Appendix of CONVERT: Contrastive Graph Clustering with Reliable Augmentation

CCS CONCEPTS

• **Computing methodologies** → **Cluster analysis**.

KEYWORDS

Attribute Graph Clustering, Contrastive Learning

ACM Reference Format:

. 2022. Appendix of CONVERT: Contrastive Graph Clustering with Reliable Augmentation. In *Proceedings of 30th ACM International Conference on Multimedia (ACM Multimedia 2022).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

A DETAILS OF THE PROPOSED METHOD

We introduce the implementation of our proposed method CONVERT with PyTorch-style pseudo codes in Algorithm 1.

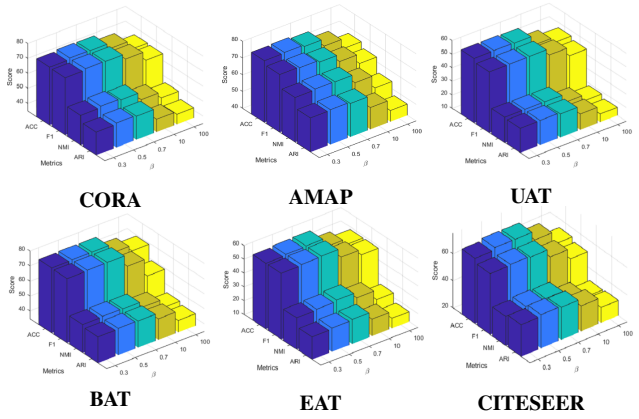

Figure 1: Sensitivity analysis of the hyper-parameter β.

B ADDITIONAL EXPERIMENTS

Due to the limitation of the original paper pages, in this section, we have conducted additional experiments to further the superiority of our proposed CONVERT, i.e., Sensitivity analysis about β, comparison experiments with other clustering methods.

We have conducted additional experiments to further the effectiveness of the proposed CONVERT. Due to the limitation of the original paper pages, in this section, we conduct additional experiments including comparison experiments and visualization analysis experiments.

ACM Multimedia 2022, October 10–14, 2018, Lisbon, Portugal

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Algorithm 1 PyTorch-style Pseudo Code of Our Method.

```
# X: Original Attribute
# A: Original Structure
# p: Perturb Network
# r: Recover Network
# h: High-confidence Pseudo Labels
# N: High-confidece Epoch
# CE: Cross-Entropy Function
# L_c: NT-Xent loss
# L_s: Semantic loss
# alpha, beta: trade-off parameters
# clu_res: Cluster Results
for epoch in range(epoch_num):

   #Smooth the feature
   X1 = graph_filter(X, A)

   # Perturb the Attribute Matrix
   X2 = p(X1)

   # Net Encoding
   E1 = F.normalization(X1,dim=1,p=2)
   E2 = F.normalization(X2,dim=1,p=2)

   #Perturb and Recover Embeddings
   H1 = r(E2)
   H2 = p(E1)

   # Calculate the similarity of embeddings
   S_r = E1 * H1
   S_p = E2 * H2

   L_s = MSE(S_r, S_p)

   # Clustering and High-confidence Pseudo Label
   clu_res, h = clustering((E1+E2/2))

   loss_c = L_c (E1, H1) + L_c (E2, H2)
   loss = loss_c + alpha * L_s

   if epoch > N:
      # Label Matching
      L_m = CE(h, softmax(E1)) + CE(h, softmax(E2))
      loss = loss_c + alpha * L_s + beta * L_m
   # optimization
   loss.backward()
   optimizer.step()
clu_res, _, _ = clustering((E1+E2/2))
return clu_res
```

B.1 Sensitivity Analysis of hyper-parameter β

As can be observed in Fig.1, we observe that the performance of CONVERT will not fluctuate greatly when the $\alpha \in [0.3, 0.7]$. When the value of β drastically change, the balance of the model will be destroyed, thus limiting the clustering performance.

Table 1: Statistics and hyper-parameter settings of six benchmark datasets.

	Dataset	CORA	CITE	AMAP	BAT	EAT	UAT	Corafull
Statistics	Type	Graph	Graph	Graph	Graph	Graph	Graph	Graph
	# Samples	2708	3327	7650	131	399	1190	19793
	# Dimensions	1433	3703	745	81	203	239	8710
	# Edges	5429	4732	119081	1038	5994	13599	63421
	# Classes	7	6	8	4	4	4	70
Hyper-parameters	τ	0.75	0.75	0.75	0.75	0.75	0.75	0.75
	α	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	β	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	Learning rate	10^{-5}	10^{-3}	10^{-3}	10^{-3}	10^{-7}	10^{-3}	10^{-4}

Table 2: Additional comparison experiments on six benchmark datasets. The clustering performance is evaluated by four metrics with mean value and standard deviation.

Dataset	Metric	Deep Clustering				Deep Graph Clustering				
		MGAE CIKM 2019	DCN ICML 2017	DEC ICML 2016	AdaGAE TPAMI 2021	DFCN AAAI 2021	GDCL IJCAI 2021	SLAPS NIPS 2021	DCRN AAAI 2022	CONVERT Ours
CORA	**ACC**	43.38±2.11	49.38±0.91	46.50±0.26	50.06±1.58	36.33±0.49	70.83±0.47	64.21±0.12	61.93±0.47	74.07±1.51
	NMI	28.78±2.97	25.65±0.65	23.54±0.34	32.19±1.34	19.36±0.87	56.60±0.36	41.16±1.24	45.13±1.57	55.57±1.12
	ARI	16.43±1.65	21.63±0.58	15.13±0.42	28.25±0.98	04.67±2.10	48.05±0.72	35.96±0.65	33.15±0.14	50.58±2.01
	F1	33.48±3.05	43.71±1.05	39.23±0.17	53.53±1.24	26.16±0.50	52.88±0.97	63.72±0.26	49.50±0.42	72.92±3.27
AMAP	**ACC**	71.57±2.48	48.25±0.08	47.22±0.08	67.70±0.54	76.82±0.23	43.75±0.78	60.09±1.14		77.19±0.55
	NMI	62.13±2.79	38.76±0.30	37.35±0.05	55.96±0.87	66.23±1.21	37.32±0.28	51.15±0.87	OOM	67.20±1.07
	ARI	48.82±4.57	20.80±0.47	18.59±0.04	46.20±0.45	58.28±0.74	21.57±0.51	42.87±0.75		60.79±1.83
	F1	68.08±1.76	47.87±0.20	46.71±0.12	62.95±0.74	71.25±0.31	38.37±0.29	47.73±0.98		74.03±1.00
BAT	**ACC**	53.59±2.04	47.79±3.95	42.09±2.21	43.51±0.48	55.73±0.06	45.42±0.54	41.22±1.25	67.94±1.45	78.02±1.36
	NMI	30.59±2.06	18.03±7.73	14.10±1.99	15.84±0.78	48.77±0.51	31.70±0.42	17.05±0.87	47.23±0.74	53.54±1.71
	ARI	24.15±1.70	13.75±6.05	07.99±1.21	07.80±0.41	37.76±0.23	19.33±0.57	06.86±2.14	39.76±0.87	51.95±2.18
	F1	50.83±3.23	46.80±3.44	42.63±2.35	43.15±0.77	50.90±0.12	39.94±0.57	37.64±0.57	67.40±0.35	77.77±1.48
EAT	**ACC**	44.61±2.10	38.85±2.32	36.47±1.60	32.83±1.24	49.37±0.19	33.46±0.18	48.62±1.65	50.88±0.55	58.35±0.18
	NMI	15.60±2.30	06.92±2.80	04.96±1.74	04.36±1.87	32.90±0.41	13.22±0.33	28.33±2.56	22.01±1.23	33.36±0.16
	ARI	13.40±1.26	05.11±2.65	03.60±1.87	02.47±0.54	23.25±0.18	04.31±0.29	24.59±0.58	18.13±0.85	27.11±0.19
	F1	43.08±3.26	38.75±2.25	34.84±1.28	32.39±0.47	42.95±0.04	25.02±0.21	40.42±1.44	47.06±0.66	58.42±0.22
UAT	**ACC**	48.97±1.52	46.82±1.14	45.61±1.84	52.10±0.87	33.61±0.09	48.70±0.06	49.77±1.24	48.70±0.06	57.36±0.55
	NMI	20.69±0.98	17.18±1.60	16.63±2.39	26.02±0.71	26.49±0.41	25.10±0.01	12.86±0.65	25.10±0.01	28.75±1.13
	ARI	18.33±1.79	13.59±2.02	13.14±1.97	24.47±0.13	11.87±0.23	21.76±0.01	17.36±0.98	21.76±0.01	27.96±0.79
	F1	47.95±1.52	45.66±1.49	44.22±1.51	43.44±0.85	25.79±0.29	45.69±0.08	10.56±1.34	45.69±0.08	54.55±1.49
CITESEER	**ACC**	61.35±0.80	57.08±0.13	55.89±0.20	54.01±1.11	69.50±0.20	66.39±0.65	64.76±0.07	66.39±0.65	68.43±0.69
	NMI	34.63±0.65	27.64±0.08	28.34±0.30	27.79±0.47	43.90±0.20	39.52±0.38	39.11±0.06	39.52±0.38	41.62±0.73
	ARI	33.55±1.18	29.31±0.14	28.12±0.36	24.19±0.85	45.50±0.30	41.07±0.96	37.54±0.12	41.07±0.96	42.77±1.63
	F1	57.36±0.82	53.80±0.11	52.62±0.17	51.11±0.64	64.30±0.20	61.12±0.70	59.64±0.05	61.12±0.70	62.39±2.15

B.2 Statistics and Hyper-parameter Settings

To guarantee reproducibility, we report the statistics summary and hyper-parameter settings of our proposed method in Table 1.

B.3 Additional Comparison Experiments

Due to the limitation of the pages,in this section, we have conducted additional experiments to further the superiority of our proposed CONVERT. Specifically, two categories methods are compared in this section, i.e. deep clustering methods (MGAE [5], DCN [7], DEC [6], AdaGAE [2]), and deep graph clustering methods (DFCN [4], GDCL [8], SLAPS [1], DCRN [3]). The experiment results are shown in Table.2. We could observe as follows. 1) Deep clustering methods are not comparable with our proposed methods. We conjecture that those methods overlook the graph structure. 2) CONVERT could achieve better performance than deep graph clustering methods. The reason is that the proposed learnable augmentation strategy ensures the semantic reliability of the augmented view, thus enhancing the supervision information capture capability of our method.

REFERENCES

[1] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. 2021. SLAPS: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems* 34 (2021), 22667–22681.

[2] Xuelong Li, Hongyuan Zhang, and Rui Zhang. 2021. Adaptive graph auto-encoder for general data clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[3] 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*.

[4] 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).

[5] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. 2017. Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 889–898.

[6] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. PMLR, 478–487.

[7] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*. PMLR, 3861–3870.

[8] 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.