Active Learning for Mesh Segmentation: Comparing uncertainty quantification methods for learning on unstructured grids

Sofia Lima
Computational Biology Department
Carnegie Mellon University
slima2@andrew.cmu.edu

Parker Simpson
Computational Biology Department
Carnegie Mellon University
psimpson@andrew.cmu.edu

Aruneshwar Venkat
Computational Biology Department
Carnegie Mellon University
aruneshv@andrew.cmu.edu

Jen Yi Wong
Computational Biology Department
Carnegie Mellon University
jenyiw@andrew.cmu.edu

1 Introduction

The purpose of this project was to assess the efficacy of active learning for the task of segmenting body parts in 3D meshes of human anatomies (Fig. 1). Obtaining segmentation labels for mesh data is an expensive and time-consuming task. A possible way to save computational resources would be to decrease the number of samples needed to train an effective segmentation model. The dataset chosen for this project comes from researchers in Perceiving Systems at the Max Planck Institute for Intelligent Systems [1]. The goal of this project was to compare the results of active learning with different query selection methods to online learning with random query selection. Our query selection methods were uncertainty sampling and query by committee. We use a graph neural network framework called GraphSAGE as the base learner.

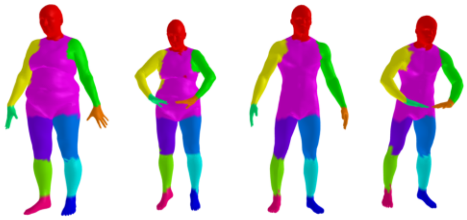

Figure 1: Sample of GNN segmentation results on MPI Faust mesh dataset.

2 Background

In this dataset, the human body is segmented into 12 regions. The task of segmentting body parts is equivalent to node-classification in a graph problem. This problem can be modeled using graph neural networks (GNNs) which are powerful for unstructured data such as social networks and protein-protein interactions. Leveraging knowledge in quantified uncertainty may provide advantages and has been demonstrated in medical image segmentation research [2]. One project describes MC dropout with a sampling based model for mesh data called PointNet++ and how running inference multiple times with active MC dropout allows the algorithm to select uncertain samples at multiple granularity levels such as scene-level and point-level [3, 4]. We aim to follow this active learning framework such that the most uncertain scene-level instance is selected at each round of the active learning algorithm.

Some work has demonstrated quantifying the model uncertainty in GNNs [5, 6]; however, very little work has been conducted exploring the potential advantages of putting the mesh segmentation task in an active learning setting. In this project, we use various uncertainty quantification (UQ) methods, including an ensemble technique [2], and recently developed techniques with monte carlo (MC) dropout during inference [3].

3 Preprocessing

The dataset was found to have a total of 6890 nodes and 41328 edges for each graph, which was prohibitively high given the time and computational constraints. In order to reduce the amount of resources required for each run, we explored two methods of creating a subgraph.

1. Edge Pruning
   A modified Breadth First Search algorithm was used to trim the number of edges in the graph. The algorithm was modified to retain all edges close to the boundaries and trim only the internal nodes within a class.
2. Node Removal
   A fixed set of nodes was removed from every graph and any associated edges were also removed.

The node removal method was found to be more effective and efficient and hence we chose to use the node removal method.

4 Methods

4.1 Active Learning

We aim to analyze the potential advantages of active learning on this mesh segmentation task. Similar to protocols in homework exercises, we will follow the prototypical active learning algorithm as follows:

Algorithm 1 Active Learning Algorithm

```
Obtain initial labeled training data, where $D_L = \{(\mathbf{x_1}, y_1), ..., (\mathbf{x_n}, y_n)\}$, where each $\mathbf{x}_i \in \mathcal{X}$ is an input instance and $y_i \in \mathfrak{C}$ its corresponding label
repeat
    Learn model: $h = Learn(D_L); h \in \mathcal{H}$
    Select the next point to label, $\mathbf{x}_i \in \mathcal{X}$, according to the data access model
    Pay ”oracle” for $\mathbf{x}_i$'s label, $y_i \in \mathfrak{C}$
    Update training data: $D_L = D_L \cup \{(\mathbf{x}_i, y_i)\}$
until stopping condition satisfied
Output final model: $h = Learn(D_L)$
```

We will apply this active learning framework to our mesh segmentation problem and compare the segmentation results based on the accuracy, DICE coefficient and mean intersection over union (meanIoU) across different models and query selection methods.

4.2 Query Selection Mode

We attempted both single query selection and batch query selection on this dataset.

4.2.1 Single Query Selection

For the single query selection methods, we explored Density-based sampling (DBS), Uncertainty sampling (UNS), Monte Carlo Dropout with UNS (MC), Query by Committee (QBC) and passive learning as a control. Each run started with 20 instances and ended when 50 instances were in the observed dataset. 5 iterations were run for each selection method.

4.2.2 Batch Query Selection

For the batch query selection method, we tested two different types of batching that were unique to our dataset: pose-wise batching and patient-wise batching. For pose-wise batching, we query all ten poses from a single patient at each round of active learning. Alternatively, for patient-wise batching, we query all 10 patients in a single pose at each round of active learning. Each of these batching methods induces a unique diversity among the training set. For example, patient-wise batching ensures that the training set contains all possible poses that may appear in the dataset but not all body types, whereas poses-wise batching ensures all possible body types but not all poses. Each run started with 30 instances and ended when 90 instances were in the observed dataset. 10 instances were selected each round. 5 iterations were run for each selection method.

4.3 Query Selection Methods

4.3.1 Uncertainty sampling (UNS)

Uncertainty sampling (UNS) was done using entropy as calculated using the equation below:

$$x^* = \text{argmax}_{x \in \mathbf{U}} - \sum_{y \in C} P(y|x) \text{log}_2 P(y|x)$$

where $\mathbf{U}$ is the set of unlabeled points and C denotes the set of possible classes for y. The point with the highest entropy, x^*, was added to the observed dataset. In theory, this method selects the point of most information from the unobserved dataset, which is also usually the least confident point.

4.3.2 Density-based sampling (DBS)

For density-based sampling (DBS), each point x^*, was chosen based on similarity to the other points as shown by the equation below.

$$x^* = \text{argmin}_{x \in \mathbf{U}} \sum_{x' \in \mathbf{u}} ||x - x'||_2^2$$

where $\mathbf{U}$ is the set of unlabeled points. The L2 norm is used as an indication of similarity. It is expected that a lower total L2 norm will correlate with higher similarity. It is often used with an addition ϕ_A query selection such as entropy, but for this project, we used only similarity without entropy.

4.3.3 MC Dropout (MC)

Traditionally when dropout is implemented in neural networks, a specified percentage of nodes are randomly chosen to have zero values during the training of the network. What differentiates MC dropout from traditional dropout is the continued use of dropout when performing inference with the model. During inference with MC dropout, multiple predictions are made per sample and the final result is an average over the distributions. For our implementation of MC dropout, we chose a dropout probability of 0.3 and a total of 5 predictions per sample. After MC dropout, a patient was queried using entropy-based uncertainty sampling.

4.3.4 Query by Committee (QBC)

Deep ensemble methods have been shown to improve node classification accuracy on GNNs [7]. Compared to the MC dropout method, an ensemble method is expected to be more robust but also more computationally expensive [2]. We chose to use a committee size of three for this project. To quantify the disagreement among our committee members, we computed the KL divergence over the prediction probability distributions from each committee member. This gave us an uncertainty score for each sample that we then used for query selection. To determine the samples in our queried batch, we found the sample with the highest uncertainty score and returned all other poses for the patient to which the most uncertain sample belonged or all other patients in the pose that the most uncertain sample is in depending on our batching preferences of the simulations.

4.4 Model

4.4.1 GraphSAGE

GraphSAGE is a frequently used GNN architecture capable of inductive node embedding[8]. Several advantages of using this model includes its ability to generalize to unseen nodes and its applicability to both feature-rich nodes and nodes with few or no features. The model architecture used consisted of 2 each node has a a feature matrix of size 3, containing the normalized position of the node in 3D space.

5 Results

5.1 About the dataset

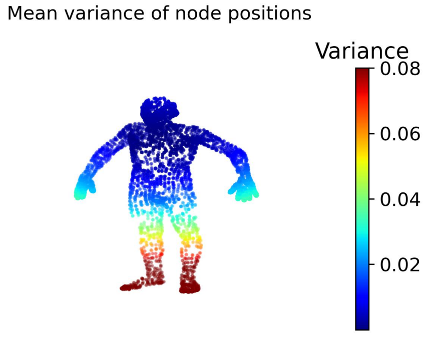

Figure 2: Variance of the position of each node across all 100 poses.

Fig.2 shows positional variance of each node in the graph. As expected, nodes denoted the lower legs and feet area show the highest variance. This was observed even between two individuals in the same pose, highlighting the difficulty in classifying the extremities.

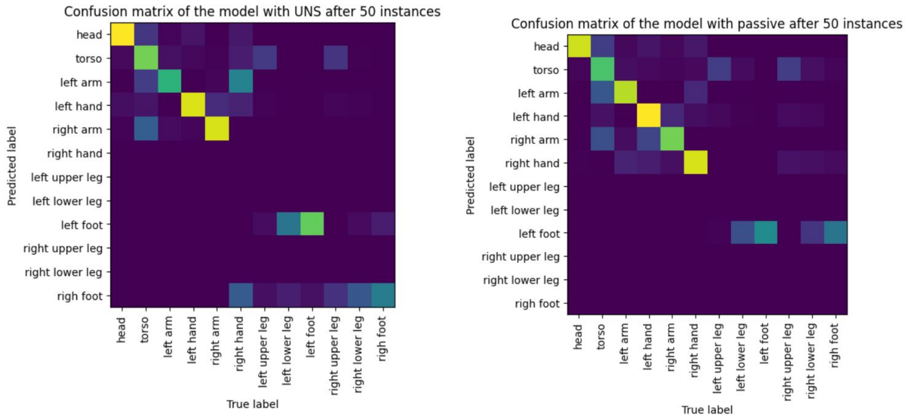

Figure 3: Representative confusion matrix from the model with UNS and passive after 50 instances.

As expected, our models perform more poorly (Fig. 3) on limbs which are in different positions for each pose. The classification may be more accurate for the feet compared to the upper and lower legs as the feet have a denser number of nodes compared to the other 2 regions.

5.2 Single query selection

Fig. 4 shows the accuracy, DICE scores, IOU and loss for each of the 4 methods on the unobserved dataset for single query selection. The figures show the mean and standard deviation of 5 randomly

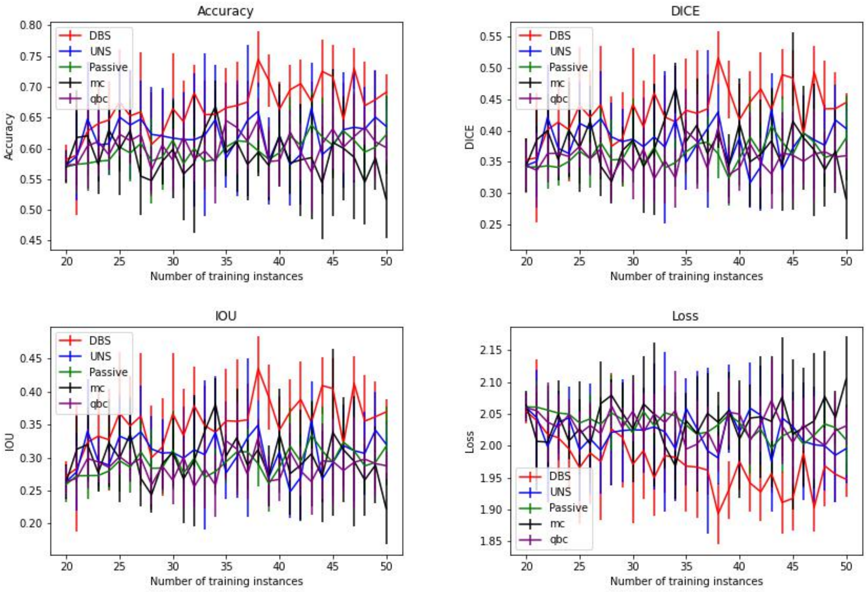

Figure 4: Accuracy, DICE, IOU and Loss for single query selection using density-based sampling (DBS), uncertainty sampling (UNS) and passive learning of the unobserved dataset.

seeded iterations. DBS performed the best, although it was not significantly better than passive, UNS and MC. This is expected as the query selection method chooses the instance with the most similarity to other instances in the unobserved dataset, thus allowing the GNN to generalize to more points in the unobserved dataset.

Surprisingly, as seen in Fig. 4, the UNS and MC methods did not perform well. This could be due to the nature of deep learning methods in general. As these methods have a tendency of overfitting, introducing a point of highest uncertainty does not help it to generalize to the unobserved dataset, unless it is extremely similar to other unobserved points. In addition to the uncertainty of the unobserved dataset, the deep learning model also introduces an additional uncertainty during random weight initialization and during the training process, as the model might be susceptible to local minimums. Thus for an instance of a model trained on 20-50 datapoints, it is uncertain how robust its prediction is on the unobserved dataset. While this additional uncertainty was supposed to be mitigated by the use of dropouts and multiple committee members, as see in Fig. 4, the QBC and MC methods only perform as well as UNS and passive.

5.2.1 Batch selection

Fig. 5 shows the evaluation metrics for the batch-wise selection of instances when grouped by patients and the evaluation metrics for the batch-wise selection by pose can be seen in Fig. 6 for the different query selection methods. Due to the fact that the batch-wise selections used MC and QBC, which were using entropy and KL divergence as a measure of disagreement, the batch-wise MC and QBC did not manage to perform better than passive learning. This may also be confounded by additional inter-pose variance and inter-patient variance for the patient-wise and pose-wise batch selection respectively.

6 Conclusion

There are several areas of future work that we were unfortunately not able to incorporate into this project. For modifications to this model, a preliminary area of focus would be to run the model with all 6890 nodes present for each sample. While this is more computationally expensive, it would be worthwhile to check if halving the number nodes during preprocessing was affecting the results generated by the model. In addition, it could be possible to examine different methods of batch selection, including diversity-based selection. Another possible extension of this project would be

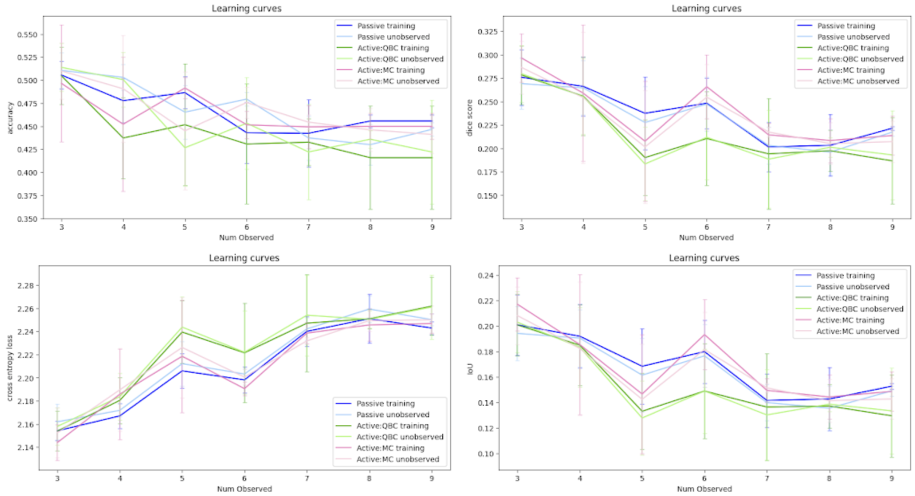

Figure 5: Accuracy, DICE, IOU and Loss for patient-wise batch selection using MC and QBC and passive learning of the unobserved dataset.

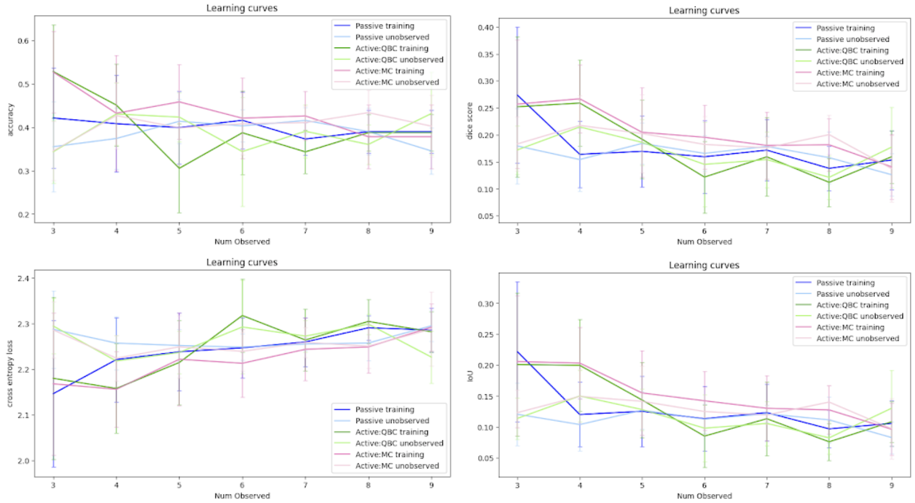

Figure 6: Accuracy, DICE, IOU and Loss for pose-wise batch selection using MC and QBC and passive learning of the unobserved dataset.

to incorporate active learning with MeshCNN, a convolutional neural network that has been used to segment mesh data. We could verify using passive learning the performance differences between the two models before incorporating active learning with MeshCNN and comparing its performance to GraphSAGE.

References

[1] Federica Bogo et al. "FAUST: Dataset and evaluation for 3D mesh registration". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE, June 2014.

[2] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297.

[3] Erik Harutyunyan. *Active Learning for 3D mesh semantic segmentation*. Last accessed 9 March 2023. 2022. URL: `https://yerevan2022.pydata.org/cfp/talk/DPVYSA/`.

[4] Charles Ruizhongtai Qi et al. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in neural information processing systems* 30 (2017).

[5] Pál András Papp et al. “DropGNN: Random Dropouts Increase the Expressiveness of Graph Neural Networks”. In: *35th Conference on Neural Information Processing Systems (NeurIPS 2021)* (2021).

[6] Sai Munikoti et al. “A general framework for quantifying aleatoric and epistemic uncertainty in graph neural networks”. In: *Neurocomputing* 521 (2023), pp. 1–10.

[7] Steven J. Krieg et al. “Deep ENSEMBLES FOR GRAPHS WITH HIGHER-ORDER DEPENDENCIES”. In: *International Conference on Learning Representations 2023* (2023).

[8] William L Hamilton, Rex Ying, and Jure Leskovec. “Inductive Representation Learning on Large Graphs”. In: *31st Conference on Neural Information Processing Systems (NIPS 2017)* (2017).