Zero-Shot Bird Species Recognition by Learning from Field Guides

Andrés C. Rodríguez¹, Stefano D’Aronco ¹, Rodrigo Caye Daudt¹, Jan D. Wegner¹,² and Konrad Schindler¹

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

Abstract

We exploit field guides to learn bird species recognition, in particular zero-shot recognition of unseen species. The illustrations contained in field guides deliberately focus on discriminative properties of a species, and can serve as side information to transfer knowledge from seen to unseen classes. We study two approaches: (1) a contrastive encoding of illustrations that can be fed into zero-shot learning schemes; and (2) a novel method that leverages the fact that illustrations are also images and as such structurally more similar to photographs than other kinds of side information. Our results show that illustrations from field guides, which are readily available for a wide range of species, are indeed a competitive source of side information. On the iNaturalist2021 subset, we obtain a harmonic mean from 749 seen and 739 unseen classes greater than 45% (@top-10) and 15% (@top-1). Which shows that field guides are a valuable option for challenging real-world scenarios with many species.

Keywords: Zero-shot learning, fine-grained classification, contrastive learning, bird species recognition, illustrations.

1 Introduction

Fine-grained species recognition is essential for biodiversity monitoring. Identifying the species of observed animals and plants is the basis for several important biodiversity indicators, e.g., the number of different species in an area, the abundance of individual species, and their geographical distribution. Many species are locally or globally threatened by human activities, making it all the more important to monitor their distributions and support conservation efforts [10].

A bottleneck has long been the collection of enough observations. In the last years, the cooperation of experts and nature enthusiasts has enabled the emergence of community science projects. Volunteers record and share images and locations of their observations, which experts can curate and organise to obtain large-scale databases for biodiversity monitoring. Examples include the iNaturalist [19] and eBirds [36] projects. The eBirds platform alone has accumulated >34 million images for bird species, from ≈800’000 contributors. Those databases open up the possibility to train automatic species recognition systems, which would be a valuable asset for scalable biodiversity monitoring.

In principle, automatic species identification can capitalise on the recent advances in computational object recognition, which now achieves human-level performance for some tasks, especially those which require specialised expertise not available to most observers. However, a necessary prerequisite for such performance is having access to a large volume of labelled training data. Due to the sheer number of species in most ecosystems, many of which are rare or at least rarely spotted, it is extremely challenging, and perhaps practically impossible, to gather a sufficient number of training samples for every one of them. For example, the iNaturalist 2021 dataset [41] comprises 1’486 bird species, yet the Birds of the World collection [7] reports over 10’000 known bird species.

¹E.g., on ImageNet computers outperform most humans when it comes to recognising different dog breeds, as well as different species of mushrooms.
When not enough data can be collected, one must resort to advanced machine learning strategies that may still be able to deliver acceptable recognition results (but typically will not attain the performance of a model supervised with enough data). For instance one can use few-shot learning if only few labelled data are available for certain classes [44]. In the extreme case, Zero-Shot Learning (ZSL) refers to the scenario where no training samples are available at all for some target classes [12, 25, 1, 46]. Clearly, this will require some sort of per-class “description” (side information) beyond images, as the training set by definition does not contain any examples that could even define such unseen classes.

Traditionally, professional as well as amateur observers rely on field guides to recognise animal and plant species in nature. This works remarkably well. Even if new formats of field guides arise, such as interactive maps and mobile apps to aid species recognition [11], the basic principle remains the same: the field guide provides a clear, representative visual example that emphasises the distinctive properties and visual cues needed to identify a species and to discriminate it from similar ones.

The question we explore in this paper is: can we exploit illustrations from field guides to mitigate the lack of training data for some classes? Field guides are easily accessible, cover a broad range of species, and although they normally contain only few images of a species – sometimes only a single illustration – they allow humans to identify it in most cases. One can think of a field guide as a collection of manually created, discriminative class prototypes: the artists who create the illustrations are highly specialised professionals, and they make a conscious effort to render each species such that the illustration not only faithfully reproduces attributes like colour and shape, but optimally typifies its peculiarities and makes it distinguishable from other species. Moreover, illustrations are available also for rare, endangered and even for extinct species.

Although naturalistic illustrations in many ways resemble photographs, a joint supervised training without additional regularisation will be biased towards photographic textures, such that the classifier will tend to collapse towards the seen classes. To tackle this problem, we propose to interpret illustrations as species-specific attribute information and leverage them in a zero-shot setting. At this point, a technical difficulty arises: Existing ZSL algorithms ingest attributes in the form of a low-dimensional vector, and it turns out that the high dimensionality of illustrations, compared to conventional binary attributes (e.g., belly shape, or eye colour), challenges existing ZSL algorithms. In this work we tackle this problem and demonstrate how illustrations from birding field guides can be exploited for zero-shot learning.

We make the following contributions: (1) We introduce the Bird Illustrations of the World (Billow) dataset for fine-grained zero-shot classification of bird species; (2) we propose a contrastive embedding of the illustrations that enables existing ZSL algorithms to leverage Billow; and (3) we propose a novel zero-shot learning scheme better suited for side-information in the form of illustrations. Its fundamental principle is to train a model that can process either illustrations or photographs and in both cases arrives at the same predictions and aligns the class prototypes from the illustrations with the photographs, see Figure 1.

We use Billow in conjunction with three other widely used datasets, namely CUB [45] and the bird
subsets of iNaturalist 2017 [42] and iNaturalist 2021 [41]. With the help of those datasets we compare our method to the state-of-the-art in ZSL as well as domain adaptation. The experiments show that Billow matches the performance of other, more structured forms of side information, confirming the hypothesis that field guides are a valuable auxiliary source of information for species recognition. We hope that our work will encourage further research into biodiversity mapping, and may serve as a first step towards unlocking the treasure trove of biological field guides, beyond Billow.

2 Related Work

The use of illustrations for zero shot learning is not new. Early work has attempted to use digital characters as side information for character recognition in ZSL [26]. In [4], authors use user generated pose graphics as side information for action recognition in a ZSL setting. Sketches of object have been used for image retrieval tasks [33].

ZSL. Early work on ZSL focused on defining class embedding spaces and visual spaces, then measuring some matching metric to predict a class [12, 25, 1]. The embedding space used for matching plays a crucial role [50, 35]. Mapping to a space closer to the class embedding can lead to a hubness problem, where a classifier is strongly biased to predict only a subset of labels.

Current state-of-the-art methods rely on generative models to map class embeddings into a visual embedding space to avoid such a problem. They use a generator to create synthetic samples that attempt to emulate real samples from unseen classes. These samples are then used to supervise the training of a machine learning algorithm along with the examples from the seen classes.

One of the first studies to use a generative approach in ZSL is [47]. They use a Generative Adversarial Network (GAN) to synthesize visual examples of the unseen classes using class descriptions as conditional information. An additional classification loss enforces that the generated features have sufficient discriminative information. TFVAEGAN [28] models the embedding space using a variational formulation. The method also has a feedback network that modulates the latent representations to further improve performance. Invertible Zero-shot recognition flows [34] use invertible layers in order to learn a mapping from the class description to the visual features. Counterfactual ZSL [49] exploits “sample attributes” from the training classes to create synthetic samples with class attributes from the unseen classes. In CE-GZSL [16] authors use a contrastive loss which allows them to use not only class-wise but also instance-wise supervision. LsrGAN [43] propose a novel semantic regularized loss which promotes visual features that reflect the semantic relationships between seen and unseen classes.

Side-information in ZSL. Generative approaches work very well in cases where the side information has low dimensional embedding, and can be used as a deterministic condition by the generator to synthesize samples from unseen classes. While there are currently many types of side information used in ZSL, all of them are rather low-dimensional. In [3], authors evaluate different supervised and unsupervised embeddings for ZSL. Such types of side information include manually created binary attributes per class [2], visual descriptions [32], automatic embeddings from Wikipedia descriptions [3], and more recently learned embeddings of DNA sequences for fine-grained species classification [5]. However, it remains unclear how to use these methods if the side information lives in a high dimensional space, as it is the case for illustrations from field guides, without a low-dimensional embedding step as preprocessing.

Domain adaptation. Given that illustrations and photograph are similar in nature (as opposed to images and text embeddings or DNA sequences, for instance) we also draw from literature regarding Domain Adaptation (DA). Such studies aim to improve the performance of a model trained on the source domain in which enough data are available for supervision and applied on a target domain in which the available data are not enough for supervision.

The most versatile form of DA is Unsupervised Domain Adaptation (UDA), in which no supervision signal is available for the target domain. Most methods aim to match the distributions of the source and target domains in a latent space, either explicitly or implicitly [37, 38, 20]. Ganin et al. proposed using a gradient reversal layer which aims to make the samples from both domains indistinguishable in a representation space [14]. Since, many other methods have been proposed that use adversarial training of a discriminator to enforce alignment of the domains in a latent space [13, 39, 51]. These methods have been shown to work well in established domain adaptation benchmarks, but use cases for fine-grained classification are somewhat
unexplored.

Other studies focus on the case where some supervised samples are available in the target domain. Well established methods exist to combine source and target data to improve a ML system’s performance on target domain data [9]. Motiian et al. propose a unified framework for domain adaptation of deep models using a Siamese architecture to align different visual domains [27]. Notably, [48, 23] use memory banks of latent representations of instances from both source and target domains in a few-shot learning setting, which are then used to create class prototypes for each domain.

However, applying these methods directly for ZSL is not straightforward, as seen classes tend to dominate predictions from the target domain if no additional regularization is done. Our novel method aims at closing the domain gap from illustrations and photographs in a ZSL setting.

3 Bird Illustrations of the World Dataset

We introduce the Bird Illustrations of the World (Billow) dataset for Generalized Zero-Shot Learning (GZSL) in fine-grained classification. The dataset consists of illustrations from the Birds of the World project [7] collected and organized by the Cornell Lab of Ornithology. Billow includes 22,351 illustrations of 10,631 different species, 2,279 genera, 249 families, and 41 orders.

All illustrations in the dataset share a standardized graphical style: side view in front of white background, in neutral pose. Most species have illustrations for a male and a female specimens, some also include a close-up of the bird’s head.

Unfortunately, the original artworks may only be accessed with a valid subscription to the Birds of the World project, and are subject to restrictive licensing conditions. To ensure reproducibility and support further research and comparisons, we will make available the latent encodings from both our approaches (see Sec. 4) Further details are given in the supplementary material.

3.1 Illustrations for Zero-Shot Learning

The list of species included in Billow covers almost all species of the CUB dataset (196 out of 200), and also the overwhelming majority of bird species from iNaturalist 2017 (895 out of 954) and iNaturalist 2021 (1485 out of 1486). Note that the opposite is not true: even the 1485 bird classes of iNaturalist 2021 are only a small fraction of the 10,631 species covered in Billow. This raises the question whether we can leverage the rich information implicit in the Billow dataset and combine it with a dataset of photographs, to advance the state-of-the-art in fine-grained (bird) species recognition.
Table 1: Zero-shot splits of iNaturalist bird classes. $N$ and $K$ denote the numbers of samples, respectively and classes, in each set.

| Dataset | Train | Val |
|---------|-------|-----|
|         | $N$   | $K$ |
| iNat2017| 97,067| 381 |
|         | 8,626 | 381 |
|         | 11,073| 514 |
|         | 2,204 | 87  |
|         | 3,613 | 177 |
|         | 2,175 | 110 |
|         | 3,081 | 140 |
| iNat2021| 211,027| 749 |
|         | 7,490 | 749 |
|         | 7,360 | 736 |
|         | 1,680 | 168 |
|         | 2,860 | 286 |
|         | 1,580 | 158 |
|         | 1,240 | 124 |

For CUB200, there is a default split into 150 seen and 50 unseen classes [46]. CUB200 uses common names, not scientific names. Hence, some previous works had to map the common names to scientific ones, e.g., to leverage the hierarchical label structure [6], or to utilize genetic information [5]. We have revised and merged these assignments, and only retain mappings for which there is a one-to-one correspondence between the common and scientific names in Wikipedia. In this way, we obtain a match in Billow for 196 out of the 200 CUB classes.

For the iNaturalist datasets, we propose a seen/unseen split. Similar to previous ZSL works that use ImageNet [13, 29], we construct several versions of the unseen portion, which have different distance to the seen classes in the label hierarchy. In this way, we can assess the performance of ZSL for unseen classes that are increasingly distant from the seen ones. We define the $i$-hop set as the set of all classes whose distance to the nearest seen class in the taxonomic tree is equal to $i$ (i.e., they belong to the same super-class at the $i$-th taxonomic level). For example, the classes in the 2-hop set share the family (2nd level) with at least one seen class, but do not share the same genus with any of them. We consider the species, genus, family and order levels to obtain 0-hop (i.e., seen classes), 1-hop, 2-hop and 3-hop sets. Classes in the 4-hop set do not have members of the same taxonomic group in any level of the seen set.

The intersection of the Aves super-class from iNaturalist 2017 with Billow contains 895 species. These are randomly split into 381 seen and 515 unseen classes. From the unseen ones we construct the 4 different $i$-hop sets for validation. We repeat the same procedure with iNaturalist 2021: the intersection of its Birds super-category with Billow has 1'485 species. Our split has 749 seen classes and 736 unseen classes. See Fig.2 for an illustration of the validation splits, and Tab.1 for the number of classes and samples in each of the splits.

4 Method

In ZSL we are given a set of classes $\mathcal{Y}$ made up of two disjoint sets $\mathcal{Y}_{\text{seen}}$ and $\mathcal{Y}_{\text{unseen}}$. Side information $s$ is available for every class in $\mathcal{Y}$. In most cases a single instance of $s$ is available for each class, although this is not a requirement. We can think of $s_y$ as an alternative (possibly incomplete) description of the class $y \in \mathcal{Y}$, e.g., a text, a list of semantic attributes, or a gene sequence. A training set of pairs $(x, y)$ is available exclusively for the seen classes, where in computer vision the $x$ are typically photographic images. The goal of ZSL is to use that information to build a classifier $F(x)$, which can recognize samples from unseen classes $\mathcal{Y}_{\text{unseen}}$. Similarly, Generalized Zero-Shot Learning (GZSL) methods aim for good classification performance on test samples from both the seen and unseen classes $\mathcal{Y} = \mathcal{Y}_{\text{seen}} \cup \mathcal{Y}_{\text{unseen}}$.

The side information $s$ provides cues about similarities (common features) between classes and serves as a bridge that shall enable the recognition of the unseen classes, despite not having training examples $x$ for them. Commonly the side information comes in the form of low-dimensional vectors, like for instance binary presence/absence flags for a number of attributes. On the contrary, our illustrations are images of a specific style / domain.

In order to utilise them for ZSL, we explore two different strategies. We first present a learned embedding of the illustrations such that one can feed them into existing ZSL methods. We then develop a more advanced method, where a single network is trained to map both illustrations and photographs to similar latent representations, in order to better leverage their similar structure.
4.1 Contrastive Encoding of Illustrations

Let $D_{\text{side}}$ be the set of pairs $(s, y)$ where $s \in S$ is an illustration associated with class $y \in \mathcal{Y}$. To turn illustrations into low-dimensional vectors, we use an encoding network $E$ that produces an embedding $z = E(s)$. These embeddings should preserve discriminative class information, so we simply add a classification head $\hat{y} = C(z)$ and optimize $E$ and $C$ with a cross-entropy loss $L_{\text{cls}}(\hat{y}, y_i)$. Here, $y$ represents the one-hot encoding of $y$.

However, the ability to discriminate classes is not enough. Rather the embeddings should also live in a metric space where pairwise differences between them are meaningful, so as to handle the zero-shot problem. One way to achieve this is to model the overall distribution of illustrations in an embedding space. For instance, one can employ a Variational Auto-encoder (VAE), that assumes the embedding $z$ to model a prior distribution in the latent space from which it is possible to draw samples and decode them to the original input space $X$. This approach, however, risks to reduce the representation power of $z$, if it is too strongly regularized by the prior distribution (see experiments in Section 5.3). An alternative is to use a contrastive loss, which promotes an embedding space with a uniform distribution over the unit-sphere and has proved useful for downstream tasks. We apply a projection and normalization head $\hat{z} = h(z)$ to our embeddings before computing the contrastive loss. Following [22], the contrastive loss function is defined as:

$$L_{\text{cont}}(\hat{z}_i) = -\frac{1}{||P(i)||_1} \sum_{p \in P(i)} \log \frac{\exp(\hat{z}_i \cdot \hat{z}_j / \tau)}{\sum_{j \in B} \exp(\hat{z}_i \cdot \hat{z}_j / \tau)},$$

where $P(i)$ is an indicator function for samples in the training batch $B$ that have the same class label as $x_i$, and $\tau \in \mathbb{R}^+$ is a tunable temperature parameter. Finally we train $F$, $C$ and $h$ with both classification and contrastive losses: $L = L_{\text{cls}} + L_{\text{cont}}$. As final representation of class $y$, we compute

$$\phi(y) = \eta \left( \sum_{s \in \mathcal{S}_y} E(s) \right),$$

where $\mathcal{S}_y$ is the set of all illustrations available for class $y$, and $\eta(z) = z/||z||$ denotes $L^2$-normalization. The embeddings $\phi(y)$ derived from illustrations can then be used as class descriptors in different existing ZSL methods.

4.2 Prototype Alignment

In contrast to other types of side information for ZSL, illustrations also belong to the visual domain. We leverage this property and propose Prototype Alignment (PA) for ZSL with visual side information. Inspired by [18], we explore a view of the problem through the lens of few-shot domain adaptation: The source domain are illustrations, the target domain are natural, photographic images.

Let $s$ and $x$ be samples from the source domain $S$ and the target domain $X$, respectively. We have access to samples from all classes $\mathcal{Y}$ in the source domain, but only to samples of the seen classes $\mathcal{Y}_{\text{seen}}$ in the target domain. Furthermore, we also do not have unlabelled samples of unseen classes in the target domain.

We train a feature extractor network $F$ that takes input samples from either domain and outputs a latent representation $z$. The last operation in $F$ is an $L^2$-normalization layer $\eta(\cdot)$, as also used in Eq. (2). During training, we keep a memory bank in each domain, with a prototype $z$ of each class. For the illustrations in the source domain, that representation can be interpreted as the class embedding $\phi(y_k)^{(s)}$ that is used for ZSL. Note that, in contrast to previous approaches [23, 18], we do not keep an instance-wise memory bank, which would lead to intractable memory demands for larger datasets.

For the sake of simplicity, we omit the domain indicator in the following where possible. In every iteration, we update the memory bank in each domain with the latent representation of the new samples, with momentum $m$:

$$\phi(y_k) \leftarrow \eta((1 - m)z_k + m\phi(y_k)).$$

To promote compact and discriminative class representations, we apply a contrastive in-domain loss similar
to Eq. 4 via a projection head $h$:

$$L_c(z_i, \phi(y_i)) = -\log \frac{\exp(h(z_i) \cdot h(\phi(y_i)) / \tau)}{\sum_{k \in C} \exp(h(z_i) \cdot h(\phi(y_k)) / \tau)}.$$

In contrast to [15], we refrain from applying a cross-domain contrastive loss to close the domain gap. Instead, we sidestep the gap by directly using the class prototypes from both domains for classification, so as to force the network $F$ to produce class-discriminative features. To obtain class logits, we compute the dot-product between an image embedding $z$ and the embeddings $\phi(Y)$ of the classes from both domains, $\hat{y}^{(z)} = z \cdot \phi(Y)^{(z)}$. These serve as input to a cross-entropy loss $L_{\text{cls}}$ for supervision:

$$L_{CE}(\hat{y}^{(z)}, y) = L_{\text{cls}}(\hat{y}^{(z)}, y) + L_{\text{cls}}(\hat{y}^{(x)}, y).$$

Eq. 5 encourages sample representations that are discriminative w.r.t. prototypes from the other domain, which in turn aligns the two domains. Note also that the second term in Eq. 5 is only computed for seen classes, as it depends on $\phi(Y_{\text{seen}})^{(z)}$. The complete loss function is $L = L^{(s)} + L^{(x)}$, such that

$$L^{(d)} = \sum_{i \in B^{(d)}} \left( \lambda_c^{(d)} L_c(z_i, \phi(y_i)) + \lambda_{ce}^{(d)} L_{CE}(\hat{y}_i^{(s)}, \hat{y}_i^{(z)}) \right),$$

where $B^{(d)}$ denotes indices of the samples from domain $d \in \{S, X\}$ in the mini-batch. Hyperparameters $\lambda_c, \lambda_{ce}$ are used to balance the different losses. At test time, we can simply use the logits $\hat{y} = F(x) \cdot \phi(Y)^{(z)}$ for classification.

5 Experiments

Experimental Setup. All of our experiments are developed in PyTorch [30] and trained with Nvidia GTX 1080 GPUs. For our contrastive encoding of illustrations we use a ResNet-18 [17] pretrained on ImageNet to create the embeddings $\phi$ of each image. A pretrained ResNet-101 backbone without fine-tuning was used to obtain a 2048-dim vector representation of each image.

The PA experiments used a pretrained ResNet-101 on ImageNet and used the Adam optimizer [24] with a learning rate of $10^{-4}$, convolutional layers’ learning rate was scaled by 0.1. All experiments with iNaturalist datasets ran for 40,000 iterations and experiments on CUB200 for 200 epochs. We define $\tau = 0.1$ in all our experiments. Following the convention in GZSL literature, we evaluate the performance of each algorithm using held out sets of samples in the seen classes (S) and unseen classes (U) separately. The harmonic mean of these two numbers (H) is also reported. We will make our all our code available for reproducibility.

5.1 Zero-Shot Species Recognition on iNaturalist 2017 and 2021

Class embeddings were created by using contrastive embedding with Billow illustrations (see Sec. 4.4). We report experiments using these embeddings with TFVAEGAN [28]. Results using Billow illustrations use our proposed PA method. On all iNaturalist datasets we observed an improved performance of PA over TFVAEGAN. This was consistent on all three datasets evaluated on all top-k metrics. With PA we observed a harmonic mean H@top-5 of 35.1% and 35.6%, for iNat2021 and its iNat2021mini respectively (see Fig. 3a). For TFVAEGAN we observed a decreased performance with the larger training dataset for iNat2021 (H@top-5 19.1% and 24.6%). This was observed over all i-hop sets consistently. These results, indicate that further regularization for large-scale datasets might be needed.

Our results show that the hierarchical distance w.r.t. to the seen classes correlates strongly with performance on the unseen datasets (see Fig. 3b). We observe a constant reduction on performance for classes with a larger taxonomic separation to seen classes. This is aligned with what has been observed in ImageNet for ZSL [13, 29, 21]. However, it seems that ZSL on ImageNet is more challenging than in iNaturalist as the label distance on ImageNet classes might not be as meaningful as with taxonomic distances of species.

Analysis of the number of synthetic samples. Generative approaches in GZSL generate synthetic examples from unseen classes, which is usually set to $N_{syn} = 100$ [28, 10, 43]. The samples are added to the
training set to supervise the training of a classifier. We investigate the effect that bigger values of $N_{syn}$ could have on larger datasets with higher numbers of unseen classes. We kept all other hyperparameters constant. Results can be found in Figure 4. We observe that when using TFVAEGAN, increasing $N_{syn}$ results in better performance on unseen classes in all the datasets, but at the cost of a decreased performance on seen classes.

5.2 Zero-Shot Species Recognition on CUB200

As described in Section 3 there are 4 out of 200 classes in CUB which are not present in the Billow dataset. We present experiments on: (1) CUB$_{196}$: the subset of 196 CUB200 classes also contained in Billow, divided into 148 seen and 48 unseen classes; (2) CUB$_{191}$: The subset of 191 CUB200 classes also present in both DNA [5] and Billow datasets. CUB$_{191}$ contains 145 seen and 46 unseen classes. In both cases, the splits respect the the proposed split by [46].

Class embedding vectors were generated from illustrations using our contrastive encoding (See Sec. 4.1). These embeddings were used with TFVAEGAN [28], CE-GZSL [10], and LsrGAN [43] to evaluate their performance as class side information $\phi(y)$ in a ZSL setting. We retrained all baselines with CUB$_{196}$ and CUB$_{192}$ using their respective original implementations. We also compare the results with those obtained by using the following sources of side information: binary attributes [15], visual descriptions [22], DNA [5], and word2vec [3]. Results can be seen in Table 2. These experiments show that the representation power of Billow’s contrastive embedding is comparable to that of word2vec and DNA embeddings. In terms of comparison among the existing methods we can observe that TFVAEGAN achieves the best results on both scenarios.

Furthermore, we show in Table 2 (bottom) a performance evaluation of various supervised and unsupervised domain adaptation methods for solving this problem. Although DANN and ProtoDA did not completely collapse towards the seen classes, they fail to fully translate knowledge from the source domain.
Figure 4: iNaturalist and Encoded Billow with TFVAEGAN. Average of 5 runs with different number of synthetic samples. Top-1: ▼, top-5: ×, top-10: ▲ into the target domain. Our PA approach on the other hand achieves the best performance on Billow illustrations.

**Ablation: backbone size.** We explore the effect of using different backbones in our prototype alignment method. The results in Table 3a show that we achieve better performance using ResNet-18 than all other methods. Larger backbones yield models that perform better in seen classes at the cost of performance on unseen classes.

**Ablation: prototype alignment components.** In Table 3c we show the effect of changing different components of the PA method. First, we observed that setting the projection head \( H \) to be a small Multi Layer Perceptron (MLP) decreased our performance w.r.t. using an identity function instead (row D). It is not fully clear what actually led to such performance drop (for seen and unseen classes). We speculate that the latent space of \( z \) is already too close to the label domain for it to benefit from a projection head. We then analysed whether computing \( \hat{y} \) with a learned linear classifier \( w_{cls} \) instead of using the dot product between domain embeddings could provide better performance (row A). We observed such modification drastically reduces the performance on the unseen classes. Another specific component that is essential to achieve good performance on the unseen classes is the contrastive in-domain loss \( \lambda_c \) (row C) . The classification loss instead seems to be more important for the recognition of the seen classes: Removing the term completely barely changes the performance on the unseen classes while drastically reducing performance on the seen ones (row B). On the other hand having a larger \( \lambda_{\text{ct}} \) tends to boost accuracy in seen classes and at the same time slightly diminish accuracy in unseen ones (row E).

### 5.3 Encoding of Illustrations

Appropriate encoding of the side information can have a strong effect in the overall performance. We evaluated different ways to encode our illustrations to be used with ZSL state-of-the-art methods. From Billow, we created a separate test set and evaluated the predictive power for species classification of each encoding. This is a challenging task, since we only have 13’000 illustrations for training to classify 8646 species. We used a pretrained ResNet-18 network, fine-tuned as described in Sec 4.1, \( \beta \)-VAE [18, 31] and VQ-VAE [40, 31] (both of which use a custom CNN), and pretrained ResNets without fine-tuning. See Tab. 4 for the results. More details can be found in the supplementary material. Larger ResNets achieved better performance at higher levels of the hierarchy (e.g. order), but our approach with ResNet-18 achieves the
Table 2: CUB200 - GZSL Results. Average of 5 runs with different random seeds, ± standard deviation. Best method for each dataset and $\phi(y)$ is bold. † denotes UDA methods that do not use target labels.

| Dataset         | $\phi(y)$       | Model         | H   | S   | U   |
|-----------------|-----------------|---------------|-----|-----|-----|
|                 | Binary attributes |               |     |     |     |
| CUB200          | CE-GZSL         | 53.5 ±0.7     | 59.8 ±1.9 | 48.4 ±0.7 |
|                 | LsrGAN          | 30.9 ±0.6     | 63.6 ±0.2 | 20.4 ±0.5 |
|                 | TFVAEGAN        | **57.6 ±0.2** | 63.4 ±2.2 | **52.8 ±1.4** |
|                 | Visual descriptions |           |     |     |     |
|                 | CE-GZSL         | 65.7 ±0.4     | 66.4 ±0.3 | 65.0 ±0.6 |
|                 | LsrGAN          | 56.3 ±0.4     | 58.7 ±0.3 | 54.2 ±0.8 |
|                 | TFVAEGAN        | **68.1 ±0.4** | **67.8 ±2.1** | **68.4 ±2.1** |
|                 | DNA             | 20.1 ±0.8     | 39.5 ±1.2 | 13.5 ±0.9 |
|                 | LsrGAN          | 7.4 ±0.4      | **69.7 ±0.1** | 3.9 ±0.2 |
|                 | TFVAEGAN        | **24.5 ±0.7** | 30.8 ±0.4 | **20.3 ±1.0** |
|                 | word2vec        | 33.9 ±0.5     | 49.1 ±1.7 | 25.9 ±0.7 |
|                 | LsrGAN          | 26.1 ±0.5     | **62.0 ±0.5** | 16.5 ±0.4 |
|                 | TFVAEGAN        | 34.1 ±0.9     | 45.6 ±1.0 | **27.2 ±0.9** |
|                 | Billow contrastive encoding | | | | |
|                 | CE-GZSL         | 34.8 ±1.0     | 42.7 ±1.5 | 27.9 ±0.9 |
|                 | LsrGAN          | 12.7 ±0.4     | **69.2 ±0.2** | 7.0 ±0.2 |
|                 | TFVAEGAN        | **35.6 ±1.1** | 45.3 ±14.1 | **31.6 ±5.4** |
|                 | Binary attributes |               |     |     |     |
| CUB106          | CE-GZSL         | 53.3 ±0.8     | 59.8 ±1.4 | 48.2 ±1.7 |
|                 | LsrGAN          | 29.7 ±0.9     | **63.8 ±0.4** | 19.4 ±0.8 |
|                 | TFVAEGAN        | **57.4 ±0.2** | 63.8 ±2.8 | **52.3 ±2.0** |
|                 | Visual descriptions |           |     |     |     |
|                 | CE-GZSL         | 65.2 ±0.2     | 66.3 ±1.5 | 64.2 ±1.5 |
|                 | LsrGAN          | 56.0 ±0.2     | 58.4 ±0.3 | 53.8 ±0.2 |
|                 | TFVAEGAN        | **68.1 ±0.3** | **68.8 ±1.3** | **67.4 ±1.1** |
|                 | Billow contrastive encoding | | | | |
|                 | CE-GZSL         | 31.5 ±1.2     | 42.0 ±1.1 | 25.2 ±1.9 |
|                 | LsrGAN          | 11.6 ±0.9     | **69.7 ±0.3** | 6.4 ±0.5 |
|                 | TFVAEGAN        | **35.8 ±1.2** | 45.5 ±13.1 | **31.5 ±5.5** |
|                 | Billow illustrations |        |     |     |     |
|                 | DANN            | 20.3 ±1.6     | 24.3 ±1.8 | 17.5 ±2.3 |
|                 | MDD             | 0.9 ±0.4      | 1.4 ±0.4 | 0.7 ±0.4 |
|                 | MCC             | 6.1 ±0.4      | 6.5 ±0.5 | 5.8 ±0.8 |
|                 | ProtoDA         | 14.4 ±2.0     | 13.8 ±0.9 | 13.8 ±1.8 |
|                 | CCSA            | 0.1 ±0.1      | **73.5 ±0.7** | 0.1 ±0.0 |
|                 | PA (Ours)       | **47.5 ±1.5** | 69.7 ±0.6 | **36.1 ±1.5** |

best performance in the most challenging case of fine-grained species recognition.

6 Conclusion

Our experiments show that using field guides as side information for ZSL is feasible, expanding the set of fine-grained ZSL experiments to datasets with more natural distributions such as iNaturalist2017 and iNaturalist2021.

In our work we focus on illustrations that although have a closer domain w.r.t. to photographs, a naïve implementation from domain adaptation might not yield the best results. iNaturalist experiments show that state-of-the-art ZSL combined with the contrastive encoded illustrations achieves reasonable results over different unseen sets but that our proposed PA has a superior advantage over current methods.

Although, visual descriptions and keywords descriptions have higher accuracies on CUB than with Billow, illustrations are a valid alternative that can be scaled up under more realistic conditions. It is easier to describe new bird species using existing illustrations of them than using the test set of CUB to obtain visual descriptions from the unseen species, which is prone to overfit to the rather small dataset.

This field of research would benefit from future work focusing on incorporating information from the taxonomic trees into the method to improve performance by explicitly modelling species similarity and patristic distances as some studies have previously explored [21]. Furthermore, birds of different sexes in the same species often have very different visual appearances. Using this information, as opposed to assuming a single representation of class visual appearance, could further improve the results.

While we have focused this work on illustrations of birds, there are many other field guides that could potentially be exploited in ZSL. We hope that our work inspires more research in this direction to assist efforts in biodiversity mapping and conservation.
Table 3: Ablations on Prototype Alignment

(a) Backbones - CUB

| Backbone  | H  | S  | U  |
|-----------|----|----|----|
| Resnet-18 | 42.0 | 48.2 | 37.2 |
| Resnet-50 | 47.8 | 64.8 | 37.8 |
| Resnet-101 | 47.5 | 69.7 | 36.1 |

(b) Backbones - iNaturalist, Top-5 Accuracies

| Dataset   | Backbone | H  | S  | U  |
|-----------|----------|----|----|----|
| iNat17    | Resnet-18 | 25.8 | 34.4 | 20.6 |
|           | Resnet-50 | 30.1 | 48.8 | 21.8 |
|           | Resnet-101 | 32.3 | 51.9 | 23.4 |
| iNat21mini | Resnet-18 | 24.2 | 29.4 | 20.6 |
|           | Resnet-50 | 31.7 | 41.9 | 25.5 |
|           | Resnet-101 | 35.6 | 46.1 | 29.0 |
| iNat21    | Resnet-18 | 23.6 | 29.2 | 19.8 |
|           | Resnet-50 | 31.0 | 41.4 | 24.8 |
|           | Resnet-101 | 35.1 | 45.5 | 28.6 |

(c) Hyperparameters - CUB

| λc | b(x)          | λc | ucls | Lcls(y’, y) | H  | S  | U  |
|----|---------------|----|-----|-------------|----|----|----|
| A  | 1 Identity    | 0.1 | learned |            | 14.7 | 44.1 | 8.9 |
| B  | 1 Identity    | 0.1 | φ(Y)(x) | ✓         | 42.7 | 50.6 | 37.0 |
| C  | 0 Identity    | 0.1 | φ(Y)(x) | ✓         | 23.3 | 52.4 | 15.2 |
| D  | 1 MLP         | 0.1 | φ(Y)(x) | ✓         | 39.8 | 50.1 | 32.3 |
| E  | 1 Identity    | 1.0 | φ(Y)(x) | ✓         | 46.3 | 69.9 | 34.6 |
| F  | 1 Identity    | 0.1 | φ(Y)(x) | ✓         | 47.5 | 69.7 | 36.1 |

Table 4: Top-1 accuracy on a test set of Billow samples with different encoders. Evaluation w.r.t. 4 levels of hierarchy.

| Method   | VQ-VAE | β-VAE | Pretrained | Pretrained | Pretrained | Pretrained | CE (ours) |
|----------|--------|-------|------------|------------|------------|------------|-----------|
| Backbone | CNN    | CNN   | ResNet-101 | ResNet-50  | ResNet-18  | ResNet-18  |           |
| Species acc. | 0.1 | 14.2 | 15.5 | 12.0 | 16.5 | 17.7 |
References

[1] Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 819–826 (2013)

[2] Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. IEEE transactions on pattern analysis and machine intelligence 38(7), 1425–1438 (2015)

[3] Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2927–2936 (2015)

[4] Antol, S., Zitnick, C.L., Parikh, D.: Zero-shot learning via visual abstraction. In: European conference on computer vision. pp. 401–416. Springer (2014)

[5] Badirli, S., Akata, Z., Mohler, G., Picard, C., Dundar, M.: Fine-grained zero-shot learning with dna as side information. Advances in Neural Information Processing Systems 34 (2021)

[6] Barz, B., Denzler, J.: Deep learning on small datasets without pre-training using cosine loss. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1371–1380 (2020)

[7] Billerman, S., Keeney, B., Rodewald, P., Schulenberg, T.: Birds of the world. cornell laboratory of ornithology (2020), https://birdsoftheworld.org/bow/home

[8] Burgess, C.P., Higgins, I., Pal, A., Mattthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in $\beta$-vae. arXiv preprint arXiv:1804.03599 (2018)

[9] DAUME III, H.: Frustratingly easy domain adaptation. In: Proc. 45th Annual Meeting of the Association for Computational Linguistics, 2007. pp. 256–263 (2007)

[10] Díaz, S., Settele, J., Bové, E., Guézou, M., Gachet, J., Arneth, A., Balvanera, P., Brauman, K., Butchart, S., Chan, K., Garibaldi, L., Ichii, K., Liu, J., Subramanian, S., Midgley, G., Miloslavich, P., Molnár, Z., Obura, D., Pfaff, A., Polasky, S., Purvis, A., Raizada, J., Reyers, B., Chowdhury, R., Shin, Y., Visseren-Hamakers, L., Willis, K., Zayas, C.: Summary for policymakers of the global assessment report on biodiversity and ecosystem services. Tech. rep., Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services (2019)

[11] Farnsworth, E.J., Chu, M., Kress, W.J., Neill, A.K., Best, J.H., Pickering, J., Stevenson, R.D., Courtney, G.W., VanDyke, J.K., Ellison, A.M.: Next-Generation Field Guides. BioScience 63(11), 891–899 (11 2013). https://doi.org/10.1525/bio.2013.63.11.8, https://doi.org/10.1525/bio.2013.63.11.8

[12] Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. Advances in neural information processing systems 26 (2013)

[13] Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M.A., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 26. Curran Associates, Inc. (2013). https://proceedings.neurips.cc/paper/2013/file/7ce553c900577442771720a370c3c723-Paper.pdf

[14] Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1180–1189. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/ganin15.html

[15] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research 17(59), 1–35 (2016). http://jmlr.org/papers/v17/15-239.html
[16] Han, Z., Fu, Z., Chen, S., Yang, J.: Contrastive embedding for generalized zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2371–2381 (2021)

[17] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)

[18] Higgins, I., Matthey, L., Pal, A., Burgess, C.P., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR (2017)

[19] iNaturalist: iNaturalist. https://www.inaturalist.org accessed: 26-05-2021

[20] Jin, Y., Wang, X., Long, M., Wang, J.: Minimum class confusion for versatile domain adaptation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 464–480. Springer International Publishing, Cham (2020)

[21] Kampffmeyer, M., Chen, Y., Liang, X., Wang, H., Zhang, Y., Xing, E.P.: Rethinking knowledge graph propagation for zero-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11487–11496 (2019)

[22] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 18661–18673. Curran Associates, Inc. (2020). https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1b8b0f2af1a94af8-Paper.pdf

[23] Kim, D., Saito, K., Oh, T.H., Plummer, B.A., Sclaroff, S., Saenko, K.: Cross-domain self-supervised learning for domain adaptation with few source labels. arXiv preprint arXiv:2003.08264 (2020)

[24] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[25] Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 951–958 (2009). https://doi.org/10.1109/CVPR.2009.5206594

[26] Larochelle, H., Erhan, D., Bengio, Y.: Zero-data learning of new tasks. In: AAAI. vol. 1, p. 3 (2008)

[27] Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: IEEE International Conference on Computer Vision (ICCV) (2017)

[28] Narayan, S., Gupta, A., Khan, F.S., Snoek, C.G., Shao, L.: Latent embedding feedback and discriminative features for zero-shot classification. In: European Conference on Computer Vision. pp. 479–495. Springer (2020)

[29] Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650 (2013)

[30] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)

[31] Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. Advances in neural information processing systems 32 (2019)

[32] Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 49–58 (2016)

[33] Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Transactions on Graphics (TOG) 35(4), 1–12 (2016)
[34] Shen, Y., Qin, J., Huang, L., Liu, L., Zhu, F., Shao, L.: Invertible zero-shot recognition flows. In: European Conference on Computer Vision. pp. 614–631. Springer (2020)

[35] Shigeto, Y., Suzuki, I., Haru, K., Shimbo, M., Matsumoto, Y.: Ridge regression, hubness, and zero-shot learning. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Soares, C., Gama, J., Jorge, A. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 135–151. Springer International Publishing, Cham (2015)

[36] Sullivan, B.L., Wood, C.L., Iliff, M.J., Bonney, R.E., Fink, D., Kelling, S.: ebird: A citizen-based bird observation network in the biological sciences. Biological conservation 142(10), 2282–2292 (2009)

[37] Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. p. 2058–2065. AAAI’16, AAAI Press (2016)

[38] Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: Hua, G., Jégou, H. (eds.) Computer Vision – ECCV 2016 Workshops. pp. 443–450. Springer International Publishing, Cham (2016)

[39] Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

[40] Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems 30 (2017)

[41] Van Horn, G., Cole, E., Beery, S., Wilber, K., Belongie, S., Mac Aodha, O.: Benchmarking representation learning for natural world image collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12884–12893 (2021)

[42] Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018)

[43] Vyas, M.R., Venkateswara, H., Panchanathan, S.: Leveraging seen and unseen semantic relationships for generative zero-shot learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 70–86. Springer International Publishing, Cham (2020)

[44] Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: A survey on few-shot learning. ACM Comput. Surv. 53(3) (Jun 2020). https://doi.org/10.1145/3386252, https://doi.org/10.1145/3386252

[45] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)

[46] Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE Transactions on Pattern Analysis and Machine Intelligence 41(9), 2251–2265 (2019). https://doi.org/10.1109/TPAMI.2018.2857768

[47] Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5542–5551 (2018)

[48] Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K., Vincentelli, A.S.: Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13834–13844 (2021)

[49] Yue, Z., Wang, T., Sun, Q., Hua, X.S., Zhang, H.: Counterfactual zero-shot and open-set visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15404–15414 (2021)

[50] Zhang, L., Xiang, T., Gong, S.: Learning a deep embedding model for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2021–2030 (2017)
[51] Zhang, Y., Liu, T., Long, M., Jordan, M.: Bridging theory and algorithm for domain adaptation. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 7404–7413. PMLR (09–15 Jun 2019), https://proceedings.mlr.press/v97/zhang19i.html
7 Supplementary Material

7.1 Dataset availability

For reproducibility, we describe here how to access all illustrations from Billow. We have used all the images available by November 1st 2021. At this point there were 10,631 species in *Birds of the World*.

The encoded dataset using our contrastive embedding can be downloaded from our repository. The illustrations of Billow can be downloaded using our python script, please note that any further use is subject to the licence of use of [7].

7.2 Experiments with DA baselines

Table 5: Caption

| Method | H       | S       | U       |
|--------|---------|---------|---------|
| DANN   | 11.46 ± 0.47 | 13.00 ± 1.06 | 10.297 ± 0.46 |
| Resnet-18 | 0.03 ± 0.05 | 1.19 ± 0.22 | 0.02 ± 0.03 |
| MCC    | 4.01 ± 0.22 | 4.38 ± 0.46 | 3.75 ± 0.39 |
| DANN   | 15.23 ± 1.07 | 16.27 ± 0.60 | 14.37 ± 1.53 |
| Resnet-50 | 0.23 ± 0.52 | 0.58 ± 0.33 | 0.85 ± 1.16 |
| MCC    | 6.17 ± 0.39 | 5.78 ± 0.21 | 6.66 ± 0.72 |
| DANN   | 20.26 ± 1.04 | 24.29 ± 1.78 | 17.55 ± 2.35 |
| Resnet-101 | 0.88 ± 0.43 | 1.38 ± 0.42 | 0.69 ± 0.45 |
| MCC    | 6.12 ± 0.43 | 6.55 ± 0.46 | 5.77 ± 0.57 |

7.3 Encoding of Billow

To evaluate the quality of our encoding method we measured how well can we predict the class at different hierarchical levels. Out of the 10'631 we take species, which genus’ has at least 5 species and create a train/val split from them. This results in 18'489 illustrations: 13362, 1908 and 3219 for train, validation and test, respectively. In all splits there are samples of 8646 species, 956 genus, 175 families and 33 orders. We evaluate top-1 and top-10 accuracy on all 4 levels of the hierarchy.

For training we explored VAE generative models to encode our dataset. VAE consist of two networks an encoder $E$ and a decoder $D$. A regular Auto-encoder would simply use the output of the encoder and feed it to the decoder to reconstruct the input. Which results in the following reconstruction loss: $L = d(x, D(E(x)))$, where $d$ is usually a euclidean distance between the input and the reconstruction. VAEs assume prior on the output of the encoder $p_z$ and maximize the log-likelihood of the reconstruction produced by $D$ over the entire prior distribution $p_z$.

Modelling such a distribution would be desirable in our case as we will use the embedding and the distances between them for ZSL. Hence we explore two variations of VAE: $\beta$-VAE [8] and VQ-VAE [31]. Our motivation to test these 2 variations of VAE is to explore the effect of different priors on the latent distribution $p \sim z$. $\beta$-VAE uses a more constrained information bottleneck on the embedding $z$ than vanilla-VAE (i.e. $\beta > 1$) to obtain a disentangled representation $z$. As a reconstruction, we slowly increase the bottleneck capacity over training as proposed by [8]. VQ-VAE on other hand imposes a discrete distribution over the embedding $z$, this allows to control the information bottle neck by imposing a very small dimension on the discrete distribution.

Additionally we fine-tuned a ResNet classifier pretrained on ImageNet. The classifier was supervised by $L = L_{cls} + L_{cont}$, where $L_{cont}$ is as defined in Eq. [1] VAE experiments were trained their corresponding reconstruction loss and the supervision loss $L_{cls}$.

Once the model was trained we evaluated its predictive power by feeding the embedding $z$ into a small MLP network and trained to predict the level of the label $k$. The results on the test set in Table 4 show that $\beta$-VAE does not perform close to the ResNet models. Along with our fine-tuned ResNet-18, we include results on pretrained models on Image-Net without fine-tuning the backbone. As expected larger networks achieved
higher performance, but our fine-tuning on ResNet-18 improved performance in the most challenging case of fine-grained species recognition. Our ResNet-18 already achieved 100% accuracy on the training set and observed over-fitting on the validation set after fine-tuning. For this reason we decided to keep the fine-tuned ResNet18 as the default encoder for billow.