Supplementary Material for Partially View-aligned Representation Learning with Noise-robust Contrastive Loss

Mouxing Yang[1], Yunfan Li[1], Zhenyu Huang[1], Zitao Liu[2], Peng Hu[1], Xi Peng[1*]
[1] College of Computer Science, Sichuan University.
[2] TAL Education Group, Beijing China.
{yangmouxing, yunfanli.gm, zyhuang.gm, zitao.jerry.liu, penghu.ml, pengx.gm}@gmail.com

1. Introduction

In this supplementary material, we elaborate on the details of the datasets used and conduct additional experiments to verify the effectiveness of MvCLN.

2. Additional Experiments

In this section, we carry out additional experiments including classification and ablation studies to further show the effectiveness of the proposed MvCLN.

2.1. Details of the Datasets

- **Scene-15** [5]: The dataset consists of 4,485 images distributed over 15 indoor and outdoor scene categories. Similar to [4], two image features are extracted as views, *i.e.*, 20-dim GIST feature and 59-dim PHOG feature;

- **Caltech-101** [8]: The dataset consists of 9,144 images associated with 101 object categories, as well as an additional background category. Following [16], two features are used as views, *i.e.*, 1,984-dim HOG feature and 512-dim GIST feature;

- **Reuters** [1]: A subset, which consists of 18,758 samples of six classes, is used. Similar to [7], we use the first two languages (English and French) as two views and apply a standard autoencoder to project the data into a 10-dim space for faster speed;

- **NoisyMNIST** [13]: The dataset consists of 70,000 samples from 10 classes. As the baselines cannot handle a large-scale dataset, we randomly select 30,000 samples for evaluation.

*Corresponding author

2.2. Classification Performance

To further verify the effectiveness of MvCLN, we perform classification on the learned representations with a comparison of nine multi-view learning methods. The used baselines include CCA [11], KCCA [3], DCCA [2], DCCAE [13], LMSC [14], MvC-DMF [17], BMVC [16], AE^2-Nets [15], and PVC [6]. Note that, SwMC [9] cannot be used in this task since it directly obtains the clustering assignments and does not explicitly learn representations for data. For CCA, KCCA, DCCA, DCCAE, and PVC, we concatenate the obtained representations for classification. For graph-based methods (LMSC and MvC-DMF), we use the spectral representations for classification. For BMVC and AE^2-Nets, we use the common representations for classification. For our MvCLN, we fix the dimensionality of representations to 32. Results with other dimensionalities are shown in Section 2.4.

To achieve classification, we use the SVM classifier contained in the Scikit-Learn package [10] with the default configurations. The representations learned are divided into training and testing sets with different proportions, denoted as $Tr_{train\_ratio}/Te_{test\_ratio}$, where Tr_{train_ratio} indicates the proportion of the training set and Te_{test_ratio} indicates the proportion of the testing set. To avoid randomness due to data partition, we perform the classification 20 times and report the mean classification accuracy.

The results are reported in Table 1, from which one could observe that

- In the partially view-aligned setting, our MvCLN remarkably outperforms all baselines by a considerable margin under different Tr/Te. Particularly, MvCLN achieves an improvement of 26.9% and 12.6% on Caltech101 and Reuters when "Tr/Te" is "8/2", comparing with the best baseline. This further verifies the effectiveness of our MvCLN.

- In the fully view-aligned setting, our MvCLN still

Table 1. Classification performance comparisons on four widely-used multi-view datasets, where the best result for each setting is in bold and "Tr / Te" denotes the size ratio of the training set to testing set. "–" indicates that the method cannot obtain the result due to over-high time or memory cost.

Aligned	Methods	Scene-15			Caltech-101			Reuters			NoisyMNIST		
		8 / 2	5 / 5	2 / 8	8 / 2	5 / 5	2 / 8	8 / 2	5 / 5	2 / 8	8 / 2	5 / 5	2 / 8
Partially	CCA (NeurIPS'03)	**52.49**	**51.52**	**47.95**	35.72	34.56	31.47	64.73	64.70	63.91	65.53	65.05	64.07
	KCCA (JMLR'02)	50.49	48.82	45.75	32.87	31.29	28.90	64.06	63.95	62.83	57.08	56.63	56.06
	DCCA (ICML'13)	51.68	50.64	46.85	35.72	33.97	31.20	**65.92**	**65.80**	**65.15**	60.95	60.90	60.16
	DCCAE (ICML'15)	46.24	45.37	43.75	31.95	30.75	28.14	61.88	61.58	60.67	47.42	47.17	46.26
	LMSC (CVPR'17)	39.15	38.10	36.88	**45.21**	**43.51**	**38.02**	45.03	44.76	44.49	–	–	–
	MvC-DMF (AAAI'17)	36.74	36.42	34.71	20.78	20.08	18.93	41.59	41.34	41.27	33.04	32.64	32.03
	BMVC (TPAMI'18)	50.35	49.83	46.39	33.56	32.83	30.09	64.69	64.20	63.27	72.49	72.03	70.92
	AE^2-Nets (CVPR'19)	48.19	47.64	42.61	23.30	22.65	20.61	62.74	62.40	60.65	**76.58**	**75.87**	**73.75**
Fully	CCA (NeurIPS'03)	57.44	56.21	51.07	37.70	36.14	32.79	69.13	68.67	67.07	87.85	86.09	82.06
	KCCA (JMLR'02)	50.19	50.18	47.26	38.50	36.95	33.72	64.75	64.63	64.63	**97.20**	**97.18**	**97.08**
	DCCA (ICML'13)	63.61	61.72	57.3	38.89	37.23	33.75	71.92	72.33	71.54	96.22	96.34	96.08
	DCCAE (ICML'15)	50.42	48.84	46.48	38.61	37.53	34.03	72.00	71.65	70.63	96.45	96.37	96.08
	LMSC (CVPR'17)	51.28	51.08	48.99	53.92	51.25	42.80	56.09	55.53	54.99	–	–	–
	MvC-DMF (AAAI'17)	43.07	42.45	40.48	48.27	46.71	40.53	42.97	43.08	76.45	75.83	74.05	49.79
	BMVC (TPAMI'18)	66.32	65.16	61.73	**58.57**	**55.69**	**49.92**	**78.65**	**78.20**	**77.73**	92.45	92.47	92.05
	AE^2-Nets (CVPR'19)	**72.03**	**69.76**	**64.66**	35.24	34.38	31.72	65.47	64.82	63.28	89.74	89.33	87.90
Partially	PVC (NeurIPS'20)	48.77	45.97	40.46	36.78	36.50	35.54	72.63	72.08	71.11	93.09	93.12	93.06
	MvCLN (Mean)	**57.93**	**57.15**	**55.52**	**46.69**	**45.89**	**43.87**	**81.77**	**81.63**	**81.11**	**96.19**	**96.18**	**96.15**

achieves competitive results even though the baselines are with ground-truth alignment whereas our method does not. Note that, MvCLN is even better than all the baselines on the Reuters dataset. The possible reason is that the category-level alignment may be more helpful to performance improvement.

Table 2. Clustering performance comparison on the whole NoisyMNIST data. The best result is in bold.

Alignment Type	Method	ACC	NMI	ARI
Partially	MvCLN	**97.50**	**93.09**	**94.57**
Fully	CCA	70.89	52.03	47.91
	KCCA	83.43	88.29	82.59
	DCCA	89.34	91.40	86.87
	DCCAE	89.09	91.37	87.82
	BMVC	91.59	83.48	83.79
	AE^2-Nets	50.83	53.14	40.55

2.3. Clustering on the Whole Dataset

As aforementioned in the manuscript, we carry out all the tested methods on a subset of NoisyMNIST due to the over-high computational cost of the Hungarian algorithm and PVC on the whole dataset. In this section, we carry out our MvCLN on the whole NoisyMNIST in the partially view-aligned setting and conduct some cost-feasible baselines on the same dataset in the fully view-aligned setting for comparison. As shown in Table 2, our MvCLN performs better on the full dataset comparing to the case of the subset, which indicates that more data could do a favor to our method. Besides, MvCLN remarkably outperforms all baselines which are even with fully ground-truth alignment.

2.4. Influence of the Dimensionality

In this section, we investigate the classification performance of representations with different dimensionalities. As shown in Table 3, a higher dimensionality often give better classification performance because more information is contained in the latent space, while giving a high computational complexity. In our implementations, we fix the dimensionality of representations to 32 for all the datasets on the classification task.

2.5. Comparison on the Time Cost

In this section, we quantitatively compare our MvCLN method with the Hungarian algorithm and PVC in terms of the time cost. In the experiments, we employ the package contained in [12] to implement the Hungarian algorithm. As shown in Table 4, our method performs remarkably better than the Hungarian algorithm and PVC in terms of time cost, which verify higher accessibility and scalability of our category-level alignment strategy comparing to the instance-level ones.

2.6. Convergence Analysis

In this section, we investigate the convergence of MvCLN by reporting its loss value, CAR, and clustering performance with the increasing training epoch. From Fig. 1, one could observe that the loss value drops fast in the first 10 epochs, and then slowly decreases until convergence. For

Table 3. Ablation studies on the dimensionality. The best result for each setting is in bold and "Tr / Te" denotes the size ratio of training set to testing set.

Dataset	Dimensionality	8/2	5/5	2/8
Scene-15	10d	47.05	46.60	45.30
	32d	57.93	57.15	55.52
	64d	58.30	57.29	**54.71**
	128d	**58.77**	**57.74**	54.65
Caltech-101	10d	33.78	33.32	32.10
	32d	46.69	45.89	43.87
	64d	**47.17**	**46.45**	**44.01**
	128d	46.97	45.98	43.28
Reuters	10d	75.24	75.10	74.84
	32d	81.77	81.63	81.11
	64d	83.04	82.87	82.32
	128d	**83.94**	**83.88**	**83.19**
NoisyMNIST	10d	95.90	95.89	95.87
	32d	96.19	96.18	96.15
	64d	96.55	96.46	96.41
	128d	**96.58**	**96.62**	**96.55**

Table 4. Time cost comparisons. The best result for each setting is in bold and "–" indicates the method does not involve this phase.

Dataset	Method	training time (s)	inferring time (s)
Scene-15	Hungarian	–	2.69
	PVC	10,907.27	2.01
	MvCLN	**155.53**	**0.72**
Caltech-101	Hungarian	–	48.87
	PVC	11,839.74	7.2
	MvCLN	**388.58**	**1.75**
Reuters	Hungarian	–	289.82
	PVC	18,715.34	30.36
	MvCLN	**790.30**	**3.48**
NoisyMNIST	Hungarian	–	3,778.39
	PVC	53,070.87	34.36
	MvCLN	**1,202.77**	**5.76**

CAR and clustering metrics, they continuously increase in the first 10 epoch and then stay at a high level.

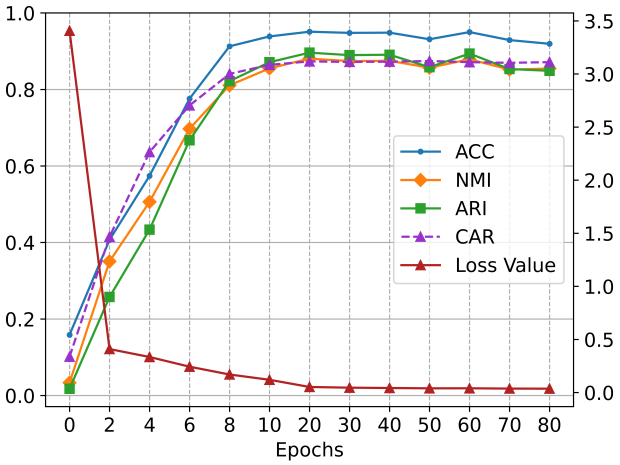

Figure 1. Convergence of MvCLN on the NoisyMNIST dataset. The left and right y-axis denote the performance results and the loss value, respectively.

References

[1] Massih-Reza Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. In *NeruIPS*, pages 28–36, 2009. 1

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013. 1

[3] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002. 1

[4] Dengxin Dai and Luc Van Gool. Ensemble projection for semi-supervised image classification. In *ICCV*, pages 2072–2079, 2013. 1

[5] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, 2005. 1

[6] Zhenyu Huang, Peng Hu, Joey Tianyi Zhou, Jiancheng Lv, and Xi Peng. Partially view-aligned clustering. *NeurIPS*, 33, 2020. 1

[7] Zhenyu Huang, Joey Tianyi Zhou, Xi Peng, Changqing Zhang, Hongyuan Zhu, and Jiancheng Lv. Multi-view spectral clustering network. In *IJCAI*, pages 2563–2569, 2019. 1

[8] F Li Fei-Fei, M Andreetto, MA Ranzato, and P Perona. Caltech101. *Computational Vision Group, California Institute of Technology*, 2003. 1

[9] Feiping Nie, Jing Li, Xuelong Li, et al. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, pages 2564–2570, 2017. 1

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. 1

[11] Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *NeurIPS*, pages 1497–1504, 2003. 1

[12] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020. 2

[13] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, pages 1083–1092, 2015. 1

[14] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. Latent multi-view subspace clustering. In *CVPR*, pages 4279–4287, 2017. 1

[15] Changqing Zhang, Yeqing Liu, and Huazhu Fu. Ae2-nets: Autoencoder in autoencoder networks. In *CVPR*, pages 2577–2585, 2019. 1

[16] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1774–1782, 2018. 1

[17] Handong Zhao and Zhengming Ding. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017. 1