

## A PARAMETER SETTING

Here we give a table of parameters when GVCLN is tested without validation set. Since there is no validation set and the number of nodes in the training set is small, we cannot stop early by setting patience. Our approach is to fix the maximum number ( $\text{epoch}_{\max}$ ) of training epochs. The training epoch set in the Cora and Citeseer datasets are 300, and the training epoch set in the PubMed datasets is 400.  $\text{hidden}_1$  and  $\text{dropout}_1$  are the hidden layers and dropout rate of Viewer 1.  $\text{hidden}_2$  and  $\text{dropout}_2$  are the hidden layers and dropout rate of Viewer 2.  $t$  represents the number of pseudo-labels added to GVCLN after pre-training. When the label rate is low, the training set is insufficient, so more pseudo-labels need to be added for training. When the label rate gradually increases, the number of pseudo tags should be gradually reduced. This operation can prevent GVCLN from misleading by the false information in pseudo labels. The function of the parameter  $\beta$  is to weigh the label loss and consistency loss. When the label rate is low, since the training set is small, the supervised loss is small, and consistency loss dominates. When the label rate gradually increases, the role of supervision loss gradually becomes more prominent, so the weight of consistency loss needs to be reduced. The setting of the filter intensity parameter  $m$  is similar to the parameter  $t$ . The filter makes the graph features smooth. In the case of low label rate, higher intensity filtering is required, and in the case of high label rate, the filter intensity needs to be reduced.

Table 1: Hyperparameters for Tests without Validation of GVCLN

Dataset	Cora						Citeseer						PubMed	
Label Rate	0.5%	1%	2%	3%	4%	5%	0.5%	1%	2%	3%	4%	5%	0.03%	0.05%
$\text{epoch}_{\max}$	300	300	300	300	300	300	300	300	300	300	300	300	400	400
$\text{hidden}_1$	16	16	16	16	16	16	16	16	16	16	16	16	16	16
$\text{dropout}_1$	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
$\text{hidden}_2$	8	8	8	8	8	8	8	8	8	8	8	8	8	8
$\text{dropout}_2$	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
$t$	200	180	160	140	120	100	200	180	160	140	120	100	1000	1000
$\beta$	0.8	0.8	0.6	0.6	0.1	0.1	0.8	0.8	0.8	0.1	0.08	0.05	0.8	0.6
$m$	15	12	12	10	10	5	15	12	12	5	5	5	15	15

## B PSEUDO LABEL SELECTION

We set a different number of pseudo-labels for each label rate, and set it to  $t$ . Let  $V_L$  denote the set of nodes with labels, and let  $V_U$  denote the set of nodes without labels, then the total set of nodes is  $V = V_L \cup V_U$ . According to the final output of the network, the predicted label confidence of the  $i$ -th node can be obtained as:

$$c_i = \max([p_{i1}, p_{i2}, \dots, p_{ik}]), \quad (1)$$

$$y'_i = \text{argmax}([p_{i1}, p_{i2}, \dots, p_{ik}]), \quad (2)$$

where  $[p_{i1}, p_{i2}, \dots, p_{ik}]$  and  $y'_i$  is prediction probability vector and predict label of the  $i$ -th node, and  $\mathbf{c} = [c_1, c_2, \dots, c_n]^\top$  is the predicted label probability of all nodes. Next, we sort the predicted probability values from large to small, and select  $t$  nodes with the highest confidence (that is, the highest predicted probability value) for each category as pseudo labels. The returned index value after sorting from largest to smallest is:

$$\mathbf{q} = \text{arg sort}(\mathbf{c}) = \text{arg sort} \left( \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \right) \quad (3)$$

where  $\mathbf{q}$  returns an index vector of the ordered confidence. Finally, we use  $\mathbf{q}_1$  and  $\mathbf{q}_2$  to obtain the common top  $t$  pseudo labels of the view 1 and the view 2.