

Supplementary Materials: Contrastive Graph Distribution Alignment for Partially View-Aligned Clustering

Xibiao Wang*
Shantou University
Shantou, Guangdong, China
w1574485261@gmail.com

Hang Gao*
Jilin University
Changchun, Jilin, China
gaohang23@mails.jlu.edu.cn

Xindian Wei
City University of Hong Kong
Hong Kong, China
xindiawei2-c@my.cityu.edu.hk

Liang Peng
Shantou University
Shantou, Guangdong, China
23lpeng@stu.edu.cn

Rui Li
Shantou University
Shantou, Guangdong, China
rli@stu.edu.cn

Cheng Liu†
Shantou University
Shantou, Guangdong, China
chengliu10@gmail.com

Si Wu
South China University of Technology
Guangzhou, Guangdong, China
cswusi@scut.edu.cn

Hau-San Wong
City University of Hong Kong
Hong Kong, China
cshswong@cityu.edu.hk

1 Introduction

In this supplementary material, we provide additional information of our approach, including details about the network architecture, optimization algorithm, convergence analysis of the model training, and effectiveness analysis of the alignment module. In addition, in the comparison experiments of the main paper, for the partially view-aligned data, we employ the Hungarian algorithm to compute the sample correspondence on the PCA projection data, and then apply the standard MVC method for clustering. For a more comprehensive comparison, we utilize autoencoders instead of PCA techniques and present the results of the baseline on partially aligned data.

2 Experiment Details

In this section, we will first introduce the details of the datasets used, our network architecture and optimization algorithm. Subsequently, we will conduct a series of experimental analyses.

2.1 Dataset

Like most partially view-aligned clustering methods, we also select two view data for experimentation. Next, we will introduce the details of the six datasets used in the experiments:

- **HandWritten** A widely-used image dataset comprises 2,000 handwritten digital images ranging from 0 to 9. Each sample is represented by six distinct feature sets: 216-dimensional FAC, 76-dimensional FOU, 64-dimensional KAR, six MORs, 240-dimensional Pix, and 47-dimensional ZER. Following by [12], we select Pix and FAC as the two views in this experiment.
- **Scene-15**. The dataset comprises a total of 4,485 images with 15 scene categories, encompassing both indoor and outdoor environments. Following [4], we utilize two features for experimentation, namely, PHOG and GIST features.

Table 1: The architecture of the autoencoder used in the contrastive cross-view representation learning module

Encoder	Decoder
Dense(ReLU, size = 128)	Dense(ReLU, size = 128)
Dense(ReLU, size = 256)	Dense(ReLU, size = 256)
Dense(ReLU, size = 128)	Dense(Softmax, size = 128)

- **BDGP**. This dataset contains 2,500 images of drosophila embryos, divided into 5 categories. Each sample is represented by visual and textual features, where we use visual features with dimension 1,750 and textual features with dimension 79 as the two views used in the experiment.
- **Caltech101**. This is a widely-used image dataset for multi-view learning, consisting of 9144 images divided into 101 object categories and one background category. For our experiments, we select two subsets, namely, **Caltech101-7** and **Caltech101-20**. **Caltech101-7** comprises 7 categories with a total of 1,474 samples. **Caltech101-20** contains 20 categories with a total of 2,386 samples. Following [17], experiments are conducted using the 1,984-dimensional HOG feature and the 512-dimensional GIST feature as the two views.
- **Reuters-dim10**. A subset of the Reuters dataset is used containing 9,379 instances over 6 categories. Following [14], German and Spanish texts serve as two distinct views.

2.2 Network Design

In this subsection, we detail the network model of our approach. The main network of this method consists of three modules, namely the representation learning module, the contrastive cross-view representation learning module and the alignment module. The representation learning module contains two autoencoders for processing two view data. The structure of the autoencoders used for each dataset as shown in Table 2. Among them, Dense represents the

*Both authors contributed equally to this research.

† Corresponding author.

fully connected layer. The contrastive cross-view representation learning module is also composed of two autoencoders. For each data set we use the same architecture, as shown in the Table 1. For the alignment module, we define two tensors for optimal solution of the final alignment matrix \mathbf{P} .

Algorithm 1: Optimization Algorithm

Input: Given a partially view-aligned dataset $\{\mathbf{X}^v\}_{v=1}^m$, the index set Ω of aligned data, the number of training epochs T , and the number of clusters K

Output: Clustering result and trained model
Initialize encoders θ_e^i , decoders θ_d^i , and $\{\mathcal{F}^v\}_{v=1}^m$. Set the learning rates and other hyperparameters.

1. Pre-training using aligned data:

Pre-train encoders and decoders based on reconstruction loss \mathcal{L}_r and contrastive loss \mathcal{L}_{cl} .

2. Training using all data:

while $t < T$ **do**

- Update $\{\mathcal{F}^v\}_{v=1}^m$ for semantic features $\{\tilde{\mathbf{Z}}^v\}_{v=1}^m$ using aligned cross-view constrative loss \mathcal{L}_{acc} .
- Learn latent features $\{\mathbf{Z}^v\}_{v=1}^m$ with losses \mathcal{L}_r , \mathcal{L}_{cl} , and \mathcal{L}_{acc} .
- Calculate similarity graphs $\{\mathbf{S}^v\}_{v=1}^m$ and their related graph distributions based on semantic features $\{\tilde{\mathbf{Z}}^v\}_{v=1}^m$ for each view.
- Learn the view alignment matrix \mathbf{P} utilizing \mathcal{L}_{alg} via stochastic gradient descent optimization.
- $t = t + 1$.

end

3. Clustering: The clustering results are obtained from semantic features with the application of K-means algorithm.

2.3 Optimization

The overall optimization process of the proposed method is summarized in Algorithm 1. The training process can be outlined as the following steps:

- We adopt a pretraining strategy to initially optimize the entire network using data with known view correspondence. In the pretraining stage, the reconstruction loss \mathcal{L}_r and contrastive loss \mathcal{L}_{cl} are utilized to train the encoder and decoder, mapping the input data to appropriate latent space.
- In the subsequent epochs, unaligned data is included to perform representation learning and alignment in an iterative order. Specifically, the contrastive representation module is employed to learn the latent features and the semantic features. The view alignment module is trained on the similarity graphs with \mathcal{L}_{alg} to obtain the alignment matrix \mathbf{P} . These two modules are referenced and learned in a cyclical manner.
- Finally, a common representation is obtained by concatenating the view-specific semantic features, followed by the application of the K-means clustering algorithm to derive the final clustering results.

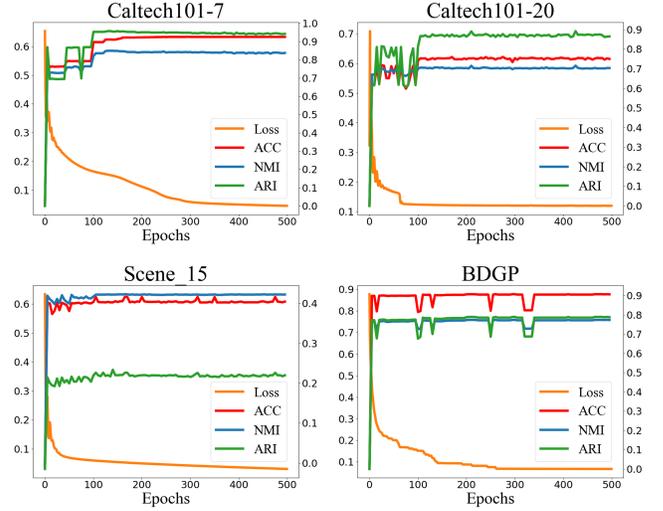


Figure 1: Convergence curve of our proposed method depicts the trajectory of both the loss value and clustering performance.

2.4 Convergence Analysis

In this subsection, we perform a convergence analysis of the proposed method. As depicted in Fig. 1, there is a discernible trend wherein the clustering effectiveness demonstrates an upward trajectory as the loss values decrease. This observation suggests that the proposed method exhibits a favorable convergence property, signifying its ability to iteratively improve and optimize clustering performance.

2.5 Experiment with AEs

In the main body of our submission (Section 4.2), we present the results on partially aligned data compared with multi-view clustering approaches, including: CCA [9], KCCA [2], DCCA [1], DCCAE[11], MvC-DMF [18], SwMC[7], GMC [10], AE²-NETs [16], LMVSC [5], SMVSC [8], OPMV [6], FastMICE [3], MVC-UM [15], PVC [4], Mv-CLN [14], SURE [13]. Among them, for the standard Multi-View Clustering (MVC) methods, we adopt PCA to project the original data into the latent space and employ the Hungarian algorithm to establish corresponding relationships, and then use the aligned data as input data. For a more complete comparison, we utilize the autoencoder illustrated in Table 4 to project the original data into a lower-dimensional space. Subsequently, we employ the Hungarian algorithm to establish cross-view correspondence. Afterwards, we perform the above baseline on the aligned data and implement clustering with the same settings as in the main paper. Among them, for our method and the other four partial view-aligned clustering methods, we directly adopt the results in the main paper. As shown in Table 3, the performance of our method outperforms other methods significantly on most datasets. Apart from achieving suboptimal results on the ARI index clustering for the Scene-15 dataset, our method achieves optimal results on the others. Specifically, in terms of NMI, our method achieves a 4.77% (HandWritten), 2.05% (Scene-15), 3.84% (BDGP), 29.78% (Caltech101-7), 6.99% (Caltech101-20),

Table 2: The architecture of the autoencoders used in our method. Here, we present only the structure of the encoders, and the decoders consists of the same layers in reverse order.

Dataset	Encoder-Layer1	Encoder-Layer2	Encoder-Layer3	Encoder-Layer4
HandWritten	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(Softmax, size = 20)
Scene-15	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(Softmax, size = 128)
BDGP	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(Softmax, size = 10)
Caltech101-7	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(Softmax, size = 128)
Caltech101-20	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(Softmax, size = 128)
Reuters	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(ReLU, size = 1024)	Dense(Softmax, size = 10)

Table 3: Comparison of clustering performance with a *partially* aligned setting on six benchmark datasets, with the best results highlighted in red and the second-best results in blue.

Methods	HandWritten			Scene-15			BDGP			Caltech101-7			Caltech101-20			Reuters		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI									
CCA	65.75	55.26	46.92	38.10	37.31	21.62	62.84	37.17	34.05	39.42	22.71	15.13	28.29	32.70	13.19	49.09	26.35	23.00
KCCA	43.00	29.86	20.15	33.29	27.80	16.01	46.40	16.34	12.19	32.90	16.97	10.24	23.30	21.96	7.18	47.86	23.33	21.42
DCCA	55.35	48.76	35.40	27.36	27.81	12.91	51.72	21.95	18.72	50.07	52.36	37.27	35.33	48.48	26.03	39.11	11.70	11.62
DCCAE	56.15	52.43	38.11	26.00	26.92	12.80	47.08	17.50	14.99	52.04	53.60	37.97	40.32	55.99	30.06	40.65	17.09	16.45
MvC-DMF	41.60	31.52	21.14	19.19	8.41	3.73	28.94	3.26	1.48	41.38	8.73	1.32	29.33	19.77	8.02	31.76	10.34	3.23
SwMC	28.60	21.97	9.42	9.74	0.66	-0.01	23.88	2.88	0.64	54.88	4.10	5.47	29.63	14.96	2.07	27.49	0.10	-0.02
GMC	21.95	19.98	2.25	11.51	5.42	0.04	28.56	12.55	2.41	54.27	14.94	5.52	37.05	25.72	4.78	31.92	9.60	-0.05
AE ² -NETs	77.60	74.07	66.85	23.92	22.12	9.90	40.68	15.82	6.34	34.19	18.00	7.45	36.04	38.79	23.28	34.88	3.23	2.91
LMVCS	71.85	68.65	59.10	35.90	35.86	18.93	44.72	25.21	19.23	63.91	35.05	19.77	47.49	62.44	33.77	45.08	20.05	18.10
SMVSC	53.40	42.92	32.09	23.99	14.60	9.00	55.12	36.83	28.28	66.15	49.24	47.79	49.33	55.42	39.11	49.86	19.44	20.36
OPMV	74.70	74.50	66.50	18.22	7.85	3.29	41.16	14.25	6.40	45.59	45.99	35.29	44.22	58.06	32.97	43.05	15.03	12.19
FastMICE	62.57	54.42	46.17	36.14	29.73	17.62	53.65	29.73	26.18	43.01	31.79	23.33	40.30	43.29	23.29	36.38	16.00	9.99
PVC	76.45	74.47	66.22	37.88	39.12	20.63	89.24	73.56	74.93	50.14	53.54	38.38	48.95	64.19	38.34	35.34	16.12	11.55
MVC-UM	71.45	69.16	60.47	25.70	27.70	11.54	46.68	21.88	8.81	55.50	45.32	37.38	43.25	60.14	32.30	36.87	13.96	15.16
MvCLN	64.55	62.29	49.32	38.53	39.90	24.26	73.04	46.15	44.28	45.52	50.34	36.87	46.19	56.69	41.43	50.63	32.69	26.77
SURE	77.31	72.42	63.01	38.67	40.00	22.53	79.29	57.95	55.87	41.00	45.98	26.79	53.44	59.30	41.90	51.01	32.11	25.76
Ours	83.16	79.24	73.01	41.63	42.05	22.57	90.74	77.40	78.84	92.33	83.32	94.69	76.02	71.18	89.27	54.66	35.93	29.90

Table 4: The architecture of the autoencoder.

Encoder	Decoder
Dense(ReLU, size = 500)	Dense(ReLU, size = 10)
Dense(ReLU, size = 500)	Dense(ReLU, size = 2000)
Dense(ReLU, size = 2000)	Dense(ReLU, size = 500)
Dense(Tanh, size = 10)	Dense(ReLU, size = 500)

and 3.24% (Reuters) progress respectively compared with the best baseline.

2.6 Effectiveness of View-Aligned Learning

The primary contribution of our proposed method lies in its capacity to learn correspondences for view-alignment. In the main body of our submission (Section 4.3.4), we verified the effectiveness of the alignment module through Sankey diagrams and heatmaps. To further validate the effectiveness of the alignment module, we assess the quality of the learned view-alignment matrix. Specifically, we compare the alignment matrices obtained through the Hungarian algorithm, PVC, and our method on the BDGP, HandWritten, and Caltech101-7 datasets, as illustrated in Fig. 2. As observed, the view

correspondences learned by our method predominantly align with the true block-diagonal structure. In contrast, the view correspondences obtained through the Hungarian algorithm and PVC may exhibit inconsistencies with the ground truth cluster alignment blocks. This outcome suggests that the alignment matrix learned by our method can accurately capture cluster-level alignment information in comparison to the Hungarian algorithm and PVC approaches.

2.7 Effectiveness of Semantic Feature Learning

Strictly contrasting features may result in the dominance of irrelevant, view-specific noise, which can distort the extraction of semantic consistency and overlook available complementary information. In contrast, our goal is to learn view correspondences based on graph distribution metrics that capture semantic view-invariant instance relationships. This approach enables the identification of view-invariant semantic structures while avoiding view-specific noise. Additionally, our semantic feature learning method employs a feature extraction strategy to generate semantic features by leveraging such view-invariant semantic structures with cross-view contrastive loss (Eq.6). Furthermore, we compare our semantic feature learning method with a latent feature

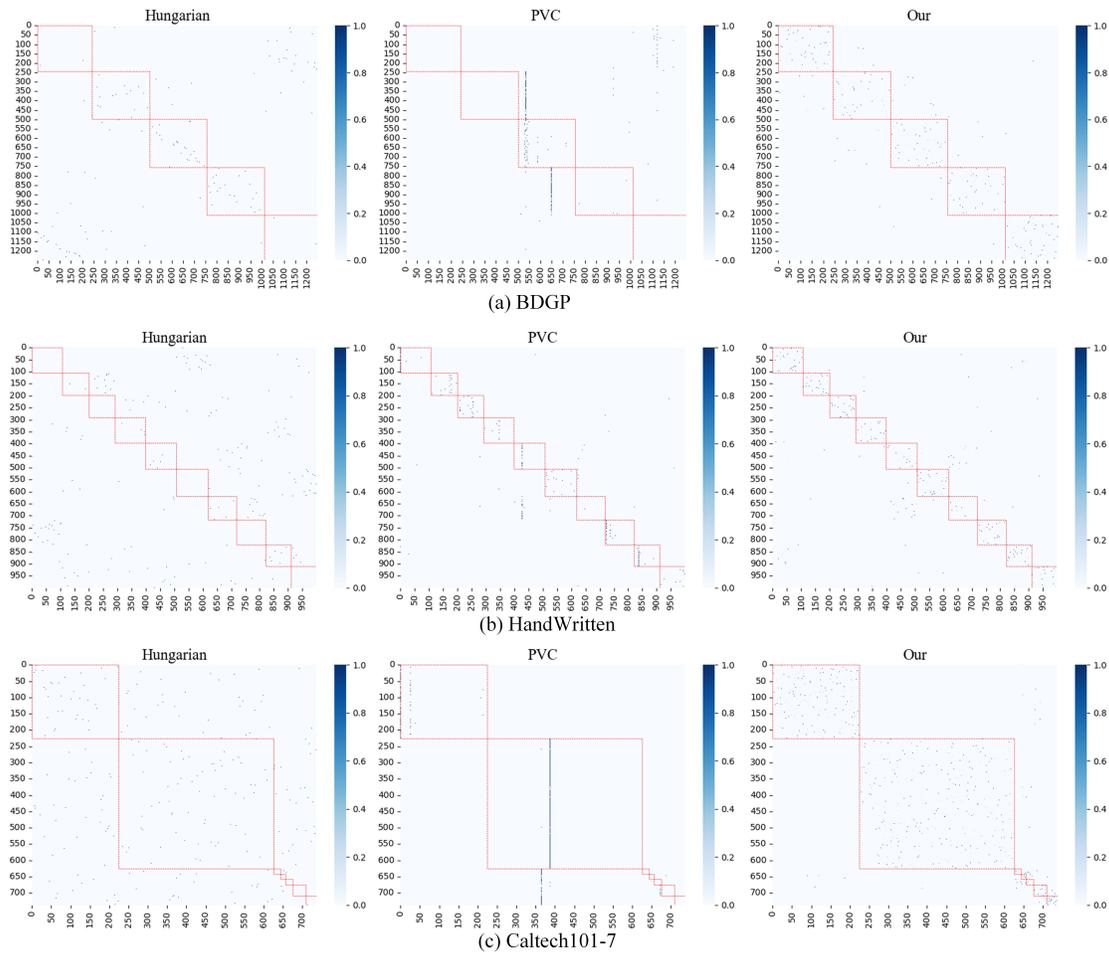


Figure 2: Convergence curve of our proposed method depicts the trajectory of both the loss value and clustering performance.

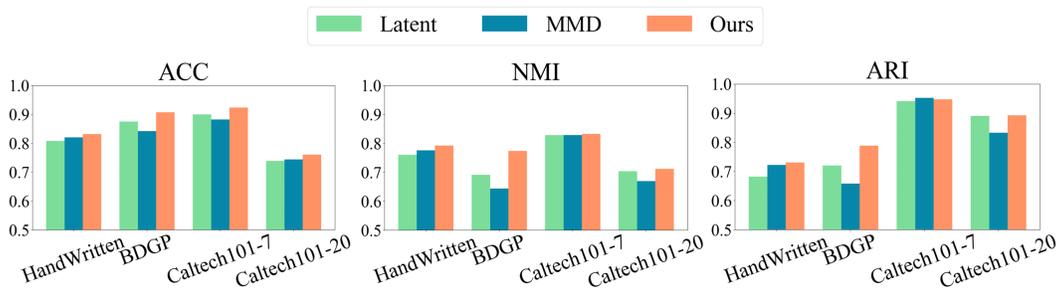


Figure 3: Comparison results of our method, latent feature learning and feature learning using MMD.

learning method and a latent feature learning method using MMD to mitigate inter-view discrepancies. As demonstrated in Fig. 3, our semantic feature learning approach outperforms the others, indicating its effectiveness in reducing inter-view discrepancies.

2.8 Comparison with Single View Learning

We compare the clustering performance of our method with the best results from all single-view under completely unaligned settings, as shown in Table 5. This demonstrates that our method significantly outperforms the best results of single-view approaches, indicating

its effectiveness of avoiding learning inaccurate view alignments which could lead to degradation of clustering performance.

Table 5: Clustering results of our method with the best results from all views (BS) under completely misaligned settings.

Type	HandWritten			BDGP			Caltech101-7		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
BS	53.70	54.06	40.58	47.20	32.94	25.97	72.93	59.14	66.69
Ours	61.95	56.21	41.68	59.80	38.39	25.34	86.49	65.31	80.24

2.9 Comparison with Euclidean-Based View-Alignment

To further evaluate the effectiveness of graph distribution alignment, we compare it with the direct optimization of Euclidean distance on latent features and similarity graphs, as shown in Table 6. The results indicate that alignment using graph distribution outperforms other methods.

Table 6: Comparison with different view-alignment learning.

Type	Caltech101-7			Caltech101-20			BDGP		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Latent Feature	83.79	77.88	87.36	69.82	61.62	80.33	90.08	74.26	76.99
Graph Node	89.28	81.00	94.74	74.31	72.97	84.83	90.08	73.85	77.09
Graph Distribution	92.33	83.32	94.69	76.01	71.18	89.27	90.74	77.40	78.84

2.10 Visualization of graph distribution comparison

Furthermore, we utilize the learned view-alignment matrix to restore the alignment of the similarity graph obtained from semantic features (Eq.(7)). We then compare this restored graph to one aligned with the ground-truth view matrix. As depicted in Figure. 4, the restored graph exhibits a block-diagonal structure similar to that aligned with the ground truth matrix. By recovering the block-diagonal similarity graph, our learned alignment is able to appropriately match related samples across views. This analysis of the aligned similarity graphs lends further support that the alignment module successfully establishes meaningful cross-view mappings between the different views.

References

- [1] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *International conference on machine learning*. PMLR, 1247–1255.
- [2] Francis R Bach and Michael I Jordan. 2002. Kernel independent component analysis. *Journal of machine learning research* 3, Jul (2002), 1–48.
- [3] Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. 2023. Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity. *IEEE Transactions on Knowledge and Data Engineering* (2023).
- [4] Zhenyu Huang, Peng Hu, Joey Tianyi Zhou, Jiancheng Lv, and Xi Peng. 2020. Partially view-aligned clustering. *Advances in Neural Information Processing Systems* 33 (2020), 2892–2902.
- [5] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. 2020. Large-scale multi-view subspace clustering in linear time. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4412–4419.
- [6] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Li Liu, Siqi Wang, Weixuan Liang, and Jiangyong Shi. 2021. One-pass multi-view clustering for large-scale data. In *Proceedings of the IEEE/CVF international conference on computer vision*. 12344–12353.

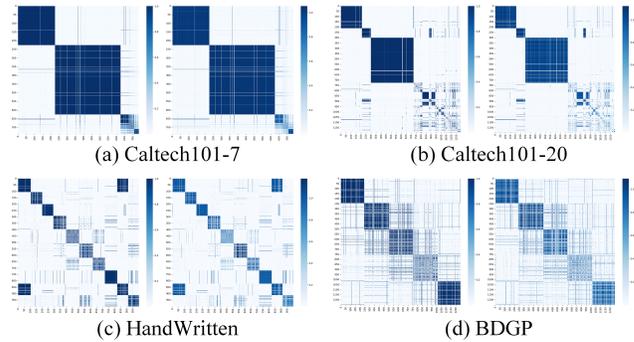


Figure 4: Comparison between the similarity graph restored using the ground truth alignment matrix and the one obtained through the learned view-alignment matrix on Caltech101-7, Caltech101-20, HandWritten, and BDGP. Please enlarge the figure for better visual results.

- [7] Feiping Nie, Jing Li, Xuelong Li, et al. 2017. Self-weighted Multiview Clustering with Multiple Graphs. In *IJCAI*. 2564–2570.
- [8] Mengjing Sun, Pei Zhang, Siwei Wang, Sihang Zhou, Wenxuan Tu, Xinwang Liu, En Zhu, and Changjian Wang. 2021. Scalable multi-view subspace clustering with unified anchors. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3528–3536.
- [9] Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. 2002. Inferring a semantic representation of text via cross-language correlation analysis. *Advances in neural information processing systems* 15 (2002).
- [10] Hao Wang, Yan Yang, and Bing Liu. 2019. GMC: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering* 32, 6 (2019), 1116–1129.
- [11] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In *International conference on machine learning*. PMLR, 1083–1092.
- [12] Yi Wen, Siwei Wang, Qing Liao, Weixuan Liang, Ke Liang, Xinhang Wan, and Xinwang Liu. 2023. Unpaired Multi-View Graph Clustering With Cross-View Structure Matching. *IEEE Transactions on Neural Networks and Learning Systems* (2023).
- [13] Mouxing Yang, Yunfan Li, Peng Hu, Jinfeng Bai, Jiancheng Lv, and Xi Peng. 2022. Robust multi-view clustering with incomplete information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2022), 1055–1069.
- [14] Mouxing Yang, Yunfan Li, Zhenyu Huang, Zitao Liu, Peng Hu, and Xi Peng. 2021. Partially view-aligned representation learning with noise-robust contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1134–1143.
- [15] Hong Yu, Jia Tang, Guoyin Wang, and Xinbo Gao. 2021. A novel multi-view clustering method for unknown mapping relationships between cross-view samples. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2075–2083.
- [16] Changqing Zhang, Yeqing Liu, and Huazhu Fu. 2019. Ae2-nets: Autoencoder in autoencoder networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2577–2585.
- [17] Xianchao Zhang, Mengyan Chen, Jie Mu, and Linlin Zong. 2023. Adaptive View-Aligned and Feature Augmentation Network for Partially View-Aligned Clustering. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 223–235.
- [18] Handong Zhao, Zhengming Ding, and Yun Fu. 2017. Multi-view clustering via deep matrix factorization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.