Efficient Subsampling for GNN Downstream Tasks

Hirad Daneshvar HIRAD.DANESHVAR@TORONTOMU.CA
Toronto Metropolitan University, Toronto, ON, Canada

Reza Samavi SAMAVI@TORONTOMU.CA
Toronto Metropolitan University, Toronto, ON, Canada
Vector Institute, Toronto, ON, Canada

Editors: Hung-yi Lee and Tongliang Liu

Abstract

While Graph Neural Networks (GNNs) have shown significant promise for data integration using graph structures, methods to support subsampling graph data are lagging. To address this gap, in this paper, we propose a novel importance-based data subsampling framework. This framework strategically identifies inputs from a primary graph dataset based on their impact on the model's learning of downstream tasks, such as graph or node classification. Our measure of impact is the predictive uncertainty of each data point. To ensure the subsample is well-representative of the original sample, we cluster them based on their learned graph representation. Finally, subsampling is performed from these identified clusters. The process favours selecting data points with greater prediction uncertainty, while preserving the diversity of the overall sample. We evaluate our approach using a multi-source, real-world dataset on child and youth mental health, comprising emergency department (ED) admissions and mental health questionnaire data. Our experimental results demonstrate that training a GNN with samples identified by the proposed framework yields a statistically significant improvement (on average, 10.13% improvement across metrics from the baseline approach) in predictive performance compared to training on a randomly selected subset of patients. The code is available at https://github.com/tailabTMU/GSS.

Keywords: Graph Subsampling; Multi-Dataset Data Integration; Uncertainty Quantification.

1. Introduction

Graph Neural Networks (GNNs) have emerged as a powerful class of deep learning models designed to handle data with complex relational structures. While GNNs have demonstrated strong potential for integrating data through graph-based representations, techniques for effectively subsampling graph data remain underdeveloped. An important application area of graph data subsampling is in medical AI, where data is often multi-modal and collected from diverse data stores. The following real-world case demonstrates the challenges of graph data subsampling.

The authors of this article, in collaboration with a clinical team, are designing a clinical decision support system for mental health using GNN where the relational structure arises from the electronic health records (EHRs) of the patients and their responses to mental health questionnaire, where the data collection scale is on the order of thousands. To improve the prediction performance of the network, the study also needs to collect audio data from a subset of patients on the order of a hundred, as the process of audio data

collection is resource-intensive. As a result, finding patients who are well-representative of the original patient data and at the same time will have the most influence on the network's performance is a major challenge this research project is facing.

Several researchers have studied the problem of graph data subsampling with the goal of improving network performance (Xu et al., 2023; Jin et al., 2022; Gupta et al., 2024; Georgiev et al., 2023; Jain et al., 2025). These studies primarily focus on compressing graphs or sampling subgraphs, thereby reducing both dataset size and the size of individual graphs. However, graph compression is generally not applicable to diverse graph integration, where data originates from various data stores, such as EHRs and mental health questionnaires, each with its own distinct graph structure. Unlike graph compression, which aims to reduce the size of a graph, graph integration often seeks to construct larger, unified graphs. A promising approach with potential for graph data subsampling is coreset selection, which is primarily used in active learning. Coreset selection offers an approach to select informative samples from unlabeled data for human labelling, which can subsequently improve network performance (Yoo and Kweon, 2019; Katharopoulos and Fleuret, 2018). An ideal core-set selection approach leverages an influence measure, such as the model's prediction uncertainty, to identify more informative samples while simultaneously encouraging diversity within the subset to ensure a well represented subsample (Yoo and Kweon, 2019).

In this paper, we propose an efficient graph subsampling approach tailored for downstream prediction tasks, such as graph classification. We are inspired by the coreset selection approach in identifying samples that are most influential for training a predictive GNN. We use the network's prediction uncertainty as a metric for identifying samples that are highly informative for network training (Katharopoulos and Fleuret, 2018; Jeong et al., 2023; Chen et al., 2024). However, to improve efficiency, we employ self-distillation in the quantification of uncertainty. Self-distillation allows for the training of a multi-classifier GNN, thereby significantly reducing the computational costs of both training and inference for uncertainty estimation (Daneshvar and Samavi, 2026). Selecting data points solely based on the highest uncertainty can lead to the undesirable outcome of mostly selecting outliers (Katharopoulos and Fleuret, 2018; Jeong et al., 2023; Settles and Craven, 2008; Chen et al., 2024). To counter this problem and promote diversity among selected samples, we utilize K-means clustering based on graph representations. Within each cluster, data points are further grouped into percentiles according to their uncertainty values. Our diversity-enabled grouping strategy ensures the inclusion of samples across the full range of uncertainty, preventing bias towards extreme outliers. Ultimately, samples are selected with a weighted emphasis on those exhibiting higher prediction uncertainty.

The key contributions of this paper are as follows. First, we propose a diversity-enabled subsampling method for GNNs specifically designed for graph classification, an underexplored direction, particularly in the context of heterogeneous graphs. Second, our method leverages self-distillation to train a multi-classifier GNN, followed by uncertainty quantification, substantially reducing the computational cost typically associated with uncertainty-based subsampling. Third, we empirically demonstrate that our framework effectively selects representative samples from a primary dataset to augment with a secondary one, thereby improving training performance in scenarios involving diverse data integration under limited-data conditions.

2. Related Work

We can categorize methods for graph subsampling into two main categories: *coreset selection* and *graph dataset compression*. In this section, we provide a review of methods in each category and conclude by outlining how our approach differs from these methods.

Coreset selection is an approach primarily used in active learning to efficiently select representative samples from an unlabeled dataset for human experts to label (Yoo and Kweon, 2019). Coreset selection is extensively studied for subsampling purposes, as it provides methods to identify and sample data points that significantly improve a network's learning ability, stemming from the understanding that not all samples contribute equally to model performance (Katharopoulos and Fleuret, 2018; Jeong et al., 2023; Xie et al., 2023; Joshi and Mirzasoleiman, 2023; Yoo and Kweon, 2019; Citovsky et al., 2021; Ash et al., 2020; Caramalau et al., 2021). Principles of coreset selection have been widely applied to various deep learning domains, including large language models (Xie et al., 2023) and computer vision (Jeong et al., 2023). Applying coreset selection to graphs remains an underexplored area, particularly in the context of graph subsampling. A closely related work, Ding et al. (2024) proposed a method for neighbourhood subgraph selection around a node using ego-graphs (Ding et al., 2024). However, the method differs from the primary focus of this paper on selecting graphs for dataset-level subsampling.

The primary focus of existing graph subsampling studies, such as KiDD (Xu et al., 2023) and DosCond (Jin et al., 2022), is **graph dataset compression**, which involves using the original dataset to create a synthetic, smaller and more compressed GNN training dataset. Both KiDD and DosCond notably utilize gradient matching for graph compression. Additionally, MIRAGE (Gupta et al., 2024) further leverages computation trees of training graphs for dataset compression. Beyond the mentioned studies, researchers have employed Tree Mover's Distance (TMD) either to select specific training sets for tasks related to neural algorithmic reasoning (Georgiev et al., 2023) or to aim to reduce both the number of graphs and the size of individual graphs for GNN training (Jain et al., 2025). However, the graph compression paradigm, which focuses on generating smaller datasets by reducing graph size, is not well-suited for scenarios involving the integration of multiple datasets, particularly when the datasets contain a limited number of samples. In such settings, the objective of graph-based data integration is to enrich the original graph by incorporating additional information, thereby enabling the GNN to exploit more relevant and complementary data during the learning process. Applying graph compression in this context may lead to the loss of valuable information that, when combined with supplementary data from other datasets, could otherwise enhance the representational capacity and performance of the GNN.

Our goal is to sample graphs that are most influential in training a GNN for a specific downstream task, such as graph classification. Our framework will enable researchers to leverage insights from our subsampling approach to strategically collect more diverse and potentially multi-source data, facilitating its integration into a unified graph structure that can improve the network's performance.

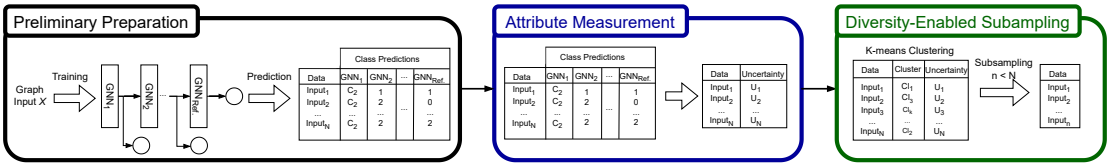

Figure 1: Overview of the proposed subsampling approach.

3. Methodology

3.1. Problem Setup

Our proposed approach is inspired by *coreset selection*, a technique primarily used in active learning to efficiently select representative samples from an unlabeled dataset for human experts to label (Yoo and Kweon, 2019). By adopting similar principles, we aim to propose a subsampling methodology for graphs that selects samples highly relevant to a given downstream task, such as graph classification, thereby making the sample more effective for GNN training.

There are three approaches to coreset selection: (1) uncertainty-based, (2) diversity-based, and (3) approaches based on expected changes in the network (Yoo and Kweon, 2019). The uncertainty-based approaches utilize the quantified uncertainty of the data points to select samples with higher uncertainty. Diversity-based approaches select diverse samples to represent the underlying distribution of the data. Finally, the approaches based on the expected changes in the network select samples with greater impact on network parameters or outputs (Yoo and Kweon, 2019). Key challenges in coreset selection include achieving computational efficiency, a common disadvantage of uncertainty-based approaches, and simultaneously ensuring sample diversity.

Our proposed approach addresses both key challenges in identifying the most influential samples for network training. Specifically: (1) the method achieves computational efficiency in uncertainty quantification by utilizing only one network instead of multiple networks (deep ensemble) or multiple passes of the input through the same network (MC Dropout). Furthermore, (2) the subsampling strategy promotes diversity among samples by grouping the data based on their graph structure and sampling from all groups. As shown in Figure 1, the approach consists of three pipelines: (1) preliminary preparation (Section 3.2), (2) attribute measurement, in which we have chosen prediction uncertainty as the attribute (Section 3.3), and (3) diversity-enabled subsampling (Section 3.4). We use GNNs for graph classification throughout this work; however, the approach is generalizable to any classification task involving GNNs.

3.2. Preliminary Preparation

The goal of the preliminary preparation phase is to train a network that can be effectively utilized to quantify the prediction uncertainty for all data points. We utilize the *self-distillation* approach to train multiple GNN networks simultaneously. Self-distillation is a special case of knowledge distillation, where the same network serves as both the teacher and the student simultaneously, facilitating knowledge transfer within the network (Zhang

et al., 2022; Gou et al., 2021). Self-distillation is especially helpful when used with over-parameterized GNNs as it reduces the computational complexity of training multiple networks separately (Chen et al., 2021). We follow the same strategy as Zhang et al. (2022) by adding a classifier after each layer of the network and utilizing the deepest classifier as the teacher.

During training, each classifier learns to mimic the deepest classifier, i.e., the teacher. As a result, the training process involves reducing the total distillation loss, which is a combination of distillation loss and feature penalty (Zhang et al., 2022; Daneshvar and Samavi, 2026). The total distillation loss for all n training graphs can be defined as

$$L = \frac{1}{n}\sum_{i=1}^{n}(L^i_{\text{dist}} + L^i_{\text{pen}}), \tag{1}$$

where L_{dist} and L_{pen} are the distillation loss and feature penalty respectively.

The distillation loss is the mean of layer-wise weighted sums of two components, cross-entropy and KL-divergence, and can be computed as

$$L^i_{\text{dis}} = \frac{1}{m}\sum_{l=1}^{m}\Big((1-\alpha_l)L^i_{\text{CE}_l} + \alpha_l L^i_{\text{KL}_l}\Big), \tag{2}$$

where $\alpha_l \in [0,1]$ is the imitation parameter for each classifier. The L_{CE} and L_{KL} are the cross-entropy of the classifier output at layer l with the true label, and the KL-divergence between the student's soft labels at layer l and the teacher's soft labels, respectively.

In GNNs, each layer aggregates information from nodes that are one step further away. The first GNN layer collects information from 1-hop neighbours, while the second layer aggregates from neighbours that already contain their own 1-hop information, resulting in 2-hop coverage. Adding more layers extends this process, so node representations include information from increasingly distant neighbours (if available). As a result, shallower layers create less informative representations (features used by the classifier) compared to deeper layers. To capture this notion, we use a penalty for features extracted by shallower models that can be computed as

$$L^i_{\text{pen}} = \frac{1}{m}\sum_{l=1}^{m}\left(\lambda_l L_2\Big(h^i_l, h^i_t\Big)\right), \tag{3}$$

where $\lambda_l > 0$ is the trade-off parameter for each classifier and L_2 is the squared ℓ_2-norm loss. h^i_l and h^i_t are features extracted in layer l and features extracted by the teacher network, respectively. Both the imitation parameter and the trade-off parameter are set to zero for the teacher network.

To make predictions for all data points, we utilize K-fold cross-validation. As a result, each data point will be included in the test dataset exactly once. The predictions of the students and the teacher network are recorded for each data point in the test dataset, which will later be used to distinguish between hard and easy examples.

3.3. Attribute Measurement

During the attribute measurement phase, we utilize the previously trained network to quantify the prediction uncertainty of each data point in the dataset. A data point would be harder for the network to classify if the prediction of shallower classifiers, i.e., classifiers after shallower layers, don't match that of the deepest classifier (Zhang et al., 2022; Daneshvar and Samavi, 2026). This is because deeper classifiers utilize deeper feature extractors, providing them with more informative information. This is especially the case with GNNs, as deeper graph layers, i.e., deeper feature extractors, aggregate information from more distant nodes (Hamilton, 2020). To rank the samples based on how hard they are for the network to classify, we utilize a metric proposed by Daneshvar and Samavi (2026) that captures disagreement between predictions of each classifier. The disagreement between the classifiers' predictions is referred to as the network's prediction uncertainty.

To quantify the network's prediction uncertainty, the metric utilizes a weighted disagreement metric based on a normalized weighted Jensen–Shannon divergence (JSD) (Daneshvar and Samavi, 2026). The uncertainty metric can be computed as

$$UC = \sum_{l=1}^{m} W(l) \times JSD(P_l||P_{teacher}), \tag{4}$$

where $W(l)$ is a bounded weight function ($1 \leq W(l) \leq 2$) that assigns a weight to the classifier at layer l. We utilize the same nonlinear weight function proposed by Daneshvar and Samavi (2026).

The goal of the weight function is to assign higher weights to classifiers based on their depth relative to the deepest classifier. The weight of a classifier at layer l is computed by

$$W(l) = (-(exp(D(l) - L)) + 2)^{\mathbb{1}_{\{y_l \neq y_{teacher}\}}}, \tag{5}$$

where L and $D(i)$ are the total number of network layers and classifier i's distance from the deepest layer, respectively. y_l and $y_{teacher}$ are the predicted class by the classifier after layer l and the predicted class by the teacher classifier, respectively. $\mathbb{1}$ is the indicator function, which returns one if the predicted class of the classifier at layer l (y_l) differs from the predicted class of the teacher classifier ($y_{teacher}$) and zero otherwise.

JSD can be computed as

$$JSD\Big(P_l||P_{outcome}\Big) = \frac{1}{2}\Bigg(KL(P_l||M) + KL(P_{outcome}||M)\Bigg), \tag{6}$$

where $M = \frac{1}{2}(P_l + P_{outcome})$ is a mixture distribution of P_l and $P_{outcome}$. JSD is bounded ($0 \leq JSD \leq \log_b(2)$) as the mixture distribution helps with averaging and smoothing out the values. The uncertainty quantification metric needs to be normalized. The upper bound of UC for a network with m layers, including the teacher network, can be computed as

$$UC_{max} = \sum_{l=1}^{m-1} W(l) \times \log_e(2), \tag{7}$$

where $\log_e$ is the natural logarithm and $W(l)$ computes layer l's. Finally, the normalized uncertainty metric can be computed as

$$UC_{norm} = \frac{UC}{UC_{max}}. \tag{8}$$

Algorithm 1 outlines the overall steps for quantifying the uncertainty associated with each data point in a graph dataset. The algorithm takes as input a graph dataset X, an untrained multi-classifier GNN network M, and the total number of layers l in the network. It outputs a set of uncertainty values U, where each element corresponds to a graph instance in the dataset X. The process begins by splitting the data into multiple folds, followed by training the network M on the training portion of each fold (lines 2–5). Subsequently, for each data point in the testing partition of the current fold, the algorithm computes its uncertainty using the trained model M and appends the resulting values to the set U (lines 6–11). Finally, the algorithm returns the complete uncertainty set U as shown in line 13.

Algorithm 1: Computing uncertainty of all data points
Input: Graph data points X, a GNN network M, total number of layers l
Output: Prediction uncertainty of the data points U

```
1  U ← ∅
2  // Step 1: Split data using K-Fold Cross Validation.
3  foreach (trainSet, testSet) ∈ k_fold(X) do
4      // Step 2: Train the network on trainSet.
5      M ← train(M, trainSet)
6      // Step 3: Compute uncertainty of testSet.
7      foreach testData ∈ testSet do
8          u ← 0
9          u ← uncertainty(M, testData, l) // Compute uncertainty.
10         U ← U ∪ {u} // Add u to the set U
11     end
12 end
13 return U
```

3.4. Diversity-Enabled Subsampling

The diversity-enabled subsampling phase ensures that the selected samples are not only representative of the dataset but also more influential in training the final predictive network. If the subsampling strategy selects only from the data points with the highest uncertainty, we risk selecting only from the outliers (Jeong et al., 2023; Settles and Craven, 2008). Therefore, to promote diversity among samples, we will first group the graphs and then sample graphs proportional to the size of each group. We can utilize unsupervised learning to find hidden patterns in the data without the need for labelled data. One fundamental unsupervised learning approach is clustering. Clustering aims to group data samples based on similarity without requiring labelled data by finding hidden patterns that might exist in the data (Xu and Wunsch, 2005; Huang, 1998). The clustering algorithm will provide disjoint clusters, each containing data that is similar (Na et al., 2010). One of the widely used algorithms in clustering is the K-means clustering algorithm (MacQueen, 1967), which is known for its simplicity.

The clustering approach in K-means clustering begins by selecting k cluster centers, usually chosen randomly. Then the algorithm assigns each data point to a cluster based on its distance to the chosen centers. Euclidean distance is a widely used metric to compute the distance of each data point to the cluster centers (Xu and Wunsch, 2005; Na et al., 2010). The value of k is arbitrary and fixed at the beginning of the algorithm. After assigning each data point to a cluster, the algorithm recalculates the cluster centers by minimizing an objective function. The objective function reduces the distance of each item in the cluster to the cluster average (cluster center), and is computed as (Na et al., 2010):

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - x_i|^2, \tag{9}$$

where C_i, x, and x_i are cluster i, data point x in cluster C_i, and the average of cluster C_i respectively. The process is repeated until there is no change in the cluster centers.

In our approach, we utilize the graphs' information for clustering. Graphs are grouped based on their graph representation. To this end, the graph representations created by the teacher network are utilized. The intuition behind grouping graphs based on their representations is to ensure the sample is representative of the underlying data distribution while taking the graph structure into account.

The sum of the squared distances of samples to their closest cluster center, also known as inertia, is used to optimize the number of clusters (k). However, having too many clusters will result in poor subsampling, as there is a risk of outliers being assigned to the same cluster, and fewer data points in each cluster, resulting in a sample that is not representative of the dataset. Therefore, k should satisfy two conditions: (1) k should be a small number, i.e., a limited number of clusters, and (2) the drop in the sum of the squared distances of samples to their closest cluster center computed with k clusters, should be significant compared to having only one cluster.

To reduce the risk of only selecting outliers in each cluster, the framework groups graphs in each cluster into four percentiles based on the uncertainty values computed by Equation (8), sorted in descending order, 0-25%, 25-50%, 50-75%, and 75-100%. The approach samples the same number of graphs from each cluster with an emphasis on the harder examples, i.e., 50-75% and 75-100% percentiles. To achieve this, in each cluster, random sampling is performed as follows: 45% of samples are drawn from the 75-100% percentile range, 30% from 50-75%, 15% from 25-50%, and 10% from 0-25%. The exact ratio of samples taken from each group is domain-specific and can be set based on the sensitivity of capturing diversity vs. more informative samples in that domain. Given that the requirement is to subsample data points with higher uncertainty, it is reasonable to expect the ratio to increase at higher percentiles. We have included a limited hyperparameter analysis study in Section 4.2.

Algorithm 2 outlines the proposed subsampling strategy. Inputs to this algorithm are the graph representations generated by the teacher model, denoted as G_{rep}, the corresponding uncertainty values U, the desired number of clusters k, and the target number of samples m. The algorithm outputs a subset of data points, X'. As outlined in line 2, the algorithm begins by clustering the data points into k groups based on their graph representations. For each cluster, the number of samples to be selected is determined proportionally to the clus-

ter's size (line 6). Within each cluster, samples are ranked based on their uncertainty values from U, and selection is more focused toward samples with higher uncertainty, specifically those falling into the upper percentiles (lines 7–9). The selected samples are appended to the output set X', which is returned as the final subsampled dataset (line 11).

Algorithm 2: Subsampling data points with higher uncertainty
Input: Graph representations provided by the teacher network G_{rep}, uncertainty of samples U, number of clusters k, and number of samples needed m
Output: Subset of the data points X' with higher uncertainty

1 // Step 1: Grouping graphs.
2 $groups = k\_means(G_{rep}, k)$
3 // Step 2: Select a subsample of size m.
4 $X' \leftarrow \emptyset$
5 **foreach** $group \in groups$ **do**
6 $\quad n = round\Big((size(group)/size(X)) * m\Big)$ // Samples needed from the group.
7 $\quad$ **foreach** $(p_1, p_2, r) \in \{(0, 25, 0.1),\ (25, 50, 0.15),\ (50, 75, 0.3),\ (75, 100, 0.45)\}$ **do**
8 $\quad\quad X' \leftarrow X' \cup subsample\Big(percentile(U,\ group,\ p_1,\ p_2),\ round(r * n)\Big)$
9 $\quad$ **end**
10 **end**
11 **return** X'

4. Experimental Evaluations

The proposed framework identifies and selects samples that provide superior utility for network training compared to random sampling. We assess the impact of our importance-based subsampling method on network performance, comparing it directly with random sampling. To do so, we evaluate the performance of an identical network trained under three conditions: (1) on the entire dataset using K-fold cross-validation, (2) on randomly sampled data, and (3) on data sampled using the proposed framework. We anticipate observing improved network performance when training with the limited data selected by our subsampling framework, compared to random sampling, and achieving comparable performance to training the network on all the training data.

4.1. Experimental Setup

Dataset: We have utilized two linked datasets containing data from medical health records and mental health questionnaires from real-world patients. The prediction task involves determining whether a patient will be admitted to the emergency department (ED) within 180 days of completing the mental health questionnaire. The datasets comprise ED visits of 1,086 unique patients, along with their responses to the mental health outpatient questionnaire. There are 281 patients who have visited the ED within 180 days of their initial outpatient visit. To address dataset imbalance, we balanced the "Not Admitted" and "Admitted" groups by repeatedly undersampling the former to 281 patients. We utilized the available information to create a graph for each patient based on their medical

history. We then utilized the mental health questionnaire responses to integrate additional data into each patient's graph. Appendix A provides more details on the dataset and the generated graph for each patient. Additionally, we assessed the generalizability of our proposed method using the Enzymes dataset (Morris et al., 2020). The details and results are included in Appendix B.

Network: We utilized two networks. The first network is a multi-classifier GNN, used by the subsampling framework to sample patients from the main dataset, i.e., the medical records dataset, trained using the self-distillation approach. The network consists of three GraphConv (Morris et al., 2021) layers, followed by a ReLU activation function and a batch normalization layer. A final readout layer is applied before each classifier, which consists of a global mean pooling layer. The purpose of utilizing a readout layer is to combine node representations into a single, final graph representation for use in graph classification. The network is trained on the same 180-day ED admission prediction task. It is worth noting that the first network is only utilized for uncertainty quantification; therefore, the network's performance is not a concern.

The second network utilizes the augmented patient graph, generated by integrating information from both datasets, to predict whether a patient would be admitted to the ED within 180 days of their initial mental health assessment. The network consists of a single layer of GraphConv (Morris et al., 2021), followed by similar ReLU and batch normalization layers. A similar readout layer is applied to the node representations to create a graph representation, which is then fed to the classifier.

Choice of Baselines: We considered two baselines to compare with our subsampling framework: the full dataset without subsampling and random subsampling of the original dataset. Existing graph subsampling techniques were excluded, as they are designed either for node classification or for extracting subgraphs from a single graph, and thus are not applicable to our task of subsampling entire graphs from a larger pool of graphs.

Training and Hardware Specifications: All networks were implemented using Python version 3.9 and PyTorch version 1.13.1. The networks have been trained on a GPU (NVIDIA GeForce RTX 3050) with CUDA version 12.5. For optimization, we used the Adam Optimizer.

4.2. Hyperparameter Analysis: Different Subsampling Ratios

We conducted a limited hyperparameter analysis study to examine the impact of different subsampling ratios within the proposed approach. The number of clusters was fixed at $k = 10$, and the network was trained on the augmented clinical dataset. Table 1 reports the performance of the network trained on data selected according to various ratios. The best results, in terms of Accuracy, Precision, Recall, and F1-Score, were obtained when selecting 45% of samples from the 75-100% percentile range, 30% from 50-75%, 15% from 25-50%, and 10% from 0-25%, as highlighted in bold. Accordingly, these ratios were adopted in our subsampling framework. The observed performance drop for ratios that emphasize a larger proportion of samples from the top percentile (75-100%) further suggests that focusing too heavily on highly uncertain data points increases the likelihood of including outliers, thereby affecting the network's ability to learn accurate patterns.

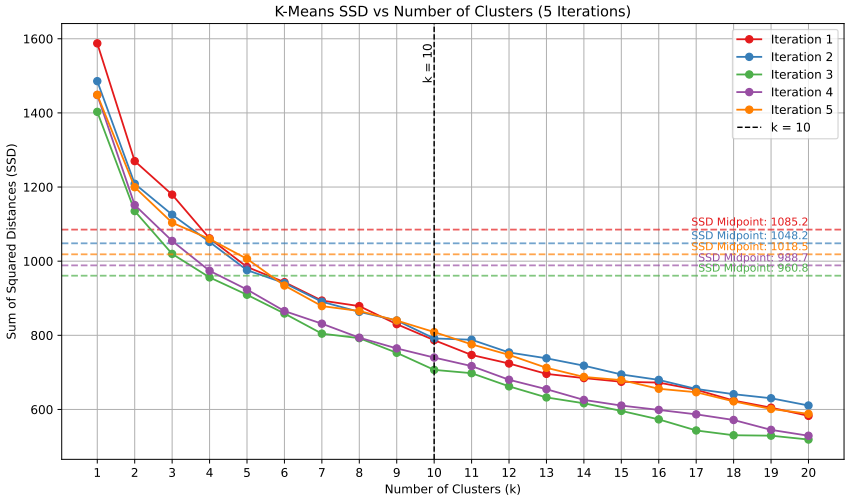

Figure 2: Candidate k versus sum of squared distances of samples to their cluster center, repeated for 5 iterations.

4.3. Hyperparameter Analysis: Different Values of k in K-Means Clustering

To select a suitable value for k, we employed the *Elbow Method*. During each iteration, we used 33% of the data to fit a K-means clustering model and computed the inertia. Figure 2 illustrates how different values of k ($1 \leq k \leq 20$) affect the inertia. As can be observed, $k = 10$ represents an 'elbow' point where the rate of decrease in inertia significantly diminishes. This observation was further validated by experimentally evaluating the effect of different k values on the overall subsampling framework. As shown in Table 2, increasing the number of clusters from 5 to 10 results in improved performance; however, performance decreases as the number of clusters approaches 20. The decline in performance occurs because a larger number of clusters can reduce the number of samples in each cluster, thereby increasing the risk of selecting more outliers and diminishing overall sample diversity. Thus, $k = 10$ yields the best performance among other candidates, as it (1) is small enough for computational efficiency yet large enough to provide sufficient diversity, and (2) substantially reduces the clustering error (by more than half), satisfying both conditions for selecting k as discussed in Section 3.4.

Table 1: Performance of the approach with different subsampling ratios.

Ratios	Accuracy	ROC AUC	Precision	Recall	F1-Score
10-15-35-40	0.56 ± 0.04	0.63 ± 0.05	0.56 ± 0.03	0.6 ± 0.17	0.57 ± 0.09
10-15-30-45	**0.61 ± 0.04**	0.63 ± 0.03	**0.6 ± 0.05**	**0.68 ± 0.03**	**0.64 ± 0.03**
10-15-25-50	0.59 ± 0.04	**0.64 ± 0.05**	0.59 ± 0.04	0.65 ± 0.07	0.62 ± 0.02
10-15-20-55	0.55 ± 0.08	0.6 ± 0.07	0.56 ± 0.08	0.57 ± 0.09	0.56 ± 0.07

4.4. Results and Discussion

Table 3 shows the performance of the network. The same number of samples was selected for both the random sampling and the proposed framework (roughly 278 samples). Additionally, we trained the same GNN network without subsampling to compare the performance of the network trained with and without subsampling. The GNN network takes the augmented patient graph as input and predicts whether a patient would be at risk of admission to the ED within 180 days of their initial mental health assessment. We have measured accuracy, ROC AUC, precision, recall, and F1 score to compare the performance of the approaches. As shown in the table, subsampling using the proposed framework has improved the results across all metrics. Note that training the network with a smaller sample size, when samples are selected using the proposed framework, yields a performance comparable to (and even slightly improved over) training the network using the entire training dataset.

The improvement in Recall and F1 Score (marked with an asterisk in Table 3) is statistically significant when comparing the proposed framework with the random sampling strategy based on t-tests at $p < 0.05$. This improvement has been verified under both Bonferroni and Benjamini-Hochberg corrections, indicating that the network's ability to identify positive cases and maintain a strong precision-recall balance correctly is a statistically reliable improvement. Our proposed subsampling framework demonstrates a consistent, albeit slight, improvement across most evaluation metrics when compared to training with the complete dataset. Of these improvements, only the gain in Recall (marked with a double dagger in Table 3) was found to be statistically significant, verified by t-tests at $p < 0.05$.

Finding: The proposed subsampling framework allows for identifying samples that are more informative in training the network. This approach enables researchers to purposefully select samples from the dataset, selecting those most helpful for training the networks, rather than relying on random selection. This is particularly important in utilizing linked datasets, as it facilitates subsampling for data integration.

5. Conclusion

In this paper, we present an efficient framework for data subsampling based on the prediction uncertainty of the available data points. The purpose of the approach is to help sample data points from a primary dataset for additional data integration from other available, and often limited, datasets. The additional data will help augment the primary dataset and potentially improve network performance. Using the primary dataset, the framework utilizes self-distillation to quantify the prediction uncertainty of the data points. Using

Table 2: Comparison of different values of k on final network performance.

Number of Clusters	Accuracy	ROC AUC	F1 Score
5	0.5 ± 0.03	0.55 ± 0.06	0.51 ± 0.05
10	$\mathbf{0.61 \pm 0.04}$	$\mathbf{0.63 \pm 0.04}$	$\mathbf{0.64 \pm 0.03}$
15	0.52 ± 0.08	0.57 ± 0.06	0.53 ± 0.07
20	0.5 ± 0.04	0.52 ± 0.11	0.54 ± 0.07

Table 3: Results of performance improvement using the proposed framework.

Method	Accuracy	ROC AUC	Precision	Recall	F1 Score
Without Subsampling	0.59 ± 0.07	0.63 ± 0.06	0.59 ± 0.07	0.61 ± 0.1	0.59 ± 0.07
Random Sampling	0.57 ± 0.08	0.59 ± 0.07	0.57 ± 0.08	0.57 ± 0.1	0.57 ± 0.08
Proposed Framework	**0.61 ± 0.04**	**0.63 ± 0.04**	**0.6 ± 0.05**	**0.68 ± 0.03***‡	**0.64 ± 0.03***

K-means clustering, the data points are grouped into k clusters based on their graph representation. In each cluster, based on the data point's prediction uncertainty, the data is grouped into 0-25%, 25-50%, 50-75%, and 75-100% percentiles. Finally, data is sampled from each cluster with an emphasis on higher percentiles. The evaluations have shown that when sampling data using the proposed framework, the network trained on the augmented dataset has a statistically significant improvement in performance compared to a network trained on randomly sampled augmented data. Additionally, the network exhibits comparable performance to one trained on the whole augmented training dataset.

As a future direction, we plan to incorporate explainability into the uncertainty quantification approach to help users better understand the data selection process. Additionally, we plan to develop an approach to find the right number of samples. Finally, we plan to evaluate the approach on additional datasets.

Acknowledgments

This study is supported by the *Pediatric Mental Health Learning Health System* research project and funded by the *Hamilton Health Sciences RFA Research Strategic Initiative Program* and *Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery* grants. We would like to extend our deepest gratitude to Dr. Roberto Sassi (Associate Professor and Division Head, Child & Adolescent Psychiatry, University of British Columbia), Dr. Paulo Pires (Psychologist and Clinical Director for the CYMHP) and Dr. Laura Duncan (Assistant Professor, Psychiatry & Behavioural Neurosciences, McMaster University), for their valuable insights. This research was also undertaken thanks in part to funding from the *Canada First Research Excellence Fund* at Toronto Metropolitan University.

References

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryghZJBKPS.

Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. Sequential graph convolutional network for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9583–9592, June 2021.

Jiayi Chen, Benteng Ma, Hengfei Cui, and Yong Xia. Think twice before selection: Federated evidential active learning for medical image analysis with domain shifts. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11439–11449, June 2024.

Yuzhao Chen, Yatao Bian, Xi Xiao, Yu Rong, Tingyang Xu, and Junzhou Huang. On self-distilling graph neural network, 2021. URL https://arxiv.org/abs/2011.02255.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11933–11944. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/64254db8396e404d9223914a0bd355d2-Paper.pdf.

Hirad Daneshvar and Reza Samavi. Gnn's uncertainty quantification using self-distillation. In Daniele Cafolla, Timothy Rittman, and Hao Ni, editors, *Artificial Intelligence in Healthcare*, pages 31–45, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-00656-1.

Mucong Ding, Yinhan He, Jundong Li, and Furong Huang. Spectral greedy coresets for graph neural networks, 2024. URL https://arxiv.org/abs/2405.17404.

Dobrik Georgiev Georgiev, Pietro Lio, Jakub Bachurski, Junhua Chen, Tunan Shi, and Lorenzo Giusti. Beyond erdos-renyi: Generalization in algorithmic reasoning on graphs. In *The Second Learning on Graphs Conference*, 2023. URL https://openreview.net/forum?id=TTxQAkg9QG.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 6 2021. ISSN 1573-1405. doi: 10.1007/s11263-021-01453-z.

Mridul Gupta, Sahil Manchanda, Hariprasad Kodamana, and Sayan Ranu. Mirage: Model-agnostic graph distillation for graph classification, 2024. URL https://arxiv.org/abs/2310.09486.

William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 09 1998.

Mika Sarkin Jain, Stefanie Jegelka, Ishani Karmarkar, Luana Ruiz, and Ellen Vitercik. Subsampling graphs with gnn performance guarantees, 2025. URL https://arxiv.org/abs/2502.16703.

Yuna Jeong, Myunggwon Hwang, and Wonkyung Sung. Training data selection based on dataset distillation for rapid deployment in machine-learning workflows. *Multimedia Tools and Applications*, 82(7):9855–9870, 2023.

Wei Jin, Xianfeng Tang, Haoming Jiang, Zheng Li, Danqing Zhang, Jiliang Tang, and Bing Yin. Condensing graphs via one-step gradient matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 720–730, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539429. URL https://doi.org/10.1145/3534678.3539429.

Siddharth Joshi and Baharan Mirzasoleiman. Data-efficient contrastive self-supervised learning: Most beneficial examples for supervised learning contribute the least. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15356–15370. PMLR, 07 2023. URL https://proceedings.mlr.press/v202/joshi23b.html.

Angelos Katharopoulos and Francois Fleuret. Not all samples are created equal: Deep learning with importance sampling. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2525–2534. PMLR, 07 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pages 281–298. University of California press, 1967.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks, 2021.

Shi Na, Liu Xumin, and Guan Yong. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 63–67, 2010. doi: 10.1109/IITSI.2010.74.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pages 1070–1079, 2008.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34201–34227. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6b9aa8f418bde2840d5f4ab7a02f663b-Paper-Conference.pdf.

Rui Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005. doi: 10.1109/TNN.2005.845141.

Zhe Xu, Yuzhong Chen, Menghai Pan, Huiyuan Chen, Mahashweta Das, Hao Yang, and Hanghang Tong. Kernel ridge regression-based graph dataset distillation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2850–2861, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599398. URL https://doi.org/10.1145/3580305.3599398.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403, 2022. doi: 10.1109/TPAMI.2021.3067100.