Generalized Category Discovery for Novel Plankton Species

Rohan Jagannathan
KTH Royal Institute of Technology
Stockholm, Sweden
rohanja@kth.se

Adhithyan Kalaivanan
KTH Royal Institute of Technology
Stockholm, Sweden
adhkal@kth.se

Josephine Sullivan
KTH Royal Institute of Technology
Stockholm, Sweden
sullivan@kth.se

***Abstract*—Understanding plankton populations is critical for ecological and environmental research, yet traditional methods of plankton classification are labor-intensive and inefficient. This paper proposes an approach for classifying plankton species with Generalized Category Discovery (GCD) and deep learning. The challenge in plankton classification lies in the incomplete datasets, where many species are underrepresented. Approaching this problem with GCD techniques addresses this by identifying new categories within image datasets, even without predefined labels. Our method employes a ResNet-18 backbone for feature extraction, fine-tuned with semi-supervised contrastive representation learning. We introduce a semi-supervised Expectation Maximization (EM) clustering algorithm with a mixture of von Mises-Fisher distributions suitable for high-dimensional spherical data. Evaluations on the WHOI-Plankton dataset demonstrate that our approach outperforms traditional clustering methods for medium to large classes in recognizing both known and novel plankton species. This work shows significant promise in automating and scaling plankton classification, which would be a valuable tool for ecological monitoring.**

I. Introduction

Plankton are a vital component of our planet's ecosystem. As a result, it is critical that we study and understand them. Understanding how their populations grow and spread can provide us with an incredible amount of information about environmental phenomena, ranging from climate change to the carbon cycle. Currently, plankton population monitoring is done manually with microscopes, which is incredibly time-consuming and inefficient. In order to solve this problem, we propose a method that uses deep learning to automatically classify images of plankton using Generalized Category Discovery.

One of the largest issues surrounding plankton classification is the lack of complete datasets. Due to their rarity or seasonal emergence, there may be some species of plankton that are not present in a dataset but are found in the area where the images were collected. Since we still want to be able to classify images of such plankton and be able to distinguish between different species of unseen plankton, we cannot employ the techniques used by traditional image classification systems that rely on predefined categories with labeled data.

To address such limitations, recent strides have been made in the emerging field of Generalized Category Discovery (GCD) [10]. GCD aims to identify new categories within image datasets, even when some categories are not labeled or known in advance. This is particularly useful for the automatic classification of plankton, where not all species may be represented in the training data.

Another underlying issue present in the problem of plankton classification is imbalanced data. Due to seasonal patterns and the distribution of species in the areas where the samples are collected, many plankton datasets are heavily imbalanced. The discrepancy between the sizes of the most and least populous classes can be massive, with it being well over 10,000:1 for some datasets. This makes some techniques that assume an even distribution of labels suffer.

In this paper, we propose a method for Generalized Category Discovery applied to the classification of plankton species. Our approach involves using deep learning to extract features plankton imaging and then applying a semi-supervised clustering algorithm to identify and classify both known and novel plankton species. The clustering algorithm we introduce is based on Expectation Maximization (EM) with a mixture of von Mises-Fisher distributions (moVMF) [2], which is well-suited for high-dimensional spherical data. In addition, unlike many of the most popular clustering algorithms, EM allows for each cluster to be of a different size. This is incredibly useful for the unbalanced plankton datasets.

We evaluate our method using the publicly available WHOI-Plankton dataset [8], which contains images of over 100 different plankton species. By comparing our approach to existing clustering methods, we demonstrate that our semi-supervised EM with a mixture of von Mises-Fisher distributions shows significant promise.

In the following sections, we review related work in the fields of GCD and EM, describe our proposed method in detail, present experimental results, and discuss the implications of our findings. We conclude with potential directions for future research.

II. Related Work

A. Generalized Category Discovery

The problem of Generalized Category Discovery (GCD) was formalized in [10]. Essentially, GCD aims to identify and define new categories and clusters within image datasets without relying on predefined labels or prior knowledge. More specifically, assume we are given a dataset $\mathcal{D}$. Suppose $\mathcal{D}$ is composed of disjoint subsets $\mathcal{D}_L$ and $\mathcal{D}_U$, where the examples

in $\mathcal{D}_L$ have class labels and those in $\mathcal{D}_U$ do not. In GCD, we want to be able to assign class labels to the examples in $\mathcal{D}_U$ even though some elements of $\mathcal{D}_U$ may belong to classes that were not present in $\mathcal{D}_L$.

Typical approaches to this problem involve training a backbone model to extract feature representations from the images and then clustering them to classify them. The backbone models are typically based on either a ResNet [7] or Vision Transformer [6] pre-trained with DINO [3]. They are then fine-tuned on the dataset with self-supervised contrastive learning. Common clustering algorithms to generate pseudo-labels include k-means, semi-supervised k-means, and expectation maximization [4].

Others have employed parametric classification approaches to GCD, which has shown some promising results [11].

B. EM Clustering with a mixture of Von Mises-Fisher Distributions

Expectation Maximization (EM) is a widely used clustering algorithm that serves as an alternative to other parametric clustering techniques like k-means. It is conceptually similar to k-means, but instead of being treated as a singular point, each cluster is a probability distribution that assigns every point in space a probability of belonging to it. The parameters of the distributions are iteratively updated based on the examples in the dataset assigned to them to better fit them with maximum likelihood estimation. Depending on the distributions used to model the clusters, EM allows for groupings with a variety of geometries to be properly clustered.

In addition to clustering data in n-dimensional Euclidean spaces, EM can be used to cluster data on the $(n-1)$-dimensional unit hypersphere, S^{n-1}, using a mixture of von Mises-Fisher distributions (moVMF) [2]. The von Mises-Fisher distribution (vMF) is a probability distribution used for modeling points on a hypersphere. It is analogous to the multivariate Gaussian distribution, but for high dimensional spherical data. Performing EM with a moVMF allows for higher dimensional data that has been normalized to lie on the unit hypersphere to be clustered in a natural way.

III. Methods

In this section, we discuss the methods we used to tackle the generalized category discovery problem for novel plankton species.

Our approach consists of two steps: feature extraction and clustering. The feature extraction component largely follows the process outlined in [10]. In the clustering step, we introduce a new way of clustering the extracted features by performing semi-supervised Expectation Maximization using a mixture of von Mises-Fisher distributions.

A. Feature Extraction

For feature extraction, we choose to use ResNet-18 [7] pre-trained with DINO on ImageNet [5] as the backbone. While others have seen superior results with vision transformers such as ViT-B/16 [6], ResNet-18 is much less demanding to train, making it a much more appealing choice.

As discussed in [10], the DINO pre-training alone is insufficient for the backbone to be ready for feature extraction, so we further fine-tune it using semi-supervised contrastive representation learning. For this task, we use the semi-supervised InfoNCE [9] loss function. This loss function is computed over each batch during training, and is defined as

$$\mathcal{L} = (1-\lambda)\sum_{i\in B}\mathcal{L}_i^u + \lambda\sum_{i\in B_\ell}\mathcal{L}_i^s \tag{1}$$

where $\mathcal{L}^u$ is the unsupervised InfoNCE loss function, $\mathcal{L}^s$ is the supervised InfoNCE loss function, λ is a smoothing coefficient, B is the current batch, and B_ℓ is the labeled subset of B.

Let $\mathbf{x}_i$ and $\mathbf{x}_i'$ be two views of the same image (a positive pair). The unsupervised InfoNCE loss function is defined as follows.

$$\mathcal{L}_i^u = -\log\frac{\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_i')/\tau)}{\sum_{k=1}^{N}\mathbb{1}_{[k\neq i]}\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_k)/\tau)} \tag{2}$$

Here, $\mathbf{z}_i = \phi(f(\mathbf{x}_i))$, where f is the backbone and ϕ is a projection head that reduces the dimensionality of the features for contrastive loss computation, which is standard in the contrastive training literature. $\mathbb{1}_{[k\neq i]}$ is an indicator function that equals 1 if the condition in the subscript is met and 0 otherwise, and τ is a temperature parameter that controls the sharpness of the loss computation. $\text{sim}(\mathbf{z}_i,\mathbf{z}_i')$ is an arbitrary function that measures the similarity between two features. In our case, we used the cosine similarity metric, defined as follows.

$$\text{sim}(\mathbf{z}_i,\mathbf{z}_i') = \frac{\mathbf{z}_i\cdot\mathbf{z}_i'}{\|\mathbf{z}_i\|\|\mathbf{z}_i'\|} \tag{3}$$

Using an angle-based similarity metric like cosine similarity during the contrastive representation learning is essential for the semi-supervised EM clustering with a mixture of von Mises-Fisher distributions to be effective.

The supervised InfoNCE loss function is similar to the unsupervised version but applied only to labeled data. Here, the positive pairs are not just crops from the same image, but are crops across different images with the same label. The supervised InfoNCE loss function is defined as follows.

$$\mathcal{L}_i^s = -\frac{1}{|\mathcal{N}(i)|}\sum_{q\in\mathcal{N}(i)}\log\frac{\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_q)/\tau)}{\sum_{k=1}^{N}\mathbb{1}_{[k\neq i]}\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_k)/\tau)} \tag{4}$$

Here, $\mathcal{N}(i)$ is the set of indices of other examples in the batch with the same label as $\mathbf{x}_i$.

We use this semi-supervised InfoNCE loss function with a cosine similarity metric to fine-tune the feature extractor over a number of epochs. This completes the training for the feature extractor.

B. Clustering

The original GCD paper, [10], uses a semi-supervised k-means algorithm to cluster the data once the features have been extracted. Like us, they use an angle-based similarity metric (dot product similarity) while performing contrastive representation learning on the feature extractor. As a result, we find that using k-means may not be ideal, as it uses the Euclidean distance between feature vectors to determine clusters rather than considering the angular similarity. Instead, we propose the use of a semi-supervised Expectation Maximization algorithm with a mixture of von Mises-Fisher distributions.

1) The von Mises-Fisher Distribution (vMF): The von Mises-Fisher distribution is the hyperspherical analog to the Gaussian distribution, and its pdf is defined as follows.

$$f_p(\mathbf{x}; \boldsymbol{\mu}, \kappa) = C_p(\kappa) e^{\kappa(\boldsymbol{\mu} \cdot \mathbf{x})} \tag{5}$$

Here, p is the dimensionality of the vector $\mathbf{x}$, and $\boldsymbol{\mu}$ and κ are parameters of the distribution. $\boldsymbol{\mu}$ is the normalized mean of the distribution (constrained to $\|\boldsymbol{\mu}\| = 1$, and indicates the direction on the $(p-1)$-dimensional hypersphere (S^{p-1}) where the vMF distribution is centered. κ is the concentration parameter and determines the "spread" of the vMF distribution. It is constrained to be nonzero ($\kappa \geq 0$), and is inversely proportionate to how "spread out" the distribution is. As κ approaches 0, the vMF distribution tends towards the uniform distribution on the p-dimensional unit hypersphere, and as it tends towards infinity, the distribution concentrates its entire probability mass on a singular point.

$C_p(\kappa)$ is a normalization constant. It is defined as follows.

$$C_p(\kappa) = \frac{\kappa^{(p/2)-1}}{(2\pi)^{p/2} I_{(p/2)-1}(\kappa)} \tag{6}$$

I_v denotes the modified Bessel function of the first kind at order v. This normalization constant ensures that the distribution is normalized over the $(p-1)$-dimensional unit hypersphere, so that

$$\int_{\mathbf{x} \in S^{p-1}} f_p(\mathbf{x}) \; d\mathbf{x} = 1 \tag{7}$$

2) Expectation Maximization (EM): Expectation Maximization (EM) is a widely used clustering algorithm. It is most commonly applied to some sort of mixture model, typically Gaussian Mixture Models. In EM, the number of clusters, k, must be specified beforehand.

The probability density function for a mixture model with k components is defined as follows.

$$p(x|\theta) = \sum_{i=1}^{k} \alpha_i f_i(x|\theta_i) \tag{8}$$

Here, f_i represents the pdf of the i-th probability distribution in the mixture model, with θ_i being the parameters of the i-th distribution. α_i is a weight assigned to the i-th probability distribution, constrained by $\sum_{i=1}^{k} \alpha_i = 1$ to ensure the mixture model is normalized.

The end goal of EM is to find the set of parameters for each distribution in the mixture model that maximizes its log-likelihood. The log-likelihood of the mixture model is given by

$$L(\theta) = \sum_{i=1}^{N} \log \Big(\sum_{j=1}^{k} \alpha_j f_j(x_i, \theta_j) \Big) \tag{9}$$

where N is the number of examples in the dataset. The way this goal is achieved is by iteratively improving the parameters by repeatedly applying two steps: expectation and maximization.

The expectation step (E-step) involves estimating the cluster memberships of each data point given the current set of parameters. Let θ_i^t be the set of parameters for the i-th distribution on iteration t. The E-step on iteration t computes the probability that each data point belongs to each of the k clusters. For data point x, the probability that it belongs to cluster i is simply $\alpha_i f_i(x|\theta_i^t)$.

The maximization step (M-step) takes the posterior probabilities calculated in the E-step and performs maximum likelihood estimation to nudge the parameters of the mixture model towards a higher log likelihood. For simplicity, assume all k distributions are of the same type, and let q be the number of parameters for each distribution. The updated j-th parameter of the i-th distribution, θ_{ij}^{t+1}, is computed by solving the following equation for θ_{ij}

$$\frac{\partial}{\partial \theta_{ij}} \Big[\sum_{n=1}^{N} \gamma_{in} \log(f_i(x_i|\theta_{i1}, \theta_{i2}, ..., \theta_{iq})) \Big] = 0 \tag{10}$$

Here, γ_{in} is the posterior probability of the n-th data point belonging to cluster i computed in the E-step. Each α_i is also updated using maximum likelihood estimation.

Each EM iteration consists of one E-step followed by one M-step. The algorithm terminates when $\sum_{i=1}^{k} \sum_{j=1}^{q} |\theta_{ij}^t - \theta_{ij}^{t+1}| < \varepsilon$ for some threshold ε, signifying convergence, or if a set maximum number of epochs is performed. At this point, we can recompute the posterior probabilities for each data point and assign each one to the cluster with the highest posterior probability.

3) EM with a mixture of vMF distributions (moVMF): Now, we will apply the principles of EM discussed in the previous section to a mixture of vMF distributions. The mixture model is defined as

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\kappa}) = \sum_{i=1}^{k} \alpha_i f_i(\mathbf{x}|\boldsymbol{\mu}_i, \kappa_i) \tag{11}$$

where each f_i is a vMF distribution whose pdf is given in Equation 5.

In the case of EM with a mixture of mVMF distributions, all points in the dataset must be normalized to be of unit length. This is imperative, as the pdf for the vMF distribution is only valid if $\|\mathbf{x}\| = 1$.

In theory, we should be able to compute the maximum likelihood estimates for the parameters of the mixture of vMF distributions using the process described in the previous

section. However, it turns out that this is impossible, as finding the maximum likelihood estimate for κ would require the computing the inverse of a ratio of Bessel functions, which there is no closed-form solution for. Banerjee et al. uses numerical methods to provide an approximation for the maximum likelihood estimates of moVMF for the M-step of EM in [2].

They are as follows:

$$\hat{\boldsymbol{\mu}}_i = \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|} \tag{12}$$

$$\hat{\kappa}_i = \frac{\bar{r}_i d - \bar{r}_i^3}{1 - \bar{r}_i^2} \tag{13}$$

$$\hat{\alpha}_i = \frac{1}{N} \sum_{j=1}^{N} q(i|\mathbf{x}_j, \boldsymbol{\theta}) \tag{14}$$

Here, $q(i|\mathbf{x}_j, \boldsymbol{\theta}) = \frac{\alpha_i f_i(\mathbf{x}_j|\boldsymbol{\theta})}{\sum_{l=1}^{k} \alpha_l f_l(\mathbf{x}_j|\boldsymbol{\theta})}$, $\mathbf{r}_i = \sum_{j=1}^{N} \mathbf{x}_j q(i|\mathbf{x}_j, \boldsymbol{\theta})$, and d is the dimensionality of the features. The quantity $\bar{r}_i$ is $\bar{r}_i = \frac{\|\mathbf{r}_i\|}{\alpha_i N}$.

Using these maximum likelihood estimates for the parameters of the mixture of vMF distributions, we can perform expectation maximization following the process detailed in the previous section.

4) Semi-Supervised moVMF EM Clustering: The unsupervised EM algorithm described in the previous sections assumes that we have no information about the prior distribution of our data. However, we know the true labels for a subset of our data, and we can leverage this information by performing a semi-supervised version of EM clustering with a mixture of vMF distributions.

The main principles of semi-supervised EM are very similar to those of unsupervised EM. The main difference is that in the E-step, we force the elements in the labeled dataset to be assigned to the correct clusters. This is done by forgoing the computations done in the unsupervised EM E-step and simply assigning the labeled data points a probability of 1 of belonging to the true cluster and a probability of 0 of being assigned to all other clusters. This allows us to impart our knowledge of the prior distribution of the data and avoid wasting time learning information we already know.

In addition to the changes made in the E-step, the maximum likelihood estimators in the M-step are updated to reflect the knowledge we have of the labeled data. The following changes are made to the maximum likelihood estimators for the mixture of vMF distributions.

$$\hat{\boldsymbol{\mu}}_i = \frac{\mathbf{r}'_i}{\|\mathbf{r}'_i\|} \tag{15}$$

$$\hat{\kappa}_i = \frac{\bar{r}'_i d - \bar{r}'^3_i}{1 - \bar{r}'^2_i} \tag{16}$$

$$\hat{\alpha}_i = \frac{1}{\|\mathcal{D}^u\| + \|\mathcal{D}_i^\ell\|} \Big(\sum_{j \in \mathcal{D}^u} q(i|\mathbf{x}_j, \boldsymbol{\theta}) + \|\mathcal{D}_i^\ell\| \Big) \tag{17}$$

Here, $\mathcal{D}^u$ is the set of indices of the unlabeled elements in the dataset and $\mathcal{D}_i^\ell$ is set of indices of the elements in the dataset that are labeled and have label i. $\mathbf{r}'_i = \sum_{j \in \mathcal{D}_i^\ell} \mathbf{x}_j + \sum_{j \in \mathcal{D}^u} \mathbf{x}_j q(i|\mathbf{x}_j, \boldsymbol{\theta})$. Similar to before, $\bar{r}'_i = \frac{\|\mathbf{r}'_i\|}{\alpha_i N}$.

Using this modified E-step and M-step, we can perform EM with a mixture of vMF distributions while leveraging our knowledge of the prior distribution of the labeled dataset.

5) Cluster Initialization: When performing Expectation Maximization, the initial parameters chosen for each cluster can heavily influence the algorithm's performance. In the unsupervised case, there are many ways of initializing $\boldsymbol{\mu}$. One option is to randomly sample points that lie on the unit hypersphere and use those as the initial μ's. Another more robust option is to choose the μ's using k-means++ initialization [1]. With no information about the prior distribution of any elements in the dataset, it is logical to initialize all of the κ's to the same value. In a similar vein, it makes sense for all α's to be initialized to the same value, $\frac{1}{k}$.

When dealing with semi-supervised moVMF EM clustering, we are able to use the same parameter initialization techniques as in the unsupervised case, but we have some additional options that leverage the distribution of the labeled data. For all clusters that correspond to labeled data, we can initialize their parameters using maximum likelihood estimation. For all other clusters, we can initialize the parameters based on the techniques described for the unsupervised case.

IV. Experiments

A. Experimental Setup

For our experiments, we will be using the publicly available WHOI-Plankton dataset provided by the Woods Hole Oceanographic Institution (WHOI) [8]. The images were collected at the Martha's Vineyard Coastal Observatory in Massachusetts and contain labeled images of over 100 different species of plankton with individual datasets every year from 2006 to 2014.

Our training data consists of all the images from 2013, and our testing data the images from 2014. For simplicity, we choose to exclude all images from the "mix" class, all of those contain multiple different plankton species in a single image. This brings the total number of plankton species in our training and testing data to 103. There are a total of 115951 images in our training dataset and 63676 in our testing dataset. In order to reduce the computational complexity of training the feature extractor, all images were converted from RGB to grayscale.

To artificially create labeled and unlabeled classes for GCD, a random subset of 50 of the 103 classes were chosen to be "labeled classes". The labeled subset of our dataset is all of the images with a label that is one of the labeled classes. All other images are part of the unlabeled dataset.

It is important to note that the WHOI-Plankton dataset is incredibly unbalanced. Between the 2013 and 2014 datasets, Detritus, the largest class, makes up almost half of the data. On the contrary, the smallest classes have less than 10 examples each.

B. Feature Extraction

The backbone model used is a ResNet-18 pre-trained with DINO on ImageNet. The backbone is fine-tuned on the training data with contrastive representation learning using the semi-supervised InfoNCE loss function and cosine similarity.

Once fine-tuned, all images in the training and testing datasets were passed through the feature extractor. The features yielded by this process are 512-dimensional vectors.

As mentioned before, the vMF distribution is only valid for unit-length input vectors. As a result, we normalize all of the feature vectors output by the feature extractor so that they are all of unit length.

C. Clustering Results

We report results on the performance of different clustering algorithms trained on the training dataset and evaluated on the testing dataset in Tables I and II. More specifically, we analyze the following clustering algorithms: k-means++, spherical k-means, semi-supervised k-means, unsupervised EM clustering with a moVMF, and semi-supervised EM clustering with moVMF. The semi-supervised k-means algorithm is the same as described in [10], the original GCD paper, and forces the labeled data to be assigned to its true cluster every iteration. An important distinction is that for the spherical k-means and all moVMF EM clustering algorithms, all features were normalized to unit-length.

For the three k-means based algorithms, the centroids are initialized using k-means++ initialization. For the EM algorithms that use k-means++ initialization, the μ's are initialized with k-means++ initialization, the κ's are initialized to 1, and the α's are initialized to $\frac{1}{k}$. For the semi-supervised moVMF EM algorithm with MLE estimation, maximum likelihood estimation is used to initialize the parameters of the clusters corresponding to labeled data. For the clusters corresponding to unlabeled data, k-means++ initialization is used to initialize the μ's, and the κ's are set to 1.

The clustering accuracy is computed by performing the Hungarian algorithm to find an optimal assignment of predicted labels to true labels, maximizing accuracy. Accuracies for the labeled and unlabeled subsets of the test data are provided for semi-supervised clustering algorithms that make use of the labels in the labeled training set.

From the data in Table I, we can clearly see that in terms of accuracy, the semi-supervised clustering algorithms outperform the unsupervised clustering algorithms. Within the semi-supervised algorithms, it is evident that the semi-supervised moVMF EM algorithm with k-means++ initialization provides the strongest performance. It outperforms the semi-supervised k-means algorithm by a few percentage points in all categories. Interestingly, the semi-supervised moVMF EM algorithm with MLE initialization boasts the highest labeled accuracy but the lowest overall and unlabeled accuracies. The high labeled accuracy is likely due to the initialization ensuring the labeled clusters start very close to their optimal positions.

Looking at the data in Table II, we can see that similar to the accuracy comparison, the semi-supervised clustering algorithms clearly outperform the unsupervised algorithms in terms of Macro F1. However, this time, the semi-supervised k-means achieves better scores than either of the semi-supervised EM algorithms. The difference in labeled Macro F1 scores is not very different across the three algorithms, but semi-supervised k-means is the achieves much better overall and unlabeled Macro F1 scores than the other two algorithms. The higher accuracies and lower Macro F1 scores of the semi-supervised moVMF EM algorithms seem to suggest that they are better at clustering medium to large classes than semi-supervised k-means but worse at clustering small classes. This would make sense, as EM allows for clusters of various sizes while k-means works best when all the clusters are similar in size. The plankton data is very unbalanced, so it makes sense that the moVMF EM clustering would be able to pick up on the larger clusters better.

D. Concentration Parameter (κ) Analysis

In this section, we examine the set of concentration parameters for the labeled data obtained after performing EM clustering with a mixture of vMF distributions. Figures 1 and 2 show comparisons of the concentration parameters obtained from various configurations of moVMF EM clustering sorted in ascending order in the labeled and unlabeled datasets, respectively. The blue line represents the set of concentration parameters that best fit the training data. These were obtained by performing maximum likelihood estimation on each class in the datasets.

This analysis will provide insight as to whether or not the predicted clusters seem to be of appropriate sizes given the dataset. If a clustering algorithm's line is very similar to the blue estimated κ's, we can infer that the predicted clusters are of an appropriate size. If they are consistently larger or smaller, then we can assume that the clustering algorithm is significantly underestimating or overestimating the sizes of the classes, respectively.

1) Labeled Dataset: From Figure 1, we can see that unsupervised moVMF EM algorithm consistently yields concentration parameters that are significantly higher than we would expect. However, the semi-supervised moVMF EM algorithm with both k-means++ and MLE initialization produce κ values that are much closer to what is expected. This makes sense, as they leverage the true labels of the labeled dataset during the clustering process. We can also see that for the semi-supervised algorithms with both initializations, the bottom 50% of kappas are almost identical. As smaller values of κ correspond to larger clusters, this appears to support the idea that the disparity between clustering accuracy and Macro F1 for the semi-supervised moVMF EM algorithms can be largely attributed to them performing relatively well on medium to large clusters and poorly on smaller clusters. The figure shows that the larger κ values are almost always overestimated by a significant margin, which indicates that the smaller clusters found by the algorithm are actually just subsets of the true clusters.

TABLE I: Comparison of Clustering Algorithms (Accuracy)

Algorithm	Initialization	Total Accuracy	Labeled Accuracy	Unlabeled Accuracy
k-means++	k-means++	0.2490	-	-
Spherical k-means	k-means++	0.2368	-	-
Semi-Sup. k-means	k-means++	0.2562	0.3656	0.2369
Unsup. EM w/ moVMF	k-means++	0.2417	-	-
Semi-Sup. EM w/ moVMF	k-means++	**0.2881**	0.4167	0.**2686**
Semi-Sup. EM w/ moVMF	MLE	0.2419	**0.4207**	0.2136

TABLE II: Comparison of Clustering Algorithms (Macro F1 Score)

Algorithm	Initialization	Total Macro F1	Labeled Macro F1	Unlabeled Macro F1
k-means++	k-means++	0.0940	-	-
Spherical k-means	k-means++	0.0976	-	-
Semi-Sup. k-means	k-means++	**0.1276**	0.1123	**0.1105**
Unsup. EM w/ moVMF	k-means++	0.0943	-	-
Semi-Sup. EM w/ moVMF	k-means++	0.1085	0.1089	0.0877
Semi-Sup. EM w/ moVMF	MLE	0.1142	**0.1126**	0.0965

E. Unlabeled Dataset

Figure 2 shows a similar comparison, this time with the unlabeled classes. In this figure, it appears that the concentration values are almost always significantly overestimated. All three clustering algorithms seem to produce similar κ values, which is expected. This is because the semi-supervised methods are unable to leverage any prior data distribution to estimate the parameters of these classes, so they fall back to the unsupervised case. There are many reasons as to why these concentrations may be so consistently large. One is that the feature extractor may be producing fragmented clusters, and the moVMF clustering algorithms are fitting the vMF distributions to the fragments. Another possible reason is that the maximum likelihood estimator for κ from [2] may be prone to overestimating the values every iteration.

The consistent underestimating of the cluster sizes largely contributes to the discrepancies between the unlabeled clustering accuracies and macro f1 scores. Because the predicted concentrations are so much higher, the smaller clusters are likely not being predicted correctly at all, whereas with the larger clusters, it is likely that subsets of the true clusters are being correctly predicted.

V. Conclusion

In this paper, we have presented an approach to the generalized category discovery problem for novel plankton species using a combination of deep learning-based feature extraction and semi-supervised clustering. From our experiments on the WHOI-Plankton dataset, it seems that semi-supervised Expectation Maximization clustering with a mixture of von Mises-Fisher distributions shows promise, and with some additional optimizations could see even better results.

The semi-supervised moVMF EM algorithm with k-means++ initialization consistently outperforms other clustering methods in terms of accuracy. However, it achieves slightly underwhelming Macro F1 scores, indicating that it has a higher capability of handling medium to large clusters but a relative weakness in managing smaller clusters.

Analysis of the concentration parameters (κ) further supports these findings, revealing that the semi-supervised moVMF Em algorithms, while seemingly effective in clustering larger groups, tend to overestimate the concentrations for the smaller clusters. This indicates a potential area for improvement, where fine-tuning the maximum likelihood estimators might enhance the algorithm's performance on underrepresented classes.

Another aspect that if improved could have a significant impact on the clustering accuracy is the feature extractor. Currently, we use a ResNet-18 as the backbone, which is far from ideal. State of the art GCD methods employ the use of much more powerful models like ViT-B/16 or ResNet-50 [7]. Using one of these models instead could drastically improve results by producing embeddings with much clearer class separation. Analysis of the tSNE visualizations between different base models conducted in [10] shows just how impactful the backbone architecture can be.

Finally, instead of randomly sampling classes to be treated as labeled data. We may see benefits in biasing the labeled data selection process to favor larger classes. This would better reflect the distribution of the known and novel classes in practical applications and likely boost the clustering accuracies

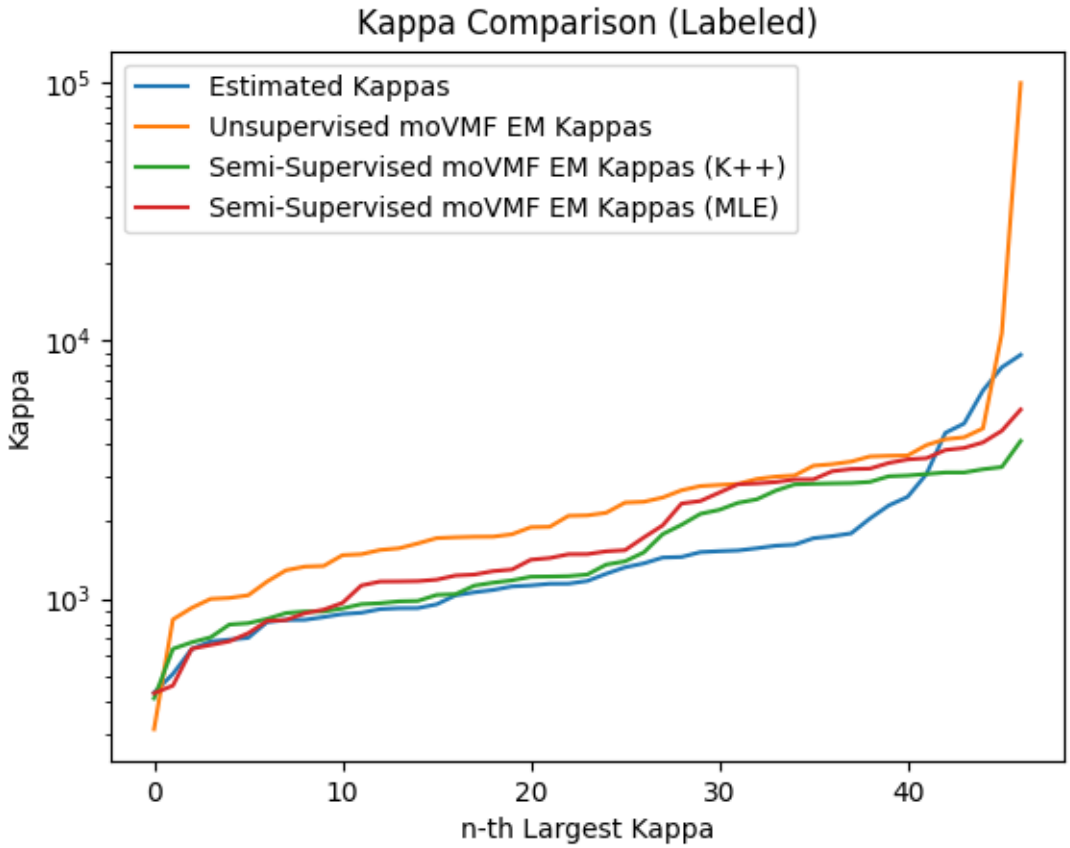

Fig. 1: Comparison of Concentration Parameters (κ)

achieved for the labeled data by semi-supervised clustering methods.

Overall, our proposed approach shows significant promise to assist with automated and scalable plankton classification, which would contribute valuable insights and tools for ecological monitoring and research. Future work could focus on refining the cluster initialization, parameter estimation, and feature extraction process to further enhance classification accuracy.

References

[1] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. volume 8, pages 1027–1035, 01 2007. doi: 10.1145/1283383.1283494.

[2] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 09 2005.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 12 2018. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1977.tb01600.x.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[8] Eric C. Orenstein, Oscar Beijbom, Emily E. Peacock, and Heidi M. Sosik. Whoi-plankton- a large scale fine grained visual recognition benchmark dataset for plankton classification, 2015.

[9] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

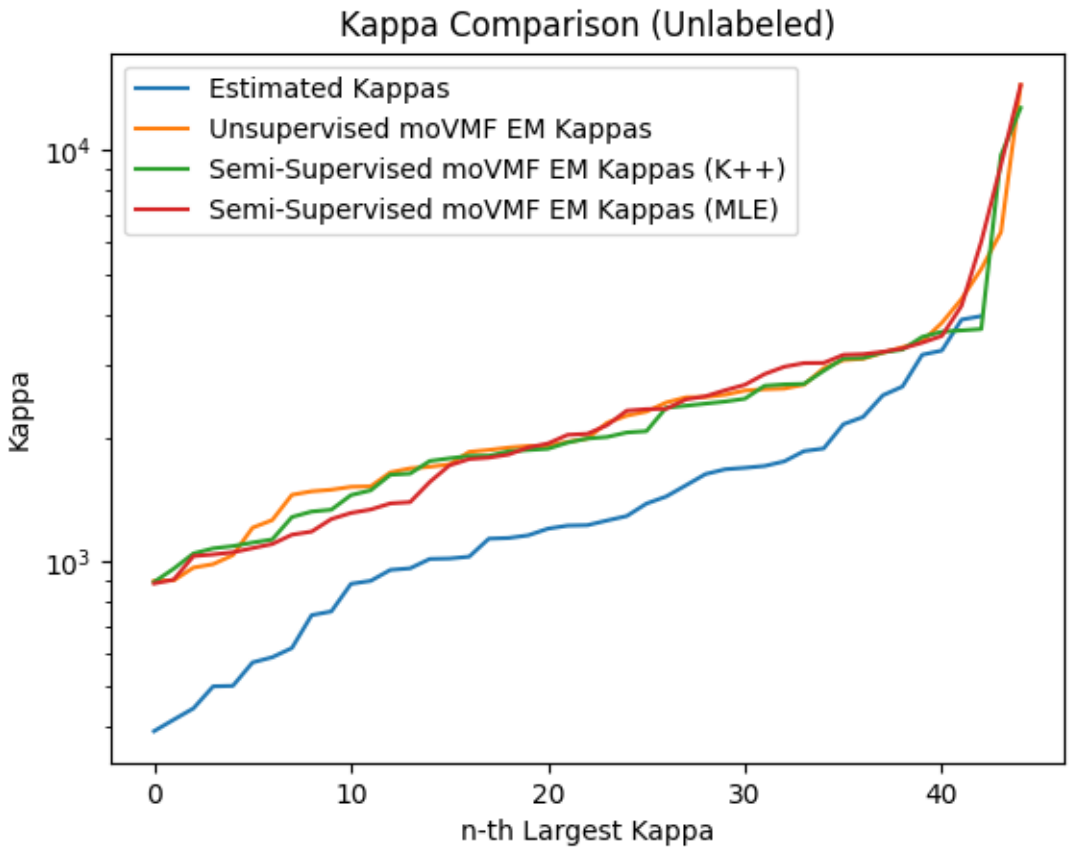

Fig. 2: Comparison of Concentration Parameters (κ)

[10] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. *CoRR*, abs/2201.02609, 2022. URL `https://arxiv.org/abs/2201.02609`.

[11] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study, 2023.