FINE-GRAINED BAYESIAN ZERO-SHOT OBJECT RECOGNITION

by

Sarkhan Badirli

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Computer Science

West Lafayette, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Murat Dundar, Co-Chair

Department of Computer and Information Science, IUPUI

Dr. Clifton W. Bingham, Co-Chair

Department of Computer Science, West Lafayette

Dr. George Mohler

Department of Computer and Information Science, IUPUI

Dr. Bedrich Benes

Department of Computer Graphics Technology, West Lafayette

Approved by:

Dr. Kihong Park

Dedicated to my inspiring parents, beloved wife, supportive brother, and future kids.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to thank my PhD advisor Dr. Murat Dundar for his tremendous guidance and continuous patience in every stage of my PhD. Apart from his insights and efforts that lead to all these good work, I am grateful for his support that helped foster my ability to learn and grow as an individual. I am leaving this institution with the hope and desire that all the knowledge, tools and advises I harvested here will lead to positive changes in the world.

I would like to extend a special gratitude to Dr. Zeynep Akata, Dr. George Mohler and Dr. Christine Picard for their valuable contributions and feedback to all the papers we published together. I would also like to thank my committee members, Dr. Christopher Clifton and Dr. Bedrich Benes for providing their constructive feedback in the course of this dissertation and to Nicole Wittlief for helping me with administrative work in these five years. I am very thankful to all my friends and colleagues in Indiana who made this long journey joyful. It was a privilege to be around so many smart, kind, helpful and awesome people.

Finally, I must express my very profound gratitude to my parents Elba, Niyazi, my brother Tural and my beautiful wife Sabina for their constant support and endless love. I am very fortunate and grateful to have them beside me. Thank you, dad, for giving me encouragement and vision to pursue my PhD in the United States.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

BOLD	Barcode of Life Database
BZSL	Bayesian Zero-shot Learning
CBZSL	Constrained BZSL classifier
CNN	Convolutional Neural Networks
COI	Cytochrome C Oxidase I gene
DL	Deep Learning
DNA	Deoxyribonucleic Acid
FGN	Feature Generating Networks
GZSL	Generalized Zero-shot Learning
H	Harmonic mean
KKCs	Known known classes
KUCs	Known unknown classes
Lp	Least populated classes
ML	Machine Learning
Mp	Most populated classes
S	Average of seen class accuracies
SotA	State of the Art
UBZSL	Unconstrained BZSL classifier
UKCs	Unknown known classes
US	Average of unseen class accuracies
UUCs	Unknown unknown classes
ZSL	Zero-shot Learning

ABSTRACT

Building machine learning algorithms to recognize objects in real-world tasks is a very challenging problem. With increasing number of classes, it becomes very costly and impractical to collect samples for all classes to obtain an exhaustive data to train the model. This limited labeled data bottleneck prevails itself more profoundly over fine grained object classes where some of these classes may lack any labeled representatives in the training data. A robust algorithm in this realistic scenario will be required to classify samples from well-represented classes as well as to handle samples from unknown origin. In this thesis, we break down this difficult task into more manageable sub-problems and methodically explore novel solutions to address each component in a sequential order.

We begin with zero-shot learning (ZSL) scenario where classes that are lacking any labeled images in the training data, i.e., unseen classes, are assumed to have some semantic descriptions associated with them. The ZSL paradigm is motivated by analogy to humans' learning process. We human beings can recognize new categories by just knowing some semantic descriptions of them without even seeing any instances from these categories. We develop a novel hierarchical Bayesian classifier for ZSL task. The two-layer architecture of the model is specifically designed to exploit the implicit hierarchy present among classes, in particular evident in fine-grained datasets. In the proposed method, there are latent classes that define the class hierarchy in the image space and semantic information is used to build the Bayesian hierarchy around these meta-classes. Our Bayesian model imposes local priors on semantically similar classes that share the same meta-class to realize knowledge transfer. We finally derive posterior predictive distributions to reconcile information about local and global priors and then blend them with data likelihood for the final likelihood calculation. With its closed form solution, our two-layer hierarchical classifier proves to be fast in training and flexible to model both fine and coarse-grained datasets. In particular, for challenging fine-grained datasets the proposed model can leverage the large number of seen classes to its advantage for a better local prior estimation without sacrificing on seen class accuracy.

Side information plays a critical role in ZSL and ZSL models hold on a strong assumption that the side information is strongly correlated with image features. Our model uses side

information only to build hierarchy, thus, no explicit correlation between image features is assumed. This in turn leads the Bayesian model to be very resilient to various side information sources as long as they are discriminative enough to define class hierarchy.

When dealing with thousands of classes, it becomes very difficult to obtain semantic descriptions for fine grained classes. For example, in species classification where classes display very similar morphological traits, it is impractical if not impossible to derive characteristic visual attributes that can distinguish thousands of classes. Moreover, it would be unrealistic to assume that an exhaustive list of visual attributes characterizing all object classes, both seen and unseen, can be determined based only on seen classes. We propose DNA as a side information to overcome this obstacle in order to do fine grained zero-shot species classification. We demonstrate that 658 base pair long DNA barcodes can be sufficient to serve as a robust source of side information for newly compiled insect dataset with more than thousand classes. The experiments is further validated on well-known CUB dataset on which DNA attributes proves to be as competitive as word vectors. Our proposed Bayesian classifier delivers state of the art results on both datasets while using DNA as side information.

Traditional ZSL framework, however, is not quite suitable for scalable species identification and discovery. For example, insects are one of the largest groups of animal kingdom with estimated 5.5 million species yet only 20% of them is described. We extend the traditional ZSL into a more practical framework where no explicit side information is available for unseen classes. We transform our Bayesian model to utilize taxonomical hierarchy of species to perform insect identification at scale. Our approach is the first to combine two different data modalities, namely image and DNA information, to perform insect identification with more than thousand classes. Our algorithm not only classifies known species with impressive 97% accuracy but also identifies unknown species and classify them to their true genus with 81% accuracy.

Our approach has the ability to address some major societal issues in climate change such as changing insect distributions and measuring biodiversity across the world. We believe this work can pave the way for more precise and more importantly the scalable monitoring of biodiversity and can become instrumental in offering objective measures of the impacts of recent changes our planet has been going through.

1. INTRODUCTION

Deep learning (DL) has become one of the key post-millennial breakthroughs and is pushing the boundary of science in a pace and scale beyond what was considered feasible [1]–[6]. Convolutional Neural Networks (CNN) in particular have been the main influence paving this DL revolution. Although CNNs were introduced in 1990's [1], [2], it was after a decade that industry and academia took full advantage of CNNs thanks to the accumulated large-scale image database [4], [7] and enhanced computational power (GPUs).

Transcending the academia, Artificial Intelligence (AI) and Machine Learning (ML) now pervade every aspect of our modern life spanning from home/ personal assistance to drug development, financial security, biodiversity measurement and many more. Transitioning from controlled lab setting, more precisely "closed-set" setup, to real world problems poses new challenges for ML algorithms by introducing open space risk [8]. Unlike "closed-set" classification setup where training and test sets are assumed to share the same set of object classes, in open-set setting, possibility of unknown objects during inference is acknowledged. That is, there might be test instances that are not represented by any object categories in training data. This problem becomes more prevalent in large scale fine-grained object recognition task as natural images have a power-low property; that is, many object classes will not be represented in our training data [4], [9], [10].

1.1 Acknowledging the Unknown

Tackling the distribution difference between train and test data and accounting for novel categories have been investigated under the hood of lifelong learning [11], [12], domain adaptation [13], [14], zero-shot learning [15], [16], few-shot learning [17], [18] and open-set recognition/ classification [8], [19]–[21]. To elucidate the contributions and draw clear boundaries between these different approaches, [8], [21] adapted the famous quote of Donald Rumsfeld [22], "there are known knowns" and suggested that recognition task should consider four basic class categories:

- *known knowns classes* (KKCs): the classes with distinctly labeled positive training samples (also serving as negative samples for other KKCs), and even have the corresponding auxiliary information like semantic attributes, etc.
- *known unknowns classes* (KUCs): labeled negative samples, not necessarily grouped into meaningful classes, such as background classes [23]
- *unknown known classes* (UKCs): classes with no available samples in training, but available auxiliary information , e.g., attributes [15], word vectors [24], text descriptors [25], DNA [26], etc.) of them during training
- *unknown unknown classes* (UUCs): classes without any samples and auxiliary information during training

Traditional multi-class classification only considers KKCs and ignores the rest. Once KUCs are included in the task then the model becomes a detector that is trained with unclassified negatives, or so called "other class" [19]. Anomaly detection [27] can be seen as a use case for this application. One-class classifiers [28]–[31] are first models attempted to capture the outliers by modeling the training data distribution so that open space can be set apart from training distribution in the feature space. However, this set-up is not optimal for addressing open space risk. First, ignoring the discriminative information from different KKCs in training data by treating them as a single class causes huge information loss. Second, even if KKCs are modeled individually by one-vs-all fashion, the thresholding step in the inference is not robust for detecting novel classes [32]. This in turn necessitates a more systematic way to model open space risk in the case of multi-class classification under the Open-set Recognition (OSR).

1.2 Open-set Recognition

In OSR setting, models are required not only to correctly classify samples from KKCs but also to efficaciously handle the UUCs. There have been several studies investigating OSR under different applications and frameworks. [33] introduced a k-NN based model to perform multi-class authentication with a rejection option to exclude a new face that has not

been enrolled in the training gallery. The authors in [34] proposed a hierarchical Support Vector based model to handle a more generic multi-class classification with a rejection option. Work in [35], [36], on the other hand, leveraged Bayesian non-exhaustive learning to perform bacteria detection in open-set setting. OSR framework has also been used in interesting applications such as tattoos detection [37], automated genre identification [38], authorship attribution [39], to name a few. Despite its implicit presence in the literature, open space risk is first formalized in OSR framework by Schierer et al. in their seminal work [8]. The authors propose a SVM based "1-vs-set machine" model to address the open space risk as a constrained minimization problem. Following this work, OSR and related work have attracted great attention.

Although OSR has a more realistic framework, the presence of UUCs hinders the progress in the domain. Most of the models developed are generally application-based and the state-of-the-art models are still far from the deployment, in particular for fine-grained object recognition task with large number of classes. Fine-grained image classification in open-set setting with large number of classes is one of the challenges we address in this thesis.

1.3 Zero-shot Learning

Inspired by the human ability, distinguishing new categories purely based on high-level descriptions, a new framework called Zero-shot Learning (ZSL)[15], [16] is promoted to tackle the fine-grained image classification with limited labeled data. Instead of dealing with completely unknown classes (UUCs), ZSL leverages semantic information for unknown classes and makes the problem at hand more manageable. ZSL considers the train and test classes, i.e seen (KKCs) and unseen classes (UKCs), as two disjoint sets and aims to classify instances from unseen classes utilizing the data from seen classes and high-level semantic descriptions from all classes. Information propagation between seen and unseen classes is achieved through these semantic descriptions, also called side or auxiliary information. Traditional ZSL, however, works under very restrictive assumption that test samples come only from unseen classes. This unrealistic conjecture is later relaxed by extending the ZSL into generalized zero-shot learning to allow test samples from both seen and unseen classes.

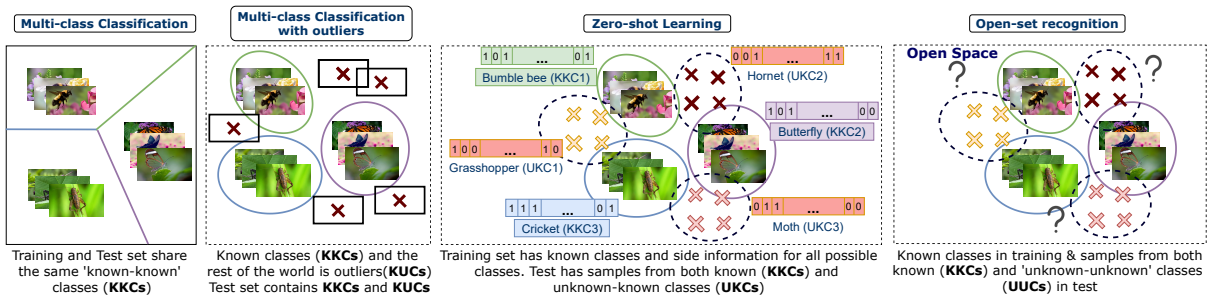


Figure 1.1. Classification problem from different openness perspective

From now on, throughout this thesis ZSL will refer to the generalized zero-shot learning framework. The Figure 1.1 depicts how the samples from unknown origin is handled in different frameworks.

An intuitive, also the most common, solution to the ZSL is to learn an embedding from image feature space to semantic space then to use various metrics to find the closest semantic description to the test instances in the semantic space [16], [40]–[42]. In order to enable the usage of multiple side information sources, few studies [43]–[47] proposed to learn a mapping from image and semantic space into a common intermediate space. A recurrent problem in these approaches is the data variance shrinkage due to embedding from high dimensional image feature space into smaller embedding space. This, in turn, leads to a *hubness* problem once nearest neighbor search is performed for the inference. That is, some classes become universal neighbors, or hubs, that appear in the query of many test samples. The hubness problem in ZSL is studied by [48], [49] and the authors showed that if the direction of embedding is changed from semantic to feature space, the problem cannot get worse and models can fully take advantage of the image space's discriminatory power. That being said, using nearest neighbor search for the inference still poses a bias towards seen classes and renders sub-optimal performance. Developing an algorithm to bypass the nearest neighbor search and to overcome the hubness problem in ZSL is another challenge that is addressed in this thesis.

Recently the course of mainstream research in ZSL has taken a turn towards generative models. The work in [50]–[56] leverage the conditional Generative Adversarial Networks (GANs) to synthesize image features for unseen classes from their semantic descriptions and

then transform the challenging ZSL into closed-set classification problem. Although GAN based models render significant performance boost in ZSL task, they have an inherent problems of training instability and mode collapse. Thus, Variational AutoEncoders (VAEs) [57]–[60] also gained popularity for learning a cross-model embedding for downstream task of image feature synthesis. Feature generating networks (FGNs) currently establish the state of the art in ZSL, nonetheless, all these models operate well only under the assumption that the side information and image features are strongly correlated.

Side information is an essential component of Zero-shot learning. It enables classes to share information among them and facilitates knowledge transfer from seen classes to unseen ones. Attributes describing visual properties of object classes are the first form of side information proposed in ZSL [15]. Despite its popularity, deriving visual attributes for large scale fine-grained datasets requires huge human labor for its manual annotation process, thus, is very costly. Therefore, there has been several studies [25], [40], [45], [61]–[63] to explore more practical sources of side information that require minimal annotation effort such as word vectors [24], text descriptions, and WordNet hierarchy [64]. Nonetheless, when dealing with fine-grained species classification with more than thousand classes, deriving subtle characteristic attributes for species using these auxiliary information sources becomes infeasible. Developing a robust and scalable solution for fine-grained zero-shot species classification is another challenge we addressed in this thesis.

1.4 Contributions

The focus of this thesis is to perform fine-grained object recognition with limited labeled data. The format of this thesis follows an article-based dissertation and the context of the thesis is systematically crafted around three research questions we seek to answer: (1) Starting with a more manageable ZSL task, can we develop a robust model that can efficaciously cope with fine-grained datasets containing a large number of classes? (2) How can we perform fine-grained zero-shot classification if all available side information sources are infeasible for the task at hand? (3) How can Zero-shot classification be extended to a more practical real-world species identification and discovery with more than 1000 classes? Our

contributions towards providing solutions for these challenges are described in the following paragraphs.

- We develop a hierarchical Bayesian model for zero-shot learning task that leverages the implicit deeper hierarchy present among fine-grained object classes. Our two-layer Normal-Normal-Inverse-Wishart generative model defines meta/ surrogate classes by forming local priors around semantically similar seen classes. These meta-classes enable information propagation from seen to unseen classes. Blending the local prior with global prior, we derive posterior predictive distribution (PPD) for each meta-class and thanks to the conjugacy, the PPD can be analytically derived in closed form as Student-t distribution. Hyperparameters from global prior render great flexibility to model datasets with different granularity and closed form PPD provides an edge over training time for the proposed method. To our best knowledge, our approach is the first fully Bayesian model in ZSL domain and bypass the hubness problem that most of the embedding based ZSL methods face. Prior to the dominance of feature generating methods, Bayesian model set the state of the art performance on six different ZSL benchmark datasets. As will be discussed in the next chapters, we also demonstrate that the Bayesian model is more robust against different sources of side information when compared against current state-of-the-art FGNs. This work was published in [65] and is described in Chapter 2.

- As our second contribution, we first demonstrate how our hierarchical Bayesian classifier utilizes the inter-class similarity between fine-grained object classes and the large number of training classes to its advantage for a robust local prior estimation and ultimately better unseen class performance without sacrificing much on seen class performance. The two-layer architecture of the model provides a great flexibility to employ various side information sources as long as they are expressive enough to exploit the class similarity. In fact, current state-of-the-art ZSL methods work under a strong assumption that side information is strongly correlated with image features and once the alternative side information sources that violate this assumption are used, their performance significantly suffers. We also show that the Bayesian classifier delivers

robust performance even the side information does not exhibit any explicit correlation with image features.

Second, we challenge the adequacy of traditional side information sources for fine-grained large scale species classification and then propose DNA as side information for the first time in ZSL to tackle this problem. We introduce a newly compiled fine-grained INSECT dataset that contains 21K image-DNA pairs with more than 1200 species to serve a new ZSL benchmark and validate our approach on. To further verify the effectiveness of the proposed approach and showcase that the DNA can be as competitive as word vectors, we extend our experiment to Caltech CUB-Bird dataset [66]. Our Bayesian model renders superior performance on both datasets while using DNA as side information compared against the state of the art ZSL techniques including FGNs. Moreover, Bayesian model also renders the best performance on CUB data if visual attributes are not used as side information.

- Our final contribution is to extend our model to perform species identification and discovery in a more realistic open-set setting. The aim of this study is to provide a scalable machine learning solution for the gigantic task of biodiversity measurement to help scientists better assess the changes our planet is going through. Since ZSL requires some sort of side information for both seen and unseen classes, it is not practical, if not impossible, to gather this information for tens of thousands of species. Our Bayesian model leverages the class taxonomy of species, in our case insects which represent the majority of the biodiversity on earth with more than estimated 5.5 million species, to not only classify species from KKCs but also identify the species from UUCs at the lowest level of abstraction possible. Our method is the first attempt to combine DNA and image information for species classification in open-set framework with large number of classes. Our experimental validation is performed on the slightly modified INSECT dataset that contains 32K image-DNA pairs from 1,040 species. The proposed approach yields an impressive 97% accuracy for classifying known species while achieving the accuracy of 81% in identifying the true genera of insect instances from unknown species. Chapter 4 of this thesis presents the details of this project.

The abstract summarizing this work is accepted as an oral talk to 30th International Congress for Conservation Biology and the full research article is under journal review.

2. BAYESIAN ZERO-SHOT LEARNING

In this chapter, we propose a Bayesian approach to zero-shot learning that introduces the notion of meta-classes and implements a Bayesian hierarchy around these classes to effectively blend data likelihood with local and global priors. Local priors driven by data from seen classes, i.e., classes available at training time, become instrumental in recovering unseen classes, i.e., classes that are missing at training time, in a generalized ZSL setting. Hyperparameters of the Bayesian model offer a convenient way to optimize the trade-off between seen and unseen class accuracy. We conduct experiments on seven benchmark datasets, including a large scale ImageNet and show that our model produces promising results in the challenging generalized ZSL (GZSL) setting. This chapter corresponds to our following work [65],

S. Badirli, Z. Akata, M. Dundar. Bayesian Zero-shot Learning. In *European Conference on Computer Vision (ECCV) Workshops*, 2020.

2.1 Introduction

Natural images exhibit a power-law property; hence, in a randomly sampled training set, no training examples are expected to be available for most of the object categories [4], [9], [10]. This restriction becomes more evident in a fine-grained object recognition task. Zero-shot learning, which considers training and test classes, i.e., seen and unseen classes, as two disjoint sets, was introduced to mitigate this limitation [15], [16]. The two groups of classes are linked through a shared set of attributes that characterize high level semantic descriptions of all classes. During the training phase, a mapping between examples of seen classes and their corresponding class-based attributes is learned. This mapping is later used to identify examples of unseen classes during the test phase.

The standard ZSL setting restricts test time search space to only unseen classes. This somewhat unrealistic stipulation was later relaxed in the generalized ZSL setting to include all classes during the test phase [67]. In GZSL, side information, i.e., attributes, are as important as the perceptual representation of images. Attribute vectors are either manually annotated [15], [68] or derived from free-form text using word embedding [24], [40], [69]. Early

line of work in ZSL [15] assumes attribute independence and uses probabilistic classifiers to assign images to test classes.

In this paper, we tackle ZSL by introducing a two layer Bayesian hierarchy manifesting over both seen and unseen classes. Our approach is designed to leverage the implicit hierarchy present among classes, especially evident in fine grained data sets [66], [70], [71]. Unlike earlier approaches, which seek to optimize an embedding between image and semantic spaces, the proposed method assumes that there are latent classes that define the class hierarchy in image space and uses semantic information to build the Bayesian hierarchy around these meta-classes. Our model uses two types of Bayesian priors: global and local. As the name suggests, global priors are shared across all classes, whereas local priors are only shared among semantically similar classes, which are identified based on the distances between attribute vectors in the Euclidean space. Unlike standard Bayesian models where the posterior predictive distribution establishes a compromise between prior and likelihood, our approach utilizes posterior predictive distributions to reconcile information about local and global priors as well as the likelihood to more effectively accommodate the class hierarchy. In this framework, unseen classes are represented by their corresponding meta classes (see Figure 2.1), and test samples are classified based on posterior predictive likelihoods computed for both seen and unseen classes. Our approach achieves significant improvements on both seen and unseen class accuracies to achieve the best results on a variety of benchmark datasets among the currently published state of the art methods.

Our contributions are as follows. (1) We propose a hierarchical Bayesian model based on the intuition that actual classes originate from their corresponding local priors, each defined by a meta-class of its own. (2) We derive the posterior predictive distribution (PPD) for a two-layer Gaussian mixture model to effectively blend local and global priors with data likelihood. These PPDs are used to implement a maximum-likelihood classifier, which represents seen classes by their own PPDs and unseen classes by meta-class PPDs. (3) Across seven datasets with varying granularity and sizes, in particular on the large-scale ImageNet dataset, we show that the proposed model is highly competitive against existing inductive techniques in the GZSL setting.

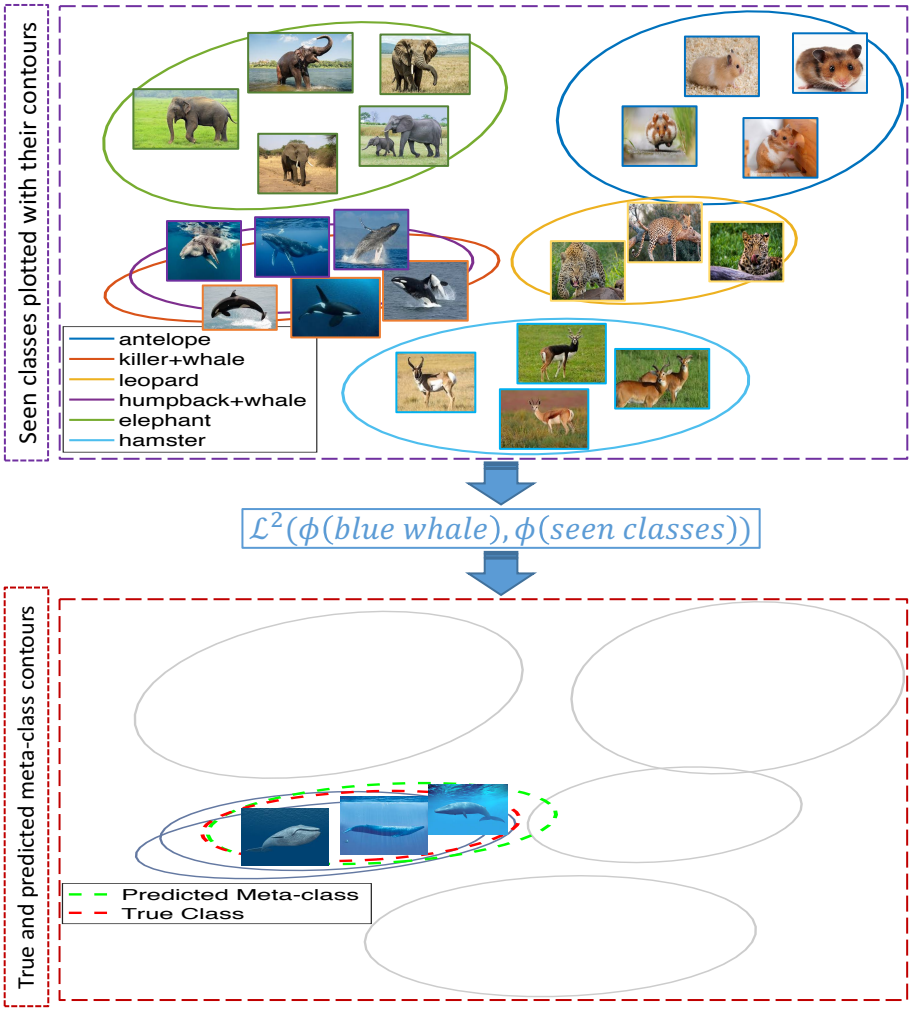


Figure 2.1. Meta-classes illustrated in 2D PCA space (reduced from 2048). Only a small subset of seen classes is shown. Contours are derived from class covariance matrices and placed at two standard deviations away from class means. Meta class for *blue whale* (unseen) predicted based on *killer* and *humpback* whales (seen).

2.2 Related Work

In this section, we discuss the prior work on zero-shot learning and hierarchical generative models related to ours.

Zero-Shot Learning. There has been an increasing interest in classifying fine grained and large-scale image datasets [4], [66], [70], [71]. This, in turn, led to a surge of interest in ZSL as labeling them is extremely costly. In their seminal paper [15], authors tackle

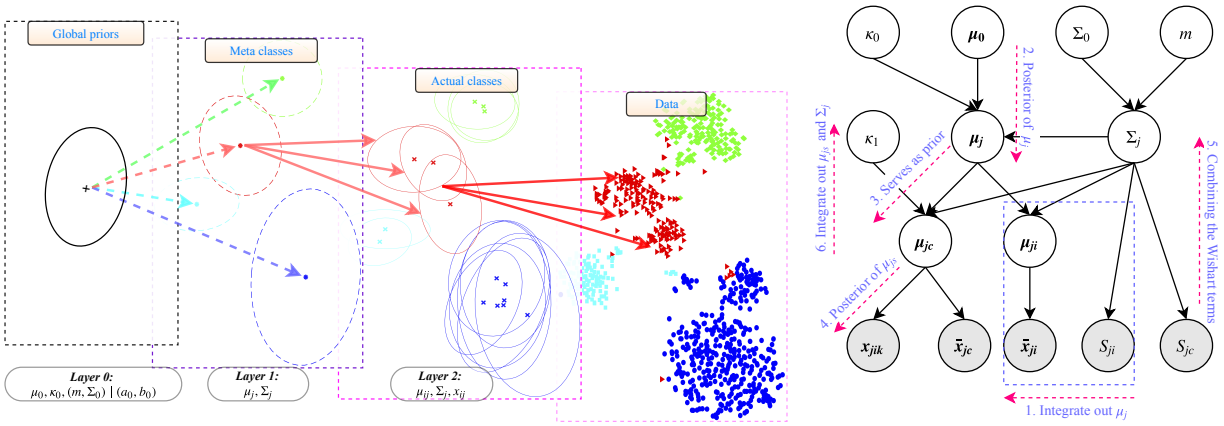
ZSL by implementing a probabilistic classifier for each attribute and then classifying test cases by aggregating attribute probabilities for each class. This approach treats attributes as independent, which is a fairly strong assumption for most real-world data sets. This work was followed by a large body of work that seeks to optimize a mapping from image space, i.e., feature vectors, onto semantic space, i.e., attribute vectors. This line of work can be categorized into two according to whether the mapping is bi-linear [40]–[42], [45], [72] or non-linear [61], [69]. Related to ours, [43], [46], [47] first maps image and semantic space into an intermediate space and represents unseen classes as a mixture of seen classes. Besides these mainline ZSL studies, a recent study evaluates an extended version of a few-shot learning algorithm for ZSL [44]. This approach learns a deep metric to query images with few shot samples. Extension to ZSL is achieved by replacing few-shot samples with one-shot class attribute vectors.

Generative models for ZSL. Although most of the early work focused on discriminative models, there are a few studies that use generative models to tackle ZSL [73], [74]. The study in [73] uses Normal distributions to model both image features and semantic vectors and learns a multimodal mapping between two spaces. This mapping is optimized by minimizing a similarity based cross domain loss function. In a similar fashion the study in [74] utilizes a regression model to optimize a mapping between class attributes and parameters of class conditional distributions. A comprehensive review of these techniques and their performance on several benchmark data sets can be found in [67].

There are also quite a few techniques that tackle ZSL in a transductive setting. Experiments in [58], [75] demonstrate that unlabeled data from unseen classes as well as training data augmented by generative adversarial nets/ variational autoencoders can notably boost the classification accuracy. We believe that this line of work should be treated under a different category as a direct comparison with current ZSL techniques is not possible since similar data augmentation techniques could have most certainly benefited these techniques.

Bayesian models. In this paper, we offer a hierarchical Bayesian perspective on ZSL as a promising alternative to earlier approaches. Although hierarchical Bayesian mixture models have been previously explored for a variety of clustering problems [76]–[79], their extension to ZSL comes with two distinct features that could help the proposed model

prevail over the large body of early work in ZSL. First, as a Bayesian model, ours offers a systematic approach to sharing information between seen and unseen classes as well as within each group through the utilization of local and global priors. Global priors are defined by hyperparameters, whereas local priors are determined by the parameters of the meta classes, which are estimated from corresponding seen classes. Second, as a hierarchical model, it can better accommodate data sets with different levels of class abstractions, i.e., fine-grained vs. coarse-grained data sets, which is particularly appealing for large-scale classification. A hierarchical Bayesian model was previously studied in a one-shot learning setting [18]. Our proposed approach differs from this model in two essential aspects. First, unlike our proposed approach, no semantic information was used when establishing the Bayesian hierarchy in [18], and class discovery was performed in a fully unsupervised fashion. Second, our approach introduces the notion of local prior, which becomes highly instrumental in defining meta-classes and modeling dispersion of classes.



(a) Our Bayesian Zero-Shot Learning Model (BZSL) (b) Graphical model

Figure 2.2. Two-layer Generative Model for Bayesian zero-shot learning (BZSL). (a) Latent (Meta) classes shown by dashed lines in layer 1 are generated from Normal distribution ($N(\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}})$). Hyperparameters in layer 0 are the priors on the mean and covariance of these Gaussians. The sufficient statistics of actual classes in layer 2 are obtained from layer 1. Finally, data samples are drawn from Gaussian distribution with the mean, $\boldsymbol{\mu}_{\mathrm{ji}}$ and covariance, Σ_{j}. (b) Conditional hierarchical Gaussian data generation (Likelihood) model and derivation of marginal likelihood.

Our work. Unlike the vast majority of early work, which seeks to optimize a mapping between image features and attribute vectors, our approach readily models class distributions in the feature space by exploiting both local and global priors defined over the parameters of these distributions. Local priors are defined by meta-classes. In the proposed approach, attribute vectors only come into play when determining meta-class memberships of actual classes. Classes with similar attribute vectors are pooled together to derive local priors.

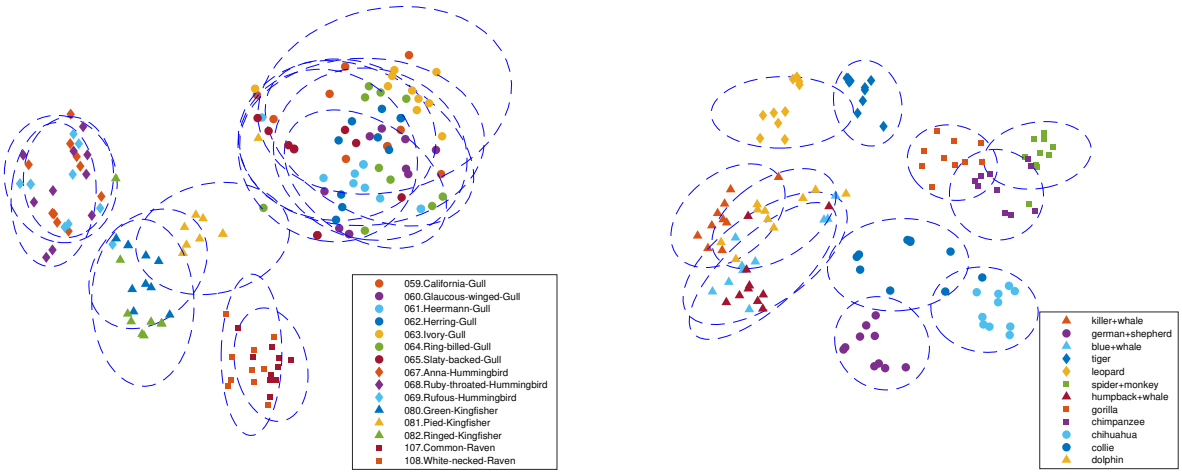
2.3 Hierarchical Bayesian Classifier

Bayesian classification places a shared prior over the parameters of class distributions, which are assumed to be generated independently conditioned on the prior. Imposing the same prior across all classes creates dependencies among them, enabling information propagation and regularization at the same time during model inference. However, in real-world applications the classes are often not generated independently, indeed for a large number of classes different levels of abstraction is expected. On the other hand, availability of semantic side information suggests that there is a deeper level of hierarchy among existing classes than a single global Bayesian prior can explain.

Images from semantically similar classes are embedded close to each other due to their shared latent parameters (See figure 2.3). When such similarities are not accounted for in the classification model, sample estimates of class parameters derived based on independence assumption among classes become nullified. In other words, knowing the parameters of the global prior may not be sufficient for achieving independence as latent parameters define deeper level hierarchical relationships among classes. Our model resolves this problem by introducing a layer of meta-classes between global prior and actual classes, paving the way for independence and enabling information sharing and propagation across classes.

2.3.1 Generative Model

Our approach to ZSL employs class similarities by a two-layer generative model. As shown in Figure 2.2, our model identifies meta-classes that determine groupings among classes. These meta-classes play a key role by acting as a local prior for individual classes,



(a) CUB (b) AWA

Figure 2.3. t-SNE visualization from AWA and CUB datasets shows that classes similar in the attribute space indeed cluster closer in the feature space as well. For example, white-necked raven clusters next to common raven instead of Gull, Hummingbird and Kingfisher species. The same phenomenon also appears in coarse grained datasets (AWA) as different kinds of dogs, monkeys, cetaceans, and carnivorous cats cluster together with their other kinds. Nevertheless, note that groups are not as intermingled as in fine grained CUB dataset.

i.e. both seen and unseen classes that belong to the same meta-class inheriting the same local prior. In our framework, the data points with the same local prior that do not belong to any of the seen classes can be considered from unseen classes. If the class groupings can be arranged such that there is only one unseen class associated with each local prior then unseen classes can be uniquely identified. Associating each unseen class with a different local prior forms the basis of our approach. Our generative model is designed as follows:

$$
\begin{aligned}
\boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}} &\sim N(\boldsymbol{\mu}_{\mathrm{ji}}, \Sigma_{\mathrm{j}}), \\
\boldsymbol{\mu}_{\mathrm{ji}} &\sim N(\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}\kappa_1^{-1}), \\
\boldsymbol{\mu}_{\mathrm{j}} &\sim N(\boldsymbol{\mu}_{\mathbf{0}}, \Sigma_{\mathrm{j}}\kappa_0^{-1}), \\
\Sigma_{\mathrm{j}} &\sim W^{-1}(\Sigma_0, m)
\end{aligned}
\tag{2.1}
$$

with the meta-class index j, the actual class index i, the image index k. We assume that images $\boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}}$ come from a Gaussian with mean $\boldsymbol{\mu}_{\mathrm{ji}}$ and covariance matrix Σ_{j}. They are generated independently conditioned not only on the global prior but also on their corresponding meta-class.

Each meta-class is characterized by the parameters $\boldsymbol{\mu}_{\mathrm{j}}$ and Σ_{j}. $\boldsymbol{\mu_0}$ is the mean of the Gaussian prior defined over the mean vectors of meta-classes, κ_0 is a scaling constant that adjusts the dispersion of the centers of meta classes around $\boldsymbol{\mu_0}$. A smaller value for κ_0 suggests that class centers are expected to be farther apart from each other whereas a larger value suggests they are expected to be closer to each other. On the other hand, Σ_0 and m dictate the expected shape of the class distributions, as under the inverse Wishart distribution assumption the expected covariance is $E(\Sigma|\Sigma_0, m) = \frac{\Sigma_0}{m-D-1}$, where D is the dimension of image feature space. The minimum feasible value of m is equal to $D+2$, and the larger the m is the less individual covariance matrices will deviate from the expected shape.

On the other hand, κ_1 is a scaling constant that adjusts the dispersion of the actual class means around their corresponding meta-class means. A larger κ_1 leads to smaller variations in class means compared to the mean of their corresponding meta classes, suggesting a fine-grained relationship among classes sharing the same meta-class. On the other hand, a smaller κ_1 dictates coarse-grained relationships among classes sharing the same meta-class. In this model, classes with the same meta-class also share the same covariance matrix Σ_{j} to preserve conjugacy of the model.

To classify test examples, we need the posterior predictive distributions (PPD) of seen and unseen classes which we will explain next. More details about the derivation are provided in the Appendix A.1.

2.3.2 Posterior Predictive Distribution

In our model, the posterior predictive distribution (PPD) incorporates three sources of information: the data likelihood that arises from the current class, the local prior that results from other classes sharing the same meta class as the current class, and global prior defined

in terms of hyperparameters. The derivation in six steps are outlined in Figure 2.2b and Algorithm[1] 1 describes a pseudo code on deriving PPD for both seen and unseen classes. Class sufficient statistics are summarized by $\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}$ and n_{ji} which represent sample mean, scatter matrix and size of class i of meta-class j, respectively. The notations $\omega_{\mathrm{j}c}$ and ω_{j} used in the Algorithm 1 represents the current seen class and unseen class, whose PPD is being derived.

In step 1, we establish the link between class sample mean $\bar{\boldsymbol{x}}_{\mathrm{ji}}$ and its corresponding meta-class mean $\boldsymbol{\mu}_{\mathrm{j}}$ by marginalizing out the intermediate class mean $\boldsymbol{\mu}_{\mathrm{ji}}$. As all of these are Gaussian, this marginalization yields a Gaussian:

$$P(\bar{\boldsymbol{x}}_{\mathrm{ji}}|\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1) \quad = \quad N(\bar{\boldsymbol{x}}_{\mathrm{ji}}|\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}(\frac{1}{n_{\mathrm{ji}}} + \frac{1}{\kappa_1})) \tag{2.2}$$

In step 2, we use Bayes rule to obtain the posterior distribution of the meta-class mean vector $\boldsymbol{\mu}_{\mathrm{j}}$:

$$\begin{gathered} P(\boldsymbol{\mu}_{\mathrm{j}}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) = N(\boldsymbol{\mu}_{\mathrm{j}}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\kappa}_{\mathrm{j}}^{-1}\Sigma_{\mathrm{j}}) \\ \bar{\boldsymbol{\mu}}_{\mathrm{j}} = \frac{\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}}+\kappa_1)}\bar{\boldsymbol{x}}_{\mathrm{ji}} + \kappa_0\boldsymbol{\mu_0}}{\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}}+\kappa_1)} + \kappa_0}, \quad \bar{\kappa}_{\mathrm{j}} = (\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}}+\kappa_1)} + \kappa_0) \end{gathered} \tag{2.3}$$

where t_{i} is the meta-class indicator for class i. Note that the mean $\bar{\boldsymbol{\mu}}_{\mathrm{j}}$ is the weighted average of the prior mean and class means share the same meta-class. In step 3, we obtain the local prior for class mean vector $\boldsymbol{\mu}_{\mathrm{j}c}$ by propagating the information from other classes sharing the same meta-class as the current class c. This is achieved by integrating out the meta-class mean vector $\boldsymbol{\mu}_{\mathrm{j}}$.

$$P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) = N(\boldsymbol{\mu}_{\mathrm{j}c}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \Sigma_{\mathrm{j}}(\bar{\kappa}_{\mathrm{j}}^{-1} + \kappa_1^{-1})) \tag{2.4}$$

[1]↑The code is publicly available at https://github.com/sbadirli/Bayesian-ZSL

Algorithm 1 Modeling seen and unseen classes in BZSL

Input: Training data, $\phi(seen)$, $\phi(unseen)$

Output: PPD parameters for each seen class $(\bar{\boldsymbol{\mu}}_{\mathrm{j}\boldsymbol{c}}, \bar{v}_{\mathrm{j}c}, \bar{\Sigma}_{\mathrm{j}c})$ and unseen class $(\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{v}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}})$

1: Set hyper-parameters: $\kappa_0, \kappa_1, m, s, K$
2: Compute $\boldsymbol{\mu_0}$ (mean of class means) and Σ_0 (mean of class covariances scaled by s)
3: **for** each seen class $\omega_{\mathrm{j}c}$ **do** ▷ Images available
4: Calculate current class params: $\bar{\boldsymbol{x}}_{\mathrm{j}\boldsymbol{c}}, n_{\mathrm{j}c}, S_{\mathrm{j}c}$
5: Find K most similar seen classes:
6: $\mathcal{L}^2(\phi(\omega_{\mathrm{j}c}), \phi(seen))$
7: **for** each selected seen class $\omega_{\mathrm{j}i}$ **do**
8: Calculate class params: $\bar{\boldsymbol{x}}_{\mathrm{j}i}, n_{\mathrm{j}i}, S_{\mathrm{j}i}$
9: **end for**
10: Calculate intermediate terms: $\tilde{\kappa}_{\mathrm{j}}, \bar{\boldsymbol{\mu}}_{\mathrm{j}}, S_{\boldsymbol{\mu}}$ (Eq 2.5,2.3,2.6)
11: Calculate PPD parameters by combining *local prior*
12: and *data driven likelihood*: $\bar{\boldsymbol{\mu}}_{\mathrm{j}\boldsymbol{c}}, \bar{v}_{\mathrm{j}c}, \bar{\Sigma}_{\mathrm{j}c}$ (Eq 2.7)
13: **end for**
14: **for** each unseen class ω_{j} **do** ▷ No image available
15: Find K most similar seen classes:
16: $\mathcal{L}^2(\phi(\omega_{\mathrm{j}}), \phi(seen))$
17: **for** each selected seen class $\omega_{\mathrm{j}i}$ **do**
18: Calculate class params: $\bar{\boldsymbol{x}}_{\mathrm{j}i}, n_{\mathrm{j}i}, S_{\mathrm{j}i}$
19: **end for**
20: Calculate intermediate terms: $\tilde{\kappa}_{\mathrm{j}}, S_{\boldsymbol{\mu}}$ (Eq 2.5, 2.6)
21: Calculate PPD parameters using only *local*
22: *prior*: $\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{v}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}}$ (Eq 2.3, 2.7)
23: **end for**

In step 4, we derive the posterior of the current class mean vector $\boldsymbol{\mu}_{\mathrm{j}\boldsymbol{c}}$ by combining current class sample mean $\bar{\boldsymbol{x}}_{\mathrm{j}\boldsymbol{c}}$ from step 1 and the local prior from step 3.

$$P(\boldsymbol{\mu}_{\mathrm{j}\boldsymbol{c}}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{j}i}\}_{t_i=\mathrm{j}}, \bar{\boldsymbol{x}}_{\mathrm{j}\boldsymbol{c}}) = N(\boldsymbol{\mu}_{\mathrm{j}\boldsymbol{c}}|\frac{n_{\mathrm{j}c}\bar{\boldsymbol{x}}_{\mathrm{j}\boldsymbol{c}} + \tilde{\kappa}_{\mathrm{j}}\bar{\boldsymbol{\mu}}_{\mathrm{j}}}{n_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}}}, \Sigma_{\mathrm{j}}(\tilde{\kappa}_{\mathrm{j}}^{-1} + n_{\mathrm{j}c}^{-1}))$$

$$\tilde{\kappa}_{\mathrm{j}} = \frac{(\sum_{\mathrm{i}:t_i=\mathrm{j}} \frac{n_{\mathrm{j}i}\kappa_1}{(n_{\mathrm{j}i}+\kappa_1)} + \kappa_0)\kappa_1}{\sum_{\mathrm{i}:t_i=\mathrm{j}} \frac{n_{\mathrm{j}i}\kappa_1}{(n_{\mathrm{j}i}+\kappa_1)} + \kappa_0 + \kappa_1} \tag{2.5}$$

In step 5, we derive the posterior distribution of the covariance matrix Σ_{j} by combining the local prior of the covariance matrix $P(\Sigma_{\mathrm{j}}|\Sigma_0, m)$ with the distribution of the scatter matrices of the classes associated with meta-class j S_{ji} and current class $S_{\mathrm{j}c}$:

$$P(\Sigma_{\mathrm{j}}|\{S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, S_{\mathrm{j}c}) = IW(\Sigma_{\mathrm{j}}|\bar{S}_c, m + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}}(n_{\mathrm{ji}} - 1) + n_{\mathrm{j}c})$$
$$\bar{S}_c = \Sigma_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S_{\mathrm{ji}} + S_{\mathrm{j}c} + S_\mu, \quad S_\mu = \frac{n_{\mathrm{j}c}\tilde{\kappa}_{\mathrm{j}}}{\tilde{\kappa}_{\mathrm{j}} + n_{\mathrm{j}c}}(\bar{\boldsymbol{x}}_{\mathrm{j}\boldsymbol{c}} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})(\bar{\boldsymbol{x}}_{\mathrm{j}\boldsymbol{c}} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})^T \tag{2.6}$$

In step 6, we derive the posterior predictive distribution by integrating out meta-class mean vector $\boldsymbol{\mu}_{\mathrm{j}}$ and covariance Σ_{j} in the form of a Student-t distribution as follows.

$$P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \bar{\boldsymbol{x}}_{\mathrm{j}c}, S_{\mathrm{j}c}, \boldsymbol{\mu_0}, \kappa_0, \kappa_1) = T(\boldsymbol{x}|\bar{\boldsymbol{\mu}}_{\mathrm{j}\boldsymbol{c}}, \bar{\Sigma}_{\mathrm{j}c}, \bar{v}_{\mathrm{j}c})$$
$$\bar{\boldsymbol{\mu}}_{\mathrm{j}\boldsymbol{c}} = \frac{n_{\mathrm{j}c}\bar{\boldsymbol{x}}_{\mathrm{j}\boldsymbol{c}} + \tilde{\kappa}_{\mathrm{j}}\bar{\boldsymbol{\mu}}_{\mathrm{j}}}{n_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}}}, \quad \bar{v}_{\mathrm{j}c} = n_{\mathrm{j}c} + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}}(n_{\mathrm{ji}} - 1) + m - D + 1$$
$$\bar{\Sigma}_{\mathrm{j}c} = \frac{\Sigma_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S_{\mathrm{ji}} + S_{\mathrm{j}c} + S_\mu}{\frac{(n_{\mathrm{j}c}+\tilde{\kappa}_{\mathrm{j}})\bar{v}_{\mathrm{j}c}}{n_{\mathrm{j}c}+\tilde{\kappa}_{\mathrm{j}}+1}} \tag{2.7}$$

where, $\bar{\boldsymbol{\mu}}_{\mathrm{j}}$, $\tilde{\kappa}_{\mathrm{j}}$ and S_μ are defined as in Equation (2.3), (2.5) and (2.6) respectively. The index c in Equation (2.7) represents the current seen class, whose PPD is being derived. Top K most similar seen classes are identified as the ones with the smallest Euclidean distance to the current class in the attribute space. If the current class is a seen class, PPD takes the form in Equation (2.7). When it is an unseen class with no images available in training, the sample statistics of the current class in (2.7) drops and PPD becomes:

$$P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \boldsymbol{\mu_0}, \kappa_0, \kappa_1) = T(\boldsymbol{x}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}}, \bar{v}_{\mathrm{j}})$$
$$\bar{v}_{\mathrm{j}} = \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}}(n_{\mathrm{ji}} - 1) + m - D + 1, \quad \bar{\Sigma}_{\mathrm{j}} = \frac{(\Sigma_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S_{\mathrm{ji}})(\tilde{\kappa}_{\mathrm{j}} + 1)}{\tilde{\kappa}_{\mathrm{j}}\bar{v}_{\mathrm{j}}} \tag{2.8}$$

where $\bar{\boldsymbol{\mu}}_{\mathrm{j}}$ and $\tilde{\kappa}_{\mathrm{j}}$ are defined as in Equation (2.3) and (2.5), respectively. In this setting, a new image is labeled by evaluating PPDs for seen and unseen classes and assigning the image to the class that generates the maximum likelihood.

2.3.3 Meta-class Formation

Meta-class for each unseen class is formed by finding K most similar seen classes to the current unseen class using $\mathcal{L}^2$distance between the attribute vectors (ϕ) of that unseen class and of seen classes. In the case of tie, the least similar class among selected seen classes (K^{th}) is replaced by the next one until tie is broken. These define a local prior in the PPD of the unseen class. Meta-class formation for a seen class follows the same procedure. We use the $\mathcal{L}^2$ distance between the current seen class attribute and other seen class attributes to find K most similar classes. As we have access to seen class samples, the PPD of the seen class (Equation 2.7) uses class samples in addition to local and global priors from its meta-class. An illustration for the formation of the meta-class associated with an unseen class *blue whale*, from AWA dataset, is shown in Figure 2.1. $\phi(blue\ whale)$ is compared against $\phi(seen)$ in the semantic space, *humpback* and *killer whale* are identified as the two closest matches. Using *humpback* and *killer whale* class samples, the meta-class for *blue whale* is formed as a local prior in the PPD for *blue whale*.

2.4 Experiments

We evaluate the performance of the proposed approach on several benchmark data sets and compare the results with the current state of the art in ZSL. **Datasets & specifications.** Experiments are evaluated on ZSL datasets widely used for benchmarking. Among those, CUB [66], FLO [70] and SUN [71] are medium scale, fine-grained datasets. AWA1 [80] and AWA2 [81] and aPY [68], on the other hand, are coarse-grained datasets. Finally, we evaluate our model on ImageNet [4] with more than 14 million images and 21K classes. SUN, AWA1, AWA2, aPY and CUB datasets come with visual attributes whereas FLO uses sentences and ImageNet uses word embeddings as class vectors. We use the publicly available image embeddings of [81], i.e. 2048-dimensional top-layer pooling units of the 101-layered ResNet [3] as feature vectors. Additional information about each dataset including the number of images, number of attributes, and sizes of train, validation, and test class splits are present in Table 2.1.

Table 2.1. $|Y^{all}|$, $|Y^{s}|$, and $|Y^{u}|$ denote the number of classes in all, seen and unseen classes, respectively. To clarify the numbers in last 3 columns, we give an illustration on FLO dataset: FLO has total of 102 classes of which 62 are training, 20 are validation (both seen during training) and 20 are test classes (unseen during training).

Dataset	#imgs	Type	#att	$\lvert Y^{all}\rvert$	$\lvert Y^{s}\rvert$	$\lvert Y^{u}\rvert$
FLO	8,189	fine	102	102	62 + 20	20
SUN	14,340	fine	102	717	580 + 65	72
CUB	11,788	fine	312	200	100 + 50	50
AWA1	30,475	coarse	85	50	27 + 13	10
AWA2	37,322	coarse	85	50	27 + 13	10
aPY	15,339	coarse	64	32	15 + 5	12
ImageNet	14M	large	500	21K	1K	20K

For ImageNet following the benchmark in [81] we use all of the images from 1K classes, i.e. seen classes, for training so that we do not violate the zero-shot assumption as ResNet-101 [3] is trained on the same 1K classes from ImageNet. We evaluate the proposed technique in nine different configurations as proposed in [81], all of which differs according to how test class subsets are chosen.

Evaluation criteria. We use the same evaluation procedure employed in [81] as described below. The standard practice in ZSL literature is to evaluate classification performance by Top-1 accuracy. To avoid large classes dominating the overall accuracy, Top-1 accuracy is separately calculated for each class and the mean of individual class accuracies is used for evaluation. GZSL setting includes both seen and unseen classes in the test phase, hence the search space includes all the classes, i.e. $|Y^{all}|$. Hence, first seen and unseen class accuracies are separately computed and then their harmonic mean is used as the final score for evaluation. For ImageNet, the final score is the average Top-1 accuracy over the images of unseen classes (although the search space is still $|Y^{all}|$) as no images from seen classes are available during testing phase.

Implementation details. We implement two versions of our model: *unconstrained* (UBZSL) and *constrained* (CBZSL) Bayesian ZSL. For large data sets, e.g. ImageNet, our model in Eq.1 suffers from the large memory requirement due to the unconstrained structure

of the class covariance matrices. To alleviate this problem, we developed a scalable version of our model where the covariance matrices are constrained to have diagonal forms. The only difference between these two models is that constrained version uses an Inverse Gamma prior on the diagonal entries of the covariance matrix as opposed to an Inverse Wishart in the unconstrained version. With this revision the generative model in Eq.1 is updated as follows.

$$
\begin{aligned}
\boldsymbol{x}^{\boldsymbol{d}}_{\mathrm{ji}\boldsymbol{k}} &\sim N(\boldsymbol{\mu}^{\boldsymbol{d}}_{\mathrm{ji}}, \Sigma^{d}_{\mathrm{j}}) \\
\boldsymbol{\mu}^{\boldsymbol{d}}_{\mathrm{ji}} &\sim N(\boldsymbol{\mu}^{\boldsymbol{d}}_{\mathrm{j}}, \Sigma^{d}_{\mathrm{j}}\kappa_1^{-1}) \\
\boldsymbol{\mu}^{\boldsymbol{d}}_{\mathrm{j}} &\sim N(\boldsymbol{\mu}^{\boldsymbol{d}}_{\boldsymbol{0}}, \Sigma^{d}_{\mathrm{j}}\kappa_0^{-1}) \\
\Sigma^{d}_{\mathrm{j}} &\sim IG(a_0, b_0)
\end{aligned}
\tag{2.9}
$$

where the superscript d is added to refer to the d^{th} component of each parameter. The Inverse Wishart parameters m and Σ_0 are replaced with the scale (a_0) and shape (b_0) parameters of the Inverse Gamma distribution. The derivation of PPD for the constrained model is in the Appendix A.1.

The hyperparameters of the model are coarsely tuned to maximize the harmonic mean score on the validation set for all datasets but ImageNet. The training, test and validation set splits for these datasets are done according to [81] to maintain a fair comparison. As hyperparameter tuning for ImageNet can be computationally unmanageable and to demonstrate the robustness of the model we used the hyperparameters of the SUN dataset for ImageNet. For CBZSL we utilize all 2048 ResNet features whereas for UBZSL we applied PCA to reduce the dimensionality to 500.

Both UBZSL and CBZSL have four hyperparameters: $\kappa_0, \kappa_1, m, s, K$. Here, K is the selected number of classes most similar to the current class in the attribute space. To simplify the parameter tuning process, we set prior mean, $\boldsymbol{\mu}_0$, to the average of class means. We set Σ_0 to the average of class scatter matrices scaled by a constant s.

Table 2.2. Ablation study (in harmonic mean) on 6 datasets. In the UBZSL (V1) we discard Bayesian aspect and in UBZSL (V2) we impose similar dispersion for meta and actual classes

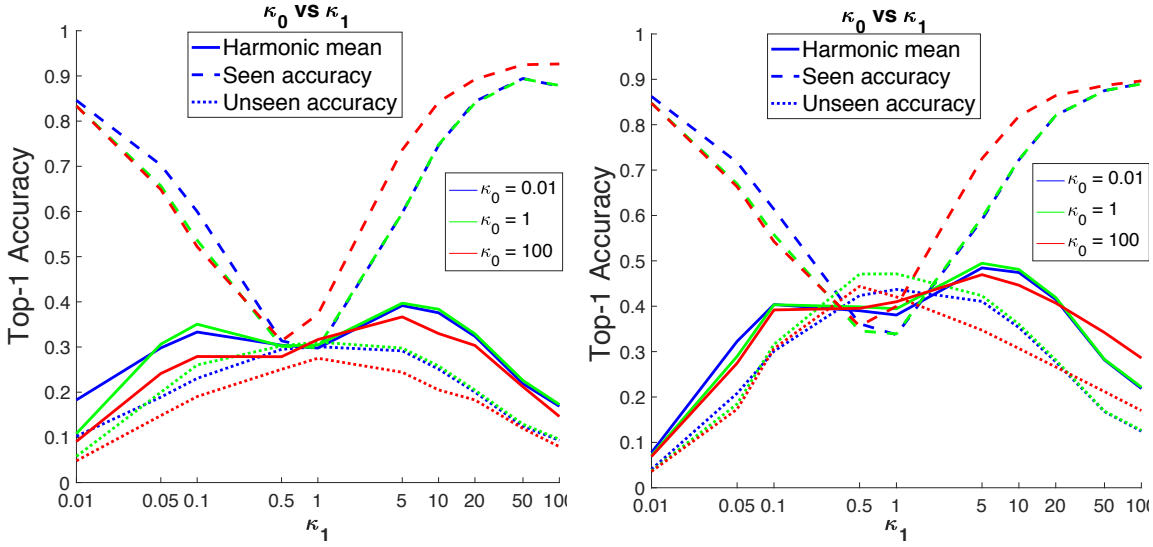
Method	SUN	CUB	AWA1	AWA2	aPY	FLO
UBZSL (V1)	32.5	24.9	21.1	29.0	10.0	20.5
UBZSL (V2)	3.0	18.3	38.0	40.3	9.5	34.1
UBZSL	32.8	37.5	49.6	49.7	35.4	40.4

2.4.1 Model Evaluation

In this section, we evaluate our model through an ablation study and investigate the tradeoff between seen and unseen class accuracies.

Model ablation. Our model formulates zero-shot learning in the framework of hierarchical Bayes. Towards this end, we validate the necessity of each component in our model by eliminating one component at a time and investigating the performance of the model with remaining components on several benchmark datasets. Our observations from Table 2.2 are as follows. (1) If we break the hierarchy by removing the meta-class layer, then actual classes are directly linked to the global prior and same PPD is assigned to all unseen classes. Thus, unseen classes can no longer be distinguished during test time. (2) If we discard the Bayesian aspect by eliminating the global and local priors, each seen class is fit a single Gaussian and each unseen class is fit a GMM with K components. We observe drastic drop in harmonic mean, almost cut in half, in all datasets but SUN (1^{st} row in Table 2.2: V1). In general, GMM works better on fine grained datasets than coarse grained ones as the distribution produced by a mixture of very similar classes can be better fit by GMM compared to a distribution produced by a mixture of relatively less similar classes. (3) Finally, if we impose similar dispersion for actual and meta classes (by improperly adjusting κ_0 and κ_1) with respect to the center of the data, harmonic mean again suffers significantly (2^{nd} row in Table 2.2: V2). In particular, results of the SUN dataset suffer the most. The impact of improper tuning of κ_1 is explained in the next section. Unlike V1, UBZSL V2 works better on coarse grained datasets (AWA1, AWA2) as class centers in these datasets are more separated

than fine grained ones. As a result, the adverse effects of setting $\kappa_1 << 1$ in experiments performed with these datasets seem to be less significant.



(a) FLO (b) AWA2

Figure 2.4. Variations in seen and unseen class accuracies and their harmonic means with respect to changes in κ_0 and κ_1. Seen and unseen class accuracies are highly sensitive to changes in κ_1 whereas minimal changes are observed w.r.t. changes in κ_0.

Effect of κ_0 and κ_1. In both of our models (*constrained* and *unconstrained*), different hyperparameter settings can be used to modify the operating point of the classifier to favor seen class accuracy over unseen one or vice versa. In this experiment we investigate the effect of κ_0 and κ_1 on seen and unseen class accuracies. Recall that κ_0 adjusts the dispersion of meta-class centers with respect to the center of the overall data and κ_1 adjusts the dispersion of actual class centers with respect to their corresponding meta class centers. The smaller these parameters are the higher the dispersion will be.

Figure 2.4 illustrates on FLO and AWA2 that unseen class accuracy is highest when κ_1 is close to 1 and drops significantly lower in both directions, i.e., for $\kappa_1 << 1$ and $\kappa_1 >> 1$. As expected, the opposite of this pattern is observed for seen class accuracy. Although both seen and unseen class accuracies are highly sensitive to the selection of κ_1, the changes are

[1]As [44] uses different set of attributes in their experiments, we rerun their algorithm with the attributes from [67] to maintain a fair comparison.

Table 2.3. GZSL results achieved by the proposed approach (CBZSL and UBZSL) along with results of several other techniques from the literature on SUN, CUB, FLO, AWA1, AWA2, aPY datasets. We measure per-class averages top-1 accuracy on seen classes (**tr**), unseen classes (**ts**) and their harmonic mean (**H**).

	SUN			CUB			AWA1			AWA2			aPY			FLO		
Method	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**	**ts**	**tr**	**H**
LATEM[61]	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	0.1	73.0	0.2	6.6	47.6	11.5
ALE[41]	21.8	33.1	26.3	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	4.6	73.7	8.7	13.3	61.6	21.9
DEVISE[40]	16.9	27.4	20.9	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	4.9	76.9	9.2	9.9	44.2	16.2
SJE[45]	14.7	30.5	19.8	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	3.7	55.7	6.9	13.9	47.6	21.5
ESZSL[42]	11.0	27.9	15.8	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	2.4	70.1	4.6	11.4	56.8	19.0
SYNC[47]	7.9	43.3	13.4	11.5	70.9	19.8	8.9	87.3	16.2	10.0	90.5	18.0	7.4	66.3	13.3	–	–	–
SAE[72]	8.8	18.0	11.8	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	0.4	80.9	0.9	–	–	–
GFZSL[74]	0.0	39.6	0.0	0.0	45.7	0.0	1.8	80.3	3.5	2.5	80.1	4.8	0.0	83.3	0.0	–	–	–
TCN[82]	31.2	37.3	**34.0**	52.6	52.0	**52.3**	49.4	76.5	**60.0**	61.2	65.8	**63.4**	24.1	64.0	35.1	–	–	–
DCN[83]	25.5	37.0	30.2	28.4	60.7	38.7	25.5	84.2	39.1	–	–	–	14.2	75.0	23.9	–	–	–
REL. NET[44][1]	11.1	20.0	14.3	14.0	35.7	20.1	22.9	76.9	35.3	18.6	87.3	30.6	11.5	60.9	19.4	13.8	73.8	23.2
CBZSL	29.0	32.7	30.7	21.1	43.5	28.5	38.9	67.2	49.3	34.1	72.5	46.4	18.8	70.8	29.6	31.3	28.5	29.8
UBZSL	31.7	34.0	32.8	31.5	46.3	37.5	38.7	69.3	49.6	37.1	75.1	49.7	24.0	67.4	**35.4**	27.2	78.2	**40.4**

marginal with respect to κ_0. Moving κ_1 towards zero encodes a local prior that imposes unrealistically large dispersion for centers of actual classes sharing the same meta-class, which violates the main assumption of our model that classes sharing the same meta class are semantically similar classes. On the other hand, moving κ_1 towards infinity encodes a local prior that imposes limited to no deviation among centers of actual classes which is another extreme that is not true for real-world datasets, i.e. classes are supposed to be statistically identifiable.

In both extremes unrealistic prior assumptions that cannot be reconciled with the characteristics of real-world data sets impede knowledge transfer between seen and unseen classes and lead to poor classification performance on unseen classes. On the other hand, the same extreme assumptions happen to help with seen class accuracies because likelihood and data-driven local priors (both of which lacks for unseen classes) outweigh the effect of unrealistic global prior in posterior predictive distributions.

2.4.2 Comparison with State of the Art

Results obtained by the proposed CBZSL and UBZSL models on SUN, CUB, FLO, AWA1, AWA2, aPY datasets are presented in Table 2.3. In addition to all SotA techniques reported in [67] we also included results of more recently published techniques [44], [82], [83] in this comparison. These results suggest that the proposed unconstrained model (UBZSL) demonstrates better performance than all other techniques but TCN. The constrained version of our model, i.e., CBZSL, also renders comparable results with the unconstrained version of the model despite its simplicity.

Results in Table 2.3 further show that in all of the experiments, unseen class accuracies achieved by our models are substantially higher than those achieved by all other techniques, but the TCN [82] model. This is achieved while maintaining a comparable performance on seen class accuracies in most of the experiments. Intuitively speaking, the two-level Bayesian hierarchy defined by meta-classes is expected to better manage the open space risk [84] by assigning an image of an unseen class to its meta class as opposed to misclassifying it into one of the seen classes.

Table 2.4. ImageNet results in nine different test phase configurations. Lp and Mp refer to least and most populated classes, respectively. 2/3 Hop represents the classes that are 2/3-hops away from 1K training classes according to the ImageNet label hierarchy. Finally, All appears for all 21K ImageNet classes. The results are in top-K accuracy.

	UBZSL			**CBZSL**			**SoA from [67]**		
Split	**1**	**5**	**10**	**1**	**5**	**10**	**1**	**5**	**10**
2Hop	2.6	13.1	20.3	**3.9**	**15.0**	**22.8**	2.2	10.3	19.3
3Hop	0.8	4.1	6.9	**1.0**	**4.1**	6.9	0.8	3.7	**7.2**
Lp500	1.8	5.1	8.6	**2.5**	**10.2**	**14.3**	1.9	6.1	10.4
Lp1K	1.2	4.6	7.3	**2.3**	**7.3**	**10.7**	1.4	4.8	8.5
Lp5K	0.5	2.0	3.5	**0.6**	**2.4**	**4.0**	0.4	2.2	3.9
Mp500	3.4	17.4	26.5	**7.5**	**25.2**	**35.0**	2.9	14.9	26.6
Mp1K	2.4	13.0	20.2	**4.8**	**17.3**	**25.5**	2.3	11.8	20.7
Mp5K	1.1	6.1	9.9	**1.5**	**6.6**	**10.5**	1.1	6.2	10.0
All	0.3	1.8	3.0	**0.4**	1.8	2.9	0.3	**2.0**	**3.4**

2.4.3 Large-Scale Experiments on ImageNet

ImageNet is currently the most challenging dataset for ZSL. Arguably it constitutes the most natural setup to evaluate ZSL learning performance as it contains 22K classes (1K of which are used to train state of the art deep neural networks) and most of these classes are sparsely populated.

Table 2.4 summarizes ImageNet results under nine different test set configurations. Our unconstrained model (UBZSL) improves over the state of the art in 2/3 Hop and highly populated test classes. Of particular importance is the highly competitive performance by the constrained model (CBZSL) that improves the current state of the art in all test configurations with respect to Top-1 accuracy (3.9% vs 2.18% on 2Hop, 7.51% vs 2.9% on Mp500, 4.78% vs 2.34% on Mp1000). Our model achieves the best results in eight of the nine test configurations for Top-5 and seven of the nine for Top-10 accuracies. Especially in the least populated (Lp500) classes the accuracy improvement is four percentage points in Top-5 and Top-10 accuracies. In most populated classes (Mp500) the accuracy gets almost doubled, i.e. 25.20% vs 14.86% on Top-5.

These results show that as the number of classes and the average number of samples per class (1300 in ImageNet vs 700 in benchmark datasets) increase, the explicit hierarchy across classes becomes more evident leading to more informative local priors. ImageNet contains both coarse- and fine-grained classes. The results suggest that our technique can be equally effective on datasets with hybrid granularity.

2.5 Conclusions

Summary of our contributions. In this study, we proposed a Bayesian approach to ZSL that relies on the consideration that classes in real-world datasets emerge at different levels of abstraction, and there are meta-classes that inherently organize the class hierarchy in the semantic space. We introduced concepts of local and global priors and showed that knowledge transfer from seen classes to unseen ones could be effectively carried out in the image space by a two-layer GMM. The proposed two-layer GMM offers extreme flexibility in modeling datasets with different characteristics by tuning its hyperparameters, each of which models a

different aspect of the data. We performed extensive experiments with benchmark datasets (fine-grained, coarse-grained, and large-scale) to demonstrate the utility of the proposed Bayesian approach for ZSL, which favors the proposed approach over other state-of-the-art inductive ZSL techniques.

Future Research Directions. Recently proposed transductive methods [58], [75], [85] have proved that generating features for unseen classes and treating ZSL as a closed-set classification can produce much better results than running ZSL in an inductive setting. Although features generated by these techniques do not seem to preserve correlation among features and are far from recovering unseen class distributions, they do preserve the relative distances among unseen classes, which in turn helps improve the performance of a softmax classifier in the closed-set setting. Thus, using prototypical feature vectors for unseen classes and integrating these into PPDs can offer significant boost for the performance of the proposed hierarchical Bayesian model. In our future work we aim to demonstrate that these prototypical feature vectors can be easily obtained by solving a simple compressed sensing problem and PPDs updated with these prototypical vectors can be used to generate new features in a probabilistic way. Such an approach can potentially preserve both the correlation among features and the relative distance between classes to generate more realistic features. Although not discussed in current work the proposed framework can be easily and effectively extended for any-shot learning problems, which will be a research direction we will pursue in parallel to probabilistic feature generation.

3. FINE-GRAINED ZERO-SHOT LEARNING WITH DNA AS SIDE INFORMATION

In the previous chapter, we introduced a novel hierarchical Bayesian classifier for ZSL task. In this chapter, we demonstrate that the current side information sources can be quite challenging to come by for fine-grained zero-shot species classification and propose a novel and efficient alternative. We further demonstrate that our proposed Bayesian classifier proves to be very resilient with various sources of side information and discuss the limitations of the current state-of-the-art ZSL methods.

Fine-grained zero-shot learning task requires some form of side-information to transfer discriminative information from seen to unseen classes. As manually annotated visual attributes are extremely costly and often impractical to obtain for a large number of classes, in this study we use DNA as side information for the first time for fine-grained zero-shot classification of species. Mitochondrial DNA plays an important role as a genetic marker in evolutionary biology and has been used to achieve near perfect accuracy in species classification of living organisms. We implement a simple hierarchical Bayesian model that uses DNA information to establish the hierarchy in the image space and employs local priors to define surrogate classes for unseen ones. On the benchmark CUB dataset we show that DNA can be equally promising, yet in general a more accessible alternative than word vectors as a side information. This is especially important as obtaining robust word representation for fine-grained species names is not a practicable goal when information about these species in free-form text is limited. On a newly compiled fine-grained insect dataset that uses DNA information from over a thousand species we show that the Bayesian approach outperforms state-of-the-art by a wide margin. This chapter corresponds to the following published work [26],

S. Badirli, Z. Akata, G. Mohler, C. Picard, and M. Dundar. Fine-grained Zero-shot Learning with DNA as Side Information. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

3.1 Introduction

Fine-grained species classification is essential in monitoring biodiversity. Diversity of life is the central tenet to biology and preserving biodiversity is key to a more sustainable life. Monitoring biodiversity requires identifying living organisms at the lowest taxonomic level possible. The traditional approach to identification uses published morphological dichotomous keys to identify the collected sample. This identification involves a tedious process of manually assessing the presence or absence of a long list of morphological traits arranged at hierarchical levels. The analysis is often performed in a laboratory setting by a well-trained human taxonomist and is difficult to do at scale. Fortunately, advances in technology have addressed this challenge to some extent through the use of DNA barcodes. DNA barcoding is a technique that uses a short section of DNA from a specific gene, such as *cytochrome C oxidase I (COI)*, found in mitochondrial DNA, and offers specific information about speciation in living organisms and can achieve nearly perfect classification accuracy at the species level [86], [87].

As it is costly to obtain the label information for fine-grained image classification of species, Zero-Shot Learning (ZSL) that handles missing label information is a suitable task. In ZSL, side information is used to associate seen and unseen classes. Popular choices for side-information are manually annotated attributes [15], [68], word embeddings [24], [40], [69] derived from free-form text or the WordNet hierarchy [45], [64]. It is often assumed that an exhaustive list of visual attributes characterizing all object classes (both *seen* and *unseen*) can be determined based only on seen classes. However, taking insects as our object classes, if no seen class species have antennae, the attribute list may not contain *antenna*, which may in fact be necessary to distinguish unseen species. In the United States alone, more than 40% of all insect species (>70,000) remain undescribed [88], which is a clear sign of the limitations of existing identification techniques that rely on visual attributes. Similarly, free-form text is unlikely to contain sufficiently descriptive information about fine-grained objects to generate discriminative vector embeddings. For example, *tiger beetle* is a class in the ImageNet dataset. However, the *tiger beetle* group itself contains thousands of known species and the Wikipedia pages for these species either do not exist or are limited

to short text that does not necessarily contain any information about species' morphological characteristics. WordNet hierarchy may not be useful either as most of the species names do not exist in WordNet.

Given that DNA information can be readily available for training [89], [90], species-level DNA information can be used as highly specific side information to replace high-level semantic information in ZSL. For seen classes, species-level DNA information can be obtained by finding the consensus nucleotide sequence among samples of a given species or by averaging corresponding sequence embeddings of samples. For unseen classes, species-level DNA information can be obtained from actual samples, if available, in the same way as seen classes, or can be simulated in a non-trivial way to represent potentially existing species.

Our approach uses DNA as side information for the first time for zero-shot classification of species. In fine-grained, large-scale species classification, no other side information can explain class dichotomy better than DNA, as new species are explicitly defined based on variations in DNA. The hierarchical Bayesian model leverages the implicit inter-species association of DNA and phenotypic traits and ultimately allows us to establish a Bayesian hierarchy based on DNA similarity between unseen and seen classes. We compare DNA against word representations for assessing class similarity and show that the Bayesian model that uses DNA to identify similar classes achieves favorable results compared to the version that uses word representations on a well-known ZSL benchmark species dataset involving slightly less than 200 bird species. In the particular case of an insect dataset with over 1000 species, when visual attributes or word representations may not offer feasible alternatives, we show that our hierarchical model that relies on DNA to establish class hierarchy significantly outperforms all other embedding-based methods and feature generating networks.

Our contributions are on three fronts. First, we introduce DNA as side information for fine-grained ZSL tasks, implement a Convolutional Neural Net (CNN) model to learn DNA barcode embeddings, and show that the embeddings are robust and highly specific for closed-set classification of species, even when training and test sets of species are mutually exclusive. We use the benchmark CUB dataset as a case study to show that DNA embeddings are competitive to word embeddings as side information. Second, we propose a fine-grained insect dataset involving $21,212$ matching image/DNA pairs from 578 genera

and $1,213$ species as a new benchmark dataset and discuss the limitations of current ZSL paradigms for fine-grained ZSL tasks when there is no strong association between side information and image features. Third, we perform extensive studies to show that a simple hierarchical Bayesian model that uses DNA as side information outperforms state-of-the-art ZSL techniques on the newly introduced insect dataset by a wide margin.

3.2 Related Work

3.2.1 Zero-Shot Learning

Early ZSL literature is dominated by methods that embed image features into a semantic space and perform various forms of nearest neighbor search to do inference [40], [41], [69]. As the dimensionality of semantic space is usually much smaller than the feature space this leads to the hubness problem where some classes become *hub* and occur as the nearest neighbor of many samples. In an effort to alleviate the hubness problem, [48], [49] changed the direction of the embedding from semantic to image feature space. This was followed by a line of work that investigates bidirectional embedding between semantic and image spaces through a latent space [43]–[47].

In [91], [92], a new strategy of synthesizing features for unseen classes and converting the challenging ZSL problem into traditional supervised learning is introduced [50]–[53], [55], [57]–[59], [93]. Although feature generating networks (FGNs) currently achieve state-of-the-art results in ZSL, they suffer from the same problem as earlier lines of work in ZSL: hypersensitivity towards side information not strongly correlated with visual attributes. The vulnerability of both embedding and FGN-based methods toward sources of side information different than visual attributes, such as word vectors or WordNet hierarchy, is investigated in [45], [58], [93]. Another limitation of FGNs is that features generated for unseen classes are significantly less dispersed than actual features due to the generator failing to span more than a small subset of modes available in the data. Recent deep generative models mitigate this problem by proposing different loss functions that can better explore inter-sample and inter-class relationships [94]–[98]. However, these methods fail to scale well with an increasing number of classes with an especially high inter-class similarity [99].

3.2.2 Side Information in ZSL

Side information serves as the backbone of ZSL as it bridges the knowledge gap between seen and unseen classes. Earlier lines of work [41], [80] use visual attributes to characterize object classes. Although visual attributes achieve compelling results, obtaining them involves a laborious process that requires manual annotation by human experts not scalable to data sets with a large number of fine-grained object classes. When dealing with fine-grained species classification, apart from scalability, a more pressing obstacle is how to define subtle attributes potentially characteristic of species that have never been observed.

As an alternative to manual annotation, several studies [25], [40], [45], [61]–[63] proposed to learn side information that requires less effort and minimal expert labor such as textual descriptions, distributed text representations, like Word2Vec [24] and GloVe [100], learned from large unsupervised text corpora, taxonomical order built from a pre-defined ontology like WordNet [64], or even human gaze reaction to images [101]. The accessibility, however, comes at the cost of performance degradation [45], [58]. A majority of ZSL methods implicitly assume strong correlation between side information and image features, which is true for handcrafted attributes but less likely to be true for text representations or taxonomic orders. Consequently, all these methods experience significant decline in performance when side information is not based on visual attributes.

3.3 Barcode of Life Data and DNA Embeddings

In this study, we present the fine-grained INSECT dataset with $21,212$ matching image/DNA pairs from $1,213$ species (see Fig. 3.1 for sample images). Unlike existing benchmark ZSL datasets, this new dataset uses DNA as side information[1] and can be best characterized with the high degree of similarity among classes. Among the existing benchmark datasets, SUN contains the largest number of classes (717) but classes in SUN represent a wide range of scene categories related to transportation, indoors and outdoors, nature, underwater etc., and as such can be considered a relatively coarse-grained dataset compared to the INSECT dataset we are introducing in this study.

[1]↑Please refer to Appendix B.2 for discussion on limitations of using DNA as side information

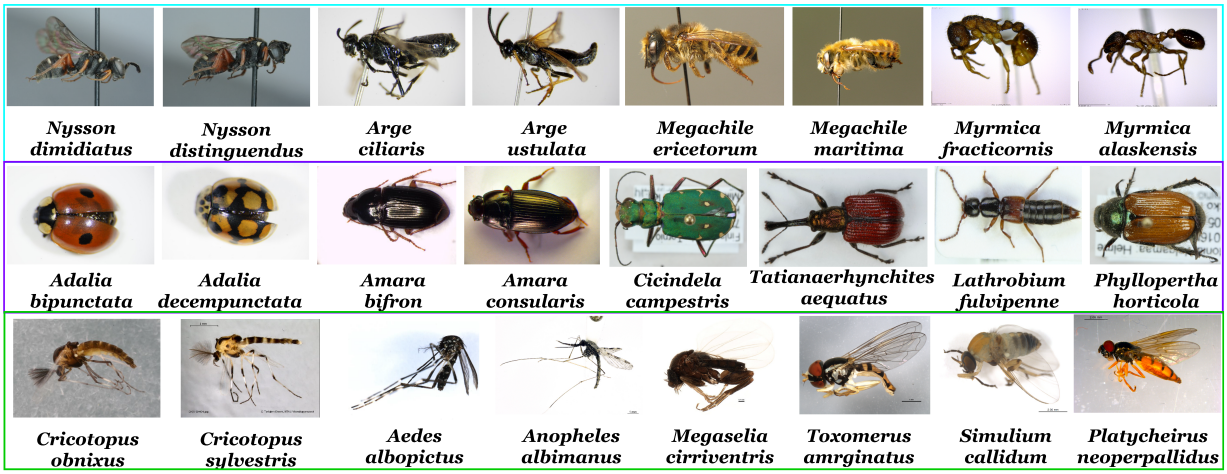


Figure 3.1. Image samples from the INSECT dataset. Rows represents a small subset of species from three orders: Hymenoptera, Coleoptera and Diptera, respectively. The first word in names indicate genus, the two words together define the species name.

All insect images and associated DNA barcodes in our dataset come from the Barcode of Life Data System (BOLD) [89], [90]. BOLD is an open-access database in which users can upload DNA sequences and other identifying information for any living organism on Earth. The database provides approximately 658 base pairs of the mitochondrial DNA barcode extracted from the *cytochrome c oxidase I* (COI) gene along with additional information such as country of origin, life-stage, order, family, subfamily, and genus/species names.

3.3.1 Data Collection

We collected image/DNA pairs of insects that originate from three orders: Diptera (true flies), Coleoptera (beetles) and Hymenoptera (sawflies, wasps, bees, and ants). While the dataset is in general clean, manual effort was devoted to further curate the dataset. Only cases with images and matching DNA barcodes of adult insects are included. Images from each species were visually inspected and poor quality images were deleted. Only species with more than ten instances were included. The final dataset consisted of $21,212$ images and $1,213$ insect species of which 254 belong to Diptera (133 genera), 564 to Coleoptera (315 genera) and 395 to Hymenoptera (130 genera). We extracted image features, namely

Table 3.1. ZSL split details. Y^s, and Y^u denote the seen and unseen test sets, whereas Y^{all} represents entire data. There are $15,262$ $(21,212-3,525-2,425)$ samples left for the training set.

	Y^{all}	Y^s	Y^u
#Images	21,212	3,525	2,425
#Classes	1,213	1,080	121

image embeddings, using a pre-trained (on ImageNet 1000 classes) ResNet101 model [3]. Images are resized to 256×256 and center-cropped before fed to the ResNet model. No other pre-processing is applied to the images.

3.3.2 Data Split

We randomly chose 10% of all species as unseen classes for the test set leading to $1,092$ seen and 121 unseen classes. Similarly, we randomly chose 10% of the $1,092$ training classes as unseen classes for the validation set. Samples from seen classes were split by an 80/20 ratio in a stratified fashion to create seen portion of the train and test datasets. In the dataset there were a few hundred cases where multiple image views (dorsal, ventral, and lateral) of the same insect were present. To avoid splitting these cases between train and test, we made sure all instances of the same insect are included in the training set. As a result, 12 of the $1,092$ seen classes in the training set were not represented in the test set. Our dataset splits are summarized in Table 3.1.

3.3.3 DNA Embeddings

Although it is the first time DNA barcodes are used as side information in ZSL domain, there have been some work investigating vector embeddings for DNA sequences. Authors of [102] trained a CNN model to do binary DNA sequence classification considering sequences as a text data. Imitating amino acid structure, each triplet of base pairs is treated as a word and sequences are converted into one-hot vector representation. Taking [24] as the base, [103] trained a shallow neural network on human genome data to generate representation for k-mers. Unlike these techniques we deal with DNA Barcodes represented by nucleotide

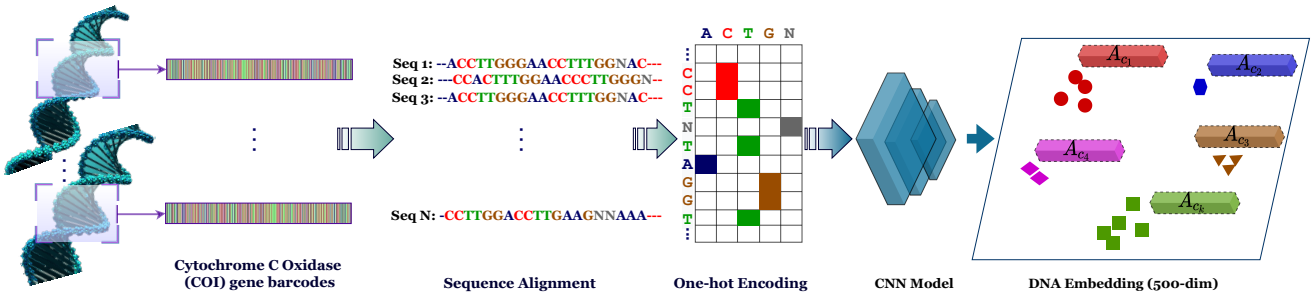


Figure 3.2. Attribute extraction from mitochondrial DNA.

sequences and aim to convert the entire character sequence into a vector embedding useful for species classification with more than 1,000 classes. Most recently, DNABERT [104] adapted the powerful text transformer model [105] to a genomic DNA setting and generated vector embeddings for long DNA sequences. In this paper, we trained a CNN model to learn a vector representation of DNA barcodes in the Euclidean space. First, the consensus sequence of all DNA barcodes in the training set with 658bp is obtained. Then, all sequences are aligned with respect to this consensus sequence using a progressive alignment technique implemented in MATLAB R2020A (Natick, MA, USA). A total of five tokens are used, one for each of the four bases, *Adenine*, *Guanine*, *Cytosine*, *Thymine*, and one for *others*. All ambiguous and missing symbols are included in the *others* token. In pre-processing, barcodes are one hot encoded into a 658x5 2D array, where 658 is the length of the barcode sequence (median of the nucleotide length of the DNA data).

To train the CNN model, a balanced subset of the training data is subsampled, where each class size is capped at 50 samples. The CNN is trained with $14,723$ barcodes from $1,092$ classes. No barcodes from the 121 unseen classes are employed during model training. The training set is further split into two as train (80%) and validation (20%) by random sampling. We used 3 blocks of convolutional layers each followed by batch normalization and 2D max-pooling. The output of the third convolutional layer is flattened, and batch normalized before feeding the data into a fully-connected layer with 500 units. The CNN architecture is completed by a softmax layer. We used the output of the fully-connected layer as the embeddings for DNA. Class level attributes are computed by the mean embedding of each class. The DNA-based attribute extraction is illustrated in Figure 3.2. The CNN model

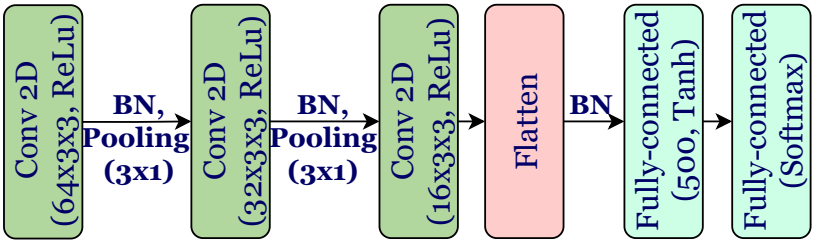


Figure 3.3. CNN model architecture

architecture is depicted in Figure 3.3. We used ADAM optimizer for training the model for five epochs with a batch size of 32 (with a step-decay initial learning rate = 0.0005 and drop factor= 0.5, $\beta_1 = 0.9$, $\beta_2 = 0.999$). The model is developed in Python with Tensorflow-Keras API.

3.3.4 Predictive Accuracy of DNA Embeddings

Although the insect barcodes we used are extracted from a single gene (COI) of the mitochondrial DNA with a relatively short sequence length of 658 base pairs, they are proven to have exceptional predictive accuracy; the CNN model achieves a 99.1% accuracy on the held-out validation set. Note that, we only used the data from training seen classes to train the CNN model. In order to validate the generalizability of embeddings to unseen data, we trained a simple K-Nearest Neighbor classifier ($K = 1$) on the randomly sampled 80% of the DNA-embeddings of unseen classes and tested on the remaining 20%. The classifier had a perfect accuracy for all 121 but one classes with an overall accuracy of 99.8%. In addition to our CNN model we have explored a DNABERT [104] model for converting DNA barcodes to vector embeddings. The pretrained DNABERT model achieves around 85% (vs 99% from CNN) top-1 KNN accuracy (averaged over 10 runs) on the unseen classes. Pretrained DNABERT can be fine-tuned for species classification however because of the vast number of parameters to tune each run takes a few hours on a relatively sophisticated GPU, significantly more than CNN training. Similarly, a simple LSTM model with half of the parameters as the CNN model is almost 5 times slower than the CNN model and requires

more epochs to reach a reasonable accuracy. Therefore, we use a simple 3-layer CNN that trains in an hour and achieves almost perfect top-1 KNN accuracy.

To demonstrate that the approach can be easily extended to larger members of the animal kingdom, we compiled approximately 26,000 DNA barcodes from 1,047 bird species to train another CNN model (ceteris paribus) to learn the DNA embeddings for CUB dataset (see the Supp. materials for details). The CNN model achieved a compelling 95.60% on the held-out validation set. The promising classification performance on both insect and bird datasets proves that DNA barcodes can be used as a reliable and accurate source of side information for fine-grained ZSL task.

3.3.5 Hierarchical Bayesian Approach

Object classes in nature already tend to emerge at varying levels of abstraction, but the class hierarchy is more evident when classes represent species and species are considered the lowest taxonomic rank of living organisms. We build our approach on a two layer hierarchical Bayesian model that was previously introduced and evaluated on benchmark ZSL datasets with promising results (See Chapter 2). The model assumes that there are latent classes that define the class hierarchy in the image space and uses side information to build the Bayesian hierarchy around these latent classes. Two types of Bayesian priors are utilized in the model: global and local. As the name suggests, global priors are shared across all classes, whereas local priors represent latent classes, and are only shared among similar classes. Class similarity is evaluated based on side information in the Euclidean space. Unlike standard Bayesian models where the posterior predictive distribution (PPD) forms a compromise between prior and likelihood, this approach utilizes posterior predictive distributions to blend local and global priors with data likelihood for each class. Inference for a test image is performed by evaluating posterior predictive distributions and assigning the sample to the class that maximizes the posterior predictive likelihood[2]. For the model details regarding surrogate class formation and PPD derivation please check the Section 2.3

[2]↑ The code and dataset are available at https://github.com/sbadirli/Fine-Grained-ZSL-with-DNA

3.3.6 Rationale for the Hierarchical Bayesian Approach and Limitations

We believe that the hierarchical Bayesian model is ideally suited for fine-grained zero-shot classification of species when DNA is used as side information for the following reasons. The performance of the model in identifying unseen classes depends on how robust the local priors can be estimated. This in turn depends on whether or not the set of seen classes contain any classes similar to unseen ones. As the number of seen classes increases, seen classes become more representative of their local priors, more robust estimates of local priors can be obtained, and thus, unseen classes sharing the same local priors as seen classes can be more accurately identified. On the other hand, if the class-level side information is not specific enough to uniquely characterize a large number of classes, then the model cannot evaluate class similarity accurately and local priors are estimated based on potentially incorrect association between seen and unseen classes. In this case having a large number of seen classes available may not necessarily help. Instead, highly specific DNA as side information comes into play for accurately evaluating class similarity (See Figure 3.4). If a unique local prior can be eventually described for each unseen class, then unseen classes can be classified during test time without the model having to learn the mapping between side information and image features beforehand. Uniqueness of the local prior can only be ensured when the number of seen classes is large compared to the number of unseen classes. Thus, the ratio of the number of seen and unseen classes becomes the ultimate determinant of performance for the hierarchical Bayesian model. The higher this ratio is the higher the accuracy of the model will be. An experiment demonstrating this effect is performed in Section 3.4.3.

If the same set of K classes is found to be the most similar for two different unseen classes, then these two unseen classes will inherit the same local prior and thus they will not be statistically identifiable during test time. The likelihood of such a tie happening for fine-grained data sets quickly decreases as the number of classes increases. In practice we deal with this problem by replacing the least similar of the K most similar seen classes by the next most similar seen class for one of the unseen classes.

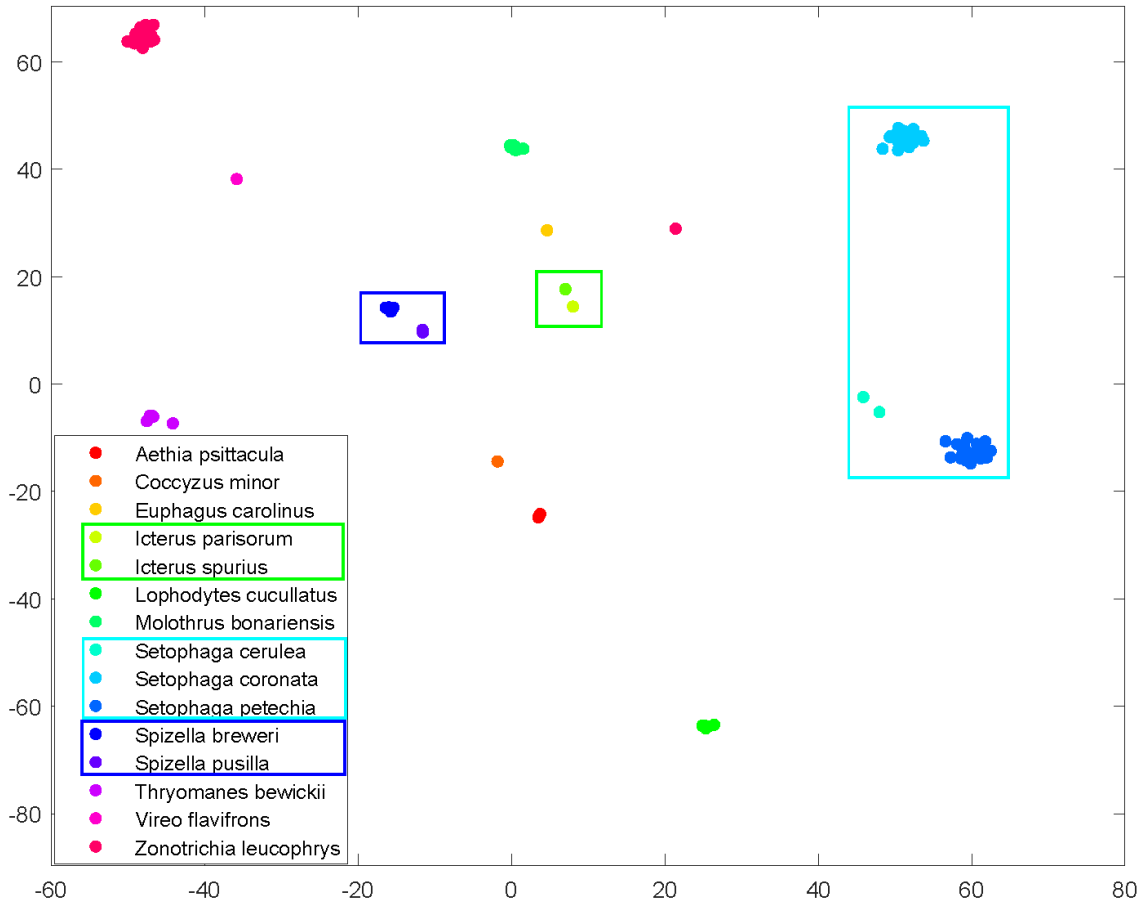


Figure 3.4. TSNE plot of DNA embeddings from CUB dataset using randomly selected 15 classes. Class names are represented by birds' scientific names. Observe that species belonging to the same genus thus sharing very akin morphological traits are nicely grouped closer to each other inside the colored rectangles. Although COI gene barcodes does not have explicit association between image features, visually similar species also cluster closer in the DNA space.

3.4 Experiments

In this section we report results of experiments with two species datasets that use DNA as side information. Details of training and hyperparameter tuning are provided in the Appendix B.3.

3.4.1 Experiments on the INSECT Dataset

We compare our model (BZSL) against state-of-the-art (SotA) ZSL methods proved to be most competitive on benchmark ZSL datasets that use visual attributes or word vector representations as side information. Selected SotA models represent various ZSL categories: (1) Embedding methods with traditional [41], [42] and end-to-end neural network [106] approaches, (2) FGNs using VAE [58] and GAN [93], and (3) end-to-end few shot learning approach extended to ZSL [44]. Table 3.2 displays seen (**S**) and unseen (**US**) accuracies

Table 3.2. Generalized ZSL results on INSECT data using DNA barcodes as attributes.

Method	**US**	**S**	**H**
CRNet [106]	13.33	19.70	15.90
ALE [41]	2.86	27.18	5.17
RelationNet [44]	3.25	24.37	5.73
CADA-VAE [58]	14.55	20.81	17.10
ESZSL [42]	3.41	18.61	5.77
LsrGan [93]	12.58	30.41	17.75
BZSL [65]	**20.83**	**38.30**	**26.99**

and their harmonic mean (**H**) on the INSECT data using DNA as the side information. Results suggest that the large number of seen classes along with the highly specific nature of DNA information in characterizing classes particularly favors the Bayesian method to more accurately estimate local priors and characterize surrogate classes. The harmonic mean achieved by the proposed method is 52% higher than the harmonic mean achieved by the second-best performing technique. Similar levels of improvements are maintained on both seen and unseen class accuracies. The next top performers are FGNs. CADA-VAE uses a VAE whereas LsrGan utilizes GAN to synthesize unseen class features, then both train a *LogSoftmax* classifier for inference. Lower unseen class accuracies suggest that FGNs struggle to synthesize meaningful features in the image space. On the other hand, CRNet that uses end-to-end neural network to learn the embedding between semantic and image spaces renders slightly worse performance than FGNs. It seems, non-linear embedding also works better than a linear (ESZSL) and bilinear (ALE) ones for this specific dataset. RelationNet is amongst the ones with the lowest performance, as the method is explicitly designed for

Few-shot learning and expects the side information to be strongly correlated with image features. The weak association between side information and image features affects the performance of both FGNs and embedding methods, but the traditional embedding methods suffer the most.

Table 3.3. Generalized ZSL results on CUB data using original visual attributes, word vectors, and DNA attributes. **US**, **S**, and **H** represent unseen, seen class accuracies and harmonic mean, respectively.

	Attributes			Word Vectors			DNA		
Method	**US**	**S**	**H**	**US**	**S**	**H**	**US**	**S**	**H**
CRNet [106]	44.28	59.84	50.89	22.75	45.92	30.43	9.27	56.56	15.93
ALE [41]	25.15	60.80	35.59	3.95	48.57	7.31	3.50	50.18	6.54
RelationNet [44]	11.66	44.81	18.50	8.67	36.16	13.99	5.33	40.83	9.42
CADA-VAE [58]	47.15	53.11	49.95	26.45	41.98	**32.45**	19.42	37.05	25.48
ESZSL [42]	15.58	50.66	23.84	2.26	23.86	4.12	5.99	5.38	5.67
LsrGan [93]	47.65	56.97	**51.89**	24.63	37.96	29.88	15.99	33.57	21.66
BZSL [65]	31.49	50.61	38.82	22.43	45.00	29.94	27.46	48.14	**34.97**

3.4.2 Experiments on the Benchmark CUB Dataset

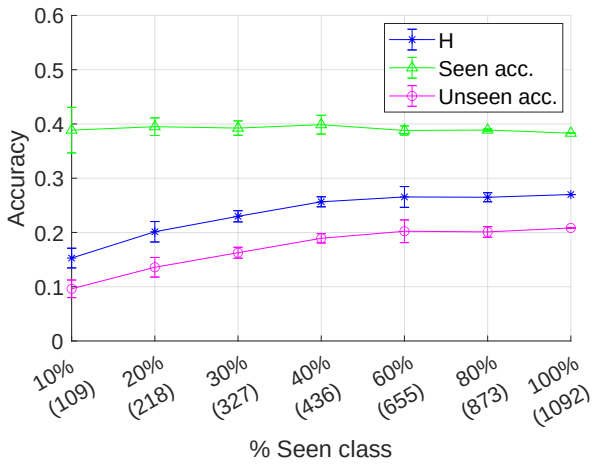
To demonstrate the utility of DNA-based attributes in a broader spectrum of species classification, we procured DNA barcodes, again from the BOLD system, for bird species in the CUB dataset. For this experiment, we derived 400 dimensional embeddings in order to have the same size with word vectors and eliminate the attribute size effect. There were 6 classes, 4 seen and 2 unseen, that did not have DNA barcodes extracted from COI gene in the BOLD system. These classes were excluded from the dataset but the proposed split from [67] is preserved otherwise.

The results shown in Table 3.3 validate our hypothesis that when side information is not strongly correlated with visual characteristics of object classes (like in word vectors or DNA) both embedding methods and FGNs display significant performance degradation. With the exception of the proposed Bayesian model, word vector representation yields better accuracy than DNA-based attributes for all models. This phenomenon can be explained by our observation that text fragments related to common animals/birds in the Wikipedi-

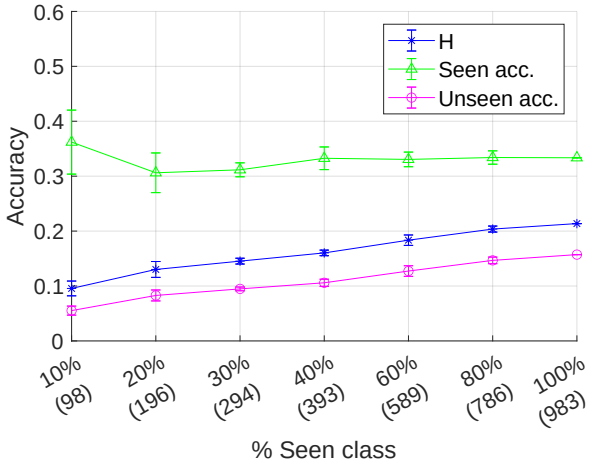
a/Internet often include some morphological traits of the underlying species. Hence, word vector representation is expected to have higher degree of correlation to visual attributes than DNA information. Our model produces the best results, 34.97% vs 32.45% when the side information is not derived from visual characteristics of classes. This outcome validates the robustness of the Bayesian model to diverse sources of side information and emphasizes the need for more robust FGN or embedding based models in more realistic scenarios where hand-crafted visual attributes are not feasible.

3.4.3 The Effect of the Number of Seen Classes on Performance

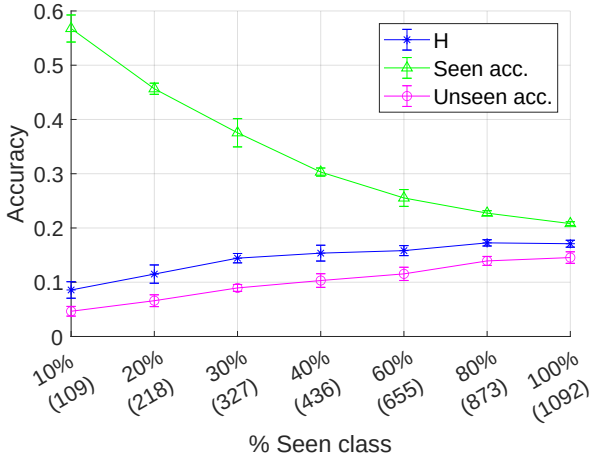
Local priors are central to the performance of the hierarchical Bayesian model. Here, we perform experiments to show that as the number of seen classes increases while the number of unseen classes is fixed, each unseen class can be associated with a larger pool of candidate seen classes and more informative local priors can potentially be obtained, which in turn leads to more accurate identification of unseen classes. To demonstrate this effect, we run two experiments. In the first experiment we use the same set of unseen classes as in Section 3.4.1 but gradually increase the number of seen classes used for training. In the second experiment we double the size of the unseen classes and gradually include the remaining classes into training as seen classes. The first experiment is also performed for CADA-VAE. LsrGan is skipped for this experiment due to long training time. To account for random subsampling of seen classes each experiment is repeated five times and error bars are included in each plot. There is a clear trend in these results that further highlights the intuition behind the hierarchical Bayesian model and explains why this model is well-suited for fine-grained ZSL. When 10% of the classes are used as unseen, unseen class accuracy improves with increasing number of seen classes until it flatlines beyond the 60% mark while seen class accuracy always maintains around the same level (see Fig. 3.5a). When 20% of the classes are used as unseen no flatlining effect in unseen class accuracy is observed even at 100% mark, which suggest that there is still room for improvement in unseen class accuracy if more seen classes become available (see Fig. 3.5b). For CADA-VAE unseen class accuracy initially improves and then flatlines beyond 80% mark but this improvement comes at the



(a) BZSL results in original setup ($Y^s_{tr} = 1,092$ and $Y^u = 121$)



(b) BZSL results with $Y^s_{tr} = 983$ and $Y^u = 230$



(c) CADA-VAE in original setup ($Y^s_{tr} = 1,092$ and $Y^u = 121$)

Figure 3.5. The effect of the number of seen classes on the performance of BZSL and CADA-VAE. Each experiment is repeated five times to account for random subsampling of seen classes.

expense of significant degradation in seen class accuracy, which suggest that as the number of seen classes increase generated features further confound the classifier as would be expected of an FGN for a fine-grained dataset.

3.5 Conclusions

Visual attributes tend to be the top preference as side information in ZSL task, yet they quickly lose their appeal with an increasing number of classes. Word vectors have been proposed as a scalable solution for this problem, nonetheless, their utility significantly diminishes once very fine-grained datasets are involved such as species datasets. Considering the tens of thousands of *described* species and even larger number of *undescribed* species, we demonstrate that the DNA barcodes are practical and effective alternative as side information to perform large-scale, fine-grained zero-shot classification of species. Leveraging our hierarchical Bayesian classifier, we show DNA barcodes' utility in evaluating class similarity for the purpose of identifying unseen classes in a fine-grained ZSL setting. The proposed Bayesian model proves to be very resilient against various sources of side information. In fact, on CUB dataset, our model delivers the best performance while using side information that is not based on visual attributes. On INSECT dataset, the proposed algorithm yields impressive 52% performance boost over the second best performing ZSL technique. We show that taking the presence of an explicit association between visual attributes and image features for granted is the main reason for SotA ZSL methods' significant performance degradation when non-visual attributes such as word vectors and WordNet are used as side information. The same effect is observed in our experiments once DNA embeddings are utilized. The experiments also reveal that Bayesian model does not require side information to have strong correlation with image features as long as they are informative enough to expose the class hierarchy. Furthermore, unlike the feature generating networks, the proposed model can use the increasing number of seen classes to its advantage for a better unseen class local prior estimation without sacrificing on seen class accuracy.

These favorable results by a simpler model suggest that as the number of classes increases along with inter-class similarity, the complexity of the mapping between side information and image attributes emerges as a major bottleneck at the forefront of zero-shot classification. A promising future research avenue appears to be implementing hierarchically organized FGNs where each sub-component only operates with a small subset of seen classes all sharing the same local prior.

4. CLASSIFYING THE UNKNOWN: IDENTIFICATION OF INSECTS BY DEEP ZERO-SHOT BAYESIAN LEARNING

In chapter 3, we discussed how DNA barcodes can be effectively used as a side information for fine-grained Zero-shot learning task. Motivated by agility of the Bayesian model and predictive capacity of DNA barcodes, in this chapter, we extend the traditional ZSL into a more realistic species identification and discovery without relying on any side information for unseen classes. Assuming a class-specific auxiliary information for unknown classes is impractical in real world discovery and not scalable if the number of estimated unseen classes are in the order of thousands. We show that once the hierarchical grouping of training classes are readily available, the proposed Bayesian classifier can leverage this hierarchy to form local priors around these groupings and facilitate scalable unseen class detection. Furthermore, we demonstrate that the increasing number of training classes makes seen classes becoming more representative of their local priors, thus, a more robust estimation of local priors which leads to a better identification of unknown classes. We also develop several benchmark techniques to build a baseline for a comparison with the Bayesian classifier. The study presented in this chapter discusses the application of our Bayesian classifier to entomology domain and corresponds to the following works [107], [108],

S. Badirli, C. Picard, G. Mohler, Z. Akata, M. Dundar. (2021). Classifying the Unknown: Identification of Insects by Deep Zero-shot Bayesian Learning. Research Square, https://doi.org/10.21203/rs.3.rs-1099185/v1.

S. Badirli, C. Picard, M. Dundar. Zero-shot Insect Identification and Discovery (Abstract), In *30th International Congress on Conservation Biology* (Oral Talk), 2021.

4.1 Introduction

Diversity of life is a central tenet to Biology, from the process of speciation to the maintenance or prevention of extinction (adaptation) and the ecosystem services biodiversity provides. Human activity threatens this, and as a result, the well-being and economics of humans are threatened. Biodiversity is important for health and medicine [109], drug

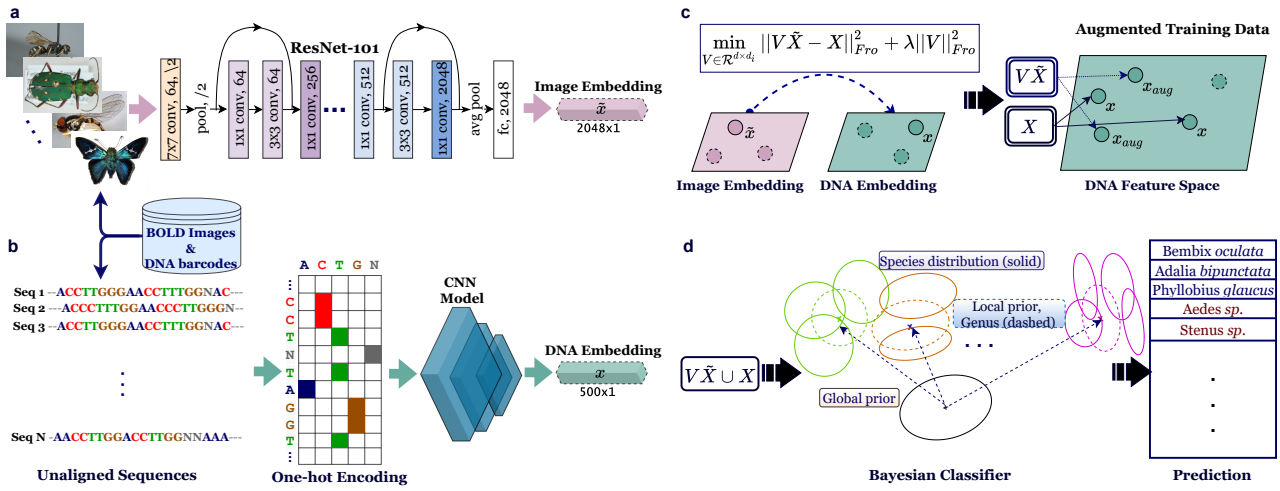


Figure 4.1. Deep Zero-shot Bayesian Classification with Unknown and Undescribed Species. **a.** Image embeddings of size 2048 are obtained using the pretrained ResNet-101 model. **b.** CNN architecture is trained using one-hot encoding representations of DNA barcodes (see Fig 3.3 for more details). **c.** Mapping from ResNet features to CNN embeddings is learned by transductive Ridge regression. Training set for the CNN embeddings is augmented by the mapped versions of ResNet features. **d.** Zero-shot Bayesian model is trained on the augmented training set and used for classification. A test sample is either assigned to one of the described species or identified as a new species belonging to one of the described genera.

discovery [110], social equality [111], ecosystem services [112], food security [113], and for life [114]. The time is now for innovative solutions to address the current and future losses of biodiversity, however, the problem is confounded with the enormous task when it comes to assessing biodiversity in taxonomic groups with large numbers of taxa for which the majority remain undescribed.

One of the largest groups of animals on the planet is the insects, and they are the most diverse, yet, so few of them are described, and they are disappearing faster than they can be identified [115]. Within Insecta, approximately 5.5 million insects species are thought to exist, yet only 20% are described, leaving a very large swath of unknown biodiversity [116]. Describing biodiversity for insects requires discovery and identification. Once an insect is collected, an individual with taxonomic knowledge will attempt to identify it to the lowest taxonomic level based on existing morphological dichotomous keys [117]. Alas, the inherent

flaw: undescribed species would not be present in a key, and only through the very thorough analysis of characters distinguishing the unknown from all others could one conclude it may be a new, undescribed species and is not attributed to plasticity or geographic isolation.

The use of newer technology, specifically, the DNA Barcoding [118], has helped confirm new species if the variation in sequence exceeds the traditional intraspecific variation or when species have indistinguishable characteristics such as cryptic species [119]. The DNA Barcode Database (BOLD) [89], [90]) will reveal that a search of Diptera yields 2.4 million records (DNA sequences) and 126,000 BINs (barcode indexed numbers), yet only 25,000 species have been identified, meaning DNA is facilitating the possibility of new species discovery, but nothing is happening to identify them. So even with DNA sequencing increasing the rate of new species discoveries, they are not being identified nor published, and the biology around these new species is not being discovered. These powerful DNA-based methods provide some possible estimates of biodiversity, yet do not contribute to the knowledge base.

The increasingly difficult challenge is due to the lack of experts in a given taxonomic field owing to the vast diversity of the insects themselves, and the decline of the art of traditional taxonomy [120]–[122]. Therefore, the only way to meaningfully scale the discovery and identification of new species is to address that point. If there is a means to perform this function across a broad scale, the insect identification problem becomes manageable, and this is where machine learning (ML) algorithms can be leveraged to find patterns from insect images and apply this to identifying insect species. Recent studies demonstrated that ML approaches can achieve human-expert level accuracy on image-based taxonomic identification [123]–[125]

Recent advances in ML have led to a surge of interest in entomology and ML methods provide potential solutions to the many challenges in the domain. Deep learning (DL) approaches, in particular those involving Convolutional Neural Networks (CNN), are utilized in pest-detection [126], [127], digitalization of Natural History Museum collections [128], [129], measuring invertebrate biodiversity [130], [131], investigating the plant-insect interactions [132] and many more applications [133]. ML methods have also been employed for a more challenging task of automatic detection of species in video and time-lapse images [134].

At the heart of the issue is that insect identification requires a method that is aware of open-set nature of the problem owing to the fact that species not currently described will not be present in an existing database, wherein all existing algorithms employ closed-set methods [21]. Current open-set classification methods have been employed on relatively small datasets and do not scale well with a larger number of classes [21]. Furthermore, such approaches have only been restricted to detect an insect sample as an outlier and cannot differentiate between different types of outliers [84], [135], [136]. This in turn limits their usefulness in entomology as insect datasets are very fine-grained and contain a large number of similar classes.

Traditional supervised learning algorithms will be inherently limited by the non-exhaustive nature of insect repositories available for training. It is impractical, often impossible, to create a training repository with a complete set of insect species for various reasons. For example, some of the insect species are not yet described, and thus well-characterized training images of insects from these species simply cannot be obtained. Similarly, when insect species are either rare in a given geographical locale, collecting samples may become impractical. And finally, insects specifically pose a challenge due to the morphologically distinct life stages of the insects.

Identifying samples of undescribed species is an ill-defined problem. However, ML models can be tailored to operate in a setting that is capable of detecting insect samples with no matching classes in the training data. In this study, we adapt the Generalized Zero-shot Learning (ZSL) setting [67] using class taxonomy as auxiliary information to facilitate identification of new insect species. In brief, we seek to answer whether recent advances in machine learning and computer vision can help extract subtle yet potentially discernible morphological characteristics, and when combined with DNA Barcode data, whether this can help facilitate more accurate identification of insects of described species while simultaneously discovering insects of undescribed origin (See Fig 4.1).

Traditional ZSL utilizes class-specific side information to identify unseen classes. The number of unseen classes can grow to arbitrarily large numbers as long as each unseen class can be described by some form of auxiliary information. However, acquiring this information is quite challenging and often infeasible for real world problems, in particular if class numbers

are in the order of thousands. Unlike the ZSL setting, in this approach, we only use class hierarchy observed from seen classes to discover new classes. This flexible and more practical setup, however, comes with a limitation that the number of new classes that can be uniquely described is limited to the number of local priors estimated from seen classes, that is, all the new classes assigned to the same local prior are treated as one class. Further details regarding classification setup are provided in Section 4.2.2.

4.2 Methods

In this section, we first introduce the dataset and how the split is performed for machine learning training. Next, Convolutional Neural Network (CNN) model for deriving DNA embeddings is presented. Finally, we lay out the zero-shot Bayesian classifier details along with the bioinformatics baseline classifier.

4.2.1 Barcode of Life Data System

Our study uses insect data from the Barcode of Life Data System (BOLD) [89], [90]. As other databases exist of genetic data (for example, [137]), they require some identification prior to depositing into the database. BOLD differs slightly in that as it allows for unidentified organisms to be uploaded into the database, and their algorithms, based on DNA sequence only, will place the unknown into a barcode index number (BIN). This allows for the quantification of the unknown and undescribed, however, no identifications are made. This data repository does not contain samples of truly undescribed species. The BOLD database using a specific searching algorithm that translates the DNA sequence to its protein sequence and searches its database. BOLD will make a species identification if the queried sequence contains less than 1% divergence to a reference specimen located in the database. If the sequence divergence is less than 3% (but greater than 1%), the database will make a match to a genus.

BOLD is an open-access database in which users can upload DNA sequences and other identifying information regarding any animal on Earth. Because the majority of the uploads are not identified species, they are classified into BINs [90]. For example, as of 8/18/2021, the

Insecta database had a total of 5,883,100 records with sequences, and about half had species names (2,561,685), meaning the remainder could not be identified to species. The data are important for assessing biodiversity, distributions of species, as well as collating other descriptive metadata and images. The limitations of this database are that it is important for the discovery of new species but does not allow for the identification of such, and simply places the outliers in an interim position, not allowing for any forward movement.

Data Collection. For this study, the data are collected based on a subset of insects that originate from four major Insecta orders: Diptera (true flies), Coleoptera (beetles), Lepidoptera (butterflies and moths), and Hymenoptera (sawflies, wasps, bees, and ants). The same data cleaning process from Section 3.3.1 is followed here as well. After the data cleaning, the final dataset consists of 1,040 insect species and a total of 32,848 insect instances (records). In the finalized dataset, we obtain 108 species of Diptera from 63 genera, 329 species of Coleoptera from 164 genera, 189 species of Hymenoptera from 59 genera, and 414 species of Lepidoptera from 82 genera (See Figure 4.2)

The same pretrained ResNet101 model [3] from previous work is used to embed images into Euclidean vector space and represent them by information-rich 2048 dimensional real-valued feature vectors. We utilize the ResNet101 model parameters that were optimized on ImageNet 1000 classes, and we have not fine-tuned the model on our dataset. Images are first resized to 256×256, then center-cropped into the ResNet model image dimension: 224×224. No other preprocessing is applied to the images.

Split details. The BOLD database does not contain truly undescribed species. To artificially create undescribed test classes, genera are chosen that have a minimum of three species, and 33% of those species are randomly chosen and set aside as undescribed species. These pseudo-undescribed species are referred to as unseen classes and described species as seen classes. For example, the genus Coelioxys has three species, and one of them (in our case C. *conoidea*) is randomly chosen as an undescribed species, leaving two as seen classes. This split leaves 243 unseen classes and 797 seen classes, where the training set does not include any images or DNA from these 243 classes. In order to create a validation set for unseen classes, in the same fashion, 33% of species are randomly chosen of genera that have at least three members from the 797 training classes. The remainder of the data is split by a

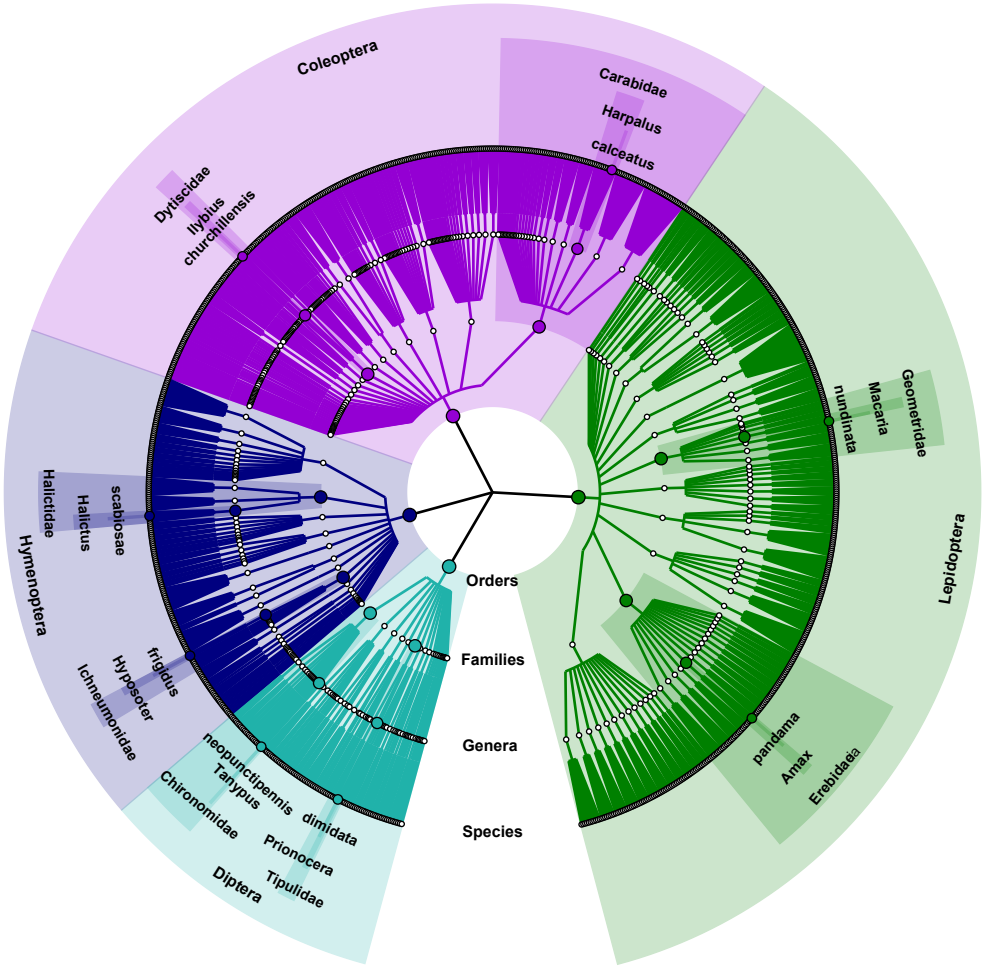


Figure 4.2. Phylogenetic tree of the 4 orders from the dataset. Two species are randomly chosen from each order with their full taxonomic hierarchy are illustrated.

70/30 ratio in a stratified fashion to obtain samples for training and test seen classes. Some of the insect classes have multiple images, each capturing a different view of the insect (for example, ventral and dorsal views), all insect classes with multiple images are restricted to the training set, leaving 27 of the seen classes with no available samples for testing. Test samples from seen and unseen classes summed up to 4,965 and 8,463 instances.

CNN Embeddings for DNA Barcodes. We trained the same CNN model from Section 3.3.3 to optimize vector representations of the DNA barcodes in the Euclidean space. The training procedure is the same with one exception: the raw DNA barcodes are used, that is, they are not aligned before feeding into the model.

4.2.2 Zero-shot Insect Classification

In our approach, we assume that there are species that are completely unknown (for example, a newly discovered species), and we introduce a framework that can identify insects at the lowest taxonomic level possible by jointly leveraging image and taxonomic information. More specifically, if an insect to be classified is a previously described species, the test sample would be classified as one of the species present in the training set. On the other hand, if the insect is undescribed and therefore not present in the training data, the taxonomic level identification would be to genus, providing clues that the insect is not a species in the current database. Thus, for undescribed insect species, the genus would be predicted, therefore indicating the database/training does not contain the species and it is likely an unknown species. This zero-shot classification approach not only significantly reduces the uncertainty surrounding traditional closed-set supervised algorithms (closed-set algorithms assume all possible classes/ species are present in the training data and therefore would misclassify all new/ undescribed insects into one of the known species), but also addresses problems with existing open-set frameworks where any undescribed species are designated as an outlier, thus no additional taxonomic level is being identified.

In traditional ZSL visual attributes [15], word vectors [40], text descriptors [25], and DNA barcodes (See Chapter 3) were previously used as class-based auxiliary information. Traditional ZSL can include an arbitrarily large number of unseen classes as long as each unseen class can be described by some form of auxiliary information. However, in our approach the number of unseen classes, i.e., undescribed species, that can be uniquely discovered is limited with the number of genera in the training dataset as our approach cannot differentiate between two undescribed species assigned to the same genus. Although this may look like a limitation of our approach, using class taxonomy to discover undescribed species is a more practicable goal because undescribed species are supposed to be the most infrequent, uncommon, and thus least known categories. They remain as undescribed because current dichotomous keys cannot uniquely define them. Therefore, the assumption that highly granular side information uniquely defining these species are available during test time is far from

being realistic and relying on taxonomic information to perform zero-shot classification is more achievable.

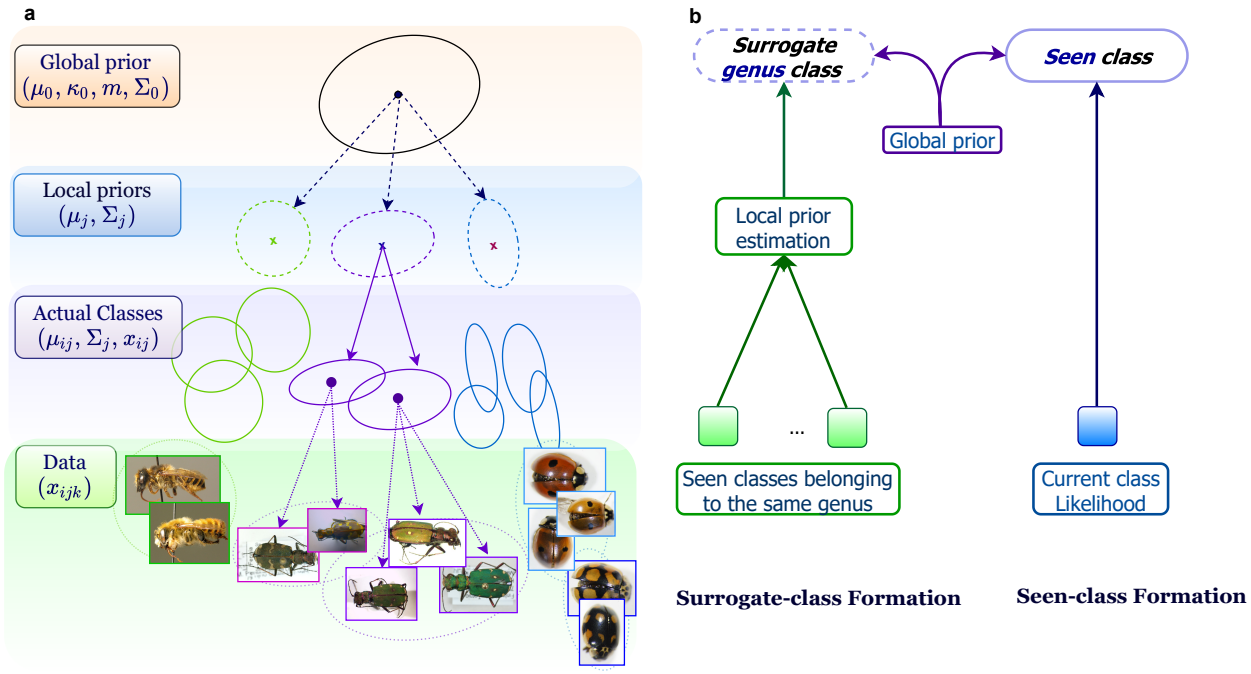


Figure 4.3. a. Generative model. Hyperparameters are defined in the Methods section. **b.** Class distribution formation for seen and surrogate genus classes.

4.2.3 Bayesian Model

Insect species have a predefined taxonomic hierarchy; species < genus < subfamily < family < order etc., although rich variety between these hierarchies carries valuable information, it is often overlooked when designing ML algorithms. A hierarchical Bayesian model is recently introduced in computer vision for zero-shot classification of object classes (See Chapters 2, 3). This model establishes a Bayesian hierarchy among object classes using visual attributes [65] or DNA [138] as auxiliary information. To identify both described and undescribed species a similar model is developed in this chapter. In this model, the class-specific side information is replaced with a predefined class hierarchy explicit in the taxonomical classification of biological organisms to define the class hierarchy. Therefore, the hyperparameter K is not required anymore as classes belonging to the same superclass,

Algorithm 2 Deriving PPD for seen and surrogate genus classes

Input: Training data

Output: PPD parameters for each seen class $(\bar{\boldsymbol{\mu}}_{\mathbf{jc}}, \bar{v}_{\mathrm{jc}}, \bar{\Sigma}_{\mathrm{jc}})$ and surrogate genus $(\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{v}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}})$

Set hyper-parameters: κ_0, κ_1, m, s
Compute $\boldsymbol{\mu_0}$ (mean of class means) and Σ_0 (mean of class covariances scaled by s)
for each seen class ω_{jc} **do**
    Calculate current class params: $\bar{\boldsymbol{x}}_{\mathbf{jc}}, n_{\mathrm{jc}}, S_{\mathrm{jc}}$
    Calculate $S_{\boldsymbol{\mu}}$ (Eq. A.34)
    Calculate PPD by combining *global prior* and *data driven likelihood*: $\bar{\boldsymbol{\mu}}_{\mathbf{jc}}, \bar{v}_{\mathrm{jc}}, \bar{\Sigma}_{\mathrm{jc}}$ (Eq 4.1)
end for
for each genus ω_{j} **do**
    for each seen class ω_{ji} belonging to the genus ω_{j} **do**
        Calculate class params: $\bar{\boldsymbol{x}}_{\mathrm{ji}}, n_{\mathrm{ji}}, S_{\mathrm{ji}}$
    end for
    Calculate intermediate term: $\tilde{\kappa}_{\mathrm{j}}$ (Eq. A.30)
    Calculate PPD parameters using only *local prior*: $\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{v}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}}$ (Eq 4.2)
end for

in our case genus, are grouped together to form the local priors. Blending the local and global priors with data likelihood, we again derive the posterior predictive distribution for each described species and surrogate genus. However, during the inference this time, a new insect sample (image or DNA) is classified to one of the described species that maximizes the posterior predictive likelihood or identified as a new species belonging to the surrogate genus class maximizing the posterior predictive likelihood.

PPD Derivation. PPD derivation follows the same procedure as described in section 2.3.2 except during implementation we did not utilized local priors for seen classes this time. PPDs for seen classes include the global prior and data likelihood (See Fig 4.3) and are derived in the form of a Student-t distribution as below,

$$P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\boldsymbol{c}}, S_c, \boldsymbol{\mu_0}, \kappa_0, \kappa_1) = T(\boldsymbol{x}|\bar{\boldsymbol{\mu}}_{\boldsymbol{c}}, \bar{\Sigma}_c, \bar{v}_c) \tag{4.1}$$

$$\bar{\boldsymbol{\mu}}_{\boldsymbol{c}} = \frac{n_c\bar{\boldsymbol{x}}_{\boldsymbol{c}} + \frac{\kappa_0\kappa_1}{\kappa_0+\kappa_1}\boldsymbol{\mu_0}}{n_c + \frac{\kappa_0\kappa_1}{\kappa_0+\kappa_1}}, \; \bar{v}_c = n_c + m - D + 1,$$

$$\bar{\Sigma}_c = \frac{(\Sigma_0 + S_c + S_\mu)(n_c + \frac{\kappa_0\kappa_1}{\kappa_0+\kappa_1} + 1)}{(n_c + \frac{\kappa_0\kappa_1}{\kappa_0+\kappa_1})\bar{v}_c}$$

where, $\bar{\boldsymbol{x}}_{\boldsymbol{c}}$, S_c and n_c are sample mean, scatter matrix and size of current seen class c. The index c in Equation (4.1) represents the current seen class, whose PPD is being derived.

Surrogate Class Formation. In this model, groupings among classes are based on local priors. Hence, once estimated from seen classes, local priors can be used to define surrogate classes for unseen classes during inference. We form a surrogate-class for each genus in our dataset by forming a local prior combining all seen classes from that genus (See Fig 4.3b). During the inference, test samples are classified based on class-conditional likelihoods evaluated for both seen and genus-level surrogate classes. The modified pseudocode is described in Algorithm 2.

PPDs for unseen classes also follow a Student-t distribution, thanks to conjugacy, given below,

$$
\begin{aligned}
&P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \boldsymbol{\mu_0}, \kappa_0, \kappa_1) = T(\boldsymbol{x}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}}, \bar{v}_{\mathrm{j}}) \qquad (4.2)\\
&\bar{\boldsymbol{\mu}}_{\mathrm{j}} = \frac{\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}}+\kappa_1)} \bar{\boldsymbol{x}}_{\mathrm{ji}} + \kappa_0 \boldsymbol{\mu_0}}{\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}}+\kappa_1)} + \kappa_0},\\
&\bar{v}_{\mathrm{j}} = \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} (n_{\mathrm{ji}} - 1) + m - D + 1, \quad \bar{\Sigma}_{\mathrm{j}} = \frac{(\tilde{\kappa}_{\mathrm{j}} + 1)}{\tilde{\kappa}_{\mathrm{j}} \bar{v}_{\mathrm{j}}} (\Sigma_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S_{\mathrm{ji}})
\end{aligned}
$$

where, $\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}$ and n_{ji} represent sample mean, scatter matrix and size of class i associated with surrogate-class j, respectively and $\tilde{\kappa}_{\mathrm{j}}$ is defined as in Eq. (A.30) from Appendix A.1.

It is worth to clarify the distinction between seen and surrogate class PPDs in the case of genera where they have only one species in the training data. The seen class distribution and surrogate genus class will have similar formulas but with 2 important distinctions. First, mean of the seen class PPD will have more weight on class sample mean whereas mean of the surrogate class will lean towards μ_0. Beside the common terms in location parameters, seen class PPDs have the term $\frac{\bar{\boldsymbol{x}}_{\boldsymbol{c}}}{1+\kappa_1/\kappa_0+\kappa_1/n_c}$ whereas surrogate class PPDs have μ_0 in replace of $\bar{\boldsymbol{x}}_{\boldsymbol{c}}$. Second, unlike surrogate class PPDs, seen class PPDs have additional term, S_μ, in their scale matrix.

4.2.4 Transductive Approach

The transductive approach leverages the unlabeled test data as well during the training process. We aim to learn a linear mapping from Image feature space to DNA feature space using Ridge regression. Figure 4.1 panel (c) outlines the transductive approach. Following the notation in the figure, $\tilde{X}$ and X represents the image and DNA embeddings, respectively. $V \in R^{d \times \tilde{d}}$ is the embedding from image space to DNA space we want to learn and λ is the regularization constant. Ridge regression with $Frobeneus$ norm has a well-known closed for solution given as, $V = X\tilde{X}^\top(\tilde{X}\tilde{X}^\top + \lambda I)^{-1}$. We leverage the learned mapping to augment auxiliary training data by embedding image features with labels into DNA feature space, mathematically $V\tilde{X}_{tr}$ and combine this data with DNA embeddings. The whole process takes two lines of a code and computational cost is infinitesimal compared to the model training time, thus this step comes as free. Nonetheless, we achieve remarkable 11% percent performance boost on unseen class accuracy while preserving seen class accuracy.

4.2.5 A Distance-based Bioinformatics Approach as a Baseline

For each described species, nucleotide sequences are aligned using training samples available for that species. Aligned sequences are then used to compute a consensus nucleotide sequence for each described species. Test samples are classified by evaluating Jukes-Cantor distance [139] between a test sequence and consensus sequences of described species. Test samples are assigned to the species with the minimum distance only if the minimum distance is smaller than a designated threshold. If the minimum distance is larger than this threshold then the test sample is treated as a sample of an undescribed species and assigned to the genus of the species with the minimum distance. Result of this approach is included in Table 4.1. The threshold is chosen by cross-validation.

4.3 Results

In this section, we first briefly discuss the predictive performance of the Convolutional Neural Network (CNN) model we developed to learn DNA embeddings. Then, Zero-shot

insect classification results are reported and, finally, the section is concluded with discussion and case studies. The core building block of our ZSL classification approach is a two-layer hierarchical Bayesian model defined over both described and undescribed species with two different types of priors: global and local. Global prior is shared by all species whereas local priors are only shared by species that are taxonomically similar and used as a surrogate class for undescribed species. Classification is performed by maximizing posterior predictive likelihood over both true and surrogate classes.

4.3.1 Predictive Accuracy of DNA Embeddings

Convolutional Neural Networks (CNN) are trained to optimize vector representations of the DNA barcodes in the Euclidean space (= embeddings). The CNN model yields impressive 99.44% accuracy on the holdout validation set that was created by reserving 20% of the training set. In the deployment of Neural Network models, it is also important to test that the model is generalizable to unseen classes/ species. To this end, we train a K-Nearest-Neighbor (KNN) classifier ($K = 1$) on randomly sampled 80% of the DNA embeddings of unseen classes (243 species) obtained from the CNN model and test on the remaining 20%. The simple KNN classifier renders 99.19% accuracy, proving the robustness of the CNN model to learning representation for undescribed species.

4.3.2 Zero-shot Bayesian Classification with Unknown/Undescribed Species (ZSBC)

No class information can be defined for undescribed species as these species are unrepresented in the training data. The only data available for training are images and DNA barcodes from described insect species (seen classes). The machine learning task at test time involves identifying insect classes originating from described species at the species level and those from undescribed species at the genus level. Several models are developed and tested. The Bayesian model is first trained and tested with CNN barcode embeddings (ZSBC-DNA) and then with ResNet101 [3] image embeddings (ZSBC-IMG). Additionally, we also develop a simple baseline using DNA sequences from the Matlab's bioinformatics tool.

Table 4.1. Zero-shot classification results. US and S represent unseen and seen class accuracy and H represents the harmonic mean of these two scores. For both seen and unseen classes, each class accuracy is calculated then the average of these class accuracies is reported. Note these results are for genus level classification for unseen classes. More precisely, during class accuracy calculations, different unseen classes belonging to the same genus are treated as the same class. Best results are displayed in bold and the second-best results are underlined. Tr, Ts_s and Ts_{us} represents train, test seen and test unseen data, respectively.

Methods	**US**	**S**	**H**
ZSBC-IMG	35.88	39.11	37.42
BioInformatics (DNA) baseline	71.85	**98.65**	83.16
ZSBC-DNA	73.39	96.15	83.24
ZSBC-DIC	77.26	97.26	86.25
ZSBC-DIL	**81.95**	<u>98.21</u>	**89.35**
ZSBC-DIT ($Tr + Ts_s + Ts_{us}$)	<u>81.39</u>	96.66	<u>88.37</u>
ZSBC-DIT ($Tr + 50\%Ts_{us}$)	79.94	96.66	87.53
ZSBC-DIT ($Tr + 25\%Ts_{us}$)	77.48	96.63	86.01

For combined approaches, the CNN barcode and ResNet image embeddings are investigated jointly to determine if image information can improve the accuracy of the DNA Barcode classifier in inductive as well as transductive settings. As a standard approach to fusing DNA and image information in the inductive setting, the DNA and image embeddings are concatenated into a single feature vector (ZSBC-DIC). Another approach in the inductive setting is the summation of normalized likelihood vectors generated by two Bayesian classifiers of CNN and ResNet embeddings (ZSBC-DIL). Finally, we develop a transductive approach that optimizes a linear mapping from image space to DNA sequence space by solving a ridge regression problem using ResNet and CNN embeddings of all available cases in test and train sets without using any class labels (ZSBC-DIT).

Table 4.1 reports the results from Zero-shot insect classification. As the number of classes increase, image classifiers alone cannot offer high performance. On the other hand, DNA data proves to be very informative for species classification. The bioinformatics baseline method using DNA alone is excellent at accurately classifying seen species (species that are present in the database) while achieving an accuracy of 72% on unseen species, a significant

reduction in comparison to using ZSBC-DIT. Although ZSBC-DNA yields a better unseen class accuracy, the performance on seen classes slightly drops.

In all three scenarios, combining image and DNA data helped the Bayesian classifier with a performance boost, in particular for unseen classes. Transductive and heuristic likelihood methods perform best with $>$ 88% harmonic mean and 81% unseen class accuracy. That being said, both inductive methods (ZSBC-DIC and ZSBC-DIL) have an inherent flaw: they require test samples to have an image with an accompanying DNA barcode. For the transductive method (ZSBC-DIT), only a fraction of test data contained an image and DNA pair, without using any labels, was enough to learn robust mapping and deliver a remarkable performance increase. The main information flow in learning the mapping in the transductive setting is coming from unseen classes. The last two rows of Table 4.1 display model performance while utilizing various fractions of image and DNA test data pairs from unseen classes for learning image to DNA embedding. Note that the model was not tuned for these configurations and employed the validation parameters used to produce ZSBC-DIT results. Using only 25% of image-DNA pairs from unseen classes to learn the Ridge regression improved the harmonic mean to 86%. This finding clearly displays how the abundance of unlabeled image and DNA pairs can be leveraged by the transductive method to significantly boost the DNA classifier performance. The transductive model (ZSBC-DIT) yields 96.66% overall accuracy of seen class classification with 4,827/4,965 correct classifications (See Table 4.2). For unseen classes, the accuracy declines, unsurprisingly, but is remarkably good for 3 of the 4 orders with >81% accuracy of assigning the unknown "species" to the correct genus. A large portion of unseen classes is misclassified in the order Diptera (Table 4.2). When examining the different family groups and their classification accuracy (Table 4.2), the Culicidae (the mosquitoes), Syrphidae (the hover flies), and Tipulidae (the crane flies) had the greatest amount of misclassifications. The number of possible species in the group does not account for the misclassifications, as species in Chironomidae were classified with 100% accuracy. With the Culicidae, 45/58 of the misclassifications are *Aedes vexans* records that classified to the *Culex* genus. When taking a random record and using the DNA sequence to BLASTn [140] in Genbank as a semi-independent test of the data, there were BOLD records that populate the hit list that are Culicinae sp., and therefore,

these records may be obstructing the classification due to the overlap in sequences. For the Syrphidae, 18 *Platycheirus neoperpallidus* records are misclassified to *Platycheirus clypeatus*. When random *P. neoperpallidus* records are aligned to other *Platycheirus* species, it is noted that there is a great deal of similarity with *P. quadratus*, a species not present in the training set, again, demonstrating the need for a more representative training dataset to ensure accuracy within certain groups. There is only one instance in which every single individual is misclassified, wherein 14 records of the Tipulidae, all belonging to a single species *Tipula coloradensis*, are completely misclassified. The majority of the misclassifications are to the same subfamily (Tipulinae), but misclassified to the *Nephrotoma* genus, and four of the 14 are misclassified to Syrphidae. What is remarkable with this dataset is that the training data contains three species of Tipula (*T. caliginosa*, *T. salicetorum*, *T. shirakii*). Sequence similarities are calculated between the three in the training set and *T. coloradensis*, and what is apparent very quickly is that *T. salicetorum* and *T. caliginosa* are closely related (interspecific sequence differences 97%), whereas the sequence similarity of *T. coloradensis* with either *T. salicetorum* or *T. caliginosa* is 88%. Further, *T. shirakii* is perhaps the most different, with 85% sequence similarities from the remainder of the Tipula species included in this analysis (data not shown). What this is indicative of is quite the vast amount of sequence variation that may exist in this genus.

4.4 Discussion

Deep learning methods are becoming more and more integrated into various fields and disciplines in the sciences. Here we present a novel method for classification of new insect species, with an eye on the future of identification through image analysis and character extraction for the entomology field explicitly, although this can be applied to any biological organism for which image and DNA data can be generated. This is the first attempt where zero-shot classification is done by integrating DNA information with image analysis on a comparatively larger number of classes (in this case, 1,040 species in four large orders). The use of image analysis alone, or DNA analysis alone, has had varying levels of success. DNA is generally viewed as strong support for new species if the sequence variation falls outside

the normal bounds of intraspecific variation. In some cases, the DNA barcode has been integral to differentiate between species that are morphologically indistinguishable, confirmed through additional nuclear DNA sequencing [141]. Image analysis alone has provided some gains in order to monitor (in real-time) insect species but suffers when background extraction is necessary. Furthermore, these methods are closed-set since the application is related to monitoring for existing species (for example, when pest management strategies are necessary [142], [143]). When using deep learning methods with images to identify seen classes of insects, accuracy gains reach 90 percent or greater [123]–[125], [144], in some cases, approaching or surpassing taxonomic specialist accuracy's [145]. However, all these methods are tested either on coarse-grained datasets or with a limited number of classes, generally less than 15 species. Furthermore, the lingering issue of identifying unseen classes and the inherent data imbalance continue to plague the ability of more efficient means of identifying new species, especially within the Insecta class, where the majority of the species continue to be unidentified and presents the most important advancement to the field of entomology, but more broadly, to better understanding ecosystems and their processes, of which insects likely play a major role [146].

The model trained on DNA embeddings (ZSBC-DNA) achieves a compelling 96.15% accuracy on seen classes where 670 out of 770 test classes are perfectly classified to their true species. The model performance dropped to 73.39% in a more challenging task of identifying unseen species and assigning them to their true genera. ZSBC-DNA completely misclassifies all samples of 24 unseen species (less than 10% of all unseen classes), yet it is worth noting that six of these classes are perfectly assigned to their true classes as the second-best option. Leveraging auxiliary image data, the transductive approach (ZSBC-DIT) significantly boosts the unseen class performance to 81.39% (an 11% increase) with a modest increase on the seen class accuracy over DNA alone (ZSBC-DNA). ZSBC-DIT classified 677 out of 770 seen classes with 100% accuracy. The model also partially recovers 14 of 24 completely missed unseen species under ZSBC-DNA model (see Fig 4.4), where nine out of 14 classes are recovered by more than 80%.

4.4.1 Striking Morphological Similarity Between Species Belonging to the Same Genus

As it is observed in Figure 4.4a, variation in some insects is nearly invisible to the human eye, especially if lacking specialized expertise, yet the models are able to extract these subtle differences from images and aid DNA embeddings to correctly classify these difficult cases. To illustrate, we present a simple challenge in Fig 4.4d where one sample from 4 different species belonging to *Agabus* genus is displayed. The task is to correctly match the images with the following species names: *A. sturmii*, *A. bipustulatus*, *A. uliginosus*, and *A. infuscatus*. The true order can be found in this footnote[1]. Out of four species, *A. infuscatus* was reserved as an unseen class. DNA classifier correctly classifies all test samples from three seen classes, however, it makes a few mistakes while assigning the samples of unseen class into its true genus, *Agabus*. The ZSBC-DIT model, on the other hand, correctly classifies with 100% accuracy all seen and unseen class test samples.

This observation also reveals that 658bp of DNA sequence (*cytochrome oxidase subunit I*) lacks the differentiation needed, yet image representation can highlight these features such as spotted in the *Lasioglossum* and *Sphecodes* cases (column 3 of Fig 4.4a). Both genera share similar DNA sequences and are members of the same tribe (Halictini), which makes it quite difficult to differentiate using DNA barcodes in the challenging generalized zero-shot classification setup. In the transductive approach (OSDB-DIT), these elusive morphological features are successfully transferred from image space to DNA space and fill the gap in the utility of DNA barcodes.

4.4.2 Effect of Image Quality and Background Noise on Model Performance

High-quality images are an integral part of any successful machine learning approach and heavily impact the model performance in computer vision tasks. It is well documented that due to cross-entropy loss they have been trained with, many state-of-the-art pretrained CNN models are sensitive to the presence of subtle noise such as Gaussian, background noises, or blurriness in the image [147]–[149]. The following interesting cases observed in our

[1]↑*A. infuscatus*, *A. sturmii*, *A. bipustulatus*, and *A. uliginosus*

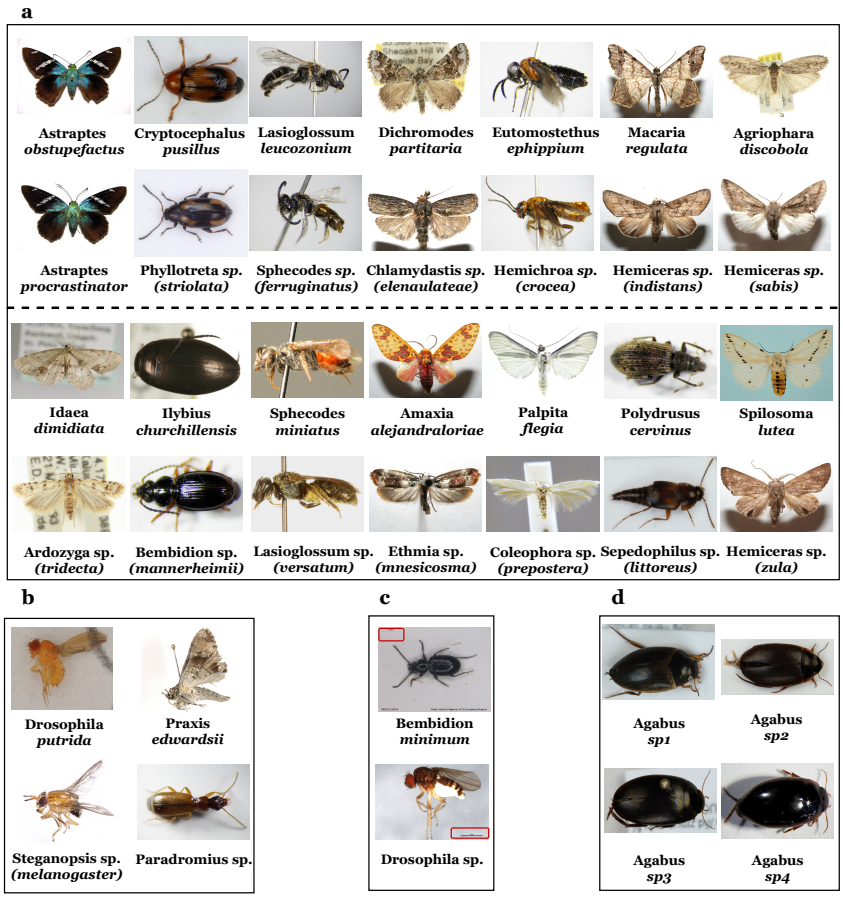


Figure 4.4. Discussion cases and phylogenetic tree. **a.** Unseen classes (14) that are completely missed by ZSBC-DNA but classified by ZSBC-DIT (some species partially and some fully). The first and third rows display the images that are covered by ZSBC-DIT and the second and fourth rows show samples for the corresponding classes that are misclassified under ZSBC-DNA. Names containing "sp." means that this is a Genus class and the image is from the species of name in parenthesis belonging to that genus. **b.** Misclassified cases due to image manipulation. **c.** Misclassified case due to background noise. **d.** Morphological resemblance between species belonging to the same genus.

experiments also verified these phenomena where a few isolated instances were misclassified to unrelated species under ZSBC-DIT classifier.

Cases in Figure 4.4b illustrate the vulnerability of the CNN models towards image manipulations. The first case is a test sample from the seen class *Drosophila putrida*, which is correctly classified by the DNA classifier, yet ZSBC-DIT misclassifies the sample to *Steganopsis* genus. Except for this case, all the cases from seen classes where the DNA model correctly

classified but DNA+IMAGE model failed are either misclassified to the true genus or to another species from the corresponding genus. In the light of this statistic, this particular case stands out as the ZSBC-DIT model misclassified this test sample (and only misclassified test case from *D. putrida*) to another genus. Inspecting the image features reveals that this figure is the only one exposed to image manipulation and was trimmed by Adobe Photoshop CS (this information can be accessed from image properties). In the same fashion, the test case from *P. ewardsii* species in Figure 4.4b is the only test sample that is misclassified by ZSBC-DIT, and also the only sample exposed to a modification from a software called CombineZP [150] (this information can be accessed from image properties). These subtle alterations are most of the time indistinguishable to humans yet can drastically alter the CNN model embeddings. Recent research suggests more robust image embeddings less sensitive to subtle alterations can be obtained using backbone architectures trained by self-supervised learning [147], [151].

Background information can sometimes dominate the relevant image features. The aberrant misclassification of the test sample from *Bembidion minimum* to a *Drosophila* genus (from a different order) is an example of this phenomenon (See Figure 4.4c). Many images from *Drosophila* genus have "1mm" text attached next to the species image to illustrate the scale, and that particular test sample (the only misclassified sample from *B. minimum*) has the same "1mm" text in the background.

4.5 Conclusion

In this study, we developed a novel framework utilizing our Bayesian classifier to facilitate the identification and discovery of insect species at scale. Insects account for a large portion of unknown biodiversity and our approach can pave the way to tackle this gigantic task. First time in the literature we proposed fusing two data modalities, namely image and DNA information, to tackle this problem and demonstrated its effectiveness on extended INSECT dataset with more than thousand classes. Unlike all the previous work, our model does not simply cast aside out-of-distribution samples by treating them as outliers but classifies them to meaningful groups exploiting the inherent taxonomic hierarchy. Our transductive

Bayesian classifier delivered 81% accuracy on identifying the correct genus of new species that have no image or DNA samples present in the training data, meanwhile classified known species with more than 96% accuracy.

Considering the transductive approach was built on regularized linear mapping, it appears there is a great potential to achieve better performance utilizing nonlinear and more sophisticated approaches like Generative Adversarial Networks [5] or Variational Autoencoders [152] to learn this mapping. Integrating GAN/ VAE would also allow training an end-to-end model by self-supervised learning that can potentially mitigate the shortcomings of the pretrained CNN models trained with cross-entropy loss.

In this proof of concept application, the focus was on new species discovery, wherein the subclasses were species, and the superclasses were genera. The Bayesian model can easily be extended to be trained on where genera/species are considered the subclasses and higher taxonomic levels are considered superclasses (e.g., family). Such a classifier will readily deal with missing/unobserved genera. That being said, a vanilla CNN with cross-entropy loss would not suffice to learn efficient embeddings for this problem. Although intra-species variation in DNA barcodes is very low, we observed that the inter-species variation is quite the opposite. Thus, a hierarchical loss that takes into account not just species information but also genus, family, and order information all at once would be necessary to train the CNN model for more robust embeddings.

Code and Data

The code can be accessed from github.com/sbadirli/Zero-shot-Insect-Discovery. The data is publicly available at http://dx.doi.org/10.7912/D2/27

Table 4.2. Seen and unseen class accuracy (from ZSBC-DIT model) by insect family that has five or more species. 'Summary' row reports the results from all families belonging to that order in our dataset.

		Seen Classes			Unseen Classes	
Order	**Family**	# training	# test samples	Accuracy	# test samples	Accuracy
Coleoptera	Brentidae	94	18	100.00%		
	Cantharindae	226	43	93.02%	77	94.81%
	Carabidae	1660	346	95.66%	128	95.31%
	Cerambycinae	210	43	100.00%		
	Chrysomelidae	564	114	99.12%	37	89.19%
	Coccinellidae	226	46	100.00%		
	Curculionidae	348	68	94.12%	55	96.36%
	Dytisicidae	146	30	100.00%	18	88.89%
	Elateridae	242	47	100.00%	12	100.00%
	Scarabaeidae	106	23	91.30%		
	Staphylinidae	714	150	92.67%	47	100.00%
	Tenebrionidae	186	24	100.00%		
Summary (C)	**37**	**5,680**	**1,143**	**95.80%**	**751**	**85.22%**
Diptera	Calliphoridae	190	35	100.00%	13	92.31%
	Chironomidae	464	96	97.92%	24	100.00%
	Culicidae	496	107	89.72%	58	22.41%
	Drosophilidae	392	85	84.71%	80	81.25%
	Muscidae	104	22	90.91%		
	Sciaridae	150	33	100.00%		
	Syrphidae	342	71	97.18%	45	60.00%
	Tipulidae	122	26	96.15%	14	0.00%
Summary (D)	**20**	**2,744**	**570**	**93.68%**	**273**	**61.17%**
Hymenoptera	Andrenidae	192	39	100.00%	53	79.25%
	Colletidae	190	32	100.00%	56	100.00%
	Crabronidae	312	66	100.00%	60	96.67%
	Eulophidae	226	47	100.00%	183	100.00%
	Halictidae	344	70	98.57%	113	80.53%
	Ichneumonidae	306	67	100.00%	12	100.00%
	Megachilidae	296	55	100.00%	28	53.57%
	Tenthredinidae	864	169	91.72%	261	66.28%
	Vespidae	106	22	100.00%	22	77.27%
Summary (H)	**19**	**3,282**	**660**	**97.27%**	**872**	**82.22%**
Lepidoptera	Coleophoridae	994	206	99.51%	170	82.35%
	Crambidae	1054	176	99.43%	482	87.14%
	Depressariidae	1836	269	100.00%	380	67.63%
	Erebidae	4288	464	97.20%	694	74.78%
	Gelechidae	268	59	96.61%	41	82.93%
	Geometridae	1170	230	96.96%	328	89.63%
	Hesperiidae	2294	14	85.71%	566	47.00%
	Noctuidae	3246	570	98.95%	525	82.10%
	Notodontidae	4068	257	100.00%	959	94.89%
	Nymphalidae	554	37	100.00%	166	84.94%
	Saturniidae	890	31	100.00%	111	99.10%
	Tortricidae	968	170	100.00%	144	96.53%
Summary (L)	**18**	**22,564**	**2,592**	**98.61%**	**6,567**	**81.01%**

5. CONCLUSION

Classifying fine-grained objects is a very challenging task especially with large number of classes. The inherent difficulty of this task is the lack of labeled data and costly process of obtaining these labels. A robust machine learning algorithm needs to not only classify well-represented object classes but also efficiently cope with the underrepresented classes. This problem is studied under the general open-set recognition framework where the presence of test samples that are not represented by any classes in training data is acknowledged and expected to be dealt by the algorithm.

5.1 Summary

In this thesis, we first approached this sophisticated problem from Zero-shot Learning perspective in which some sort of side information is provided to cognize these unrepresented classes. The presence of side information made the problem more manageable and led to more publicity for this challenging problem in the ML and computer vision communities. We developed a Bayesian hierarchical model for ZSL task based on the consideration that real-life fine-grained object classes emerge at different levels of abstraction and a two-layer Gaussian mixture model can take an advantage of this hierarchy. With the touch of the Bayesian approach, the proposed model (BZSL) offers several advantages over the other ZSL methods. With its closed form solution, BZSL has a much faster training time. With only 5 hyperparameters, it is easy to tune the model and also these parameters offer huge flexibility to model both coarse and fine-grained datasets. Finally, our model does not assume strong correlation between side information and image features which is the main reason for current state of the art ZSL models' poor performance when the side information is not based on visual attributes.

As the number of classes increases along with inter-class similarity, the complexity of obtaining characteristic visual attributes for unseen classes emerges as the major bottleneck for zero-shot classification. This problem becomes more evident in the zero-shot species classification with thousands of classes. Furthermore, due to lack of enough text descriptions of many species, it is also not possible to derive word vectors for them. To overcome this

challenge, we introduced DNA as side information for zero-shot species classification. We showed that a simple CNN model can extract information-rich feature vectors from these small fragments of mitochondrial DNA to serve as characteristic attributes for both seen and unseen species. Moreover, our experiments demonstrated that Bayesian model can better utilize DNA attributes to exploit the inherent class hierarchy among species and outperform state of the art ZSL methods. We also concluded that without an explicit association between DNA barcodes and image features, FGNs struggle to learn meaningful representation for unseen classes which leads to inferior performance compared to Bayesian model on both INSECT and CUB datasets.

In the final chapter of the thesis, encouraged with the discriminative power of DNA and flexibility of Bayesian model we took one step further to tackle the fine-grained species identification and discovery without relying on any side information for unseen classes. The results from our experiments led to three conclusions: (1) COI gene barcodes are more powerful than images for species classification although they are harder to obtain. For a successful fine-grained image classification with large number of classes, high resolution images are necessary and self-supervised learning might be necessary to overcome some limitations of pretrained Deep CNN models. (2) Combining image information with DNA can boost the performance by almost 10% for unknown species identification without reducing the seen class classification accuracy. (3) Just using image-DNA pairs and taxonomical order of seen classes, Bayesian model can correctly classify seen species with impressive 97% accuracy while identifying new species and associating them with their true genus with 81% accuracy.

5.2 Limitations

The proposed Bayesian classifier is specifically designed to leverage the implicit hierarchy present in fine-grained datasets. That is, within our Bayesian framework the local priors for unseen classes can be best estimated once similar seen classes are available in the dataset. This, in turn, would require having large number of seen classes to ensure local priors are robust and unique. Furthermore, our model relies on informative enough side information

to accurately build hierarchy and evaluate class similarities. Without effective auxiliary information or inherent class hierarchy, even having large number of classes cannot help. Nonetheless, this is not a problem unique to our model but rather an impediment in ZSL.

On the application side, we investigated the utility of DNA barcodes in ZSL as side information and in a more realistic species identification and discovery task. Although it is much easier to extract and process the mitochondrial DNA than the nucleic DNA, the procedure still requires experts in the domain to handle it. This challenges the use of DNA data for in-situ species identification and emphasizes the necessity of image data which is easier to obtain. That being said, doing fine-grained image recognition calls for higher quality images and very robust CNN models.

5.3 Future Directions

In this section, we discuss two new directions along which the work conducted within this thesis can be readily extended to and a problem that has been long overlooked in ZSL literature.

The focus of this thesis was on fine grained object classification particularly in the presence of limited labeled training data. In fact, we dealt with the most extreme case where not a single training example was available to train some classes. In practice, however, there might be some scenarios where this extreme case can be loosened. A closely related challenging problem to zero-shot learning is the few-shot learning where instead of side information some classes have one or only a few samples to train. This severe imbalance between classes poses a great challenge for ML algorithms. Our Bayesian model can be easily extended to operate in the few-shot learning domain. The feature representation of scarce samples can be used to build the hierarchy to form local priors for those imbalanced classes. Indeed, using directly image feature prototypes for building the hierarchy can help to better estimate local priors which would lead more accurate PPDs.

The second research direction we want to pursue is morphing our Bayesian model into a Hierarchical Dirichlet Process Gaussian Mixture Model and performing open-set recognition by local priors without using any side information. The main idea is to quantify the open

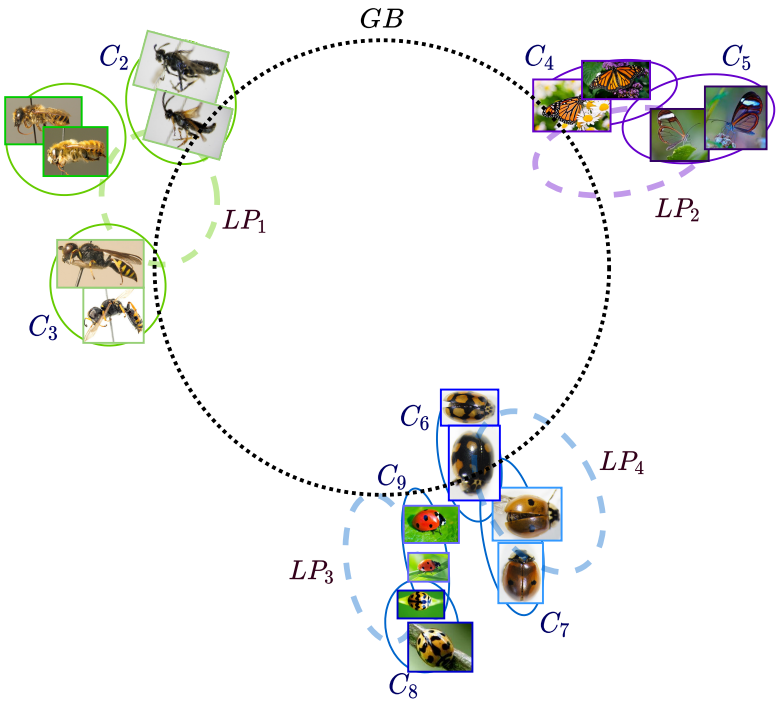


Figure 5.1. Open-set recognition with local priors. Beside minimizing the open-space risk, introducing local priors also helps to cognize test samples from unseen classes. For example, if a test sample gets classified to local prior 3, but does not belong to any seen classes associated with that local prior, then we may infer that we find a new ladybug species.

space risk by the hyperparameters of the Bayesian model and then minimize it by introducing local priors. This way we will not only identify the samples from UUCs but can also cognize them by the local priors they are assigned to. In this two-layer model, the top layer defines a global Dirichlet Process (DP) over local priors while the bottom layer defines local DPs over each group of class distributions. Under this setup, the open-set classification of a test sample can be performed by a two-stage classification. In the first stage, the posterior probability would be calculated to predict whether the sample belongs to one of the local priors or to a global prior in which case the test sample would be considered as an out-of-distribution sample associated with new local prior. For the former case, the classification would proceed to the second stage to identify either the sample belongs to one of the seen classes associated with that local prior or it originates from an unseen class associated with that local prior. The Figure 5.1 illustrates class formations in 2D for this model.

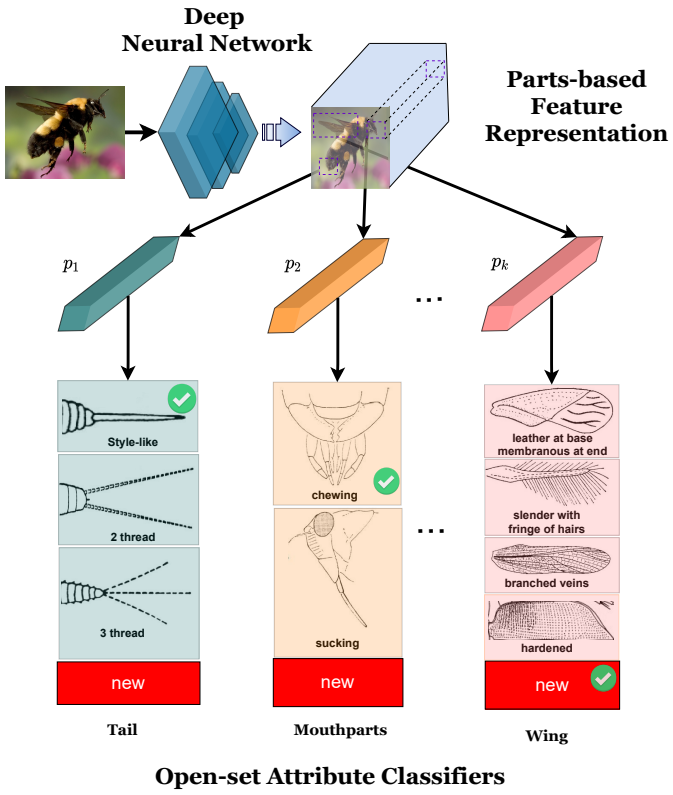


Figure 5.2. Filling in semantic gap in ZSL. p_i represents learned vector embeddings for body-part i.

The main motivation of ZSL can be related to human learning process, that is, identifying new categories by just knowing their semantic descriptions. Although the idea is fascinating, the current ZSL paradigm holds on a strong assumption: an exhaustive list of attributes can be obtained based only on seen classes and highly granular semantic descriptions are defined not just for seen classes but for unseen ones as well. Nevertheless, due to the power-law property of object categories in nature, unseen object categories are supposed to be the most infrequent, and thus least known categories. Thus, for a fine-grained object classification, it is not realistic to assume that semantic descriptions of unseen classes are well-defined, or even exist. A more practicable yet more challenging task would be to assume that unseen classes are partially unknown and predict their semantic descriptions as deviations from the patterns of semantic descriptions characterizing seen classes. To tackle this problem, an end-to-end deep learning model can be trained to identify locally discriminative regions of species [153], [154]. The semantic attributes defining seen classes will be grouped according

to these locally discriminative regions, i.e., body parts. For each group, an open-set classifier can be trained and during inference the classifier will identify semantic attributes of species instances from their images. As these classifiers are designed to perform in open-set setting, they will not only identify existing semantic attributes but also new attribute patterns that deviate from existing ones as determined by the classifier (See Figure 5.2).

REFERENCES

[1] Y. LeCun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 1989, pp. 396–404.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 4, pp. 2278–2324, 1998.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[5] I. Goodfellow, J. P. Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[6] J. Jumper, R. Evans, and A. e. a. Pritzel, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, pp. 583–589, 2021.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[8] W. J. Scheirer, A. R. Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *TPAMI*, vol. 35, 2013.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[10] 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.

[11] S. Thrun and T. M. Mitchell, "Lifelong robot learning," *Robotics and Autonomous Systems*, vol. 15, pp. 25–46, 1995.

[12] A. Pentina and C. H. Lampert, "A pac-bayesian bound for lifelong learning," in *ICML*, 2014.

[13] M. Yamada, L. Sigal, and Y. Chang, "Domain adaptation for structured regression," *International Journal of Computer Vision*, vol. 109, pp. 126–145, 2014.

[14] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32, pp. 53–69, 2015.

[15] C. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009.

[16] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *NeurIPS*, 2009.

[17] F. F. Li, R. Fergus, and P. Perona, "A bayesian approach to unsupervised one-shot learning of object categories," in *ICCV*, 2008.

[18] R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "One-shot learning with a hierarchical nonparametric bayesian model," in *JMLR workshop*, 2012.

[19] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 2317–2324, 2014.

[20] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *ECCV*, 2014.

[21] C. Geng, S. J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE TPAMI*, 2020.

[22] N. A. Ross, "Known knowns, known unknowns and unknown unknowns: A 2010 update on carotid artery disease," *Journal of the Royal Colleges of Surgeons of Edinburgh and Ireland*, vol. 8, pp. 79–86, 2010.

[23] A. R. Dhamija, M. Gunther, and T. Boult, "Reducing network agnostophobia," in *NeurIPS*, 2018.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013.

[25] M. Elhoseiny, B. Saleh, and A. Elgammal, "Write a classifier: Zeroshot learning using purely textual descriptions," in *ICCV*, 2013.

[26] S. Badirli, Z. Akata, G. Mohler, C. Picard, and M. Dundar, "Fine-grained zero-shot learning with dna as side information," in *NeurIPS*, 2021.

[27] D.Smolyak, K. Gray, S. Badirli, and G. Mohler, "Coupled igmm-gans with applications to anomaly detection in human mobility data," *ACM Transactions on Spatial Algorithms and Systems*, vol. 6, 2018.

[28] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, pp. 1443–1471, 2001.

[29] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *Journal of machine Learning research*, vol. 2, pp. 139–154, 2001.

[30] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, pp. 45–66, 2004.

[31] H. Jin, Q. Liu, and H. Lu, "Face detection using one-class-based support vectors," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[32] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler, "Kernel null space methods for novelty detection," in *CVPR*, 2013.

[33] F. Li and H. Wechsler, "Open set face recognition using transduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1686–1697, 2005.

[34] Y. C. F. Wang and D. Casasent, "A support vector hierarchical method for multi-class classification and rejection," in *In International Joint Conference on Neural Networks*, 2009.

[35] F. Akova, D. Hirleman, A. K. Bhunia, B. Raiwa, and M. Dundar, "Non-exhaustive learning for bacteria detection," in *International Conference on Network-Based Information Systems*, 2009.

[36] M. Dundar, F. Akova, Y. Qi, and B. Raiwa, "Bayesian non-exhaustive learning for online discovery and modeling of emerging classes," in *ICML*, 2012.

[37] B. Heflin, W. Scheirer, and T. E. Boult, "Detecting and classifying scars, marks, and tattoos found in the wild," in *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems*, 2012.

[38] D. A. Pritsos and E. Stamatatos, "Open-set classification for automated genre identification," in *European Conference on Advances in Information Retrieval*, 2013.

[39] S. Badirli, M. B. Ton, A. Gungor, and M. Dundar, "Open set authorship attribution toward demystifying victorian periodicals," in *ICDAR*, 2021.

[40] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *NeurIPS*, 2013.

[41] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *TPAMI*, 2016.

[42] B. Romera-Paredes and P. H. Torr, "An embarrassingly simple approach to zero-shot learning," in *ICML*, 2015.

[43] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *ICCV*, 2015.

[44] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018.

[45] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *CVPR*, 2015.

[46] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean, "Zero-shot learning by convex combination of semantic embeddings," in *ICLR*, 2014.

[47] S.Changpinyo, W.-L.Chao, B.Gong, and F.Sha, "Synthesized classifiers for zero-shot learning," in *CVPR*, 2016.

[48] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *CVPR*, 2017.

[49] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto, "Ridge regression, hubness, and zero-shot learning," in *ECML/PKDD*, 2015.

[50] J. Li, M. Jin, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," in *CVPR*, 2019.

[51] L. Chen, H. Zhang, J. Xiao, W. Liu, and S. Chang, "Zero-shot visual recognition using semantics-preserving adversarial embedding networks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1043–1052, 2018.

[52] R. Felix, V. Kumar, I. Reid, and G. Carneiro, "Multi-modal cycle-consistent generalized zero-shot learning," in *ECCV*, 2018.

[53] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *CVPR*, 2018.

[54] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1004–1013, 2018.

[55] Y. Zhu, M. Elhoseiny, B. Liu, and A. Elgammal, "Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning," *ICCV*, 2019.

[56] M. R. Vyas, H. Venkateswara, and S. Panchanathan, "Leveraging seen and unseen semantic relationships for generative zero-shot learning," in *ECCV*, 2020.

[57] A. Mishra, M. K. Reddy, A. Mittal, and H. Murthy, "A generative model for zero shot learning using conditional variational autoencoders," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2269–22 698, 2018.

[58] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrel, and Z. Akata, "Generalized zero- and few-shot learning via aligned variational autoencoders," in *CVPR*, 2019.

[59] G. Arora, V. Verma, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4281–4289, 2018.

[60] M. Bucher, S. Herbin, and F. Jurie, "Generating visual representations for zero-shot classification," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2666–2673, 2017.

[61] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *CVPR*, 2016.

[62] R. Qiao, L. L. N. C. Shen, and A. van den Hengel, "Less is more: Zero-shot learning from online textual documents with noise suppression," in *CVPR*, 2016.

[63] J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov, "Predicting deep zero-shot convolutional neural networks using textual descriptions," *ICCV*, 2015.

[64] G. A. Miller, "Wordnet: A lexical database for english," *CACM*, vol. 38, 1995.

[65] S. Badirli, Z. Akata, and M. Dundar, "Bayesian zero-shot learning," in *European Conference on Computer Vision Workshops*, Springer, Cham, 2020, pp. 687–703.

[66] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, *Caltech-ucsd birds 200*, Caltech, Technical Report, CNS-TR- 2010-001, 2010.

[67] Y. Xian, C. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly," *TPAMI*, 2018.

[68] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *CVPR*, 2009.

[69] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *NeurIPS*, 2013.

[70] M. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *ICCVGI*, 2008.

[71] G. Patterson and J. Hay, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *CVPR*, 2012.

[72] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *CVPR*, 2017.

[73] T. Mukherjee and T. Hospedales, "Gaussian visual-linguistic embedding for zero-shot recognition," in *EMNLP*, 2016.

[74] V. K. Verm and P. Rai, "A simple exponential family framework for zero-shot learning," in *ECML*, 2017.

[75] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *CVPR*, 2018.

[76] S. Kim and P. Smyth, "Hierarchical dirichlet processes with random effects," in *NeurIPS*, 2007.

[77] M. Dundar, F. Akova, H. Z. Yerebakan, and B. Rajwa, "A non-parametric bayesian model for joint cell clustering and cluster matching: Identification of anomalous sample phenotypes with random effects," *BMC bioinformatics*, vol. 15, no. 1, p. 314, 2014.

[78] M. Dundar, H. Z. Yerebakan, and B. Rajwa, "Batch discovery of recurring rare classes toward identifying anomalous samples," in *SIGKDD*, ACM, 2014.

[79] H. Z. Yerebakan, B. Rajwa, and M. Dundar, "The infinite mixture of infinite gaussian mixtures," in *NeurIPS*, 2014.

[80] C. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *TPAMI*, vol. 36, no. 3, pp. 453–465, 2013.

[81] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero shot learning a comprehensive evaluation of the good, the bad and the ugly," in *CVPR*, 2017.

[82] H. Jiang, R. Wang, S. Shan, and X. Chen, "Transferable contrastive network for generalized zero-shot learning," in *ICCV*, 2019.

[83] S. Liu, M. Long, J. Wang, and M. I. Jordan, "Generalized zero-shot learning with deep calibration network," in *NeurIPS*, 2018.

[84] W. J. Scheirer and T. E. Boult, "Statistical methods for open set recognition," in *CVPR Tutorial*, 2016.

[85] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. Radke, and O. Camps, "Towards visually explaining variational autoencoders," in *CVPR*, 2020.

[86] D. H. Lunt, D. X. ZHang, J. M. Szymura, and G. M. Hewitt, "The insect cytochrome oxidase i gene: Evolutionary patterns and conserved primers for phylogenetic studies," *PLoS biology*, vol. 5, 1996.

[87] P. D. Hebert, E. H. Penton, J. M. Burns, D. H. Janzen, and W. Hallwachs, "Ten species in one: Dna barcoding reveals cryptic species in the neotropical skipper butterfly astraptes fulgerator," *PNAS*, vol. 101, pp. 14 812–14 817, 2004.

[88] N. E. Stork, "How many species of insects and other terrestrial arthropods are there on earth?" *Annual review of entomology*, vol. 63, pp. 31–45, 2018.

[89] S. Ratnasingham and P. D. Hebert, "Bold: The barcode of life data system (http://www. barcodinglife. org)," *Molecular ecology notes*, vol. 7, no. 3, pp. 355–364, 2007.

[90] S. Ratnasingham and P. D. Hebert, "A dna-based registry for all animal species: The barcode index number (bin) system," *PloS one*, vol. 8, no. 7, e66213, 2013.

[91] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, and J. Han, "From zero-shot learning to conventional supervised classification: Unseen visual data synthesis," in *CVPR*, 2017.

[92] Y. Guo, G. Ding, J. Han, and Y. Gao, "Synthesizing samples for zero-shot learning," in *IJCAI*, 2017.

[93] M. R. Vyas, H. Venkateswara, and S. Panchanathan, "Leveraging seen and unseen semantic relationships for generative zero-shot learning," in *ECCV*, 2020.

[94] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[95] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Cvae-gan: Fine-grained image generation through asymmetric training," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2745–2754.

[96] Y. L. Cacheux, H. L. Borgne, and M. Crucianu, "Modeling inter and intra-class relations in the triplet loss for zero-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 333–10 342.

[97] H. Jiang, R. Wang, S. Shan, and X. Chen, "Transferable contrastive network for generalized zero-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9765–9774.

[98] M. R. Vyas, H. Venkateswara, and S. Panchanathan, "Leveraging seen and unseen semantic relationships for generative zero-shot learning," in *European Conference on Computer Vision*, Springer, 2020, pp. 70–86.

[99] J. Liu, Z. Zhang, and G. Yang, "Cross-class generative network for zero-shot learning," *Information Sciences*, vol. 555, pp. 147–163, 2021.

[100] J. Pennington, R. Socher, and C. D. Manning., "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[101] N. Karessli, Z. Akata, B. Schiele, and A. Bulling, "Gaze embeddings for zero-shot image classification," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6412–6421, 2017.

[102] N. G. Nguyen, V. A. Tran, D. L. Ngo, D. Phan, F. R. Lumbanraja, M. R. Faisal, B. Abapihi, M. Kubo, and K. Satou, "Dna sequence classification by convolutional neural network," *Journal of Biomedical Science and Engineering*, vol. 9, 2016.

[103] P. Ng, "Dna2vec: Consistent vector representations of variable-length k-mers," *ArXiv*, vol. abs/1701.06279, 2017.

[104] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "Dnabert: Pre-trained bidirectional encoder representations from transformers model for dna-language in genome," *Bioinformatics*, vol. 37, 2021.

[105] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[106] F. Zhang and G. Shi, "Co-representation network for generalized zero-shot learning," in *ICML*, 2019.

[107] S. Badirli, C. Picard, G. Mohler, Z. Akata, and M. Dundar, *Classifying the unknown: Identification of insects by deep zero-shot bayesian learning*, https://doi.org/10.21203/rs.3.rs-1099185/v1, 2021.

[108] S. Badirli, C. Picard, and M. Dundar, "Zero-shot insect identification and discovery (abstract)," in *International Congress on Conservation Biology (Oral Talk)*, 2021.

[109] F. Keesing, R. D. Holt, and R. S. Ostfeld, "Effects of species diversity on disease risk," *Ecology letters*, vol. 9, no. 4, pp. 485–498, 2006.

[110] M. Tulp and L. Bohlin, "Functional versus chemical diversity: Is biodiversity important for drug discovery?" *Trends in pharmacological sciences*, vol. 23, no. 5, pp. 225–231, 2002.

[111] J. Schmidhuber and F. N. Tubiello, "Global food security under climate change," *Proceedings of the National Academy of Sciences*, vol. 104, no. 50, pp. 19 703–19 708, 2007.

[112] G. M. Mace, K. Norris, and A. H. Fitter, "Biodiversity and ecosystem services: A multilayered relationship," *Trends in ecology & evolution*, vol. 27, no. 1, pp. 19–26, 2012.

[113] D. Pilling, J. Bélanger, and I. Hoffmann, "Declining biodiversity for food and agriculture needs urgent global action," *Nature Food*, vol. 1, no. 3, pp. 144–147, 2020.

[114] W. R. Erdelen, "Shaping the fate of life on earth: The post-2020 global biodiversity framework," *Global Policy*, vol. 11, no. 3, pp. 347–359, 2020.

[115] M. J. Costello, R. M. May, and N. E. Stork, "Can we name earth's species before they go extinct?" *Science*, vol. 339, no. 6118, pp. 413–416, 2013.

[116] N. E. Stork, "How many species of insects and other terrestrial arthropods are there on earth?" *Annual Review of Entomology*, vol. 63, pp. 32–45, 2018.

[117] M. Buck, N. E. Woodley, A. Borkent, D. M. Wood, T. Pape, J. Vockeroth, V. Michelsen, and S. Marshall, "Key to diptera families-adults," *Manual of Central American Diptera*, vol. 1, pp. 95–156, 2009.

[118] P. D. Hebert, A. Cywinska, S. L. Ball, and J. R. Dewaard, "Biological identifications through dna barcodes," *Proc. R. Soc. Lond. B Biol. Sci.*, vol. 270, pp. 313–321, 2003.

[119] J. M. Burns, D. H. Janzen, M. Hajibabaei, W. Hallwachs, and P. D. Hebert, "Dna barcodes and cryptic species of skipper butterflies in the genus perichares in area de conservacion guanacaste, costa rica," *PNAS*, vol. 105, pp. 6350–6355, 2008.

[120] M. S. Lee, "A worrying systematic decline," *Trends in Ecology & Evolution*, vol. 15, no. 8, p. 346, 2000.

[121] G. Hopkins and R. P. Freckleton, "Declines in the numbers of amateur and professional taxonomists: Implications for conservation," *Animal Conservation*, vol. 5, no. 3, pp. 245–249, 2002.

[122] M. C. ORR, J. S. ASCHER, M. BAI, D. CHESTERS, and C.-D. ZHU, "Three questions: How can taxonomists survive and thrive worldwide?" *Megataxa*, vol. 1, no. 1, pp. 19–27, 2020.

[123] J. Raitoharju and K. Meissner, "On confidences and their use in (semi-)automatic multi-image taxa identification," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019.

[124] M. Valan, K. Makónyi, A. Maki, D. Vondráček, and F. Ronquist, "Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks," *Systematic Biology*, vol. 68, pp. 876–895, 2019.

[125] D. Milošević, A. Milosavljevic, B. Predic, A. Medeiros, D. Savić-Zdravković, M. S. Piperac, T. Kostić, F. Spasić, and F. Leese, "Application of deep learning in aquatic bioassessment: Towards automated identification of non-biting midges.," *The Science of the total environment*, vol. 711, pp. 135–160, 2020.

[126] Y. Sun, X. Liu, M. Yuan, L. Ren, J. Wang, and Z. Chen, "Automatic in-trap pest detection using deep learning for pheromone-based dendroctonus valens monitoring," *Biosystems Engineering*, vol. 176, pp. 140–150, 2018.

[127] W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management," *Computers and Electronics in Agriculture*, vol. 123, pp. 17–28, 2016.

[128] B. P. H. et al., "Digitization and the future of natural history collections," *Bioscience*, vol. 70, pp. 243–251, 2020.

[129] E. K. Meineke, C. Tomasi, and K. M. P. S. Yuan, "Applying machine learning to investigate long-term insect-plant interactions preserved on digitized herbarium specimens," *Applications in Plant Sciences*, vol. 8, 2020.

[130] M. Mayo and A. T. Watson, "Automatic species identification of live moths," *Knowledge-Based Systems*, vol. 20, pp. 195–202, 2007.

[131] J. Wang, C. Lin, L. Ji, and A. Liang, "Automatic species identification of live moths," *Knowledge-Based Systems*, vol. 33, pp. 102–110, 2012.

[132] D. T. Tran, T. T. Høye, M. Gabbouj, and A. Iosifidis, "Automatic flower and visitor detection system," in *European Signal Processing Conference (Eusipco)*, 2018.

[133] T. T. Høye, J. Ärje, K. Bjerge, O. L. Hansen, A. Iosifidis, F. Leese, H. M. Mann, K. Meissner, C. Melvad, and J. Raitoharju, "Deep learning and computer vision will transform entomology," *Proceedings of the National Academy of Sciences*, vol. 118, no. 2, 2021.

[134] L. Pegoraro, O. Hidalgo, I. J. Leitch, J. Pellicer, and S. E. Barlow, "Automated video monitoring of insect pollinators in the field," *Emerging Topics in Life Sciences*, vol. 4, pp. 87–97, 2020.

[135] A. Bendale and T. E. Boult, "Towards open set deep networks," in *CVPR*, 2016.

[136] P. Perera and V. M. Patel, "Deep transfer learning for multiple class novelty detection," in *CVPR*, 2019.

[137] R. Agarwala and et al., "Database resources of the national center for biotechnology information," *Nucleic Acids Research*, vol. 46, pp. D8–D13, 2018.

[138] S. Badirli, Z. Akata, G. Mohler, C. J. Picard, and M. Dundar, "Fine-grained zero-shot learning with dna as side information," in *NeurIPS*, 2021.

[139] T. H. Jukes, C. R. Cantor, *et al.*, "Evolution of protein molecules," *Mammalian protein metabolism*, vol. 3, pp. 21–132, 1969.

[140] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, *Blastn (from ncbi)*, https://blast.ncbi.nlm.nih.gov/Blast.cgi, 2021.

[141] D. H. Janzen, J. M. Burns, Q. Cong, W. Hallwachs, T. Dapkey, R. Manjunath, M. Hajibabaei, P. D. Hebert, and N. V. Grishin, "Nuclear genomes distinguish cryptic species suggested by their dna barcodes and ecology," *Proceedings of the National Academy of Sciences*, vol. 114, no. 31, pp. 8313–8318, 2017.

[142] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8787–8796.

[143] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8769–8778.

[144] F. Visalli, T. Bonacci, and N. A. Borghese, "Insects image classification through deep convolutional neural networks," in *Progresses in Artificial Intelligence and Neural Systems*, Springer, 2021, pp. 217–228.

[145] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[146] L. H. Yang and C. Gratton, "Insects as drivers of ecosystem processes," *Current Opinion in Insect Science*, vol. 2, pp. 26–32, 2014.

[147] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *NeurIPS*, 2020.

[148] G. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," in *NeurIPS*, 2018.

[149] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[150] Informer Technologies, Inc, *Combinezp*, https://combinezp.software.informer.com/, 2021.

[151] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[152] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2014.

[153] Y. Zhu, J. Xie, Z. Tang, X. Peng, and A. Elgammal, "Semantic-guided multi-attention localization for zero-shot learning," in *NeurIPS*, 2019.

[154] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata, "Attribute prototype network for zero-shot learning," in *NeurIPS*, 2020.

[155] S. Balaban, C. Li, and M. Balaban, *Deep Learning GPU Benchmarks - Tesla V100 vs RTX 2080 Ti vs GTX 1080 Ti vs Titan V*, https://lambdalabs.com/blog/best-gpu-tensorflow-2080-ti-vs-v100-vs-titan-v-vs-1080-ti-benchmark/, [Online; accessed 31-May-2021], 2018.

[156] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *ICML*, 2013.

A. BZSL APPENDICES

A.1 Posterior Predictive Distribution (PPD) Derivation

The proposed model (BZSL) assumes Gaussian data model and Normal-Inverse Wishart (NIW) prior on class parameters. Derivations for each meta class (seen and unseen) are treated independently as our model imposes independence on data generation conditioned on meta-class parameters and global hyperparameters. By preserving the conjugacy, the assumption of common covariance among classes sharing the same meta-class further simplified the derivations. The order of derivations follows the same sequence from the main text. Please refer to the Table A.1 for all variables and parameters used in the calculations.

Table A.1. The notation used in the derivation of PPD.

Parameters	**Description**
D	image feature space dimension
d	superscript refering to the d^{th} component of a variable
j	meta-class index
i	actual class index
k	image index
c	index of current class
t_{i}	meta-class indicator for class i
t_k	class indicator for data point k
K	# of neighbors of current class in meta-class
$\boldsymbol{\mu}_{\mathrm{j}}$	mean of meta-class j
Σ_{j}	covariance of meta-class j
$\boldsymbol{\mu}_{\mathrm{ji}}$	mean of class i of meta-class j
$\bar{\boldsymbol{x}}_{\mathrm{ji}}$	sample mean of class i of meta-class j
S_{ji}	scatter matrix of class i of meta-class j
n_{ji}	size of class i of meta-class j
$\boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}}$	data point k from class i of meta-class j

Sufficient Statistics

As Gaussian distribution requires only mean and covariance to be uniquely identified, hence data in classes can be summarized by their sufficient statistics; sample means and covariances.

$$
\begin{aligned}
P(\boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}}|\boldsymbol{\mu}_{\mathrm{ji}}, \Sigma_{\mathrm{j}}) &\sim N(\boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}}|\boldsymbol{\mu}_{\mathrm{ji}}, \Sigma_{\mathrm{j}}) && \text{(A.1)}\\
\bar{\boldsymbol{x}}_{\mathrm{ji}} &= \frac{1}{n_{\mathrm{ji}}} \sum_{k:t_k=\mathrm{i}} \boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}} && \text{(A.2)}\\
\bar{\boldsymbol{x}}_{\mathrm{ji}} &\sim N(\boldsymbol{\mu}_{\mathrm{ji}}, \Sigma_{\mathrm{j}} n_{\mathrm{ji}}^{-1}) && \text{(A.3)}\\
S_{\mathrm{ji}} &= \frac{1}{n_{\mathrm{ji}}} \sum_{k:t_k=\mathrm{i}} (\boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}} - \bar{\boldsymbol{x}}_{\mathrm{ji}})(\boldsymbol{x}_{\mathrm{ji}\boldsymbol{k}} - \bar{\boldsymbol{x}}_{\mathrm{ji}})^T && \text{(A.4)}\\
(n_{\mathrm{ji}} - 1)S_{\mathrm{ji}} &\sim W(\Sigma_{\mathrm{j}}, n_{\mathrm{ji}} - 1) && \text{(A.5)}
\end{aligned}
$$

(3) follows from eq. (2) and independence assumption given meta-class parameters. (5) is a very definition of Wishart distribution as S_{ji} is scatter matrix of class i from meta-class j.

Step 1: Marginal Likelihood

The class sample means $\bar{\boldsymbol{x}}_{\mathrm{ji}}{}'s$ are connected to their meta class (j) by integrating out the intermediate class parameter $\boldsymbol{\mu}_{\mathrm{ji}}$. Note that all three parameters $(\bar{\boldsymbol{x}}_{\mathrm{ji}}, \boldsymbol{\mu}_{\mathrm{ji}}, \boldsymbol{\mu}_{\mathrm{j}})$ are Normally distributed and terms depend on meta-class covariance (Σ_{j}) are treated constant for this derivation.

$$
\begin{aligned}
&P(\bar{\boldsymbol{x}}_{\mathrm{ji}}|\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1) = \int P(\bar{\boldsymbol{x}}_{\mathrm{ji}}|\boldsymbol{\mu}_{\mathrm{ji}}, n_{\mathrm{ji}}, \Sigma_{\mathrm{j}}) P(\boldsymbol{\mu}_{\mathrm{ji}}|\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1) d\boldsymbol{\mu}_{\mathrm{ji}} && \text{(A.6)}\\
&= \int N(\bar{\boldsymbol{x}}_{\mathrm{ji}}|\boldsymbol{\mu}_{\mathrm{ji}}, \Sigma_{\mathrm{j}} n_{\mathrm{ji}}^{-1}) N(\boldsymbol{\mu}_{\mathrm{ji}}\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}} \kappa_1^{-1}) && \text{(A.7)}\\
&= \int (2\boldsymbol{\pi})^{-\frac{d}{2}} |\Sigma_{\mathrm{j}}/n_{\mathrm{ji}}|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\bar{\boldsymbol{x}}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{ji}})^T (\Sigma_{\mathrm{j}}/n_{\mathrm{ji}})^{-1} (\bar{\boldsymbol{x}}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{ji}})) && \text{(A.8)}\\
&* (2\boldsymbol{\pi})^{-\frac{d}{2}} |\Sigma_{\mathrm{j}}/\kappa_1|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\boldsymbol{\mu}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{j}})^T (\Sigma_{\mathrm{j}}/\kappa_1)^{-1} (\boldsymbol{\mu}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{j}})) d\boldsymbol{\mu}_{\mathrm{ji}}\\
&= \int C_1 C_2 exp(-\frac{1}{2}(\boldsymbol{\mu}_{\mathrm{ji}} - \frac{\kappa_1 \boldsymbol{\mu}_{\mathrm{j}} + n_{\mathrm{ji}} \bar{\boldsymbol{x}}_{\mathrm{ji}}}{\kappa_1 + n_{\mathrm{ji}}})^T (\frac{\Sigma_{\mathrm{j}}}{n_{\mathrm{ji}} + \kappa_1})^{-1} (\boldsymbol{\mu}_{\mathrm{ji}} - \frac{\kappa_1 \boldsymbol{\mu}_{\mathrm{j}} + n_{\mathrm{ji}} \bar{\boldsymbol{x}}_{\mathrm{ji}}}{\kappa_1 + n_{\mathrm{ji}}}) + C_3) d\boldsymbol{\mu}_{\mathrm{ji}} && \text{(A.9)}
\end{aligned}
$$

$$= C_1 C_2 \exp(C_3) \int exp(-\frac{1}{2}(\boldsymbol{\mu}_{\mathrm{ji}} - \frac{\kappa_1 \boldsymbol{\mu}_{\mathrm{j}} + n_{\mathrm{ji}} \bar{\boldsymbol{x}}_{\mathrm{ji}}}{\kappa_1 + n_{\mathrm{ji}}})^T (\frac{\Sigma_{\mathrm{j}}}{n_{\mathrm{ji}} + \kappa_1})^{-1} (\boldsymbol{\mu}_{\mathrm{ji}} - \frac{\kappa_1 \boldsymbol{\mu}_{\mathrm{j}} + n_{\mathrm{ji}} \bar{\boldsymbol{x}}_{\mathrm{ji}}}{\kappa_1 + n_{\mathrm{ji}}}) d\boldsymbol{\mu}_{\mathrm{ji}} \tag{A.10}$$

$$P(\bar{\boldsymbol{x}}_{\mathrm{ji}} | \boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1) = C_1 C_2 \exp(C_3) (2\boldsymbol{\pi})^{\frac{d}{2}} |\frac{\Sigma_{\mathrm{j}}}{\kappa_1 + n_{\mathrm{ji}}}|^{\frac{1}{2}} \tag{A.11}$$

$$C_1 = (2\boldsymbol{\pi})^{-\frac{d}{2}} |\Sigma_{\mathrm{j}} / n_{\mathrm{ji}}|^{-\frac{1}{2}} \tag{A.12}$$

$$C_2 = (2\boldsymbol{\pi})^{-\frac{d}{2}} |\Sigma_{\mathrm{j}} / \kappa_1|^{-\frac{1}{2}} \tag{A.13}$$

$$C_3 = -\frac{1}{2}(\bar{\boldsymbol{x}}_{\mathrm{ji}}{}^T (\Sigma_{\mathrm{j}} / n_{\mathrm{ji}})^{-1} \bar{\boldsymbol{x}}_{\mathrm{ji}} + \boldsymbol{\mu}_{\mathrm{j}}{}^T (\Sigma_{\mathrm{j}} / \kappa_1)^{-1} \boldsymbol{\mu}_{\mathrm{j}} - \frac{\kappa_1 \boldsymbol{\mu}_{\mathrm{j}} + n_{\mathrm{ji}} \bar{\boldsymbol{x}}_{\mathrm{ji}}}{\kappa_1 + n_{\mathrm{ji}}}{}^T (\frac{\Sigma_{\mathrm{j}}}{n_{\mathrm{ji}} + \kappa_1})^{-1} \frac{\kappa_1 \boldsymbol{\mu}_{\mathrm{j}} + n_{\mathrm{ji}} \bar{\boldsymbol{x}}_{\mathrm{ji}}}{\kappa_1 + n_{\mathrm{ji}}}) \tag{A.14}$$

$$C_3 = -\frac{1}{2}((\bar{\boldsymbol{x}}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{j}})^T \frac{n_{\mathrm{ji}} \kappa_1 \Sigma_{\mathrm{j}}^{-1}}{(n_{\mathrm{ji}} + \kappa_1)} (\bar{\boldsymbol{x}}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{j}})) \tag{A.15}$$

$$P(\bar{\boldsymbol{x}}_{\mathrm{ji}} | \boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1) = (2\boldsymbol{\pi})^{-\frac{d}{2}} |\frac{\Sigma_{\mathrm{j}}(\kappa_1 + n_{\mathrm{ji}})}{n_{\mathrm{ji}} \kappa_1}|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\bar{\boldsymbol{x}}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{j}})^T \frac{n_{\mathrm{ji}} \kappa_1 \Sigma_{\mathrm{j}}^{-1}}{(n_{\mathrm{ji}} + \kappa_1)} (\bar{\boldsymbol{x}}_{\mathrm{ji}} - \boldsymbol{\mu}_{\mathrm{j}})) \tag{A.16}$$

$$P(\bar{\boldsymbol{x}}_{\mathrm{ji}} | \boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1, n_{\mathrm{ji}}) = N(\bar{\boldsymbol{x}}_{\mathrm{ji}} | \boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}} (\frac{1}{n_{\mathrm{ji}}} + \frac{1}{\kappa_1})) \tag{A.17}$$

(6), (7) and (8) follow the model assumption and definition of Normal distribution. (9) is derived by completing the (8) into normal distribution and combining extra elements into constant C_3. New term in the (11) comes from evaluation of integral from (10) as the exponential term is in the Gaussian form. Combining the similar terms in (14), we get (15), hence marginal likelihood (16). Hereon, it is trivial to observe that the likelihood is in Gaussian form with mean and covariance as in (17).

Step 2: Posterior of $\boldsymbol{\mu_k}$

We combined the sufficient statistics (means) of classes sharing the same meta-class in the posterior distribution of meta-class mean $\boldsymbol{\mu}_{\mathrm{j}}$.

$$P(\boldsymbol{\mu}_{\mathrm{j}}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) \propto P(\boldsymbol{\mu}_{\mathrm{j}}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0) \prod_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} P(\bar{\boldsymbol{x}}_{\mathrm{ji}}|\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1) \tag{A.18}$$

$$P(\boldsymbol{\mu}_{\mathrm{j}}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) \propto exp(-\frac{1}{2}(\boldsymbol{\mu}_{\mathrm{j}} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})^T(\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}} + \kappa_1)} + \kappa_0)\Sigma_{\mathrm{j}}^{-1}(\boldsymbol{\mu}_{\mathrm{j}} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})) \tag{A.19}$$

$$P(\boldsymbol{\mu}_{\mathrm{j}}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) = N(\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\kappa}_{\mathrm{j}}^{-1}\Sigma_{\mathrm{j}}) \tag{A.20}$$

$$\bar{\boldsymbol{\mu}}_{\mathrm{j}} = \frac{\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}}+\kappa_1)}\bar{\boldsymbol{x}}_{\mathrm{ji}} + \kappa_0\boldsymbol{\mu_0}}{\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}}+\kappa_1)} + \kappa_0} \tag{A.21}$$

$$\bar{\kappa}_{\mathrm{j}} = (\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} \frac{n_{\mathrm{ji}}\kappa_1}{(n_{\mathrm{ji}} + \kappa_1)} + \kappa_0) \tag{A.22}$$

Applying Bayes rule, posterior can be proportioned as in (18). As meta–class mean and sample mean are Normal distributed (from step 1), we get (19). Completing square procedure used in previous step would give the exact normalization, hence, posterior can be written in a closed form of Gaussian as in (20). The last part can also be verified by observing all exponential terms are quadratic, indeed Gaussian.

Step 3: Updated prior of $\boldsymbol{\mu}_{\mathrm{j}c}$

As new information is available from classes sharing the same meta-class, current class mean ($\boldsymbol{\mu}_{\mathrm{j}c}$) can leverage this information by updating its prior. Marginalizing out meta-class mean $\boldsymbol{\mu}_{\mathrm{j}}$ would render this information propagation as below,

$$P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) = \int P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}, \kappa_1)P(\boldsymbol{\mu}_{\mathrm{j}}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}})d\boldsymbol{\mu}_{\mathrm{j}} \tag{A.23}$$

$$P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) = \int N(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu}_{\mathrm{j}}, \Sigma_{\mathrm{j}}\kappa_1^{-1})N(\boldsymbol{\mu}_{\mathrm{j}}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}})d\boldsymbol{\mu}_{\mathrm{j}} \tag{A.24}$$

$$P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}) = N(\boldsymbol{\mu}_{\mathrm{j}c}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}} + \Sigma_{\mathrm{j}}\kappa_1^{-1}) \tag{A.25}$$

Step 4: Posterior on $\boldsymbol{\mu}_{\mathrm{j}c}$

Combining the local prior from step 3 with current class sample mean $\bar{\boldsymbol{x}}_{\mathrm{j}c}$ from step 1, we derive the posterior for current class mean $\boldsymbol{\mu}_{\mathrm{j}c}$. Analogously, applying Bayes rule and observing that both distributions are Normal, we obtain another Gaussian.

$$P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{j}i}\}_{t_i=\mathrm{j}}, \bar{\boldsymbol{x}}_{\mathrm{j}c}) \tag{A.26}$$

$$\propto P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{j}i}\}_{t_i=\mathrm{j}}) P(\bar{\boldsymbol{x}}_{\mathrm{j}c}|\boldsymbol{\mu}_{\mathrm{j}c}, \Sigma_{\mathrm{j}} n_{\mathrm{j}c}^{-1}) \tag{A.27}$$

$$\propto N(\boldsymbol{\mu}_{\mathrm{j}c}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}} + \Sigma_{\mathrm{j}}\kappa_1^{-1}) N(\bar{\boldsymbol{x}}_{\mathrm{j}c}|\boldsymbol{\mu}_{\mathrm{j}c}, \Sigma_{\mathrm{j}} n_{\mathrm{j}c}^{-1}) \tag{A.28}$$

$$P(\boldsymbol{\mu}_{\mathrm{j}c}|\boldsymbol{\mu_0}, \Sigma_{\mathrm{j}}, \kappa_0, \kappa_1, \{\bar{\boldsymbol{x}}_{\mathrm{j}i}\}_{t_i=\mathrm{j}}, \bar{\boldsymbol{x}}_{\mathrm{j}c}) = N(\frac{n_{\mathrm{j}c}\bar{\boldsymbol{x}}_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}}\bar{\boldsymbol{\mu}}_{\mathrm{j}}}{n_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}}}, \Sigma_{\mathrm{j}}(\tilde{\kappa}_{\mathrm{j}}^{-1} + n_{\mathrm{j}c}^{-1})) \tag{A.29}$$

$$\tilde{\kappa}_{\mathrm{j}}^{-1} = \bar{\kappa}_{\mathrm{j}}^{-1} + \kappa_1^{-1} \tag{A.30}$$

Note that in order to simplify the notation, κ terms are collected in $\tilde{\kappa}_{\mathrm{j}}^{-1}$.

Before stepping into the covariance related derivations, we would like to draw your attention to an expression similar to C_3 from equation (A.15) that appears with $\bar{\boldsymbol{\mu}}_{\mathrm{j}}$ in the remaining terms while deriving posterior.

$$(\bar{\boldsymbol{x}}_{\mathrm{j}c} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})^T \frac{n_{\mathrm{j}c}\tilde{\kappa}_{\mathrm{j}}}{(n_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}})} \Sigma_{\mathrm{j}}^{-1} (\bar{\boldsymbol{x}}_{\mathrm{j}c} - \bar{\boldsymbol{\mu}}_{\mathrm{j}}) = \frac{n_{\mathrm{j}c}\tilde{\kappa}_{\mathrm{j}}}{n_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}}} tr((\Sigma_{\mathrm{j}}^{-1})(\bar{\boldsymbol{x}}_{\mathrm{j}c} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})(\bar{\boldsymbol{x}}_{\mathrm{j}c} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})^T)) \tag{A.31}$$

Note that $\bar{\boldsymbol{x}}_{\mathrm{j}c}$ and $\bar{\boldsymbol{\mu}}_{\mathrm{j}}$ are observed values, and the equation depends only on Σ_{j}. This formula stays in the exponent creating a factor that contributes to Wishart distribution and is denoted as $P(S_\mu|\Sigma_{\mathrm{j}})$.

Step 5: Wishart Terms

As it is observed from graphical model in Figure 2.2, meta-class covariance Σ_{j} is shared among its all classes. Even though Wishart terms are independent of mean variables, the residual terms in the posterior calculations create additional Wishart distributions. These

distributions and the ones from data scatter matrices are combined in the posterior of Σ_{j} as below,

$$P(\Sigma_{\mathrm{j}}|\{S_{\mathrm{ji}}, \bar{\boldsymbol{x}}_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, S_{\mathrm{j}c}, \bar{\boldsymbol{x}}_{\mathbf{j}\boldsymbol{c}}) \propto P(\Sigma_{\mathrm{j}}|\Sigma_0, m)P(S_{\mathrm{j}c}|\Sigma_{\mathrm{j}}, n_{\mathrm{j}c})P(S_\mu|\Sigma_{\mathrm{j}}) \prod_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} P(S_{\mathrm{ji}}|\Sigma_{\mathrm{j}}, n_{\mathrm{ji}}) \tag{A.32}$$

$$= IW(\Sigma_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S_{\mathrm{ji}} + S_{\mathrm{j}c} + S_\mu,\ m + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} (n_{\mathrm{ji}} - 1) + n_{\mathrm{j}c}) \tag{A.33}$$

$$S_\mu = \frac{n_{\mathrm{j}c}\tilde{\kappa}_{\mathrm{j}}}{\tilde{\kappa}_{\mathrm{j}} + n_{\mathrm{j}c}}(\bar{\boldsymbol{x}}_{\mathbf{j}\boldsymbol{c}} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})(\bar{\boldsymbol{x}}_{\mathbf{j}\boldsymbol{c}} - \bar{\boldsymbol{\mu}}_{\mathrm{j}})^T \tag{A.34}$$

Since the prior on Σ_{j} is Inverse-Wishart and scatter matrices are Wishart distributed, the posterior will be exactly Inverse-Wishart.

Step 6: Integration of remaining parameters

As you notice from Step 4 and 5, the posterior distribution is Normal-Inverse-Wishart and from model assumption, data is Gaussian. Hence, the integration in PPD can analytically be derived. Integration with respect to $\boldsymbol{\mu}$ will render another multivariate Normal distribution due to conjugacy, thereafter integration w.r.t Σ completes to Inverse-Wishart. Finally, arranging the terms yields the posterior predictive distribution in the form of Student-t.

$$P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \bar{\boldsymbol{x}}_{\mathbf{j}\boldsymbol{c}}, S_{\mathrm{j}c}, \boldsymbol{\mu}_{\mathbf{0}}, \kappa_0, \kappa_1) \tag{A.35}$$

$$= \int\int P(\boldsymbol{x}|\boldsymbol{\mu}_{\mathbf{j}\boldsymbol{c}}, \Sigma_{\mathrm{j}})P(\boldsymbol{\mu}_{\mathbf{j}\boldsymbol{c}}, \Sigma_{\mathrm{j}}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \bar{\boldsymbol{x}}_{\mathbf{j}\boldsymbol{c}}, S_{\mathrm{j}c}, \boldsymbol{\mu}_{\mathbf{0}}, \kappa_0, \kappa_1)d\boldsymbol{\mu}_{\mathbf{j}\boldsymbol{c}}d\Sigma_{\mathrm{j}}$$

$$= T(\boldsymbol{x}|\bar{\boldsymbol{\mu}}_{\mathbf{j}\boldsymbol{c}}, \bar{\Sigma}_{\mathrm{j}c}, \bar{v}_{\mathrm{j}c}) \tag{A.36}$$

$$\begin{aligned}
\bar{\boldsymbol{\mu}}_{\mathbf{j}\boldsymbol{c}} &= \frac{n_{\mathrm{j}c}\bar{\boldsymbol{x}}_{\mathbf{j}\boldsymbol{c}} + \tilde{\kappa}_{\mathrm{j}}\bar{\boldsymbol{\mu}}_{\mathrm{j}}}{n_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}}} \\
\bar{v}_{\mathrm{j}c} &= n_{\mathrm{j}c} + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} (n_{\mathrm{ji}} - 1) + m - d + 1 \\
\bar{\Sigma}_{\mathrm{j}c} &= \frac{n_{\mathrm{j}s} + \tilde{\kappa}_{\mathrm{j}} + 1}{(n_{\mathrm{j}c} + \tilde{\kappa}_{\mathrm{j}})\bar{v}_{\mathrm{j}c}}(\Sigma_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S_{\mathrm{ji}} + S_{\mathrm{j}c} + S_\mu)
\end{aligned}$$

As it is explained in the main text, depending on the current class status, seen or unseen, the PPD can take 2 forms by either containing current class sufficient statistics as above (A.36) or not as below (A.37).

$$
\begin{aligned}
P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \boldsymbol{\mu_0}, \kappa_0, \kappa_1) &= T(\boldsymbol{x}|\bar{\boldsymbol{\mu}}_{\mathrm{j}}, \bar{\Sigma}_{\mathrm{j}}, \bar{v}_{\mathrm{j}}) \\
\bar{v}_{\mathrm{j}} &= \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} (n_{\mathrm{ji}} - 1) + m - d + 1 \\
\bar{\Sigma}_{\mathrm{j}} &= \frac{(\tilde{\kappa}_{\mathrm{j}} + 1)}{\tilde{\kappa}_{\mathrm{j}} \bar{v}_{\mathrm{j}}} (\Sigma_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S_{\mathrm{ji}})
\end{aligned}
\tag{A.37}
$$

where $\bar{\boldsymbol{\mu}}_{\mathrm{j}}$ is from (A.21).

The previous 6 steps are for the unconstrained model PPD derivation. The generative model for constrained model is illustrated as below

$$
\begin{aligned}
x^d_{\mathrm{ji}k} &\sim N(\mu^d_{\mathrm{ji}}, \Sigma^d_{\mathrm{j}}) \\
\mu^d_{\mathrm{ji}} &\sim N(\mu^d_{\mathrm{j}}, \Sigma^d_{\mathrm{j}} \kappa_1^{-1}) \\
\mu^d_{\mathrm{j}} &\sim N(\mu^d_0, \Sigma^d_{\mathrm{j}} \kappa_0^{-1}) \\
\Sigma^d_{\mathrm{j}} &\sim IG(a_0, b_0)
\end{aligned}
\tag{A.38}
$$

where superscript d denotes the d^{th} component of each parameter. For the constrained model where explicit calculation of full covariance matrices are reduced to only diagonal values, most of the derivations is analogous. Indeed, first 4 steps are identical as they involve mean vectors. The 5^{th} step, nevertheless, follows slightly different path.

Following the same fashion as in step 5, we combined all scatter matrices and additional terms in the posterior of Σ_{j}, but this time for each diagonal value separately. Moreover, in constrained model, as mentioned before in the main text, there is no explicit scatter matrix calculations, hence scatter matrices in this model are assumed diagonal via putting Gamma priors on them to preserve conjugacy.

$$(n_{\mathrm{ji}}-1)S^d_{\mathrm{ji}} \sim G(\Sigma^d_{\mathrm{j}}, n_{\mathrm{ji}}-1) \tag{A.39}$$

$$(n_{\mathrm{j}c}-1)S^d_{\mathrm{j}c} \sim G(\Sigma^d_{\mathrm{j}}, n_{\mathrm{j}c}-1) \tag{A.40}$$

$$P(S^d_\mu|\Sigma^d_{\mathrm{j}}) = \frac{n_{\mathrm{j}c}\tilde{\kappa}_{\mathrm{j}}}{n_{\mathrm{j}c}+\tilde{\kappa}_{\mathrm{j}}}((\bar{x}^d_{\mathrm{j}c}-\bar{\mu}^d_{\mathrm{j}})^2/\Sigma^d_{\mathrm{j}}) \tag{A.41}$$

$$P(\Sigma^d_{\mathrm{j}}|\{S^d_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, S^d_{\mathrm{j}c}) \propto P(\Sigma^d_{\mathrm{j}}|a_0,b_0)P(S^d_{\mathrm{j}c}|\Sigma^d_{\mathrm{j}}, n_{\mathrm{j}c})P(S^d_\mu|\Sigma^d_{\mathrm{j}})\prod_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} P(S^d_{\mathrm{ji}}|\Sigma^d_{\mathrm{j}}, n_{\mathrm{ji}}) \tag{A.42}$$

$$= IG(b_0+\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S^d_{\mathrm{ji}} + S^d_{\mathrm{j}c} + S^d_\mu,\ a_0+\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}}(n_{\mathrm{ji}}-1)+n_{\mathrm{j}c}) \tag{A.43}$$

$$S^d_\mu = \frac{n_{\mathrm{j}c}\tilde{\kappa}_{\mathrm{j}}}{n_{\mathrm{j}c}+\tilde{\kappa}_{\mathrm{j}}}(\bar{x}^d_{\mathrm{j}c}-\bar{\mu}^d_{\mathrm{j}})^2 \tag{A.44}$$

Note that unlike unconstrained model, scatter matrices herein are not calculated, indeed they are random variables from Gamma distribution as specified above. Note that, only S^d_μ is calculated as it appears in the derivation. Analogously, since prior on Σ is Inverse-Gamma and likelihood is Gamma distributed together with data from Gaussian, the posterior for component d can be analytically derived as univariate-t distribution as well. Constrained model PPD for seen classes is provided as below,

$$\begin{aligned} P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \bar{\boldsymbol{x}}_{\mathrm{j}c}, S_{\mathrm{j}c}, \boldsymbol{\mu_0}, \kappa_0, \kappa_1) &= \prod_d T(x^d|\bar{\mu}^d_{\mathrm{j}c}, \bar{\Sigma}^d_{\mathrm{j}c}, \bar{v}_{\mathrm{j}c}) \\ \bar{\mu}^d_{\mathrm{j}c} &= \frac{n_{\mathrm{j}c}\bar{x}^d_{\mathrm{j}c}+\tilde{\kappa}_{\mathrm{j}}\bar{\mu}^d_{\mathrm{j}}}{n_{\mathrm{j}c}+\tilde{\kappa}_{\mathrm{j}}} \\ \bar{v}_{\mathrm{j}c} &= 2(n_{\mathrm{j}c}+\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}}(n_{\mathrm{ji}}-1)+a_0) \\ \bar{\Sigma}^d_{\mathrm{j}c} &= \frac{n_{\mathrm{j}s}+\tilde{\kappa}_{\mathrm{j}}+1}{(n_{\mathrm{j}c}+\tilde{\kappa}_{\mathrm{j}})\bar{v}_{\mathrm{j}c}}(b_0+\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S^d_{\mathrm{ji}} + S^d_{\mathrm{j}c} + S^d_\mu) \end{aligned} \tag{A.45}$$

As it is observed from the equation (A.45), complete PPD is the multiplication of student-t distributions for each component of test sample $\boldsymbol{x}$. Note that degrees of freedom, $\bar{v}_{\mathrm{j}c}$ doesn't depend on dimension and S^d_μ is from (A.44). The equation (A.46) illustrates the PPD for unseen classes in which current class related terms dropped,

$$
\begin{aligned}
P(\boldsymbol{x}|\{\bar{\boldsymbol{x}}_{\mathrm{ji}}, S_{\mathrm{ji}}\}_{t_{\mathrm{i}}=\mathrm{j}}, \boldsymbol{\mu_0}, \kappa_0, \kappa_1) &= \prod_d T(x^d|\bar{\mu}^d_{\mathrm{j}}, \bar{\Sigma}^d_{\mathrm{j}}, \bar{v}_{\mathrm{j}}) \\
\bar{v}_{\mathrm{j}} &= 2(\sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} (n_{\mathrm{ji}} - 1) + a_0) \\
\bar{\Sigma}^d_{\mathrm{j}} &= \frac{(\tilde{\kappa}_{\mathrm{j}} + 1)}{\tilde{\kappa}_{\mathrm{j}} \bar{v}_{\mathrm{j}}} (b_0 + \sum_{\mathrm{i}:t_{\mathrm{i}}=\mathrm{j}} S^d_{\mathrm{ji}})
\end{aligned}
\tag{A.46}
$$

where $\bar{\mu}^d_{\mathrm{j}}$ is from (A.21).

A.2 Implementation and Tuning Details

Coupled with 2 layer structure, hyper-parameters offer great flexibility for encoding different aspects of domain knowledge into the model. Referencing the main text, both constrained and unconstrained models have 5 parameters to tune of which $\{\kappa_0, \kappa_1, K\}$ are common in both. The hyperparameter κ_0 adjusts the separation between meta-class centers, on the other hand, κ_1 adjusts the dispersion between class centers of meta-classes. On top of these 3 parameters, unconstrained model has 2 more parameters $\{m, s\}$ where m controls the degree of deviation of individual Σ_{j}'s from the $E[\Sigma_{\mathrm{j}}]$ and s is the scale constant. As the expected value of Σ_{j} is $\frac{\Sigma_0}{m-d-1}$, larger values for m assumes classes with more spherical shapes, whereas smaller values creates resilience to learn more flexible shapes. Constrained model has 2 more hyper-parameters as well, scale and shape parameters of Inverse-Gamma: $\{a_0, b_0\}$.

Hyper-parameters are coarsely tuned to maximize the Harmonic mean. For the sake of reproducibility, the range of each parameters and the best quintuplets associated with each dataset are depicted in the Tables A.2, A.3.

Table A.2. Parameter ranges used in hyper-parameter tuning

HP	**Range**	
	UBZSL	**CBZSL**
κ_0	$\{0.01, 0.1, 1, 10\}$	$\{0.01, 0.1, 1, 10\}$
κ_1	$\{0.01, 0.1, 1, 10\}$	$\{0.01, 0.1, 1, 10\}$
K	$\{2, 3, 4\}$	$\{2, 3, 4\}$
m	$\{10d, 50d, 500d\}$	—
s	$\{1, 3, 5, 7, 9\}$	—
a_0	—	$\{1, 10, 100\}$
b_0	—	$\{1, 10, 100\}$

Table A.3. Best quintuplets from tuning with the order of $\{\kappa_0, \kappa_1, m, s, K\}$ for UBZSL and $\{\kappa_0, \kappa_1, a_0, b_0, K\}$ for CBZSL

Datasets	**Best quintuplets**	
	UBZSL	**CBZSL**
FLO	$\{10, 10, 50d, 3, 3\}$	$\{0.01, 0.1, 10, 1, 3\}$
SUN	$\{0.1, 10, 50d, 7, 2\}$	$\{1, 10, 100, 10, 2\}$
CUB	$\{1, 10, 50d, 3, 3\}$	$\{1, 10, 10, 1, 2\}$
AWA1	$\{10, 10, 500d, 3, 2\}$	$\{1, 10, 10, 1, 2\}$
AWA2	$\{10, 10, 500d, 3, 2\}$	$\{1, 10, 10, 1, 2\}$
aPY	$\{10, 10, 500d, 9, 4\}$	$\{0.1, 10, 10, 1, 4\}$
ImageNet	$\{0.1, 10, 50d, 7, 2\}$	$\{1, 10, 100, 10, 2\}$

B. ZSL WITH DNA APPENDICES

B.1 BOLD Database

Since BOLD is an open-access database, a manual effort is needed to further curate a clean dataset. Figure B.1 displays a small subset of images deleted during cleaning process. Note that, for INSECT dataset, only cases with images and matching DNA barcodes are included whereas for CUB dataset, we did not impose this restriction as we only needed DNA information. Consequently, we retrieved all DNA barcodes from bird species (extracted from COI genes only) present in BOLD.

Figure B.2 presents further details on INSECT dataset such as number of species per genera and number of samples per species. From Figure B.2(a), one can observe that dataset can be considered balanced as more than 90% of species have samples between 10 and 30. Fine-grained nature of the dataset, on the other hand, can be seen from Figure B.2(b). Out of 578 genera, 270 of them have at least 2 species in the dataset. In 50 genera, there are more than 4 species coming from the same genus, which makes the data challenging yet, at the same time, provides a chance to find similar seen classes for the unseen classes.

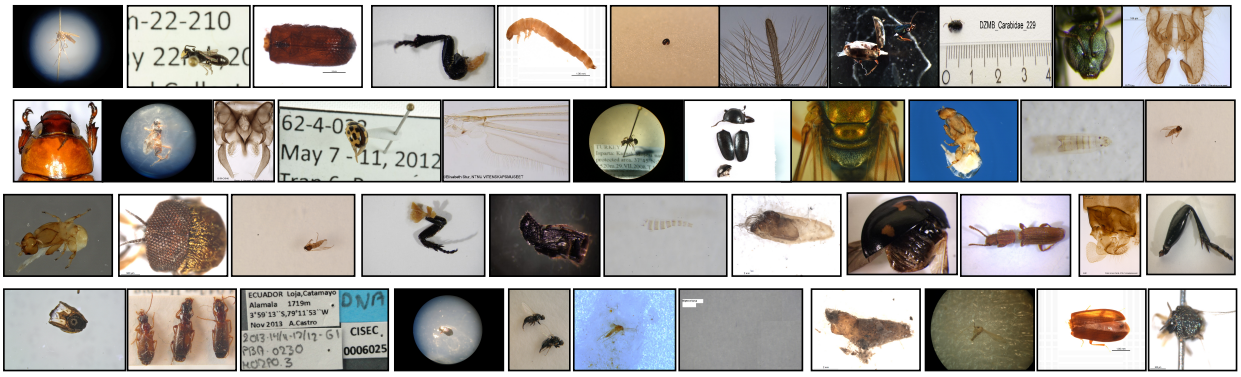
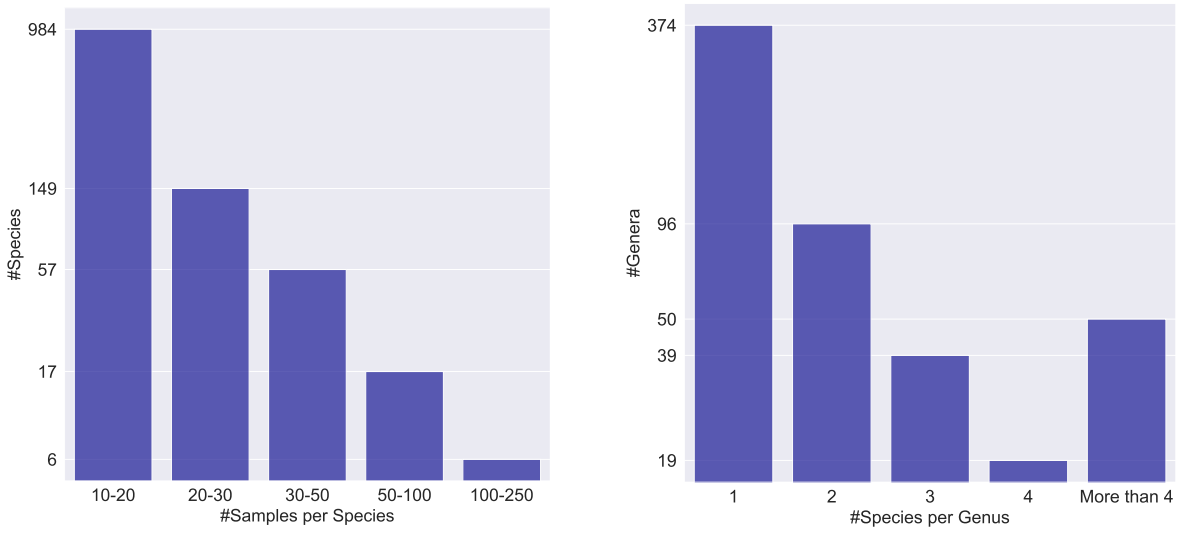


Figure B.1. Small subset of sample images deleted from INSECT dataset during data cleaning. Images inside of a circle are taken from microscope camera, thus, had very low resolution. Some images display only body parts, which is enough to extract DNA information but useless for image classification. There are many images in which insects are positioned very far from camera, hence almost no morphological characteristics were visible.



(a) Samples per Species

(b) Species per Genera

Figure B.2. INSECT Data statistics

B.2 Discussion on Limitations of DNA as Side Information

Starting with stating the obvious, DNA barcodes can only be used as side information for living organisms which span from simple bacteria to plants, fungi and all animals. Moreover, specialized information is required to extract and use DNA as side information. However, this is not much different than using visual attributes as side information, which also requires specialized information and well-trained taxonomists. For example, for annotating insects with visual attributes one has to go through a long list of morphological traits as described in this document (A Pictorial Key to the Order of Adult Insects). This list does not even cover emerging morphological traits in unseen species. Beside DNA is much more precise than visual attributes in identifying species, mainly because speciation occurs as a result of changes in DNA. We believe that our efforts in this direction will open new avenues for research, particularly in areas involving DNA synthesis and variants leading to speciation.

Another possible handicap in DNA barcoding is the labeling. Genbank, BOLD and even museum collections are known to contain some labeling errors, particularly for challenging-to-identify species. There are classes of insects that contain a lot of biodiversity and they cannot be reliably identified, requiring as genitalia dissections and likely DNA data. These

errors can then be translated into incorrectly labeled DNA sequences. Nonetheless, the limitation is the humans doing the identifications. This is one of the Future work directions we want to pursue. The goal is to generate better (3D) images, and then extract digital attributes from these high quality images themselves.

B.3 Training Details

In this section, we provide training details regarding implementation and hyperparameter tuning for CNN model, BZSL model and other state-of-the-art ZSL methods.

B.3.1 Implementation

Experiments are run on two machines: (1) a Dell PC with Windows 10, Intel(R) Core(TM) i9-9900 CPU @ 3.10 GHz, 64 GB RAM, and NVIDIA GeForce RTX 2080 GPU, 8GB RAM (2) School server with NVIDIA Tesla V100-PCIE GPU, 16GB RAM. LambdaLab [155] bechnmarking tests reveal that on average RTX 2080 is 56% as fast as Tesla V100 GPU for Deep Learning training. All models except FGNs are run on the PC. CADA-VAE and LsrGan are run on the school server.

CNN Model

The same model and parameters are utilized in learning both INSECT and CUB datasets' DNA embeddings, except the sequence length. Since in INSECT dataset more than 90% of DNA barcodes have a length of 658, we transformed barcodes into 658x5 2D arrays during one-hot endcoding, whereas for bird dataset, we utilized maximum sequence length, 1500, for one-hot encoding 2D array. The reason for the latter was that bird DNA barcode lengths have a high variance and results on validation set maximized once we use maximum sequence length. For the padding of missing bases, we simply used the label *others* which represents missing and ambiguous symbols.

BZSL

The model is developed in both Python and Matlab. Please check the Github page for instructions to run the experiments.

Other ZSL Methods

We compared the proposed model against six state-of-the-art ZSL approaches and five of them have a publicly available code. We got the code for ALE [41] from the authors themselves and it is developed in MATLAB. Methods and links to their codes are presented below:

- CADA-VAE [58]: https://github.com/edgarschnfld/CADA-VAE-PyTorch. The code is developed by authors in *PyTorch*. GPU enabled code.
- LsrGan [93]: https://github.com/Maunil/LsrGAN. The code is developed by authors in *PyTorch*. GPU enabled code.
- CRNet [106]: https://github.com/Fezaries/CRnet. The code is developed in *Python*. GPU enabled code.
- RelationNet [44]: https://github.com/lzrobots/LearningToCompare_ZSL. The code is developed by authors in *PyTorch*. GPU enabled code.
- ESZSL [42]: https://github.com/mvp18/Popular-ZSL-Algorithms. The code is developed in *Python*.

B.3.2 Hyperparameter Tuning

CNN Model

We very coarsely tuned the CNN model. After fixing the model architecture, we tuned initial learning rate, batch size and number of epochs between the ranges given below:

- Learning rate: $\{0.1, 0.01, 0.05, 0.001\}$

Table B.1. Parameter ranges used in hyperparameter tuning

HP	Range
κ_0	$\{0.1, 1\}$
κ_1	$\{1, 10, 25\}$
K	$\{1, 2, 3\}$
m	$\{5d, 25d, 100d, 500d\}$
s	$\{1, 5, 10\}$

Table B.2. Best quintuplets from tuning with the order of $\{\kappa_0, \kappa_1, m, s, K\}$.

Datasets	Best quintuplets
INSECT	$\{0.1, 10, 5d, 10, 3\}$
CUB (w. Att)	$\{1, 25, 500d, 10, 3\}$
CUB (w. w2v)	$\{0.1, 25, 5d, 5, 2\}$
CUB (w. DNA)	$\{0.1, 25, 25d, 5, 3\}$

- Batch size: $\{32, 64\}$
- Number of epochs: $\{5, 10\}$

We also tried *SGD* optimizer but *ADAM* was superior.

BZSL

Referencing the main text, unconstrained model has 5 parameters to tune: $\{\kappa_0, \kappa_1, K, m, s\}$. As you may recall from the main text, the hyperparameter κ_0 adjusts the separation between local prior centers, on the other hand, κ_1 adjusts the dispersion between class centers inheriting the same local prior. Moreover, m controls the degree of deviation of individual Σ_j's from the $E[\Sigma_j]$ and s is the scale constant. As the expected value of Σ_j is $\frac{\Sigma_0}{m-d-1}$ where d is the dimension of the data, larger values for m assumes classes with more spherical shapes, whereas smaller values creates resilience to learn more flexible shapes.

Model hyperparameters are coarsely tuned to maximize the Harmonic mean on validation set. The only preprocessing we did was to apply PCA to reduce the dimensionality of the data from 2048 to 500. The range of each parameter and the best quintuplets associated with each dataset are depicted in the Tables B.1 and B.2, respectively. Running time of the model was 33 seconds per trial.

Other Methods

Out of six ZSL methods, four of them are based on neural networks and it is not feasible, if not impossible, to tune all hyperparameters. Thus, we utilized Hyperopt [156], a distributed hyperparameter optimization technique developed in Python to tune all SotA ZSL methods, except ALE. All models are tuned to maximize the harmonic mean.

ALE was built on MATLAB and has only two parameters to tune: learning rate and margin constant, hence no need to use sophisticated software for tuning. Learning rate was tuned in the set of $\{1, 0.5, 0.25, 0.1, 0.05, 0.025, 0.01, 0.005, 0.001\}$, whereas margin constant was tuned from the set of $\{0.01, 0.1, 0.5, 1\}$. The model is run 100 epochs with early stop is set to 30 epochs.

For other methods, hyperparameters and their range are listed below (ranges are arranged by observing parameters used by authors for different benchmark datasets and then slightly extended):

CADA-VAE (100 runs):

- Classifier learning rate (LR) (*lr_cls*): *log-uniform* from range of $[log(1e{-}5), log(1e{-}2)]$
- Generative model LR (*lr_gen_model*): *log-uniform* from range of $[log(1e{-}5), log(1e{-}2)]$
- Classifier training steps (*cls_train_steps*): *random-sampling* from set of $\{4 : 1 : 40\}$
- Batch size (*batch_size*): *random-sampling* from set of $\{32, 64, 96, 128\}$
- Latent space dimensionality (*latent_size*): *random-sampling* from set of $\{32, 64, 96, 128\}$
- Regularizer loss (*loss*): *random-sampling* from set of $\{L1, L2\}$

LsrGan (25 runs due to slow training time, see Table B.3):

- Class weight (*cls_weight*): *log-uniform* from range of $[log(1e-3), log(1e-1)]$
- LR (*lr*): *log-uniform* from range of $[log(1e-6), log(1e-3)]$
- Unseen class weight (*unseen_cls_weight*): *log-uniform* from range of $[log(1e{-}2), log(0.9)]$

- Epsilon (*epsilon*): *log-uniform* from range of $[log(1\text{e}-2), log(0.9)]$
- Upper epsilon (*upper_epsilon*): *log-uniform* from range of $[log(1\text{e}-2), log(0.9)]$
- Number of synthesized features per class (*syn_num*): *random-sampling* from set of $\{100:100:3000\}$
- Number of epochs (*nepoch*): *random-sampling* from set of $\{10:5:50\}$
- Correlation penalty (*correlation_penalty*): *random-sampling* from set of $\{5:10:50\}$
- Pretrained classifier for inference (*no_classifier*): *random-sampling* from set of $\{True, False\}$

CRNet (50 runs)

- LR (*LEARNING_RATE*): *log-uniform* from range of $[log(1\text{e}-6), log(1\text{e}-2)]$
- Weight decay (*WEIGHT_DECAY*): *log-uniform* from range of $[log(1\text{e}{-}20), log(1\text{e}{-}3)]$
- Number of clusters (K): *random-sampling* from set of $\{1:1:21\}$

RelationNet (50 runs)

- LR (*learning_rate*): *log-uniform* from range of $[log(1\text{e}-6), log(1\text{e}-3)]$
- Number of episodes (*episode*): *random-sampling* from set of $\{50,000:10,000:200,001\}$
- Step size of learning rate scheduler (*lr_step_size*): *random-sampling* from set of $\{10,000:10,000:200,001\}$

ESZSL (100 runs)

- Regularization constant alpha (*alpha*): *uniform* from range of $[-10, 10]$
- Regularization constant gamma (*gamma*): *uniform* from range of $[-10, 10]$

B.4 Additional Experiments

B.4.1 Model Runtime Analysis

Table B.3 displays runtime of all ZSL methods on CUB dataset while visual attributes are used as side information. All models but LsrGan are run on the PC for this experiment. On server, average runtime per trial for LsrGan was $1,834$ seconds and using the LambdaLab research as a reference point an approximate runtime of LsrGan on PC is calculated as $1,834/0.56 = 3,274$ seconds. ESZSL and BZSL has the lowest running time per trial and BZSL is 30 times faster than the next fastest method, CADA-VAE. Both methods owe the super fast training time to their closed form solutions.

Table B.3. Running time in seconds per trial on CUB dataset (our version in which 6 classes are not present).

Method	**Ave. runtime per trial**
CADA-VAE	280
LsrGan	$3,274$
CRNet	$1,625$
RelationNet	882
ALE	513
ESZSL	3.6
BZSL	8.6

B.4.2 Visualization of Synthesized Features from FGNs.

To better understand the effect of various side information sources on FGNs' performance, we visualized generated unseen class features from CADA-VAE and LsrGan on CUB data using 3 different side information sources: visual attributes (common one), word2vecs and DNA embeddings. For each experiment we sampled 60 points for TSNE training and then randomly sampled 20 unseen classes for visualization. Figure B.3 displays the TSNE plot from CUB data. Since the same randomly selected 20 unseen classes are used for all figures, we only put one legend and it is in the Figure B.3. Please note that, we utilized the best setup from tuning for the model training and we only sampled 20 classes after training done for

only visualization purposes. TSNE plots of synthesized features from these 2 models using visual, word2vec and DNA attributes are presented in Figures B.4, B.5, B.6, respectively.

The first thing to notice is the bizarre distribution of synthesized features from CADA-VAE. The model to some extent is able to transfer the relative inter-class proximity from the attribute space to image feature space and achieves competitive results yet the generated features lack quality. The superior results of LsrGan when visual attributes are available is visible by the TSNE plots in Figure B.4. However, once the correlation between side information and image features decreases, the quality of generated features suffers. Both methods display mode collapse once word vectors are used as side information (see red circle in Figure B.5). On the other hand, with DNA embeddings, features from LsrGan are scattered all around and fail to form meaningful clusters. Despite their strange shapes, features generated by CADA-VAE seem to be slightly more reasonable and it is reflected in their better performance with DNA embeddings.

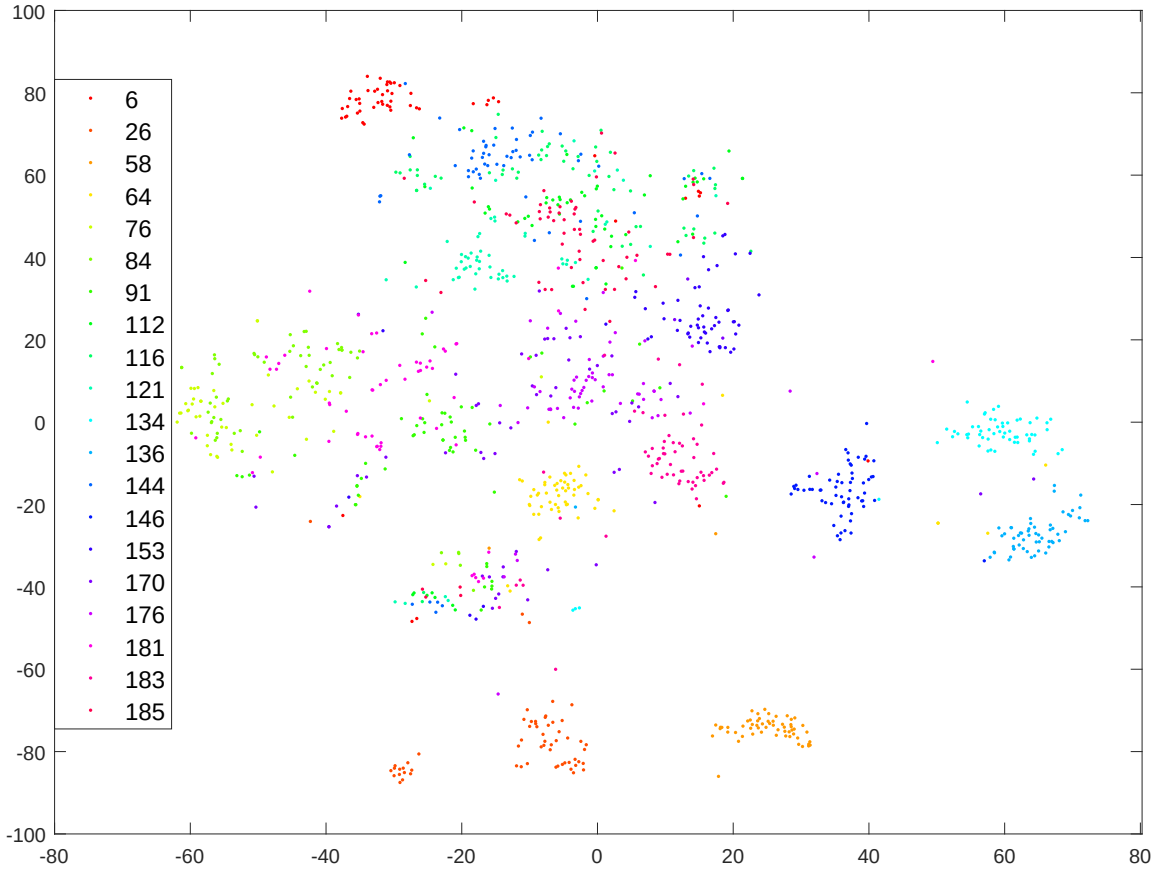
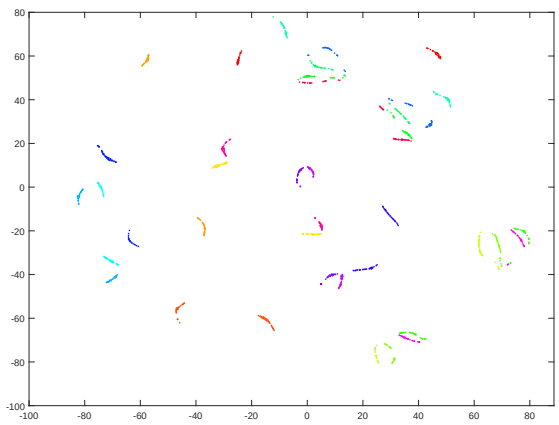
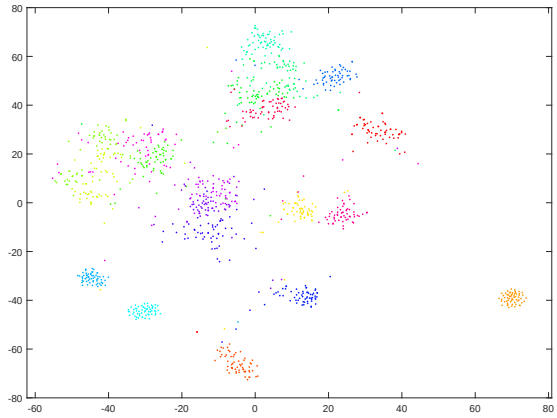


Figure B.3. TSNE plot of randomly sampled 20 unseen classes from CUB data

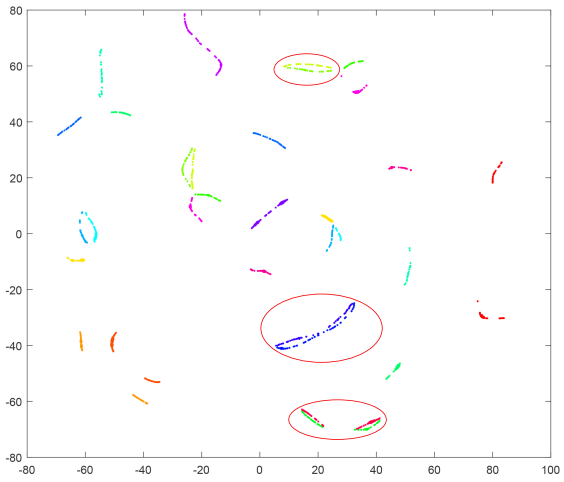


(a) Synthesized unseen class features from CADA-VAE

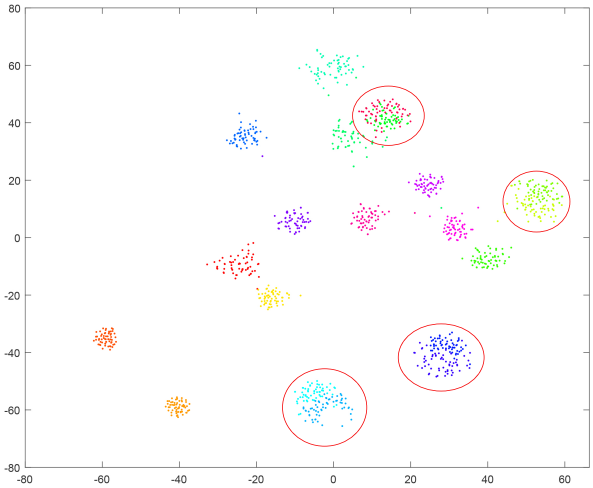


(b) Synthesized unseen class features from LsrGan

Figure B.4. TSNE plots using visual attributes as side information during model training on CUB data

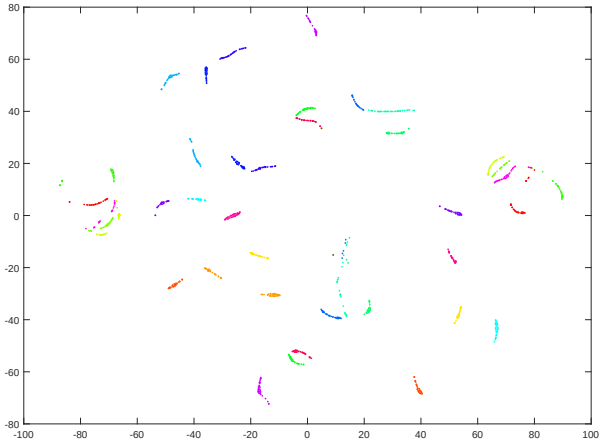


(a) Synthesized unseen class features from CADA-VAE

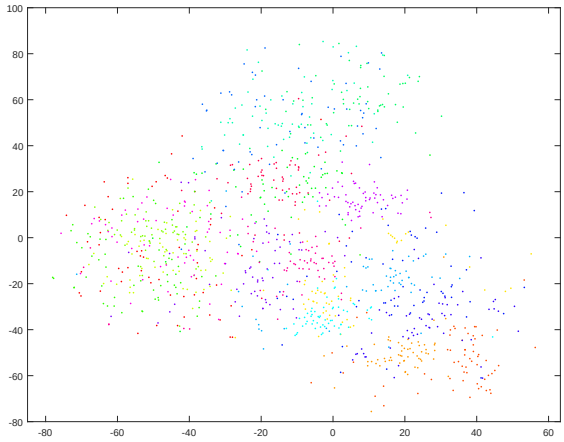


(b) Synthesized unseen class features from Lsr-Gan

Figure B.5. TSNE plots using word2vec as side information during model training on CUB data



(a) Synthesized unseen class features from CADA-VAE



(b) Synthesized unseen class features from LsrGan

Figure B.6. TSNE plots using DNA as side information during model training on CUB data