Inflated Episodic Memory with Region Self-Attention for Long-Tailed Visual Recognition

Linchao Zhu[1,2] and Yi Yang[2*]
[1] Baidu Research [2] ReLER, University of Technology Sydney
{linchao.zhu,yi.yang}@uts.edu.au

Abstract

There have been increasing interests in modeling long-tailed data. Unlike artificially collected datasets, long-tailed data are naturally existed in the real-world and thus more realistic. To deal with the class imbalance problem, we introduce an Inflated Episodic Memory (IEM) for long-tailed visual recognition. First, our IEM augments the convolutional neural networks with categorical representative features for rapid learning on tail classes. In traditional few-shot learning, a single prototype is usually leveraged to represent a category. However, long-tailed data has higher intra-class variances. It could be challenging to learn a single prototype for one category. Thus, we introduce IEM to store the most discriminative feature for each category individually. Besides, the memory banks are updated independently, which further decreases the chance of learning skewed classifiers. Second, we introduce a novel region self-attention mechanism for multi-scale spatial feature map encoding. It is beneficial to incorporate more discriminative features to improve generalization on tail classes. We propose to encode local feature maps at multiple scales, and the spatial contextual information should be aggregated at the same time. Equipped with IEM and region self-attention, we achieve state-of-the-art performance on four standard long-tailed image recognition benchmarks. Besides, we validate the effectiveness of IEM on a long-tailed video recognition benchmark, i.e.*, YouTube-8M.*

1. Introduction

Recently, visual recognition models [18, 12] have achieved significant success with the renaissance of deep convolutional neural networks (ConvNets). These models are usually trained on large datasets, *e.g.*, ImageNet [28] and Kinetics [16], demonstrating satisfying generalization capabilities in various tasks, *e.g.*, object detection [27], object segmentation [22], video localization [10], video question answering [42]. These datasets are artificially collected to be balanced, where the number of training examples in each category is roughly the same. However, data in real-world applications usually follow a "long-tailed" distribution [2, 6]. In this distribution, a large number of examples are data-scarce with only a few training examples. Specifically, a few "head" classes contain thousands of examples per category, while few instances exist for "tail" classes. Human exhibits remarkable generalization capabilities in recognizing rare examples. They can identify examples by only observing objects a few times or even never seeing them before. This generalization capability is essential in deploying deep networks to real-world applications. Cur-

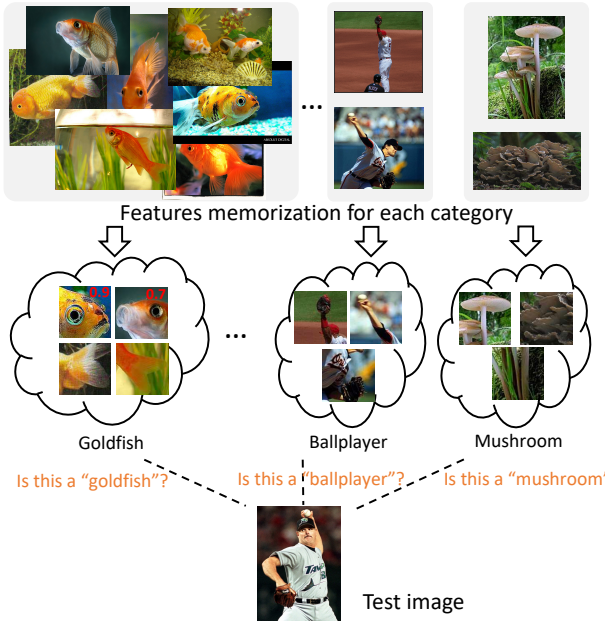

Figure 1: Illustration of inflated episodic memory. The visual cues are stored separately based on their categorical information.

*This work was done when Linchao Zhu visited Baidu Research. Yi Yang is the corresponding author.

rent deep recognition models largely ignore the long-tailed visual phenomena, making it challenging to extract robust information from real data. Consequently, the performance of "tail" classes can be significantly degenerated due to data deficiency.

There have been a few attempts in modeling long-tailed data to enhance generalization on tail classes. One promising direction is to transfer knowledge from head classes to tail classes [7, 38, 40]. Wang *et al*. [38] adaptively trained a meta-network on head classes and then applied it to tail classes. Liu *et al*. [21] introduced a dynamic meta-embedding to improve the robustness of tail recognition. They also introduced to evaluate performance on the open-set data. In their Open Long-Tailed Recognition (OLTR) setting, the goal is to learn from both long-tailed and open-ended data. However, the meta-embedding is not dynamically updated during training, and each category corresponds to a single embedding vector. The single embedding vector might fail to represent the data distribution. In this paper, we introduce a new framework with an inflated episodic memory to tackle the OLTR problem.

First, we propose an Inflated Episodic Memory (IEM) to augment ConvNets with multiple memory banks (Figure 1). Each memory is independent. We utilize a differentiable memory block for each category. Each memory bank records the most discriminative features for the corresponding category. It can be a natural design choice for extremely imbalanced datasets as each memory bank is updated independently. We are motivated by episodic training for few-shot classification in [36, 29], where a prototype is calculated for each class in an episode. In the few-shot regime, the number of categories for each episode is small. Thus, a single prototype is sufficient to represent a category. Liu *et al*. [21] extended the idea by leveraging a global memory structure that stores prototypes ("centroids") for all categories. Different from few-shot learning, OLTR involves with more training examples and more classes. Due to higher intra-class variances, it is more challenging to learn a single prototype for all examples in a category. Our introduced IEM enables more robust representation learning of the prototypes and provides a powerful mechanism for imbalanced data modeling.

Second, we propose to extract discriminative region features with our novel region self-attention mechanism (RSA). Our region self-attention considers features at different scales. It is beneficial to exploit local region features and utilize the most discriminative feature to improve recognition of tail categories. Contextual information is exploited during spatial feature encoding. RSA leverages contextual relationships using the self-attention mechanism during feature encoding. With region self-attention, IEM records stronger discriminative features for all categories. The performance is boosted when more visual cues are explored.

Third, we keep both the global feature and the RSA-encoded feature in two separate banks. In this way, the local features and the global features are updated independently, offering a more feasible way for network weight training and memory writing. We evaluate IEM on both long-tailed video classification and long-tailed image classification tasks. We achieve the state-of-the-art performance on five datasets.

2. Related Work

Imbalanced visual recognition. Data resampling is a straightforward approach to model imbalanced data, where [8] introduced a class rectification loss to discover sparsely sampled boundaries of tail classes. Another direction is to transfer knowledge from head classes to tail classes [38, 7, 21]. [21] used instance-balanced sampling to learn representations and used a class-balanced sampling for long-tailed classification. Cao *et al*. [4] introduced a margin loss that expands the decision boundaries of tail classes. [38] proposed to transfer meta-knowledge in a progressive manner, from head classes to tail classes. Recent 2D [30, 41] and 3D ConvNets for video classification [34, 5, 35] evaluate on balanced datasets, *e.g*., Kinetics [5]. The studies of imbalanced video classification are largely ignored. YouTube-8M [1] is a large-scale long-tailed dataset. However, NetVLAD [23] is still the prevailing method, which does not consider long-tailed nature in YouTube-8M. We investigate IEM for long-tailed video classification.

Few-shot classification. In few-shot classification, the goal is to generalize to novel categories given few examples [9, 36, 29, 33, 11, 26, 43]. The tail classes in long-tailed classification contain only a few examples, and they perform worse than head classes. It is promising to improve long-tailed recognition by enhancing generalization on tail classes. [9] introduced to use a cosine classifier between the feature representations and the classification weight vectors. [31] proposed to generate a prototype for each class during few-shot classification. Liu *et al*. [21] extended the idea of prototypical learning to long-tailed visual classification. Our IEM leverages a memory for each class, enabling the learning of more robust representations for each category.

Memory-augmented networks. Memory-augmented neural networks have made remarkable achievements in recent years [32, 15, 25]. Kaiser *et al*. [15] proposed a key-value memory module to update the memory via element inserting. All examples are written to a global memory, and the memory is updated via a ranking loss. In contrast, we propose inflated episodic memory to store categorical visual cues independently.

3. Our Method

We design a new framework focusing on learning multi-scale local features and global features for visual cues memorization. We aim to improve the robustness of recognition on tail classes and generalization in open classes. Due to data imbalance, the typical classifier is highly skewed towards the head classes. We tackle this problem by introducing a novel module named inflated episodic memory. We first introduce IEM with lookup and update operations in Section 3.1. IEM can effectively store features in the long-term. In Section 3.2, we introduce our region self-attention mechanism to learn multi-scale local representations. We present the whole framework in Section 3.3.

3.1. Inflated Episodic Memory

For each category in a dataset, there is a corresponding inflated episodic memory. IEM rapidly integrates the visual representations and the corresponding confidences, which can be retrieved quickly for future predictions.

IEM follows the key-value form. [25] introduced a similar approach, but they applied it in the reinforcement learning scenario. We denote the l-th IEM as $M_l = (K_l, V_l)$, where K_l is the key memory and V_l is the value memory. The key memory saves the encoded features, while the value memory stores the probability of the features belonging to category l. Each slot in memory K_l corresponds to a memory slot in V_l. K_l contains arrays of vectors with variable sizes. The memory is extended when new items needed to be written. For each M_l, there is a memory size limitation. It is used to avoid the out of memory problem.

In [15], the memory is reconstructed for each episode. However, our IEM is not cleared, and it persists during the whole learning process. The information in IEM can be leveraged at inference.

Reading. Similar to most memory networks [32], the lookup operation in IEM is based on the soft attention mechanism. For each lookup, the output is a weighted sum of the values in the value memory. The weights are generated by the similarity measurement between the lookup query and the related keys in the key memory. Given a query $\mathbf{q}$, the output p is generated by,

$$p = \frac{\sum_i s(\mathbf{q}, \mathbf{k}_i) v_i}{\sum_i s(\mathbf{q}, \mathbf{k}_i)} \tag{1}$$

where v_i is the i-th prediction score in the value memory V_l, and $\mathbf{k}_i$ is i-th key vector in the key memory K_l. The similarity function $s(\mathbf{a}, \mathbf{b})$ measures the distances between two vectors. Following [25], we leverage inversed squared Euclidean distance,

$$s(\mathbf{a}, \mathbf{b}) = \frac{1}{||\mathbf{a} - \mathbf{b}||_2^2 + \delta}, \tag{2}$$

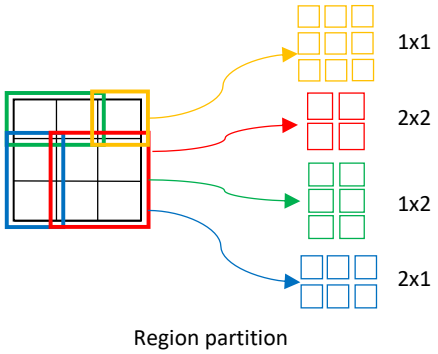

Figure 2: The feature map is partitioned into regions. The feature map is 3×3, and the maximum allowed region is 2×2. Four regions are generated after partition.

where δ is added to avoid division by zero. This similarity function is found to be robust to tail examples that are not similar to the given query. We leverage kd-tree [3] to build indices on a large number of entries for fast retrieval. It enables efficient access when the memory size grows. Furthermore, we select top-n related slots for each memory reading to avoid slow updates occurred in all memory slots. This process further speeds up the access process.

Writing. We introduce the writing operation. In [32, 36], there are no explicit writing operations. We introduce the writing operation to enable dynamic memory representation updates. The memory acts like a linked list. When a new pair needed to be written, we simply append the new pair to the original memory. Specifically, keys and values are written to IEM by appending them onto the end of the memory K_l and V_l, respectively,

$$K_l = \text{Concat}(K_l, \mathbf{k}_i), V_l = \text{Concat}(V_l, v_i). \tag{3}$$

"Concat" is the concatenation operation. If the key has already existed in the memory, the new key is not appended. Meanwhile, its corresponding value is updated. We guarantee that there are no duplicated key vectors in the memory bank. The size of the memory is dynamically altered. To avoid out of GPU memory, we set the memory with a maximum capacity. When the memory's maximum capacity is reached, the oldest key-value pair is deleted. The oldest key-value pair is the least frequently accessed slot. We use an age vector to record the access frequency for each memory slot, following [15]. The value is updated by,

$$v^{t+1} = \gamma * v^n + (1 - \gamma) v^t, \tag{4}$$

where v^t is the original value in the memory, v^n is the new value to be written, v^{t+1} is the updated value, and γ is weighting parameter.

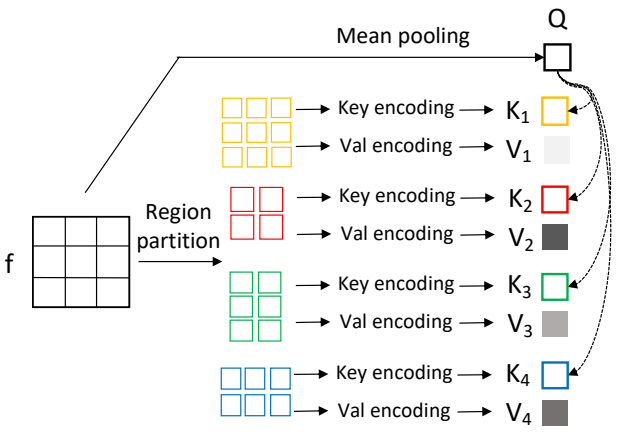

Figure 3: Region self-attention. Query, key and value are generated given a feature map $\mathbf{f}$. The global representation is generated by region self-attention.

Training Loss. When a reading operation is made, we obtain a retrieved prediction. The loss is calculated to evaluate the distance between the retrieved predictions and the ground-truth label. We use a mean squared loss (MSE),

$$\text{MSE}(p, y) = ||p - y||_2^2, \tag{5}$$

where p is the prediction and y is the ground-truth. Both the key memory and the value memory are updated via back-propagation.

3.2. Region Self-Attention

In this section, we introduce a multi-level region encoding mechanism to extract representative features for subsequent recognition. The global feature aggregated by average pooling is a single compact vector, while the local region information is ignored at the pooling stage. It is beneficial to exploit local features for each image and utilize the most discriminative feature for improving the recognition performance of tail categories. Liu *et al*. [21] proposed modulated attention to locate discriminative cues from spatial features. The motivation is that discriminative region information is distributed in various locations. However, they still leveraged a single global vector after the attention process. We instead propose a new region self-attention (RSA) mechanism to extract local features from the feature map. Contextual relationships are considered during feature encoding. When the attention weight is learned, we keep region features at multiple scales in the IEM. Our IEM provides stronger discriminative features for all categories, improving the recognition ability at the feature-level.

RSA is inspired by [19], and it is a variant of self-attention. RSA produces local features and encodes region information effectively. During training, we insert RSA at the last convolutional block before the final classification. Our region encoding function produces region statistics like mean, variances, region shape. The region statistics are aggregated to generate a feature for regions at each scale.

Region Partition. We first divide the original feature map into multi-scale parts. We introduce multiple kernels to scan over the whole feature map to incorporate multi-scale features. We denote a feature map as $\mathbf{f}$. H and W are the height and width of the feature map, respectively. Each position has feature of $\mathbf{f}_{i,j}$, where $i = \{1, \ldots, H\}$ and $j = \{1, \ldots, W\}$. The maximum allowed region on the feature map is of shape $h_{max} \times w_{max}$. The RSA kernels are $(k_h,\ k_w)$, where $k_h = \{1, 2, \ldots, h_{max}\}$ and $k_w = \{1, 2, \ldots, w_{max}\}$. In Figure 2, we illustrate multi-scale kernels on a 3×3 feature map, and the maximum allowed region is 2×2. We introduce four kernels, *i.e*., 1×1, 1×2, 2×1, 2×2. Each kernel divides the region at different scales and covers diverse multi-scale information.

Region Feature Encoding. We illustrate the encoding process for each partitioned region (Figure 3). We denote a region as r with height r_h and width r_w. The region feature is denoted as $\mathbf{r}$. We obtain the (key, value)=($\mathbf{k}$, $\mathbf{v}$) pair by transforming $\mathbf{r}$ with a linear layer. $\mathbf{k}_{i,j}$ is used to produce a region feature by a region encoding function. The region encoding function considers the size of the region and incorporates feature variances. The most straightforward method to aggregate region features is to simply average or sum them all. We denote the summation operator *sum* as $\sum_{i=1;j=1}^{i=r_h;j=r_w} \mathbf{k}_{i,j}$.

We introduce a richer and stronger representation to encode each region key,

$$\mu = \frac{1}{r_h \times r_w} \sum_{i=1;j=1}^{i=r_h;j=r_w} \mathbf{k}_{i,j}, \tag{6}$$

$$\sigma = \sqrt{\frac{1}{r_h \times r_w} \sum_{i=1;j=1}^{i=r_h;j=r_w} (\mathbf{k}_{i,j} - \mu)^2}, \tag{7}$$

$$p = \text{Concat}[\text{One-hot}(r_h)\mathbf{W}_h, \text{One-hot}(r_w)\mathbf{W}_w], \tag{8}$$

$$g = \text{ReLU}(\text{Concat}[\mu, \sigma, p]\mathbf{W}_o)\mathbf{W}_d, \tag{9}$$

where μ is the mean feature of region r, σ is the standard deviation of vectors within this region, and $\mathbf{W}_*$ are learnable weights. μ and σ are two important statistics about this region. We leverage σ to show the variances in the region. We incorporate the region shape as a region feature by encoding r_h and r_w with one-hot encoding. The one-hot vectors are embedded by an embedding matrix. We concatenate the height vector and the width vector to be the region shape representation p. When we obtain μ, σ and p to reflect region features, we generate the final region representation by concatenating them, followed by a linear layer.

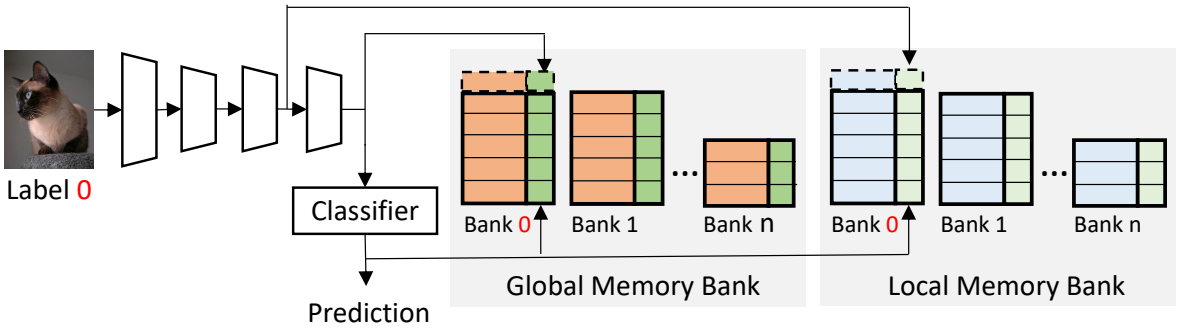

Figure 4: The features and confidences are written to IEM. Given an image with label 0, global memory 0 and local memory 0 are updated. The global memory stores global representations and the local memory records region features.

It is further activated by a ReLU activation and transformed by a linear layer. The final region representation g extracts helpful features from local regions. We denote the process as Region-Key-Encoding.

The region self-attention process is illustrated as follows. We obtain the query by mean pooling the whole feature map $\mathbf{f}$. The key map for each region r is obtained by encoding the whole region with the aforementioned region key encoding mechanism,

$$\mathbf{Q} = \frac{1}{H \times W} \sum_{i=1;j=1}^{i=H;j=W} \mathbf{f}_{i,j}, \tag{10}$$

$$\mathbf{K} = \text{Region-Key-Encoding}(\mathbf{r}), \tag{11}$$

$$\mathbf{V} = \sum_{i=1;j=1}^{i=r_h;j=r_w} \mathbf{v}_{i,j} \tag{12}$$

We then follow the standard self-attention mechanism (SA) to obtain the global representation for the whole feature map,

$$\text{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{13}$$

where d is the input channel size.

3.3. IEM for Long-tailed Classification

In this section, we explain the use of the aforementioned modules for long-tailed visual recognition. For a dataset with n classes, we leverage $2 \times n$ IEM banks. There are two IEM blocks for each category, *i.e.*, a global IEM and a local IEM. The global block stores the global representation calculated by global average pooling. Meanwhile, the local IEM saves the region features from the feature vectors encoded by region self-attention mechanism (Section 3.2).

Memory Warmup. We first illustrate the memory warmup stage. Initially, the memory is randomly initialized. At this stage, both global and local visual features are incorporated into the global memory bank and local memory bank, respectively (Figure 4). Specifically, given an image x with label y, the classifier generates a logits $\mathbf{y}'$ with n dimensions. We leverage a convolutional network to obtain feature $\mathbf{f}$. The region encodings are $\mathbf{r}_1, \ldots, \mathbf{r}_c$, where c is the number of generated regions. The key-value pair of $(\frac{1}{H \times W} \sum_{i=1;j=1}^{i=H;j=W} \mathbf{f}_{i,j}, \mathbf{y}'[y])$ is appended to the global memory block. For local region features, there are pairs of $\{(\mathbf{r}_1, \mathbf{y}'[y]), (\mathbf{r}_2, \mathbf{y}'[y]), \ldots, (\mathbf{r}_c, \mathbf{y}'[y])\}$, which are appended to local memory block sequentially.

Memory Update. The loss for memory update consists of two parts. The first loss is the cross-entropy loss (CE), which is used to update the weights of the convolutional network. The second loss is the MSE loss. It is used to calculate the gradient to update all memory blocks. For an input x with label y, we not only calculate the loss for memory bank y but select a hard negative memory block $\bar{y}$ which is ranked highest in logit $\mathbf{y}'$ except the ground-truth label. We constrain the retrieved prediction to have low scores. The loss is defined as,

$$p_{pos} = \text{READ}(M_y), \tag{14}$$

$$p_{neg} = \text{READ}(M_{\bar{y}}), \tag{15}$$

$$L = \text{CE}(y, \mathbf{y}') + \text{MSE}(p_{pos}, 1) + \text{MSE}(p_{neg}, 0). \tag{16}$$

READ is the memory reading operation described in Section 3.1. The network is optimized end-to-end via back-propagation.

Inference. Given a test image at the inference stage, we iterate over all memory blocks and retrieve the prediction scores from all blocks. The n prediction scores are averaged with the standard classifier prediction.

4. Experiments

We evaluate our model on both long-tailed image classification and long-tailed video classification. We show that

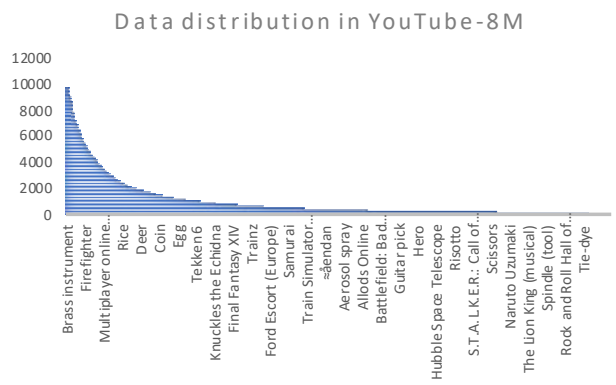

Figure 5: Data distribution on YouTube-8M. YouTube-8M is a realistic dataset with a long-tailed distribution

our model can be generalized in both images and videos.

4.1. Datasets

We conduct quantitative experiments on long-tailed image and video recognition datasets.

Long-tailed image classification. We evaluate on standard datasets, *i.e.*, ImageNet-LT [21], Places-LT [21], SUN-LT [21] for long-tailed image classification. Places-LT and ImageNet-LT are designed for long-tailed recognition evaluation, and they are sampled from the original balanced datasets. The dataset details can be found in [21]. We follow [21] to construct the ImageNet-LT dataset. In ImageNet-LT, there are 1K categories and 115.8K images, with maximally 1,280 images per class and minimally 5 images per class. The test set is balanced. The open set is constructed by the additional classes of images in ImageNet-2010. We conduct experiments on iNaturalist 2018 [14], which is a fine-grained object recognition dataset with high-class imbalance.

Long-tailed video classification. We use YouTube-8M [1] for long-tailed video classification. YouTube-8M is a realistic dataset with the long-tailed distribution. It contains diverse videos with 3,696 classes. Each video may have multiple classes. The average video length is 200 second, and the maximum video length is 300 second. [1] provided the frame-level audio and visual features for each frame sampled at 1 FPS. This dataset is extremely imbalanced (Figure 5). The maximum number of examples per category is 788,288, while the minimum number of examples per category is 123. The ratio between the head class and the tail class is more than 5,000. In Figure 5, we removed classes that have more than 10,000 training examples to visualize the distribution clearer. We report the accuracy on the original validation set. We cross-validate the hyper-parameters on the held-out validation set, where we randomly sampled 5% videos from the training data. We used Global Average Precision (GAP) as the evaluation metric [1],

$$GAP = \sum_{i=1}^{P} p(i)\nabla r(i), \tag{17}$$

where P is the number of top predictions, $p(i)$ is the precision at prediction i, $\nabla r(i)$ is the change in the recall at prediction i. P is set to 20. We also use Mean Average Precision (mAP) as the metric. We report both metrics for YouTube-8M. We conduct experiments on YouTube-8M to demonstrate the effectiveness of our framework for video classification.

Method	Accuracy
Plain Model [12]	48.0
Cost-Sensitive [13]	52.4
Model Reg. [37]	54.7
MetaModelNet [38]	57.3
OLTR [21]	58.7
Ours	**60.2**

Table 1: Comparisons on SUN-LT. Our IEM achieves the best. We outperform OLTR [21] by 1.5%.

Method	ResNet-50
CB-Focal [4]	61.1
LDAM [4]	64.6
LDAM+DRW [4]	68.0
Ours	**70.2**

Table 2: Comparisons on iNaturalist. We achieve substantial improvements.

Method	GAP	mAP
NetVLAD baseline [23]	86.1	52.7
Ours	**87.7**	**56.5**

Table 3: Evaluation results on YouTube-8M dataset. Our IEM outperforms the baseline on both GAP and mAP significantly. We obtain a 3.8% improvement on mAP.

4.2. Implementation Details

We set the number of nearest neighbors in IEM to 50 for all experiments. For each IEM, we set the maximum memory size to 50,000. The update momentum γ is set to 0.99. We use a small learning rate of 1×10^{-5} to update the memory. The memory is updated with Adam optimizer [17]. To train the backbone network, we use stochastic gradient descent (SGD) with momentum of 0.9, and the batch size is 256. The shorter side of each image is first resized to 256, and then we randomly sample a 224 $\times$ 224 crop from the resized image. The images are randomly flipped. We train

Methods	Closed-set setting				Open-set setting			
	> 100 **Many**	$\leqslant 100$ & > 20 **Medium**	< 20 **Few**	**Overall**	> 100 **Many**	$\leqslant 100$ & > 20 **Medium**	< 20 **Few**	**F-measure**
Plain Model [12]	40.9	10.7	0.4	20.9	40.1	10.4	0.4	0.295
Lifted Loss [24]	35.8	30.4	17.9	30.8	34.8	29.3	17.4	0.374
Focal Loss [20]	36.4	29.9	16	30.5	35.7	29.3	15.6	0.371
Range Loss [39]	35.8	30.3	17.6	30.7	34.7	29.4	17.2	0.373
FSLwF [9]	40.9	22.1	15	28.4	40.8	21.7	14.5	0.347
OLTR [21]	43.2	35.1	18.5	35.6	41.9	33.9	17.4	0.474
Ours	**48.9**	**44.0**	**24.4**	**43.2**	**46.1**	**42.3**	**20.1**	**0.525**

(a) Classification results on ImageNet-LT.

Methods	Closed-set setting				Open-set setting			
	> 100 **Many**	$\leqslant 100$ & > 20 **Medium**	< 20 **Few**	**Overall**	> 100 **Many**	$\leqslant 100$ & > 20 **Medium**	< 20 **Few**	**F-measure**
Plain Model [12]	45.9	22.4	0.36	27.2	45.9	22.4	0.36	0.366
Lifted Loss [24]	41.1	35.4	24	35.2	41	35.2	23.8	0.459
Focal Loss [20]	41.1	34.8	22.4	34.6	41	34.8	22.3	0.453
Range Loss [39]	41.1	35.4	23.2	35.1	41	35.3	23.1	0.457
FSLwF [9]	43.9	29.9	29.5	34.9	38.1	19.5	14.8	0.375
OLTR	44.7	37	25.3	35.9	44.6	36.8	25.2	0.464
Ours	**46.8**	**39.2**	**28.0**	**39.7**	**48.8**	**42.4**	**28.9**	**0.486**

(b) Classification results on Places-LT.

Table 4: Evaluation results on ImageNet-LT and Places-LT. We achieve better classification performance on both datasets, and on both the close-set and the open-set settings.

the network for 90 epochs with an initial learning rate of 0.1. We anneal the learning rate at epoch 30 and 60 [21].

For ImageNet-LT, following [21], we evaluate with the ResNet-10 model that is randomly initialized. We follow the original learning rate scheduling in [21], where the initial learning rate is set to 0.1 and decayed by 0.1 every 10 epoch. We train the model with 30 epochs. For Places-LT and SUN-LT, we leverage ResNet-152, and the initial learning rate is 0.01. For iNaturalist, we train ResNet-50 with an initial learning rate of 0.1. The total training epoch is 90, and the learning rate is annealed at epoch 30 and 60.

For YouTube-8M, we train a NetVLAD model with the same set of hyperparameters in [23]. We train it with 256 clusters and the size of the hidden layers is 2,048. During training, we use a batch size of 80. The initial learning rate of 0.0002 is used. The learning rate is exponentially decayed at the rate of 0.8.

4.3. Experimental Results

4.3.1 Long-tailed Image Classification

The results on ImageNet-LT and Places-LT are shown in Table 4. The results on SUN-LT and iNaturalist are shown in Table 1 and Table 2, respectively.

The effectiveness of IEM is validated across various datasets. We obtain substantial improvements. For instance, we outperform OLTR by 7.6% in the closed-set setting on ImageNet-LT. We obtain a 3.8% improvement on overall classification on Places-LT closed-set setting. Significant improvements are also observed in the open-set settings. For instance, we outperform OLTR [21] by 0.051 on F-measure on the open-set of ImageNet-LT.

For the cases of tail classes ("few"), we observe a significant improvement compared to OLTR. For example, on the closed-setting of Places-LT, we obtain 3.3% performance gain. For the case of ImageNet-LT, the improvement is 5.9%. The results show IEM with region self-attention is extremely helpful for tail class recognition. It shows that our IEM can learn from tail classes more effectively. Similar improvements can be obtained in SUN-LT (Table 1). In addition to ImageNet-LT, Places-LT, and SUN-LT, we conduct experiments on iNaturalist, which is a more natural long-tailed dataset. Notably, we achieve a 2.2% improvement compared to [4]. All these results clearly show that our IEM with region self-attention can alleviate the effect of imbalanced data distribution. The learned model also generalizes better in the open-set setting.

4.3.2 Long-tailed Video Classification

We also conduct experiments on YouTube-8M, which is a video classification dataset. We apply IEM to the NetVLAD

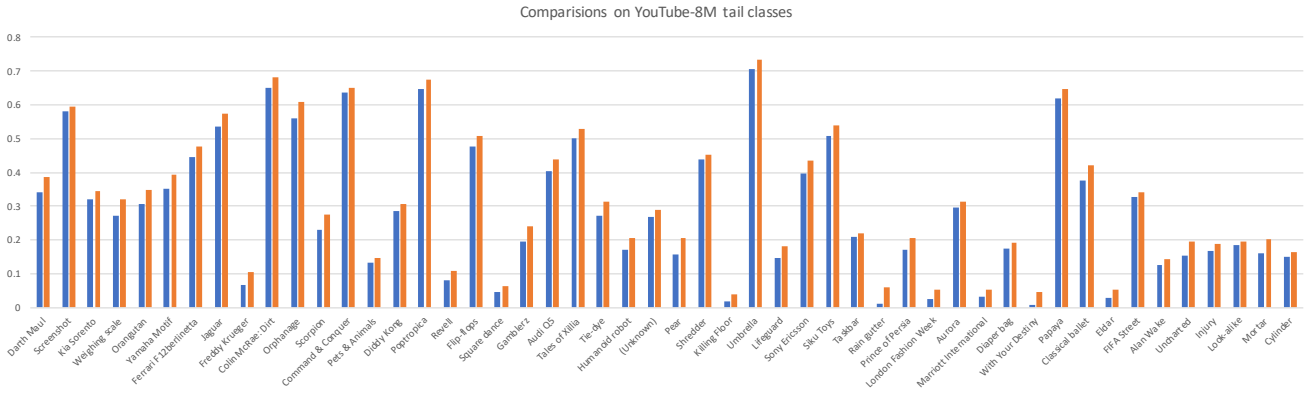

Figure 6: Comparisons on YouTube-8M tail classes. We can observe a consistent accuracy improvement on the tail classes. Note that there are tail classes that have low accuracy, showing the difficulties in modeling long-tailed data.

Method	ImageNet-LT	Places-LT	SUN-LT	iNaturalist
w/o RSA	41.7	31.1	59.3	69.5
w/o local memory	38.1	34.6	57.9	67.0
w/o global memory	40.8	36.9	58.9	68.1
Ours	**43.2**	**39.7**	**60.2**	**70.2**

Table 5: Ablation studies across the datasets. We study the effectiveness of the region self-attention mechanism (RSA), the effectiveness of local IEM, and the effectiveness of global IEM. The results demonstrate the effectiveness of RSA components and the design of IEM.

network. Note that the provided features for YouTube-8M are compact vectors. We are unable to obtain the original spatial feature map for each video. We do not leverage local memory for YouTube-8M. The results are shown in Table 3. Notably, compared to the NetVLAD baseline, we achieve a 4.8% improvement on the YouTube-8M dataset. It demonstrates the effectiveness of our IEM in modeling long-tailed distributions. Furthermore, the GAP metric is also improved, which demonstrates that IEM can improve overall generalization on both head and tail classes. The improvements on tail classes are shown in Figure 6. We observe consistent improvements compared to the baseline.

4.4. Ablation Studies

We study some key elements in our IEM. We mainly study the effectiveness of the region self-attention mechanism, the effectiveness of local IEM, and the effectiveness of global IEM. To demonstrate the effectiveness of the region self-attention mechanism, we replace the region self-attention mechanism with a simple average pooling function. For ImageNet-LT and Places-LT, we conduct the ablations on the closed-set setting.

The results are shown in Table 5. We observe that local IEM is essential to the success of long-tailed classification, where local features are helpful for learning discriminative features from a few examples. The results show that our RSA is beneficial to encode multi-scale region features. When it is replaced by a simple average pooling, the performance drops across all the datasets. Note that the global memory is also an important component. When the global memory is removed, the performance drops more than 2% on ImageNet-LT, Places-LT, and iNaturalist. It shows that global information is important for classification on these datasets. Global representation offers a straightforward view for recognition as some classes are about general scenes. These results demonstrate the effectiveness of our IEM and RSA.

5. Conclusion

In this paper, we introduce a novel Inflated Episodic Memory (IEM) module for long-tailed visual recognition. IEM augments convolutional neural networks with categorical representative features. We investigate the effectiveness of region self-attention (RSA) for region feature encoding. We validate the effectiveness of IEM and RSA on both long-tailed image classification and long-tailed video classification. In the future, we will focus on designing better re-sampling strategies that complement our framework.

Acknowledgements. This work is supported by ARC DP200100938.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 2, 6

[2] Rohit Babbar and Bernhard Schölkopf. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, 2019. 1

[3] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 1975. 3

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019. 2, 6, 7

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2

[6] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. Dnn or k-nn: That is the generalize vs. memorize question. *arXiv preprint arXiv:1805.06822*, 2018. 1

[7] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, 2018. 2

[8] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *ICCV*, 2017. 2

[9] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 2, 7

[10] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 1

[11] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6, 7

[13] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016. 6

[14] iNatrualist. The inaturalist 2018 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2018, 2018. 6

[15] Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *arXiv preprint arXiv:1703.03129*, 2017. 2, 3

[16] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[19] Yang Li, Lukasz Kaiser, Samy Bengio, and Si Si. Area attention. *arXiv preprint arXiv:1810.10126*, 2018. 4

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 7

[21] 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. 2, 4, 6, 7

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[23] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 2, 6, 7

[24] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 7

[25] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *ICML*, 2017. 2, 3

[26] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard S Zemel. Incremental few-shot learning with attention attractor networks. *arXiv preprint arXiv:1810.07218*, 2018. 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[29] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 2

[30] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 2

[31] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2

[32] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *NeurIPS*, 2015. 2, 3

[33] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2

[34] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 2

[35] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 2

[36] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. 2, 3

[37] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*. Springer, 2016. 6

[38] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, 2017. 2, 6

[39] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, 2017. 7

[40] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *CVPR*, 2019. 2

[41] Linchao Zhu, Zhongwen Xu, and Yi Yang. Bidirectional multirate reconstruction for temporal modeling in videos. In *CVPR*, 2017. 2

[42] Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. Uncovering the temporal context for video question answering. *IJCV*, 124(3):409–421, 2017. 1

[43] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *ECCV*, 2018. 2