

Sequential Graph Convolutional Network for Active Learning

Razvan Caramalau¹, Binod Bhattarai¹ and Tae-Kyun Kim^{1,2}

¹Imperial College London, UK

²KAIST, South Korea

{ r.caramalau18, b.bhattarai, tk.kim}@imperial.ac.uk

Abstract

We propose a novel pool-based Active Learning framework constructed on a sequential Graph Convolution Network (GCN). Each image feature from a pool of data represents a node in the graph and the edges encode their similarities. With a small number of randomly sampled images as seed labelled examples, we learn the parameters of the graph to distinguish labelled vs unlabelled nodes by minimising the binary cross-entropy loss. GCN performs message-passing operations between the nodes, and hence, induces similar representations of the strongly associated nodes. We exploit these characteristics of GCN to select the unlabelled examples which are sufficiently different from labelled ones. To this end, we utilise the graph node embeddings and their confidence scores and adapt sampling techniques such as CoreSet and uncertainty-based methods to query the nodes. We flip the label of newly queried nodes from unlabelled to labelled, re-train the learner to optimise the downstream task and the graph to minimise its modified objective. We continue this process within a fixed budget. We evaluate our method on 6 different benchmarks: 4 real image classification, 1 depth-based hand pose estimation and 1 synthetic RGB image classification datasets. Our method outperforms several competitive baselines such as VAAL, Learning Loss, CoreSet and attains the new state-of-the-art performance on multiple applications.

1. Introduction

Deep learning has shown great advancements in several computer vision tasks such as image classification [15, 22] and 3D Hand Pose Estimation (HPE) [26, 41, 25]. This has been possible due to the availability of both the powerful computing infrastructure and the large-scale datasets. Data annotation is a time-consuming task, needs experts and is also expensive. This gets even more challenging to some of the specific domains such as robotics or medical image analysis. Moreover, while optimizing deep neural network architectures, a gap is present concerning the representative-

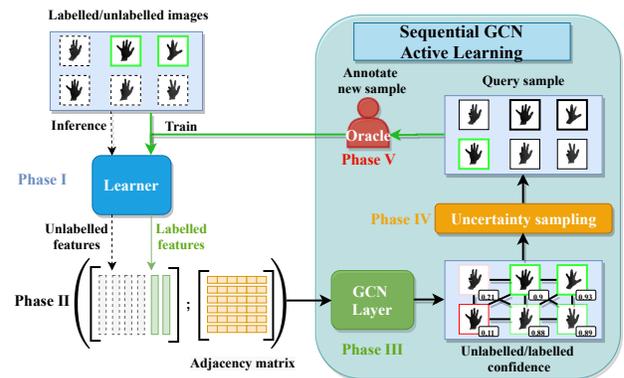


Figure 1: The diagram outlines the proposed pipeline. **Phase I:** Train the learner to minimise the objective of downstream task from the available annotations, **Phase II:** Construct a graph with the representations of images extracted from the learner as vertices and their similarities as edges, **Phase III:** Apply GCN layers to minimise binary cross-entropy between labelled and unlabelled, **Phase IV:** Apply uncertainty sampling to select unlabelled examples to query their labels, **Phase V:** Annotate the selection, populate the number of labelled examples and repeat the cycle.

ness of the data [4]. To overcome these issues, *active learning* [10, 20] has been successfully deployed to efficiently select the most meaningful samples.

Essentially, there are three distinct components in any Active Learning (AL) framework. These components are *learner*, *sampler*, and *annotator*. In brief, a *learner* is a model trained to minimize the objective of the target task. The *sampler* is designed to select the representative unlabelled examples within a fixed budget to deliver the highest performance. Finally, an *annotator* labels the queried data for learner re-training. Based on the relationship between learner and sampler, AL frameworks can be categorised into two major groups: task-dependent and task-agnostic. Task-dependents are the ones where the sampler is specially designed for a specific task of the learner. Majority of the previous works in active learning [8, 10, 1, 38, 6, 11] are

task-dependent in nature. In other words, the sampling function is dependent on the objective of the learner. This design limits the model to become optimal for a specific type of task while suffering from its scalability problem. Some of the recent works such as VAAL [33] and Learning Loss [42] tackle such a problem. VAAL trains a variational auto-encoder (VAE) that learns a latent space for better discrimination between labelled and unlabelled images in an adversarial manner. Similarly, Learning Loss introduces a separate loss-prediction module to be trained together with the learner. The major drawback of these approaches is the lack of a mechanism that exploits the correlation between the labelled and the unlabelled images. Moreover, VAAL has no way to communicate between the learner and the sampler. Graph Convolutional Networks (GCNs) [18, 5] are capable of sharing information between the nodes via message-passing operations. In the AL domain, the application of GCNs [6, 11, 1, 38] is also slowly getting priority. However, these methods are applied only to specific kind of datasets i.e. graph-structured data such as Cora, CiteSeer, and PubMed [40]. In this work, we are exploring the image domain beyond graph-structured data.

To address the above-mentioned issues, we propose a sequential GCN for Active Learning in a task-agnostic manner. Figure 1 shows the pipeline of the proposed method. In the Figure, Phase I implements the learner. This is a model trained to minimize the objective of the downstream task. Phase II, III and IV compose our sampler where we deploy the GCN and apply the sampling techniques on graph-induced node embeddings and their confidence scores. Finally, in Phase V, the selected unlabelled examples are sent for annotation. At the initial stage, the learner is trained with a small number of seed labelled examples. We extract the features of both labelled and unlabelled images from the learner parameters. During Phase II, we construct a graph where features are used to initialise the nodes of the graph and similarities represent the edges. Unlike VAAL[33], the initialisation of the nodes by the features extracted from the learner creates an opportunity to inherit uncertainties to the sampler. This graph is passed through GCN layers (Phase III) and the parameters of the graph are learned to identify the nodes of labelled vs unlabelled example. This objective of the sampler is independent of the one from the learner. We convolve on the graph which does message-passing operations between the nodes to induce the higher-order representations. The graph embedding of any image depends primarily upon the initial representation and the associated neighbourhood nodes. Thus, the images bearing similar semantic and neighbourhood structure end up inducing close representations which will play a key role in identifying the sufficiently different unlabelled examples from the labelled ones. The nodes after convolutions are classified as labelled or unlabelled. We sort the examples based on the

confidence score, apply an uncertainty sampling approach (Phase IV), and send the selected examples to query their labels (Phase V). We called this sampling method as **UncertainGCN**. Figure 2 simulates the UncertainGCN sampling technique. Furthermore, we adapt the higher-order graph node information under the CoreSet [31] for a new sampling technique by introducing latent space distancing. In principle, CoreSet [31] uses risk minimisation between core-sets on the learner feature space while we employ this operation over GCN features. We called this sampling technique as **CoreGCN**. Our method has a clear advantage due to the aforementioned strengths of the GCN which is demonstrated by both the quantitative and qualitative experiments (see Section 4). Traversing from Phase I to Phase V as shown in Figure 1 completes a cycle. In the next iteration, we flip the label of annotated examples from unlabelled to labelled and re-train the whole framework.

We evaluated our sampler on four challenging real domain image classification benchmarks, one depth-based dataset for 3D HPE and a synthetic image classification benchmark. We compared with several competitive sampling baselines and existing state-of-the-arts methods including CoreSet, VAAL and Learning Loss. From both the quantitative and the qualitative comparisons, our proposed framework is more accurate than existing methods.

2. Related Works

Model-based methods. Recently, a new category of methods is explored in the active learning paradigm where a separate model from the learner is trained for selecting a subset of the most representative data. Our method is based on this category. One of the first approaches [42] attached a loss-learning module so that loss can be predicted offline for the unlabelled samples. In [33], another task-agnostic solution deploys a variational auto-encoder (VAE) to map the available data on a latent space. Thus, a discriminator is trained in an adversarial manner to classify labelled from unlabelled. The advantage of our method over this approach is the exploitation of the relative relationship between the examples by sharing information through message-passing operations in GCN.

GCNs in active learning. GCNs [18] have opened new active learning methods that have been successfully applied in [1, 38, 6, 11]. In comparison to these methods, our approach has distinguished learner and sampler. It makes our approach task-agnostic and also gets benefited from model-based methods mentioned just before. Moreover, none of these methods is trained in a sequential manner. [38] proposes K-Medoids clustering for the feature propagation between the labelled and unlabelled nodes. A regional uncertainty algorithm is presented in [1] by extending the PageRank [27] algorithm to the active learning problem. Similarly, [6] combines node uncertainty with graph centrality

for selecting the new samples. A more complex algorithm is introduced in [11] where a reinforcement scheme with multi-armed bandits decides between the three query measurements from [6]. However, these works [6, 11, 38] derive their selection mechanism on the assumption of a Graph learner. This does not make them directly comparable with our proposal where a GCN is trained separately for a different objective function than the learner.

Uncertainty-based methods. Earlier techniques for sampling unlabelled data have been explored through uncertainty exploration of the convolutional neural networks (CNNs). A Bayesian approximation introduced in [9] produce meaningful uncertainty measurements by variational inference of a Monte Carlo Dropout (MC Dropout) adapted architecture. Hence, it is successfully integrated into active learning by [10, 16, 19, 28]. With the rise of GPU computation power, [2] ensemble-based method outperformed MC Dropout.

Geometric-based methods. Although there have been studies exploring the data space through the representations of the learning model ([20, 35, 14]), the first work applying it for CNNs as an active learning problem, CoreSet, has been presented in [31]. The key principle depends on minimising the difference between the loss of a labelled set and a small unlabelled subset through a geometric-defined bound. We furthermore represent this baseline in our experiments as it successfully over-passed the uncertainty-based ones.

3. Method

In this section, we describe the proposed method in details. First, we present the learners in brief for the image classification and regression tasks under the pool-based active learning scenario. In the second part, we discuss our two novel sampling methods: **UncertainGCN** and **CoreGCN**. UncertainGCN is based on the standard AL method uncertainty sampling [31] which tracks the confidence scores of the designed graph nodes. Furthermore, CoreGCN adapts the highly successful CoreSet [31] on the induced graph embeddings by the sequentially trained GCN network.

3.1. Learner

In Figure 1, Phase I depicts the learner. Its goal is to minimise the objective of the downstream task. We have considered both the classification and regression tasks. Thus, the objective of this learner varies with the nature of the task we are dealing with.

Classification: For the classification tasks, the learner is a CNN image classifier. We deploy a deep model \mathcal{M} that maps a set of inputs $\mathbf{x} \in \mathbf{X}$ to a discriminatory space of outputs $\mathbf{y} \in \mathbf{Y}$ with parameters θ . We took ResNet-18 [15] as the CNN model due to its relatively higher performance

in comparison to other networks with comparable parameter complexity. Any other model like VGG-11[32] can also be easily deployed (refer to Supplementary Material B.4). A loss function $\mathcal{L}(\mathbf{x}, \mathbf{y}; \theta)$ is minimized during the training process. The objective function of our classifier is cross-entropy defined as below:

$$\mathcal{L}_{\mathcal{M}}^c(\mathbf{x}, \mathbf{y}; \theta) = -\frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{y}_i \log(f(\mathbf{x}_i, \mathbf{y}_i; \theta)), \quad (1)$$

where N_l is the number of labelled training examples and $f(\mathbf{x}_i, \mathbf{y}_i; \theta)$ is the posterior probability of the model \mathcal{M} .

Regression: To tackle the 3D HPE, we deploy a well-known *DeepPrior* [26] architecture as model \mathcal{M} . Unlike the previous case, we regress the 3D hand joint coordinates from the hand depth images. Thus, the objective function of the model changes as in Equation 2. In the Equation, J is the number of joints to construct the hand pose.

$$\mathcal{L}_{\mathcal{M}}^r(\mathbf{x}, \mathbf{y}; \theta) = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\frac{1}{J} \sum_{j=1}^J \|\mathbf{y}_{i,j} - f(\mathbf{x}_{i,j}, \mathbf{y}_{i,j}; \theta)\|^2 \right), \quad (2)$$

To adapt our method to any other type of task, we just need to modify the learner. The rest of our pipeline remains the same which we discuss in more details in the following Sections.

3.2. Sampler

Moving to the second Phase from Figure 1, we adopt a pool-based scenario for active learning. This has become a standard in deep learning system due to its successful deployment in recent methods [2, 31, 33, 42]. In this scenario, from a pool of unlabeled dataset \mathbf{D}_U , we randomly select an initial batch for labelling $\mathbf{D}_0 \subset \mathbf{D}_U$. Without loss of generality, in active learning research, the major goal is to optimize a sampler’s method for data acquisition, \mathcal{A} in order to achieve minimum loss with the least number of batches \mathbf{D}_n . This scope can be simply defined for n number of active learning stages as following:

$$\min_n \min_{\mathcal{L}_{\mathcal{M}}} \mathcal{A}(\mathcal{L}_{\mathcal{M}}(\mathbf{x}, \mathbf{y}; \theta) | \mathbf{D}_0 \subset \dots \subset \mathbf{D}_n \subset \mathbf{D}_U). \quad (3)$$

We aim to minimise the number of stages so that fewer samples (\mathbf{x}, \mathbf{y}) would require annotation. For the sampling method \mathcal{A} , we bring the heuristic relation between the discriminative understanding of the model and the unlabelled data space. This is quantified by a performance evaluation metric and traced at every querying stage.

3.2.1 Sequential GCN selection process

During sampling as shown in Figure 1 from Phase II to IV, our contribution relies on sequentially training a GCN

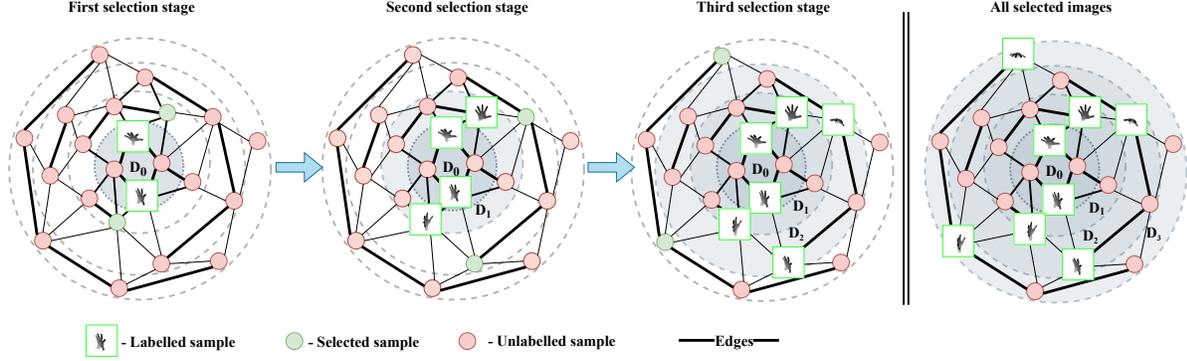


Figure 2: This Figure simulates the sampling behaviour of the proposed **UncertainGCN** sampler at its first three selection stages. We run a toy experiment just taking 100 examples from ICVL [34] Hand Pose dataset. Each node is initialised by the features extracted from the learner and edges capture their relation. Each concentric circle represents a cluster of strongly connected nodes. Here, in our case, a group of images having similar viewpoints are in a concentric circle. Considering two labelled examples as seed labelled examples in the centre-most circle, in the first selection stage, our algorithm selects samples from another concentric circle which is out-of-distribution than selecting the remaining examples from the innermost circle. Similarly, in the second stage, our sampler chooses images residing in another outer concentric circle which are sufficiently different from those selected before.

initialised with the features extracted from the learner for both labelled and unlabelled images at every active learning stage. As stated before, similar to VAAL [33], we consider this methodology as model-based where a separate architecture is required for sampling. Our motivation in introducing the graph is primarily in propagating the inherited uncertainty on the learner feature space between the samples (nodes). Thus, message-passing between the nodes induces higher-order representations of the nodes after applying convolution on the graph. Finally, our GCN will act as a binary classifier deciding which images are annotated.

Graph Convolutional Network. The key components of a graph, \mathcal{G} are the nodes, also called vertices \mathcal{V} and the edges \mathcal{E} . The edges capture the relationship between the nodes and encoded in an adjacency matrix A . The nodes $\mathbf{v} \in \mathbb{R}^{(m \times N)}$ of the graph encode image-specific information and are initialised with the features extracted from the learner. Here, N represents the total number of both labelled and unlabelled examples while m represents the dimension of features for each node. After we apply l_2 normalisation to the features, the initial elements of A result as vector product between every sample of \mathbf{v} i.e. $(S_{ij} = \mathbf{v}_i^\top \mathbf{v}_j, \{i, j\} \in N)$. This propagates the similarities between nodes while falling under the same metric space as the learner’s objective. Furthermore, we subtract from S the identity matrix I and then we normalise by multiplying with its degree D . Finally, we add the self-connections back so that the closest correlation is with the node itself. This can simply be summarised under:

$$A = D^{-1}(S - I) + I. \quad (4)$$

To avoid over-smoothing of the features in GCN [18], we adopt a two-layer architecture. The first GCN layer can be described as a function $f_{\mathcal{G}}^1(A, \mathcal{V}; \Theta_1) : \mathbb{R}^{N \times N} \times \mathbb{R}^{m \times N} \rightarrow \mathbb{R}^{h \times N}$ where h is number of hidden units and Θ_1 are its parameters. A rectified linear unit activation [24] is applied after the first layer to maximise feature contribution. However, to map the nodes as labelled or unlabelled, the final layer is activated through a sigmoid function. Thus, the output of $f_{\mathcal{G}}$ is a vector length of N with values between 0 and 1 (where 0 is considered unlabelled and 1 is for labelled). We can further define the entire network function as:

$$f_{\mathcal{G}} = \sigma(\Theta_2(\text{ReLU}(\Theta_1 A)A)). \quad (5)$$

In order to satisfy this objective, our loss function will be defined as:

$$\mathcal{L}_{\mathcal{G}}(\mathcal{V}, A; \Theta_1, \Theta_2) = -\frac{1}{N_l} \sum_{i=1}^{N_l} \log(f_{\mathcal{G}}(\mathcal{V}, A; \Theta_1, \Theta_2)_i) - \frac{\lambda}{N - N_l} \sum_{i=N_l+1}^N \log(1 - f_{\mathcal{G}}(\mathcal{V}, A; \Theta_1, \Theta_2)_i), \quad (6)$$

where λ acts as a weighting factor between the labelled and unlabelled cross-entropy.

UncertainGCN: Uncertainty sampling on GCN. Once the training of the GCN is complete, we move forward to selection. From the remaining unlabelled samples \mathbf{D}_U , we can draw their confidence scores $f_{\mathcal{G}}(\mathbf{v}_i; \mathbf{D}_U)$ as outputs of the GCN. Similarly to uncertainty sampling, we propose to

select with our method, **UncertainGCN**, the unlabelled images with the confidence depending on a variable s_{margin} . While querying a fixed number of b points for a new subset \mathbf{D}_L , we apply the following equation:

$$\mathbf{D}_L = \mathbf{D}_L \cup \arg \max_{i=1 \dots b} |s_{margin} - f_G(\mathbf{v}_i; \mathbf{D}_U)|. \quad (7)$$

For selecting the most uncertain unlabelled samples, s_{margin} should be closer to 0. In this manner, the selected images are challenging to discriminate, similarly to the adversarial training scenario [12]. This stage is repeated as long as equation 3 is satisfied. Algorithm 1 summarises the GCN sequential training with the UncertainGCN sampling method.

Algorithm 1 UncertainGCN active learning algorithm

- 1: **Given:** Initial labelled set \mathbf{D}_0 , unlabelled set \mathbf{D}_U and query budget b
 - 2: **Initialise** $(\mathbf{x}_L, \mathbf{y}_L), (\mathbf{x}_U)$ - labelled and unlabelled images
 - 3: **repeat**
 - 4: $\theta \leftarrow f(\mathbf{x}_L, \mathbf{y}_L)$ ▷ Train learner with labelled
 - 5: $\mathcal{V} = [\mathbf{v}_L, \mathbf{v}_U] \leftarrow f(\mathbf{x}_L \cup \mathbf{x}_U; \theta)$ ▷ Extract features for labelled and unlabelled
 - 6: *Compute adjacency matrix A according to Equation 4*
 - 7: $\Theta \leftarrow f_G(\mathcal{V}, A)$ ▷ Train the GCN
 - 8: **for** $i = 1 \rightarrow b$ **do**
 - 9: $\mathbf{D}_L = \mathbf{D}_L \cup \arg \max_i |s_{margin} - f_G(\mathbf{v}; \mathbf{D}_U)|$ ▷ Add nodes depending on the label confidence
 - 10: **end for**
 - 11: *Label \mathbf{y}_U given new \mathbf{D}_L*
 - 12: **until** Equation 3 is satisfied
-

CoreGCN: CoreSet sampling on GCN. To integrate geometric information between the labelled and unlabelled graph representation, we approach a CoreSet technique [31] in our sampling stage. This has shown better performance in comparison to uncertainty-based methods [38]. [31] shows how bounding the difference between the loss of the unlabelled samples and the one of the labelled is similar to the k -Centre minimisation problem stated in [37].

In this approach, the sampling is based on the l_2 distances between the features extracted from the trained classifier. Instead of that, we will make use of our GCN architecture by applying CoreSet method on the features represented after the first layer of the graph. To this, the CoreSet method benefits from the cyclical dependencies. The sampling method is adapted to our mechanism for each b data point under the equation:

$$\mathbf{D}_L = \mathbf{D}_L \cup \arg \max_{i \in \mathbf{D}_U} \min_{j \in \mathbf{D}_L} \delta(f_G^1(A, \mathbf{v}_i; \Theta_1), f_G^1(A, \mathbf{v}_j; \Theta_1)), \quad (8)$$

where δ is the Euclidean distance between the graph features of the labelled node \mathbf{v}_i and the ones from the unlabelled node \mathbf{v}_j . We define this method as **CoreGCN**.

Finally, given the model-based mechanism, we claim that our sampler is task-agnostic as long as the learner is producing a form of feature representations. In the following section, we will experimentally demonstrate the performance of our methods quantitatively and qualitatively.

4. Experiments

We performed experiments on sub-sampling RGB and grayscale real images for classification, depth real images for regression and RGB synthetic-generated for classification tasks. We describe them in details below.

4.1. Classification

Datasets and Experimental Settings. We evaluated the proposed AL methods on four challenging image classification benchmarks. These include three RGB image datasets, CIFAR-10[21], CIFAR-100[21] and SVHN[13], and a grayscale dataset, FashionMNIST[39]. Initially, for every benchmark, we consider the entire training set as an unlabelled pool (\mathbf{D}_U). As a cold-start, we randomly sample a small subset and query their labels, \mathbf{D}_L . For CIFAR-10, SVHN and FashionMNIST, the size of the seed labelled examples is 1,000. Whereas, for CIFAR-100 we select 2,000 due to their comparatively more number of classes (100 vs 10). We conduct our experiments for 10 cycles. At every stage, the budget is fixed at 1,000 images for the 10-class benchmarks and at 2,000 for CIFAR-100 which is a 100-class benchmark. Similar to the existing works of [2, 42], we apply our selection on randomly selected subsets $\mathbf{D}_S \subset \mathbf{D}_U$ of unlabelled images. This avoids the redundant occurrences which are common in all datasets [4]. The size of \mathbf{D}_S is set to 10,000 for all the experiments.

Implementation details. ResNet-18 [15] is the favourite choice as learner due to its relatively higher accuracy and better training stability. During training the learner, we set a batch size of 128. We use Stochastic Gradient Descent (SGD) with a weight decay 5×10^{-4} and a momentum of 0.9. At every selection stage, we train the model for 200 epochs. We set the initial learning rate of 0.1 and decrease it by the factor of 10 after 160 epochs. We use the same set of hyper-parameters in all the experiments. For the sampler, GCN has 2 layers and we set the dropout rate to 0.3 to avoid over-smoothing [43]. The dimension of initial representations of a node is 1024 and it is projected to 512. The objective function is binary cross-entropy per node. We set the value of $\lambda = 1.2$ to give more importance to the larger unlabelled subset. We choose Adam [17] optimizer with a weight decay of 5×10^{-4} and a learning rate of 10^{-3} . We initialise the nodes of the graph with the features of the images extracted from the learner. We set the value of s_{margin}

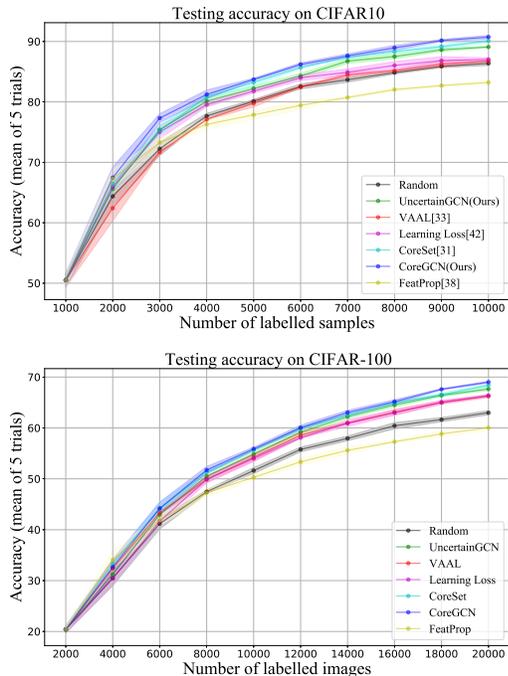


Figure 3: Quantitative comparison on CIFAR-10(top) and CIFAR-100(bottom) (Zoom in the view)

to 0.1. For the empirical comparisons, we suggest readers to refer Supplementary Material.

Compared Methods and Evaluation Metric: We compare our method with a wide range of baselines which we describe here. Random sampling is by default the most common sampling technique. CoreSet[31] on learner feature space is one of the best performing geometric techniques to date and it is another competitive baseline for us. VAAL [33] and Learning Loss [42] are two state-of-the-art baselines from task-agnostic frameworks. Finally, we also compare with FeatProp [38] which is a representative baseline for the GCN-based frameworks. This method is designed for cases where a static fixed graph is available. To approximate their performance, we construct a graph from the features extracted from learner and similarities between the features as edges. We then compute the k-Medoids distance on this graph. For quantitative evaluation, we report the mean average accuracy of 5 trials on the test sets.

Quantitative Comparisons. We train the ResNet-18 learner with all the available training examples on every dataset separately and report the performance on the test set. Our implementation obtains 93.09% on CIFAR-10, 73.02% on CIFAR-100, 93.74% on FashionMNIST, and 95.35% on SVHN. This is comparable as reported on the official implementation [15]. These results are also set as the upper-bound performance of the active learning frameworks.

Figure 3 (left) shows the performance comparison of

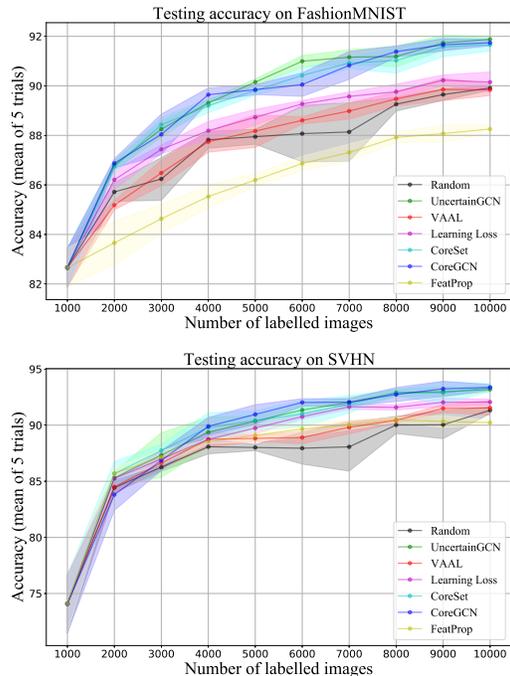


Figure 4: Quantitative comparison on FashionMNIST(top) and SVHN(bottom) (Zoom in the view)

UncertainGCN and CoreGCN with the other five existing methods on **CIFAR-10**. The solid line of the representation is the mean accuracy and the faded colour shows the standard deviation. Both our sampling techniques surpass almost every other compared methods in every selection stage. CoreSet is the closest competitor for our methods. After selecting 10,000 labelled examples, the CoreGCN achieves 90.7% which is the highest performance amongst reported in the literature [42, 33]. Likewise, Figure 3 (right) shows the accuracy comparison on **CIFAR-100**. We observe almost similar trends as on CIFAR-10. With only 40% of the training data, we achieve 69% accuracy by applying CoreGCN. This performance is just 4% lesser than when training with the entire dataset. Compared to CIFAR-10, we observe the better performance on VAAL in this benchmark. The reason is that VAE might favour a larger query batch size ($>1,000$). This exhaustively annotates large batches of data when the purpose of active learning is to find a good balance between exploration and exploitation as we constrain the budget and batches sizes.

We further continue our evaluation on the image classification by applying our methods on FashionMNIST and SVHN. In Figure 4, the left and the right graphs show the comparisons on **FashionMNIST** and **SVHN** respectively. As in the previous cases, our methods achieve at minimum similar performance to that of existing methods or outperforming them. From the studies on these datasets, we

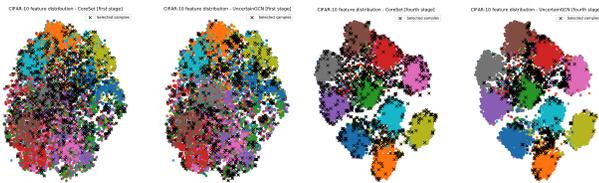


Figure 5: Exploration comparison on CIFAR-10 between CoreSet and UncertainGCN (Zoom in the view)

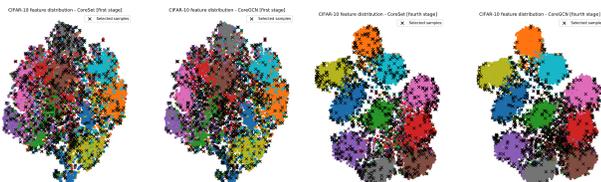


Figure 6: Exploration comparison on CIFAR-10 between CoreSet and CoreGCN (Zoom in the view)

observed consistent modest performance of FeatProp [38]. This may be because it could not generalise on the unstructured data like ours.

Qualitative Comparisons. To further analyse the sampling behaviour of our method we perform qualitative comparison with existing method. We choose CoreSet for its consistently better performance in empirical evaluations when compared to the other baselines. We made this comparison on CIFAR-10. For the two algorithms, we generate the t-SNE [36] plots of both labelled and unlabelled extracted features from the learner at the first and fourth selection stage. To make a distinctive comparison of sampling behaviour from early stage, we choose to keep a difference of 3 stages. Figure 5, t-SNE plots, compares the sampling behaviour of CoreSet and UncertainGCN. In the first sampling stage, the selected samples distribution is uniform which is similar for both techniques. Without loss of generality, the learner trained with a small number of seed annotated examples is sub-optimal, and, hence the features of both labelled and unlabelled are not discriminative enough. This makes the sampling behaviour for both methods near random. At the fourth selection stage, the learner becomes relatively more discriminative. This we can notice from the clusters representing each class of CIFAR-10. Now, these features are robust to capture the relationship between the labelled and unlabelled examples which we encode in the adjacency matrix. Message-passing operations on GCN exploits the correlation between the labelled and unlabelled examples by inducing similar representations. This enables our method to target on the out-of-distribution unlabelled samples and areas where features are hardly distinguished. This characteristics we can observe on the plot of the fourth selection stage of UncertainGCN. Similarly, in Figure 6, we

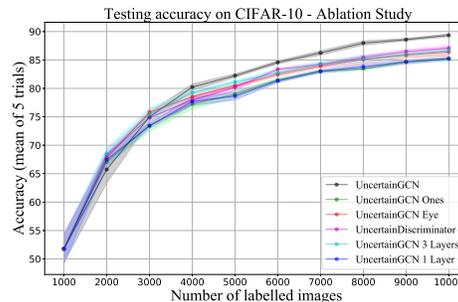


Figure 7: Ablation studies (Please zoom in)

continue the qualitative investigation for the CoreGCN acquisition method. CoreGCN avoids over-populated areas while tracking out-of-distribution unlabelled data. Compared to UncertainGCN, the geometric information from CoreGCN maintains a sparsity throughout all the selection stages. Consequently, it preserves the message passing through the uncertain areas while CoreSet keeps sampling closer to cluster centres. This brings a stronger balance in comparison to CoreSet between in and out-of-distribution selection with the availability of more samples.

Ablation Studies To further motivate the GCN proposal, we conduct ablation studies on the sampler architecture. In Figure 7, still on CIFAR-10, we replace the GCN with a 2 Dense layer discriminator, *UncertainDiscriminator*. This approach over-fits at early selection stages. Although, GCN with 2 layers [18] has been a de-facto optimal design choice, we also report the performance with 1 layer (hinders long-range propagation) and 3 (over-smooths). However, to further quantify the significance of our adjacency matrix with feature correlations, we evaluate GCN samplers with identity (UncertainGCN Eye) and filled with 1s matrices (UncertainGCN Ones). Finally, a study on two important hyper-parameters: drop-out (0.3, 0.5, 0.8) and the number of hidden units (128, 256, 512) is in the Supplementary B.2. We also fine-tune these parameters to obtain the optimal solution.

4.2. Regression

Dataset and Experimental Settings: We further applied our method on a challenging dataset for 3D Hand Pose Estimation benchmarks from depth images. ICVL [34] contains 16,004 hand depth-images in the training set and the test set has 1,600. At every selection stage, similar to the experimental setup of image classification, we randomly pre-sample 10% of entire training examples D_S and apply the AL methods on this subset of the data. Out of this pre-sampled images subset, we apply our sampler to select the most influencing 100 examples.

Implementation Details: 3D HPE is a regression problem which involves estimating the 3D coordinates of the hand

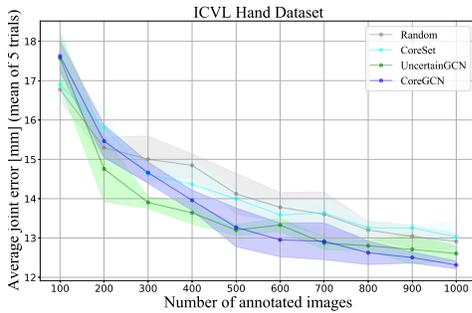


Figure 8: Quantitative comparison on 3D Hand Pose Estimation (lower is better)

joints from depth images. Thus, we replace ResNet-18 by commonly used *DeepPrior* [26] as learner. The sampler and the other components in our pipeline remain the same as in the image classification task. This is yet another evidence for our sampling method being task-agnostic. For all the experiments, we train the 3D HPE with Adam[17] optimizer and with a learning rate of 10^{-3} . The batch size is 128. As pre-processing, we apply a pre-trained U-Net [30] model to detect hands, centre, crop and resize images to the dimension of 128x128.

Compared Methods and Evaluation Metric: We compare our methods from the two ends of the spectrum of baselines. One is random sampling which is the default mechanism. The other is CoreSet[31], one of the best performing baselines from the previous experiments. We report the performance in terms of mean squared error averaged from 5 different trials and its standard deviation.

Quantitative Evaluations: Figure 8 shows the performance comparison on ICVL dataset. In the Figure, we can observe that both our sampling methods, CoreGCN and UncertainGCN, outperform the CoreSet and Random sampling consistently from the second selection stage. The slope of decrease in error for our methods sharply falls down from the second till the fifth selection stage for UncertainGCN and till the sixth for CoreGCN. This gives us an advantage over the other methods when we have a very limited budget. At the end of the selection stage, CoreGCN gives the least error of 12.3 mm. In terms of performance, next to it is UncertainGCN.

4.3. Sub-sampling of Synthetic Data.

Unlike previous experiments of sub-sampling real images, we applied our method to select synthetic examples obtained from StarGAN [7] trained on RaFD [23] for translation of face expressions. Although Generative Adversarial Networks [12] are closing the gap between real and synthetic data [29], still the synthetic images and its associated labels are not yet suitable to train a downstream discriminative model. Recent study [3] recommends sub-sampling the

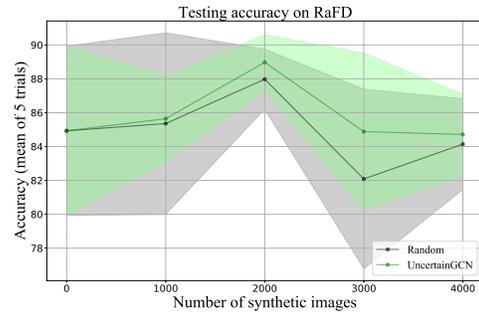


Figure 9: Performance comparison on sub-sampling synthetic data to augment real data for expression classification

synthetic data before augmenting to the real data. Hence, we apply our algorithm to get a sub-set of the quality and influential synthetic examples. The experimental setup is similar to that of the image classification which we described in our previous Section. Learner and Sampler remain the same. The only difference will be in the nature of the pool images. Instead of real data, we have StarGAN synthetic images. Figure 9 shows the performance comparison of random selection vs our UncertainGCN method in 5 trials. From the experiment, we can observe our method achieving higher accuracy with less variance than commonly used random sampling. The mean accuracy drops for both the methods from the fourth selection stage. Only a small fraction of synthetic examples are useful to train the model [3]. After the fourth stage, we force sampler to select more examples which may end up choosing noisy data.

5. Conclusions

We have presented a novel methodology of active learning in image classification and regression using Graph Convolutional Network. After systematical and comprehensive experiments, our adapted sampling techniques, UncertainGCN and CoreGCN, produced state-of-the-art results on 6 benchmarks. We have shown through qualitative distributions that our selection functions maximises informativeness within the data space. The design of our sampling mechanism permits integration into other learning tasks. Furthermore, this approach enables further investigation in this direction where optimised selection criteria can be combined GCN sampler.

Acknowledgements

This work is partially supported by Huawei Technologies Co. and by EPSRC Programme Grant FACER2VM (EP/N007743/1).

References

- [1] Roy Abel and Yoram Louzoun. Regional based query in graph active learning, 2019. 1906.08541v1.
- [2] William H Beluch Bcai, Andreas Nürnberger, and Jan M Khler Bcai. The power of ensembles for active learning in image classification. In *CVPR*, 2018.
- [3] Binod Bhattarai, Seungryl Baek, Rumeysa Bodur, and Tae-Kyun Kim. Sampling strategies for gan synthetic data. In *ICASSP*, 2020.
- [4] Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio. Semantic redundancies in image-classification datasets: The 10% you don't need. *CoRR*, 2019.
- [5] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 2017.
- [6] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. Active Learning for Graph Embedding, 2017. 1705.05085v1.
- [7] Yunje Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.
- [8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 1995.
- [9] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *ICML*, 2016.
- [10] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. In *ICML*, 2017.
- [11] Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. Active discriminative network representation learning. In *IJCAI*, 2018.
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.
- [13] Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks, 2013. 1312.6082v4.
- [14] Sarel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In *SCG*, 2005.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [16] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Mt Lengyel. Bayesian Active Learning for Classification and Preference Learning, 2011. 1112.5745v1.
- [17] Diederik P Kingma and Jimmy Lei Ba. ADAM: A Method for Stochastic Optimization. In *ICLR*, 2015.
- [18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [19] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In *NeurIPS*, 2019.
- [20] Aryeh Kontorovich, Sivan Sabato, and Ruth Uerner. Active nearest-neighbor learning in metric spaces. In *NeurIPS*, 2016.
- [21] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.
- [22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.
- [23] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 2010.
- [24] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.
- [25] Markus Oberweger and Vincent Lepetit. Deeprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV*, 2017.
- [26] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. In *CVWW*, 2015.
- [27] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [28] Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and Jos Miguel Hernandez-Lobato. Bayesian Batch Active Learning as Sparse Subset Approximation. In *NeurIPS*, 2019.
- [29] Suman Ravuri and Oriol Vinyals. Seeing is not necessarily believing: Limitations of biggans for data augmentation. In *ICLR*, 2019.
- [30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, 2015.
- [31] Ozan Sener and Silvio Savarese. Active Learning for Convolutional Neural Networks: A Core-set approach. In *ICLR*, 2018.
- [32] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Network for Large-scale image recognition. In *ICLR*, 2015.
- [33] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational Adversarial Active Learning. In *ICCV*, 2019.
- [34] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *CVPR*, 2014.
- [35] Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: Fast svm training on very large data sets. *JMLR.*, 2005.
- [36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne, 2008. *JMLR*.
- [37] Gert Wolf. Facility location: concepts, models, algorithms and case studies. In *Contributions to Management Science*, 2011.
- [38] Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. Active Learning for Graph Neural Networks via Node Feature Propagation, 2019. 1910.07567v1.
- [39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017. 1708.07747v2.

- [40] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *PMLR*, 2016.
- [41] Qi Ye, Shanxin Yuan, and Tae-kyun Kim. Spatial Attention Deep Net with Partial PSO for Hierarchical Hybrid Hand Pose Estimation. In *ECCV*, 2016.
- [42] Donggeun Yoo and In So Kweon. Learning Loss for Active Learning. In *CVPR*, 2019.
- [43] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling over-smoothing in gnns. *ICLR*, 2020.