Retrieval Augmented Classification for Long-Tail Visual Recognition*

Alexander Long[1,3], Wei Yin[2], Thalaiyasingam Ajanthan[1], Vu Nguyen[1], Pulak Purkait[1], Ravi Garg[1], Alan Blair[3], Chunhua Shen[4], Anton van den Hengel[1,2]

[1] Amazon [2] The University of Adelaide, Australia [3] University of New South Wales [4] Zhejiang University, China

Abstract

*We introduce Retrieval Augmented Classification (RAC), a generic approach to augmenting standard image classification pipelines with an explicit retrieval module. RAC consists of a standard base image encoder fused with a parallel retrieval branch that queries a non-parametric external memory of pre-encoded images and associated text snippets. We apply RAC to the problem of long-tail classification and demonstrate a significant improvement over previous state-of-the-art on Places365-LT and iNaturalist-2018 (*14.5% *and* 6.7% *respectively), despite using only the training datasets themselves as the external information source. We demonstrate that RAC's retrieval module, without prompting, learns a high level of accuracy on tail classes.This, in turn, frees the base encoder to focus on common classes, and improve its performance thereon. RAC represents an alternative approach to utilizing large, pretrained models without requiring fine-tuning, as well as a first step towards more effectively making use of external memory within common computer vision architectures.*

1. Introduction

Large Transformer [48] models have arrived in Computer Vision, with parameter counts and pretraining dataset size increasing rapidly [11,26,34,42,44,53]. The distributed representations learned by such models result in significant performance gains on a range of tasks, however come with the drawback of storing world knowledge implicitly within their parameters, making post-hoc modification [8] and interpretability [4] challenging. In addition, real-world data is long-tailed by nature, and implicitly storing every visual cue present in the world appears futile with current hardware constraints. As an alternative to this fully parametric approach, we propose augmenting standard classification pipelines with an explicit external memory, thus separating model performance from parameter count, and facilitating the dynamic addition and removal of information explicitly with no changes to model weights.

To evaluate our approach, we focus on the problem of Long-Tail visual recognition, as it shares many of the properties likely to be encountered by a general agent. Specifically, the data distributions are highly skewed on a per-class basis, with a majority of classes containing a small number of samples. The number of samples in these small classes, commonly referred to as the "tail", can far outweigh those in the relative minority of high sample classes (referred to as the "head"). In this situation, learning is challenging due to both the lack of information provided for tail classes, and the tendency for head classes to dominate the learning process. Long-tail learning is a well-studied [2, 20, 39] instance of the more general label shift problem [41], where the shift is static and known during both training and testing. Despite being well-studied, commonly occurring, and of great practical importance, classification performance on long-tail distributions lags significantly behind the state-of-the-art for better balanced classes [24].

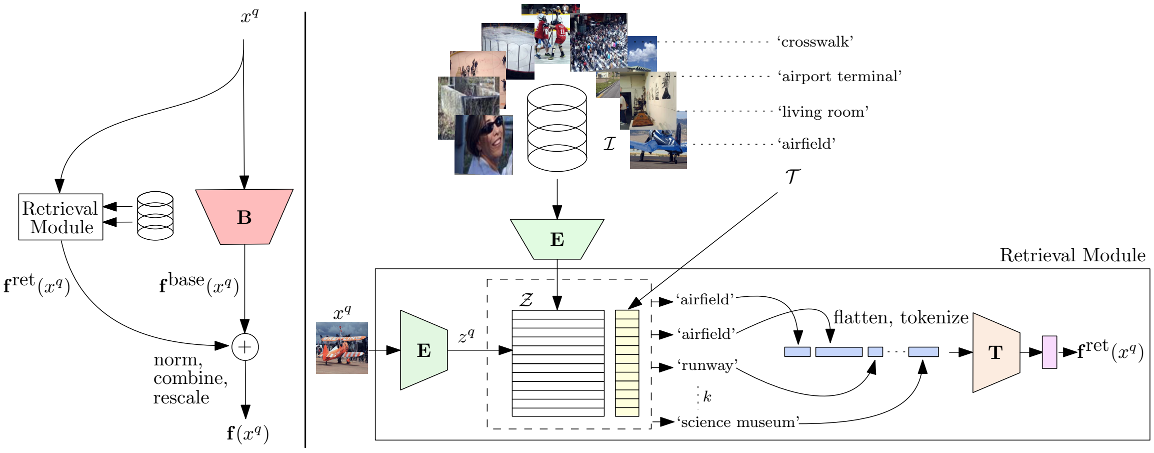
Base approaches are largely variants of the same core idea—that of "adjustment", where the learner is encouraged to focus on the tail of the distribution. This can be achieved implicitly, via over/under-weighting samples during training [3, 12, 19, 22] or cluster-based sampling [6], or explicitly via logit [10, 36, 57] or loss [21, 36] modification. Such approaches largely focus on *consistency*, ensuring minimizing the training loss corresponds to a minimal error on the known, balanced, test distribution.

An alternative approach focuses on ensembling models. Instead of disregarding knowledge of the test distribution, recent work [17, 52, 59] use ensembling models to induce invariance to the test distribution. This is typically done by training separate models under different losses or re-sampling techniques, and combining them at test time.

We introduce a third approach, Retrieval Augmented Classification (RAC), motivated by the desire to explicitly store tail knowledge, as a retrieval-based augmentation to standard classification pipelines.

RAC's retrieval module is multi-modal, making use of image representations as retrieval keys, and returning en-

*Part of this work was done when WY was with Amazon and CS was with The University of Adelaide.



(a) The proposed RAC architecture

(b) Retrieval module

Figure 1. (a) RAC overview. RAC consists of a retrieval module that augments a standard encoder $\mathbf{B}(\cdot)$ with explicit external memory. (b) The retrieval module consists of external images $\mathcal{I}$ encoded by a fixed, pretrained image encoder $\mathbf{E}(\cdot)$, and associated text $\mathcal{T}$ queried using an approximate k-NN and encoded via a text encoder $\mathbf{T}(\cdot)$. The logits of the retrieval encoder are then combined with those of the base network. In our instantiation, $\mathbf{B}$ and $\mathbf{E}$ are ViT's, and $\mathbf{T}$ is a BERT-like text encoder.

coded textual information associated with each image. We place no limitation on the nature of this text; it may be the labels from a supervised training set, descriptions, captions etc. In the simplest case, the images in the index, and associated text, can be the images and labels from the dataset of interest alone.

RAC jointly trains a standard base encoder, and a separate retrieval branch. We demonstrate empirically that the retrieval branch learns, without explicit prompting, to focus on tail classes. This frees the base encoder from modelling these sparse classes, as they are already effectively represented by the non-parametric memory of the retrieval module. This in turn allows the base encoder to achieve a higher level of performance on the head classes.

RAC achieves state-of-the-art performance on common Long-Tail classification benchmarks, even out-performing approaches such as LACE [36] that are provably consistent with regard to the class-balanced error, and Bayes-optimal under Gaussian class priors. *A major benefit of RAC is its ability to use large, pretrained models for inference* (for index and retrieval encoding), leveraging their rich representations to improve the classification performance of a base learner. This broadens the applicability of such models due to the large cost of fine-tuning.

Our contributions are summarised as follows:

1. The first demonstration of effective external memory within long-tail visual recognition setting.
2. A novel method for Long-Tail classification that significantly improves on the current state-of-the-art.
3. Insight into the proposed method, with the reimplementation of strong baselines that also exceed current state-of-the-art.

2. Related Work

Resampling and Logit Adjustment Over-sampling sparse classes [3, 19] is one of the oldest approaches to addressing distribution bias, but one that is still in common use. Under-sampling common classes [12], applying additional data-augmentation to sparse classes in pixel, or feature space [5, 32], or sampling uniformly from pre-computed clusters [22], have also been suggested. Hong *et al*. [21] propose a distribution aware weight regularizer that is applied more heavily to head classes than tail classes, in a similar vein to weight normalization. However, empirically, the resulting model (LADE) only produces marginal gains over straight-forward balanced softmax. Recent work [36] has unified many empirically successfully approaches under a Fisher-consistent scorer for the balanced error, and additionally shown that weight normalization fails when used with the ADAM optimizer. Zhang *et al*. [54] adopt a two stage approach, and propose a class-specific learnable (from the samples) reweighting (via a single layer NN) of the frozen pretrained logits based on a generalized formulation of the class-balanced softmax. They show that transforming the classification head, as opposed to re-training it, performs better. PaCo [6], the current state-of-the-art for long-tail classification, combines learnable logit adjustment with contrastive learning [38]. Despite their simplicity, adjusted logit methods (LACE, LDAM, LADE) remain strong solutions to the long tail problem, typically achieving within 1-2% top-1 accuracy of state-of-the-art ensemble approaches (see Table 1, Table 2).

Ensemble Methods In proposing TADE [56], Zhang *et al*. explicitly train three heads with standard, balanced, and inversely weighted softmax losses, linearly combining their predictions at test time, weighted by a measure of confidence derived from each head's stability under data-augmentation. Wang *et al.* [50] in contrast combine multiple independently trained classification heads that are pushed to be decorrelated in their predictions via a (class balanced) KL loss, with a small routing network that improves computational efficiency during inference.

External Memory One of the first models to successfully combine deep networks with external memory was the Neural Turing Machine [15]. The purpose of that model was symbolic manipulation, however, which renders its architecture quite different to that of RAC. Gong *et al.* [14] proposed a similar retrieval-module architecture for anomaly detection, but without RAC's corresponding base module. Recently, in the NLP domain, several works have proposed the augmentation of large language models with a non-parametric memory to allow explicit access to external data [18, 30]. While such approaches make use of differential retrievers, which introduces the problem of lookup/representation drift, they are still closely related to RAC. k-NN Language Models (LMs) [27] are most similar to our work, which directly interpolate a retrieval distribution with the next token distribution produced by a base LM, resulting in reduced combined model perplexity.

Latent retrieval has been applied to textual open-domain QA [25, 29]. The central difference is such approaches return information that is most similar to the retrieval key, whereas RAC returns information (text) attached to retrieved samples. An approach similar to that of RAC has been applied to knowledge-intensive QA [49], where a 'fact memory' consisting of triples from a symbolic Knowledge Base (KB) is directly encoded and queried using the final representation of a language model as keys. In computer vision, non-parametric retrieval has been used to assist in addressing the fine-grained retrieval problem, such as in enforcing instance-level retrieval loss in [46]. The Open-world Long-tail model proposed in [33] also makes use of a retrieval module, but primarily as a mechanism to distinguish between seen, and unseen samples in the 'open world' setting, not to boost performance on seen classes as we do.

3. Method

3.1. Preliminaries

In long-tailed visual recognition, the model has access to a set of N training samples $\mathcal{S} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, where $\mathbf{x}_n \in \mathcal{X} \subset \mathbb{R}^D$ and labels $\mathcal{Y} = \{1, 2, .., L\}$. Training class frequencies are defined as $N_y = \sum_{(x_n, y_n)\in\mathcal{S}} \mathbb{1}_{y_n = y}$ and the test-class distribution is assumed to be sampled from a uniform distribution over $\mathcal{Y}$[1], but is not explicitly provided during training. The goal is thus to minimize the balanced error, of a scorer $\mathbf{f} : \mathcal{X} \to \mathbb{R}^L$, defined as;

$$\text{BE}(\mathbf{x}, \mathbf{f}(\cdot)) = \sum_{y\in\mathcal{Y}} \mathbf{P}_{\mathbf{x}|y} \left(y \notin \operatorname*{arg\,max}_{y'\in\mathcal{Y}} \mathbf{f}_{y'}(\mathbf{x}) \right) \tag{1}$$

where $\mathbf{f}_y(x)$ is the logit produced for true label y for sample $\mathbf{x}$. Traditionally this is done by minimizing a proxy loss, the Balanced Softmax Cross Entropy (BalCE):

$$\ell_{\text{BalCE}}(\mathbf{x}, y, \mathbf{f}_y(\cdot)) = -\frac{1}{N_y} \log \frac{e^{\mathbf{f}_y(\mathbf{x})}}{\sum_{y'\in\mathcal{Y}} e^{\mathbf{f}_{y'}(\mathbf{x})}}. \tag{2}$$

This is a form of *re-weighting*, where the contribution of each label's individual loss is scaled by an approximation of $\mathbf{P}(y)$, which for ℓ_{BalCE}, is the inverse class frequency. BalCE remains a strong baseline in this domain (see Sec. 4.3).

3.2. LACE Loss

An alternative to re-weighting is to adjust the logits themselves, however the two can be done in conjunction, resulting in the general form of the re-weighted (via α_y) and adjusted (via Δy) softmax cross entropy loss;

$$\ell(\mathbf{x}, y, \mathbf{f}_y(\cdot)) = -\alpha_y \log \frac{e^{\mathbf{f}_y(\mathbf{x}) + \tau\cdot\Delta y}}{\sum_{y'\in\mathcal{Y}} e^{\mathbf{f}_{y'}(\mathbf{x}) + \tau\cdot\Delta y'}} \tag{3}$$

where τ is a constant temperature scaling parameter. Several recent works focusing on long-tail learning exploit special cases of this loss. If $\alpha_y = 1$ and, Logit Adjusted Cross-Entropy (LACE) [36], can be recovered with $\Delta_y = \log(N_y/N)$ and LDAM [1] with $\alpha_y = {}^1\!/\!{}_{N_y}$ and

$$\Delta y = \begin{cases} N_y^{-1/4} & \text{if} \quad y' = y, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Both are Fisher-consistent with respect to the balanced loss. The optimal τ can be found with a holdout set, or set to 1 if the logits are calibrated [16]. In our experiments, this calibration is achieved through label smoothing [37], which has been shown to *implicitly* calibrate neural networks [45]. This property is important in the design of RAC as we use LACE as the base loss and do not apply manual temperature adjustment, setting $\tau = 1$, unless otherwise specified.

3.3. Retrieval Augmented Classification

The overall idea of RAC is very simple—split the scorer into two branches (see Fig. 1), where one branch (retrieval) exhibits implicit invariance to class frequency. The two branches are trained under a common LACE loss, with their individual logits combined with a norm, addition and rescale operation to ensure that one does not override the other during training.

[1] While this is true for Places365-LT, iNaturalist2018 has a fixed number of test samples for each class ($N_i^{\text{test}} = 3, \quad \forall i \in \mathcal{Y}$)

The base branch encoder $\mathbf{B}(\cdot)$ can be any choice of a standard backbone network. In our experiments, we primarily use the ViT-B-16 variant of Visual Image Transformer [11], transforming the final token embedding via a standard linear layer. The retrieval module (see Sec. 3.4) takes a raw image and performs a latent-space lookup on an index of precomputed embeddings, returning the text attached to the top k most similar images to the image currently being considered, $\mathbf{x}^q$. This text is then fed through a text encoder and transformed by another linear layer into logits $\mathbf{f}^{\text{ret}}(\mathbf{x}^q)$.

We make use of the pretrained BERT-like text encoder (63M parameters, 12-layer 512-wide model with 8 attention heads) from CLIP [42], which we choose due to the broad (400M) range of images, alt-text pairs used during pretraining, and the compatibility with other language models due to the preservation of masked self attention in the architecture. In our experiments, the choice of text encoder is not critical as the textual information being retrieved (labels) is not highly complex, and off-the-shelf word embeddings, and even random encodings still perform reasonably well (see Fig. 3). This choice does increase training time due to the larger parameter count (see Table 5), but allows RAC to scale to more complex retrieved text.

To combine base and retrieval branches, we normalize each branch's outputs to the unit norm and add them together. To ensure training dynamics are not altered (via lower logit magnitudes) in comparison to the baselines we rescale the combined logits by a constant factor (dependent on L due to final layer Xavier initialization [13] also being dependent on L).

$$\mathbf{f}(\mathbf{x}) = \frac{L}{2}\left(\frac{\mathbf{f}^{\text{ret}}(\mathbf{x})}{||\mathbf{f}^{\text{ret}}(\mathbf{x})||_2} + \frac{\mathbf{f}^{\text{base}}(\mathbf{x})}{||\mathbf{f}^{\text{base}}(\mathbf{x})||_2}\right), \tag{5}$$

where $\mathbf{f}^{\text{base}}(\mathbf{x})$ represents the logits produced by the image encoder backbone, and $\mathbf{f}^{\text{ret}}(\mathbf{x})$ is the output of the retrieval module. This straightforward setup has the benefit of being able to treat the branch outputs as individual logits, increasing the interpretability of RAC, and allowing us to precisely evaluate the per-class accuracy of each branch (see Fig. 2). While there are many ways to combine the branches such as confidence or distance based weightings, attention mechanisms etc., we found this approach sufficient, with the weighting of each branch done implicitly by the learned sharpness of the logits.

3.4. Retrieval Module

The retrieval module consists of a frozen pretrained image encoder $\mathbf{E}(\cdot)$, a pre-existing set of external images $\mathcal{I} = \{\mathbf{i}_j\}_{j=1}^{J}$, with associated text $\mathcal{T} = \{\mathbf{t}_j\}_{j=1}^{J}$, which may be labels, descriptions, captions etc. Unless otherwise specified, $\mathbf{E}(\cdot)$ is a ViT-B-16, pretrained on ImageNet following [43]. Prior to training RAC, the retrieval module is initialized by producing image keys $\mathcal{Z} = \{\mathbf{z}_j\}$ such that $\mathbf{z}_j = \mathbf{E}(\mathbf{i}_j)\ \forall j$, and storing the resultant representations in a fast approximate k-NN index.

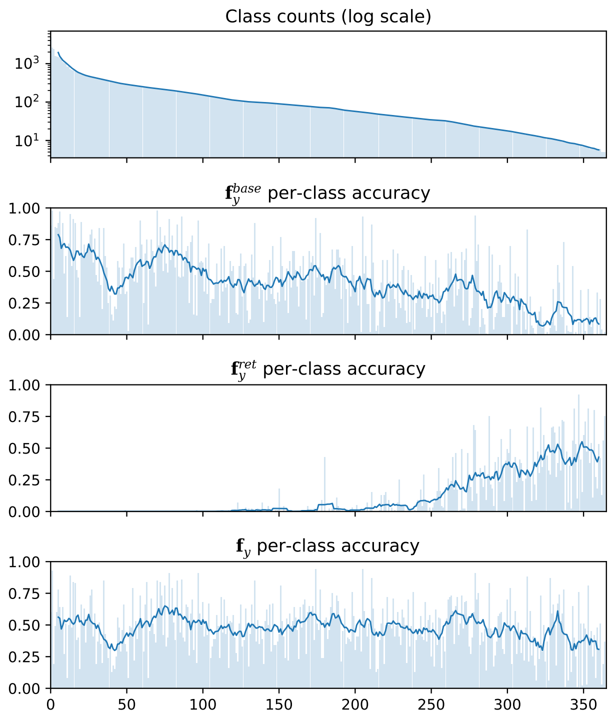


Figure 2. Per-class top-1 accuracy on Places365-LT from each branch's output. Without prompting, the retrieval module learns to focus on tail classes. The 20 sample moving average over classes (solid line) is shown for clarity.

During training, we produce features $\mathbf{z}^q = \mathbf{E}(\mathbf{x}^q)$ for each image $\mathbf{x}^q$ in the training batch. The k-NN is queried for each $\mathbf{z}^q$ and returns a list of indices of the k closest keys in $\mathcal{Z}$, where cosine similarity is the distance metric. The text element in $\mathcal{T}$ is recovered for every such index, generating k text elements for each query. These text elements are then encoded by a text encoder $\mathbf{T}(\cdot)$ which produces the retrieval branch's (fixed length) logits, $\mathbf{f}^{\text{ret}}(\mathbf{x}^q)$.

Text strings are truncated after 76 tokens, and the resultant batches are zero-padded. This approach allows for a single text-encoder call per batch, as opposed to k which would be required if each text snippet was encoded separately, and would result in a significant slowdown. The use of a large-scale transformer ensures that RAC can scale to longer text snippets if the external information is expanded to contain additional sources beyond simply labels.

A key feature of the retrieval module is its ability to include otherwise unconnected data-sources simply via their labels. In this way, we can dynamically add or remove datasets from $\mathcal{I}$, and if new examples are similar (from the point of view of the encoder), they can directly impact clas-

Method	Backbone	Many	Med	Few	All
		Input: 224 × 224			
OLTR [33]	RN50	59	64.1	64.9	63.9
Dec. LWS [24] †	RN50	65.0	66.3	65.5	65.9
LADE [21] †	RN50	-	-	-	70.0
ALA [57] †	RN50	71.3	70.8	70.4	70.7
LACE [36]	RN50	-	-	-	71.9
RIDE [50]	RN50	70.9	72.4	73.1	72.6
TADE [56]	RN50	74.4	72.5	73.1	72.9
DisAlign [54]	RN152	-	-	-	74.1
PaCo [6]	RN152	75.0	75.5	74.7	75.2
RAC (ours)	ViT-B-16	**75.92**	**80.47**	**81.07**	**80.24**
		Input: 384 × 384			
Grafit	RegNetY	-	-	-	81.2
RAC (ours)	ViT-B-16	**82.91**	**85.71**	**86.06**	**85.56**

Table 1. Historical performance on iNat under varying backbones and training schemes. †Results reproduced from [57].

Method	Backbone	Many	Med	Few	All
Focal Loss [31] †	RN152	41.1	34.8	22.4	34.6
Range Loss [55] †	RN152	41.1	35.4	23.2	35.1
OLTR [33] †	RN152	44.7	37	25.3	35.9
Dec. LWS [24] †	RN152	40.6	39.1	28.6	37.6
LADE [21] †	RN152	42.8	39	31.2	38.8
DisAlign [54]	RN152	40.4	42.4	30.1	39.3
ALA [57]	RN152	43.9	40.1	32.9	40.1
TADE [56]	RN152	43.1	42.4	33.2	40.9
PaCo [6]	RN152	36.1	47.9	35.3	41.2
RAC (ours)	ViT-B-16	**48.69**	**48.31**	**41.76**	**47.17**

Table 2. Historical performance on Places365-LT under varying backbones and training schemes. †Results reproduced from [57].

sification accuracy, providing an alternative to fine-tuning in order to incorporate new information.

For fast querying of the index, we make use of the FAISS implementation [23] of the Hierarchical Navigable Small World (HNSW) approximate k-NN lookup [35]. We construct the index with default settings aside from the hyperparameter $M = 32$, which sets the number of bidirectional links per node and increases the complexity of the index, but allows for higher recall. During training, we drop the first result, as when training data is included in the index, the first result is often the original image, which causes the text encoder to place undue weight on the first retrieved label when creating predictions.

4. Experiments

We establish RAC's high level of performance on common benchmark datasets iNaturalist2018 (Table 1) and Places365-LT (Table 2)[2] with no additional external information aside from the datasets used to pretrain the individual encoders. Note that these tables report results from the literature which were obtained under varying architectures and training schemes. We ablate the benefit of RAC's improved training pipeline in Table 3 where we reimplement class-balanced softmax Cross Entropy (BalCE) and LACE [36] as baselines. We consider 'Base only' trained under the LACE loss [36] as our primary baseline, due to LACE's strong theoretical grounding, provable consistency, and high level of previously reported empirical performance. We report overall accuracy as well as per-class accuracy bucketed into the few ($<$ 20), medium ($\leq$ 100) and many ($>$ 100) shot categories. The full per-class distribution curve is also shown in Fig. 2. We then focus specifically on the design choices of the retrieval module and how the choice of data for the index affects RAC in Sec. 4.7.

Method	Backbone	Many	Med	Few	All
		Places365-LT			
CE	RN50	-	-	-	32.14
Balanced CE	RN50	-	-	-	38.31
CE	ViT-B-16	**50.81**	33.83	19.51	37.16
Balanced CE	ViT-B-16	49.03	45.72	29.05	43.67
Retrieval	-	43.50	41.99	26.83	39.58
Base only	ViT-B-16	44.57	45.06	40.77	44.05
RAC	ViT-B-16	48.69	**48.31**	**41.76**	**47.17**
		iNaturalist 2018			
CE	RN50	-	-	-	61.7
Balanced CE	RN50	-	-	-	69.8
CE	ViT-B-16	**81.53**	76.62	69.82	74.44
Balanced CE	ViT-B-16	72.39	76.06	73.05	74.49
Retrieval	-	50.10	52.77	52.45	52.37
Base only	ViT-B-16	74.41	78.95	78.55	78.32
RAC	ViT-B-16	75.92	**80.48**	**81.07**	**80.24**

Table 3. Comparison of top-1 accuracy against baselines under a common training scheme. 'Base only' is equivalent to LACE with implicit temperature scaling via label smoothing.

In all experiments, unless otherwise indicated, **E** is a ViT-B-16 encoder, with the weights from [11]. The weights are obtained from pretraining on ImageNet21k (IM21k), a larger (11M samples) variant of the original 1.2M images ImageNet [9] dataset, with more granular classes. We make use of IM21k to expand the index in some experiments, and use the variant introduced in [28].

[2] We do not compare against CIFARLT and ImageNetLT, as in these scenarios training is typically performed from scratch, and RAC requires a pretrained network for the retrieval module. While it is possible to train the base network from scratch, this is not a fair comparison and RAC significantly outperforms other methods due to the information present in **E**.

4.1. Places365-LT

Places365-LT is a synthetic long-tail variant of Places-2 [60] introduced in [33]. It consists of 365 high-level scene classes such as 'airport', 'basement', etc. across 62.5K samples at 256×256 resolution. The minimum number of samples per class is 5, with a training set that, while balanced, is not perfectly uniform. The dataset contains a significant amount of label noise, which makes it appealing, as logit adjustment methods typically assume fully separable classes in their theoretical motivation.

We observe that with no explicit prompting, the retrieval network learns to highly skew its accuracy towards the few-shot classes (Fig. 1), confirming our hypothesis that it will be beneficial in this case. Note that there is no explicit signal pushing the retrieval network to learn infrequent classes over common ones, or for the supervised network to prefer common classes, as both are trained under the common LACE loss. Interestingly, RAC's learned strategy is similar to the hard-coded ensembling utilized in TADE [56], which is the previous state-of-the-art.

4.2. iNaturalist-2018

iNaturalist-2018 (iNat) [47] consists of 437K images and 6 levels of label granularity (kingdom, genus etc.). Following other work, we consider only the most granular labels (species), which constitutes 8142 unique classes with a naturally occurring class imbalance. In many cases the labels are very fine-grained, making it a challenging dataset even without the long-tailed property. The test set is perfectly balanced, with 3 samples per class.

In addition to the 224×224 resolution commonly studied, we report the results with 384×384 images, which was used in GRAFIT [46] and is currently state-of-the-art for this task. We found the use of 16×16 patch size to be of major importance on iNat, boosting retrieval accuracy by 21.6% (see Table 4), likely due to the fine-grained nature of the dataset.

4.3. Ablation

Historical performance numbers reported in Tables 1 and 2 contain a mix of backbones and training schemes that make comparison difficult as a modernized training scheme alone can significantly boost performance [51]. In Table 3 we baseline RAC's performance against class-balanced softmax cross entropy (BalCE) and with the retrieval branch removed ('Base only'). In this case, 'Base only' is equivalent to LACE augmented with implicit temperature scaling via label smoothing. Even against this very strong baseline (already far past state-of-the-art), RAC improves on overall accuracy by 7.01% for Places and 2.45% for iNat.

One question is how RAC is able to so outperform methods that are provably consistent, such as LACE [36]. We hypothesize that, in addition to the non-convexity introduced by using Neural Networks as the scorer, this is due to the fact that sample frequency alone does not indicate classification 'difficulty' from the perspective of a balanced learner [58]. Instead, a small number of samples may still define a sufficiently clear decision boundary if the volume of semantic space covered by that class is small and distinct [7], and hence in a truly balanced model, both inter- and intra-class distributions must be considered. Accounting for the intra-class distribution being difficult, however, given no prior on this quantity is typically provided, aside from the labels themselves. Instead, the majority of prior work has either ignored this factor, or assumed the class distributions to be Gaussian. In our formulation, these "easy" classes get picked up by the retrieval model, leaving the base branch to focus on examples that are difficult, where the difficulty is a combination of presentation frequency and class complexity. Previous methods have attempted this by correlating stability under augmentation with confidence [56], however, this correlation is weak.

Encoder	Many	Med	Few	All	CT(m:s)
		Places365-LT			
RN50	31.73	16.28	8.65	20.34	0:46
RN152d	33.52	17.71	10.03	21.89	2:07
ViT-B-32	38.34	24.82	15.83	27.92	0:20
ViT-B-32*	39.95	26.12	16.87	29.28	0:49
ViT-B-16	39.97	26.74	18.65	29.91	0:53
ViT-B-16*	40.79	27.23	19.25	30.55	3:15
		iNaturalist 2018			
RN50	26.8	20.8	21.15	21.56	5:14
RN152d	38.95	29.56	28.45	30.09	17:50
ViT-B-32	48.14	43.69	44.1	44.31	2:35
ViT-B-16	59.38	53.92	52.42	53.89	4:19
ViT-B-16*	66.15	61.54	60.92	61.77	22:14

Table 4. Analysis of standard retrieval performance and Construction Time (CT) which includes both image encoding and HNSW indexing. $^*384 \times 384$ resolution variants.

4.4. Retrieval

Quantifying retrieval accuracy is important because if standard retrieval performance is significantly lower than that of a balanced supervised learner such as the LACE baseline, it is unlikely to be beneficial. In Table 4 we perform standard retrieval with ImageNet pretrained encoders on both datasets, encoding the training set and then querying it with encoded test images, returning the label of the closest image in the training set as the prediction. All comparisons are done on exact match indexes with the ℓ_2 distance, no data augmentation and consistent crop, interpola-

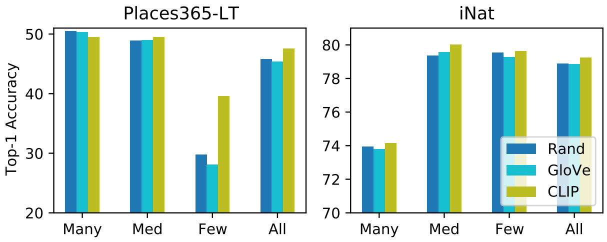


Figure 3. Effect of the choice of text encoder on performance. The overall impact is minor, however the CLIP LM significantly boosts few-shot performance on Places365-LT, where labels are natural language terms.

tion and normalization constants. z_q has length 2048 for the ResNet models, and 768 for ViTs. Despite being trained on the same data, we show that ViTs significantly outperform ResNets, and are hence critical to RAC's performance.

4.5. Importance of the Text Encoder

RAC makes use of a large BERT-like text encoder to learn a mapping from retrieved labels to class logits. Here we quantify the importance of this model relative to two alternatives: (i) Bag-of-words (BoW) GLoVe [40] embeddings, and (ii) BoW cached random embeddings.

Both are of dimension 300 vs. 512 for the CLIP encoder. The random embeddings are sampled from a uniform distribution over the interval $[0, 1)$ and cached for each word in the input string. That is, the embeddings for individual words are consistent, but have no inherent semantics.

We observe that a higher capacity $\mathbf{T}$ does improve performance, particularly on the Places365-LT few-shot classes, but that overall this benefit is minor. This is likely due to the input to the encoder not being overly complex, and more detailed information such as captions were returned, this effect may be more pronounced.

4.6. Effect of k

Given that our choice of k in approximate k-NN search is larger than the minimum number of samples present per-class for both Places365-LT and iNat, we question whether the additional returned samples, which cannot be the correct class (in the few-shot case), degrade retrieval performance. To study this, we experimented on only the retrieval branch, with no base encoder, and utilized an index that contained the training set only. As can be seen in Fig 4, increasing k consistently increases accuracy, indicating the text encoder $\mathbf{T}(\cdot)$ is able to learn to disregard the common classes. Note the few-shot performance is low here, as the retrieval branch is still trained under the LACE loss, and hence pushed towards balanced performance across all classes. It is thus not free (via the base branch) to focus on the tail classes. This indicates that newer transformer architectures, that facilitate longer sequence length, may be beneficial when applied to RAC, especially when the associated text snippets themselves are longer.

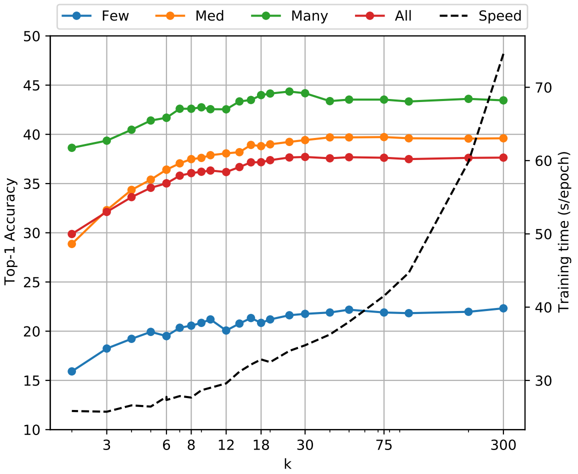


Figure 4. Effect of the number of retrieved text snippets, k, on Places365-LT top-1 accuracy for the retrieval only branch, querying an index containing only the Places365-LT training set. Higher k consistently improves performance until the cutoff induced by the text encoding truncation (76 tokens), however it does come at the cost of (linearly) higher training time. We choose $k = 30$ in our experiments. x-axis is log-scaled.

4.7. Impact of Index Content

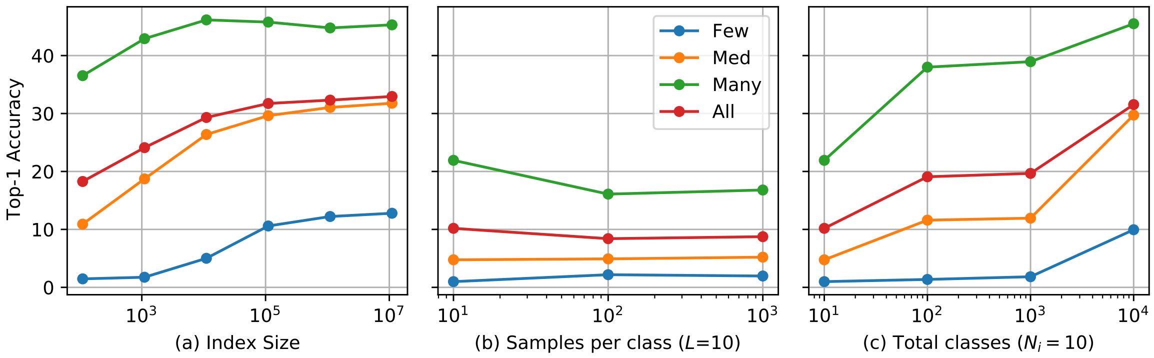
To quantify how index content affects RAC, we carried out three experiments in which we trained only the retrieval module on the Places365-LT dataset, with variants of the ImageNet21k dataset used for the index. Training was done with the same final LACE loss as complete RAC, with a ViT-B-16 as $\mathbf{E}$. Specifically, we alter:

(a) the index size via directly sub-sampling from the full ImageNet21k dataset.

(b) the number of training examples per-class while keeping the number of classes constant.

(c) the number of classes while keeping the number of sample per class constant.

Results are shown in Fig. 5. While naively increasing index size does increase performance, this effect diminishes as more samples are added. This is likely caused by the information content of the labels passed to $\mathbf{T}$ not increasing—as once most labels are present $\mathbf{E}$ is more likely to find a similar image, however from the perspective of $\mathbf{T}$, which is not distance or image aware, the information is the same. This is supported by sub-figures (b) and (c), in which the total amount of samples in the index is increased consistently between both plots, but adding samples by via new labels



(a) Index Size

(b) Samples per class (L=10)

(c) Total classes ($N_i = 10$)

Figure 5. Effect of index content on performance ($k = 30$) on the retrieval branch only, trained under the LACE loss on Places365-LT. Here, the index contains no Places365-LT data, only variants of the ImageNet21k dataset.

has a disproportionately larger effect than adding new samples with the number of labels constant. This is promising in that it indicates increasing label granularity, through the use of image captions or associated text, is likely to increase RAC's performance even further.

4.8. Runtime Consideration

Nearest neighbour searches can be computationally infeasible for large datasets. We show here, however, that a lookup over a sample index of size >10M can be performed for each training sample with negligible overhead, although the additional label encoding (and subsequent backprop), does increase the training time by a factor of $1.5 - 2\times$. We report the precise run-times of models with and without retrieval augmentation on Places365-LT in Table 5. Given that the index is static, the number of iterations per second is constant throughout training. All training is carried out on a single node, containing $8\times$ A100 GPUs (32GB Mem).

Moving a tensor from the GPU to CPU, querying the index, then moving the resultant tensor back to GPU maybe expected to slowdown training. However, we find that the impact is minor with the majority of overhead coming from the additional text encoder (the random encoder, 'Rand.', contains no additional parameters). To facilitate multi-node training, RAC keeps separate, complete copies of each index in memory for each node, ensuring querying the index is never the bottleneck, which we found to be essential. While it is possible to do the search entirely on GPU, due to the low overhead we do not do this, instead using the free GPU memory to facilitate a large batch size. Due to the use of HNSW, index query time is logarithmic with respect to the index size, and a standard exhaustive search is prohibitively slow.

Index Data	Size	Text Enc.	Speed (s/epoch)
None	None	None	23.6
Places	184K	Rand.	28.3
Places	184K	CLIP	44.3
Places, IM21k	11.2M	CLIP	47.0

Table 5. Effect of additional text encoder, and lookup on training wall-time for RAC ($k = 50$) on Places365-LT. Top row indicates the use of the base encoder only. The majority of added overhead comes from use of the text encoder, rather than the lookup itself.

5. Limitations

While RAC demonstrates robust performance for both naturally occurring (iNat) and constructed (Places365-LT) long-tailed class distributions, the analysis could be further expanded to include additional long-tailed datasets. The performance of RAC on balanced datasets is also of interest and not explored. Finally, while RAC clearly demonstrates the benefit of an explicit retrieval component, the data being retrieved (labels) is of limited value and imposes a cap on RACs performance—a natural extension is to query for whole paragraphs or captions. However, the 76 token limit imposed by the CLIP text encoder prevents this, and would need to be increased. We leave this for future work.

6. Conclusion

We have introduced RAC, a generic approach to augmenting standard classification pipelines with an explicit retrieval module. RAC's retrieval module, without prompting, achieves a high level of accuracy on tail classes, freeing up the base encoder to focus on common classes. RAC improves upon the state-of-the-art results on the iNat and Places365-LT benchmarks by a large margin for the task of long-tail image classification. We hope that RAC represents a step towards more effectively making use of external memory within common computer vision architectures, and we predict its use for other vision tasks, particularly, such as one/few shot learning, and continual learning without catastrophic forgetting.

References

[1] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv: Comp. Res. Repository*, 2019. 3

[2] Claire Cardie and Nicholas Howe. Improving minority class prediction using case-specific feature weights. In *Proc. Int. Conf. Mach. Learn.*, 1997. 1

[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *J. Artificial Intelligence Research*, 16:321–357, 2002. 1, 2

[4] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 782–791, 2021. 1

[5] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *Proc. Eur. Conf. Comp. Vis.*, pages 694–710, 2020. 2

[6] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 715–724, 2021. 1, 2, 5

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9268–9277, 2019. 6

[8] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv: Comp. Res. Repository*, 2021. 1

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 248–255. Ieee, 2009. 5

[10] Zongyong Deng, Hao Liu, Yaoxing Wang, Chenyang Wang, Zekuan Yu, and Xuehong Sun. PML: Progressive margin loss for long-tailed age classification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 10503–10512, 2021. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv: Comp. Res. Repository*, 2020. 1, 4, 5

[12] Chris Drumnond. Class imbalance and cost sensitivity: Why undersampling beats oversampling. In *ICML-KDD Workshop: Learning from Imbalanced Datasets*, volume 3, 2003. 1, 2

[13] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. Int. Conf. Artificial Intelligence and Statistics*, pages 249–256, 2010. 4

[14] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1705–1714, 2019. 3

[15] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv: Comp. Res. Repository*, 2014. 3

[16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proc. Int. Conf. Mach. Learn.*, pages 1321–1330, 2017. 3

[17] Hao Guo and Song Wang. Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 15089–15098, 2021. 1

[18] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv: Comp. Res. Repository*, 2020. 3

[19] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proc. Int. Conf. Intelligent Computing*, pages 878–887, 2005. 1, 2

[20] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowledge & Data Engineering*, 21(9):1263–1284, 2009. 1

[21] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6626–6636, June 2021. 1, 2, 5

[22] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5375–5384, 2016. 1, 2

[23] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *arXiv: Comp. Res. Repository*, 2017. 5

[24] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv: Comp. Res. Repository*, 2019. 1, 5

[25] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv: Comp. Res. Repository*, 2020. 3

[26] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv: Comp. Res. Repository*, 2021. 1

[27] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *Proc. Int. Conf. Learn. Representations*, 2020. 3

[28] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Proc. Eur. Conf. Comp. Vis.*, pages 491–507. Springer, 2020. 5

[29] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *arXiv: Comp. Res. Repository*, 2019. 3

[30] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv: Comp. Res. Repository*, 2020. 3

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2980–2988, 2017. 5

[32] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2970–2979, 2020. 2

[33] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2537–2546, 2019. 3, 5, 6

[34] 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 *Proc. Eur. Conf. Comp. Vis.*, pages 181–196, 2018. 1

[35] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836, 2018. 5

[36] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv: Comp. Res. Repository*, 2020. 1, 2, 3, 5, 6

[37] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv: Comp. Res. Repository*, 2019. 3

[38] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv: Comp. Res. Repository*, 2018. 2

[39] Yoon-Joo Park and Alexander Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proc. ACM Conf. Recommender Systems*, pages 11–18, 2008. 1

[40] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proc. Conf. Empirical Methods in Natural Language Process.*, pages 1532–1543, 2014. 7

[41] Joaquin Quiñonero-Candela, Masashi Sugiyama, Neil D. Lawrence, and Anton Schwaighofer. *Dataset shift in machine learning*. Mit Press, 2009. 1

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv: Comp. Res. Repository*, 2021. 1, 4

[43] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? Data, augmentation, and regularization in vision transformers. *arXiv: Comp. Res. Repository*, 2021. 4

[44] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 843–852, 2017. 1

[45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2818–2826, 2016. 3

[46] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. Grafit: Learning fine-grained image representations with coarse labels. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 874–884, 2021. 3, 6

[47] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 8769–8778, 2018. 6

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 5998–6008, 2017. 1

[49] Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. Adaptable and interpretable neural MemoryOver symbolic knowledge. In *Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3678–3691, June 2021. 3

[50] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv: Comp. Res. Repository*, 2020. 3, 5

[51] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv: Comp. Res. Repository*, 2021. 6

[52] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *Proc. Eur. Conf. Comp. Vis.*, pages 247–263. Springer, 2020. 1

[53] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv: Comp. Res. Repository*, 2021. 1

[54] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2361–2370, 2021. 2, 5

[55] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5409–5418, 2017. 5

[56] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv: Comp. Res. Repository*, 2021. 2, 5, 6

[57] Yan Zhao, Weicong Chen, Xu Tan, Kai Huang, Jin Xu, Changhu Wang, and Jihong Zhu. Adaptive logit adjustment loss for long-tailed visual recognition. *arXiv: Comp. Res. Repository*, 2021. 1, 5

[58] Yan Zhao, Weicong Chen, Xu Tan, Kai Huang, Jin Xu, Changhu Wang, and Jihong Zhu. Improving long-tailed clas-

sification from instance level. *arXiv: Comp. Res. Repository*, 2021. 6

[59] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9719–9728, 2020. 1

[60] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2017. 6