CONVERT: Contrastive Graph Clustering with Reliable Augmentation

Xihong Yang  
xihong_edu@163.com  
National University of Defense Technology  
Changsha, Hunan, China

Cheng Tan  
Westlake University  
Hangzhou, Zhejiang, China

Yue Liu  
Ke Liang  
National University of Defense Technology  
Changsha, Hunan, China

Siwei Wang  
Sihang Zhou  
National University of Defense Technology  
Changsha, Hunan, China

Jun Xia  
Stan Z.Li  
Westlake University  
Hangzhou, Zhejiang, China

Xinwang Liu∗  
En Zhu∗  
National University of Defense Technology  
Changsha, Hunan, China

ABSTRACT

Contrastive graph node clustering via learnable data augmentation is a hot research spot in the field of unsupervised graph learning. The existing methods learn the sampling distribution of a pre-defined augmentation to generate data-driven augmentations automatically. Although promising clustering performance has been achieved, we observe that these strategies still rely on pre-defined augmentations, the semantics of the augmented graph can easily drift. The reliability of the augmented view semantics for contrastive learning cannot be guaranteed, thus limiting the model performance. To address these problems, we propose a novel CONtrastive Graph ClusterEring network with Reliable AugmenTation (CONVERT). Specifically, in our method, the data augmentations are processed by the proposed reversible perturb-recover network. It distills reliable semantic information by recovering the perturbed latent embeddings. Moreover, to further guarantee the reliability of semantics, a novel semantic loss is presented to constrain the network via quantifying the perturbation and recovery. Lastly, a label-matching mechanism is designed to guide the model by clustering information through aligning the semantic labels and the selected high-confidence clustering pseudo labels. Extensive experimental results on seven datasets demonstrate the effectiveness of the proposed method. We release the code and appendix of CONVERT at https://github.com/xihongyang1999/CONVERT on GitHub.

KEYWORDS

Attribute Graph Clustering, Contrastive Learning

ACM Reference Format:

1 INTRODUCTION

Thanks to the strong representation capacity, graph learning algorithms have attracted more attention in many fields of multimedia, including the recommendation system [3–5, 12, 38, 40], knowledge graph [16–18], temporal graph [19–21], molecular graph [52–55] and so on. Among those directions, graph contrastive clustering [9, 11, 22–24, 26, 48, 56] has become a hot research spot, which encodes the nodes with graph neural networks into the embeddings and divides them into disjoint clusters in the unsupervised scenario. In general, prevailing graph contrastive clustering algorithms initially create augmented graph views through perturbations in node attributes or edges. Subsequently, these methods aim to bring identical samples in different views closer together while simultaneously distancating distinct samples from each other. Graph data augmentations are adopted as a crucial technique to construct contrastive views. More recently, learnable graph data augmentation has gained significant attention.

Specifically, through a Bayesian manner, JOAO [50] proposed an augmentation strategy to automatically select augmentations among many pre-defined candidates for graph classification. Although verified effective, the augmentations selected still depend on the pre-defined schemes and are not learnable. To further alleviate this problem, AD-GCL [31] designed a learnable edge augmentation by Bernoulli distribution. NCLA [30] proposed a learnable topology augmentation method with the multi-head graph attention mechanism. However, the learnable strategy for node level is neglected. Moreover, by acquiring knowledge of a probability distribution, AutoGCL [49] introduced an auto-augmentation approach that involves masking or dropping nodes. The model performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3611809
is guaranteed by the proposed learnable augmentation strategy. However, previous methods still rely on pre-defined augmentations. The semantics of the augmented view easily drift. As shown in Fig. 1 (a), we observe that when dropping nodes in the molecule graph, the semantic information will dramatically change. Similarly, Fig. 1 (b) demonstrates that the relationship will be represented incorrectly by dropping the connection in the social network. The semantic reliability of the constructed views can not be guaranteed, thus limiting model clustering performance.

To address these issues, we propose a novel CONTRastive Augment-free Graph Clustering (CONVERT). In our method, a reversible perturb-recover network is designed to generate augmented views by neural network optimization. Concretely, the features of the augmented view are extracted by the perturb and recover operations in the latent space, thus improving the reliability of the semantics. In addition, we propose a novel semantic loss to further guarantee the reliability of semantics by quantifying the perturbation and the recovery of the reversible network. Moreover, the neural networks are guided with clustering information by aligning the selected high-confidence clustering pseudo labels and the semantic labels.

In those manners, we guarantee the reliability of the embedding semantics in dual aspects. Firstly, for the network aspect, we design a reversible perturb-recover network. The recover network restores the embeddings generated by the perturbed network. Thus the semantics of the original view and the augmented view are more similar. Secondly, for the optimization aspect, we design a semantic loss to further guarantee reliability by pushing close the similarity matrix of the embeddings. We summarize the key contributions of this paper as follows:

- We propose a contrastive graph clustering method with reliable augmentation, termed CONVERT, by designing a reversible perturb-recover network to generate the augmented view with reliable semantics.
- To further guarantee the reliability of the semantic, we design a semantic loss by quantifying the perturbation and recovery.
- In order to guide the model with clustering information, a label-matching mechanism is proposed to align the selected high-confidence pseudo labels and semantic labels.
- Extensive experimental results on seven datasets have demonstrated that CONVERT outperforms the existing state-of-the-art deep graph clustering algorithms. Moreover, the effectiveness of the proposed modules is verified by ablation studies.

2 RELATED WORK

2.1 Deep Graph Clustering

Graph learning methods [39, 46] have attracted great attention recently. Deep graph node clustering is an important unsupervised downstream task. It aims to learn the graph underlying semantic information and divide the nodes into different clusters. The existing clustering algorithms can be classified into three categories, i.e., generative algorithms [2, 8, 33, 35], adversarial algorithms [29, 32], and contrastive algorithms [6, 11, 23, 25, 56]. Contrastive learning algorithms have obtained great success in the field of graph [43, 45, 57]. In this paper, we mainly focus on contrastive clustering algorithms. Data augmentation plays a crucial role in contrastive clustering algorithms. To be specific, MVGRL [11], GDCL [56], and DCRN [23] utilize the graph diffusion matrix as the augmented view. Different from the above algorithms, SCAGC [44] conducts random edge perturbation to construct the augmented view. Regarding feature operation, both DCRN and SCAGC execute augmentations on node attributes through attribute corruption. Despite their proven effectiveness, the strong clustering performance of these techniques is intricately tied to the judicious selection of data augmentations. In recent developments, certain graph augmentation methodologies [15] emphasize that specific data augmentations may trigger semantic drift. To alleviate this problem, in our paper, we design a reversible network to generate the augmented view in a perturb-recover way. The semantics of embeddings are guaranteed by the reversible network, thus avoiding semantic drift.

2.2 Graph Data Augmentation

Graph data augmentation [41, 42, 51] has emerged as a dominant technique in graph contrastive learning. The existing data augmentation methods could roughly be divided into three classes. 1) Augmentation-free methods. AFGRL [15] devises the augmented view by identifying nodes equipped with local and global data, eschewing the need for augmentation. However, this approach doesn’t assure the augmented view’s reliability, potentially resulting in subpar performance. 2) Adaptive augmentation techniques. In the realm of graph classification, JOAO [50] harnesses the potential to autonomously choose data augmentation through learning the sampling distribution of predefined augmentations. Furthermore, GCA [57] enhances augmentation adaptability by integrating diverse
Figure 2: Illustration of CONVERT. In our method, we design a reversible perturb-recover network to generate the augmented view. The semantic information is guaranteed by the perturbation and recovery operation in the latent space. Besides, we design a semantic loss to further improve the reliability of the augmented views. Detailed descriptions are shown in Fig.3. Lastly, we design a label-matching mechanism to utilize the clustering information. The selected high-confidence clustering pseudo labels align the semantic labels $p_{se}$ with the label matching loss $L_M$, thus guiding the model to have better performance.

3 METHOD
In this section, we introduce a novel Contrastive Graph Clustering method with Reliable Augmentation (CONVERT). The comprehensive CONVERT framework is illustrated in Figure 2. Primarily, CONVERT encompasses two key modules: the Learnable Augmentation Module with Reliable Augmentation and the Label-Matching Module. Detailed definitions for these modules will be presented in subsequent sections.

3.1 Notations
For an undirected graph $G = \{X, A\}$, consider $V = \{v_1, v_2, \ldots, v_N\}$ as a set of $N$ nodes categorized into $K$ classes. Here, $X \in \mathbb{R}^{N \times D}$ denotes the attribute matrix, while $A \in \mathbb{R}^{N \times N}$ stands for the attribute matrix and the original adjacency matrix. The degree matrix is symbolized as $D = \text{diag}(d_1, d_2, \ldots, d_N) \in \mathbb{R}^{N \times N}$, where $d_i = \sum_{(v_i, v_j) \in E} a_{ij}$. The graph Laplacian matrix is defined as $L = D - A$. Utilizing the renormalization technique $\tilde{A} = A + I$ from GCN [14], the symmetric normalized graph Laplacian matrix is represented as $\tilde{L} = \mathbf{I} - \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$. A summary of essential notations is provided in Table 1.

3.2 Reliable Augmentation Module
In this subsection, we design a reversible network to obtain the augmented views with reliable semantics. Specifically, we propose a perturb network and a recover network.

To avoid the entanglement of graph convolutional filters and weight matrices, following AGE [6], we firstly adopt the Laplacian filter at the embedding level. Through a reversible network, we generate the augmented view, allowing the network to optimize the view’s quality during training.

$$\hat{X}_1 = (I - \tilde{L})^t X,$$

where $\tilde{L}$ represents the symmetric normalized graph Laplacian matrix, while $t$ denotes the layer number of the Laplacian filter. Subsequently, we leverage the perturbation network $p(\cdot; \theta)$ to apply priors targeting the graph’s topological and semantic attributes. However, the augmentation is still not optimized by the neural network in the adaptive augmentation methods. 3) Learnable data augmentation. An edge-level learnable strategy is designed in AD-GCL [31] while neglecting the augmentations on the node level. Similarly, NCLA [30] proposed a learnable topology by the multi-head graph attention mechanism. However, those two methods neglected the learnable strategy for the node level. Subsequently, AutoGCL [49] introduced a probability-driven learnable strategy. Despite yielding improved the model performance, AutoGCL remains reliant on existing and predefined data augmentations. In contrast, our work introduces a learnable augmentation approach at the embedding level. Through a reversible network, we generate the augmented view, allowing the network to optimize the view’s quality during training.
perturbations to the attribute matrix, leading to the formation of the perturbed attribute matrix $\tilde{X}_2$, formulated as follows:

$$\tilde{X}_2 = p(\tilde{X}_1; \theta).$$  

(2)

After that, we extract the original embeddings $E^{v1}$ and the perturbed embeddings $E^{v2}$ for the $\tilde{X}_1$ and $\tilde{X}_2$ with the encoder network:

$$E^{v1} = \text{Encoder}(\tilde{X}_1); E^{v2}_i = \frac{E^{v1}_i}{\|E^{v1}_i\|_2}, i = 1, 2, \ldots, N$$  

(3)

$$E^{v2} = \text{Encoder}(\tilde{X}_2); E^{v2}_j = \frac{E^{v2}_j}{\|E^{v2}_j\|_2}, j = 1, 2, \ldots, N.$$  

We adopt the multi-layers perception (MLPs) as the encoder network. To obtain the reliable semantic augmented views, we design a recover network $r(\cdot; \theta)$ to recover the perturbed embeddings as follows:

$$H^{r1} = r(E^{v1}; \theta),$$  

(4)

where $H^{r1}$ denotes the recovered embeddings. Here, we adopt the two layers of multi-layers perception as the perturb network and the recover network, respectively. In this manner, the perturbed embeddings could be recovered by $r(\cdot; \theta)$. In embedding space, the recover network restores the perturbed semantics. Therefore, the recovered embeddings have more similar semantics to the original embeddings. In addition, we implement the perturb to the original embeddings $E^{v1}$ by the perturb network $p(\cdot; \theta)$ as formulated:

$$H^{r2} = p(E^{v1}; \theta),$$  

(5)

where $H^{r2}$ is the perturbed embeddings. Since the smoothed attributed matrix $\tilde{X}$ has been perturbed, we conduct the same operation for the original embeddings to obtain the perturbed embeddings. In this way, the semantics of the perturbed embeddings are similar to the same perturbed operation in latent space.

Aiming to further guarantee reliable semantics of the augmented view, as shown in Fig. 3, we calculate the similarity matrix $S$ between $i$-th sample in the first view and $j$-th sample in the second view as follows:

$$S(E^{v1}, H^{r1}) \xleftarrow{L_S} S(E^{v1}, H^{r2})$$  

Figure 3: Illustration of the semantic loss $L_S$. Through pulling close $S(E^{v1}, H^{r1})$ and $S(E^{v1}, H^{r2})$, the perturbation and recovery could be quantified, thus guaranteeing the reliability.

**Algorithm 1 CONVERT**

**Input:** The input graph $G = \{X, A\}$; The iteration number $I$; High-confidence epoch $N$

**Output:** The clustering result $R$.

1. for $i = 1$ to $I$
2. Acquire the smoothed attribute $\tilde{X}_1$ and perturbed attribute matrix $\tilde{X}_2$.
3. Encoder the attribute to obtain the embeddings $E^{v1}$ and $E^{v2}$.
4. Obtain the perturbed embedding $H^{r1}$ and the recovered embedding $H^{r2}$ with Eq. (4) and (5).
5. Fuse $E^{v1}$ and $E^{v2}$ to yield $E$, followed by K-means for clustering results.
6. Calculate the similarity matrix using Eq. (6).
7. Obtain the semantic labels $p^{*}$ and high-confidence pseudo labels $h$ via Eq. (10) and (9).
8. if $i > N$ then
9. Calculate the label-matching loss $L_{MI}$ via Eq. (11)
10. end if
11. Calculate the semantic loss $L_S$ with Eq. (7).
12. Calculate the contrastive loss $L_{C}$ with Eq. (12).
13. Update the entire network by minimizing $L$ in Eq. (13).
14. end for
15. Perform K-means on $E$ to obtain the final clustering result $R$.
16. return $R$

$$S(E^{v1}, H^{r1}) = E^{v1}_i \odot H^{r1}_{rj},$$  

$$S(E^{v1}, H^{r2}) = E^{v1}_i \odot H^{r2}_{pj},$$  

(6)

where $\odot$ denotes the Hadamard product. Besides, $S(E^{v1}, H^{r1})$ represents the similarity matrix between the original embeddings and the recovered embeddings. $S(E^{v1}, H^{r2})$ denotes the similarity matrix of the original embeddings and the perturbed embeddings. $S$ can better reveal the changing of semantics by quantifying perturbation and recovery. Subsequently, we enhance the reliability of embedding semantics by forcing the similarity matrix to pull close:

$$L_S = \|S(E^{v1}, H^{r1}) - S(E^{v1}, H^{r2})\|^2_F.$$  

(7)

Through Eq. (7), we constrain the network to restore the perturbed embeddings. In this manner, the quantification of perturb and restore is mutually reinforcing, thus generating the augmented view with more reliable semantics.

Different from previous augmentation methods, our method can construct more reliable semantic augmented views. The reasons are as follows. On the one hand, instead of using normal graph augmentations, e.g., feature masking or edge perturbing, the augmented views are generated by the reversible network. The process of the augmented view is learnable and can be optimized by the network. On the other hand, the semantics of the augmented view is guaranteed by the perturb-recover procedure. The perturbed embeddings are restored through the recover network.

### 3.3 Label-Matching Module

After encoding, we firstly fuse the embeddings to obtain consensus embeddings $E$ as below:
\[ E = \frac{1}{2}(E^p + E^n). \]  

Then we implement K-means [10] to obtain clustering results. To utilize more reliable clustering information, we obtain the high-confidence pseudo labels \( h \) through selecting clustering pseudo labels \( p \). Formally,

\[ h = \text{top}(p), \]  

where \( \text{top}(\cdot) \) is the confidence measure function to select the top \( \tau \) clustering pseudo labels.

In the unsupervised clustering scene, the supervision information could be extracted by the high-confidence pseudo labels. We design a matching mechanism between the semantic labels and the high-confidence pseudo labels to further guide the network. Specifically, we conduct the Softmax for the embeddings to obtain the semantic labels \( p^{se} \) as follows:

\[ p^{se}_i = \text{Softmax}(E^{si}), i \in 1, 2. \]  

Subsequently, we match the semantic labels \( p^{se} \) and the high-confidence pseudo labels \( h \) by:

\[ L_M = CE(p^{se}_i, h), i \in 1, 2, \]  

where \( CE(\cdot) \) is the Cross-Entropy loss [7]. Through the matching mechanism, the quality of the semantics can be further improved with the high-confidence pseudo labels. Furthermore, in order to enhance the precision of clustering pseudo labels, we employ a two-stage training strategy aimed at bolstering the network’s discriminative capability. Specifically, during the second stage, we opt for high-confidence pseudo labels to enact the matching mechanism.

### 3.4 Loss Function

The proposed CONVERT jointly optimizes three objectives, including the semantic loss \( L_S \), the label matching loss \( L_M \), and the contrastive loss \( L_C \). More precisely, we employ \( L_C \) to amplify the similarity among positive samples while diminishing it among negative samples. The contrastive loss for the \( i \)-th sample in the \( j \)-th view can be articulated as follows:

\[ \ell(E_i, E_j) = -\log \frac{e^{\text{sim}(E_i, E_j)}}{e^{\text{sim}(E_i, E_j)} + \sum_{k \neq i} e^{\text{sim}(E_k, E_j)}}, \]  

\[ L_C = \frac{1}{2N} \sum_{i=1}^{N} \left[ \ell(E_i, E_j), \ell(E_j, E_i) \right] \]  

where \( \text{sim}(\cdot) \) represents the function to calculate the similarity, i.e., cosine similarity. Here, we calculate the contrastive loss for the view pairs \( (E^p_i, H^{se}_i) \) and \( (E^n_i, H^{se}_i) \). In summary, the objective of CONVERT is formulated as:

\[ L = L_C + \alpha L_S + \beta L_M. \]  

where \( \alpha \) and \( \beta \) are the trade-off hyper-parameters. The detailed learning procedure of CONVERT is illustrated in Algorithm 1.

### 4 EXPERIMENT

In this section, we conduct the experiments to verify the effectiveness of the proposed CONVERT through answering the following questions:

- **RQ1**: How effective is CONVERT for graph node clustering?
- **RQ2**: How does the proposed module influence the performance of CONVERT?
- **RQ3**: How about the efficient about CONVERT?
- **RQ4**: How do the hyper-parameters impact the performance of CONVERT?
- **RQ5**: What is the clustering structure revealed by CONVERT?

#### 4.1 Datasets & Metric

**Benchmark Datasets** We conduct extensive experiments to verify the effectiveness of CONVERT on seven benchmark datasets, including CORA \(^1\), CITESEER \(^2\), BAT [48], EAT [47], UAT [48], AMAP [28], and CORAFULL [25]. Detailed dataset statistics are summarized in Table 2.

**Evaluation Metrics** The evaluation of clustering performance encompasses four extensively utilized metrics: Accuracy (ACC), Normalized Mutual Information (NMI), Average Rand Index (ARI), and macro F1-score (F1) [36, 37].

#### 4.2 Experimental Setup

The experimental setup comprises a desktop computer equipped with an Intel Core i7-7820x CPU, an NVIDIA GeForce RTX 2080Ti GPU, 64GB RAM, and the PyTorch deep learning platform. To minimize the influence of randomness, each method is executed ten times, and the results are reported in terms of mean values with corresponding standard deviations. Training all methods is continued for 400 epochs until convergence. We employ the Adam optimizer to minimize the total loss and subsequently perform K-means on the acquired embeddings. A two-stage training strategy is adopted to ensure dependable clustering pseudo labels. In the first stage, the model’s discriminative capacity is improved, while the second stage involves the utilization of high-confidence clustering pseudo labels within the matching module. Comprehensive parameter settings are elaborated in Table 1 of the Appendix.

---

\(^1\)https://relational.fit.cvut.cz/dataset/CORA

\(^2\)http://citeseerx.ist.psu.edu/index
4.3 Performance Comparison (RQ1)

In this subsection, we conduct experiments to demonstrate the superiority of CONVERT with 12 baselines on seven datasets. To be specific, the compared clustering algorithms could roughly be divided into three classes, i.e., classical graph clustering algorithms (DAEGC [35], ARGA [29], SDCN [2]), contrastive graph clustering algorithms (AGE [6], MVGRL [11], AGC-DRR [9]), and graph augmentation clustering algorithms (GCA [57], AFGR [15], AutoSSL [13], SUBLIME [27], NCLA [30]).

Table 3 presents the attribute clustering performance comparison. From those results, we observe that 1) compared with classical deep graph clustering algorithms, our method achieves state-of-the-art performance. We conjecture the reason is that those methods rarely consider the topological information; 2) Thanks to the learnable augmentation with reliable semantic strategy, our method obtains better performance with contrastive deep graph clustering methods; 3) The graph augmentation methods achieve unsupervised performance. We analyze the reason is that the semantic information of those methods drafts after the augmentations. Overall, our proposed CONVERT achieves better performance on most metrics on seven datasets. Using the EAT dataset as an example, CONVERT outperforms the second-best method by margins of 6.22%, 5.00%, 5.63%, and 5.67% in ACC, NMI, ARI, and F1 scores respectively. Furthermore, due to space constraints, more comparison results of eight baseline methods are available in Table 2 of the Appendix. These supplementary results reaffirm the efficacy of our proposed CONVERT.
Table 4: Ablation studies of CONVERT are conducted across six datasets. The notations (w/o) L_M, (w/o) R_S, and (w/o) R_N denote reduced models obtained by excluding the label-matching module, reliable semantic loss, and the reversible network, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>(wo) L_M</th>
<th>(wo) R_S</th>
<th>(wo) R_N</th>
<th>Feature Mask</th>
<th>Edge Remove</th>
<th>Edge Add</th>
<th>Diffusion</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORA</td>
<td>ACC</td>
<td>73.42±1.39</td>
<td>73.13±1.42</td>
<td>73.36±1.19</td>
<td>69.56±1.91</td>
<td>65.76±3.43</td>
<td>64.77±1.95</td>
<td>70.68±1.48</td>
<td>74.07±1.51</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>55.69±0.87</td>
<td>55.48±0.84</td>
<td>55.93±0.77</td>
<td>52.28±2.50</td>
<td>50.92±1.75</td>
<td>49.53±2.22</td>
<td>53.15±0.84</td>
<td>55.57±1.12</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>50.35±1.38</td>
<td>49.71±1.41</td>
<td>50.24±1.22</td>
<td>45.10±2.50</td>
<td>41.46±2.21</td>
<td>40.76±1.27</td>
<td>48.39±1.34</td>
<td>50.58±2.01</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>72.62±1.93</td>
<td>72.58±1.55</td>
<td>72.73±1.30</td>
<td>68.72±2.78</td>
<td>63.94±3.30</td>
<td>62.20±2.55</td>
<td>68.66±1.67</td>
<td>72.92±3.27</td>
</tr>
<tr>
<td>AMAP</td>
<td>ACC</td>
<td>75.96±0.58</td>
<td>74.93±1.03</td>
<td>74.95±0.77</td>
<td>67.55±1.13</td>
<td>72.52±0.62</td>
<td>68.21±1.81</td>
<td>63.49±2.17</td>
<td>77.19±0.55</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>65.80±1.25</td>
<td>63.76±1.76</td>
<td>65.53±1.23</td>
<td>55.66±1.32</td>
<td>59.72±0.96</td>
<td>55.25±1.97</td>
<td>51.88±1.99</td>
<td>67.20±1.07</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>57.28±1.98</td>
<td>55.55±1.99</td>
<td>56.38±2.07</td>
<td>45.48±1.33</td>
<td>51.58±1.35</td>
<td>46.12±1.95</td>
<td>41.45±2.71</td>
<td>60.79±1.83</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>72.33±2.26</td>
<td>71.80±1.59</td>
<td>70.53±1.82</td>
<td>64.39±1.57</td>
<td>67.67±1.88</td>
<td>64.78±2.70</td>
<td>61.07±2.50</td>
<td>74.03±1.00</td>
</tr>
<tr>
<td>BAT</td>
<td>ACC</td>
<td>76.11±1.96</td>
<td>69.69±3.17</td>
<td>68.85±2.97</td>
<td>61.60±2.32</td>
<td>49.47±2.48</td>
<td>64.96±2.74</td>
<td>64.50±3.28</td>
<td>78.02±1.36</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>52.04±1.86</td>
<td>45.55±2.73</td>
<td>43.96±2.82</td>
<td>36.11±2.06</td>
<td>19.44±3.09</td>
<td>39.27±3.18</td>
<td>39.84±3.61</td>
<td>53.54±1.71</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>49.70±2.89</td>
<td>41.38±2.78</td>
<td>39.53±3.12</td>
<td>31.82±2.92</td>
<td>12.72±3.39</td>
<td>33.86±3.27</td>
<td>34.92±4.16</td>
<td>51.95±2.18</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>75.61±2.11</td>
<td>69.10±3.48</td>
<td>67.87±3.54</td>
<td>60.33±2.40</td>
<td>48.23±2.12</td>
<td>64.08±3.40</td>
<td>63.56±3.99</td>
<td>77.77±1.48</td>
</tr>
<tr>
<td>EAT</td>
<td>ACC</td>
<td>54.59±0.46</td>
<td>53.93±0.58</td>
<td>53.83±0.88</td>
<td>50.25±0.93</td>
<td>40.38±1.56</td>
<td>52.31±0.92</td>
<td>54.99±0.63</td>
<td>58.35±0.18</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>28.36±1.16</td>
<td>27.49±0.70</td>
<td>27.55±0.74</td>
<td>22.24±0.99</td>
<td>12.59±1.92</td>
<td>26.29±1.30</td>
<td>27.47±0.50</td>
<td>33.36±0.16</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>22.58±0.51</td>
<td>21.77±1.04</td>
<td>22.07±1.54</td>
<td>16.59±1.78</td>
<td>17.81±1.69</td>
<td>19.29±1.05</td>
<td>24.84±0.36</td>
<td>27.11±0.19</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>55.05±0.53</td>
<td>54.43±0.66</td>
<td>53.71±1.53</td>
<td>50.80±1.40</td>
<td>39.29±1.76</td>
<td>53.10±1.01</td>
<td>53.27±1.01</td>
<td>58.42±0.22</td>
</tr>
<tr>
<td>UAT</td>
<td>ACC</td>
<td>49.99±1.32</td>
<td>49.49±1.34</td>
<td>48.39±0.57</td>
<td>47.76±2.79</td>
<td>49.80±1.02</td>
<td>50.00±1.13</td>
<td>52.67±1.84</td>
<td>57.36±0.55</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>20.09±1.73</td>
<td>20.99±1.83</td>
<td>19.93±2.39</td>
<td>21.61±2.27</td>
<td>18.81±1.03</td>
<td>24.18±1.39</td>
<td>24.29±1.65</td>
<td>28.75±1.13</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>20.02±2.26</td>
<td>20.10±2.31</td>
<td>18.55±1.65</td>
<td>15.84±1.83</td>
<td>16.06±1.22</td>
<td>14.39±1.88</td>
<td>23.33±2.23</td>
<td>27.96±0.79</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>46.52±1.42</td>
<td>45.68±1.12</td>
<td>45.84±1.42</td>
<td>46.72±3.52</td>
<td>47.20±1.84</td>
<td>50.54±2.61</td>
<td>48.08±2.84</td>
<td>54.55±1.49</td>
</tr>
<tr>
<td>CITESEER</td>
<td>ACC</td>
<td>64.99±1.62</td>
<td>63.00±1.62</td>
<td>64.43±1.55</td>
<td>63.62±1.10</td>
<td>66.00±1.47</td>
<td>64.16±1.06</td>
<td>65.74±0.56</td>
<td>68.43±0.69</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>38.14±1.72</td>
<td>38.15±1.71</td>
<td>37.68±1.44</td>
<td>30.13±1.17</td>
<td>39.16±1.44</td>
<td>39.35±1.13</td>
<td>40.98±0.57</td>
<td>41.62±0.73</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>39.12±2.32</td>
<td>39.14±2.32</td>
<td>38.20±1.78</td>
<td>37.09±1.73</td>
<td>38.66±2.24</td>
<td>37.78±1.43</td>
<td>39.66±0.91</td>
<td>42.77±1.63</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>61.22±1.38</td>
<td>61.23±1.38</td>
<td>60.82±1.62</td>
<td>60.36±0.85</td>
<td>58.50±1.24</td>
<td>60.39±1.01</td>
<td>62.00±0.81</td>
<td>62.39±2.15</td>
</tr>
</tbody>
</table>

4.4 Ablation Studies (RQ2)

4.4.1 Effectiveness of the proposed module. As shown in Table 4, we conduct experiments to verify the effectiveness of the reversible network. To be specific, we adopt "(w/o) L_M", "(w/o) R_S", and "(w/o) R_N" to denote the reduced models by removing the label matching module, the reliable semantic loss, the reversible network respectively. Without any of our proposed modules, the clustering performance will decrease, indicating that each module makes contributions to boosting the performance. We further analyze the reasons as follows: 1) the network is better guided by the high-confidence clustering pseudo labels through the label-matching mechanism. 2) The proposed semantic loss and the reversible network guarantee the reliability of semantic information, thus avoiding semantic drifting.

4.4.2 Effectiveness of the Learnable Augmentation Module. To validate the effectiveness of the learnable augmentation module, we conduct the ablation studies shown in Table 4. Here, we adopt the same backbone for all experiments and four commonly used graph augmentations, e.g., randomly masking 10% feature (Feature Mask),...
Table 5: Training Time Comparison on six datasets. Avg. denotes the average time cost on six datasets. Besides, OOM represents Out-Of-Memory during the training process.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DEC</th>
<th>AE</th>
<th>DAEGC</th>
<th>MGAE</th>
<th>AGE</th>
<th>SDCN</th>
<th>MVGRL</th>
<th>MCCG</th>
<th>SCAGC</th>
<th>CONVERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORA</td>
<td>91.13</td>
<td>47.31</td>
<td>12.97</td>
<td>7.38</td>
<td>46.63</td>
<td>11.32</td>
<td>14.72</td>
<td>118.07</td>
<td>54.08</td>
<td>16.89</td>
</tr>
<tr>
<td>CITESEER</td>
<td>223.95</td>
<td>74.69</td>
<td>14.70</td>
<td>6.69</td>
<td>70.63</td>
<td>11.90</td>
<td>18.31</td>
<td>126.06</td>
<td>58.00</td>
<td>57.23</td>
</tr>
<tr>
<td>BAT</td>
<td>21.37</td>
<td>7.46</td>
<td>4.79</td>
<td>3.83</td>
<td>2.49</td>
<td>11.50</td>
<td>3.19</td>
<td>2.28</td>
<td>93.79</td>
<td></td>
</tr>
<tr>
<td>EAT</td>
<td>26.99</td>
<td>9.56</td>
<td>5.14</td>
<td>4.64</td>
<td>3.86</td>
<td>12.12</td>
<td>3.32</td>
<td>2.87</td>
<td>47.79</td>
<td></td>
</tr>
<tr>
<td>UAT</td>
<td>42.30</td>
<td>29.57</td>
<td>6.44</td>
<td>4.75</td>
<td>8.95</td>
<td>10.64</td>
<td>4.27</td>
<td>23.10</td>
<td>64.70</td>
<td></td>
</tr>
<tr>
<td>AMAP</td>
<td>264.20</td>
<td>94.48</td>
<td>39.62</td>
<td>18.64</td>
<td>377.49</td>
<td>19.28</td>
<td>131.38</td>
<td>OOM</td>
<td>150.54</td>
<td>46.98</td>
</tr>
<tr>
<td>Avg.</td>
<td>111.66</td>
<td>43.85</td>
<td>13.94</td>
<td>7.66</td>
<td>85.81</td>
<td>12.64</td>
<td>29.20</td>
<td>-</td>
<td>76.82</td>
<td>22.95</td>
</tr>
</tbody>
</table>

Figure 7: Illustration of the gpu memory cost of CONVERT with five algorithms on six datasets.

randomly dropping 10% graph edges (Edge Remove), randomly adding 10% graph edges (Edge Add), and graph diffusion with 0.10 teleportation rate (Diffusion). From the results, we conclude that since utilizing the common graph augmentations, the clustering performance is limited by the drifting semantic [15]. In summary, the experiment results demonstrate the effectiveness of the learnable augmentation module.

4.5 Efficiency analysis (RQ3)

In this subsection, we compare the efficiency of CONVERT and other state-of-the-art clustering algorithms. As can be observed in Fig.7 and Table.5, CONVERT has a comparable GPU memory cost and training time with other clustering methods. In summary, the efficiency of CONVERT is acceptable. We analyze the reason that we adopt the graph filter to extract the feature, avoiding the complex convolution and aggregation operations.

4.6 Hyper-parameter Analysis (RQ4)

4.6.1 Sensitivity Analysis of hyper-parameter $\alpha$. As can be observed in Fig.5, we observe that the performance of CONVERT will not fluctuate greatly when the $\alpha \in [0.3, 0.7]$. When the value of $\alpha$ drastically changes, the balance of the model will be destroyed, thus limiting the clustering performance. Moreover, we investigate the influence of $\beta$. CONVERT is not sensitive to $\beta$. Experiments can be found in Fig. 1 in Appendix.

4.6.2 Sensitivity Analysis of hyper-parameter $\tau$. Besides, we investigate the impact of $\tau$. As shown in Fig.6, we observe that CONVERT achieve better performance when $\tau \in [0.55, 0.75]$. There are two reasons. Firstly, the discriminative capacity of CONVERT is limited when $\tau < 0.55$ since the low confidence of the pseudo labels. Secondly, when $\tau > 0.75$, the model easily leads to confirmation bias due to the over-confidence pseudo labels [1].

4.7 Visualization Analysis (RQ5)

To unveil the inherent clustering structure, this subsection employs visualization to depict the distribution of the learned embeddings. Specifically, experiments are conducted using the t-SNE algorithm [34] on CORA and AMAP datasets. As depicted in Figure 4, the visual results highlight that CONVERT exhibits an enhanced clustering structure.

5 CONCLUSION

In this paper, we present a learnable augmentation strategy for attribute clustering with reliable semantics, termed CONVERT. Specifically, we design a reversible network to generate augmented views. The perturb-recover embedding operation avoids semantic drift. Then, we design a semantic loss to further guarantee the reliability of the semantic. Moreover, we propose a label-matching mechanism to align the semantic labels and high-confidence pseudo labels, thus utilizing the clustering information to guide the model. Extensive experiments have demonstrated the effectiveness of CONVERT.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China 2020AAA0107100 and the National Natural Science Foundation of China (project no. 62325604, 62276271).

REFERENCES
