Diffusion-Based Sampling for Deep Active Learning

***Abstract*—The remarkable performance of deep neural networks depends on the availability of massive labeled training data. To alleviate the load of data annotation with labels, deep active learning aims to sample a minimal set of training points to be labelled which yields maximal model accuracy. We propose an efficient sampling criterion to sample data for annotation, which automatically shifts from an exploration type of sampling to a class-decision-boundary refinement. Our criterion relies on a process of diffusing the existing label information over a graph constructed from the hidden representation of the data. This graph representation captures the intrinsic geometry of the approximated labeling function. We analyze our sampling criterion and its exploration - refinement transition in light of the eigen-spectrum of the diffusion operator. Additionally, we provide a comprehensive sample complexity analysis that captures the two phases of exploration and refinement. The diffusion-based sampling criterion is shown to be advantageous over state-of-the-art criteria for deep active learning on synthetic and real benchmark data.**

I. Introduction

Deep learning has provided unprecedented performance in various semi-supervised learning tasks ranging from speech recognition to computer vision and natural language processing. Deep Convolutional Neural Networks (CNN), in particular, have demonstrated object recognition that exceeds human's performance. However, this success comes with the requirement for massive amounts of labelled data. While data collection at large scale has become easier, its annotation with labels has become a bottleneck for execution in many real-life problems.

Active learning provides a plethora of techniques that allow to sample a set of data points for labeling, which optimally minimizes the error probability under a fixed budget of a labeling effort (see [16] for review). A well known trade-off in active learning is between the exploration and refinement sampling strategies. Exploration aims at mapping the joint data and label distribution in order to identify decision boundaries, and typically yields higher model accuracy at the earlier stage of active learning over other baselines criteria. Refinement (also referred to as exploitation), on the other hand, samples labels at the proximity of a discovered class decision-boundary, improving its localization. Active learning exhibits improved performance when the balance between exploration and refinement is optimal.

Incorporating active learning into deep learning is yet a challenging task for several reasons. First, the network representation does not allow to construct a simplistic probabilistic sampling criterion that incorporates exploration and refinement. Thus, state-of-the-art focus on either exploration or on a refinement type criteria, which often leads to sub-optimal results. This state drives active deep learning methodology to focus mostly on simple refinement-type uncertainty sampling criteria (e.g. [5, 17]), which do not require using the classifier at the querying stage. Or, on using geometric criteria (e.g. [15]) that only explores the data distribution in its feature space.

We propose a graph Diffusion-based Deep Active Learning (DDAL) criterion. We utilize the graph diffusion process, learnt on a graph constructed from the penultimate layer, to derive our active sampling criterion. The penultimate representation is highly correlated with the labeling function, and hence provides an optimal representation to sample the labeling function. The main contributions of DDAL are:

1) **Exploration-refinement trade-off**: We derive a sampling criterion which exhibits an automatic switch from exploration to refinement-based sampling. Our criterion explores graph nodes that are unreachable via diffusion, and then refines the boundary where diffused labels are ambivalent. This natural switch avoids the need in active learning for an additional optimization machinery (e.g. [19]) to decide which criterion type to use. We analyze the exploration-refinement switch in light of the eigen-spectrum of the diffusion iterant. We provide a sketch of the application of our method in Fig 1.
2) **Sample complexity analysis**: We provide sample complexity analysis for the two phases of exploration and refinement. The analysis depicts complexity that is logarithmic in N, and linear in the decision boundary set.
3) **Empirical validation:** DDAL competitive performance is demonstrated empirically on benchmark data and compared with eight state-of-the-art active learning criteria. In particular, those constructed for deep learning.

II. Problem Set-up

Let $\mathcal{D}$ be a probability distribution over $\mathbb{R}^d \times [C]$, where $[C] = \{1, ..., C\}$ and $C \geq 2$ indicates the number of classes. Our aim is to find a classifier $f : \mathbb{R}^d \to \Delta_C$ (Δ_C denotes the space of probability measures over $[C]$) that minimizes the classification error probability

$$\mathcal{E}(f) = \mathbb{P}_{(x,y)\sim\mathcal{D}}\left\{(\arg\max_{c\in[C]} f_c(x)) \neq y\right\}.$$

In deep learning, we consider the parametric functions

$$f_\theta = g_{\theta_{n+1}} \circ h^n_{\theta_n} \circ \cdots \circ h^1_{\theta_1},$$

where $h^i_{\theta_i} : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$ (with $d_0 = d$) and $g_{\theta_{n+1}} : \mathbb{R}^{d_n} \to \Delta_C$. The parameter $\theta = \{\theta_i\}_i$ defines our final model f_θ.

Active learning setting. In active learning, one is given a *pool* of input data points $X_{\text{pool}} = \{x_i\}_{i=1}^N$, and a budget Q to select for labeling. Optionally, a labelled subset $X_\ell \subset X_{\text{pool}}$ may also be given as input. In the following, we denote $X_u \doteq X_{\text{pool}} \backslash X_\ell$

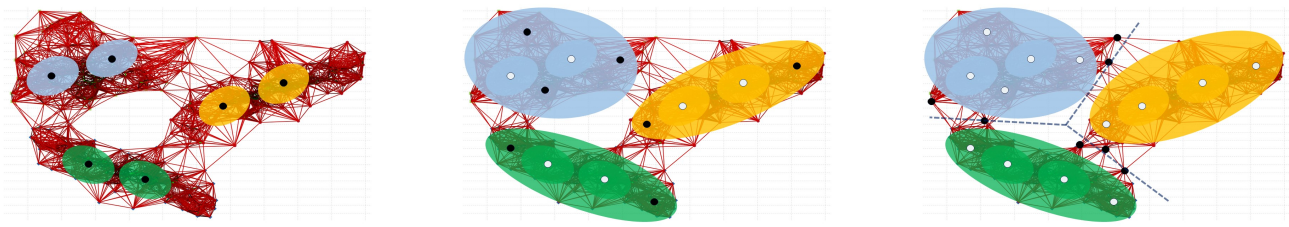


Figure 1: **Schematic evolution of diffusion-based active sampling on the graph $G(V, E, W)$ constructed on the penultimate layer**. The graph in red represents the constructed graph over all points $X_\ell \cup X_{\text{pool}}$ Black points - current samples, white points - previous samples, ovals - area of low uncertainty in class label for 3 different color coded labels. Left: active sampling in early exploration stage. Center: advanced exploration stage of sampling captures the label related clusters. Right: Refinement (exploitation) stage: black samples focus on the decision boundary between class where highest uncertainty prevails.

as the set of unlabeled points in X_{pool}. The aim of active learning is to minimize the error $\mathcal{E}(f_\theta)$ while querying only Q points from X_{pool}, and training f_θ with them.

III. CLASSIFICATION VIA GRAPH DIFFUSION

We use the optimized latent representation to construct a KNN-based weighted proximity graph $G = (V, E)$, where

$$W_{ij} = m\left(-\frac{\rho(f_\theta^n(x_i), f_\theta^n(x_j))}{\sigma_{ij}}\right) I\{j \in N(i)\} \tag{1}$$

with m as a similarity metric, and ρ as a distance metric. In our setting $\rho(\rho(f_\theta^n(x_i), f_\theta^n(x_j)) = \|f_\theta^n(x_i) - f_\theta^n(x_j)\|_2^2$, $\sigma_{ij} = \max_{j\in N(i)} \rho(\rho(f_\theta^n(x_i), f_\theta^n(x_j))$, and $N(i)$ as the set of K-NN neighbours of $f_\theta^n(x_i)$. We define a graph transition matrix $P = D^{-1}W$ which stands for the transition probabilities of a Markov random walk on G. The matrix D is a diagonal where $D_{ii} = \sum_j W_{ij}$.

Label diffusion can be considered as a Markov process to propagate the label information of X_l to X_u. The transition probability of one step between state i and j can be denoted as P_{ij}. To facilitate notation, consider a binary classification, we associate the classification probability $P(y(x_i) = 1|f_\theta^n(x_i))$ with $P_t(y(x) = 1|i)$. Namely, a label to $f_\theta^n(x_i)$ is assigned by following a t-step random walk with the above transition probabilities.

Let $\chi_i = 2P_t(y(z) = 1|i) - 1 \in [-1, 1]$. The signal of χ stands for the binary class. Denote

$$D = \begin{pmatrix} D_{ll} & 0 \\ 0 & D_{uu} \end{pmatrix}, W = \begin{pmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{pmatrix}.$$

In matrix form $\chi_u = [D_{uu}^{-1}W_{ul}|D_{uu}^{-1}W_{uu}] \begin{pmatrix} \chi_l \\ \chi_u \end{pmatrix}$.
Let the graph laplacian $L = D - W$ in the system, then

$$L_{uu}\chi_u = W_{ul}\chi_l \Longleftrightarrow L_{uu}X_{uu} = -L_{ul}y_l. \tag{2}$$

(2) above can be solved via the iteration:

$$\chi_i^{(t+1)} = \frac{1}{L_{uu,ij}}\left(-(L_{ul}y_l)_i - \sum_{j\neq i} L_{uu,ij}\chi_j^{(t)}\right). \tag{3}$$

(3) is transducing a label χ_i^{t+1} to χ_i as a weighted average of the labels of its neighbors with the transition weights. The labels are propagated to X_u gradually for t steps. At the t-th step,

$$\chi_{ic}^{(t)} = \begin{cases} 1 & \text{if } x_i \in X_l \ \& \ c = y_i \\ -1 & \text{if } x_i \in X_l \ \& \ c \neq y_i \\ 0 \ if \ t = 0, \ else \ (M\chi_{:,c}^{(t-1)})_i & \text{if } x_i \in X_u. \end{cases}$$

We provide the multi-class case in the supplementary material.

IV. ACTIVE LEARNING

We begin with introducing our query selection criterion and prove its important transition from exploration to refinement.

A. Query criterion.

The matrix $\chi^{(T)}$ of propagated values can be interpreted as *uncertainties* measured by the absolute value $|\chi_{c,i}^{(T)}|$. Specifically, the absolute value magnitude represents a measure of uncertainty on whether vertex i belongs to class c. The magnitude can be used to select a new batch to query as

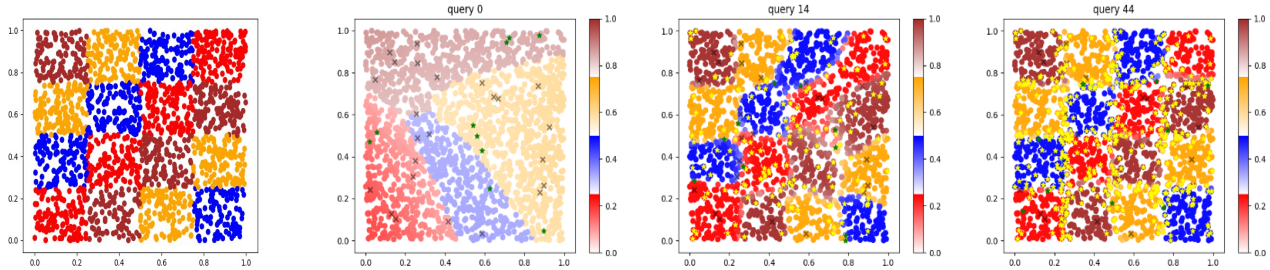
$$\hat{X} = \arg\min_{i\in X_u}^{B} \min_{c\in[C]} \left|\chi_{c,i}^{(T)}\right|, \tag{4}$$

where $\min^B$ denotes the B smallest elements. The selection criterion in (4) fits the one-vs-all approach above. In supplementary material other variants are suggested.

B. Exploration and Refinement.

The query criterion (4) coupled with the diffusion process allows to explore the data set at early stages of active learning and switch to refinement when exploration has saturated. We prove in the following that the diffusion iterant χ^t converges to the second eigenvector ϕ_2 of the graph's Laplacian as $t \to \infty$. As we show below, asymptotically ϕ_2 provides a relaxed solution to the minimal normalized cut problem, where the cut corresponds to the decision boundary between the two classes in $G(V, E, W)$.

At early stages of label acquisition low magnitude entries in χ correspond to data points that are unreachable from the training set via diffusion and need to be explored. At later stages all unlabelled data points $f_\theta^n(X_u)$ are reachable via diffusion



Ground truth | First query batch | 14 query batches | 44 query batches

Figure 2: Exploration & Refinement in Multiclass Checkerboard. Left to right: Ground truth, 1st batch (exploration, 30% accuracy), 14 batches (exploration, 82% accuracy) and 44 batches (Refinement, 95% accuracy). Color coding corresponds to label assignment and its intensity level to the magnitude of the diffused labels. Light yellow points represent so-far queried points, black points represent current query. Transition from exploring the distribution in (0-14 batches) to focus on boundaries (44) batches

from the labelled set $f_\theta^n(X_l)$. At this stage low magnitude χ entries correspond to the transition between the two classes -1 and 1. These nodes capture the eigenvector's transition from negative to positive entries. Therefore, sampling these points corresponds to the refinement of the decision boundary. We provide the convergence result to the second eigenvector of the graph Laplacian in Lemma (1):

Lemma 1. *Let $\lambda_1, ..., \lambda_n$, $\phi_1, ..., \phi_n$ be the solutions to the system: $L\phi = (1-\lambda)D\phi$. Then $\chi^{(t)}$ converges to ϕ_2 via the iteration (8) with M, as $t \to \infty$.*

The eigenvector ϕ_2 is a solution to the relaxed Minimal Normalized Cut problem (*MinNCut*) [3] in a graph $G(V, E, W)$ constructed from data representation in the hidden layer f_θ^n:

$$MinNCut = \arg\min_{h:h^T D\mathbf{1}=0} \frac{h^T Lh}{h^T Dh}. \tag{5}$$

We demonstrate the utility of our criterion in switching from exploration to refinement on a checkerboard toy example - Fig. 2. Exploration is observed to become saturated, as the joint data-label distribution is sufficiently mapped. At that stage DDAL automatically switches to refining the decision boundaries. We provide the pseudo code of DDAL in algorithm 1, for sampling batches of query points via our criterion.

V. SAMPLE COMPLEXITY ANALYSIS

We analyze two fundamental graph structures that model the two stages of exploration and refinement in the hidden layer graph representation. We address the binary setting to facilitate the analysis. The first case addresses the early exploration stage, as the hidden layers representation is not sufficiently adapted to the class function, and therefore separation between classes is not detectable. We model this setting as a K-regular graph or a grid (similarly to the work of [4]). The binary class function divides the grid to two regions in which each class resides. Localization of the decision boundary is still not achieved by the learner at this exploratory stage, as we show that the query complexity of a full recovery of the graph-cut scales linearly in the size of the boundary set and logarithmic in N.

Algorithm 1 DDAL

input: data $X_{\text{pool}} = \{x_i\}_{i=1}^N$, labeled subset X_ℓ^0, model f_θ trained on X_ℓ^0, Q, B, M, T
compute K-NN graph $G = (V, E, W)$ using the top hidden layer representations $\{f_\theta^n(x_i)\}_{i=1}^N$
while budget Q is not exhausted: **do**
  initialize $\hat{X}_\ell^0 = \emptyset$
  for $i = 1$ **to** $R = \frac{Q}{B}$ **do**
    initialize $\chi^{(0)}$ according to $X_\ell^0 \cup \hat{X}_\ell^{i-1}$
    for $t = 1$ **to** T **do**
      $\chi^{(t)} = M\chi^{(t-1)}$
      $\chi^{(t)} = \chi^{(t-1)}|_{(X_\ell^0 \cup \hat{X}_\ell^{i-1})}$
    end for
    query $\mathcal{S} = \arg\min_{k \in \hat{X}_u^i}^B \min_{c \in [C]} \left|\chi_{c,k}^{(T)}\right|$
    update $\hat{X}_\ell^i = \mathcal{S} \cup \hat{X}_\ell^{i-1}$
  end for
  retrain f_θ with $\hat{X}_\ell^i$
end while

The second case captures a learning stage of a converging network with a larger training set at the mature refinement stage. In this phase, well-separated clusters emerge in the hidden layer representation, which correspond to label assignments.

A. The non-separable grid case.

Definition 1. A **d-dimensional weighted grid graph** is defined as the weighted graph $G(V, E, W)$ on the $\sqrt[d]{N} \times ... \times \sqrt[d]{N}$ grid of N nodes constructed via the graph Cartesian product on the $\sqrt[d]{N}$ line graphs. The edge weights W_{ij} are fixed.

Definition 2. The **balancedness** β of a labelled set $C = \{(x_i, y_i)\}_{i=1}^N$ is defined as $\beta = \min_j \frac{|C_j|}{N}$.

Theorem 1. *Let the graph $G(V, E, W)$ correspond to a d-dimensional regular grid, and assume that the grid is partitioned by the cut set C of size $|\partial C|$ with balancedness β. Then the sample complexity of $BatchQuery$ for recovering*

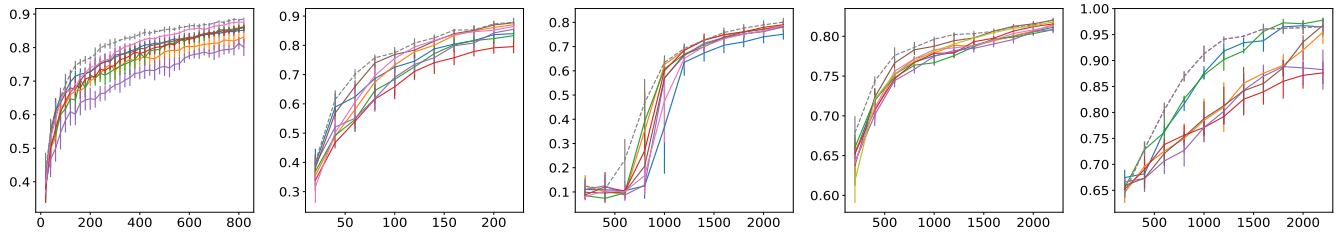


Figure 3: Accuracy vs. queried points. From left to right: MNIST (fully connected), MNIST (convolutional), SVHN, CIFAR10, and Openml155 datasets. The x-axis represent the labels queried, while the y-axis represents test set accuracy. Legend: (–) random, (–) uncertainty, (–) coreset, (–) bayes-entropy, (–) entropy, (–) margin, (–) bayes-uncertainty, (–) badge, (- -) diffusion.

the cut with probability at least 1 - δ for any $\delta > 0$ is

$$\frac{\log \frac{1}{\beta\delta}}{\log \frac{1}{1-\beta}} + O(d \log_K N + |\partial C|). \tag{6}$$

B. The separable case.

At the advanced refinement stage the hidden layer representation has structure comprising separated clusters that correspond to different label assignments. The K-NN graph, in this case, has a minimal cut that corresponds to the decision boundary between the two classes. We analyze the query complexity of well-separated class samples via a Gaussian Mixture Model (GMM) $w_1\mathcal{N}(m_1, \Sigma_1) + w_2\mathcal{N}(m_2, \Sigma_2)$ modelling two densities in the hidden layer representation. The separation S is defined as the minimal distance between the centroids, relative to the covariance:

$$S = \frac{\|\mu_i - \mu_j\|}{\sqrt{d}\left(\sqrt{\lambda_{\max}(\Sigma_i)} + \sqrt{\lambda_{\max}(\Sigma_j)}\right)}. \tag{7}$$

Theorem 2. *Consider a Gaussian Mixture Model of 2 well-separated ($S >> 1$) Gaussians of balancedness β. Let the graph $G(V, E, W)$ be the KNN graph constructed from the GMM samples with weights according to eq. (1). Then $BatchQuery$ will recover the decision boundary with probability $1 - \delta$ by querying $\log \frac{1}{\beta\delta} \left(\log \frac{1}{1-\beta}\right)^{-1}$ points.*

VI. EXPERIMENTS

Benchmark evaluation. In Figure 3 we provide a performance comparison with other active learning methods. We selected fundamental approaches: refinement (e.g. uncertainty [10] and entropy [6]), exploratory (e.g. Coreset [15]), and a combined approach (e.g. 'margin' [14, 11] and Badge [1, 2]). Our selections were guided by i) comparing the exploration-refinement trade-off between different active learning approaches, and ii) comparing deep active learning approaches. We experiment with the following benchmark classification problems: MNIST, CIFAR10, Openml155 [18], and SVHN. Additional experimental details are reported in the appendix.

The advantage of DDAL is prominent during the early exploration and transition to refinement. The competitive accuracy persists into the longer tail of the refinement stage.

Robustness study. We examine the robustness of DDAL and state-of-the-art for the choice of batch size and number of epochs in Figure 4. For large batch size the criterion in (4) becomes more exploratory. Namely, it probes points with larger distance between them. This is more advantageous on the early stages of learning. Small batches may focus more on refinement of the decision boundary and may give better gains at later stages. On the other hand, when the batches are small they are revisited more often by the optimizer in each active learningiterations, and therefore faster convergence is observed.

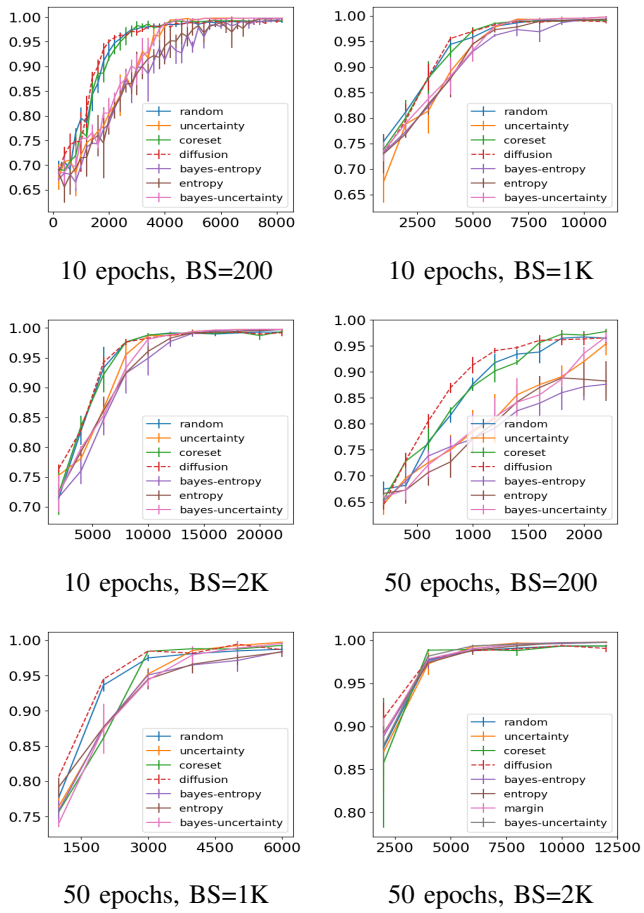


Figure 4: Effect of query size & epochs on performance in Openml 155.

References

[1] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Project. https://github.com/JordanAsh/badge, 2019.

[3] F. R. K Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, 92 edition, 1996.

[4] X. Zhu G. Dasarathy, R. Nowak. S2: An efficient graph based active learning algorithm with application to nonparametric classification. In *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 503–522. PMLR, 03–06 Jul 2015.

[5] Islam Riashat Ghahramani Zoubin Gal, Yarin. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[6] Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[7] Szegedy C. Ioffe, S. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[8] A. Krishnamurthy J. Langford A. Agarwal J. T. Ash, Chicheng Z. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

[9] Ba J. Kingma, D. P. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[10] David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pages 13–19. ACM New York, NY, USA, 1995.

[11] Tong Luo, Kurt Kramer, Dmitry B Goldgof, Lawrence O Hall, Scott Samson, Andrew Remsen, Thomas Hopkins, and David Cohn. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6(4), 2005.

[12] H. H. Zhou M. Löffler, A. Y. Zhang. Optimality of spectral clustering for gmm, 2019.

[13] Gross S. Chintala S. Chanan G. Yang E. DeVito Z. Lin Z. Desmaison A. Antiga L. Lerer A. Paszke, A. Automatic differentiation in PyTorch. In *NIPS Workshop*, 2017.

[14] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer, 2001.

[15] Savarese S. Sener, O. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[16] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[17] Hazırbas C. Triebel R. Cremers D. Stark, F. Captcha recognition with active deep learning. In *Workshop new challenges in neural computation*, page 94, 2015.

[18] Jan van Rijn. Normalized version of the pokerhand data set.

[19] Ma Z. Nie F. Chang X. Hauptmann A. G. Yang, Y. Multi-class active learning by uncertainty sampling with diversity maximization. *Int. J. Comput. Vision*, 113(2):113–127, June 2015.