

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 can not be guaranteed, thus limiting the model performance. To address these problems, we propose a novel **CON**trastiVe Graph ClustEring network with **Reliable AugmenTation (COVERT)**. 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.

CCS CONCEPTS

• **Theory of computation** → **Unsupervised learning and clustering**; • **Computing methodologies** → **Cluster analysis**.

*Corresponding author

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 are not made or distributed for profit or commercial advantage and 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](https://www.acm.org/permissions).

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0108-5/23/10...\$15.00

<https://doi.org/10.1145/3581783.3611809>

KEYWORDS

Attribute Graph Clustering, Contrastive Learning

ACM Reference Format:

Xihong Yang, Cheng Tan, Yue Liu, Ke Liang, Siwei Wang, Sihang Zhou, Jun Xia, Stan Z.Li, Xinwang Liu, and En Zhu. 2023. CONVERT: Contrastive Graph Clustering with Reliable Augmentation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*, October 29–November 3, 2023, Ottawa, ON, Canada. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3581783.3611809>

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 distancing 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

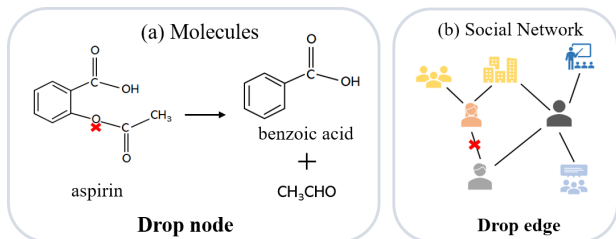


Figure 1: Illustration of the semantic drift.

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 **CON**trasti**VE** Graph Clust**ER**ing network with **Rel**iable Augmen**T**ation (**COVERT**). 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.

Table 1: Notation summary.

Notations	Meaning
$X \in \mathbb{R}^{N \times D}$	The Attribute Matrix
$\tilde{X}_1 \in \mathbb{R}^{N \times D}$	The Perturbed Attribute Matrix
$A \in \mathbb{R}^{N \times N}$	The Adjacency Matrix
$D \in \mathbb{R}^{N \times N}$	The Degree Matrix
$E \in \mathbb{R}^{N \times D}$	The Node Embeddings
$H_p^{o_2} \in \mathbb{R}^{N \times D}$	The Perturbed Embeddings
$H_r^{o_1} \in \mathbb{R}^{N \times D}$	The Recovered Embeddings
$S \in \mathbb{R}^{N \times N}$	The Similarity Matrix
h	High-confidence Clustering Pseudo Label
p^{se}	The Semantic Label

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) Augment-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

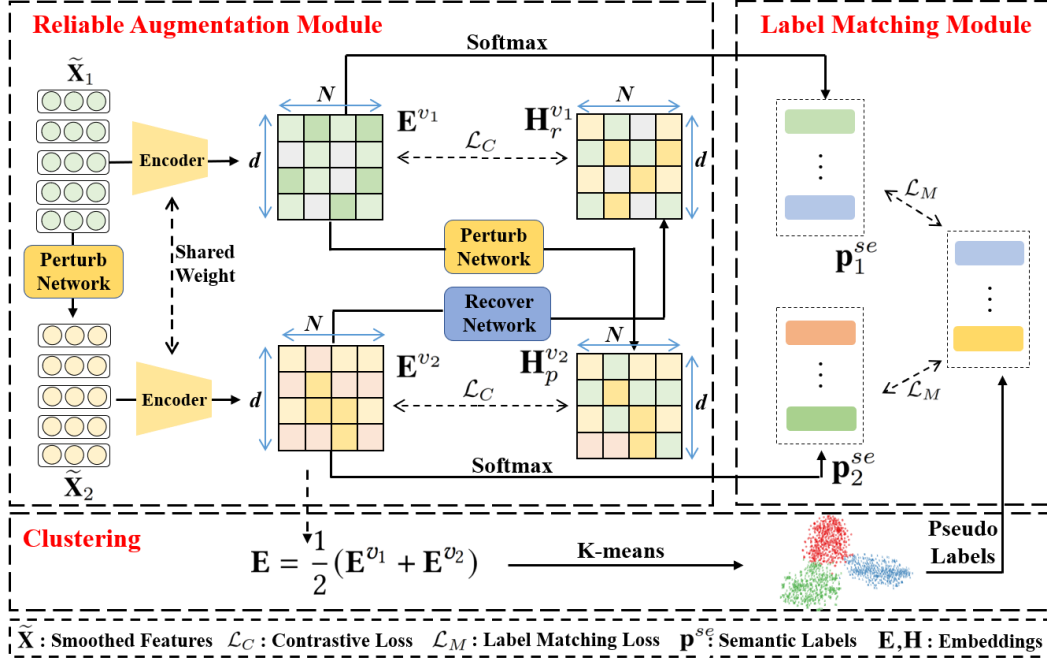


Figure 2: Illustration of CONVERT. In our method, we design a reversible perturb-recover network to generate the augmented view. The semantic information is guaranteed by the perturbation and recovery operation in the latent space. Besides, we design a semantic loss to further improve the reliability 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 \mathbf{p}^{se} with the label matching loss \mathcal{L}_M , thus guiding the model to have better performance.

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.

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 $\mathbf{G} = \{\mathbf{X}, \mathbf{A}\}$, consider $\mathbf{V} = \{v_1, v_2, \dots, v_N\}$ as a set of N nodes categorized into K classes. Here, $\mathbf{X} \in \mathbb{R}^{N \times D}$ denotes the attribute matrix, while $\mathbf{A} \in \mathbb{R}^{N \times N}$ stands for the attribute matrix and the original adjacency matrix. The degree matrix is symbolized as $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_N) \in \mathbb{R}^{N \times N}$, where $d_i = \sum_{(v_i, v_j) \in \mathcal{E}} a_{ij}$. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Utilizing the renormalization technique $\widehat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ from GCN [14], the symmetric normalized graph Laplacian matrix is represented as $\widetilde{\mathbf{L}} = \mathbf{I} - \widehat{\mathbf{D}}^{-\frac{1}{2}} \widehat{\mathbf{A}} \widehat{\mathbf{D}}^{-\frac{1}{2}}$. 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 to obtain the smoothed attribute matrix $\widetilde{\mathbf{X}}_1$ as follows:

$$\widetilde{\mathbf{X}}_1 = (\mathbf{I} - \widetilde{\mathbf{L}})^t \mathbf{X}, \quad (1)$$

where $\widetilde{\mathbf{L}}$ 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

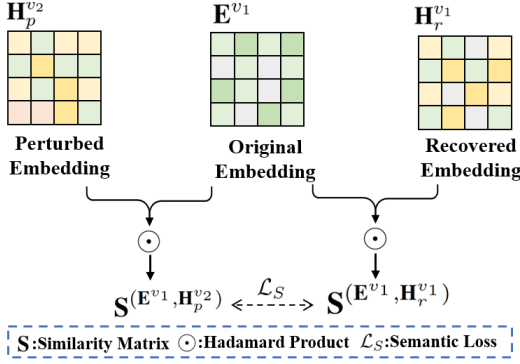


Figure 3: Illustration of the semantic loss \mathcal{L}_S . Through pulling close $\mathbf{S}^{(E^{v1}, H_r^{v1})}$ and $\mathbf{S}^{(E^{v1}, H_p^{v2})}$, the perturbation and recovery could be quantified, thus guaranteeing the reliability.

perturbations to the attribute matrix, leading to the formation of the perturbed attribute matrix $\tilde{\mathbf{X}}_2$, formulated as follows:

$$\tilde{\mathbf{X}}_2 = p(\tilde{\mathbf{X}}_1; \theta). \quad (2)$$

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

$$\begin{aligned} \mathbf{E}^{v1} &= \text{Encoder}(\tilde{\mathbf{X}}_1); \mathbf{E}_i^{v1} = \frac{\mathbf{E}_i^{v1}}{\|\mathbf{E}_i^{v1}\|_2}, i = 1, 2, \dots, N \\ \mathbf{E}^{v2} &= \text{Encoder}(\tilde{\mathbf{X}}_2); \mathbf{E}_j^{v2} = \frac{\mathbf{E}_j^{v2}}{\|\mathbf{E}_j^{v2}\|_2}, j = 1, 2, \dots, N. \end{aligned} \quad (3)$$

We adopt the multi-layers perceptions (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:

$$\mathbf{H}_r^{v1} = r(\mathbf{E}^{v2}; \theta), \quad (4)$$

where \mathbf{H}_r^{v1} denotes the recovered embeddings. Here, we adopt the two layers of multi-layers perceptions 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 \mathbf{E}^{v1} by the perturb network $p(\cdot; \theta)$ as formulated:

$$\mathbf{H}_p^{v2} = p(\mathbf{E}^{v1}; \theta), \quad (5)$$

where \mathbf{H}_p^{v2} is the perturbed embeddings. Since the smoothed attributed matrix $\tilde{\mathbf{X}}$ 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 \mathbf{S} between i -th sample in the first view and j -th sample in the second view as follows:

Algorithm 1 CONVERT

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

Output: The clustering result \mathbf{R} .

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

$$\begin{aligned} \mathbf{S}_{ij}^{(E^{v1}, H_r^{v1})} &= \mathbf{E}_i^{v1} \odot \mathbf{H}_{rj}^{v1}, \\ \mathbf{S}_{ij}^{(E^{v1}, H_p^{v2})} &= \mathbf{E}_i^{v1} \odot \mathbf{H}_{pj}^{v2}, \end{aligned} \quad (6)$$

where \odot denotes the Hadamard product. Besides, $\mathbf{S}^{(E^{v1}, H_r^{v1})}$ represents the similarity matrix between the original embeddings and the recovered embeddings. $\mathbf{S}^{(E^{v1}, H_p^{v2})}$ denotes the similarity matrix of the original embeddings and the perturbed embeddings. \mathbf{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:

$$\mathcal{L}_S = \left\| \mathbf{S}^{(E^{v1}, H_r^{v1})} - \mathbf{S}^{(E^{v1}, H_p^{v2})} \right\|_F^2. \quad (7)$$

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 \mathbf{E} as below:

$$\mathbf{E} = \frac{1}{2}(\mathbf{E}^{v_1} + \mathbf{E}^{v_2}). \quad (8)$$

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

$$\mathbf{h} = \text{top}(\mathbf{p}), \quad (9)$$

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

In the unsupervised clustering scene, the supervision information could be extracted by the high-confidence clustering 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 \mathbf{p}^{se} as follows:

$$\mathbf{p}_i^{se} = \text{Softmax}(\mathbf{E}^{v_i}), i \in 1, 2. \quad (10)$$

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

$$\mathcal{L}_M = \text{CE}(\mathbf{p}_i^{se}, \mathbf{h}), i \in 1, 2, \quad (11)$$

where $\text{CE}(\cdot)$ 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 \mathcal{L}_S , the label matching loss \mathcal{L}_M , and the contrastive loss \mathcal{L}_C . More precisely, we employ \mathcal{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:

$$\begin{aligned} \ell(\mathbf{E}_i, \mathbf{E}_j) &= -\log \frac{e^{\text{sim}(\mathbf{E}_i, \mathbf{E}_j)}}{e^{\text{sim}(\mathbf{E}_i, \mathbf{E}_j)} + \sum_{k \neq i} e^{\text{sim}(\mathbf{E}_k, \mathbf{E}_i)}}, \\ \mathcal{L}_C &= \frac{1}{2N} \sum_{i=1}^N [\ell(\mathbf{E}_i, \mathbf{E}_j), \ell(\mathbf{E}_j, \mathbf{E}_i)] \end{aligned} \quad (12)$$

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 $(\mathbf{E}^{v_1}, \mathbf{H}_r^{v_1})$ and $(\mathbf{E}^{v_2}, \mathbf{H}_p^{v_2})$. In summary, the objective of CONVERT is formulated as:

$$\mathcal{L} = \mathcal{L}_C + \alpha \mathcal{L}_S + \beta \mathcal{L}_M, \quad (13)$$

where α and β are the trade-off hyper-parameters. The detailed learning procedure of CONVERT is illustrated in Algorithm 1.

Table 2: Statistics summary of seven datasets.

Dataset	Type	Sample	Dimension	Edge	Class
CORA	Graph	2708	1433	5429	7
CITSEER	Graph	3327	3703	4732	6
AMAP	Graph	7650	745	119081	8
CORAFULL	Graph	19793	8710	63421	70
BAT	Graph	131	81	1038	4
EAT	Graph	399	203	5994	4
UAT	Graph	1190	239	13599	4

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 CORA¹, CITSEER², 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>

Table 3: The clustering performance is gauged through ten runs, encompassing mean values and standard deviations. Notably, the most exceptional and second-best outcomes are denoted by red and blue values correspondingly. "OOM" signifies out-of-memory during training.

Dataset	Metric	Classical Graph Clustering Methods			Constrastive Clustering Methods			Graph Augmentation Methods					CONVERT Ours
		DAEGC IJCAI 19	ARGA TCYB 19	SDCN WWW 20	AGE SIGKDD 20	MVGRL ICML 20	AGC-DRR IJCAI 22	GCA WWW 21	AFGRL AAAI 22	AutoSSL ICLR 22	SUBLIME WWW 22	NACL AAAI 23	
CORA	ACC	70.43±0.36	71.04±0.25	35.60±2.83	73.50±1.83	70.47±3.70	40.62±0.55	53.62±0.73	26.25±1.24	63.81±0.57	71.14±0.74	51.09±1.25	74.07±1.51
	NMI	52.89±0.69	51.06±0.52	14.28±1.91	57.58±1.42	55.57±1.54	18.74±0.73	46.87±0.65	12.36±1.54	47.62±0.45	53.88±1.02	31.80±0.78	55.57±1.12
	ARI	49.63±0.43	47.71±0.33	07.78±3.24	50.10±2.14	48.70±3.94	14.80±1.64	30.32±0.98	14.32±1.87	38.92±0.77	50.15±0.14	36.66±1.65	50.58±2.01
	F1	68.27±0.57	69.27±0.39	24.37±1.04	69.28±1.59	67.15±1.86	31.23±0.57	45.73±0.47	30.20±1.15	56.42±0.21	63.11±0.58	51.12±1.12	72.92±3.27
AMAP	ACC	75.96±0.23	69.28±2.30	53.44±0.81	75.98±0.68	41.07±3.12	76.81±1.45	56.81±1.44	75.51±0.77	54.55±0.97	27.22±1.56	67.18±0.75	77.19±0.55
	NMI	65.25±0.45	58.36±2.76	44.85±0.83	65.38±0.61	30.28±3.94	66.54±1.24	48.38±2.38	64.05±0.15	48.56±0.71	06.37±1.89	63.63±1.07	67.20±1.07
	ARI	58.12±0.24	44.18±4.41	31.21±1.23	55.89±1.34	18.77±2.34	60.15±1.56	26.85±0.44	54.45±0.48	26.87±0.34	05.36±2.14	46.30±1.59	60.79±1.83
	F1	69.87±0.54	64.30±1.95	50.66±1.49	71.74±0.93	32.88±5.50	71.03±0.64	53.59±0.57	69.99±0.34	54.47±0.83	15.97±1.53	73.04±1.08	74.03±1.00
BAT	ACC	52.67±0.00	67.86±0.80	53.05±4.63	56.68±0.76	37.56±0.32	47.79±0.02	54.89±0.34	50.92±0.44	42.43±0.47	45.04±0.19	47.48±0.64	78.02±1.36
	NMI	21.43±0.35	49.09±0.54	25.74±5.71	36.04±1.54	29.33±0.70	19.91±0.24	38.88±0.23	27.55±0.62	17.84±0.98	22.03±0.48	24.36±1.54	53.54±1.71
	ARI	18.18±0.29	42.02±1.21	21.04±4.97	26.59±1.83	13.45±0.03	14.59±0.13	26.69±2.85	21.89±0.74	13.11±0.81	14.45±0.87	24.14±0.98	51.95±2.18
	F1	52.23±0.03	67.02±1.15	46.45±5.90	55.07±0.80	29.64±0.49	42.33±0.51	53.71±0.34	46.53±0.57	34.84±0.15	44.00±0.62	42.25±0.34	77.77±1.48
EAT	ACC	36.89±0.15	52.13±0.00	39.07±1.51	47.26±0.32	32.88±0.71	37.37±0.11	48.51±1.55	37.42±1.24	31.33±0.52	38.80±0.35	36.06±1.24	58.35±0.18
	NMI	05.57±0.06	22.48±1.21	08.83±2.54	23.74±0.90	11.72±1.08	07.00±0.85	28.36±1.23	11.44±1.41	07.63±0.85	14.96±0.75	21.46±1.80	33.36±0.16
	ARI	05.03±0.08	17.29±0.50	06.31±1.95	16.57±0.46	04.68±1.30	04.88±0.91	19.61±1.25	06.57±1.73	02.13±0.67	10.29±0.88	21.48±0.64	27.11±0.19
	F1	34.74±0.16	52.75±0.07	33.42±3.10	45.54±0.40	25.35±0.75	35.20±0.17	48.22±0.33	30.53±1.47	21.82±0.98	32.31±0.97	31.25±0.96	58.42±0.22
UAT	ACC	52.29±0.49	49.31±0.15	52.25±1.91	52.37±0.42	44.16±1.38	42.64±0.31	39.39±1.46	41.50±0.25	42.52±0.64	48.74±0.54	45.38±1.15	57.36±0.55
	NMI	21.33±0.44	25.44±0.31	21.61±1.26	23.64±0.66	21.53±0.94	11.15±0.24	24.05±0.25	17.33±0.54	17.86±0.22	21.85±0.62	24.49±0.57	28.75±1.13
	ARI	20.50±0.51	16.57±0.31	21.63±1.49	20.39±0.70	17.12±1.46	09.50±0.25	14.37±0.19	13.62±0.57	13.13±0.71	19.51±0.45	21.34±0.78	27.96±0.79
	F1	50.33±0.64	50.26±0.16	45.59±3.54	50.15±0.73	39.44±2.19	35.18±0.32	35.72±0.28	36.52±0.89	52.94±0.87	46.19±0.87	30.56±0.25	54.55±1.49
CITSEER	ACC	64.54±1.39	61.07±0.49	65.96±0.31	68.73±0.24	62.83±1.59	68.32±1.83	60.45±1.03	31.45±0.54	66.76±0.67	64.14±0.65	59.23±2.32	68.43±0.69
	NMI	36.41±0.86	34.40±0.71	38.71±0.32	44.93±0.53	40.69±0.93	40.28±1.41	36.15±0.78	15.17±0.47	40.67±0.84	39.08±0.25	36.68±0.89	41.62±0.73
	ARI	37.78±1.24	34.32±0.70	40.17±0.43	45.31±0.41	34.18±1.73	45.34±2.33	35.20±0.96	14.32±0.78	38.73±0.55	39.27±0.78	33.37±0.53	42.77±1.63
	F1	62.20±1.32	58.23±0.31	63.62±0.24	64.45±0.27	59.54±2.17	64.82±1.60	56.42±0.94	30.20±0.71	58.22±0.68	61.00±0.15	52.67±0.64	62.39±2.15
CORAFULL	ACC	34.35±1.00	22.07±0.43	26.67±0.40	39.62±0.13	31.52±2.95		31.19±0.57			32.63±1.24		43.53±0.96
	NMI	49.16±0.73	41.28±0.25	37.38±0.39	52.38±0.17	48.99±3.95		50.23±1.54			38.46±0.25		54.86±0.33
	ARI	22.60±0.47	12.38±0.24	13.63±0.27	24.46±0.48	19.11±2.63		19.17±0.47			31.41±0.55		30.39±1.04
	F1	26.96±1.33	18.85±0.41	22.14±0.43	31.22±0.87	26.51±2.87		25.44±0.99			28.94±1.72		35.58±0.98

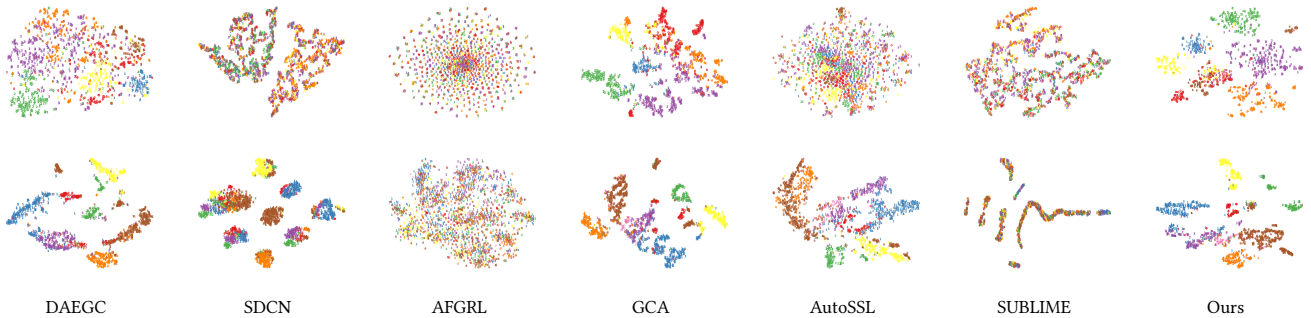


Figure 4: T-SNE visualization is employed to illustrate the performance of seven methods across two benchmark datasets. The first row pertains to the CORA dataset, while the second row corresponds to the AMAP dataset.

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], AFGRL [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 unpromising 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.

Dataset	Metric	(w/o) L_M	(w/o) R_S	(w/o) R_N	Feature Mask	Edge Remove	Edge Add	Diffusion	Ours
CORA	ACC	73.42±1.39	73.13±1.42	73.36±1.19	69.56±1.91	65.76±3.43	64.77±1.95	70.68±1.48	74.07±1.51
	NMI	55.69±0.87	55.48±0.84	55.93±0.77	52.28±2.50	50.92±1.75	49.55±2.22	53.15±0.84	55.57±1.12
	ARI	50.35±1.38	49.71±1.41	50.24±1.22	45.10±2.50	41.46±2.21	40.76±1.27	48.39±1.34	50.58±2.01
	F1	72.62±1.93	72.58±1.55	72.75±1.30	68.72±2.78	63.94±4.30	62.20±2.55	68.66±1.67	72.92±3.27
AMAP	ACC	75.96±0.58	74.93±1.03	74.95±0.77	67.55±1.13	72.52±0.62	68.21±1.81	63.49±2.17	77.19±0.55
	NMI	65.80±1.25	63.76±1.76	65.53±1.23	55.66±1.32	59.72±0.96	55.25±1.97	51.88±1.99	67.20±1.07
	ARI	57.28±1.98	55.55±1.99	56.38±2.07	45.48±1.33	51.58±1.35	46.12±1.95	41.45±2.71	60.79±1.83
	F1	72.33±2.26	71.80±1.59	70.53±1.82	64.39±1.57	67.67±1.88	64.78±2.70	61.07±2.50	74.03±1.00
BAT	ACC	76.11±1.96	69.69±3.17	68.85±2.97	61.60±2.32	49.47±2.48	64.96±2.74	64.50±3.28	78.02±1.36
	NMI	52.04±1.86	45.55±2.73	43.96±2.82	36.11±2.06	19.44±3.09	39.27±3.18	39.84±3.61	53.54±1.71
	ARI	49.70±2.89	41.38±2.78	39.53±3.12	31.82±2.92	12.72±3.39	33.86±3.27	34.92±4.16	51.95±2.18
	F1	75.61±2.11	69.10±3.48	67.87±3.54	60.33±2.40	48.23±2.12	64.08±3.40	63.56±3.99	77.77±1.48
EAT	ACC	54.59±0.46	53.93±0.58	53.83±0.88	50.25±0.93	40.38±1.56	52.31±0.92	54.99±0.63	58.35±0.18
	NMI	28.36±0.51	27.49±0.70	27.55±0.74	22.24±0.99	12.59±1.92	26.29±1.30	27.47±0.50	33.36±0.16
	ARI	22.58±0.51	21.77±1.04	22.07±1.54	16.59±1.78	17.81±1.69	19.29±1.05	24.84±0.36	27.11±0.19
	F1	55.05±0.53	54.43±0.66	53.71±1.53	50.80±1.32	39.68±1.98	53.10±1.01	53.27±1.01	58.42±0.22
UAT	ACC	49.99±1.32	49.49±1.34	48.39±0.57	47.76±2.79	49.80±1.02	50.00±1.13	52.67±1.84	57.36±0.55
	NMI	20.93±1.67	20.99±1.83	19.93±2.39	21.61±2.27	18.81±1.03	24.18±1.39	24.29±1.65	28.75±1.13
	ARI	20.02±2.26	20.10±2.31	18.35±1.65	15.84±1.83	16.06±1.22	14.39±1.88	23.33±2.23	27.96±0.79
	F1	46.52±1.42	45.68±1.12	45.84±1.42	46.72±3.52	47.20±1.84	50.54±2.61	48.08±2.84	54.55±1.49
CITSEER	ACC	64.99±1.62	65.00±1.62	64.43±1.35	63.62±1.10	66.00±1.47	64.16±1.06	65.74±0.56	68.43±0.69
	NMI	38.14±1.72	38.15±1.71	37.68±1.44	39.13±1.17	39.46±1.44	39.35±1.13	40.98±0.57	41.62±0.73
	ARI	39.12±2.32	39.14±2.32	38.20±1.78	37.09±1.73	38.66±2.24	37.78±1.43	39.66±0.91	42.77±1.63
	F1	61.22±1.38	61.23±1.38	60.82±1.62	60.36±0.85	58.50±1.24	60.39±1.01	62.00±0.81	62.39±2.15

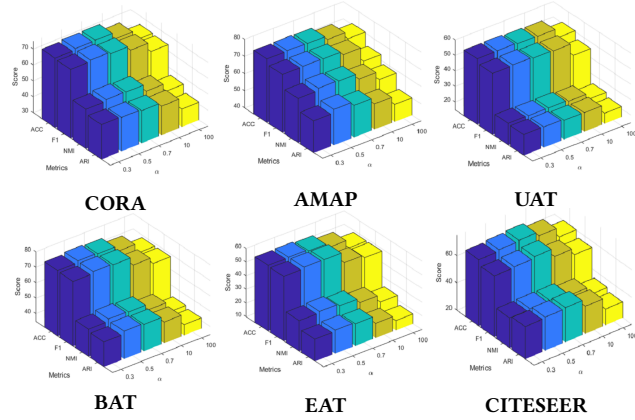


Figure 5: Sensitivity analysis of the hyper-parameter α .

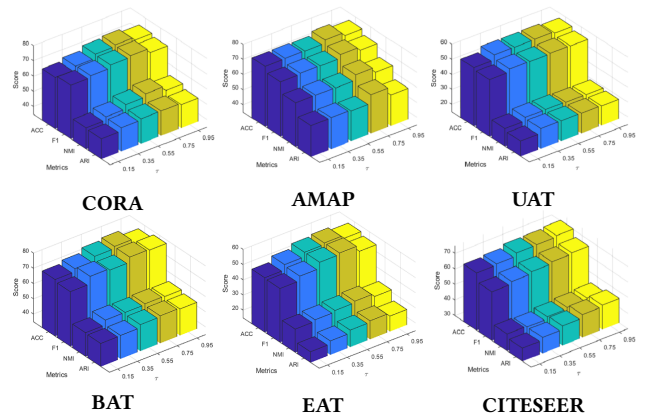


Figure 6: Sensitivity analysis of the hyper-parameter τ .

4.4 Ablation Studies (RQ2)

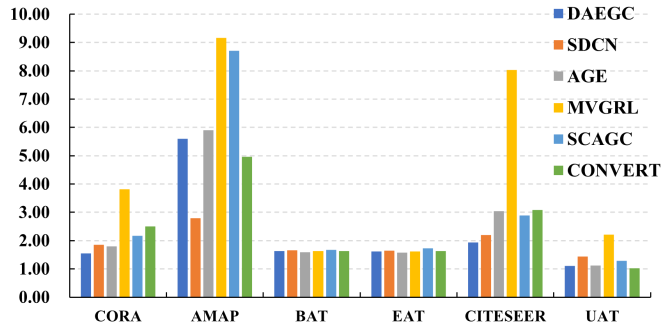
4.4.1 Effectiveness of the proposed module. As shown in Table.4, we conduct experiments to verify the effectiveness of the reversible network. To be specific, we adopt “(w/o) L_M”, “(w/o) R_S” and “(w/o) R_N” to denote the reduced models by removing the label matching module, the reliable semantic loss, 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 are 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.

Dataset	DEC	AE	DAEGC	MGAE	AGE	SDCN	MVGRL	MCGC	SCAGC	CONVERT
	ICML 2016	ICML 2017	IJCAI 2019	SIGKDD 2019	SIGKDD 2020	WWW 2020	ICML 2020	NeurIPS 2021	TMM 2022	Ours
CORA	91.13	47.31	12.97	7.38	46.65	11.32	14.72	118.07	54.08	16.89
CITSEER	223.95	74.69	14.70	6.69	70.63	11.00	18.31	126.06	50.00	57.23
BAT	21.37	7.46	4.79	3.83	2.49	11.50	3.19	2.28	93.79	5.15
EAT	26.99	9.56	5.14	4.64	3.86	12.12	3.32	2.87	47.79	5.18
UAT	42.30	29.57	6.44	4.75	8.95	10.64	4.27	23.10	64.70	6.26
AMAP	264.20	94.48	39.62	18.64	377.49	19.28	131.38	OOM	150.54	46.98
Avg.	111.66	43.85	13.94	7.66	85.01	12.64	29.20	-	76.82	22.95

**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 is 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 α . 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 α drastically changes, the balance of the model will be destroyed, thus limiting the clustering performance. Moreover, we investigate the influence of β . CONVERT is not sensitive to β . Experiments can be found in Fig. 1 in Appendix.

4.6.2 Sensitivity Analysis of hyper-parameter τ . Besides, we investigate the impact of τ . 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

- [1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
- [2] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. 2020. Structural deep clustering network. In *Proceedings of The Web Conference 2020*. 1400–1410.
- [3] Jin Chen, Defu Lian, Binbin Jin, Xu Huang, Kai Zheng, and Enhong Chen. 2022. Fast variational autoencoder with inverted multi-index for collaborative filtering. In *Proceedings of the ACM Web Conference 2022*. 1944–1954.
- [4] Jin Chen, Defu Lian, Binbin Jin, Kai Zheng, and Enhong Chen. 2022. Learning Recommenders for Implicit Feedback with Importance Resampling. In *Proceedings of the ACM Web Conference 2022*. 1997–2005.
- [5] Jin Chen, Defu Lian, Yucheng Li, Baoyun Wang, Kai Zheng, and Enhong Chen. 2022. Cache-Augmented Inbatch Importance Resampling for Training Recommender Retriever. *Advances in Neural Information Processing Systems* 35 (2022), 34817–34830.
- [6] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 976–985.

- [7] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinfeld. 2005. A tutorial on the cross-entropy method. *Annals of operations research* 134 (2005), 19–67.
- [8] Shifei Ding, Benyu Wu, Xiao Xu, Lili Guo, and Ling Ding. 2023. Graph clustering network with structure embedding enhanced. *Pattern Recognition* (2023), 109833.
- [9] Lei Gong, Sihang Zhou, Xinwang Liu, and Wenxuan Tu. 2022. Attributed Graph Clustering with Dual Redundancy Reduction. In *IJCAI*.
- [10] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.
- [11] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*. PMLR, 4116–4126.
- [12] Xu Huang, Defu Lian, Jin Chen, Liu Zheng, Xing Xie, and Enhong Chen. 2023. Cooperative Retriever and Ranker in Deep Recommenders. In *Proceedings of the ACM Web Conference 2023*. 1150–1161.
- [13] Wei Jin, Xiaorui Liu, Xiangyu Zhao, Yao Ma, Neil Shah, and Jiliang Tang. 2021. Automated self-supervised learning for graphs. *arXiv preprint arXiv:2106.05470* (2021).
- [14] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.
- [15] Namkyeong Lee, Junseok Lee, and Chanyoung Park. 2021. Augmentation-Free Self-Supervised Learning on Graphs. *arXiv preprint arXiv:2112.02472* (2021).
- [16] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *arXiv preprint arXiv:2212.05767* (2022).
- [17] Ke Liang, Lingyuan Meng, Sihang Zhou, Siwei Wang, Wenxuan Tu, Yue Liu, Meng Liu, and Xinwang Liu. 2023. Message Intercommunication for Inductive Relation Reasoning. *arXiv preprint arXiv:2305.14074* (2023).
- [18] Ke Liang, Jim Tan, Dongrui Zeng, Yongzhe Huang, Xiaolei Huang, and Gang Tan. 2023. Abslearn: a gnn-based framework for aliasing and buffer-size information retrieval. *Pattern Analysis and Applications* (2023), 1–19.
- [19] Meng Liu, Ke Liang, Bin Xiao, Sihang Zhou, Wenxuan Tu, Yue Liu, Xihong Yang, and Xinwang Liu. 2023. Self-Supervised Temporal Graph learning with Temporal and Structural Intensity Alignment. *arXiv preprint arXiv:2302.07491* (2023).
- [20] Meng Liu and Yong Liu. 2021. Inductive representation learning in temporal networks via mining neighborhood and community influences. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2202–2206.
- [21] Meng Liu, Jiaming Wu, and Yong Liu. 2022. Embedding global and local influences for dynamic graphs. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 4249–4253.
- [22] Yue Liu, Ke Liang, Jun Xia, Sihang Zhou, Xihong Yang, Xinwang Liu, and Z. Stan Li. 2023. Dink-Net: Neural Clustering on Large Graphs. In *Proc. of IJML*.
- [23] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. 2022. Deep Graph Clustering via Dual Correlation Reduction. In *AAAI Conference on Artificial Intelligence*.
- [24] Yue Liu, Jun Xia, Sihang Zhou, Siwei Wang, Xifeng Guo, Xihong Yang, Ke Liang, Wenxuan Tu, Z. Stan Li, and Xinwang Liu. 2022. A Survey of Deep Graph Clustering: Taxonomy, Challenge, and Application. *arXiv preprint arXiv:2211.12875* (2022).
- [25] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Siwei Wang, Ke Liang, Wenxuan Tu, and Liang Li. 2023. Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems* (2023).
- [26] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. 2023. Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. 8914–8922.
- [27] Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. 2022. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*. 1392–1403.
- [28] Yue Liu, Sihang Zhou, Xinwang Liu, Wenxuan Tu, and Xihong Yang. 2022. Improved Dual Correlation Reduction Network. *arXiv preprint arXiv:2202.12533* (2022).
- [29] Shirui Pan, Ruiqi Hu, Sai-fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Learning graph embedding with adversarial training methods. *IEEE transactions on cybernetics* 50, 6 (2019), 2475–2487.
- [30] Xiao Shen, Dewang Sun, Shirui Pan, Xi Zhou, and Laurence T Yang. 2023. Neighbor Contrastive Learning on Learnable Graph Augmentation. *arXiv preprint arXiv:2301.01404* (2023).
- [31] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. 2021. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems* 34 (2021), 15920–15933.
- [32] Zhiqiang Tao, Hongfu Liu, Jun Li, Zhaowen Wang, and Yun Fu. 2019. Adversarial graph embedding for ensemble clustering. In *International Joint Conferences on Artificial Intelligence Organization*.
- [33] Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, Jieren Cheng, et al. 2020. Deep Fusion Clustering Network. *arXiv preprint arXiv:2012.09600* (2020).
- [34] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [35] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Attributed graph clustering: A deep attentional embedding approach. *arXiv preprint arXiv:1906.06532* (2019).
- [36] Siwei Wang, Xinwang Liu, Li Liu, Sihang Zhou, and En Zhu. 2021. Late fusion multiple kernel clustering with proxy graph refinement. *IEEE Transactions on Neural Networks and Learning Systems* (2021).
- [37] Siwei Wang, Xinwang Liu, Xinzhou Zhu, Pei Zhang, Yi Zhang, Feng Gao, and En Zhu. 2021. Fast Parameter-Free Multi-View Subspace Clustering With Consensus Anchor Guidance. *IEEE Transactions on Image Processing* 31 (2021), 556–568.
- [38] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal representation learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*. 3562–3571.
- [39] Yiqi Wang, Chaozhuo Li, Wei Jin, Rui Li, Jianan Zhao, Jiliang Tang, and Xing Xie. 2022. Test-time training for graph neural networks. *arXiv preprint arXiv:2210.08813* (2022).
- [40] Yiqi Wang, Chaozhuo Li, Mingzheng Li, Wei Jin, Yuming Liu, Hao Sun, Xing Xie, and Jiliang Tang. 2022. Localized graph collaborative filtering. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 540–548.
- [41] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. 2021. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*. 3663–3674.
- [42] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, Juncheng Liu, and Bryan Hooi. 2020. Nodeaug: Semi-supervised node classification with data augmentation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 207–217.
- [43] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. *arXiv preprint arXiv:2202.03104* (2022).
- [44] Wei Xia, Qianqian Wang, Quanxue Gao, Ming Yang, and Xinbo Gao. 2022. Self-consistent Contrastive Attributed Graph Clustering with Pseudo-label Prompt. *IEEE Transactions on Multimedia* (2022). <https://doi.org/10.1109/TMM.2022.3213208>
- [45] Xihong Yang, Xiaochang Hu, Sihang Zhou, Xinwang Liu, and En Zhu. 2022. Interpolation-Based Contrastive Learning for Few-Label Semi-Supervised Learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–12. <https://doi.org/10.1109/TNNLS.2022.3186512>
- [46] Xihong Yang, Yue Liu, Sihang Zhou, Xinwang Liu, and En Zhu. 2022. Mixed Graph Contrastive Network for Semi-Supervised Node Classification. *arXiv preprint arXiv:2206.02796* (2022).
- [47] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Xinwang Liu, and En Zhu. 2022. Contrastive Deep Graph Clustering with Learnable Augmentation. *arXiv preprint arXiv:2212.03559* (2022).
- [48] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. 2023. Cluster-guided Contrastive Graph Clustering Network. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 10834–10842.
- [49] Yihang Yin, Qingzhong Wang, Siyu Huang, Haoyi Xiong, and Xiang Zhang. 2022. Autogcl: Automated graph contrastive learning via learnable view generators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8892–8900.
- [50] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph contrastive learning automated. In *International Conference on Machine Learning*. PMLR, 12121–12132.
- [51] Lu Yu, Shichao Pei, Lizhong Ding, Jun Zhou, Longfei Li, Chuxu Zhang, and Xiangliang Zhang. 2022. SALL: Self-Augmented Graph Contrastive Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8927–8935.
- [52] Zaixi Zhang and Qi Liu. 2023. Learning Subpocket Prototypes for Generalizable Structure-based Drug Design. *arXiv preprint arXiv:2305.13997* (2023).
- [53] Zaixi Zhang, Qi Liu, Chee-Kong Lee, Chang-Yu Hsieh, and Enhong Chen. 2023. An equivariant generative framework for molecular graph-structure Co-design. *Chemical Science* (2023).
- [54] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. 2021. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems* 34 (2021), 15870–15882.
- [55] Zaixi Zhang, Jiaxian Yan, Qi Liu, and Enhong Che. 2023. A Systematic Survey in Geometric Deep Learning for Structure-based Drug Design. *arXiv preprint arXiv:2306.11768* (2023).
- [56] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. 2021. Graph debiased contrastive learning with joint representation clustering. In *Proc. IJCAI*. 3434–3440.
- [57] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*. 2069–2080.