Supplementary Material for COMPLETER: Incomplete Multi-view Clustering via Contrastive Prediction

Yijie Lin[1], Yuanbiao Gou[1], Zitao Liu[2], Boyun Li[1], Jiancheng Lv[1], Xi Peng[1*]
[1] College of Computer Science, Sichuan University, China
[2] TAL Education Group, China
{linyijie.gm,gouyuanbiao,zitao.jerry.liu,liboyun.gm,pengx.gm}@gmail.com; lvjiancheng@scu.edu.cn

1. Introduction

In this supplementary material, we provide additional information including mathematical notations, mathematical derivation of our loss, network architectures, and implementation details. To investigate the effectiveness of our method, we also conduct some additional experimental analysis.

2. Notations and Definitions.

In this section, we summarize the mathematical notations used throughout the manuscript in Table 1 for a clear reference. The bi-view dataset $\bar{\mathbf{X}}$ consists of three parts, *i.e.*, $\bar{\mathbf{X}}^{1,2}$, $\bar{\mathbf{X}}^1$, and $\bar{\mathbf{X}}^2$, where $\bar{\mathbf{X}}^{1,2}$, $\bar{\mathbf{X}}^1$, and $\bar{\mathbf{X}}^2$ denote the examples presented in all views, the first view only, and the second view only, respectively. n and m denote the number of the data points of the whole dataset $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}^{1,2}$, respectively. More specifically, Fig. 1 visually illustrates our setting and some notations by taking a dataset as a showcase.

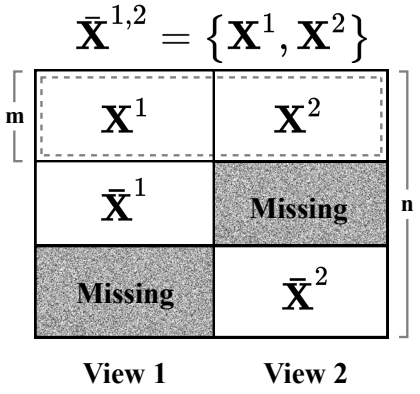

Figure 1. Illustrations of the incomplete bi-view dataset $\bar{\mathbf{X}}$.

Notation	Definition
$\bar{\mathbf{X}}$	Incomplete bi-view dataset where $\bar{\mathbf{X}} = \{\bar{\mathbf{X}}^{1,2}, \bar{\mathbf{X}}^1, \bar{\mathbf{X}}^2\}$.
$\bar{\mathbf{X}}^{1,2}$	Set of examples presented in both views.
$\bar{\mathbf{X}}^v$	Set of examples only presented in view v.
$\mathbf{X}^v$	The v-th view of complete samples $\bar{\mathbf{X}}^{1,2}$, *i.e.*, $\bar{\mathbf{X}}^{1,2} = \{\mathbf{X}^1, \mathbf{X}^2\}$.
$\mathbf{Z}^v$	The representations of $\mathbf{X}^v$.
n	Number of examples presented in $\bar{\mathbf{X}}$.
m	Number of examples presented in $\bar{\mathbf{X}}^{1,2}$.
α	Trade-off parameter of information entropy.
λ_1	Trade-off parameter of dual prediction loss.
λ_2	Trade-off parameter of reconstruction loss
$f^{(v)}$	Encoder of v-th view.
$g^{(v)}$	Decoder of v-th view.
$G^{(i)}$	Predictor which recovers missing representation $\mathbf{Z}^j$ from complete one $\mathbf{Z}^i$.

Table 1. Mathematical notations in the manuscript.

3. Theoretical Derivation on Our Loss

In this section, we elaborate on the derivations of our loss that are omitted in the manuscript due to the space limitation.

3.1. Cross-view Contrastive Loss

We have imported the information entropy to the standard definitions of mutual information in the manuscript. Mathematically, the contrastive loss is defined as,

$$\mathcal{L}_{cl} = -\sum_{t}^{m} \left(I\left(\mathbf{Z}_t^1, \mathbf{Z}_t^2\right) + \alpha \left(H\left(\mathbf{Z}_t^1\right) + H\left(\mathbf{Z}_t^2\right) \right) \right). \quad (1)$$

As illustrated in the manuscript, the representations of t-th sample $\mathbf{Z}_t^1 \in \mathbb{R}^D$ and $\mathbf{Z}_t^2 \in \mathbb{R}^D$ can be interpreted as the distribution of discrete random variables z and z' over D classes, respectively. In other words, the probability distributions $\mathcal{P}(z = d)$ is the d-th element of $\mathbf{Z}_t^1$ where

*Corresponding author

$1 \leq d \leq D$. Hence, considering the case of a data set or batch, the joint probability distribution $\mathcal{P}(z, z')$ of variable z and z' could be defined by $\mathbf{P} \in \mathcal{R}^{D \times D}$, *i.e.*,

$$\mathbf{P} = \frac{1}{m} \sum_{t=1}^{m} \mathbf{Z}_t^1 \left(\mathbf{Z}_t^2\right)^\top. \tag{2}$$

Thus, for discrete distributions, the mutual information and information entropy are given as below:

$$\begin{aligned} I(\mathbf{Z}^1, \mathbf{Z}^2) &= \sum_{z=1}^{D} \sum_{z'=1}^{D} \mathcal{P}(z, z') \log \left(\frac{\mathcal{P}(z, z')}{\mathcal{P}(z)\mathcal{P}(z')} \right) \\ &= \sum_{d=1}^{D} \sum_{d'=1}^{D} \mathbf{P}_{dd'} \ln \frac{\mathbf{P}_{dd'}}{\mathbf{P}_d \cdot \mathbf{P}_{d'}}, \end{aligned} \tag{3}$$

$$\mathrm{H}(\mathbf{Z}^1) = -\sum_{z=1}^{D} \mathcal{P}(z) \log \mathcal{P}(z) = -\sum_{d=1}^{D} \mathbf{P}_d \ln \mathbf{P}_d, \tag{4}$$

$$\mathrm{H}(\mathbf{Z}^2) = -\sum_{z'=1}^{D} \mathcal{P}(z') \log \mathcal{P}(z') = -\sum_{d'=1}^{D} \mathbf{P}_{d'} \ln \mathbf{P}_{d'}, \tag{5}$$

where $\mathbf{P}_d$ and $\mathbf{P}'_d$ denote the marginal probability distributions $\mathcal{P}(z = d)$ and $\mathcal{P}(z' = d')$ which could be obtained by summing over the d-th row and d'-th column of $\mathbf{P}$, respectively. By substituting Eq. (3), Eq. (4), and Eq. (5) into Eq. (1), we could obtain the final form of our cross-view contrastive loss as below:

$$\begin{aligned} \mathcal{L}_{cl} &= -\left(I\left(\mathbf{Z}^1, \mathbf{Z}^2\right) + \alpha\left(H\left(\mathbf{Z}^1\right) + H\left(\mathbf{Z}^2\right)\right)\right) \\ &= -\left(\sum_{d=1}^{D}\sum_{d'=1}^{D} \mathbf{P}_{dd'} \ln \frac{\mathbf{P}_{dd'}}{\mathbf{P}_d \cdot \mathbf{P}_{d'}} - \right. \\ &\qquad \left. \alpha\left(\sum_{d=1}^{D} \mathbf{P}_d \ln \mathbf{P}_d + \sum_{d'=1}^{D} \mathbf{P}_{d'} \ln \mathbf{P}_{d'}\right)\right) \\ &= -\left(\sum_{d=1}^{D}\sum_{d'=1}^{D} \mathbf{P}_{dd'} \ln \frac{\mathbf{P}_{dd'}}{\mathbf{P}_d \cdot \mathbf{P}_{d'}} + \right. \\ &\qquad \left. \alpha\left(\sum_{d=1}^{D}\sum_{d'=1}^{D} \mathbf{P}_{dd'} \ln \frac{1}{\mathbf{P}_d} + \sum_{d'=1}^{D}\sum_{d=1}^{D} \mathbf{P}_{dd'} \ln \frac{1}{\mathbf{P}_{d'}}\right)\right) \\ &= -\sum_{d=1}^{D}\sum_{d'=1}^{D} \mathbf{P}_{dd'} \left(\ln \frac{\mathbf{P}_{dd'}}{\mathbf{P}_d \cdot \mathbf{P}_{d'}} + \alpha\left(\ln \frac{1}{\mathbf{P}_d} + \ln \frac{1}{\mathbf{P}_{d'}}\right)\right) \\ &= -\sum_{d=1}^{D}\sum_{d'=1}^{D} \mathbf{P}_{dd'} \ln \frac{\mathbf{P}_{dd'}}{\mathbf{P}_d^{\alpha+1} \cdot \mathbf{P}_{d'}^{\alpha+1}}. \end{aligned} \tag{6}$$

3.2. Cross-view Dual Prediction Loss

To infer the missing views $\mathbf{Z}^i$ from $\mathbf{Z}^j$, we propose to minimize the conditional entropy $H(\mathbf{Z}^i|\mathbf{Z}^j) = -\mathbb{E}_{\mathcal{P}_{\mathbf{Z}^i,\mathbf{Z}^j}}[\log \mathcal{P}(\mathbf{Z}^i|\mathbf{Z}^j)]$. As it is intractable to solve such a problem, we introduce a variational distribution $\mathcal{Q}(\mathbf{Z}^i|\mathbf{Z}^j)$ and further maximize the lower bound of $\mathbb{E}_{\mathcal{P}_{\mathbf{Z}^i,\mathbf{Z}^j}}[\log \mathcal{P}(\mathbf{Z}^i|\mathbf{Z}^j)]$, *i.e.*, $\mathbb{E}_{\mathcal{P}_{\mathbf{Z}^i,\mathbf{Z}^j}}\left[\log \mathcal{Q}\left(\mathbf{Z}^i \mid \mathbf{Z}^j\right)\right]$. To be specific, we assume $\mathcal{Q}$ as a Gaussian distribution $\mathcal{N}\left(\mathbf{Z}^i \mid G^{(j)}\left(\mathbf{Z}^j\right), \sigma\mathbf{I}\right)$, where $G^{(j)}(\cdot)$ is the predictor which maps $\mathbf{Z}^j$ to $\mathbf{Z}^i$ and $\sigma\mathbf{I}$ is a variance matrix. As a result, we have

$$\begin{aligned} &\max\ \mathbb{E}_{\mathcal{P}_{\mathbf{Z}^i,\mathbf{Z}^j}}\left[\log \mathcal{Q}\left(\mathbf{Z}^i \mid \mathbf{Z}^j\right)\right] = \\ &\qquad \mathbb{E}_{\mathcal{P}_{\mathbf{Z}^i,\mathbf{Z}^j}}\left[\log\left(\frac{1}{\sqrt{\sigma I}\sqrt{2\pi}} e^{-\frac{\left(\mathbf{Z}^i - G^{(j)}\left(\mathbf{Z}^j\right)\right)^2}{2(\sigma\mathbf{I})}}\right)\right], \end{aligned} \tag{7}$$

which could be equivalent to the following optimization problem:

$$\max\ \mathbb{E}_{\mathcal{P}_{\mathbf{Z}^i,\mathbf{Z}^j}}\left[-\frac{\left(\mathbf{Z}^i - G^{(j)}\left(\mathbf{Z}^j\right)\right)^2}{2\sigma\mathbf{I}} + \log \frac{1}{\sqrt{2\pi\sigma I}}\right]. \tag{8}$$

By ignoring the constant $\log \frac{1}{\sqrt{2\pi\sigma I}}$ and scaling factor $2\sigma\mathbf{I}$, we could obtain the prediction loss as below,

$$\max\ -\mathbb{E}_{\mathcal{P}_{\mathbf{Z}^i,\mathbf{Z}^j}} \left\| \left(\mathbf{Z}^i - G^{(j)}\left(\mathbf{Z}^j\right)\right) \right\|_2^2. \tag{9}$$

Hence, for a given bi-view dataset, the dual prediction loss is accordingly defined as

$$\mathcal{L}_{pre} = \left\| G^{(1)}(\mathbf{Z}^1) - \mathbf{Z}^2 \right\|_2^2 + \left\| G^{(2)}(\mathbf{Z}^2) - \mathbf{Z}^1 \right\|_2^2. \tag{10}$$

4. Experiment Details

In this section, we elaborate on the implementation details of our method and the experimental settings.

4.1. Network Architectures of COMPLETER

The proposed method contains two training modules, *i.e.*, view-specific auto-encoders and cross-view prediction networks. Table 2 and 3 have presented the details of the network architectures in these two training modules, respectively. For these two modules, we simply adopt a dense (*i.e.*, fully-connected) network where each layer is followed by a batch normalization layer and a ReLU layer. The softmax activation function is used at the last layer of the encoders and prediction modules. The structures of autoencoders $f^{(\cdot)}$, $g^{(\cdot)}$, and predictors $G^{(\cdot)}$ for different views are the same. Specifically, the size of the output of the encoder and predictor should be the same and set to 64 or 128 according to the dataset.

4.2. Implementation Details for Clustering

To perform clustering, we adopt k-means to compute the cluster assignments on the common representation which is

Table 2. The architecture of the autoencoders in COMPLETER.

Dataset	Encoder	Decoder
Caltech101-20	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (Softmax, size = **128**)	Dense (BatchNorm1d, ReLU, size = input)
LandUse-21	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (Softmax, size = **64**)	Dense (BatchNorm1d, ReLU, size = input)
Scene-15	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (Softmax, size = **128**)	Dense (BatchNorm1d, ReLU, size = input)
Noisy MNIST	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (BatchNorm1d, ReLU, size = 1024)	Dense (BatchNorm1d, ReLU, size = 1024)
	Dense (Softmax, size = **64**)	Dense (BatchNorm1d, ReLU, size = input)

Table 3. The architecture of dual prediction in COMPLETER.

Structure
Dense (BatchNorm1d, ReLU, size = 128)
Dense (BatchNorm1d, ReLU, size = 256)
Dense (BatchNorm1d, ReLU, size = 128)
Dense (BatchNorm1d, ReLU, size = 256)
Dense (BatchNorm1d, ReLU, size = 128)
Dense (Softmax, size = input)

Table 4. Influence of dimensionality.

Dataset	Dimension	ACC	NMI	ARI
Caltech101-20	32	43.48	60.31	41.50
	64	51.99	62.88	47.91
	128	68.44	**67.39**	**75.44**
	256	**69.56**	65.63	74.54
Scene-15	32	37.30	40.79	21.41
	64	37.60	41.01	20.55
	128	**39.50**	**42.35**	**23.51**
	256	36.37	41.87	22.10

obtained by simply concatenating all view-specific representations together. Specifically, we use the k-means contained in the Scikit-Learn package [3] with the default configuration. For a fair comparison, we run all the used methods five times with different initializations and data partitions to obtain the common representations. For each run, we further conduct k-means 10 times on the representation to obtain the clustering results. In all experiments, we adopt three evaluation metrics implemented by Scikit-Learn to evaluate the clustering performance, namely, ACC, NMI, and ARI.

5. Additional Experiments

This section presents two experimental studies including: i) the influence of dimensionality of latent representations and ii) clustering performance on the full datasets.

5.1. Influence of Dimensionality

In the proposed method, we treat each element of the representation as an over-cluster class probability like [1, 2, 4]. To evaluate the effectiveness of such over-clustering strategy, we change the dimensionality of the representation in the range of $\{32, 64, 128, 256\}$. The missing rate η is fixed to 0.5 and the results are shown in Table 4. The results demonstrate that a too large or too small dimensionality will cause performance degradation. The former is of insufficient representability and the latter may have some redundant information.

5.2. Experiment on the Full Datasets

In the main body of the manuscript, we only report the results on the 20k subsets of the Noisy MNIST dataset because most of the baselines are inefficient to handle large scale datasets. In this evaluation, we carry out clustering experiments on the whole Noisy MNIST dataset and report the results compared with scalable methods including DCCA [5], DCCAE [5], BMVC [7], and AE^{2}Nets [6]. Similarly, we also test these methods in both *Incomplete* ($\eta = 0.5$) and *Complete* ($\eta = 0$) settings. As shown in Table 5, COMPLETER still outperforms all baselines.

Table 5. Performance comparisons on full Noisy MNIST.

Missing Type	Method	ACC	NMI	ARI
Incomplete	DCCA	45.32	48.73	25.70
	DCCAE	49.44	48.49	25.31
	AE^{2}Nets	37.76	35.53	20.57
	BMVC	46.42	36.23	22.34
	COMPLETER	**94.28**	**87.39**	**88.12**
Complete	DCCA	89.29	91.35	87.04
	DCCAE	89.03	91.40	87.77
	AE^{2}Nets	50.70	53.26	40.49
	BMVC	91.57	83.55	83.83
	COMPLETER	**97.17**	**94.19**	**93.58**

References

[1] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *CVPR*, pages 8849–8858, 2020. 3

[2] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, pages 9865–9874, 2019. 3

[3] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[4] Xi Peng, Hongyuan Zhu, Jiashi Feng, Chunhua Shen, Haixian Zhang, and Joey Tianyi Zhou. Deep clustering with sample-assignment invariance prior. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4857–4868, 2020. 3

[5] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, pages 1083–1092, 2015. 3

[6] Changqing Zhang, Yeqing Liu, and Huazhu Fu. Ae2-nets: Autoencoder in autoencoder networks. In *CVPR*, pages 2577–2585, 2019. 3

[7] 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, 2019. 3