of ImageNet [4] by sampling a subset following the Pareto distribution with power value $\alpha = 6$. It contains 115.8K images from 1000 categories, with class cardinality ranging from 5 to 1280. Places-LT is a long-tailed version of the Places dataset [38]. It contains 184.5K images from 365 categories with class cardinality in $[5, 4980]$. iNatrualist18 is a long-tailed dataset, which contains 437.5K images from 8141 categories with class cardinality in $[2, 1000]$. Following [19], we present classification accuracies for both the entire dataset and three groups of classes: *many shot* (more than 100 training samples), *medium shot* (between 20 and 100), and *few shot* (less than 20 training samples).

Baselines. Following [19], we consider three metric-learning baselines, based on the lifted [21], focal [15], and range [37] losses, and one state-of-the-art method, FSLwF [8], for learning without forgetting. We also include long-tailed recognition methods designed specifically for these datasets, OLTR [19] and Distill [35], plus the recent state of the art Decoupling method [14]. The model with standard random sampling and end-to-end training is denoted as the *Plain Model* for comparison.

Training Details. ResNet-10 and ResNeXt-50 [11, 36] are used on ImageNet-LT; ResNet-152 is used on Places-LT; and ResNet-50 is used on iNatrualist18. The model is trained with SGD, using momentum 0.9, weight decay 0.0005, and a learning rate that cosine decays from 0.2 to 0. Each iteration uses class-balanced and random sampling mini-batches of size 512. One epoch is defined when the random sampling iterates over the entire training data. Under Strong Breadcrumb Sampling, class-balanced sampling is applied in the initial classifier training epochs, when there are not enough previous epochs to back-track. Codes are attached in supplementary.

Table 1. Ablation of Breadcrumb components, on the ImageNet-LT. For many-shot $t > 100$, for medium-shot $t \in (20, 100]$, and for few-shot $t \leq 20$, where t is the number of training samples.

Method	Overall	Many-Shot	Medium-Shot	Few-Shot
Decoupling [14]	41.4	51.8	38.8	21.5
+ back-tracking	41.2	50.4	38.5	23.8
+ class-specific	41.3	50.8	38.1	24.6
Weak Breadcrumb	43.2	53.6	39.8	25.1
Strong Breadcrumb	**44.0**	53.7	**41.0**	**26.4**
Breadcrumb	**44.0**	53.7	**41.0**	**26.4**
Breadcrumb(var.)	43.9	**53.8**	40.8	26.0
Breadcrumb(agn.)	38.5	47.3	35.6	24.0

Table 2. Results on ImageNet-LT and Places-LT. ResNet-10/152 are used for all methods. For many-shot $t > 100$, for medium-shot $t \in (20, 100]$, and for few-shot $t \leq 20$, where t is the number of training samples.

Method	ImageNet-LT, ResNet-10				Places-LT, ResNet-152			
	Overall	Many-Shot	Medium-Shot	Few-Shot	Overall	Many-Shot	Medium-Shot	Few-Shot
Plain Model	23.5	41.1	14.9	3.6	27.2	**45.9**	22.4	0.36
Lifted Loss [21]	30.8	35.8	30.4	17.9	35.2	41.1	35.4	24.0
Focal Loss [15]	30.5	36.4	29.9	16.0	34.6	41.1	34.8	22.4
Range Loss [37]	30.7	35.8	30.3	17.6	35.1	41.1	35.4	23.2
FSLwF [8]	28.4	40.9	22.1	15.0	34.9	43.9	29.9	29.5
OLTR [19]	35.6	43.2	35.1	18.5	35.9	44.7	37.0	25.3
Distill [35]	38.8	47.0	37.9	19.2	36.2	39.3	39.6	24.2
Decoupling(cRT) [14]	41.4	51.8	38.8	21.5	37.9	37.8	40.7	31.8
Breadcrumb	**44.0**	**53.7**	**41.0**	**26.4**	**39.3**	40.6	**41.0**	**33.4**

Table 3. Results on ImageNet-LT, ResNeXt-50. For many-shot $t > 100$, for medium-shot $t \in (20, 100]$, and for few-shot $t \leq 20$, where t is the number of training samples.

Method	Overall	Many-Shot	Medium-Shot	Few-Shot
OLTR [19]	41.9	51.0	40.8	20.8
Decoupling(NCM) [14]	47.3	56.6	45.3	28.1
Decoupling(cRT) [14]	49.6	61.8	46.2	27.4
Decoupling(τ) [14]	49.4	59.1	46.9	30.7
Decoupling(LWS) [14]	49.9	60.2	47.2	30.3
Causal [27]	50.6	62.3	46.9	30.6
LADE [12]	51.9	62.3	49.3	31.2
Breadcrumb	**51.0**	**62.9**	**47.2**	**30.9**

Table 4. Results on the iNaturalist 2018. All methods are implemented with ResNet-50.

Method	Accuracy
CB-Focal [3]	61.1
LDAM+DRW [1]	68.0
Decoupling(cRT) [14]	68.2
Decoupling(τ) [14]	69.3
Decoupling(LWS) [14]	69.5
Causal [27]	64.4
LADE [12]	70.0
BBN [39]	66.3
Breadcrumb	**70.3**

5.2 Ablation Study

Several ablations were performed to study the effectiveness of the various components of Breadcrumb. In this study, all models are trained and evaluated on the training and test set of ImageNet-LT, respectively, using a ResNet-10 backbone.

Component ablation. Starting from the baseline Decoupling (cRT) [14] method, we incrementally add feature back-tracking, class-specific augmentation, class alignment (leading to Weak Breadcrumb Sampling), and Strong Breadcrumb Sampling. Results are shown in Table 1. When only back-tracking is applied, all snapshots are collected from the last 10 epochs of image-balanced training (first stage), and the classifier trained (in the second stage) using this feature set and class-balanced sampling. No class alignment is applied. Compared to the baseline, back-tracking gives a reasonable gain on few-shot classes but harms many-shot performance. This can be explained by the fact that, for many-shot classes, features from the final epoch are replaced by those from prior epochs. Since the corresponding embeddings are sub-optimal, the augmented features are inferior to the final ones. This, however, is not the case in few-shot, where augmented features replace *duplicated* features.

The combination of back-tracking and class-specific augmentation, where different classes have different back-tracking lengths, is denoted as "+ class-specific" in Table 1. Surprisingly, without class alignment, the performance on many-shot does not improve, even though no augmented features are introduced into those classes. We believe this is due to the fact that when few-shot features are augmented without alignment, those augmented features take up position in feature space that should not be assigned to them. This decreases the accuracy of many-shot classes. When class-alignment is applied (Weak Breadcrumb Sampling) we observe an improvement over all class partitions, with gains of 1.8% (Many), 1% (Medium), and $3,6$% (Few-Shot) and an overall improvement of 1.8% over the baseline. Finally, Strong Breadcrumb Sampling enables another 0.8% overall gain, for a total gain of 2.6% over the baseline.

Class alignment ablation. Since alignment makes a significant difference, we considered three different alignment choices. In Sec 3.2, only class-specific mean alignment is presented. It is also possible to align the feature variances. This is denoted as Breadcrumb(var.) in Table 1 and has a negligible difference. Hence,

we only apply mean alignment unless otherwise noted. Another possibility is class-agnostic alignment, where only one mean is computed over all classes. This is listed as Breadcrumb(agn.) in Table 1. Its poor performance implies that class-agnostic alignment cannot fully eliminate the differences between epochs.

5.3 Comparison to the state of the art

Table 2 presents a final comparison to the methods in the literature on ImageNet-LT, using a ResNet-10, and Places-LT, using a ResNet-152. In these experiments we use Strong Breadcrumb Sampling, which is shown to outperform all other methods on both datasets. It achieves the best performance on 5 of the 6 partitions and is always better than the next overall best performer (Decoupling(cRT)). It is only outperformed by the Plain Model on the Many-Shot split of Places-LT, where this model severely overfits to the Many-Shot classes, basically ignoring the Few-Shot ones, and achieving overall performance 12.1% weaker than Breadcrumb. Compared to the best models Breadcrumb also achieves significant gains on few-shot classes, especially on ImageNet-LT, where it beats the next best method by 4.9%. This suggests that previous methods over-fit for few-shot classes, a problem that is mitigated by the introduction of EMANATE and Strong Breadcrumb Sampling. Table 3 shows that these results are fairly insensitive to the backbone network. Breadcrumb achieves the best overall performance and the best performance on all partitions with a ResNeXt-50 backbone. Finally Table 4 shows that Breadcrumb again achieves the overall best results for a ResNet-50 on iNaturalist.

6 Conclusion

This work discussed the long-tailed recognition problem. A new augmentation framework, Breadcrumb, was proposed to increase feature variety and classifier robustness. Breadcrumb is based on EMANATE, a feature back-tracking procedure that aligns features vectors produced across several epochs of embedding training, to compose a class-balanced feature set for training the classifier at the top of the network. It is inspired by the the recent success of class-balanced training schemes. However, unlike previous schemes, it is shown to be an adversarial sampling scheme, a property that encourages better generalization. A comparison of two sampling schemes based on EMANATE confirmed this property, resulting in best performance for the Strong Breadcrumb Sampling technique, where feature snapshots are collected while the embedding is evolving. Breadcrumb was shown to achieve state-of-the-art performance on three popular long-tailed datasets with different CNN backbones. Furthermore, Breadcrumb introduces no extra model, which means that it adds no computational overhead or convergence issues to the baseline model.

Acknowledgement. This work was partially funded by NSF awards IIS-1924937 and IIS-2041009. and the use of the Nautilus platform for some of the experiments discussed above. Gang Hua was supported partly by National Key R&D Program of China Grant 2018AAA0101400 and NSFC Grant 61629301.

References

1. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: Advances in Neural Information Processing Systems. pp. 1565–1576 (2019)
2. Chen, Z., Fu, Y., Wang, Y.X., Ma, L., Liu, W., Hebert, M.: Image deformation meta-networks for one-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8680–8689 (2019)
3. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9268–9277 (2019)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
5. Felix, R., Kumar, V.B., Reid, I., Carneiro, G.: Multi-modal cycle-consistent generalized zero-shot learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 21–37 (2018)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1126–1135. JMLR. org (2017)
7. Finn, C., Xu, K., Levine, S.: Probabilistic model-agnostic meta-learning. In: Advances in Neural Information Processing Systems. pp. 9516–9527 (2018)
8. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4367–4375 (2018)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
10. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3018–3027 (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., Chang, B.: Disentangling label distribution for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6626–6636 (2021)
13. Huang, C., Li, Y., Chen, C.L., Tang, X.: Deep imbalanced learning for face recognition and attribute prediction. IEEE transactions on pattern analysis and machine intelligence (2019)
14. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: Eighth International Conference on Learning Representations (ICLR) (2020)
15. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
16. Liu, B., Wang, X., Dixit, M., Kwitt, R., Vasconcelos, N.: Feature space transfer for data augmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
17. Liu, J., Sun, Y., Han, C., Dou, Z., Li, W.: Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2970–2979 (2020)

18. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)
19. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2537–2546 (2019)
20. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
21. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4004–4012 (2016)
22. Qi, H., Brown, M., Lowe, D.G.: Low-shot learning with imprinted weights. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5822–5830 (2018)
23. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=BJgklhAcK7
24. Sharma, S., Yu, N., Fritz, M., Schiele, B.: Long-tailed recognition using class-balanced experts. arXiv preprint arXiv:2004.03706 (2020)
25. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in neural information processing systems. pp. 4077–4087 (2017)
26. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1199–1208 (2018)
27. Tang, K., Huang, J., Zhang, H.: Long-tailed classification by keeping the good and removing the bad momentum causal effect. Advances in Neural Information Processing Systems **33**, 1513–1524 (2020)
28. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018)
29. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in neural information processing systems. pp. 3630–3638 (2016)
30. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7278–7286 (2018)
31. Wang, Y.X., Hebert, M.: Learning to learn: Model regression networks for easy small sample learning. In: European Conference on Computer Vision. pp. 616–634. Springer (2016)
32. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. In: Advances in Neural Information Processing Systems. pp. 7029–7039 (2017)
33. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5542–5551 (2018)
34. Xian, Y., Sharma, S., Schiele, B., Akata, Z.: f-vaegan-d2: A feature generating framework for any-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10275–10284 (2019)

35. Xiang, L., Ding, G., Han, J.: Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In: European Conference on Computer Vision. pp. 247–263. Springer (2020)
36. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
37. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5409–5418 (2017)
38. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in neural information processing systems. pp. 487–495 (2014)
39. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9719–9728 (2020)
40. Zhu, X., Anguelov, D., Ramanan, D.: Capturing long-tail distributions of object subcategories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 915–922 (2014)