Retrieval Augmented Classification for Long-Tail Visual Recognition*

Alexander Long¹,², Wei Yin², Thalaiyasingam Ajanthan¹, Vu Nguyen¹, Pulak Purkait¹, Ravi Garg¹, Alan Blair³, Chunhua Shen⁴, Anton van den Hengel¹,²

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 [50] models have arrived in Computer Vision, with parameter counts and pretraining dataset size increasing rapidly [12,27,36,44,46,54]. 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 [9] 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 [2,21,41] instance of the more general label shift problem [43], 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 [25].

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 [3,13,20,23] or cluster-based sampling [7], or explicitly via logit [11,38,58] or loss [22,38] 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 [18,53,60] 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-
2. Related Work

Resampling and Logit Adjustment Over-sampling sparse classes [3, 20] is one of the oldest approaches to addressing distribution bias, but one that is still in common use. Under-sampling common classes [13], applying additional data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed

Resampling and Logit Adjustment Over-sampling sparse classes [3, 20] is one of the oldest approaches to addressing distribution bias, but one that is still in common use. Under-sampling common classes [13], applying additional data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed data-augmentation to sparse classes in pixel, or feature space [5, 33], or sampling uniformly from pre-computed
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. [52] 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 [16]. The purpose of that model was symbolic manipulation, however, which renders its architecture quite different to that of RAC. Gong et al. [15] 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 [19, 31]. While such approaches make use of differential retrievers, which introduces the problem of external data [19, 31]. 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) [28] 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 [26, 30]. 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 [51], 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 [48]. The Open-world Long-tail model proposed in [34] 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 \( S = \{(x_n, y_n)\}_{n=1}^N \), where \( x_n \in X \subset \mathbb{R}^D \) and labels \( Y = \{1, 2, ..., L\} \). Training class frequencies are defined as \( N_y = \sum_{(x_n, y_n) \in S} I_{y_n=y} \) and the test-class distribution is assumed to be sampled from a uniform distribution over \( Y \), but is not explicitly provided during training. The goal is thus to minimize the balanced error, of a scorer \( f : X \rightarrow \mathbb{R}^L \), defined as:

\[
BE(x, f(\cdot)) = \sum_{y \in Y} P_{x|y} \left( y \notin \arg \max_{y' \in y} f_{y'}(x) \right)
\]

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

\[
\ell_{\text{BalCE}}(x, y, f_y(\cdot)) = -\frac{1}{N_y} \log \frac{e^{f_y(x)}}{\sum_{y' \in y} e^{f_{y'}(x)}}.
\]

This is a form of re-weighting, where the contribution of each label’s individual loss is scaled by an approximation of \( P(y) \), which for \( \ell_{\text{BalCE}} \), 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 \( \alpha_y \)) and adjusted (via \( \Delta y \)) softmax cross entropy loss:

\[
\ell(x, y, f_y(\cdot)) = -\alpha_y \log \frac{e^{f_y(x)+\tau \Delta y}}{\sum_{y' \in y} e^{f_{y'}(x)+\tau \Delta y}}
\]

where \( \tau \) 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) [38], can be recovered with \( \Delta y = \log(N_y/N) \) and LDAM [1] with \( \alpha_y = 1/N_y \) and

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

Both are Fisher-consistent with respect to the balanced loss. The optimal \( \tau \) can be found with a holdout set, or set to 1 if the logits are calibrated [17]. In our experiments, this calibration is achieved through label smoothing [39], which has been shown to implicitly calibrate neural networks [47]. 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 re-scale 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 (\( N_{x,t}^{\text{test}} = 3 \), \( \forall i \in Y \))
The base branch encoder \( B(\cdot) \) can be any choice of a standard backbone network. In our experiments, we primarily use the ViT-B-16 variant of Visual Image Transformer [12], 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, \( x^q \). This text is then fed through a text encoder and transformed by another linear layer into logits \( f^\text{ret}(x^q) \).

We make use of the pretrained BERT-like text encoder (63M parameters, 12-layer 512-wide model with 8 attention heads) from CLIP [44], 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. 4). 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 [14] also being dependent on \( L \)).

\[
f(x) = \frac{L}{2} \left( \frac{f^\text{ret}(x)}{\| f^\text{ret}(x) \|_2} + \frac{f^\text{base}(x)}{\| f^\text{base}(x) \|_2} \right),
\]

where \( f^\text{base}(x) \) represents the logits produced by the image encoder backbone, and \( f^\text{ret}(x) \) 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 \( \mathcal{E}(\cdot) \), a pre-existing set of external images \( \mathcal{I} = \{i_j\}_{j=1}^J \), with associated text \( \mathcal{J} = \{t_j\}_{j=1}^J \), which may be labels, descriptions, captions etc. Unless otherwise specified \( \mathcal{E}(\cdot) \) is a ViT-B-16, pretrained on ImageNet following [45]. Prior to training RAC, the retrieval module is initialized by producing image keys \( \mathcal{Z} = \{z_j\} \) such that \( z_j = \mathcal{E}(i_j) \forall j \), and storing the resultant representations in a fast approximate \( k \)-NN index.

During training, we produce features \( z^q = \mathcal{E}(x^q) \) for each image \( x^q \) in the training batch. The \( k \)-NN is queried for each \( z^q \) 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 \( \mathcal{J} \) is recovered for every such index, generating \( k \) text elements for each query. These text elements are then encoded by a text encoder \( \mathcal{T}(\cdot) \) which produces the retrieval branch’s (fixed length) logits, \( f^\text{ret}(x^q) \).

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 \( \mathcal{J} \), and if new examples are similar (from the point of view of the encoder), they can directly impact clas-
Table 1. Historical performance on iNat. †Results reproduced from [58].

| Method     | Many | Med | Few | All |
|------------|------|-----|-----|-----|
| OLTR [34]  | 59   | 64.1| 64.9| 63.9|
| Decouple-LWS [25] † | 70.2 | 71.3 | 71.7 | 71.4 |
| LADE [22] † | 70.9 | 72.4 | 73.1 | 72.6 |
| Grafit [48] | 74.4 | 72.5 | 73.1 | 72.9 |
| ALA [58]  | 71.3 | 70.8 | 70.4 | 70.7 |
| RIDE [52] (2 experts) | 40.6 | 39.1 | 28.6 | 37.6 |
| LADE [38] † | 42.8 | 39  | 31.2 | 38.8 |
| RIDE [52] (4 experts) | 40.4 | 42.4 | 30.1 | 39.3 |
| Grafit [48] | 43.9 | 40.1 | 32.9 | 40.1 |
| ALA [58]  | 43.1 | 42.4 | 33.2 | 40.9 |
| DisAlign [55] | 36.1 | 47.9 | 35.3 | 41.2 |
| RAC (ours) | 75.92 | 80.47 | 81.07 | 80.24 |

Table 2. Historical performance on Places365-LT. †Results reproduced from [58].

| Method     | Many | Med | Few | All |
|------------|------|-----|-----|-----|
| CE RN50    | -    | -   | -   | 32.14 |
| BalCE RN50 | -    | -   | -   | 38.31 |
| CE ViT-B-16 | 50.81 | 33.83 | 19.51 | 37.16 |
| BalCE ViT-B-16 | 49.03 | 45.72 | 29.05 | 43.67 |
| Retrieval | -    | 43.50 | 41.99 | 26.83 |
| Base ViT-B-16 | 44.57 | 45.06 | 29.05 | 43.67 |
| RAC ViT-B-16 | 48.69 | 48.31 | 41.76 | 47.17 |

Table 3. Comparison of top-1 accuracy against baselines under a common training scheme. Column B indicates the architecture of the base branch.

| Method | B | Many | Med | Few | All |
|--------|---|------|-----|-----|-----|
| Focal Loss [32] † | 41.1 | 34.8 | 22.4 | 34.6 |
| Range Loss [36] † | 41.1 | 35.4 | 23.2 | 35.1 |
| OLTR [34] † | 44.7 | 37  | 25.3 | 35.9 |
| Decouple-LWS [25] † | 40.6 | 39.1 | 28.6 | 37.6 |
| LADE [22] † | 42.8 | 39  | 31.2 | 38.8 |
| DisAlign [55] | 40.4 | 42.4 | 30.1 | 39.3 |
| ALA [58]  | 43.9 | 40.1 | 32.9 | 40.1 |
| TADE [57] | 43.1 | 42.4 | 33.2 | 40.9 |
| PaCo [7]  | 36.1 | 47.9 | 35.3 | 41.2 |
| RAC (ours) | 48.69 | 48.31 | 41.76 | 47.17 |

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 [38] as baselines. We consider ‘Base’ trained under the LACE loss [38] 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 (≤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.

In all experiments, unless otherwise indicated, E is a ViT-B-16 encoder, with the weights from [12]. The weights are obtained from pretraining on ImageNet21k (IM21k), a

²We 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.
larger (11M samples) variant of the original 1.2M images ImageNet [10] dataset, with more granular classes. We make use of IM21k to expand the index in some experiments, and use the variant introduced in [29].

4.1. Places365-LT

Places365-LT is a synthetic long-tail variant of Places-2 [61] introduced in [34]. 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 [57], which is the previous state-of-the-art.

4.2. iNaturalist-2018

iNaturalist-2018 (iNat) [49] 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 [48] 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

We baseline RAC’s performance against class-balanced softmax cross entropy (BalCE) and with the retrieval branch removed (Base only variant) under a common training setup in Table 3. We also include final accuracies for ResNet models for comparison. RAC consistently increases all-class top-1 accuracy by 8.04% on Places365-LT and 7.72% on iNat over BalCE, and by 25.48% over standard cross entropy on Places365-LT. These improvements are most pronounced on the tail classes where RAC improves over BalCE by 30.42% on Places365-LT and 10.98% on iNat.

One question is how RAC is able to so outperform methods that are provably consistent, such as LACE [38]. 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 [59]. 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 [8], 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 [57], however, this correlation is weak.

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 query-

| 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 × 384 resolution variants.
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 [42] 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 $T$ 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 3, increasing $k$ consistently increases accuracy, indicating the text encoder $T(\cdot)$ 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.

### 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 $E$. 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 $T$ not increasing— as once most labels are present $E$ is more likely to find a
similar image, however from the perspective of $T$, 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 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 $>10^7$ 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 slow down 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.

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 prompt-
ing, 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 Aperidis, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv: Comp. Res. Repository*, 2019. 3, 12

[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 oversampling 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 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] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proc. IEEE Comp. Vis. Patt. Recogn. Workshops*, pages 702–703, 2020. 13

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

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

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

[10] 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 Comp. Vis. Patt. Recogn.*, pages 248–255. Ieee, 2009. 6

[11] 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 Comp. Vis. Patt. Recogn.*, pages 10503–10512, 2021. 1

[12] 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

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

[14] 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

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

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

[17] 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

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

[19] 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

[20] 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

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

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

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

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

[25] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale image database. In *arXiv: Comp. Res. Repository*, 2019. 1, 5

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

[28] 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.

[29] 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. Vix., pages 491–507. Springer, 2020.

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

[31] 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.

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

[33] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Tsung-Yi Lin. Revisiting unreasonable effectiveness of data in deep learning era. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pages 843–852, 2017.

[34] 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.

[35] 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.

[36] 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.

[37] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Tsung-Yi Lin. Revisiting unreasonable effectiveness of data in deep learning era. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pages 843–852, 2017.

[38] 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.

[39] 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.

[40] 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.
[56] 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.

[57] 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.

[58] 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.

[59] Yan Zhao, Weicong Chen, Xu Tan, Kai Huang, Jin Xu, Changhu Wang, and Jihong Zhu. Improving long-tailed classification from instance level. arXiv: Comp. Res. Repository, 2021.

[60] 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.

[61] 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.
Appendix

We provide more information here.

A. Additional Experiments

A.1. Re-weighting and Temperature Scaling

As a top level loss for RAC, we make use of unscaled logit adjustment exclusively, with no reweighting (i.e. $\alpha_y = 1 \ \forall y \in \mathcal{Y}$) and no temperature scaling ($\tau = 1$) in Eq. (3). This loss is theoretically well-grounded [38], and is appealing due to its simplicity. Nevertheless, other works have noted that the combination of logit adjustment with re-weighting often leads to higher empirical performance [1]. While optimizing the top-level loss is not the focus of RAC, we include here a comparison of RAC performance under various re-weighting schemes when combined with logit adjustment for completeness. Specifically, we consider inverse log

$$\alpha_y = \frac{1}{\log N_y}$$  \hspace{1cm} (6)

and inverse square-root

$$\alpha_y = \frac{1}{\sqrt{N_y}}$$  \hspace{1cm} (7)

class-frequency based re-weighting of individual sample losses.

We confirm that the same effect is present for RAC on Places365-LT, with overall accuracy increasing with the use of both re-weighting schemes, with improvement most pronounced for tail classes (Table S1). However, this trend does not hold on iNat.

For Places365-LT, we also perform a sweep across $\tau$, to evaluate if the same effect can be achieved with manual temperature scaling (Fig. S1). Higher $\tau$ does result in slightly higher overall accuracy, however this effect is minor in comparison to re-weighting. This divergence from theory is likely due to the non-separability of many classes in Places365-LT due to the high label noise.

A.2. Index Ablations

We examine the effect of the distance metric and index type on RAC’s lookup performance and speed in Table S2.

To quantify error induced by an approximate index, we include the lookup accuracy on the index content itself (training set) in addition to the validation accuracy. Query Time (QT) includes encoding samples with a ViT-B-16 model, which is the primarily overhead on small indexes. We observe that the choice of distance metric ($\ell_2$ vs. cosine) has little effect, as may be expected, given the high dimensionality of the index. Cosine distance does introduce a minor computational overhead due to the need to normalize the embeddings prior to querying. The drop in accuracy due to use of an approximate (HNSW) instead of exact $k$-NN is also minor, but comes with a significant ($2\times$) speedup on large index’s. Construction time is constant across all indexes at 0.02μs per sample, with the exception of large-index HNSW, which requires 0.05μ per sample. In summary, performance differences are minor when querying small indexes, however as the index size grows, the choice of HNSW becomes critical to ensure lookup time does not bottleneck training.

A.3. Per-class Accuracy on iNat

We include the same per-class visualization presented in the main body for Places365-LT (Fig. 2), for iNat. Note that for iNat only 3 samples are present for each class in the validation set, hence the square-wave appearance of the plots (Fig. S2). Nevertheless, the same trend is clearly visible in the sliding window moving average, with the retrieval

|                  | Many | Med | Few | All  |
|------------------|------|-----|-----|------|
| None             | 49.72| 49.34| 40.46| 47.75|
| Inverse sqrt.    | 47.12| 49.58| 47.65| 48.32|
| Inverse log      | 44.88| 51.35| 47.54| 48.29|

|                  | Many | Med | Few | All   |
|------------------|------|-----|-----|-------|
| None             | 75.92| 80.48| 81.07| 80.24|
| Inverse sqrt.    | 71.13| 80.02| 81.19| 79.56|
| Inverse log      | 67.17| 80.04| 81.67| 79.35|

Table S1. Effect of common re-weighting schemes when combined with logit adjustment for RAC’s top-level loss.

Figure S1. Effect of $\tau$ within the LACE loss on balanced performance on the Places365-LT dataset.
Table S2. Index ablations on Places365-LT. QT indicates Query Time. Large-sample indices are filled with the ImageNet21k dataset.

Table S3. Dataset Details

Table S4. ViT model architecture details.

Table S5. ResNet model architecture details.

B. Further Details

For Places365-LT we use the training and validation splits during development and report final numbers on the test set, with no validation samples used during final training. For iNaturalist2018, following prior work, we report results on the released validation split, as labels for the test set are not publicly available.

In the main text, we use several common model variants. While the architectures for these models are standard, we specify the high-level design choices in Tables S4 and S5. These choices are consistent across both the 224 × 224 and 384 × 384 variants.

All training is carried out on 8× 32GB A100 GPU’s. We largely follow the procedures outlined in [45], with the following alterations. We finetune with AdamW [35] instead of SGD, and make use of low-magnitude RandAugment [6] alone for data augmentation, with no Color-Jitter, Mixup,
C. Retrieval Branch Visualization

While Table 4 quantifies top-1 performance of the retrieval branch, non-exact match snippets will also effect RAC as they are still likely to be informative. If ‘plane’, ‘runway’, ‘concrete’, ‘sky’, ‘propeller’ etc. are returned for example, it is not difficult for B to place a high score on ‘airport’. We visualize the returned strings for random samples by distance and frequency in Figs. S3 and S4. For all runs, k = 30 as in the main work.

Each column displays (from left to right):

1. query image $x_q$,
2. retrieved labels sorted by distance (distance shown in brackets, exact matches colored green),
3. retrieved labels and occurrence counts, sorted from most to least frequent,
4. histogram of distances to all returned samples,
5. the correct label.

We restrict the lists of retrieved labels to the top eight for visualization purposes. Note that as the distance metric is cosine, a higher value corresponds to more similar samples.

Cutmix, RandomErase or Augmix applied. Full hyperparameters are shown in Table S7. While we found the use of Mixup and Cutmix does boost performance on standard ViTs trained under a BalCE loss, their use of combined targets requires special treatment to make compatible with the LACE loss, which requires hard targets in order to assign the class adjustment. While one approach may be to apply the ‘merged’ class adjustment, the performance benefit is marginal and hence we simply did not include either approach in RAC’s data augmentation pipeline.
Figure S3. Retrieval branch visualization for randomly drawn samples from Places365-LT. For detailed explanation see Appendix C.
Figure S4. Retrieval branch visualization for randomly drawn samples from iNaturalist2018. For detailed explanation see Appendix C.