This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

Fine-grained Species Recognition with Privileged Pooling: Better Sample Efficiency Through Supervised Attention

Andrés C. Rodríguez[1], Stefano D'Aronco[1], Konrad Schindler[1], and Jan Dirk Wegner[1,2]

[1]EcoVision Lab - Photogrammetry and Remote Sensing, ETH Zurich, Switzerland
[2]Institute for Computational Science, University of Zurich, Switzerland

Abstract—We propose a scheme for supervised image classification that uses privileged information, in the form of keypoint annotations for the training data, to learn strong models from small and/or biased training sets. Our main motivation is the recognition of animal species for ecological applications such as biodiversity modelling, which is challenging because of long-tailed species distributions due to rare species, and strong dataset biases such as repetitive scene background in camera traps. To counteract these challenges, we propose a visual attention mechanism that is supervised via keypoint annotations that highlight important object parts. This privileged information, implemented as a novel privileged pooling operation, is only required during training and helps the model to focus on regions that are discriminative. In experiments with three different animal species datasets, we show that deep networks with privileged pooling can use small training sets more efficiently and generalize better.

Index Terms—privileged pooling, supervised attention, training set bias, fine-grained species recognition, camera trap images

1 Introduction

Learning under privileged information is a paradigm where, exclusively for the training samples, one has access to supplementary information beyond the target outputs [1], [2], [3], [4]. The idea is to use this side information to guide the training towards a model that achieves lower generalization error. Such an approach can be beneficial in two situations: *(i)* compared to standard supervised learning it is in general possible to achieve better performance with the same (typically small) number of training samples; *(ii)* it is possible to steer the learning so as to overcome potential biases in the training set. Both situations arise in many domains, but are particularly challenging for the fine-grained classification of animal species: due to difficulties of observing and photographing animals, practical training sets will unavoidably suffer from observational biases and also have limited sample sizes for certain classes.

The concept of privileged information during training was originally introduced in [1] to improve the estimation of slack variables and the convergence rate of Support Vector Machines (SVMs). Subsequent works [2], [3], [4] have adapted this idea to a variety of visual tasks, by adding bounding boxes, attributes or sketches as privileged information [5], [6]. Technically, one can interpret learning under privileged information as a regularization of the model parameters with additional knowledge about the training samples.

Many common CNN architectures, like ResNet [7] or Inception [8], employ a global average pooling layer before the final fully connected layer(s), in order to reduce the number of parameters and to make the model applicable to input images of varying size. However, much information is lost during feature averaging, as features of the object of interest (in our case the animal) are merged with background features. Evidently, this can lead to noisy representations, particularly if the training set for some target classes is small – a frequent situation when dealing with skewed data distributions where some classes, for instance certain animal species [9], are much rarer than others. A similar problem arises when the procedure used to acquire the training data induces sampling biases, which then may cause the network to learn spurious correlations that are irrelevant, or even harmful, for the task [10], [11]. Specifically, global pooling operations harm generalization if different categories of interest often appear in the same context, thus complicating the conceptually simple task to focus on a small, relevant region; such as for instance in our application, where animals are surrounded by similar vegetation. We thus advocate the use of privileged information during training to guide the model's attention.

We introduce *privileged pooling (PrPool)* a visual attention mechanism for animal species recognition that leverages privileged information in the form of keypoint locations to learn a weighted pooling operator. It is intuitive that annotations of important body parts facilitate learning from small training sets. We use point-level part annotations, which are relatively cheap to collect and at the same time directly relevant to discriminate animal species that look alike. A few works have investigated self-supervised attention as a means to improve image understanding [12], [13], [14]. We are not aware of any systematic inquiry into supervised learning of the pooling operator from privileged information. Notably, [15] derive a scheme for action recognition

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

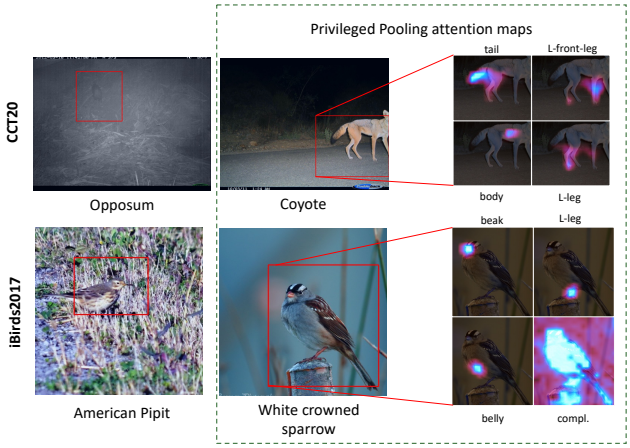


Fig. 1. Predicted classes using privileged pooling on CCT20-Cis test dataset (top) and iBirds2017 test dataset (bottom). Bounding boxes are computed using the predicted attention maps. Attention maps (bounding-box cropped for visualization) depict the encoded privileged information from different keypoints provided at train time. The bottom right-most attention map is not supervised by any keypoint and acts as complementary to other animal regions.

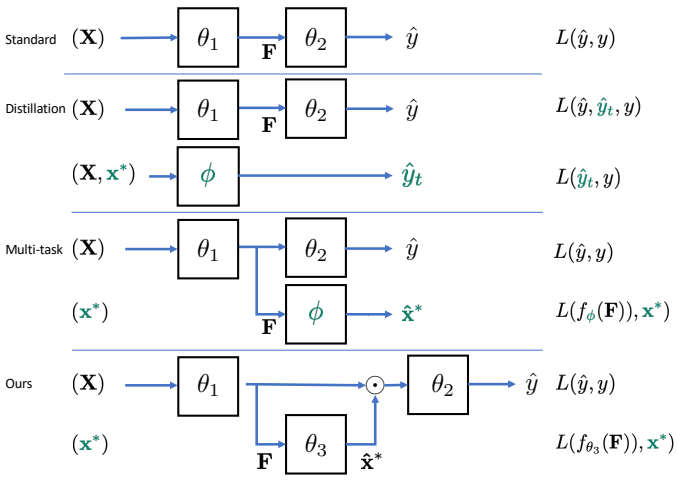


Fig. 2. Comparison of learning strategies. In the standard network with parameters θ, an input $\mathbf{x}$ is mapped to a latent encoding $\mathbf{F}$ and then on to a prediction $\hat{y}$. Distillation first learns a teacher network with parameters ϕ using also privileged information $\mathbf{x}^\star$, then learns the weights θ to approximate that teacher network. Multi-task learning jointly learns to predict also $\mathbf{x}^\star$ with a decoder with parameters ϕ. The proposed framework adds an attention mechanism with parameters θ_3 and supervises it with $\mathbf{x}^\star$. Green denotes quantities used only during training.

that relies on keypoints to learn a very specific pooling operation (a low-rank approximation of bilinear pooling).

Here, we explore the role of attention maps as a general tool to capture spatially explicit privileged information. In our case, the additional annotations come in the form of animal keypoints, which are used to train attention maps as a soft gating mechanism that selects relevant features. Fig. 1 shows an example how such an attention map emphasizes different keypoints on a test sample, like the tail, head, and body. In this way we obtain a generic attention module that can be combined with any commonly used pooling operator to improve classification performance. We also provide two new datasets with part annotations: (1) Caltech CameraTrap-20+ (CCT20+), obtained by augmenting a subset of the Caltech CameraTrap-20 dataset [16]. With this challenging dataset, we demonstrate that our method is able to counteract inherent background biases and thereby improve classification performance. (2) iBirds2018+, a subset of rare bird species from iNaturalist2018 [9]. With this dataset we show that our method improves classification in the long tail of rarely observed species. Furthermore, we also experiment with the CUB200 [17] birds dataset under a scarce data regime, and also outperform state-of-the-art methods based on both privileged information and few-shot learning. To assess generalization, we extract the matching subset of the *aves* (bird) family (termed *iBirds2017*) from the iNaturalist-17 dataset [9] and test our model trained on CUB200. In that experiment the advantage of our model is even bigger. Overall, we show that supervising the pooling with privileged information affords better generalization with fewer training samples, and is also a powerful alternative to few-shot learning when labeled training data is scarce. Code and the corresponding CCT20+ and iBirds2018+ annotations will be made available at github.com/ac-rodriguez/privilegedpooling.

2 Related Work

Learning under Privileged Information attempts to leverage additional information $\mathbf{x}^\star$ during training, but does not rely on it at test time, see Figure 2. How to best exploit such side information is not obvious. Several algorithms have been developed for SVMs, for tasks including action [5] and image [18] recognition. Applications in the context of deep learning include object detection [19] and face verification [20]. Also simulated data has been interpreted as privileged information [3], and (heteroscedastic) dropout has been used as a way of injecting, at training time, privileged information into the network [4].

Knowledge Distillation (KD) [21], [22], originally introduced for model compression, is closely related to the concept of privileged information [23], see Figure 2. KD trains a student network to imitate the output of (usually much bigger) teacher network pre-trained on the same task. To distil knowledge of both high- and low-level features from a pre-trained teacher, different variants of KD match feature maps at varying stages of the networks, usually to obtain more compact models [24], [25], [26].

Multitask Learning could be viewed as a naïve way of incorporating privileged side information, by training an auxiliary task to predict the side information, see Figure 2. The hope is that a shared feature representation will benefit the target task, because it profits from the additional supervision afforded by the auxiliary task. Examples in medical imaging [27] and video description [28] use bounding boxes for such purposes. There is a risk that tasks will instead compete for model capacity, leading to decreased performance. Several works focus on the non-trivial task of correctly balancing them [29], [30], [31].

Few-shot Learning deals with the extension of an already trained classifier to a novel class for which there are only few examples. The hope is that the new class, when

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

embedded in the previously learned feature space, has a simple distribution that can be learned from few samples [32]. One way to achieve this is to enforce compositionality of the feature space [33], [34] by exploiting additional attributes of the training data, which can be seen as a form of privileged information.

Pooling. Virtually all image classification methods use some sort of pooling over a feature map extracted from a feature extractor – nowadays a deep backbone. Beyond simple average- or max-pooling, other methods like bilinear pooling [35], [36], covariance pooling [37], [38] and higher-order estimators [39], [40] have been proposed. These methods are collectively referred to as *second-order methods*, since they estimate second-order statistics of the features distribution. Empirically this can improve discriminative power [38], [39]. Using a form of attention, [15] tackle action recognition using a bilinear pooling based on [35], [36] and use one of the low-rank vectors to encode pose as privileged information into the network. Unlike ours, that approach is not applicable to first order pooling or to other forms of second-order pooling such as covariance pooling.

Despite recent developments in transfer learning and pooling, it is an open question how to leverage sparse, but highly informative privileged information at training time. We address this with a simple yet effective *privileged pooling* scheme.

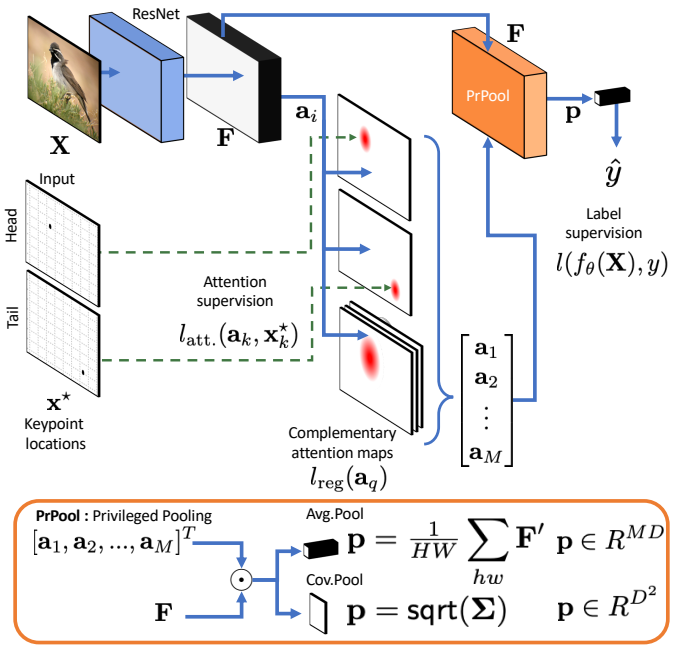


Fig. 3. Privileged Pooling (**PrPool**) illustration. M attention maps with K supervised and Q complementary ones. F' is the extended feature map obtained using all attention maps (See Eq. 8.) $\text{sqrt}(\mathbf{\Sigma})$ is the square-root normalized covariance matrix of the expanded feature map F'. Green-dotted lines denote quantities used only during training.

3 Method

3.1 Background and Problem Statement

Consider a supervised image classification task, with inputs $\mathbf{X} \in \mathcal{X}$ represented as 3D tensors of size $W_\mathbf{X} \times H_\mathbf{X} \times 3$, and outputs y from a label space $\mathcal{Y}$. The goal is to learn a function $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$ with parameters θ, for instance a convolutional network (CNN), that minimises the expected loss $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$:

$$\underset{\boldsymbol{\theta}}{\text{argmin}} \quad \mathbb{E}_{(\mathbf{X},y)\sim P(\mathbf{X},Y)}\big[l(f_{\boldsymbol{\theta}}(\mathbf{X}), y)\big]. \tag{1}$$

In the paradigm of learning under privileged information, we have access to additional side information denoted by $\mathbf{x}^\star \in \mathcal{X}^\star$ for the training examples (but *not* for the test data). For the case of image classification, the supplementary information $\mathbf{x}^\star$ typically has much lower dimension than the input image. Here, we consider annotated keypoint locations. In other words, the training set is composed of triplets of the form $\{\mathbf{X}, \mathbf{x}^\star, y\}$. As we will only have access to $\mathbf{X}$ at prediction time, the overall goal is to minimise the risk of Eq. (1). However, we would want to leverage the information offered by $\mathbf{x}^\star$ to regularise the training procedure. This leads to the new optimisation problem:

$$\underset{\boldsymbol{\theta}}{\text{argmin}} \quad \mathbb{E}_{(\mathbf{X},y)\sim P(\mathbf{X},Y)}\big[l(f_{\boldsymbol{\theta}}(\mathbf{X}), y)\big] + g(\boldsymbol{\theta}, p(\mathbf{X}, \mathbf{x}^\star, y)), \tag{2}$$

where g represents a regulariser that depends on the learned parameters θ and on the joint distribution of the triplets $p(\mathbf{X}, \mathbf{x}^\star, y)$. The challenge is to come up with an appropriate regulariser g that alters the parameters θ in such a way that the generalization error for unseen data $\mathbf{X}$ is reduced at test time. For many CNNs, $f_{\boldsymbol{\theta}}$ can be decomposed into a feature extractor $f_{\boldsymbol{\theta}_1}(\mathbf{X})$ that yields a feature map $\mathbf{F}$ of size $W \times H \times D$, followed by a (first-order) pooling operation $\text{pool}(\mathbf{F})$ that yields a feature vector $\mathbf{p}$ of size D, and finally a multi-layer perceptron $f_{\boldsymbol{\theta}_2}$ that outputs a vector $\mathbf{y}$ of class scores. The spatial dimension of the feature map $\mathbf{F}$ depends on the feature extractor, often it is smaller than the input image $\mathbf{X}$. It is straight-forward to interpolate the input and/or the feature map so that their spatial dimensions match.

As discussed in Section 2, privileged information encompasses several forms of transfer learning, e.g., in the case of multi-task learning the second term of the objective function corresponds to $\mathbb{E}_{p(\mathbf{x}^\star,\mathbf{X})}\left[l(g_{\boldsymbol{\theta}_1}(\mathbf{X}), x^\star)\right]$, where $\boldsymbol{\theta}_1$ are the shared parameters of the common feature extractor. This formulation, however, does not guarantee that the main task makes full use of the privileged information predicted for a test sample. Our architecture does exactly that: the attention maps can be seen as a strategy to pass the keypoint predictions learned from the privileged information back to the classification head, so as to highlight the most important visual features for animal species recognition. In this way we obtain a *visual attention* mechanism [41], [42] that steers the focus of the main network $f_{\boldsymbol{\theta}}$ towards locations in the image that contain important features for the classification task, see Fig. 3.

3.2 Supervision of Attention Maps

The purpose of attention mechanisms is to emphasize image evidence that supports prediction [14], [43]. In images this is commonly done by means of a 1×1 convolution that outputs a weight for re-weighting features before passing them to the next network layer. For example, [12], [13] use attention maps to learn feature gating for fine-grained classification without additional supervision other than the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

image-level class label. Here we explore a supervised attention mechanism: privileged information in the form of keypoint annotations is available at training time and serves to teach the network how to identify locations of interest in the latent feature representation.

As annotations we provide, for every training image, the desired output label as well as a set of K keypoint locations. Keypoints are ordered and every point has a fixed semantic meaning, in our case a specific body part of the animal. We found that scheme particularly effective for our application, as it delivers highly informative privileged information with fairly low annotation effort.

The framework is depicted in Figure 3. We add a network branch that derives K attention maps $\mathbf{a}_k$ from the feature map $\mathbf{F}$. In contrast to previous approaches we rely on 3×3 convolutions to produce the attention maps. This is necessary since we need to have a larger receptive field to produce attention maps that re-weight based on surrounding pixels using higher level concepts from the image (i.e. head, tail, etc.) instead of just the feature vector itself.

Keypoint locations can be represented in different ways. A simple idea would be a list of K image coordinates (x, y) that locate the keypoints in the image. A second possibility that better suits our approach is to create a set of K binary maps with the same spatial dimensions as the input image, with pixels set to 1 at keypoint locations and 0 otherwise. This allows us to train our attention maps $\mathbf{a}_k$ with a binary cross-entropy loss:

$$l_{\text{BCE}}(\mathbf{a}_k, \mathbf{x}_k^\star) = \frac{-1}{WH} \sum_{w,h} \left(x^\star_{whk} \log(a_{whk}) + (1 - x^\star_{whk}) \log(1 - a_{whk}) \right), \quad (3)$$

where $\mathbf{x}_k^\star$ represents the binary map for the k-th keypoint location, and $\mathbf{a}_k$ is the predicted attention map with values in the continuous interval $[0, 1]$. Note that the keypoint locations $\mathbf{a}_k$ can easily be interpolated to fit feature maps $\mathbf{F}$ of different resolutions, depending on the network architecture.

Because the keypoint annotation might sometimes not be exactly at the right position, it is convenient to adopt a multi-scale loss. Attention map and keypoint map are passed through max-pooling operators $\{\mathcal{S}_1..\mathcal{S}_J\}$, with different kernel-sizes to account for different scales. Finally all the losses (Eq. 3) are combined:

$$l_{\text{attention}}(\mathbf{a}_k, \mathbf{x}_k^\star) = \sum_j l_{\text{BCE}}(\mathcal{S}_j(\mathbf{a}_k), \mathcal{S}_j(\mathbf{x}_k^\star)). \quad (4)$$

The multi-scale attention loss $l_{\text{attention}}$ (Eq. 4) is then applied separately to all K keypoint maps.

Note that the keypoints are not always all visible in the input image. If the k-th keypoint is missing, then $\mathbf{x}_k^\star$ is set to 0 everywhere. This choice reflects the preference that, if a keypoint is not visible, the network should learn to predict its absence, corresponding to an empty attention map.

Complementary attention maps are also included. Although they are not supervised by any keypoint annotation, they allow the network to attend to potentially important regions not indicated by keypoints. For these additional attention maps, proper regularisation is necessary, otherwise the optimization may converge to the trivial solution of a uniform map. The center loss [44] has been successfully used to enforce a single feature center per label and penalize distances from deep features to their corresponding center, see [12]. We empirically found that, in our case, a much simpler regularization that maximizes the variance within each attention map yields better results:

$$\bar{a}_q = \frac{\sum_{w,h} \mathbf{a}_{whq}}{WH}, \qquad l_{\text{reg}}(\mathbf{a}_q) = \bar{a}_q \cdot (1 - \bar{a}_q). \quad (5)$$

In Eq. (5), $\bar{a}_q$ represents the average value of the complementary attention map q. If that map is trivial (constant), then $\bar{a}_q$ will tend to either 0 or 1. A simple way to penalize extreme values is to maximize $l_{\text{reg}}(\mathbf{a}_q)$. Intuitively, if we consider $\bar{a}_q$ as the parameter of a Bernoulli distribution, $l_{\text{reg}}(\mathbf{a}_q)$ corresponds to its variance: a larger value translates to a more heterogeneous attention map. The proposed regulariser ultimately imposes a bias against trivial maps that are overly diffuse (or, in the extreme case, uniform). We have also tested various other, more complex, regularisations; but did not observe empirical improvements over the proposed one, see Appendix B for details. The final loss for a model with a total of M attention maps, including K supervised and Q complementary attention maps is defined as:

$$L = l(f_\theta(\mathbf{X}), y) + \frac{1}{K} \sum_k^K l_{\text{attention}}(\mathbf{a}_k, \mathbf{x}_k^\star) - \frac{1}{Q} \sum_q^Q l_{\text{reg}}(\mathbf{a}_q). \quad (6)$$

3.3 Attention Pooling

Pooling operations integrate information over the spatial dimensions of a feature map $\mathbf{F}$ with size $H \times W \times D$, to obtain a vector $\mathbf{p}$, assumed to have the necessary representative power for image-level classification. For average pooling, the d-dimensional vector $\mathbf{p}$ is simply the per-channel mean of $\mathbf{F}$:

$$\mathbf{p}^{\text{AvgPool}} = \frac{1}{HW} \sum_{h,w} \mathbf{F}_{hw}. \quad (7)$$

Attention map serve to increase the representative power of $\mathbf{p}$. To that end, we first expand the dimension of the feature map $\mathbf{F}$ using the attention maps. We denote the expanded feature vector as $\mathbf{F}'$ with size $H \times W \times M \times D$, computed as

$$\mathbf{F}'_{md} = \mathbf{F}_d \odot \mathbf{a}_m \; ; \quad (8)$$

with $\odot$ the Hadamard product over the spatial dimensions h and w. The M attention maps determine from which image regions features are be emphasized, respectively ignored.

This simple formulation allows the new feature map $\mathbf{F}'$ to be used together with a range of pooling operations, including traditional first order pooling, low-rank bilinear pooling and covariance pooling approximations.

First Order Pooling comprises average and max pooling, the most common pooling operations. [45] demonstrate that combining average and max pooling operations yields better results on the CUB200 dataset for fine-grained classification.

Given the expanded feature map $\mathbf{F}'$, we can take samples coming from the same attention map to perform M average pooling operations:

$$\mathbf{p}^{\text{AvgPrPool}} = \frac{1}{HW} \sum_{hw} \mathbf{F}'. \quad (9)$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

As the feature vectors $\mathbf{p}_m^{\text{AvgPrPool}}$ are collected from different regions according to $\mathbf{a}_m$, they preserve some degree of locality in the features . Note that total number of elements in the pooled representation is DM.

Second Order Pooling regards each feature vector $\mathbf{F'}_{hwm}$ in the expanded feature map $\mathbf{F'}$ as a sample and computes a covariance matrix among the features. The mean and covariance matrix are computed by averaging over the spatial dimensions, h and w, as well as over the attention map dimension m. If the feature dimension D is too large, a 1×1 convolution on $\mathbf{F'}$ can be used to reduce its size to $\tilde{D} \ll D$, to save computational resources as previously proposed by [37]. After reshaping the feature map $\mathbf{F'}$ to $\tilde{D} \times S$, where $S = HWM$, the covariance matrix can be computed as:

$$\mathbf{\Sigma} = \frac{1}{S}(\mathbf{F'} - \bar{\mathbf{F}}')(\mathbf{F'} - \bar{\mathbf{F}}')^T \ , \tag{10}$$

with $\bar{\mathbf{F}}'$ the average feature value over the S samples. Furthermore, [38] showed that normalising $\mathbf{\Sigma}$ by taking its square root, denoted here as $\mathsf{sqrt}(\mathbf{\Sigma})$, drastically improves the representation power of the features. An effective method consists in using the Newton-Schulz iterative matrix square root computation, which can be efficiently implemented on GPU [37].

Finally, note that one can backpropagate through the whole operation by using Newton-Schulz iterative square root computation, which enables end-to-end training of the network, with fine-tuning of the attention maps. We use the square root as our feature vector $\mathbf{p}$ of size $\tilde{D}^2$:

$$\mathbf{p}^{\text{CovPrPool}} = \mathsf{sqrt}(\mathbf{\Sigma}). \tag{11}$$

Other types of Second Order Pooling focus on a low rank approximation to perform bilinear pooling, although it can easily be computed in combination with the feature map $\mathbf{F'}$ we observed that the best representative power came from the covariance pooling described above. Finally, note that covariance pooling is also possible over the original (not expanded) feature map $\mathbf{F}$, leading to standard covariance pooling without attention.

4 Experiments

We evaluate the proposed method on three different datasets, showing how it improves over Average Pooling and Covariance Pooling, and compare its performance against other methods that also leverage privileged information.

4.1 Datasets

CUB200 [17] is a dataset of 200 different bird species, with a total of 5,994 training images and 5,794 test images. Each image comes with 15 keypoint annotations for body parts such as beak, belly, wings, etc. CUB200 has been extensively used for fine-grained image classification and highly specialised architectures have been designed for it. Still, a vanilla ResNet-50 pretrained on ImageNet achieves 86% accuracy (note that ImageNet includes some bird classes, and there might even be some common images between the two datasets).

Our focus is to evaluate keypoint annotations as privileged information to *(i)* train a model in data-scarce settings, and *(ii)* improve the generalization to other data. Images in CUB200 are usually centered on the bird and depicted in "standard" poses suitable for recognition, as in a field guide. To test generalization, we use images from the iNaturalist-2017 [9] dataset, which features less curated, more challenging images of birds. 156 birds species are shared between CUB200 and iNaturalist-2017, for those species there are in total 3,407 images (average 22 samples/species) in iNaturalist-2017, which we use as an additional test set, termed **iBirds2017**.

iBirds2018 is a subset of iNaturalist2018 [9], comprising 1258 bird classes with a total of 143,950 training and 3,744 validation samples. As this **iBirds2018** dataset exhibits a particularly long-tailed (realistic) class distribution, we separate it into different subsets according to the number N of samples available per class. First, as commonly done in the literature we consider 4 large sub-sets: 308 many-shot ($N > 100$), 563 mid-shot ($20 \leq N \leq 100$), 387 low-shot20 ($N < 20$) and 188 low-shot15 ($N < 15$). Second, in a similar way, we also consider 9 sub-sets for a more fine-grained analysis. Furthermore we introduce **iBirds2018+**, where we annotated 5 samples in each low-shot15 class with keypoints. For consistency we used the same keypoints as in CUB200. One day of student work was required to write and use a small Python script for keypoint annotation to annotate 1,014 samples.

Caltech CameraTrap-20+ (CCT20+) is a reduced version of CCT20 [16] augmented with privileged information. CCT20 is a set of 57,000 images captured at 20 different camera locations and showing 15 different animal classes.[1] Images from 10 camera locations and taken on even days form the training set (13,000 samples). There are two different test scenarios. The "Cis" split (15,000 samples) consists of the odd days of the same cameras used for training, to test generalization across time for a fixed set of viewpoints. The "Trans" split (23,000 samples) are images from camera locations not seen during training, to test generalization to new viewpoints. As validation data, a single day (3,400 samples) for the Cis scenario, respectively a single location (3,400 samples) for the Trans scenario, is held out from the training data.

CCT20+ augments CCT20 with privileged information that we manually annotated for 1,182 images across all species and all cameras of the training set. We chose keypoints that have the same semantic meaning across the different species of animals in the dataset: head, left-front-leg, right-front-leg, left-back-leg, right-back-leg, tail and body-center.

In our experiments, we do not use the sequence information but treat every image independently. Moreover, we disregard images with more than one animal species and images without any annotated bounding box (the bounding box is not used in our system, we only use it as an indication that an animal was visible for the human annotator). Further details on the dataset annotation and statistics are available in Table 1.

1. CCT20 is a subset of the bigger Caltech CameraTrap set with 243,000 images from 140 locations.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

TABLE 1
Samples per Class in the CCT20 and CCT20+ (marked as Train+) Dataset After Disregarding Cars Class and Sequence Information

	Datasplit					
			Val		Test	
Class	Train	Train+	Cis	Trans	Cis	Trans
All	13,139	1,182	3,408	1,605	15,469	22,626
opossum	2,470	178	346	425	3,988	4,614
rabbit	2,190	189	320	9	1,461	669
empty	2,122	0	1,860	192	3,922	6,355
coyote	1,200	88	161	43	1,096	1,706
cat	1,164	111	169	70	1,455	1,233
squirrel	1,024	109	146	0	496	779
raccoon	845	105	116	108	869	4,314
bobcat	673	110	92	624	751	1,901
dog	580	77	89	80	627	631
bird	353	78	36	5	360	127
rodent	260	44	46	0	140	22
skunk	212	73	24	49	162	263
deer	38	12	2	0	136	0
fox	5	5	0	0	2	1
badger	3	3	1	0	4	11

TABLE 2
Comparison of Model Size

Model	FLOPS [G]	Params [M]	Time [ms]
Avg.Pool	39.9	45.0	15.6
Avg.PrPool	48.3	68.3	19.5
WSDAN	40.0	57.7	17.9
S3N*	90.1	104.3	47.1
Cov.Pool	40.3	49.6	16.6
Cov.PrPool	84.3	52.0	36.8
TransFG	107.56	86.85	116.28

CUB200 default hyperparameters, ResNet-101 as backbone for all baselines except TransFG. *: s3N can only use ResNet-50, see text. The displayed times are average inference times on a Nvidia TitanX GPU

4.2 Implementation Details

All experiments have been implemented in PyTorch [46], with ResNet-101 pre-trained on ImageNet [47] as a backbone. *WSDAN* [12] was implemented in Tensorflow [48] and uses InceptionV3 as a backbone. WSDAN experiments with ResNet-101 did not yield comparable results, which is why we included WSDAN with its original backbone. See the Supplementary material for more details. The results from WSDAN differ slightly ($< 1.6\%$) from the 89.4% accuracy on CUB200 reported in the original publication, note that, in contrast to the original publication, we present an average of different random seeds using original code. All our models are trained with SGD with momentum 0.9 and weight-decay 10^{-4}, batch-size 10 and initial learning-rate of 0.01, a backbone multiplier of 0.01, and exponential decay by a factor 0.9 every 1,000 iterations.

Some methods, such as [12], leverage the predicted attention maps to compute a bounding box around the attended area, crop the image and re-feed it through the network. The final output is obtained by averaging the two predictions. This general strategy is orthogonal to what we propose in this paper and can be applied also with methods that predict attention maps. This practice is sometimes beneficial, but not always, depending on the dataset. We report results for all datasets with re-feeding, except for CCT. See the ablation study in Section 4.8 for an analysis of this practice.

4.3 Baselines

4.3.1 No-$\mathbf{x}^\star$ Methods

As baselines we consider several recent works that pretrain only on ImageNet, and whose network architecture has a complexity similar to ours.

AvgPool: is a vanilla Resnet-101 pre-trained on ImageNet, with average pooling and a single fully connected layer for final prediction.

WSDAN Avg [12] is a method that reaches state-of-the-art results on several fine-grained image classification datasets including CUB200 [17] FGVC-Aircraft [49], Stanford Cars [50] and Stanford Dogs [51]. It uses unsupervised learning of attention maps to perform weighted pooling.

iSQRT [37] (*CovPool*) proposes to change the usual average pooling of feature maps with a square-root normalized covariance pooling. Empirically this is particularly useful for fine-grained classification, where 2^{nd}-order information can be highly informative. The dimension of the feature map is always reduced from 2048 to 256 to lower the computational cost, in all our experiments.

S3N [52] proposes a way to select peaks of the feature response, so as to force the network to explore those peaks, which can be especially informative for the prediction. That method also achieves state-of-the-art performance on CUB200, but it does so at a high cost: d^2 additional parameters are required for the peak sampling layers from a feature map of size $W \times W \times D$. S3N is the only method that does not use ResNet-101 in its original implementation. To stay within GPU memory limits, it must be used with ResNet-50, because of its large number of additional parameters.

LSTM [53] uses a weakly supervised object detector to detect different relevant parts of the image, and in a second step all these detected objects are feeded as a time-sequence of images to a LSTM to finally perform a prediction of an object class.

TransFG [54] is transformer-based network tuned for fine-grained classification, it uses an overlapping window to create patches over the image to select and discriminative image regions.

4.3.2 $\mathbf{x}^\star$ Methods

We test our method *PrPool* in combination with two different options: *AvgPrPool* is a first order alternative where we compute mean and max pooling operations. *CovPrPool* is a second order method using covariance pooling as described in the methods section and a 1x1 convolution to reduce the dimensionality of the feature map from 2048 to 256. For CUB200 we follow WSDAN and use a total of 32 attention maps (in our approach, 15 are supervised by keypoints, the remaining 17 are complementary). For CCT20 we used a total of 8 attention maps (7 supervised and 1 complementary).

The *Multitask* architecture is identical to *AvgPool*, except that the output of the backbone is also connected to a fully connected layer that predicts the keypoints locations. This baseline represents a sort of "lower bound" for the impact of privileged information that is available only during training.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

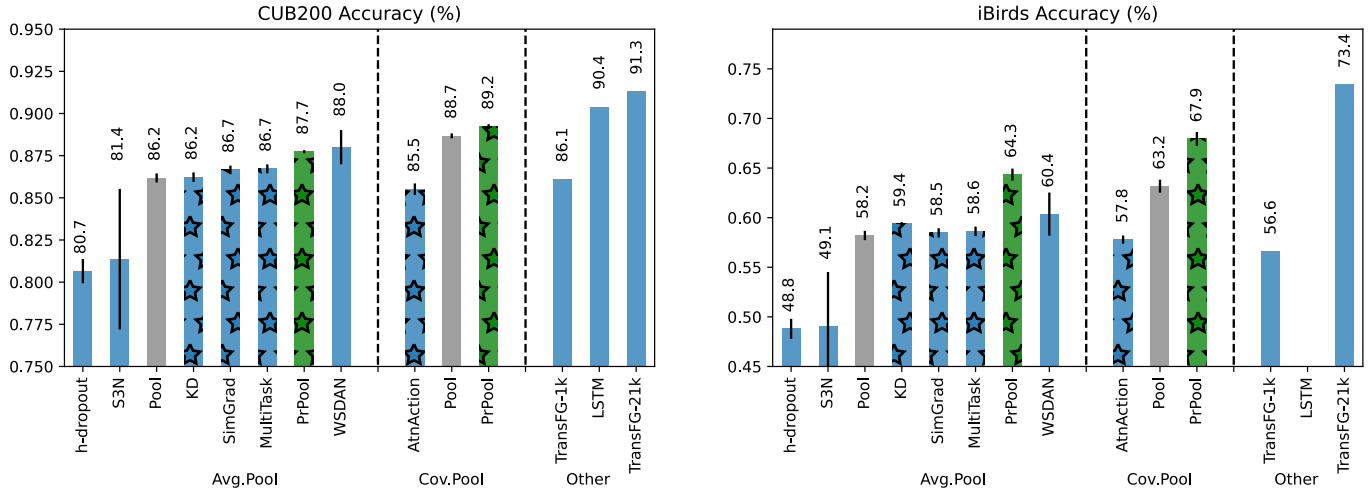


Fig. 4. Top-1 accuracy for CUB200 (left) and iBirds2017 (right) test datasets. Mean accuracy and standard deviation (error bars) over 5 runs. In gray the baselines methods *AvgPool* and *CovPool*; green indicates our methods trained with *PrPool* (ours). Bars with $\star$ indicate use of privileged information at training time ($\mathbf{x}^\star$ methods). LSTM values are taken directly from [53], TransFG-21k uses [54] pretrained on ImageNet-21k and TransFG-1k is pretrained on ImageNet-1k

SimGrad [31] is an improvement of the multitask architecture that aims at reducing the risk that the auxiliary task harms the main task. After separately computing the gradients of the two tasks w.r.t. the shared parameters, the gradients are averaged only if the cosine similarity between them is positive. Otherwise the auxiliary loss is ignored, with the intuition that it should not influence the fitting if it is in conflict with the primary loss.

For knowledge distillation *(KD)* we train a classification network that has two input channels, one for the RGB images and one for the keypoint masks. Once trained, we distill the output of that teacher network into our baseline Resnet101 as student model.

Heteroscedastic dropout (*h-dropout*) [4] highlights how learning under privileged information can be implemented via a dropout regularization. We implemented this method in its original form and in conjunction with other attention-based methods (e.g., [12]), but found that, despite our best effort, the noise injected into the fully connected layer made training unstable for masked inputs with bounding boxes (as in the original implementation), and also with masked inputs around keypoints with different diameters. We show results with the keypoint annotation version, because it empirically performed better.

We have also tested Attentional pooling for action recognition (*AtnAction*) [15] on our datasets. This method uses a low-rank bilinear approximation and takes one of the low-rank vectors as attention maps to encode the privileged information. By default that method sets the rank to $L = 768$, as we did not observed any significant effect when using larger values for L, so we kept the original value in all our experiments.

A main goal of our work is to learn classes for which we only have few training examples, so it is also related to few-shot learning. We test compositional *FewShot* recognition [34], which uses class-level labels to enforce compositionality (see Section 2). We use the same 5 random splits into base classes and novel classes as [34] and run our *PrPool* network on them, uniformly sampling from the batches when creating a batch, in order to deal with the imbalance between novel and base classes.

In Table 2 we report the FLOPs, number of parameters and inference time for different baseline methods as well as for the proposed *PrPool* model. As can be seen the overhead needed by *PrPool* is limited compared to other methods such as *S3N* or *TransFG*. We have used the original implementations of *WSDAN*, *CovPool*, *S3N*, *h-dropout* and *TransFG*; and our own re-implementations of all other methods. All presented results are from experiments we ran ourselves, unless stated otherwise. *LSTM* could not be included in this analysis as there is no open code. It is also not directly comparable conceptually because it uses a recurrent network over a sequence of images in two different steps.

4.4 Fine-grained Classification

Figure 4 presents the results on the CUB200 test set. The numbers are averages over 5 runs with different random initialisations. The average pooling baseline (*AvgPool*) achieves 86.2%. With average pooling supervised by privileged information (*PrPool*), this increases to 87.7%. Among the first order pooling methods in this case the best performance is achieved by *WSDAN* with 88.0%

In line with the literature [37] we find that covariance pooling is superior to average pooling for fine-grained-classification, reaching 88.7%. The best result with pooling methods for the CUB200 dataset is achieved by the proposed privileged pooling, which improves the result to 89.2%. These improvements may seem comparatively small. When, however, testing the trained networks on iBirds2017, the gains are amplified, i.e., our models with supervised attention generalize better. The relative improvements are quite significant, up to 16.7% over the average pooling baseline. While most methods show a mild improvement over the baseline without attention, they do not reach the performance of *PrPool*. This includes other methods that also use privileged information at training time, but do not

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

TABLE 3
Top1 Accuracy for CUB200 and iBirds2017 test set

	Samples per class	iBirds2017 5	iBirds2017 10	iBirds2017 15	CUB200 5	CUB200 10	CUB200 15
Avg	S3N	38.8	49.6	54.0	67.2	79.6	82.4
	Pool	39.3	49.1	54.6	66.0	77.9	81.7
	SimGrad	34.6	47.4	53.2	62.7	77.7	81.8
	MultiTask	33.6	47.2	52.5	61.7	77.5	81.5
	WSDAN	38.5	51.8	56.0	68.8	80.4	84.1
	PrPool (ours)	48.5	58.1	60.3	73.8	82.5	84.6
Cov	AtnAction	38.4	49.1	53.6	68.7	78.1	81.6
	Pool	45.5	56.2	58.8	72.0	82.4	85.0
	PrPool (ours)	**49.6**	**59.6**	**62.8**	**74.6**	**83.3**	**85.4**

Best performance in **bold**. Second best underlined.

show a significant improvement on CUB200 and even fall behind the baseline in iBirds2017. Seemingly they are to some degree overfitted to the CUB200 distribution and not able to generalize.

For completeness, we also include architectures with neither ResNet-101 nor pooling mechanisms that have recently demonstrated high performance on CUB200. *LSTM* [53] is based on a two stage prediction scheme and reports 90.4% top-1 accuracy on CUB200. We could not evaluate that method on iBirds2017 nor its time-complexity because no code is available (and accurate re-implementation is not possible given the description in the paper).

More recently, TransFG [54], a transformer-based network tuned for fine-grained classification, was able to achieve 91.7% accuracy on CUB200. The network is pretrained on ImageNet-21k instead of the usual 1k classes which, by itself, is likely to improve transfer learning [55]. Beyond the sheer number of classes, ImageNet-21k contains >350,000 samples from >350 different bird species, including >60 species also present in CUB200. This overlap suggests that pre-training on ImageNet-21k gives a substantial advantage when processing CUB200 that previously may have been underestimated. We found that TransFG indeed did not perform well when pretrained only on ImageNet-1k as can be observed under category "Other" in Figure 4. A detailed analysis can be found in Appendix C.

4.5 Data Efficiency

Privileged information especially improves the data efficiency in data-scarce regimes. We consider two main scenarios for experimental evaluation: a few-shot learning scenario with CUB200 (as done previously in the literature) and a long-tailed scenario with iBirds2018.

Few-shot learning. We draw n samples per class out of the CUB200 training samples. Table 3 shows that *PrPool* consistently outperforms all competing approaches, with increasing benefits as the training set gets smaller. Note that in the most challenging 5-shot case, there are only 1000 samples to learn 200 classes. We also find that in the small data regime, the naïve multitask loss does not improve performance, and also other baselines become rather inconsistent.

Given the performance achieved with only 5 samples per class, we also compare to few-shot learning. We use

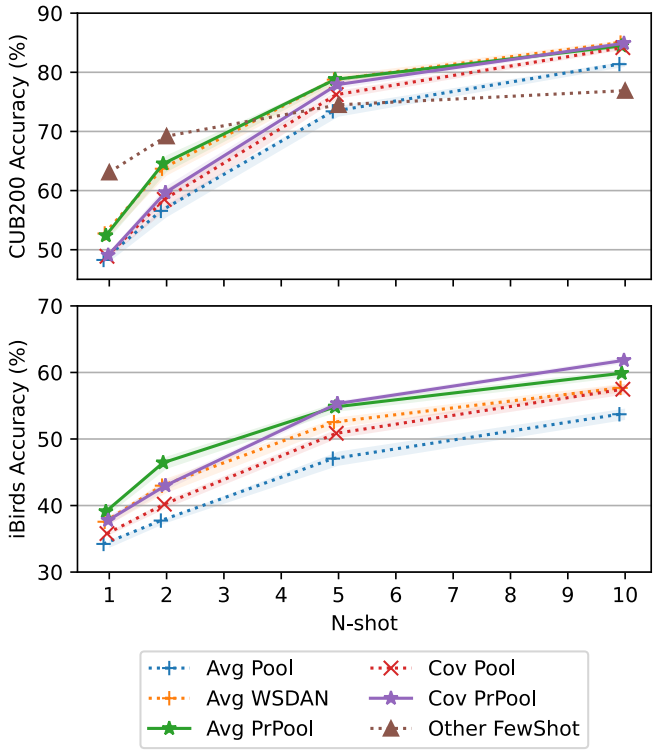


Fig. 5. CUB200 (top) and iBirds2017 (bottom) Top1 Accuracy. 100 base classes, with 100-way new classes with n-shots. Average over five random novel/base splits with standard deviation as shaded areas. Methods with ⋆ marker denote PrPool (ours). FewShot results taken directly from [34]

the same evaluation strategy of [34] with 100 base classes and 100 new classes with only n shots each. Results are displayed in Figure 5. As expected, dedicated few-shot learning – based on a set of well-trained base classes and some form of distance learning to add the new classes – is superior in the extreme 1-shot and 2-shot scenario. But already in the 5-shot case, we find that even simple average pooling is competitive with few-shot learning, and our privileged pooling already outperforms it. At 10 samples the difference is accentuated, as one moves further away from the extreme few-shot setting. Apparently the privileged information can, already at this low sample number, compensate the reduced sampling of the pose and appearance space, by steering the learning towards sub-regions with a well-defined semantic meaning across classes.

As before we evaluate the implemented methods on the iBirds2017 dataset, see Figure 5. Results are consistent with the previous ones, *PrPool* proves to be very effective at increasing performance in low-data regime and improves the generalization power of the network. See Appendix D for more baseline comparisons.

Long-tailed dataset. In general, our method is not specifically designed to deal with long-tailed class distributions. We still evaluate on such a scenario nonetheless using **iBirds2018** to understand its impact on classes with scarce training labels. Table 4 shows the top-1 and top-5 performance for the whole dataset and for different subsets with progressively lower number of training samples. Generally, *CovPrPool* seems to best leverage the privileged information

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

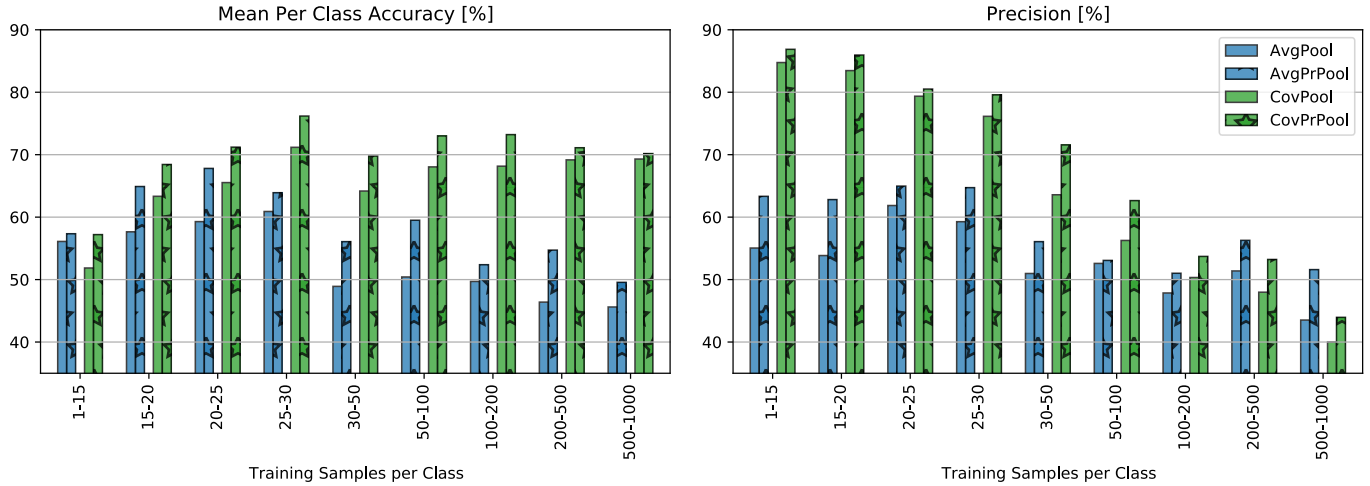


Fig. 6. Mean per class accuracy and precision for iBirds2018. Results are grouped in sub-sets according to the number of available training samples for each class.

TABLE 4
iBirds2018 Results

Pool	Model	Top-1	Top-5	N-shot subsets			
				Many	Mid	Low	Low15
Avg	S3N	47.9	71.5	50.8	49.0	43.9	41.1
	Pool	54.0	77.9	47.4	55.7	56.7	55.1
	MultiTask	58.9	80.7	69.5	58.8	50.5	47.2
	WSDAN	64.8	84.5	**76.2**	65.5	54.6	51.0
	PrPool (ours)	59.5	81.0	52.6	62.6	60.3	**57.4**
Cov	Pool	64.1	85.0	68.8	67.4	55.6	50.5
	PrPool (ours)	**68.8**	**87.8**	71.6	**72.5**	**61.2**	55.7

Best performance in **bold**. Second best underlined

$\mathbf{x}^\star$, reaching a top-1 accuracy of 68.8%, 4 points higher than the best baseline method without $\mathbf{x}^\star$. Importantly, the privileged information available boosts performance for the considered sub-groups. Figure 6 shows Mean per Class Accuracy and Precision for more fine-grained sub-sets according to the number of training samples available. We observe that leveraging privileged information boosts precision consistently for all class sub-sets. It also becomes clear that *CovPool* and *CovPrPool* yield higher class accuracy than *AvgPool* approaches on classes with $>$ 15 training samples; for classes with $>$ 200 training samples *CovPool* and *CovPrPool* also have higher accuracies but at the cost of a reduced precision (Figure 6, left). This effect is somewhat reduced for lower-shot classes though. *CovPrPool* outperformed consistently all other forms of pooling considered in Figure 6. These experiments, on a highly popular dataset, demonstrate performance of our method for a realistic case where the label distribution is not uniform. Our method shows to be effective for cases with few labels at training time. The results support our hypothesis that a moderate labelling effort – a handful of keypoints in a small subset of training images – does lead to a significant performance boost.

4.6 Generalization with biased datasets

The CCT20 dataset is very challenging, due to bad illumination, frequent occlusions, camouflage and extreme perspective that arise in camera traps. Moreover, the highly repetitive scenes are an "invitation to overfit" and learn spurious correlations, which then hinder generalization to new scenarios (e.g., unseen camera locations). Our *PrPool* method achieves the best performance. In this case, with fewer and more distinct classes, first-order pooling works better than *CovPool* (but also the latter outperforms the baselines). Figure 7 shows that *AvgPrPool* outperforms other types of pooling in several classes, such as squirrel, cat and dog. Table 5 shows the results on CCT20. In Cis locations the Mean per Class Accuracy has a slight increase with AvgPool methods when using the CCT dataset (13k training samples) instead of CCT+ (1k training samples): *AvgPool* improves 2.8 points, *AvgPrPool* improves 0.9 points. CovPool methods, on the other hand, have a larger improvement. *CovPool* improves by 9.5 points while *CovPrPool* improves by 12.9 points. These findings are aligned with earlier results we observed on iBirds2018 where *CovPool* methods yield overall better results with more training samples than *AvgPool* methods. We also find that attention-cropping at test time decreases performance for both *WSDAN* and *PrPool* for CCT datasets and therefore omitted that step. Please refer to Section 4.8 for an ablation study on this effect.

Note that our method trained only on the CCT20+ subset with ≈1000 samples outperforms the *AvgPool* baseline even when the latter is trained with 10× more samples. We see two reasons for this, *(i)* the superior data efficiency through Privileged Pooling, and *(ii)* the low diversity of samples in Camera Trap data, where more samples can in fact reinforce inherent dataset biases. In the most challenging Trans-location setting *PrPool* reaches 75% test set accuracy.

We now explore the impact that privileged information has when using only the keypoint annotated images from CCT20+ dataset. As expected, the differences in performance with respect to the baseline methods are larger in this case. See Figure 8 for more details on the per class performance.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

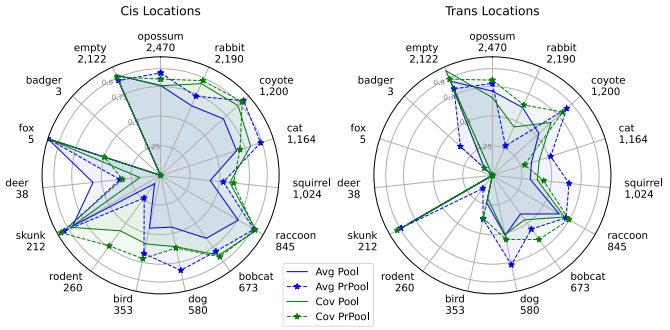


Fig. 7. Accuracy per class in CCT20 test datasets. Training with CCT20: 13K samples including 1,180 samples with keypoint annotations. Classes sorted according to number of training samples per class (indicated below each animal class). Methods with ⋆ marker denote PrPool (ours).

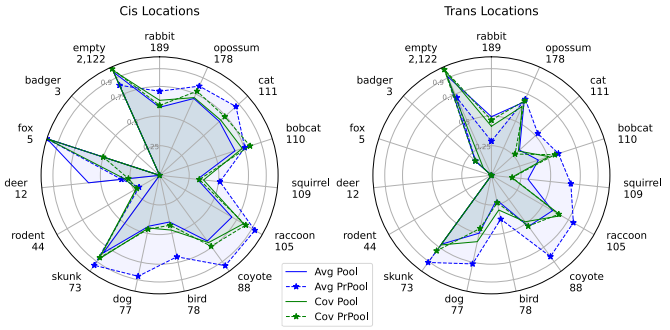


Fig. 8. Accuracy per class in CCT20 test datasets. Training with CCT20+: 1,180 samples, all training samples are provided with keypoint annotations. Classes sorted according to number of training samples per class (indicated below each animal class). Methods with ⋆ marker denote PrPool (ours).

TABLE 5
Overall Accuracy and Mean per Class Accuracy Results of Models Trained on CCT20 and CCT20+

	Dataset	Acc		$\text{Acc}_{\text{class}}$	
		CCT+	CCT	CCT+	CCT
Cis Locations					
Avg	S3N	75.2	75.5	61.7	66.7
	Pool	71.4	73.6	57.0	59.8
	MultiTask	72.2	76.4	59.5	65.8
	WSDAN	72.9	76.2	52.2	63.9
	PrPool (ours)	**81.0**	**83.5**	**69.4**	**70.3**
Cov	Pool	74.3	81.4	59.8	69.3
	AtnAction	72.7	77.4	56.3	68.0
	PrPool (ours)	76.7	82.9	57.0	69.9
Trans Locations					
Avg	S3N	64.2	70.6	40.2	50.9
	Pool	65.1	66.1	42.0	43.6
	MultiTask	65.6	67.2	42.6	50.0
	WSDAN	62.6	61.4	34.7	42.4
	PrPool (ours)	**70.6**	72.3	**52.7**	**54.8**
Cov	Pool	67.2	70.1	41.9	43.8
	AtnAction	64.4	73.0	43.2	51.2
	PrPool (ours)	68.5	**75.0**	42.8	51.6

CCT dataset has 13K samples, including 1K samples with keypoints annotations. CCT+ only has 1K, all of them with keypoints annotations. Best performance in **bold**

TABLE 6
Top-1 Accuracy Results of Attention Maps With Different Supervisions.

	Supervision Type		
	No	Reg	Pr
iBirds2017			
AvgPool	60.3	61.7	65.2
CovPool	62.3	66.5	67.0
CUB200			
AvgPool	84.3	87.0	88.1
CovPool	87.1	88.5	89.0

No Supervision, **Reg**ularized Supervision ($l_{\text{reg}}(\mathbf{a}_q)$) and **Pr**ivileged Supervision ($l_{\text{attention}}(\mathbf{a}_k, \mathbf{x}_k^\star)$)

4.7 Ablation study: attention maps

We go on to analyze how important attention map supervision is in our architecture, in combination with both average and covariance pooling. To that end we train exactly the same architecture as *PrPool*, but without the supervision signal $l_{\text{attention}}$ from keypoint annotations. The results in Table 6 confirm that the privileged information plays an important role and significantly increases prediction performance. Moreover, we observe that already the regulariser alone improves over totally unconstrained self-attention, as expected.

We also vary the number of attention maps used for the CUB200 dataset, see Table 7. As it can be seen from the table, performance tend to slightly increase when complementary attention maps are added to the model, however too many might lead to a minor model overfit.

4.8 Ablation study: Attention Cropping at Test Time

Using the attention maps from *WSDAN* and *PrPool*(ours) it is straight forward to create a bounding box around the areas in the image that the network is attending to. We used this bounding box to crop the image and re-feed it at test time. We observed this is a key element of *WSDAN* and has usually a positive effect, see Table 8. Intuitively this technique should increase the performance as it creates a higher-resolution attention map from the cropped input image (The original image is of size 488^2 and the attention map is 28^2). The results for the CCT20 dataset can be seen in Table 9, in this case the effect of the re-feeding seems to be hurting the overall performance for both models *WSDAN* and *PrPool*. We observed class-specific effects. For instance, when using CCT+ as training dataset the rodent class showed an improvement from 20.0% to 55.0% after attention cropping at test time (Figure 8). This is likely a similar effect as observed for CUB200, where the high-resolution attention map centered around the (rather small) animal helps the identification; on the other hand, some classes were negatively affected by attention cropping, in particular larger animals such as dog, coyote and raccoon. For these, it sometimes happens that keypoints are not identified correctly and the cropping removes relevant information; see Figure 9 for selected samples where this is the case.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

TABLE 7
Top-1 Accuracy when Training With Different Numbers of Attention Maps.

	N. Attention Maps			
	15	16	32	64
iBirds2017				
AvgPool	64.0	64.2	64.3	63.8
CovPool	62.5	66.8	67.9	67.2
CUB200				
AvgPool	87.2	87.1	87.7	87.5
CovPool	85.6	88.6	89.2	88.9

In all cases 15 attention maps are supervised by keypoints, the rest are left for complimentary attention maps: we used 0, 1, 17 and 47 complimentary attention maps.

TABLE 8
Effect of Re-feeding Attention-cropped Images at Test Time on CUB200 and iBirds2017 Test Datasets.

Pool	Model	Test-crop	iBirds2017	CUB200
Avg	PrPool	No	61.4	87.7
		Yes	65.2	88.1
	WSDAN	No	54.9	87.4
		Yes	62.4	88.6
Cov	PrPool	No	64.8	**89.4**
		Yes	**67.0**	89.0

Best performance for each case is underlined. In **Bold** the best overall performance. This re-feeding is only possible with *WSDAN* and *PrPool* models.

TABLE 9
Effect of Re-feeding Attention-cropped Images at Test Time on CCT20 Test datasets

Pool	Model	Test-crop	Acc		Acc_{class}	
			CCT+	CCT	CCT+	CCT
Cis Locations						
Avg	PrPool	No	**81.0**	**83.5**	**69.4**	70.3
		Yes	80.3	82.1	66.8	**73.3**
	WSDAN	No	72.9	76.2	52.2	63.9
		Yes	71.7	75.7	50.8	64.9
Cov	PrPool	No	76.7	82.9	57.0	69.9
		Yes	68.8	75.9	46.7	56.3
Trans Locations						
Avg	PrPool	No	70.6	72.3	52.7	54.8
		Yes	**72.0**	68.5	**55.1**	**60.2**
	WSDAN	No	62.6	61.4	34.7	42.4
		Yes	60.3	62.5	33.5	42.5
Cov	PrPool	No	68.5	**75.0**	42.8	51.6
		Yes	64.8	70.5	41.2	49.5

CCT dataset has 13K samples, including 1K samples with keypoints annotations. CCT+ only has 1K, all of them with keypoints annotations. Best performance for each case is underlined. In **Bold** the best overall performance. This re-feeding is only possible with *WSDAN* and *PrPool* models.

4.9 Qualitative Evaluation of Attention Maps

Attention Maps make it easy to visualize which parts of the input image are being used to make a prediction. For *CovPrPool*, we computed the mean of the attention maps (separately for the supervised and the complementary maps). For the baseline model *AvgPool* with ResNet101 backbone, we used GradCam [43]. In Figure 10, we show random examples from the iBirds2017 dataset. One can see that the baseline focusses on small, specific patterns on the bird, in some cases even on areas not on the bird (see fourth row from Figure 10). This provides an intuitive example how the privileged information can help generalization, by paying attention to relevant, representative parts of the bird.

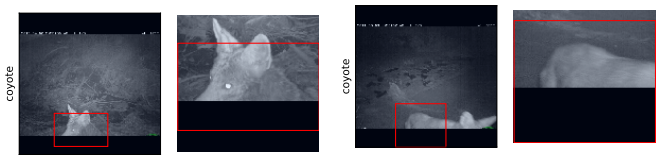


Fig. 9. Selected Samples for Coyote class, where attention cropping hurts performance. The red bounding boxes are derived from the predicted attention maps.

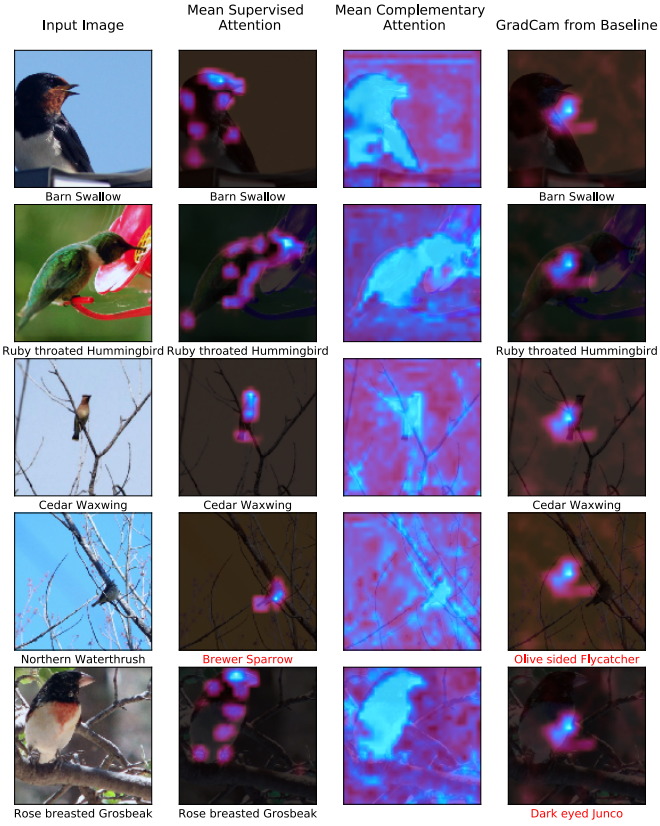


Fig. 10. Comparison GradCam from *AvgPool* and Mean Attention Maps from *CovPrPool*. Random samples from iBirds2017 test dataset. Left most column is input image. Below each sample GT class, and predictions are shown. Misclassified samples in red

Furthermore, we can see that the Complementary attention map learn to largely ignore pixels outside the bird's silhouette, despite not being explicitly trained for this.

We show selected test samples from low-shot of iBirds2018 in Figure 11. The attention maps with *AvgPool* show slight overfitting to background features, which *CovPrPool* seems to effectively ignore.

Example attention maps of trained models. See Figure 12 for random samples of predictions and attention maps over the CCT Cis Test set. For these samples, we observe that the attention maps clearly highlight the different

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

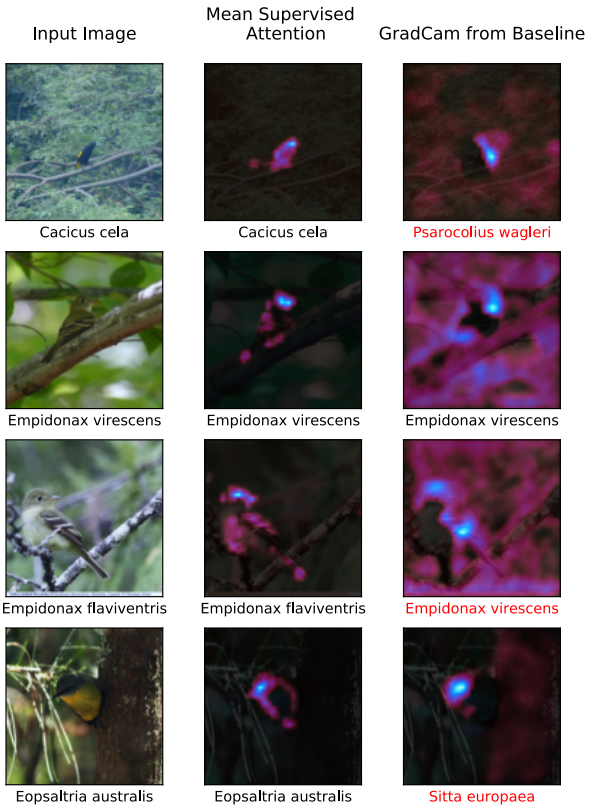


Fig. 11. Comparison GradCam from *AvgPool* and Mean Attention Maps from *CovPrPool*. Selected samples from low-shot classes from iBirds2018 test dataset. Left most column is input image. Below each sample GT class, and predictions are shown. Misclassified samples in red

keypoints, even when the animal is difficult to distinguish from the background. Figure 13, on the other hand, shows samples from the iBirds2017 test dataset. Here the attention map on the bottom right is complementary to the keypoints and effectively performs a foreground-background separation.

5 Conclusions

The aim of learning under privileged information is to exploit collateral information that is available only for the training data, so as to learn predictors that generalize better. We have examined the case where the privileged information comes in the form of keypoint locations, a natural and fairly frequent situation in image analysis. By using keypoints as supervision for attention maps, they can be effectively leveraged to support image classification. Privileged information to steer a model's attention is particularly effective when labeled training data is scarce, and when it exhibits strong biases. Moreover, it turns out that in some small-data scenarios a moderate amount of privileged information may serve as an alternative to few-shot learning.

On a more general note, we see it as an important message of our work that gathering more data is not the only option to fix an under-trained deep learning model. While additional training data is almost always welcome, there are important applications where it is inherently hard to come by. It is encouraging that, with the right design, more elaborate labeling of the existing data can also present a way forward.

References

[1] D. Pechyony and V. Vapnik, "On the theory of learnining with privileged information," in *NeurIPS*, 2010.
[2] H. Yang, J. Tianyi Zhou, J. Cai, and Y. Soon Ong, "Miml-fcn+: Multi-instance multi-label learning via fully convolutional networks with privileged information," in *CVPR*, 2017.
[3] K.-H. Lee, G. Ros, J. Li, and A. Gaidon, "SPIGAN: Privileged adversarial learning from simulation," in *ICLR*, 2019.
[4] J. Lambert, O. Sener, and S. Savarese, "Deep learning under privileged information using heteroscedastic dropout," in *CVPR*, 2018.
[5] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Information bottleneck learning using privileged information for visual recognition," in *CVPR*, 2016.
[6] Y. Chen, X. Jin, J. Feng, and S. Yan, "Training group orthogonal neural networks with privileged information," in *IJCAI*. AAAI Press, 2017, pp. 1532–1538.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.
[9] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The iNaturalist species classification and detection dataset," in *CVPR*, 2018.
[10] A. Torralba and A. A. Efros, "Unbiased look at dataset bias." in *CVPR*, 2011.
[11] S. Ritter, D. G. Barrett, A. Santoro, and M. M. Botvinick, "Cognitive psychology for deep neural networks: A shape bias case study," in *ICML*, 2017.
[12] T. Hu and H. Qi, "See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification," *arXiv preprint arXiv:1901.09891*, 2019.
[13] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *ICCV*, 2017.
[14] K. Li, Z. Wu, K. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *CVPR*, 2018.
[15] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 34–45. [Online]. Available: http://papers.nips.cc/paper/6609-attentional-pooling-for-action-recognition.pdf
[16] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita," in *ECCV*, 2018.
[17] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.
[18] V. Sharmanska, N. Quadrianto, and C. H. Lampert, "Learning to rank using privileged information," in *ICCV*, 2013.
[19] J. Hoffman, S. Gupta, and T. Darrell, "Learning with side information through modality hallucination," in *CVPR*, 2016.
[20] G. Borghi, S. Pini, F. Grazioli, R. Vezzani, and R. Cucchiara, "Face verification from depth using privileged information." in *BMVC*, 2018.
[21] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
[22] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *ACM SIGKDD*, 2006.
[23] D. Lopez-Paz, B. Schölkopf, L. Bottou, and V. Vapnik, "Unifying distillation and privileged information," in *ICLR*, 2016.
[24] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.
[25] Y. Jang, H. Lee, S. J. Hwang, and J. Shin, "Learning what and where to transfer," *arXiv preprint arXiv:1905.05901*, 2019.
[26] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

[27] E. Pesce, S. J. Withey, P.-P. Ypsilantis, R. Bakewell, V. Goh, and G. Montana, "Learning to detect chest radiographs containing pulmonary lesions using visual attention networks," *Medical image analysis*, vol. 53, pp. 26–38, 2019.

[28] L. Zhou, Y. Kalantidis, X. Chen, J. J. Corso, and M. Rohrbach, "Grounded video description," in *CVPR*, 2019.

[29] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *NeurIPS*, 2018.

[30] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018.

[31] Y. Du, W. M. Czarnecki, S. M. Jayakumar, R. Pascanu, and B. Lakshminarayanan, "Adapting auxiliary losses using gradient similarity," *arXiv preprint arXiv:1812.02224*, 2018.

[32] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[33] J. Andreas, "Measuring compositionality in representation learning," in *ICLR*, 2019.

[34] P. Tokmakov, Y.-X. Wang, and M. Hebert, "Learning compositional representations for few-shot recognition," in *ICCV*, 2019.

[35] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard product for low-rank bilinear pooling," *arXiv preprint arXiv:1610.04325*, 2016.

[36] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *CVPR*, 2016.

[37] P. Li, J. Xie, Q. Wang, and Z. Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *CVPR*, 2018.

[38] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *ICCV*, 2017.

[39] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel pooling for convolutional neural networks," in *CVPR*, 2017.

[40] S. Cai, W. Zuo, and L. Zhang, "Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization," in *ICCV*, 2017.

[41] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[42] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[43] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.

[44] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*, 2016.

[45] L. Zhang, S. Huang, W. Liu, and D. Tao, "Learning a mixture of granularity-specific experts for fine-grained categorization," in *ICCV*, 2019.

[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[49] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *arXiv preprint arXiv:1306.5151*, 2013.

[50] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

[51] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, no. 1, 2011.

[52] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, and J. Jiao, "Selective sparse sampling for fine-grained image recognition," in *ICCV*, 2019, pp. 6599–6608.

[53] W. Ge, X. Lin, and Y. Yu, "Weakly supervised complementary parts models for fine-grained image classification from the bottom up," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3034–3043.

[54] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, and A. Yuille, "Transfg: A transformer architecture for fine-grained recognition," *arXiv preprint arXiv:2103.07976*, 2021.

[55] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. [Online]. Available: https://openreview.net/forum?id=Zkj_VcZ6ol

[56] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[57] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your vit? data, augmentation, and regularization in vision transformers," *arXiv preprint arXiv:2106.10270*, 2021.



Andrés C Rodríguez recieved his BS in Industrial Engineering from the Pontificia Universidad Javeriana in Bogota, Colombia in 2012. Then recieved his MSc in Statistics from ETH Zurich in 2017. He joined the EcoVision Lab at ETH Zurich in 2017 as a PhD student. His research interest is at the intersection of machine learning and remote sensing with the focus on large scale analysis for biodiversity, sustainablity and environmental applications.



Stefano D'Aronco received his BS and MS degrees in electronic engineering from the Università degli studi di Udine, in 2010 and 2013 respectively. He then joined the Signal Processing Laboratory (LTS4) in 2014 as a PhD student under the supervision of Prof. Pascal Frossard. He received his PhD in Electrical Engineering from École Polytechnique Fédérale de Lausanne in 2018. He is Postdoctoral researcher in the EcoVision group at ETH Zurich since 2018. His research interests include several machine learning topics, such as Bayesian inference method and deep learning, with particular emphasis on applications related to remote sensing an environmental monitoring.



Konrad Schindler (M'05–SM'12) received the Diplomingenieur (M.Tech.) degree from Vienna University of Technology, Vienna, Austria, in 1999, and the Ph.D. degree from Graz University of Technology, Graz, Austria, in 2003. He was a Photogrammetric Engineer in the private industry and held researcher positions at Graz University of Technology, Monash University, Melbourne, VIC, Australia, and ETH Zuürich, Zürich, Switzerland. He was an Assistant Professor of Image Understanding with TU Darmstadt, Darmstadt, Germany, in 2009. Since 2010, he has been a Tenured Professor of Photogrammetry and Remote Sensing with ETH Zürich. His research interests include computer vision, photogrammetry, and remote sensing.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

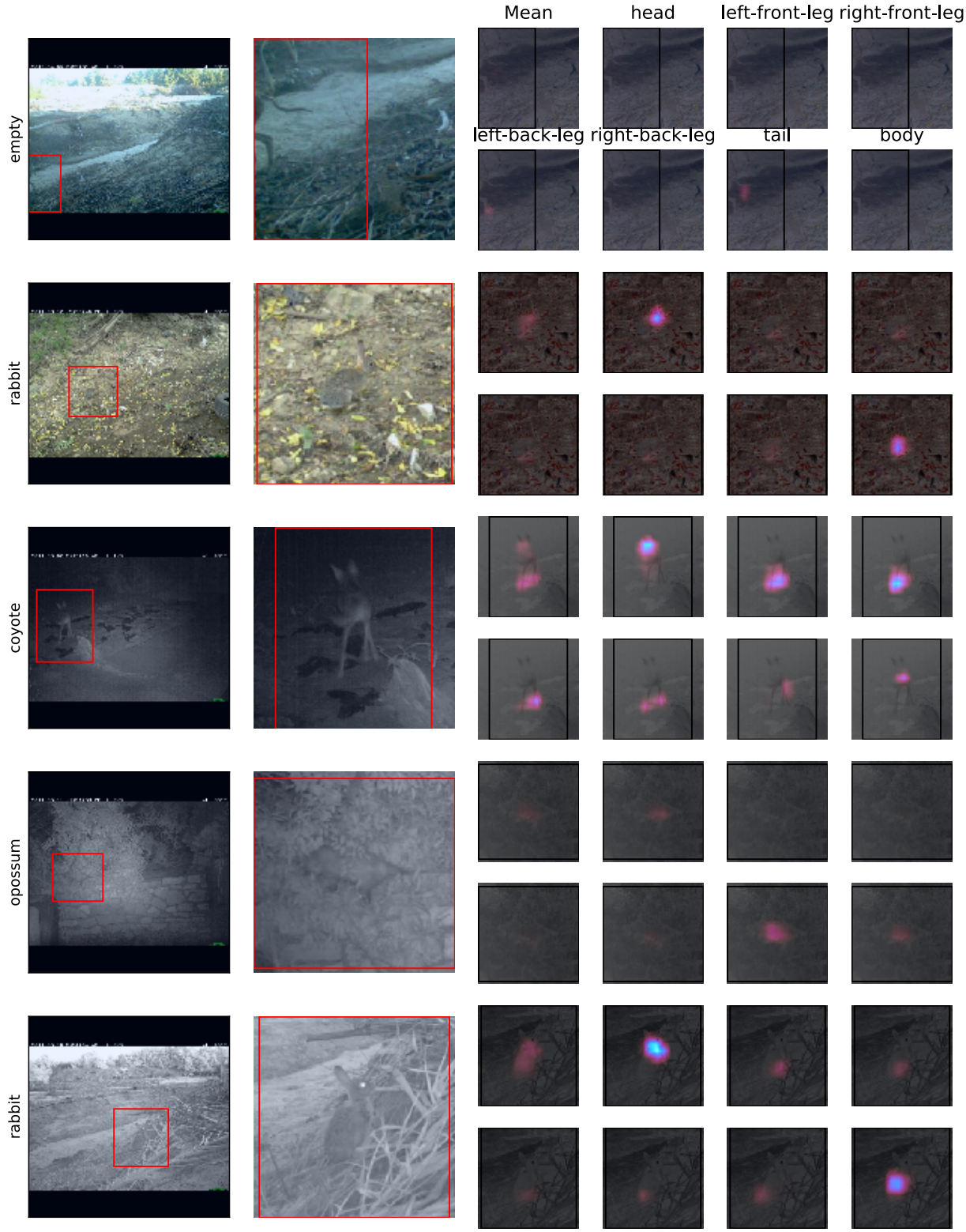


Jan Dirk Wegner is associate professor at University of Zurich and head of the EcoVision Lab at ETH Zurich. Jan was PostDoc (2012-2016) and senior scientist (2017-2020) in the Photogrammetry and Remote Sensing group at ETH Zurich after completing his PhD (with distinction) at Leibniz Universität Hannover in 2011. He was granted multiple awards, among others an ETH Postdoctoral fellowship and the science award of the German Geodetic Commission. Jan was selected for the WEF Young Scientist Class 2020 as one of the 25 best researchers world-wide under the age of 40 committed to integrating scientific knowledge into society for the public good. He is vice-president of ISPRS Technical Commission II, chair of ISPRS II/WG 6 ”Large-scale machine learning for geospatial data analysis”, director of the PhD graduate school ”Data Science” at University of Zurich, associated faculty of the ETH AI Center, and his professorship is part of the Digital Society Initiative at University of Zurich.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

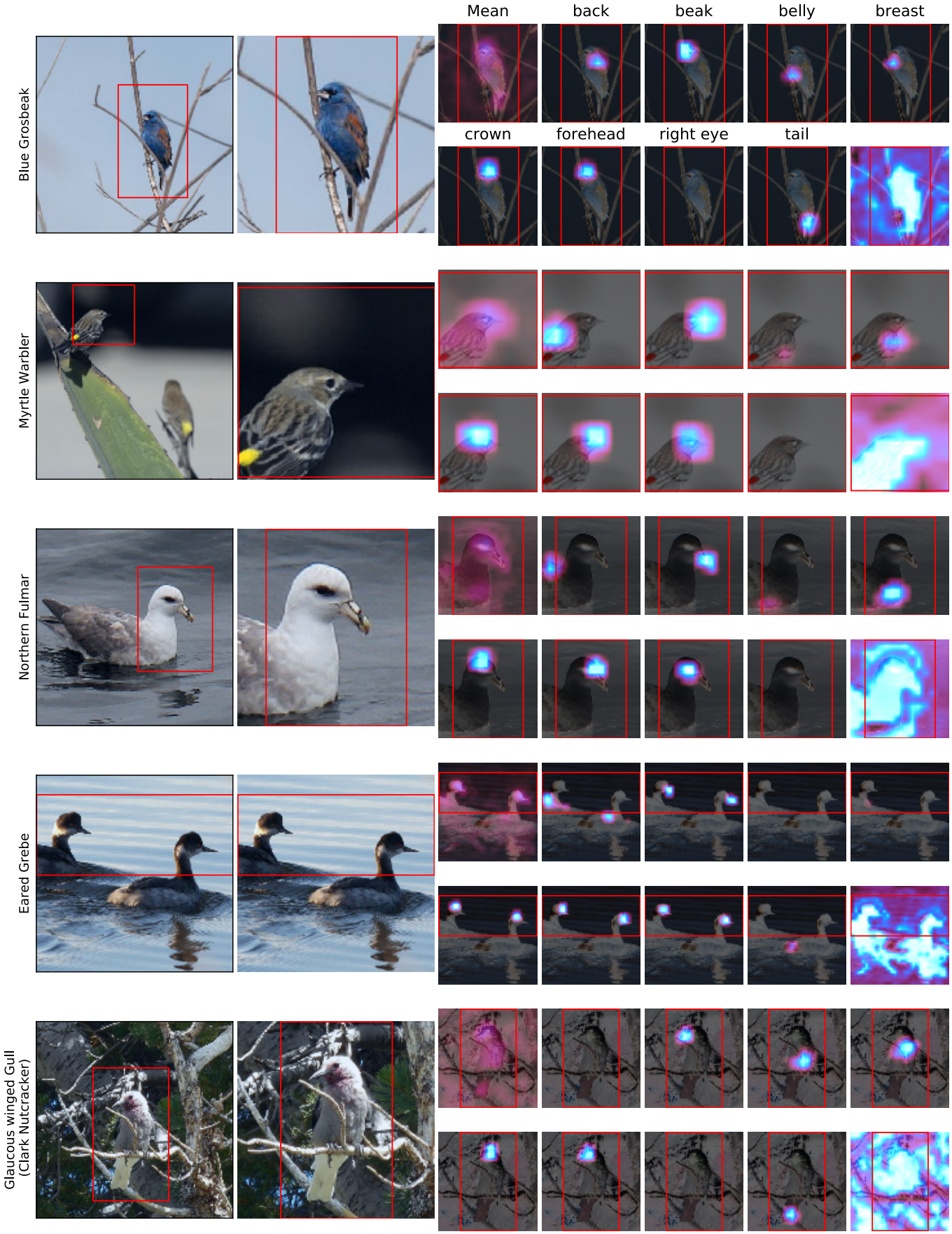
Fig. 12. Random samples from CCT Cis Test Set. Red bounding-box is derived from the predicted attention maps. From left to right: Input image (marked with predicted class), zoom to attended region and keypoint specific attention maps.



This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3316718

Fig. 13. Random samples from iBirds2017 Test Set. The red bounding-box is derived from the predicted attention maps. From left to right: Input image (with predicted class and misclassified samples include true class in parenthesis), zoom to attended region, keypoint specific attention maps, and average of complementary attention maps.



This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/