Hierarchical Consensus Network for Multiview Feature Learning

Chengwei Xia[1], **Chaoxi Niu**[2], **Kun Zhan**[1]*

[1]School of Information Science and Engineering, Lanzhou University
[2] Australian Artificial Intelligence Institute, University of Technology Sydney
kzhan@lzu.edu.cn

Abstract

Multiview feature learning aims to learn discriminative features by integrating the distinct information in each view. However, most existing methods still face significant challenges in learning view-consistency features, which are crucial for effective multiview learning. Motivated by the theories of CCA and contrastive learning in multiview feature learning, we propose the hierarchical consensus network (HCN) in this paper. The HCN derives three consensus indices for capturing the hierarchical consensus across views, which are classifying consensus, coding consensus, and global consensus, respectively. Specifically, classifying consensus reinforces class-level correspondence between views from a CCA perspective, while coding consensus closely resembles contrastive learning and reflects contrastive comparison of individual instances. Global consensus aims to extract consensus information from two perspectives simultaneously. By enforcing the hierarchical consensus, the information within each view is better integrated to obtain more comprehensive and discriminative features. The extensive experimental results obtained on four multiview datasets demonstrate that the proposed method significantly outperforms several state-of-the-art methods.

1 Introduction

Multiview data are universal in the real world and typically contain multiple views of the same underlying semantic information. Generally, a single view cannot provide adequate information for feature learning. To address this, multiview feature learning aims to integrate the common and complementary information contained in multiple views to generate more discriminative data features than a single view (Chen et al. 2023; Xu et al. 2022).

Multiview feature learning methods are roughly divided into two categories, i.e., traditional and deep methods. However, traditional methods typically suffer from limited representation capacity, and high computational complexity for complex data scenarios. To alleviate these problems, many works propose to perform multiview feature learning based on deep neural networks and achieve superior performance compared to the traditional methods.

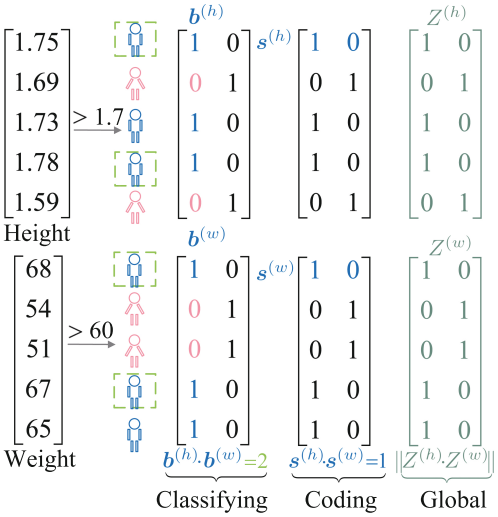


Figure 1: The hierarchical consensus: Consider a scenario with five students, where we collect their height and weight. We employ two simple classifiers: one based on height (assigning 'boy' if height >1.7m and 'girl'; otherwise) and another based on weight (assigning 'boy' if weight >60kg and 'girl' otherwise). Each view produces binary predictions. Our hierarchical consensus objective involves deriving consensus indices through inner product computations. (1) **Classifying Consensus**: In the first column of the matrix, $\boldsymbol{b}^{(h)}$ and $\boldsymbol{b}^{(w)}$ denote boys' predictions from the two views. The result of 2 counts the number of boys. The objective of classifying consensus is to align predictions for gender quantity. (2) **Coding Consensus**: Moving to the first row of the second matrix, $\boldsymbol{s}^{(h)}$ and $\boldsymbol{s}^{(w)}$ represent the gender coding for a same student. The result of 1 indicates consensus prediction of the student. The objective of coding consensus is to align gender coding of the student. (3) **Global Consensus**: To the whole matrix perspective, a global consensus is established.

Most deep learning methods focus on learning consistency between different view features to obtain a common representation through concatenation or adaptive weighted fusion (Trosten et al. 2021; Yang et al. 2023; Xu et al. 2024b). Canonical Correlation Analysis (CCA) (Hotelling 1936) and co-training (Blum and Mitchell 1998) are two representa-

*Corresponding Author.

tive methods in consistency learning, achieving promising results in exploring common features across views (Andrew et al. 2013; Wang et al. 2015; Lin et al. 2023). As a promising unsupervised representation learning method, contrastive learning has also gained increasing attention (Chen et al. 2020; He et al. 2020), and several multiview feature learning methods based on contrastive learning have been proposed, e.g., (Trosten et al. 2021) directly conducts alignment-based instance-level contrast across multiple views. (Chen et al. 2023; Xu et al. 2022) rely on additional components such as an MLP projector to obtain a cluster assignment probability, and achieve consistency by contrasting the cluster assignments across views. Moreover, (Yan et al. 2023; Xu et al. 2024a) propose a selection of negative pairs and reweighting strategies to improve contrastive learning performance under multiview scenarios, resulting in additional costs.

In this paper, building on the insight of CCA and contrastive learning in multiview feature learning, we propose the hierarchical consensus network (HCN) which derives three consensus indices to explore hierarchical consensus between views, i.e., classifying, coding, and global consensus. In Figure 1, our motivation is clearly and intuitively presented, making it easy to see the connections among HCN, CCA (Hotelling 1936), and contrastive learning. The objective of classifying consensus is conceptually similar to CCA, while the objective of coding consensus closely resembles contrastive learning. The objective of global consensus aims to capture consensus information from two perspectives simultaneously. Specifically, classifying consensus aims to reinforce class-level correspondence between views. Coding consensus is proposed to achieve contrastive comparison of individual instances between different views. Global consensus minimizes the difference between the different views. Overall, HCN fully characterizes the consensus between views from different perspectives.

To apply the proposed HCN for multiview learning, we employ a view-specific autoencoder to learn the distinct and common information within each view. Besides, we further apply the augmentation technique to increase the training samples for each view. Original and augmented data are fed into the view-specific encoder to obtain latent features. Then, hierarchical consensus learning is conducted on the latent features of the original and the augmented data across views. Specifically, for classifying consensus learning, we minimize the conditional entropy of the class probability in one view conditioned by the other view. For the coding consensus, we employ a weak-to-strong pseudo-supervision cross-entropy between the original and the augmented data in each view. For the global consensus, we minimize the difference between the two latent features. Compared to state-of-the-art (SOTA) methods, HCN explores consensus information in multiview feature learning from novel insights, with a lower complexity, and does not require the introduction of additional components or sample selection and weighting strategies.

The main contributions are summarized as follows: ❶ We introduce hierarchical consensus to explore multiple consistencies between views, providing promising insights into multiview feature learning. ❷ We design the Hierarchical Consensus Network (HCN) for multiview feature learning. HCN effectively learns a comprehensive and discriminative feature by capturing hierarchical consensus among multiple views. ❸ Experiments on four multiview datasets demonstrate the effectiveness of HCN over other SOTA baselines.

2 Related Work

Most multiview feature learning methods suffer from drawbacks such as high complexity and limited performance (Chen et al. 2021; Zhan et al. 2018; Wang et al. 2019; Liu et al. 2020). Recently, several consistency-based multiview feature learning methods have been proposed, aiming to maximize consistency between different views. Inspired by the strategy to maximize view consistency between two sets in CCA (Hotelling 1936), (Andrew et al. 2013) maps multiview features into a common space and concatenates the low-dimensional features as the common representation. (Wang et al. 2015) further introduce autoencoders in multiview feature learning compared with (Andrew et al. 2013). Besides, by leveraging co-training strategy (Blum and Mitchell 1998), (Lin et al. 2023) design dual contrastive prediction to learn the cross-view consistency.

As one of the most effective consistent learning paradigms, contrastive learning has achieved SOTA performance (Chen et al. 2020; He et al. 2020). The basic idea of contrastive learning is learning a feature space from raw data by maximizing the similarity between positive pairs while minimizing that of negative pairs. In recent, some methods have shown the success of contrastive learning in multiview feature learning (Trosten et al. 2021; Xu et al. 2022), where similarities of positive pairs are maximized and that of negative pairs are minimized via NT-Xent (Chen et al. 2020). (Trosten et al. 2021) learns common representation by aligning representations from different views at the sample level. (Yan et al. 2023) learns the global structural relationships between samples and utilizes them to obtain consistent data representations. Simultaneously, structural information is utilized to select negative pairs for cross-view contrastive learning. In (Xu et al. 2024a), the weights are optimized based on the discrepancy between pairwise representations, performing self-weighted contrastive learning. Considering consistency between the cluster assignments among multiple views, (Chen et al. 2023) proposes a cross-view contrastive learning method to learn view-invariant representations by contrasting cluster assignments among multiple views. Moreover, contrastive clustering (Li et al. 2021) designed for single-view clustering tasks, constructs two distinct views through data augmentation and subsequently projects them into feature space. Using two separate projection heads, the method conducts contrastive learning at different levels in the row and column space to jointly learn representations and cluster assignments. (Xu et al. 2022) conducts multi-level features contrast from multiple views to achieve consistency. Furthermore, some recent contrastive learning works, notably BYOL (Grill et al. 2020) and SimSiam (Chen and He 2021), have shown the remarkable ability to learn powerful representations using only positive pairs, which has proven to be a simple and effective method (Tian, Chen, and Ganguli 2021).

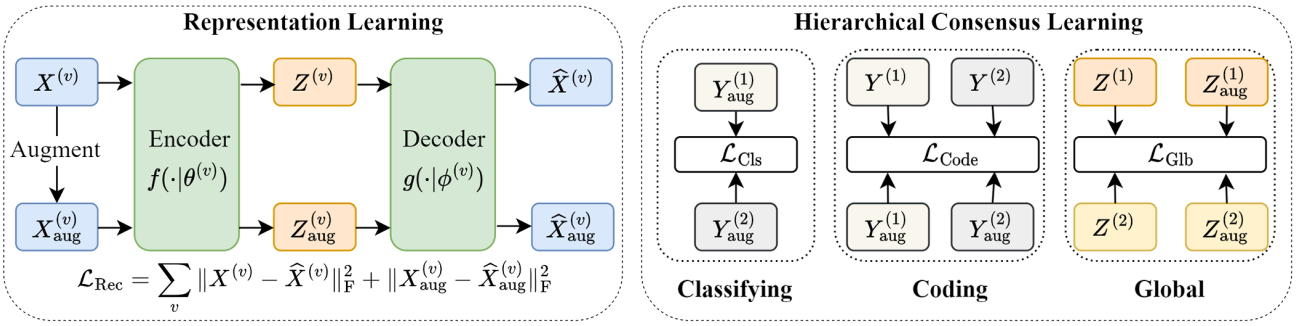


Figure 2: The HCN framework. Each view contains a view-specific autoencoder, i.e., an encoder $f(\cdot|\theta^{(v)})$ and decoder $g(\cdot|\phi^{(v)})$. The representations $Z^{(v)}$ and $Z^{(v)}_{\text{aug}}$ are learned by minimizing the construction error $\mathcal{L}_{\text{Rec}}$. Besides, $Z^{(v)}$ and $Z^{(v)}_{\text{aug}}$ are fed into the softmax to obtain the class posterior probabilities, i.e., $Y^{(v)}$ and $Y^{(v)}_{\text{aug}}$. Given the representations and class probabilities of the two views, HCN aims to capture hierarchical consensus between them. Specifically, classifying consensus ensures the consistency of class distributions across views, coding consensus enhances the same prediction for the same sample, and global consensus minimizes the difference between learned representations from different views.

3 Hierarchical Consensus Learning

We incorporate a hierarchical structure by defining objectives at the matrix, row, and column levels: maximizing the global consensus between the matrices of dual-view learned features aligns the overall representations; maximizing the pairwise coding consensus between rows enables the contrastive comparison of individual instances; and maximizing the classifying consensus between columns aligns class-level representations, similar CCA.

3.1 Notations

A multiview dataset $\mathcal{X} = \{X^{(1)}, X^{(2)}, \ldots, X^{(n_v)}\}$ typically consists of n samples across n_v views. Specifically, the data of v-th view is represented as $X^{(v)} = [\boldsymbol{x}_1^{(v)}, \boldsymbol{x}_2^{(v)}, \ldots, \boldsymbol{x}_n^{(v)}]^\top \in \mathbb{R}^{n \times d_v}$, where $\boldsymbol{x}^{(v)}$ denotes the samples with d_v dimensional raw feature. Since each view contains distinct information about the dataset, we use $f(\cdot|\theta^{(v)})$ to denote nonlinear mappings implemented by the encoder for view v, with the corresponding parameters $\theta^{(v)}$. Let $Z^{(v)} = [\boldsymbol{z}_1^{(v)}, \boldsymbol{z}_2^{(v)}, \ldots, \boldsymbol{z}_n^{(v)}]^\top \in \mathbb{R}^{n \times k}$ denote the learned feature from the input in view v by $f(X^{(v)}|\theta^{(v)})$ where k denotes the feature dimension. We aim to capture hierarchical consensus between the learned features from multiple views in an unsupervised way and generate a more comprehensive and discriminative feature for $\mathcal{X}$.

3.2 Classifying Consensus Learning

We begin with classifying consensus learning between $Z^{(1)}$ and $Z^{(2)}$ which aims to enhance the consistency of class distribution between views. Taking the first view as an example, we model the class posterior probabilities $y_{ij}^{(1)}$ or $p(c_j^{(1)}|\boldsymbol{x}_i^{(1)})$ of sample $\boldsymbol{x}_i^{(1)}$ belonging to the j-th class by applying the softmax layer on $Z^{(1)}$: $y_{ij}^{(1)} = p(c_j^{(1)}|\boldsymbol{x}_i^{(1)}) = \frac{\exp(z_{ij}^{(1)})}{\sum_j \exp(z_{ij}^{(1)})}$ where $z_{ij}^{(1)}$ represents the j-th entry of the i-th sample in $Z^{(1)}$.

In the same way, we obtain $y_{ij}^{(2)}$ or $p(c_j^{(2)}|\boldsymbol{x}_i^{(2)})$ based on $Z^{(2)}$ for the second view.

Referring to the similarity kernel function defined in (Haussler 1999) and (Bishop 2006), the joint probability of sample belonging to distinct classes is given by

$$\widetilde{p}(c_j, c_h) = \int p(c_j|\boldsymbol{x}_i)p(c_h|\boldsymbol{x}_i)p(\boldsymbol{x}_i)\mathrm{d}\boldsymbol{x}_i \tag{1}$$

where c_j and c_h denote different classes. Similar to the classifying consensus shown in Figure 1, we define classifying consensus probability. By extending Eq. (1) to the discrete scenario and multiview learning, the joint class probability of discrete random variables between two different views (Ji, Henriques, and Vedaldi 2019; Lin et al. 2021) is defined by

$$\widetilde{p}(c_j^{(1)}, c_h^{(2)}) = \frac{1}{n}\sum_{i=1}^{n} y_{ij}^{(1)} y_{ih}^{(2)}, \tag{2}$$

Eq. (2) performs a column-wise inner product between cross-view matrices $Y^{(1)} = [y_{ij}^{(1)}]$ and $Y^{(2)} = [y_{ij}^{(2)}]$ to measure similarity, similar to CCA. This measure is normalized by

$$p(c_j^{(1)}, c_h^{(2)}) = \frac{\widetilde{p}(c_j^{(1)}, c_h^{(2)})}{\sum_{j,h=1}^{k} \widetilde{p}(c_j^{(1)}, c_h^{(2)})}. \tag{3}$$

Based on the obtained cross-view joint probability $p(c_j^{(1)}, c_h^{(2)})$, it is straightforward to derive $p(\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)})$, $p(\boldsymbol{c}^{(1)})$, $p(\boldsymbol{c}^{(2)})$, $p(\boldsymbol{c}^{(2)}|\boldsymbol{c}^{(1)})$, and $p(\boldsymbol{c}^{(1)}|\boldsymbol{c}^{(2)})$.

For the classifying consensus, the softmax operation ensures independence between the prediction columns, satisfying the decorrelation constraint in CCA. Similar to the CCA objective of maximizing cross-view correlations, classifying consensus aims to align the predictions w.r.t. the same class between two views. To achieve this goal, we minimize the conditional entropy of the class probability of one view conditioned by that of the other view. In other words, if the classes of data points in one view are known, the uncertainty about the classes of corresponding data points in the other view is

minimized by the conditional entropy. Then, the classifying consensus is formulated as the conditional entropy between two views:

$$\min \mathrm{H}(\boldsymbol{c}^{(1)}|\boldsymbol{c}^{(2)}) + \mathrm{H}(\boldsymbol{c}^{(2)}|\boldsymbol{c}^{(1)}) \tag{4}$$

where $\mathrm{H}(\cdot)$ denotes the entropy operator. However, directly optimizing Eq. (4) induces the risk that all samples are assigned to a particular class. To address this issue, we introduce a regularization term to encourage the class distribution to be balanced in each view. Specifically, we propose to maximize the entropy of the class probability in each view as

$$\max \mathrm{H}(\boldsymbol{c}^{(1)}) + \mathrm{H}(\boldsymbol{c}^{(2)}) \,. \tag{5}$$

By combining Eqs. (4) and (5), we obtain the final objective for classifying consensus learning:

$$\mathcal{L}_{\mathrm{cls}} = \alpha \mathrm{H}(\boldsymbol{c}^{(1)}|\boldsymbol{c}^{(2)}) - \beta \mathrm{H}(\boldsymbol{c}^{(1)}) - \gamma \mathrm{H}(\boldsymbol{c}^{(2)}) \tag{6}$$

where α, β and γ are trade-off hyperparameters. In this way, we achieve the classifying consensus between views and avoid assigning all samples to a particular class.

3.3 Coding Consensus Learning

Different views characterize the same sample from different perspectives with the semantics of the sample shared between views. Based on this, we further propose coding consensus learning. Formally, given the class posterior probability obtained from softmax layer, we employ the cross-entropy for coding consensus learning (Han et al. 2024):

$$\mathcal{L}_{\mathrm{code}} = -\sum_{i=1}^{n} \big((\boldsymbol{y}_i^{(2)})^\top \log \boldsymbol{y}_i^{(1)} + (\boldsymbol{y}_i^{(1)})^\top \log \boldsymbol{y}_i^{(2)} \big) \tag{7}$$

where $\boldsymbol{y}$ is the softmax prediction $\boldsymbol{x}$.

3.4 Global Consensus Learning

In addition to classifying and coding consensus learning based on the posterior probabilities of data classes, global consensus learning aims to minimize the difference between the two latent features encoded from different views. Thus, the objective of global consensus is given by:

$$\mathcal{L}_{\mathrm{glb}} = -\mathrm{tr}\Big(\big(Z^{(1)}\big)^\top Z^{(2)} \Big) \,. \tag{8}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operation.

3.5 Hierarchical Structure

Eq. (4) defines the column-wise classifying consensus learning objective, corresponding to the column vectors shown in Figure 1. Eq. (7) represents a row-wise contrastive comparison of individual instances, as indicated by the row vectors in Figure 1. Lastly, Eq. (8) maximizes the global consensus, aligning overall representations between views, illustrated by the entire matrix in Figure 1.

The classifying consensus aligns column vectors from different views, similar to CCA, which maximizes cross-view correlations. This correlation maximization is achieved in Eq. (2). Additionally, the decorrelation constraint in each view, typical of CCA, is incorporated here through the softmax operation, which enforces independence across columns in each view. Thus, classifying consensus closely aligns with the principles of CCA.

By optimizing the coding consensus objective, we ensure that the coding assignment of sample i remains consistent across views, in line with the definition of multiview data. The connection between coding consensus learning and contrastive learning is formally established in Theorem 1.

Theorem 1. *Coding-consensus learning is equivalent to contrastive learning with positive pairs.*

The proofs and further corollaries are in Appendix B.2.

As shown in Figure 1, if we denote the learned matrix as $Z = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k] = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n]^\top$, we have the relationship:

$$\mathrm{tr}\big((Z^{(1)})^\top Z^{(2)} \big) = \sum_{j=1}^{k} (\boldsymbol{b}_j^{(1)})^\top \boldsymbol{b}_j^{(2)} = \sum_{i=1}^{n} (\boldsymbol{s}_i^{(1)})^\top \boldsymbol{s}_i^{(2)} \tag{9}$$

where we focus on the inner products, and then argue that the first term of Eq. (9) operates at the whole matrix level, capturing global alignment; the second term aligns cross-view columns, reinforcing class-level correspondence; and the third term reflects the effect of instance-wise row contrastive comparison, focusing on individual sample alignment. Viewed from another perspective, the global consensus simultaneously captures both the coding and classifying effects, as shown in Eq. (9).

4 Hierarchical Consensus Network

The general framework is shown in Figure 2. Given multiview data, a view-specific autoencoder is employed to exploit the distinct information (Lin et al. 2021) within each view. Specifically, we denote the encoder as $f(X^{(v)}|\theta^{(v)})$ that takes the data $X^{(v)}$ as input and output the latent feature $Z^{(v)}$ of view v. Inversely, the decoder $g(Z^{(v)}|\phi^{(v)})$ aims to reconstruct the input data based on the latent feature, denoted as $\widehat{X}^{(v)}$. We implement the encoder with a four-layer MLP followed by a softmax. The decoders have the same architecture as the encoders. Then, a reconstruction objective is employed to learn the semantic information of the input.

For each view, drop-feature augmentation is applied to the input data $X^{(v)}$ to enrich to input data of view v, which is implemented as randomly dropping certain feature dimensions. Formally, we first sample a random vector $\boldsymbol{m} \in \{0,1\}^{d_v}$ with each entry being drawn from a Bernoulli distribution independently, i.e., $m_j \sim \mathrm{Bern}(1-\rho)$, where m_j is the j-th entry of $\boldsymbol{m}$ and ρ is the drop rate. Therefore, the augmented data of view v is represented as $X_{\mathrm{aug}}^{(v)} = X^{(v)}[:, \mathbb{I}(\boldsymbol{m})]$ where $\mathbb{I}(\cdot)$ indicates whether each entry of $\boldsymbol{m}$ is 1 or 0. The latent feature $Z_{\mathrm{aug}}^{(v)}$ of $X_{\mathrm{aug}}^{(v)}$ is also obtained from the view-specific encoder, i.e., $Z_{\mathrm{aug}}^{(v)} = f(X_{\mathrm{aug}}^{(v)}|\theta^{(v)})$ and is used to reconstruct the augmented data.

Specifically, the reconstruction objective is defined by:

$$\mathcal{L}_{\mathrm{Rec}} = \sum_{v=1}^{n_v} \big(\|X^{(v)} - \widehat{X}^{(v)}\|_{\mathrm{F}}^2 + \|X_{\mathrm{aug}}^{(v)} - \widehat{X}_{\mathrm{aug}}^{(v)}\|_{\mathrm{F}}^2 \big) \,. \tag{10}$$

Given the learned latent features $Z^{(1)}$, $Z^{(2)}$, $Z^{(1)}_{\text{aug}}$, and $Z^{(2)}_{\text{aug}}$, we obtain the corresponding class probabilities by applying the softmax operation on them, i.e., $Y^{(1)}$, $Y^{(2)}$, $Y^{(1)}_{\text{aug}}$ and $Y^{(2)}_{\text{aug}}$. Then, we detail how to conduct consensus learning between and within the two views.

First, we perform the classifying consensus learning between the augmented data from two views. Formally, the joint class probability $p(\boldsymbol{c}^{(1)}_{\text{aug}}|\boldsymbol{c}^{(2)}_{\text{aug}})$ is obtained by Eq. (3), and the marginal probability distributions $p(\boldsymbol{c}^{(1)}_{\text{aug}})$ and $p(\boldsymbol{c}^{(2)}_{\text{aug}})$ are calculated by summing along rows and columns of the joint probability matrix respectively. Then, the classifying consensus objective is given by:

$$\mathcal{L}_{\text{Cls}} = \sum_{\substack{u,v=1 \\ u>v}}^{n_v} \alpha \text{H}(\boldsymbol{c}^{(u)}_{\text{aug}}|\boldsymbol{c}^{(v)}_{\text{aug}}) - \beta \text{H}(\boldsymbol{c}^{(u)}_{\text{aug}}) - \gamma \text{H}(\boldsymbol{c}^{(v)}_{\text{aug}}). \quad (11)$$

Second, within each view, we conduct coding consensus learning between the augmented data and the original data. The mechanism behind this is similar to contrastive learning which pulls the positive samples closer. Formally, inspired by weak-to-strong pseudo supervision in semi-supervised learning (Sohn et al. 2020; Xu et al. 2024c), we transform the posterior class probabilities Y of the original data into hard labels $\hat{T} = [\hat{t}_{ij}]$. $\hat{t}^{(v)}_{ij}$ denotes pseudolabel of the i-th data point belonging to the j-th class in the v-th view.

The coding consensus is defined by a weak-to-strong pseudo-supervision loss (Xu et al. 2024c):

$$\mathcal{L}_{\text{Code}} = -\sum_{v=1}^{n_v} \sum_{i=1}^{n} \left(\hat{\boldsymbol{t}}^{(v)}_i\right)^\top \log \boldsymbol{y}^{(v)}_{i,\text{aug}} \quad (12)$$

where $\hat{\boldsymbol{t}}^{(v)}_i$ is the one-hot pseudolabel and $\boldsymbol{y}^{(v)}_{i,\text{aug}}$ represent the i-th row of $Y^{(v)}_{\text{aug}}$. By enforcing the feature of the augmented data consistent with that of the raw data, Eq. (12) requires the encoder to learn the invariant semantics of each view.

Finally, we propose the global consensus to capture global alignment between two views by minimizing the difference between the latent features. The objective is defined by:

$$\mathcal{L}_{\text{Glb}} = -\sum_{\substack{u,v=1 \\ u>v}}^{n_v} \text{tr}\Big(\left(Z^{(u)}\right)^\top Z^{(v)} + \left(Z^{(u)}_{\text{aug}}\right)^\top Z^{(v)}_{\text{aug}}\Big). \quad (13)$$

Overall, by minimizing the above inter-view objective, the model is optimized to generate similar features for the same samples between different views. The overall loss objective of the proposed method is given by

$$\mathcal{L} = \mathcal{L}_{\text{Rec}} + \mathcal{L}_{\text{Cls}} + \lambda_1 \mathcal{L}_{\text{Code}} + \lambda_2 \mathcal{L}_{\text{Glb}} \quad (14)$$

where λ_1 and λ_2 denote trade-off hyperparameters.

After the model optimization, the latent embeddings outputted by the encoder of each view are concatenated to obtain the final features for downstream tasks, i.e.,

$$Z = \left[Z^{(1)}, Z^{(2)}, \ldots, Z^{(n_v)}\right]. \quad (15)$$

The proposed method is summarized in Algorithm 1.

Algorithm 1: The HCN algorithm.

Input: Multiview dataset $\mathcal{X} = \{X^{(v)}\}_v^{n_v}$, drop rate ρ, hyperparameters λ_1, λ_2, α, β, and γ.

Initialization: E and $\{\theta^{(v)}, \phi^{(v)}, \forall v \in \{1, 2, \ldots, n_v\}\}$.

while $e < E$ **do**
  for mini-batch samples in $\mathcal{X}$ **do**
    Obtain $X^{(v)}_{\text{aug}}$ for each view v;
    Compute $Z^{(v)}$, $Z^{(v)}_{\text{aug}}$, $Y^{(v)}$ and $Y^{(v)}_{\text{aug}}$ for each view by the view-specific encoder and the softmax layer;
    Calculate $\mathcal{L}_{\text{Rec}}$ and $\mathcal{L}_{\text{Cls}}$ by Eqs. (10) and (11);
    Calculate $\mathcal{L}_{\text{Code}}$ and $\mathcal{L}_{\text{Glb}}$ by Eqs. (12) and (13);
    Update $\{\theta^{(v)}, \phi^{(v)}, \forall\, v\}$ by Eq. (14);
    $e = e + 1$;
  end for
end while

Output: Fused feature: $Z = \left[Z^{(1)}, Z^{(2)}, \ldots, Z^{(n_v)}\right]$.

4.1 Computational Complexity

The complexity of HCN is $\mathcal{O}(nn_v^2k^2b + 2nn_vd_vh^{(l+1)})$ where b, h, l, n, and k denote the mini-batch size, the maximum number of hidden layers, the layer number, the number of samples, and the feature dimension, respectively. The derivation process is presented in Appendix B.3.

5 Experiments

5.1 Experimental Setup

Datasets. We conduct multiview clustering on four widely used datasets to evaluate the proposed HCN, i.e., (1) LandUse-21 (Yang and Newsam 2010) contains 2,100 satellite images across 21 categories, using PHOG and LBP features as two views; (2) Caltech101-20 (Fei-Fei, Fergus, and Perona 2004) includes 2,386 RGB images from 20 subjects, with HOG and GIST features as two views; (3) Scene-15 (Fei-Fei and Perona 2005) comprises 4,485 images from 15 scene categories, utilizing PHOG and GIST as two views; (4) Noisy MNIST (Wang et al. 2015) uses the original 70k MNIST images as one view and within-class images with white Gaussian noise as the second view. In three views experiments, we use the HOG, GIST, and LBP features. For the Scene-15 and LandUse-21 datasets for the Caltech101-20 dataset, we use the PHOG, LBP, and GIST features.

Comparison Methods. We compare HCN with the following SOTA methods on multiview clustering. The comparison includes Spectral clustering (Ng, Jordan, and Weiss 2001), traditional multiview clustering methods such as BMVC (Zhang et al. 2018), PIC (Wang et al. 2019) and EERIMVC (Liu et al. 2020), based on CCA or CCA-related methods DCCA (Andrew et al. 2013), DCCAE (Wang et al. 2015) and AE^{2}Nets) (Zhang, Liu, and Fu 2019). Besides, contrastive learning methods are also included, i.e., DCP (Lin et al. 2023), MFLVC (Xu et al. 2022), CVCL (Chen et al. 2023), GCFAggMVC (Yan et al. 2023), DealMVC (Yang et al. 2023), and SEM (Xu et al. 2024a). Finally, a more recent method, MVCAN (Xu et al. 2024b) is employed, which considers the case of noisy views based on deep embedding

Method	LandUse-21			Caltech101-20			Scene-15			Noisy MNIST		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SC_{Agg}[NeurIPS01]	24.69	30.10	10.23	48.78	60.98	34.68	35.26	35.92	20.20	44.10	40.51	27.16
DCCA [ICML13]	15.51	23.15	4.43	41.89	59.14	33.39	36.18	38.92	20.87	85.53	89.44	81.87
DCCAE [ICML15]	15.62	24.41	4.42	44.05	59.12	34.56	36.44	39.78	21.47	81.60	84.69	70.87
BMVC[TPAMI19]	25.34	28.56	11.39	42.55	63.63	32.33	40.50	41.20	24.11	81.27	76.12	71.55
PIC[IJCAI20]	24.86	29.74	10.48	62.27	67.93	51.56	38.72	40.46	22.12	-	-	-
AE^2Nets [CVPR19]	24.79	30.36	10.35	49.10	65.38	35.66	36.10	40.39	22.08	56.98	46.83	36.98
EERIMVC[TPAMI20]	24.92	29.57	12.24	43.28	55.04	30.42	39.60	38.99	22.06	65.47	57.69	49.54
MFLVC [CVPR22]	23.67	27.50	11.27	49.25	41.40	45.63	41.49	42.28	24.41	96.91	92.44	93.36
CVCL [ICCV23]	25.40	29.59	11.78	34.77	59.93	25.70	38.43	39.58	22.53	97.87	94.18	**97.87**
DCP [TPAMI23]	26.23	30.65	13.70	70.18	68.06	76.88	41.07	45.11	24.78	82.78	84.86	74.83
GCFAgg [CVPR23]	28.06	32.44	14.40	34.12	53.20	23.16	39.72	41.37	23.01	91.44	86.56	83.16
DealMVC[MM23]	10.41	7.11	1.69	39.56	56.91	36.04	38.96	42.26	24.21	32.57	28.12	13.72
SEM[NeurIPS23]	30.02	34.75	15.93	37.33	59.95	28.44	40.53	42.48	25.04	60.04	64.69	43.15
MVCAN [CVPR24]	23.94	29.57	10.70	48.63	66.80	44.85	41.54	44.38	25.63	78.46	79.01	70.51
HCN (ours)	**32.81**	**38.58**	**17.86**	**77.39**	**74.64**	**88.70**	**46.05**	**45.56**	**28.54**	**98.07**	**94.83**	95.79

Table 1: The clustering results with two views on four datasets. "-" indicates unavailable results due to the out-of-memory issue. The best and the second result are **bold** and underlined respectively.

Method	LandUse-21			Caltech101-20			Scene-15		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SC_{Agg}[NeurIPS01]	26.44	32.73	11.91	48.00	60.40	33.82	34.05	34.83	19.73
MFLVC [CVPR22]	22.38	27.23	9.91	53.43	39.33	44.06	40.15	41.41	24.11
CVCL [ICCV23]	23.47	27.50	10.61	36.93	56.70	26.24	44.59	42.17	24.11
DCP [TPAMI23]	26.66	32.74	13.50	70.58	**69.59**	**76.93**	41.81	**45.23**	25.84
GCFAgg [CVPR23]	24.95	29.06	10.68	35.24	56.04	24.62	44.14	43.40	23.99
DealMVC[MM23]	12.04	9.39	2.74	37.76	47.40	34.35	40.02	42.99	24.16
SEM[NeurIPS23]	25.12	30.12	11.55	37.34	62.51	28.66	42.45	41.25	26.67
MVCAN [CVPR24]	26.18	32.53	12.57	50.04	66.04	44.85	42.07	44.38	25.57
HCN (ours)	**33.30**	**38.55**	**18.82**	**71.35**	68.61	73.25	**45.20**	44.52	**28.18**

Table 2: The multiview clustering results on three datasets. The best and the second result are **bold** and underlined, respectively.

HCN w/o	LandUse-21			Noisy MNIST		
	ACC	NMI	ARI	ACC	NMI	ARI
$\mathcal{L}_{\text{Rec}}$	32.27	37.82	17.23	97.93	94.60	95.51
$\mathcal{L}_{\text{Cls}}$	24.57	26.22	10.11	25.62	24.58	9.79
$\mathcal{L}_{\text{Glb}}$	32.21	38.02	17.73	97.30	93.41	94.20
$\mathcal{L}_{\text{Code}}$	32.62	38.19	17.79	97.82	94.44	95.29
DA	31.95	38.06	17.63	97.19	93.11	93.96
HCN	**32.81**	**38.58**	**17.86**	**98.07**	**94.83**	**95.79**

Table 3: Ablation study of each component in HCN.

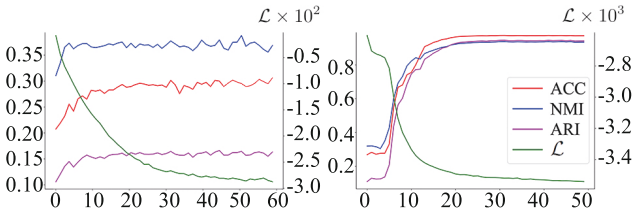
clustering (Xie, Girshick, and Farhadi 2016).

Implementation Details. We select the best performance for each experiment and report the average performance of five runs with different seeds. k-means clustering algorithm is utilized to obtain the clustering results. To have a fair comparison, all the methods use the same views on each dataset, and use three views in multiview setting. In addition, following the DCP (Lin et al. 2023), we only use a subset of Noisy MNIST consisting of 10k validation images and 10k testing images in the experiments. More details are in Appendix C.1.

Evaluation Metrics. To evaluate the clustering for our HCN, three widely used clustering metrics are employed, i.e., Normalized Mutual Information(NMI), Accuracy (ACC), and Adjusted Rand Index (ARI). For these metrics, a higher numerical value represents a better clustering.

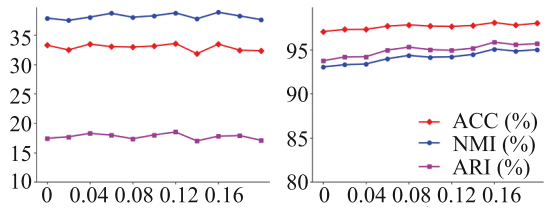
5.2 Experimental Results

The clustering results on four datasets are shown in Tables 1 and 2, where Table 1 reports the performance with two views and Table 2 presents the results of three views. From the tables, we have the following observations: (1) Our approach significantly outperforms SOTA baselines on nearly all settings. Remarkably, in Table 1, our approach demonstrates outstanding performance on the Caltech101-20 and Scene-15 datasets, significantly outperforming the best comparison



(a) LandUse-21 (b) Noisy MNIST

Figure 3: Convergence and clustering performance of HCN with increasing epoch on LandUse-21 and Noisy MNIST.



(a) LandUse-21 (b) Noisy MNIST

Figure 4: Parameter sensitivity of drop rate ρ.

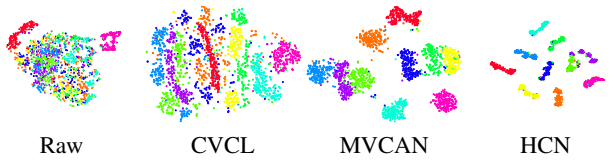
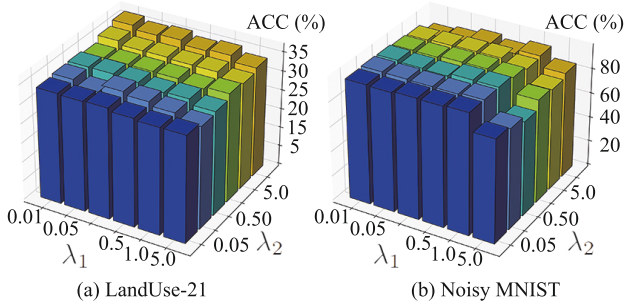


Figure 5: Visualizations on Noisy MNIST with baselines.



(a) LandUse-21 (b) Noisy MNIST

Figure 6: Parameter sensitivity of λ_1 and λ_2.

method in ACC and NMI. (2) Compared to contrastive multiview clustering methods, the proposed approach achieves SOTA results. This is attributed to our hierarchical consensus network capturing consensus at different levels, resulting in more comprehensive and discriminative features.

5.3 Ablation Study

To validate the importance of each component in HCN, we conducted ablation studies by discarding each component. The results are shown in Table 3, where DA means we apply classifying consensus objective without data augmentation. It can be observed that classifying consensus learning plays a more vital role in multiview feature learning than others as the clustering performance is significantly improved by adding classifying consensus objective. The coding consensus and the global consensus objective also improve clustering performance. Furthermore, Table 3 also demonstrates that data augmentation helps learn more robust, invariant, and discriminative features of the multiview data.

5.4 Model Analysis

Convergence Analysis. We investigate the convergence of the proposed method. As illustrated in Figure 3, we provide the dynamics of loss value and three clustering metrics with increasing epochs on the LandUse-21 and Noisy MNIST datasets. From Figure 3, we see that the loss value decreases rapidly in a few iterations until convergence is achieved. Meanwhile, the clustering performance metrics quickly increase and stabilize after several epochs. This demonstrates the promising convergence property of HCN.

Analysis of Hyperparameters. Without loss of generality, we investigate the sensitivity of the proposed method w.r.t the trade-off hyperparameters λ_1 and λ_2 on the Noisy MNIST and LandUse-21 datasets, where λ_1 and λ_2 range from $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$. As shown in Figure 6, the accuracy values achieve relatively stable with most combinations of λ_1 and λ_2. In addition, Figure 4 presents the sensitivity of HCN w.r.t the drop rate ρ which varies in a range of [0, 0.2] with 0.02 intervals. The results show that the results are relatively stable with ρ varying within a wide range.

5.5 Visualization

In Figure 5, we provide the t-SNE visualizations of our approach and other baselines on Noisy MNIST. Specifically, we visualize the fused features learned by the encoder of the proposed HCN. From the figure, we observe that raw features are non-discriminative at the initial stage. After training with HCN, the learned features become more discriminative among different classes and each class becomes more compact compared to baselines, demonstrating the effectiveness of HCN for multiview feature learning. The visualizations of other comparison methods are in Appendix C.3.

6 Conclusion

In this paper, we explore hierarchical consensus learning for multiview feature learning and introduce three consensus indices across views, i.e., classifying, coding, and global consensus, providing promising insights into multiview feature learning. Specifically, classifying consensus explores consistency between views from a CCA perspective, while the coding consensus closely resembles contrastive learning. Global consensus simultaneously captures both the coding and classifying effects. Based on these, we propose HCN for multiview feature learning. HCN effectively captures hierarchical consensus between different views from different perspectives, resulting in more comprehensive and discriminative features. Extensive experiments and ablation studies on four multiview datasets validate the superiority and effectiveness of our HCN method.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62176108, Natural Science Foundation of Qinghai Province of China under No. 2022-ZJ-929, Science Foundation of National Archives Administration of China under No. 2024-B-006, and Supercomputing Center of Lanzhou University.

References

Andrew, G.; Arora, R.; Bilmes, J.; and Livescu, K. 2013. Deep canonical correlation analysis. In *ICML*, 1247–1255.

Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, 92–100.

Chen, J.; Mao, H.; Woo, W. L.; and Peng, X. 2023. Deep multiview clustering by contrasting cluster assignments. In *ICCV*, 16752–16761.

Chen, J.; Yang, S.; Mao, H.; and Fahy, C. 2021. Multiview subspace clustering using low-rank representation. *TCyb*, 52(11): 12364–12378.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*, 15750–15758.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR workshop*, 178–178.

Fei-Fei, L.; and Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, 524–531.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, volume 33, 21271–21284.

Han, Q.; Tian, Z.; Xia, C.; and Zhan, K. 2024. InfoMatch: Entropy neural estimation for semi-supervised image classification. In *IJCAI*, volume 33, 4089–4097.

Haussler, D. 1999. Convolution kernels on discrete structures. Technical report.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika*.

Ji, X.; Henriques, J. F.; and Vedaldi, A. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 9865–9874.

Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J. T.; and Peng, X. 2021. Contrastive clustering. In *AAAI*, volume 35, 8547–8555.

Lin, Y.; Gou, Y.; Liu, X.; Bai, J.; Lv, J.; and Peng, X. 2023. Dual contrastive prediction for incomplete multi-view representation learning. *TPAMI*, 45(4): 4447–4461.

Lin, Y.; Gou, Y.; Liu, Z.; Li, B.; Lv, J.; and Peng, X. 2021. COMPLETER: Incomplete multi-view clustering via contrastive prediction. In *CVPR*, 11174–11183.

Liu, X.; Li, M.; Tang, C.; Xia, J.; Xiong, J.; Liu, L.; Kloft, M.; and Zhu, E. 2020. Efficient and effective regularized incomplete multi-view clustering. *TPAMI*, 43(8): 2634–2646.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, volume 14.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, volume 33, 596–608.

Tian, Y.; Chen, X.; and Ganguli, S. 2021. Understanding self-supervised learning dynamics without contrastive pairs. In *ICML*, 10268–10278.

Trosten, D. J.; Lokse, S.; Jenssen, R.; and Kampffmeyer, M. 2021. Reconsidering representation alignment for multi-view clustering. In *CVPR*, 1255–1265.

Wang, H.; Zong, L.; Liu, B.; Yang, Y.; and Zhou, W. 2019. Spectral Perturbation Meets Incomplete Multi-view Data. In *IJCAI*, 3677–3683.

Wang, W.; Arora, R.; Livescu, K.; and Bilmes, J. 2015. On deep multi-view representation learning. In *ICML*, 1083–1092.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *ICML*, 478–487.

Xu, J.; Chen, S.; Ren, Y.; Shi, X.; Shen, H.; Niu, G.; and Zhu, X. 2024a. Self-weighted contrastive learning among multiple views for mitigating representation degeneration. In *NeurIPS*, volume 36.

Xu, J.; Ren, Y.; Wang, X.; Feng, L.; Zhang, Z.; Niu, G.; and Zhu, X. 2024b. Investigating and mitigating the side effects of noisy views for self-supervised clustering algorithms in practical multi-view scenarios. In *CVPR*, 22957–22966.

Xu, J.; Tang, H.; Ren, Y.; Peng, L.; Zhu, X.; and He, L. 2022. Multi-Level feature learning for contrastive multi-View clustering. In *CVPR*, 16051–16060.

Xu, S.; Zhang, X.; Zhang, P.; and Zhan, K. 2024c. Structure-Aware Consensus Network on Graphs with Few Labeled Nodes. *arXiv*.

Yan, W.; Zhang, Y.; Lv, C.; Tang, C.; Yue, G.; Liao, L.; and Lin, W. 2023. GCFAgg: Global and cross-view feature aggregation for multi-view clustering. In *CVPR*, 19863–19872.

Yang, X.; Jiaqi, J.; Wang, S.; Liang, K.; Liu, Y.; Wen, Y.; Liu, S.; Zhou, S.; Liu, X.; and Zhu, E. 2023. Dealmvc: Dual contrastive calibration for multi-view clustering. In *ACM Multimedia*, 337–346.

Yang, Y.; and Newsam, S. 2010. Bag-of-visual-words and spatial extensions for land-use classification. In *SIGSPATIAL*.

Zhan, K.; Nie, F.; Wang, J.; and Yang, Y. 2018. Multiview consensus graph clustering. *TIP*, 28(3): 1261–1270.

Zhang, C.; Liu, Y.; and Fu, H. 2019. AE^2-Nets: Autoencoder in autoencoder networks. In *CVPR*, 2577–2585.

Zhang, Z.; Liu, L.; Shen, F.; Shen, H. T.; and Shao, L. 2018. Binary multi-view clustering. *TPAMI*, 41(7): 1774–1782.