Abstract—Hyperspectral image (HSI) classification is a topic of active research. One of the main challenges of HSI classification is the lack of reliable labelled samples. Various semi-supervised and unsupervised classification methods are proposed to handle the low number of labelled samples. Chief among them are graph convolution networks (GCN) and their variants. These approaches exploit the graph structure for semi-supervised and unsupervised classification. While several of these methods implicitly construct edge-weights, to our knowledge, not much work has been done to estimate the edge-weights explicitly. In this article, we estimate the edge-weights explicitly and use them for the downstream classification tasks - both semi-supervised and unsupervised. The proposed edge-weights are based on two key insights - (a) Ensembles reduce the variance and (b) Classes in HSI datasets and feature similarity have only one-sided implications. That is, while same classes would have similar features, similar features do not necessarily imply same classes. Exploiting these, we estimate the edge-weights using an aggregate of ensembles of subsamples of features. These edge weights are evaluated for both semi-supervised and unsupervised classification tasks. The evaluation for semi-supervised tasks uses Random-Walk based approach. For the unsupervised case, we use a simple filter using a graph convolution network (GCN). In both these cases, the proposed edge weights outperform the traditional approaches to compute edge-weights - Euclidean distances and cosine similarities. Fascinatingly, with the proposed edge-weights, the simplest GCN obtained results comparable to the recent state-of-the-art.

Index Terms—Hyperspectral Image Classification

I. INTRODUCTION

HYPERSPECTRAL Image (HSI) Classification is an area of active research [1]–[4], thanks to it’s wide application ranging from mineral exploration [5] to military reconnaissance [6]. This is due to the rich spatial and spectral information available within an HSI dataset. However, HSI classification depends largely on the ability to obtain noise-free ground-truth labels. This is usually costly and sometimes infeasible. Hence, several research studies focussed on semi-supervised or unsupervised classification techniques.

Semi-supervised methods try to use the large number of unlabelled data points along with limited labelled data for classification. A common approach is active learning where the data points are actively selected and labelled [7]–[9]. This procedure is iteratively repeated until all points are labelled. On the other hand, unsupervised methods do not utilize the labelled data but instead identify the conspicuous classes within the dataset and map them to ground truth labels. [10] provides a detailed review of the clustering approaches used for HSI clustering.

Graph based methods for HSI classification have been widely used for both semi-supervised or unsupervised approaches. [11] proposes a semi-supervised graph based HSI classification. [12] proposes semi-supervised graph neural networks, [13], [14] utilizes graph convolution networks for semi-supervised HSI classification. In case of unsupervised approaches, the most widely used approach is that of spectral clustering [15]. Recently in [1], [2] graph based methods are combined with other approaches such as subspace clustering to obtain better results. [16] constructs the graph using manifold-based sparse representation and graph embedding. [17] proposes multi-structure unified discriminative embedding (MUDE) which designs the intraclass and interclass neighborhood structure graphs.

While several graph based approaches implicitly compute edge-weights, we are not aware of any works which explicitly estimates the edge-weights. In this article we propose an approach to estimate the edge-weights for HSI datasets referred to as ESW Edge-Weights - Ensemble Stochastic Watersheds based Edge-Weights. Figure [1] visualizes the ESW Edge-Weights obtained by our proposed approach. Observe that the object boundaries are easily discernible. We show that ESW edge-weights perform better than existing approaches such as Euclidean distances or cosine similarities for both semi-supervised and unsupervised classification. Moreover, we also show that using the proposed edge-weights, even the simplest GCN approach results in scores better than the recent state-of-the-art.

The main contributions in this article are as follows:

(i) We propose a novel approach to estimate the edge-weights within HSI datasets, referred to as ESW Edge-Weights.
(ii) We show that these edge-weights are superior than the classical ones such as Euclidean distance or cosine similarity using Random Walk semi-supervised classification.
(iii) We show how vanilla graph convolution networks (GCNs) (as proposed in [18]) can be modified to incorporate edge-weights. This results in an unsupervised learning scheme better than the unweighted approach. Moreover
we show that using the proposed edge-weights, we obtain results which are better than the recent state-of-the-art.

II. ESTIMATE EDGE WEIGHTS USING ENSEMBLES OF STOCHASTIC WATERSHEDS

Notation: Let $X$ denote the HSI dataset with $nc$ columns, $nr$ rows and $nz$ bands. Let $G = (V, E, W)$ denote the edge-weighted graph, where $V$ denotes the set of vertices. Each vertex corresponds to a pixel within HSI dataset. Hence for the graph corresponding to dataset $X$, one would have $nc \times nr$ number of vertices. $E \subseteq V \times V$ denotes the subset of edges, and $W : E \rightarrow \mathbb{R}^+$ denotes the set of edge weights. In the rest of this article we use the 4-adjacency edges as $E$, unless explicitly stated otherwise.

In this section we ask and answer the question - Given and edge $e = (x, y)$, what is the best estimate of the edge weight $\hat{W}(e)$? An ideal edge-weight would reflect the chance that the vertices in the edge belong to a different class. That is,

$$\hat{W}(e) = P(C(x) \neq C(y))$$

(1)

where $C(.)$ denotes the class to which the vertex belongs. Before stating the algorithm, we review the \textit{seeded watershed} algorithm \[19\] in algorithm \[1\]. Stated simply, seeded watershed labels all the vertices greedily using the Euclidean edge-weights.

\begin{algorithm}[H]
\caption{Seeded Watershed}
\textbf{Input:} $V$ is the set of vertices, $E$ is the set of edges and $X$ denotes the HSI dataset. A subset of labelled points $V_k \subset V$.
\textbf{Output:} Labels for each of the vertices $\hat{C}(.)$.
\begin{algorithmic}[1]
\State Initialize the labels $\hat{C}(v)$ for all $v \in V_k$, and $\hat{C}(v) = \text{NULL}$ for all $v \notin V_k$.
\State Set $W(x, y) = \|X(x) - X(y)\|$, i.e initialize the edge weights using Euclidean distances for all edges $e = (x, y)$.
\State Sort the edges $E$ in increasing order $\text{w.r.t } W$.
\State Initialize the union-find data structure UF,
\For {$e = (x, y)$ in sorted edge set $E$}
\If {both $x$ and $y$ are labelled}
\State do nothing
\Else
\State UF.union$(x, y)$
\State Assign same label for $x$ and $y$.
\EndIf
\EndFor
\State Label each vertex of the connected component using labels $V_k$.
\State \textbf{return} Labels of the vertices.
\end{algorithmic}
\end{algorithm}

\textit{Stochastic Watershed} uses the seeded watershed algorithm using random subset of features and random subset of vertices as seeds. Algorithm \[2\] ensembles several stochastic watersheds. Each vertex within the sampled vertices is given a distinct label. These labels are propagated using the seeded watershed. This procedure is repeated several times to obtain ESW (Ensemble Stochastic Watershed) Edge-Weights.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Visualizing the ESW Edge-Weights. (a) Groundtruth objects in the Indianpines dataset. (b) Groundtruth objects in Salinas dataset. (c) Proposed ESW Edge-Weights on Indianpines dataset. (d) Proposed ESW Edge-Weights on Salinas dataset. Observe that the boundaries are easily discernible in the edge weights plots. To visualize the edge-weights we use the 2D cubical complexes \[20\].}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.png}
\caption{Distribution of edge-weights for same class and different class on Indianpines dataset. Here we plot this histogram of Euclidean distances on various subset of features between vertices of same/different class. Observe that the edge-weights for same class lie at the lower end, while edge-weights for different class are uniformly distributed.}
\end{figure}

Why does algorithm \[2\] work?

The two main insights which lead to algorithm \[2\] are
\begin{enumerate}
\item A commonly used statistical observation - averaging over samples from related distribution reduces the variance.
\item Typically, one expects that feature distances and classes have two sided equivalence - That is, similar features imply same classes and same classes imply similar features. However, in case of HSI datasets, we have that on a subset of features
\end{enumerate}

\begin{align*}
\text{Similar features on a subset} & \Rightarrow \text{Same Class} \quad (2) \\
\text{Similar features on a subset} & \Rightarrow \text{Same Class} \quad (3)
\end{align*}

This is verified in figure \[2\] Observe that the edge-weights for same class lie at the lower end, while edge-weights
for different class are uniformly distributed. Thus, one cannot simply consider an average of edge weights for ensembles. Moreover, note that average over Euclidean distances using random subset of features result in Euclidean distances, that is

\[
\frac{1}{N} \sum_{i=1}^{N} \|X_{s_i}(e_x) - X_{s_i}(e_y)\|_2 \approx \|X(e_x) - X(e_y)\|_2 \tag{4}
\]

where \(e = (e_x, e_y)\) denotes a specific edge, \(s_i\) denotes a random subset of features and \(N\) denotes the size of the ensemble.

\[
\text{Algorithm 2 Computing ESW (Ensemble Stochastic Watershed) Edge-Weights}
\]

**Input:** \(V\) is the set of vertices, \(E\) is the set of edges and \(X\) denotes the HSI dataset. \(N\) denotes the number of repetitions.

**Output:** Estimated weights, \(\hat{W} : E \to \mathbb{R}^+\)

1. Initialize \(\hat{W}(e) = 0\) for all edges \(e\). Let \(c\) (initialized to 0) denote the count.
2. **repeat**
   3. Select a subset of features of size \(\kappa_{f_1}, S_{k_1}\).
   4. Select a random sample from the set of vertices of size \(\kappa_{v}, V_{k}\).
   5. Assign distinct label (pseudo-labels) to each randomly selected vertex. There would be \(\kappa_v\) distinct labels.
   6. Propagate the labels using seeded watershed as described in algorithm [1]. Use the subset of sampled vertices \((V_{k})\) as seeds and the sampled subset of features \((S_{k})\) for computing edge-weights. Let \(\hat{C}(\cdot)\) denote the labelling obtained.
   7. Add 1 to the estimated edge weight if the labels of its vertices are different. That is,

\[
\hat{W}(e_x, e_y) = I(\hat{C}(e_x) \neq \hat{C}(e_y))
\]

where \(I(\cdot)\) denotes the indicator function. Also, add 1 to the count \(c\).
8. **until** \(c\) is less than \(N\)
9. Set \(\bar{W}(e) = \hat{W}(e)/N\) to obtain an estimate between \([0, 1]\).

In this article we consider an alternate approach to estimating edge-weights using ensembles of stochastic watershed, described in algorithm [2]. Here we consider the artificial labels constructed using the random samples and features (lines 4, 5 and 6 of algorithm [2]). These artificial labels are then converted to edge-weights using the indicator function (line 7 in algorithm [2]). These edge-weights are then averaged to obtain the final estimate (line 9 in algorithm [2]).

Figure [1] visualizes the edge weights obtained using algorithm [2]. In what follows, we quantitatively evaluate the edge weights obtained using Random Walk (RW) and vanilla Graph Convolution Network (GCN). The code is available at https://github.com/ac20/EnsembleEdgeWeightsHSI

**Remark on Datasets:** We consider the following three HSI datasets for evaluation. All these datasets have been downloaded from http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

1) **Indianpines:** This is acquired by AVIRIS spectograph, and has the size \(145 \times 145\). There are 200 spectral bands. The groundtruth consists of 16 different land cover classes.

2) **Salinas:** This is acquired by AVIRIS spectograph, and has the size \(512 \times 217\). There are 204 spectral bands. The groundtruth consists of 16 different land cover classes.

3) **Pavia Centre:** This is acquired by the ROSIS sensor and has the size \(1096 \times 490\). There are 102 spectral bands and groundtruth consists of 9 different land cover classes.

III. Evaluating ESW Edge-Weights Using Random Walk (Semi-Supervised)

Random Walk (RW) is a classical tool for image segmentation [21]. It uses an edge-weighted graph and few labelled pixels to obtain the segmentation of the image. This is similar to the semi-supervised learning paradigm. In this section we compare our edge-weights - ESW Edge-Weights with the classic measures - Euclidean distance and cosine similarity.

**Remark:** For sake of simplicity we describe the RW procedure for 2 labels - \([0, 1]\). It extends to multi-label scenario naturally as described in [21].

To recap, let \(G = (V, E, W)\) denote the edge-weighted graph. Construct the Laplacian matrix (called Random Walk Laplacian [11]), \(L\), where \(L_{ij} = \sum_j w_{ij}\) if \(i = j\) and \(L_{ij} = -w_{ij}\) if \(i \neq j\). Here \(w_{ij}\) indicates the edge-weight between pixel \(i\) and \(j\). Let \(S_0\) denote the set of vertex-indices labelled 0, \(S_1\) denote the set of vertex-indices labelled 1, and \(U\) denote the unlabelled vertex-indices. We then solve

\[
\begin{align*}
\text{minimize} & \quad Tr(x^T L x) \\
\text{subject to} & \quad x[S_0] = 0 \quad \text{and} \quad x[S_1] = 1
\end{align*}
\]

Classical approaches to obtain edge-weights \(w_{ij}\) are - (a) Using Euclidean distances

\[
\begin{align*}
w_{ij} &= \exp(-\beta \|X(i) - X(j)\|_2) \tag{6}
\end{align*}
\]

where \(X(i)\) indicates the feature vector of pixel \(i\) and \(\|X(i) - X(j)\|_2\) indicates the Euclidean distance, and (b) Cosine similarity

\[
\begin{align*}
w_{ij} &= \frac{X(i)^T X(j)}{\|X(i)\| \|X(j)\|} \tag{7}
\end{align*}
\]

We compare the proposed ESW edge-weights obtained by algorithm [2] with the Euclidean edge-weights [6] and cosine edge-weights (7). Figure [3] shows the plots of overall accuracy versus number of samples used as seeds. Observe that in all these cases, the proposed edge-weights outperform the traditional methods. (Remark: These results are obtained using the average score over 50 iterations.)

IV. Evaluating ESW Edge-Weights Using Vanilla Graph Convolution Network (Unsupervised)

While the previous section uses a semi-supervised technique for evaluating edge-weights, this section uses an unsupervised method - Graph Convolution Networks (GCN). We use the most basic version of GCN as described in [18]. The approach
I eigenvalues of $\Lambda$

Here $L$ best score is recorded. OA is computed after every iteration and correctly classified. OA is measured as a percentage of samples with the groundtruth clusters using Hungarian algorithm [22]. Overall accuracy (OA) is measured as a percentage of samples covering both semi-supervised and unsupervised scenarios.

Firstly, using accuracies obtained via Random-Walk, we compare the proposed edge-weights with unweighted GCN. The proposed edge-weights provides significant improvement over the unweighted approach. Table I reports the optimal results we obtained for each dataset. As a baseline we provide the results from [1] which is the recent state-of-the-art. Observe that proposed Weighted GCN outperforms the baseline method.

V. CONCLUSION AND PERSPECTIVES

To summarize, in this article we propose a novel approach to estimate the edge-weights of the HSI Datasets. This approach exploits 2 fundamental insights - (a) Ensembles reduce the variance and (b) In hyperspectral datasets we have similar features.

The above procedure is repeated 10 times and the average is computed for each iteration. This is plotted in figure 4. We compare the proposed edge-weights with unweighted GCN. The proposed edge-weights provides significant improvement over the unweighted approach. Table I reports the optimal results we obtained for each dataset. As a baseline we provide the results from [1] which is the recent state-of-the-art. Observe that proposed Weighted GCN outperforms the baseline method.

V. CONCLUSION AND PERSPECTIVES

To summarize, in this article we propose a novel approach to estimate the edge-weights of the HSI Datasets. This approach exploits 2 fundamental insights - (a) Ensembles reduce the variance and (b) In hyperspectral datasets we have similar features.

We now modify the above approach for edge-weighted graph. For general edge-weights $w_{ij}$ eigenvalues of Laplacian no longer belong to $[0, 2]$, but belong to $[0, \lambda_{max}]$ where $\lambda_{max}$ denotes the maximum eigenvalue. Thus we define the one step graph convolution as

$$X^{(k+1)} = (I - \frac{1}{\lambda_{max}}L^{norm})X^{(k)}$$

These features are then used for spectral clustering to obtain class labels.

**Evaluation Procedure**: To evaluate the proposed ESW Edge-weights we perform GCN with and without weights for 200 iterations. After each iteration, the spectral clustering is performed to get the clusters. These clusters are matched with the groundtruth clusters using Hungarian algorithm [22]. Overall accuracy (OA) is measured as a percentage of samples correctly classified. OA is computed after every iteration and best score is recorded.

![Graph](image)

**Fig. 3.** Random Walk evaluation of ESW Edge-Weights. We compare the proposed edge-weights with the traditional edge-weights obtained by Euclidean distances [4] and cosine similarity [7]. We evaluate using (a) Indianpines dataset. (b) Pavia Centre dataset and (c) Salinas dataset. Observe that in all cases ESW Edge-weights outperform the traditional measures.

**TABLE I**

| Method          | Indianpines | PaviaCentre | Salinas |
|-----------------|-------------|-------------|---------|
| Baseline [1]    | 53.09       | 90.09       | 75.31   |
| GCN (Unweighted)| 54.07       | 90.62       | 75.31   |
| GCN (Weighted)  | 56.26       | 90.62       | 76.80   |

The above procedure is repeated 10 times and the average is computed for each iteration. This is plotted in figure 4. We compare the proposed edge-weights with unweighted GCN. The proposed edge-weights provides significant improvement over the unweighted approach. Table I reports the optimal results we obtained for each dataset. As a baseline we provide the results from [1] which is the recent state-of-the-art. Observe that proposed Weighted GCN outperforms the baseline method.

V. CONCLUSION AND PERSPECTIVES

To summarize, in this article we propose a novel approach to estimate the edge-weights of the HSI Datasets. This approach exploits 2 fundamental insights - (a) Ensembles reduce the variance and (b) In hyperspectral datasets we have similar features.

The above procedure is repeated 10 times and the average is computed for each iteration. This is plotted in figure 4. We compare the proposed edge-weights with unweighted GCN. The proposed edge-weights provides significant improvement over the unweighted approach. Table I reports the optimal results we obtained for each dataset. As a baseline we provide the results from [1] which is the recent state-of-the-art. Observe that proposed Weighted GCN outperforms the baseline method.

V. CONCLUSION AND PERSPECTIVES

To summarize, in this article we propose a novel approach to estimate the edge-weights of the HSI Datasets. This approach exploits 2 fundamental insights - (a) Ensembles reduce the variance and (b) In hyperspectral datasets we have similar features.

We now modify the above approach for edge-weighted graph. For general edge-weights $w_{ij}$ eigenvalues of Laplacian no longer belong to $[0, 2]$, but belong to $[0, \lambda_{max}]$ where $\lambda_{max}$ denotes the maximum eigenvalue. Thus we define the one step graph convolution as

$$X^{(k+1)} = (I - \frac{1}{\lambda_{max}}L^{norm})X^{(k)}$$

These features are then used for spectral clustering to obtain class labels.

**Evaluation Procedure**: To evaluate the proposed ESW Edge-weights we perform GCN with and without weights for 200 iterations. After each iteration, the spectral clustering is performed to get the clusters. These clusters are matched with the groundtruth clusters using Hungarian algorithm [22]. Overall accuracy (OA) is measured as a percentage of samples correctly classified. OA is computed after every iteration and best score is recorded.
The main outcome of the above article is the emphasis on using edge-weights. While several works use graphs and related approaches, not much work was done to incorporate edge-weights. In this article we estimated the edge-weights explicitly using ensembles of stochastic watershed. Incorporating these edge-weights improved the results of the existing approaches such as GCN. This is mainly due to the fact that edge-weights have the ability to capture not just second order relations (edges) but also higher order relations (hyperedges). Adapting other graph based approaches to incorporate edge-weights is considered for future work.

ACKNOWLEDGMENT
AC and SD would like to thank APPCAIR and BITS Pilani K K Birla Goa Campus. The work of AC was supported by BITS-Pilani, K. K. Birla Goa Campus, under Grant BPGC/RIG/2021-22/09-2021/02. SD would like to acknowledge the funding received from BPGC/RIG/2020-21/11-2020/01 (Research Initiation Grant), GOA/ACG/2021-22/Nov/05 (Additional Competitive Grant) both provided by BITS-Pilani K K Birla Goa Campus.

REFERENCES
[1] N. Huang, L. Xiao, J. Liu, and J. Chanussot, “Graph convolutional sparse subspace clustering with nonnegative orthogonal factorization for large hyperspectral images,” IEEE Transactions on Geoscience and Remote Sensing, pp. 1–16, 2021.
[2] Y. Cai, Z. Zhang, Z. Cai, X. Liu, X. Jiang, and Q. Yan, “Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image,” IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 5, pp. 4191–4202, 2021.
[3] H. Patel and K. P. Uplab, “A shallow network for hyperspectral image classification using an autoencoder with convolutional neural network,” Multimedia Tools and Applications, vol. 81, no. 1, pp. 695–714, 2022. [Online]. Available: https://doi.org/10.1007/s11042-021-11422-w
[4] P. Ghuismi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, “Advanced spectral classifiers for hyperspectral images: A review,” IEEE Geoscience and Remote Sensing Magazine, vol. 5, no. 1, pp. 8–32, 2017.
[5] F. Kruse, J. Boardman, and J. Huntington, “Comparison of airborne hyperspectral data and co-1 hyperion for mineral mapping,” IEEE Transactions on Geoscience and Remote Sensing, vol. 41, no. 6, pp. 1388–1400, 2003.
[6] M. Shimoni, R. Haelterman, and C. Perneel, “Hyperspectral imaging for military and security applications: Combining myriad processing and sensing techniques,” IEEE Geoscience and Remote Sensing Magazine, vol. 7, no. 2, pp. 101–117, 2019.
[7] L. Yang, S. Yang, P. Jin, and R. Zhang, “Semi-supervised hyperspectral image classification using spatio-spectral laplacian support vector machine,” IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 3, pp. 651–655, 2014.
[8] C. Liu, J. Li, and L. He, “Superpixel-based semisupervised active learning for hyperspectral image classification,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 1, pp. 357–370, 2019.
[9] H. Wu and S. Prasad, “Semi-supervised deep learning using pseudo labels for hyperspectral image classification,” IEEE Transactions on Image Processing, vol. 27, no. 3, pp. 1259–1270, 2018.
[10] H. Zhai, H. Zhang, P. Li, and L. Zhang, “Hyperspectral image clustering: Current achievements and future lines,” IEEE Geoscience and Remote Sensing Magazine, pp. 0–0, 2021.
[11] G. Camps-Valls, T. V. Buades Marsheva, and D. Zhou, “Semi-supervised graph-based hyperspectral image classification,” IEEE Transactions on Geoscience and Remote Sensing, vol. 45, no. 10, pp. 3044–3054, 2007.
[12] Y. Ding, X. Zhao, Z. Zhang, W. Cai, N. Yang, and Y. Zhan, “Semi-supervised locality preserving dense graph neural network with arma filters and context-aware learning for hyperspectral image classification,” IEEE Transactions on Geoscience and Remote Sensing, pp. 1–12, 2021.
[13] H. Wang, Y. Cheng, C. L. P. Chen, and X. Wang, “Semi-supervised classification of hyperspectral image based on graph convolutional broad network,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 2995–3005, 2021.
[14] A. Sellami and S. Tabbone, “Deep neural networks-based relevant latent representation learning for hyperspectral image classification,” Pattern Recognition, vol. 121, p. 108224, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320321004052
[15] U. von Luxburg, “A tutorial on spectral clustering,” Statistics and Computing, vol. 17, no. 4, pp. 395–416, 2007. [Online]. Available: https://doi.org/10.1007/s11222-007-9033-2
[16] F. Luo, H. Huang, Z. Ma, and J. Liu, “Semi-supervised sparse manifold discriminative analysis for feature extraction of hyperspectral images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6197–6211, 2016.
[17] F. Luo, Z. Zou, J. Liu, and Z. Lin, “Dimensionality reduction and classification of hyperspectral image via multistructure unified discriminative embedding,” IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–16, 2022.
[18] X. Zhang, H. Liu, Q. Li, and X.-M. Wu, “Attributed graph clustering via adaptive graph convolution,” in Proceedings of the 28th International Joint Conference on Artificial Intelligence, ser. IJCAI’19. AAAI Press, 2019, p. 4327–4333.
[19] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, “Minimum spanning forests and the drop of water principle,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 8, pp. 1362–1374, 2009.
[20] V. Kovalevsky, “Finite topology as applied to image analysis,” Computer Vision, Graphics, and Image Processing, vol. 46, no. 2, pp. 141–161, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0734189X89901655
[21] L. Grady, “Random walks for image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pp. 1768–1783, 2006.
[22] H. W. Kuhn, “The hungarian method for the assignment problem,” Naval Research Logistics Quarterly, vol. 2, no. 1–2, p. 83–97, Mar 1955. [Online]. Available: http://dx.doi.org/10.1002/nav.3800020109

Fig. 4. GCN evaluation of ESW Edge-weights. We use the vanilla GCN [18] modified to incorporate edge-weights. We plot the best score till iteration k. The number of iterations is shown on the x-axis and the max score is shown on the y-axis. Three datasets were used - (a) Indianpines (b) Pavia Centre and (c) Salinas. For Pavia Centre we stop after iteration 30 since both methods have converged. Observe that the weighted GCN has significant improvement over unweighted GCN.

![Graph Image](image-url)