nmODE-MVC: Neural Memory ODE-Based Multi-View Clustering

1st Yan Zheng
School of Computer Science and Engineering,
University of Electronic Science and Technology of China
Chengdu, China
yan9zheng9@gmail.com

2nd Song Wu
School of Computer Science and Engineering,
University of Electronic Science and Technology of China
Chengdu, China
songwu.work@outlook.com

3rd Yazhou Ren*
School of Computer Science and Engineering,
Shenzhen Institute for Advanced Study,
University of Electronic Science and Technology of China
yazhou.ren@uestc.edu.cn

4th Xiaorong Pu*
School of Computer Science and Engineering,
Shenzhen Institute for Advanced Study,
University of Electronic Science and Technology of China
puxiaor@uestc.edu.cn

5th Zenglin Xu
School of Computer Science and Technology,
Harbin Institute of Technology Shenzhen
Shenzhen, China
zenglin@hit.edu.cn

6th Lifang He
Department of Computer Science and Engineering,
Lehigh University
Bethlehem, USA
lih319@lehigh.edu

7th Zhang Yi
Intelligent Interdisciplinary Research Center and College
of Computer Science, Sichuan University
Chengdu, China
zhangyi@scu.edu.cn

2023 7th Asian Conference on Artificial Intelligence Technology (ACAIT) | 979-8-3503-5914-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ACAIT60137.2023.10528630

***Abstract*—Deep multi-view clustering (MVC) has recently gained significant interest for its capability to harness complementary information across multiple views through deep neural networks, enhancing clustering performance. However, existing deep MVC methods still have two issues: (1) They separate the feature learning objective and the clustering objective, potentially leading to suboptimal embedding features due to the neglect of the global clustering structure. (2) Most works utilize the target distribution obtained by the clustering objective to supervise feature learning. However, they simply sharpen the soft clustering distribution to obtain the target distribution, lacking the dynamics of the clustering process. To address these issues, in this paper we propose a novel end-to-end self-supervised multi-view clustering approach based on NeuralODEs, namely nmODE-MVC. Our approach operates in two stages: In the first stage, all view-embedded features are extracted, and then unified them into a global feature, facilitating global clustering assignments by exploiting multi-view data's complementarity. The second stage leverages the dynamic attributes of NeuralODEs to dynamically refine the data features and generate a target distribution that actively drives the clustering decisions. We formulate these learning processes as a unified optimization problem, facilitating iterative training and refinement. The effectiveness of nmODE-MVC is validated through extensive experiments on multiple real-world multi-view datasets.**

***Index Terms*—Deep Clustering, Multi-View Clustering, Neural Ordinary Differential Equation, Self-Supervised Learning**

*Corresponding author

I. Introduction

Clustering is a fundamental task in unsupervised learning, which can find extensive applications in diverse areas. With the advent of deep learning technologies, researchers have gradually focused on applying deep models to improve clustering performance. Specifically, classical methods such as Deep Embedded Clustering (DEC) [1] and Structural Deep Clustering Network (SDCN) [2] have shown promise in leveraging the capabilities of deep neural networks for clustering task. Despite the advancements in deep clustering, traditional deep learning models often entail discretized data propagation within the network. This approach can inhibit the model's ability to naturally encapsulate continuous processes and introduce discrepancies between the learned and accurate data distributions. To address these issues, Neural Ordinary Differential Equation (NeuralODE) [3] can elegantly mirror the continuous dynamics of real-world processes and enhance model flexibility. This continuous modeling potentially facilitates better representation learning for the clustering task.

In the real world, data often come with multiple views. The principle of multi-view clustering (MVC) is rooted in the notions of consistency and complementarity. Well-known MVC algorithms such as Simple Multi-View Clustering (SiMVC) [4], Contrastive Multi-View clustering (CoMVC) [4] and Self-Supervised Discriminative Feature Learning for Deep Multi-View Clustering (SDMVC) [5], have demonstrated efficacy in leveraging these principles. However, while these MVC methods have found success, they typically rely on static frameworks that lack the adaptability to dynamically changing data distributions. NeuralODEs [3], [6]–[8] bring a dynamic perspective into play, allowing for network parameters to change according to input data, rather than remaining static. Unlike traditional deep neural networks where each layer is associated with a fixed set of weights and biases, in NeuralODEs, all parameters exist within a dynamical system that evolves over time. This design grants neural networks enhanced adaptability, enabling them to better handle complex, changing input data. This dynamic trait of NeuralODEs can be particularly beneficial for multi-view clustering, as it could dynamically adjust to different views' characteristics and the complementary information they offer.

Based on this, we make the following two suggestions: First, the continuous data flow inherent in NeuralODEs aligns well with the notion of seamless information propagation across different views, fostering a cohesive representation space. Second, the dynamism offered by NeuralODEs can potentially adjust and recalibrate the clustering scheme in response to the idiosyncrasies of each view and the supplementary information they introduce. To test these conjectures, we proposed **n**eural **m**emory **ODE**-based **M**ulti-**V**iew **C**lustering (nmODE-MVC), an end-to-end self-supervised multi-view clustering approach based on NeuralODEs.

Building upon the work of [5], we first utilize an autoencoder to extract the salient features for each view, and concatenate the embedding features across different views, thereby forming an aggregated global feature representation. Then, the K-means algorithm is employed to perform a clustering process on these global embeddings, which yields pseudo-labels. Subsequently, inspired by [6], we leverage the nmODE model. The global embedded features derived from clustering are used as inputs for the nmODE model, and the pseudo-labels supervise model learning more discriminative target distributions. This integration enables nmODE-MVC to exhibit explicit dynamism, allowing for a global convergence trajectory from the input to the target distribution. Specifically, our proposed framework alternate between training the autoencoder and the nmODE model for obtaining more reliable the target distribution. Finally, through a carefully designed set of experiments, we have corroborated our theoretical claims and demonstrated the efficacy of our nmODE-MVC. The main contributions of this work are summarized as follows:

- We have achieved seamless information propagation across different views to establish a unified representation space, effectively integrating multi-view clustering into a dynamic system while capitalizing on the continuous data flow.
- Our method can adapt and fine-tune the target distribution in our clustering strategy to suit the unique characteristics of each view and the complementary information it contributes.
- Our approach has been rigorously tested through extensive experiments on multiple real-world multi-view datasets. The results validate the efficacy of our proposed method, showcasing its superiority in comparison to several state-of-the-art methods.

II. Related Work

A. Multi-View Clustering

Traditional clustering algorithms, limited by simple distance metrics, tend to be ineffective when the input dimensionality is high (e.g., K-means [9], spectral clustering [10]). To this end, deep clustering methods [1], [11]–[13] aim to employ the superior representation learning capabilities of deep models (e.g., autoencoders (AEs) [14] and variational autoencoders (VAEs) [15]) to learn clustering-oriented low-dimensional representations from complicated high-dimensional data. These deep methods focus on performing end-to-end clustering tasks, where the clustering objective and embedded feature representations are jointly optimized. One of the most representative works is DEC [1] which designs a clustering objective inspired by t-SNE [16], and jointly learns the clustering assignments and embedded features of autoencoders. After then, improved DEC [11] introduces a trade-off between the clustering objective and reconstruction objective to further optimize the multi-objective learning process. Moreover, some studies have attempted to incorporate the learning paradigm of deep clustering into other clustering representation tasks. Cai et al. [12] propose an efficient deep embedded subspace clustering (EDESC) aiming to learn the subspace bases from deep representation in an iterative refining manner while the refined subspace bases in return improve the representation learning of the deep models. EDESC achieves linear time and space complexity. Yang et al. [17] attempt to incorporate graph information that captures local data structures into a deep Gaussian mixture model (GMM), and combine them facilitates the deep network to learn powerful representations for upstream clustering task.

The aforementioned deep clustering methods cater exclusively to single-view data. However, in practical clustering tasks, the input data often possess multiple views. Multi-view clustering methods [18]–[23] propose exploiting the complementary information among multiple views to enhance clustering performance. Deep multi-view clustering algorithms can be categorized into three types, based on their foundational clustering theory: DEC-based, subspace clustering-based, and GNN-based [24]. Xu et al. [18] propose a pioneering co-training framework for Deep Embedded Multi-View Clustering (DEMVC). This framework optimizes the reconstruction loss by defining a switching shared auxiliary target distribution, thereby preserving the variations across multiple views. To

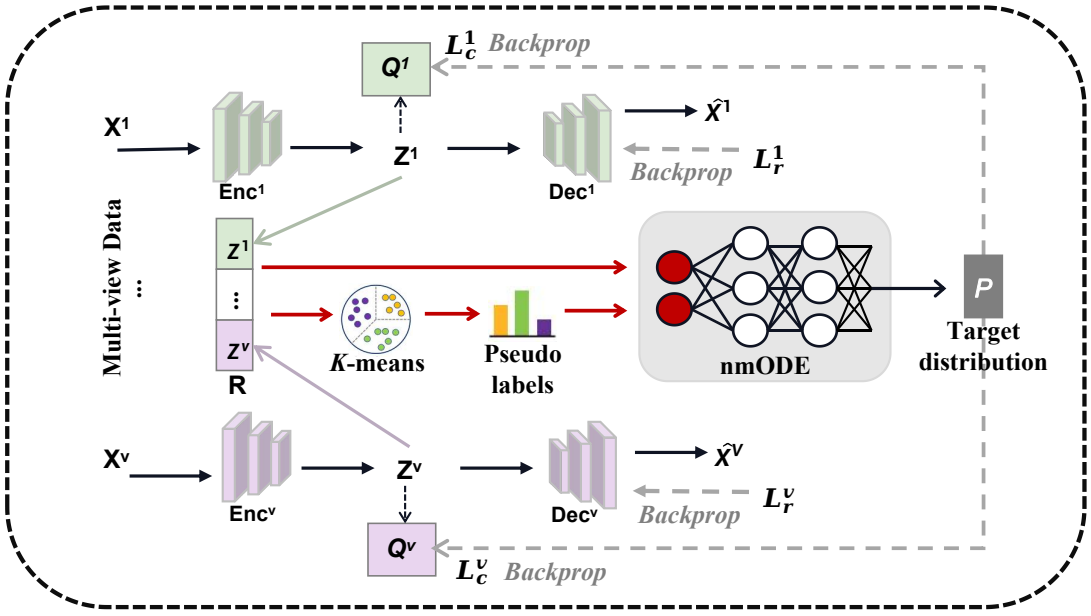


Fig. 1. nmODE-MVC, designed to dynamically update and adapt to previously learned internal data features, facilitates seamless cross-view information learning through the alternate training of the autoencoder and nmODE. Within the process, the feature concatenation is illustrated by dashed arrows from the embedded features, represented by green for the first view (Z^1) and purple for the v-th view (Z^v), towards the global feature $\boldsymbol{R}$. The subsequent process, where the global feature $\boldsymbol{R}$ and the soft clustering pseudo-labels are jointly introduced into the nmODE module, is indicated by red dashed lines. Then we utilize the dynamic properties of the nmODE module to generate the target distribution $\boldsymbol{P}$. The reconstruction loss is specifically designed on view-internal representations Z^v, while the clustering loss is constructed to utilize the target distribution $\boldsymbol{P}$ for optimizing the individual view's soft clustering Q^v.

amalgamate subspace learning methods with the latest breakthroughs in graph convolution networks, Muhammad et al. [20] introduce a Graph-based Convolution Network (Multi-GCN). It presents an effective strategy for adapting Graph-Based Semi-Supervised Learning (GSSL) to a multi-view context, bridging the gap between conventional learning methods and multi-view data processing.

B. NeuralODE

Within the realm of traditional deep learning models, data propagation through the network transpires discretely. Contrasting this established model, Chen et al. [3] pioneer the concept of NeuralODEs, interpret t as continuous and the dynamics of the hidden state $h(t)$ are depicted by an ODE system formulated as

$$\frac{dh(t)}{dt} = f(h(t), t; \omega), \tag{1}$$

where f is a neural network defined by the parameters ω. A significant merit of NeuralODEs rests on their inherent continuity, a property enabling amplified precision in computation and backpropagation, resulting in enhanced storage efficiency. From this conceptual cornerstone, a surge of NeuralODE models [6]–[8] has emerged, along with a suite of diverse task extensions founded on NeuralODEs [25]–[27].

For instance, Zhang et al. [7] propose a strategic advancement of the NeuralODE model, extending it to a coupled ODE system. This system permits the parameters of the model to evolve and activate dynamically over time. Furthermore, Zhang et al. [25] unveil the Self-Attention ODE Solver, a model adept at learning continuous hidden states endowed with positional information, while simultaneously training global representation matrices with high parameter efficiency. Another practical implementation is by Niemeyer et al. [26] and Jiang et al. [27], where NeuralODEs are trained to reconstruct temporally deformed 3D objects. More recently, Zhang [6] postulates a theory centered on a memory-based NeuralODE, which distinguishes learning neurons from memory neurons, thereby elucidating its dynamic behavior. The nmODE can establish a nonlinear mapping from the external input to its corresponding attractor, thereby eliminating recurring issues of learning features homeomorphic to the input data space found in most existing NeuralODEs.

The novel intersection of NeuralODEs and clustering has remained relatively unexplored. Up until now, only one piece of literature has addressed this emerging area of study. The NODE-EDM [28] unveils a methodology for performing NeuralODE evolutionary subspace clustering on time-series data. This approach enables the learned NODE to function as a universal solver for affinity matrices C_t, of any sequence X_t, originated from similar video contexts. This cross-context generalization exhibits promising potential for efficient and robust analysis of time-series data in complex video sequences.

Despite the significant potential, the integration of NeuralODEs with clustering is a domain that remains largely untapped. The scarcity of work done in this area highlights the wealth of opportunities that lie ahead for further exploration. The convergence of these two fields has the potential to create transformative solutions to pressing problems in machine

learning and data science, catalyzing innovation in a breadth of applications.

III. Proposed Method

In this section, we start by clearly explaining the research problem we are focusing on. Then, we give a thorough description of how neural memory Ordinary Differential Equation (nmODE [6]) and multi-view clustering work, both of which play a central role in our approach. Finally, we explain in detail how alternate training operates within our end-to-end nmODE-based multi-view clustering framework. The overview architecture of the proposed method is shown in Figure 1.

A. Problem Formulation

Multi-view clustering aims to categorize the instances into K distinct clusters. We consider a multi-view dataset $\left\{\boldsymbol{x}_i^1 \in \mathbb{R}^{D_1}, \boldsymbol{x}_i^2 \in \mathbb{R}^{D_2}, \ldots, \boldsymbol{x}_i^V \in \mathbb{R}^{D_V}\right\}_{i=1}^N$, which signifies the data from V views. N is the number of examples, and $D_1, D_2, \ldots, D_V$ are the dimensionalities of the views.

B. Definitions

Incorporating a nonlinear dynamical system into clustering tasks invites the conceptions:

• **Attractor**. If the embedded features beginning from a certain initial state, eventually converge towards a specific state or a set of states, such a state or set of states is termed the clustering task's attractor. This concept is of considerable importance within nonlinear dynamical systems, as it mirrors the system's stability characteristics after a prolonged period of evolution.

• **Global Attractor**. The basin of an attractor consists of all initial states that will eventually enter this attractor. To put it into perspective, imagine a water basin where all drops eventually gather at the lowest point; similarly, all initial states in the basin will eventually gravitate toward the attractor. If a basin encompasses the entire clustering space, then this attractor is referred to as a global attractor. The existence of global attractor does not preclude the presence of local attractors. Within the framework, multiple local attractors may be formed. These local attractors can represent the typical characteristics of each view.

Within the nmODE-MVC paradigm, attractors serve as a stable portrayal of clustering decision results. To illustrate, in a multi-view clustering model's latent space, each view's data may eventually gravitate towards a specific attractor. These attractors symbolize the typical characteristics of each view, aiding in distinguishing different categories. The presence of a global attractor does not imply an immediate convergence of all data. It is a dynamic process, during which data may form transient clusters. These temporary clusters can provide valuable information about the inherent structure of the data.

C. Multi-View Feature Learning Encoder

Due to the natural heterogeneity of multi-view datasets, our first step embeds distinct data from each view into a lower-dimensional space. This process aims to reduce dimensionality and noise, and to extract meaningful, representative information. We implement it by learning view-specific encoder and decoder, denoted as $f_{\theta^v}^v$ and $g_{\phi^v}^v$ respectively. For each view, the unique features and information inherent to each perspective can be effectively captured. Specifically, θ^v and ϕ^v are the parameters of the encoder and decoder pairs for each view, and these non-linear mappings translate the high-dimensional input data into a more compact and informative low-dimensional feature representation as:

$$\boldsymbol{z}_i^v = f_{\theta^v}^v\left(\boldsymbol{x}_i^v\right), \tag{2}$$

where $\boldsymbol{z}_i^v \in \mathbb{R}^{d_v}$ denotes as the salient embedding of $\boldsymbol{x}_i^v$ in the d_v-dimensional feature space. Further, the decoder $g_{\phi^v}^v$ reconstructs the salient embedding as $\hat{x}_i^v \in \mathbb{R}^{D_v}$ by decoding the low-dimensional feature representation z_i^v:

$$\hat{\boldsymbol{x}}_i^v = g_{\phi^v}^v\left(\boldsymbol{z}_i^v\right). \tag{3}$$

Here, the reconstruction objective $\mathcal{L}_r^v$ is implemented by forcing the decoded output to be consistent with the original input, which can be expressed as:

$$\mathcal{L}_r^v = \sum_{i=1}^N \left\|\boldsymbol{x}_i^v - g_{\phi^v}^v\left(f_{\theta^v}^v\left(\boldsymbol{x}_i^v\right)\right)\right\|_2^2. \tag{4}$$

Considering that clustering objectives can effectively enhance salient representation learning, we firstly transform the feature distribution into a soft clustering distribution. Inspired by student t-distribution [29], the probability of the i-th sample being a part of the j-th cluster is as follows:

$$Q_{ij}^v = \frac{(1 + \|\boldsymbol{z}_i^v - \boldsymbol{\mu}_j^v\|^2)^{-1}}{\sum_j (1 + \|\boldsymbol{z}_i^v - \boldsymbol{\mu}_j^v\|^2)^{-1}}, \tag{5}$$

where μ_j^v denotes the learnable cluster centroids, and we initialize them by K-means. In subsequent processes, Q_{ij}^v will be guided by the target distribution $\boldsymbol{P}$, derived by nmODE in Eq. (12), which will be introduced in Section III-D in detail.

Since single view data generally lack global information, to increase confidence of the clustering structure, we concatenate all the embedded features from different views to form a comprehensive global feature $\boldsymbol{R}$ for the final downstream clustering task:

$$\boldsymbol{r}_i = \left[\boldsymbol{z}_i^1, \boldsymbol{z}_i^2, \ldots, \boldsymbol{z}_i^V\right] \in \mathbb{R}^{\sum_{v=1}^V d_v}. \tag{6}$$

We denote $\boldsymbol{R}$ as $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N\}$. The K-means is applied to $\boldsymbol{R}$ as it effectively computes the cluster centroids, each denoted by $\boldsymbol{c}_j$, representing the nucleus of the j-th cluster in our data:

$$\min_{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K} \sum_{i=1}^N \sum_{j=1}^K \left\|\boldsymbol{r}_i - \boldsymbol{c}_j\right\|^2. \tag{7}$$

Similarly, we compute the global cluster assignment for the global feature $\boldsymbol{R}$, and further transform it into the pseudo-label s_i, which will guide nmODE to dynamically learn

latent clustering distribution. Then, the pseudo-label s_i can be computed by the following equation:

$$l_{ij} = \frac{\left(1 + \|\boldsymbol{z}_i - c_j\|^2\right)^{-1}}{\sum_j \left(1 + \|\boldsymbol{z}_i - c_j\|^2\right)^{-1}}, \tag{8}$$

$$s_i = \arg\max_j \left(l_{ij}\right). \tag{9}$$

Acting as proxies for the true data labels, these pseudo-labels are incorporated into the supervisory information for the nmODE's alternating training scheme. This setup allows for the facilitation of iterative learning, where the model dynamically adjusts and refines its label predictions based on insights gained from previous outputs.

D. nmODE Determines the Clustering Result

In conventional deep learning models, the propagation of data through the network occurs in a discrete fashion. However, NeuralODE models adopt a different approach, where data transmission is treated as a continuous process. Building upon the work of nmODE [6], we conceptualize the mapping from x to g as:

$$\dot{g} = -g + \sin^2\left[g + W^{(1)}x + b\right], \tag{10}$$

where $g \in R^n, x \in R^m, b \in R^n, W^{(1)} = \left(w_{ij}^{(1)}\right)_{n\times m} \in R^{n\times m}$ and $\dot{g}$ is the global attractor. Nevertheless, directly solving Eq. (10) to facilitate the mapping is commonly untenable due to computational limitations and the inherent complexity of the function. Consequently, the nmODE defines the mathematical model for each one-dimensional ODE using the representation in:

$$\dot{p} = -p + \sin^2(p + \omega), \tag{11}$$

where $p \in R^1$ and ω denote the perceptual input of the neuron. This form of a one-dimensional ODE coined as neuronODE, forms the building blocks of the nmODE. This formulation allows the model to break down the complex mapping problem into simpler and more manageable subproblems, each handled by a separate ODE, hence offering a more computationally feasible approach.

The pseudo-labels s_i of the clustering results and the global embedded features $\boldsymbol{R}$ are used as input of the nmODE, every memory neuron i independently yields output $g_i(t)$ in the t-th iteration of training, coalescing to form the overall pattern $g(t)$. Then, we delineate as:

$$\begin{cases} h_i(t) = \sum_{j=1}^{n} w_{ij}^{(2)} g_j(t) \\ \boldsymbol{P} = softmax\left(h_i(t)\right) \end{cases}, \tag{12}$$

where $W^{(2)} = \left(w_{ij}^{(2)}\right)_{r\times n}$ represents an additional learning parameter set. NeuronODEs signifying output target distribution $\boldsymbol{P}$ are identified as decision neurons. The intention behind this decision objective is to amplify the accuracy of the clustering outcomes produced in the initial half of the framework, whilst enabling nmODE to assimilate this target.

In pursuit of this, nmODE devises learning rules to update parameters $w_{ij}^{(2)}$, $w_{ij}^{(1)}$, and b_i as follows:

$$\begin{cases} w_{ij}^{(2)} \longleftarrow w_{ij}^{(2)} - \alpha \cdot \frac{\partial J}{\partial h_i(t)} \cdot g_j(t) \\ w_{ij}^{(1)} \longleftarrow w_{ij}^{(1)} - \beta \cdot \frac{\partial L}{\partial w_{ij}^{(1)}} \\ b_i \longleftarrow b_i - \beta \cdot \frac{\partial L}{\partial b_i} \end{cases}, \tag{13}$$

where α and β are the learning rates. Detailed elaborations regarding the procedure for updating parameters, along with the formulation of the loss L and cost functions J, could be found in [6].

In contrast to traditional target distribution, the $\boldsymbol{P}$ obtained by nmODE, is instrumental in enhancing the quality of embedded features gleaned by all autoencoders. Consequently, $\boldsymbol{P}$ is utilized across all views to force the soft clustering assignment of each view to approach the target distribution, which in turn better improving the feature representation learning. Specifically, for a given v-th view, the clustering loss $\mathcal{L}_c^v$ is computed as the Kullback-Leibler divergence. This divergence is calculated between the uniform target distribution $\boldsymbol{P}$ and its own clustering assignment distribution Q^v, as delineated as follows:

$$\mathcal{L}_c^v = D_{KL}(P||Q^v) = \sum_{i=1}^{N}\sum_{j=1}^{K} p_{ij} \log \frac{p_{ij}}{q_{ij}^v}. \tag{14}$$

Finally, we integrate feature representation learning and clustering goals into a unified framework. The total loss of each view consists of two parts:

$$\mathcal{L}^v = \mathcal{L}_r^v + \gamma \mathcal{L}_c^v, \tag{15}$$

where γ is the trade-off coefficient.

Hence how can the dynamics of nerualODE be exploited during training? In contrast to traditional networks where parameters remain fixed, nmODE introduces adaptability by allowing the parameters to change based on the fluctuations in input data. To this end, we alternately train the feature learning autoencoders and nmODE. During the t-th round of alternate training, nmODE exhibits learning behavior by updating parameters dynamically, in accordance with the knowledge it has acquired previously.

Specifically, we optimize autoencoders by the total loss $\mathcal{L}^v$, which the reliable pseudo-labels obtained by autoencoders make nmODE producing more precise target distribution compared to the initial input. The target distribution $\boldsymbol{P}$ encapsulates a range of probabilities, each corresponding to the likelihood of a particular datapoint belonging to a specific cluster. When considering the clustering result $\boldsymbol{y}$, it is derived from $\boldsymbol{P}$ by selecting the maximum value from each row of $\boldsymbol{P}$. In other words, for each datapoint, the cluster with the highest likelihood in the target distribution $\boldsymbol{P}$ is chosen to be its cluster assignment in $\boldsymbol{y}$. This approach aligns with the intuitive understanding that each datapoint should ideally be assigned to the cluster to which it is most likely to belong, based on the learned distribution.

Algorithm 1 The optimization of nmODE-MVC.

Require: Multi-View Dataset; Cluster Number K.

Pretrain autoencoders by minimizing Eq. (4).
Compute each view's embedded feature $\boldsymbol{z}_i^v$ by Eq. (2), global embedded feature $\boldsymbol{R}$ by Eq. (6).
Initialize pseudo-labels s_i by Eqs. (7)-(9).
Initialize parameters $W^{(1)}, W^{(2)}, b, \alpha$ and β.
while Not reach iteration T **do**
  while Not reach nmODE iteration T_{ode} **do**
    Solve each i-th corresponding one-dimensional ODE in Eq. (10) to get g_i.
    Update $w_{ij}^{(1)}, w_{ij}^{(2)}$ and b_i by Eq. (13).
  end while
  Generate the target distribution $\boldsymbol{P}$ by Eq. (12).
  Calculate each view's clustering loss $\mathcal{L}_c^v$ by Eq. (14).
  for fixed target distribution $\boldsymbol{P}$ **do**
    $\boldsymbol{\mu}_j^v = \boldsymbol{\mu}_j^v - \frac{\lambda}{n}\sum_{i=1}^{n}\frac{\partial \mathcal{L}_c^v}{\partial \boldsymbol{\mu}_j^v}$,
    $\phi^v = \phi^v - \frac{\lambda}{n}\sum_{i=1}^{n}\frac{\partial \mathcal{L}_r^v}{\partial \phi^v}$, and
    $\theta^v = \theta^v - \frac{\lambda}{n}\sum_{i=1}^{n}\left(\frac{\partial \mathcal{L}_r^v}{\partial \theta^v} + \gamma\frac{\partial \mathcal{L}^v}{\partial \theta^v}\right)$.
  end for
  Update each view's embedded feature $\boldsymbol{z}_i^v$ by Eq. (2), global embedded feature $\boldsymbol{R}$ by Eq. (6).
  Update pseudo-labels s_i by Eqs. (7)-(9).
end while
Compute the final target distribution $\boldsymbol{P}$ by Eq. (12).
Select the maximum value from each row of $\boldsymbol{P}$ to derive cluster assignments $\boldsymbol{y}$.

Ensure: Cluster Assignments $\boldsymbol{y}$. =0

E. Optimization

Algorithm 1 summarizes our overall optimization procedure. Initially, we minimize the reconstruction loss indicated in Eq. (4) to pretrain autoencoders for all views. Following this, we apply K-means clustering on the concatenated global embedded features, resulting in pseudo-labels s_i according to Eqs. (7)-(9). These pseudo-labels serve as a semantic label with information, guiding nmODE to generate more reliable target distribution, which in turn will optimize the feature learning. During the training process, the unique dynamical properties inherent in nmODE are leveraged, leading to capture comprehensive global information. This enables a more accurate target distribution $\boldsymbol{P}$, thereby augmenting the efficacy of data clustering distribution. Specifically, in a bid to optimize our model's performance, the training process incorporates an automatic iterative update mechanism, which adjusts the quantity and weights of each neuronODE. This dynamic adaptation allows our model to flexibly react and evolve based on the input data's specific characteristics, thereby enhancing the overall effectiveness of our clustering approach.

TABLE I
THE STATISTICS OF THE DATASETS.

Datasets	Samples	Views	Dimension	Classes
BDGP	2,500	2	[1750, 79]	5
HW	2,000	6	[216, 76, 64, 6, 240, 47]	10
Caltech-2V	1,400	2	[40, 254]	7
Caltech-3V	1,400	3	[40, 254, 928]	7
Caltech-4V	1,400	4	[40, 254, 928, 512]	7
Caltech-5V	1,400	5	[40, 254, 928, 512, 1984]	7

IV. EXPERIMENT

A. Experimental Setup

1) Datasets: In this study, we utilize three multi-view datasets. The statistics of these datasets are shown in Table I and the detailed descriptions are as follows:

• **BDGP** [30] includes 2500 samples from 5 different types of fruit fly embryos. Each sample has two views, corresponding to visual and text features.

• **HW** [1] contains 2000 samples from 10 categories corresponding to the digits 0-9. Each sample is composed of six visual views.

• **Caltech** [31] consists of five features from RGB images, including Wavelet Moments (WM), CENsus TRansform hISTogram (CENTRIST), Local Binary Pattern (LBP), GeneralIzed Search Trees (GIST), and Histogram of Oriented Gradients (HOG). **Caltech-2V** has the feature of WM [32] and CENTRIST [33], where each kind of feature is regarded as a view; **Caltech-3V** adds another feature of LBP [34] in comparison to the Caltech-2V; **Caltech-4V** adds another feature of GIST [35] in comparison to Caltech-3V; **Caltech-5V** adds another feature of HOG [36] in comparison to Caltech-4V.

2) Baseline methods: We compare our proposed nmODE-MVC method with several state-of-the-art multi-view clustering methods. The comparison includes the single view clustering method K-means [9] and five advanced MVC methods: DEMVC (Deep Embedded Multi-View Clustering with collaborative training [18]), SiMVC (Simple multi-view clustering [4]), CoMVC (trosten2021reconsidering [4]), DSMVC (Deep Safe Multi-View Clustering [19]), and SDMVC (Self-Supervised Discriminative Feature Learning for Deep Multi-View Clustering [5]).

3) Evaluation metrics: The clustering performance is evaluated by three metrics: clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI). For all these metrics, a higher value indicates better performance.

4) Implementation Details: In terms of network configurations, we set the structure of the encoder is: Input - Fc500 - Fc500 - Fc2000 - Fc10, and the decoder is symmetric with the encoder, where Fc denotes the fully connected layer. The autoencoder is alternately trained with the nmODEs 15 times. All experiments are performed on WindowsPC with Intel(R)Core(TM) i5-12600K CPU@3.69GHz, 32.0GB RAM,

[1]https://archive.ics.uci.edu/ml/datasets.php

TABLE II
RESULTS FOR ALL METHODS ON CALTECH DATASET WITH DIFFERENT VIEWS. "-*X*V " REPRESENTS THAT IT CONSISTS OF X VIEWS.

Datasets	Caltech-2V			Caltech-3V			Caltech-4V			Caltech-5V		
Evaluation metrics	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means [9] (1967)	0.416	0.305	0.351	0.463	0.313	0.403	0.546	0.467	0.489	0.574	0.491	0.511
DEMVC [18] (2021)	0.486	0.342	0.486	0.505	0.366	0.512	0.454	0.315	0.474	0.457	0.378	0.489
SiMVC [4] (2021)	0.499	0.451	0.543	0.561	0.495	0.584	0.622	0.541	0.657	0.714	0.677	0.722
CoMVC [4] (2021)	0.462	0.417	0.503	0.543	0.511	0.584	0.583	0.527	0.614	0.667	0.576	0.697
DSMVC [19] (2022)	0.584	0.466	**0.589**	0.729	0.629	**0.729**	**0.830**	0.768	**0.830**	**0.899**	0.815	**0.899**
SDMVC [5] (2022)	0.478	0.374	0.406	0.416	0.306	0.379	0.435	0.305	0.401	0.421	0.282	0.387
nmODE-MVC (ours)	**0.614**	**0.478**	0.403	**0.806**	**0.720**	0.665	0.825	**0.774**	0.690	0.885	**0.820**	0.806

TABLE III
RESULTS FOR ALL METHODS IN TERMS OF ACC, NMI AND ARI ON BDGP AND HW DATASETS.

Datasets	BDGP			HW		
Evaluation metrics	ACC	NMI	ARI	ACC	NMI	ARI
K-means [9] (1967)	0.432	0.569	0.260	0.754	0.785	0.667
DEMVC [18] (2021)	0.751	0.750	0.751	0.676	0.706	0.588
SiMVC [4] (2021)	0.754	0.670	0.754	0.640	0.821	0.665
CoMVC [4] (2021)	0.812	0.733	0.813	0.739	0.834	0.727
DSMVC [19] (2022)	0.658	0.444	0.658	-	-	-
SDMVC [5] (2022)	0.978	0.934	0.948	0.971	**0.944**	0.939
nmODE-MVC (ours)	**0.991**	**0.969**	**0.975**	**0.974**	0.940	**0.942**

and GeForce RTX 3070ti GPU (8GB caches). For fair comparison, all baselines are tuned to the best performance according to the corresponding papers.

B. Experimental Results

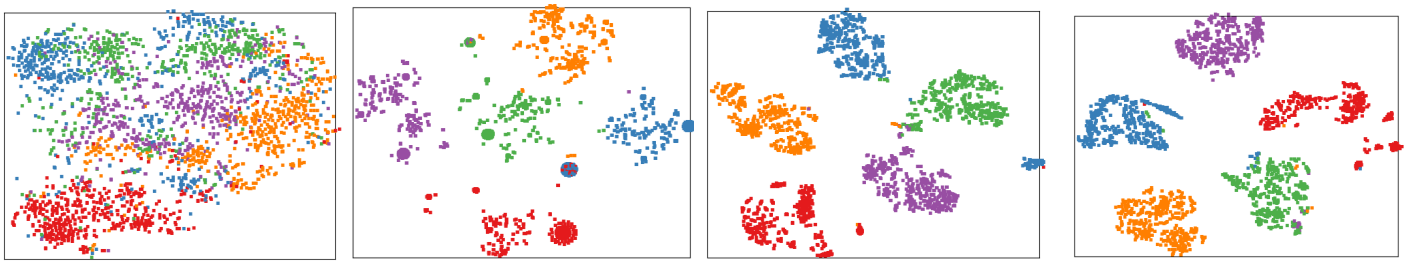
In this section, we perform comprehensive experiments, comparing our proposed nmODE-MVC method against existing state-of-the-art clustering algorithms. Clustering outcomes for the BDGP and HW multi-view datasets are elucidated in Table III, and those pertinent to the Caletch multi-view datasets are detailed in Table II. The superior results in each column are emboldened, and results of the runner-up are underlined. When employing the K-means algorithm, we initiated ten individual runs and then computed the average.

In the initial stages, we opted for the BDGP dataset encompassing two views and the HW dataset containing six views. These datasets are marked by a considerable variance in the number of views they present. The primary motive of such a selection was to examine if our proposed model could efficiently adapt to handle datasets with disparate view data, thereby assessing its versatility and potential for generalization. Following this, we proceeded with the selection of the Caltech-5V dataset. The dataset underwent subsequent processing to be segregated into 2V, 3V, 4V, and 5V subsets, where 'V' denotes the number of views. This process was designed to further affirm the ability of our model to retain or even enhance its performance as the quantity of views, and consequently, the volume of information escalates.

1) Clustering Performance Comparison: Upon inspecting the findings presented in Table III and Table II, it becomes clear that our proposed nmODE-MVC demonstrates notable efficacy on the multi-view datasets when evaluated via three established metrics: ACC, NMI, and ARI. Remarkably, nmODE-MVC achieves better ACC than the state-of-the-art MVC model on four multi-view data clustering tasks, i.e., BDGP (+1.3%), HW (+0.3%), Caltech-2V (+3.0%) and Caltech-3V (+7.7%). On the other two datasets, nmODE-MVC achieves ACC results close to the SOTA MVC model and slightly exceeds SOTA on NMI: Caltech-4V (+0.6%) and Caltech-5V (+0.5%).

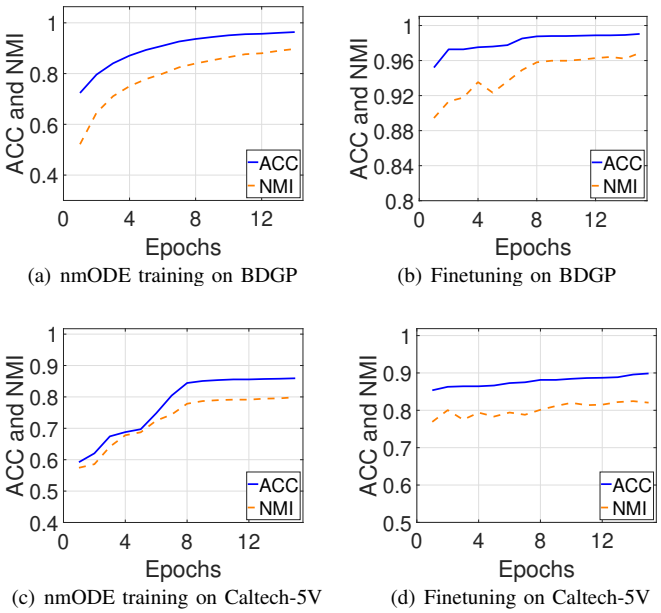
2) Visualization of Learning Process: In order to intuitively gauge the effectiveness of the proposed nmODE-MVC, we have utilized the t-SNE algorithm [16] for the visualization of varying layers' global embedded distributions. While the target distribution $\boldsymbol{P}$ remains not visually representable, the global embedded feature $\boldsymbol{R}$ distribution nonetheless offers a discernable representation of data separability. We provide a clear illustration of data separability and inseparability across four distinct stages – these include the original dataset, pretrain phase, the 5-th stage of alternating training, and the 15-th stage of alternating training. As evident in Figure 2, the initial separability of the original dataset appears relatively low. However, after the initial five rounds of alternating training involving the encoder and nmODE, there's a marked improvement in the dataset's separability. Furthermore, the clustering structure becomes increasingly pronounced with additional alternating training cycles, thus validating the effectiveness of the proposed nmODE-MVC.

3) Analysis of Training Process: This section aims to investigate the process of training across different datasets and explore how the iteration number impacts the evolution of clustering performance at varying stages. As depicted in Figure 3, the ACC and NMI are respectively represented by blue and orange lines. Our experimental findings suggest that due to the dynamic characteristics inherent in nmODE-MVC, which facilitates the dynamic updating of data features and distributions previously learned, there's a rapid enhancement in the clustering performance during the nmODE training phase. During the fine-tuning phase, we notice a moderate improvement in the clustering performance. This is attributed to the usage of the target distribution $\boldsymbol{P}$, as generated by nmODE, to aid in the fine-tuning of all autoencoders through the minimization of the loss $\mathcal{L}$, thereby assisting in their



(a) Raw data (b) Pretrain autoencoders (c) Alternate training 5 times (d) Alternate training 15 times

Fig. 2. Visualization of the embedded global features $\boldsymbol{R}$ for BDGP dataset.



(a) nmODE training on BDGP (b) Finetuning on BDGP

(c) nmODE training on Caltech-5V (d) Finetuning on Caltech-5V

Fig. 3. Clustering performance during different training processes. (a) and (c) indicate the nmODE training on the BDGP and Caltech-5V datasets, respectively. (b) and (d) indicate the finetuning stage on the BDGP and Caltech-5V datasets, respectively.

updates. Furthermore, the nmODE-MVC's clustering results tend to stabilize as the iteration number increases, showing no significant fluctuations, thereby indicating the robustness of our proposed model.

4) Ablation Experiments: We perform an ablation study on the nmODE-MVC model by comparing it to four simpler versions to understand the significance of each component in our primary model. (1) "w/o $\mathcal{L}_c$ $\mathcal{L}_r$": excluding both clustering loss $\mathcal{L}_c$ and reconstruction loss $\mathcal{L}_r$; (2) "w/o $\mathcal{L}_r$": sans $\mathcal{L}_c$ loss, employing solely $\mathcal{L}_r$; (3) "w/o $\mathcal{L}_c$": devoid of $\mathcal{L}_r$, utilizing exclusively $\mathcal{L}_c$; (4) devoid of the nmODE dynamics module, relying only on the autoencoder. We executed ablation experiments on three distinct datasets, delivering both ACC and NMI scores as illustrated in Table IV. The outcomes of the initial three ablation baselines elucidate the essential contributions made by the $\mathcal{L}_c$ and $\mathcal{L}_r$ loss functions obtained by the model towards the overall clustering results. The fourth ablation baseline underscores when the nmODE module is removed,

TABLE IV
THE ABLATION STUDY RESULTS OF NMODE-MVC ON THREE DATASETS. THE ORIGINAL RESULTS ARE SHOWN IN BOLD.

Model	BDGP		HW		Caltech-5V	
	ACC	NMI	ACC	NMI	ACC	NMI
w/o $\mathcal{L}_c$ $\mathcal{L}_r$	0.703	0.602	0.699	0.671	0.618	0.534
w/o $\mathcal{L}_r$	0.867	0.724	0.851	0.808	0.762	0.710
w/o $\mathcal{L}_c$	0.896	0.752	0.862	0.826	0.770	0.704
w/o nmODE	0.897	0.701	0.674	0.702	0.473	0.504
nmODE-MVC	**0.991**	**0.969**	**0.974**	**0.940**	**0.898**	**0.820**

the model degrades to SDMVC [5]. However, the nmODE-MVC only requires 15 iterations while SDMVC necessitates numerous rounds of iterations. This considerable reduction in computational iterations not only speeds up the process but also contributes to a more stable and robust model, as evidenced by the higher accuracy of nmODE-MVC compared to the SDMVC variant in our ablation studies.

5) Complexity Analysis: Let M be the representation for the maximum quantity of neurons embedded within the autoencoder's hidden layers, Z symbolize the maximum dimensionality of the embedding features, T symbolize the number of iterations for the outer loop, T_{ode} symbolize the number of iterations for the inner loop (ODE solving step), and K, V, and N stand for the numbers of clusters, views, and examples respectively. The K-means and target distribution computations are performed once per outer loop iteration. Thus, their contribution to the total complexity will be $T \times O(NZK)$, giving us $O(TNZK)$. The autoencoder adheres to a complexity of O($TNVM^2$). The nmODE model is architectured on the foundational blocks of neuronODE (one-dimensional) and invODE (three-dimensional). Owing to the relatively lower dimensions of these modules, their computational tasks can be executed with relative ease, giving us $O(TNT_{ode})$. The total algorithm adheres to a complexity of $O(TN(ZK + VM^2 + T_{ode}))$.

V. CONCLUSION AND FUTURE WORK

This work introduces a novel, end-to-end, self-supervised multi-view clustering technique underpinned by NeuralODEs. It overcomes the challenge of suboptimal embedding features, commonly associated with autoencoders, by jointly implementing the feature learning objective and the clustering

objective In terms of addressing the noted limitations of the K-means algorithm, particularly its inherent randomness and instability, we have strategically refrained from utilizing it directly to derive clustering outcomes. Instead, it is harnessed for the generation of soft clustering pseudo-labels, and the final clustering assignment is generated by nmODE model. We have introduced the element of NeuralODE to our model, which confers dynamical attributes to our clustering algorithm. The effectiveness of the proposed method is validated through experiments on multiple real-world multi-view datasets. Additionally, one of the primary advantages of nmODE is its efficiency, notably demonstrated by its ability to achieve desirable results with a lower number of iterations compared to other models, which is evidenced by the higher accuracy of nmODE-MVC compared to the SDMVC [5] for the same number of iterations in our ablation studies.

In the future, we are set on amplifying the capabilities of our existing ODE-based multi-view clustering method. We contemplate attaching an nmODE unit at the termination of each view, which empowers it to dynamically tune its parameters in response to the specific data characteristics of each view. This enhancement will capacitate our model to more proficiently encapsulate the distinct features prevalent in each view, which, in turn, is anticipated to markedly improve the precision of our clustering outcomes. The measure of this enhanced accuracy will be subsequently validated through rigorous evaluation methodologies

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China (No. 2018AAA0100204) and Shenzhen Science and Technology Program (Nos. JCYJ20230807120010021 and JCYJ20230807115959041). Lifang He is partially supported by the NSF grants (MRI-2215789, IIS-1909879, IIS-2319451), NIH grant under R21EY034179, and Lehigh's grants under Accelerator and CORE.

REFERENCES

[1] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016, pp. 478–487.
[2] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *WWW*, 2020, pp. 1400–1410.
[3] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *NeurIPS*, 2018.
[4] D. J. Trosten, S. Lokse, R. Jenssen, and M. Kampffmeyer, "Reconsidering representation alignment for multi-view clustering," in *CVPR*, 2021, pp. 1255–1265.
[5] J. Xu, Y. Ren, H. Tang, Z. Yang, L. Pan, Y. Yang, X. Pu, P. S. Yu, and L. He, "Self-supervised discriminative feature learning for deep multi-view clustering," *TKDE*, 2022.
[6] Z. Yi, "nmode: neural memory ordinary differential equation," *Artificial Intelligence Review*, pp. 1–36, 2023.
[7] T. Zhang, Z. Yao, A. Gholami, J. E. Gonzalez, K. Keutzer, M. W. Mahoney, and G. Biros, "Anodev2: A coupled neural ode framework," *NeurIPS*, 2019.
[8] B. Zhang, X. Li, S. Feng, Y. Ye, and R. Ye, "Metanode: Prototype optimization as a neural ode for few-shot learning," in *AAAI*, no. 8, 2022, pp. 9014–9021.
[9] J. MacQueen, "Classification and analysis of multivariate observations," in *BSMSP*, 1967, pp. 281–297.
[10] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *NeurIPS*, 2001.
[11] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation." in *IJCAI*, 2017, pp. 1753–1759.
[12] J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, and Z. Zhang, "Efficient deep embedded subspace clustering," in *CVPR*, 2022, pp. 1–10.
[13] Z. Yang, Y. Ren, Z. Wu, M. Zeng, J. Xu, Y. Yang, X. Pu, S. Y. Philip, and L. He, "Dc-fuda: Improving deep clustering via fully unsupervised domain adaptation," *Neurocomputing*, vol. 526, pp. 109–120, 2023.
[14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *JMLR*, no. 12, 2010.
[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[16] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *JMLR*, no. 11, 2008.
[17] L. Yang, N.-M. Cheung, J. Li, and J. Fang, "Deep clustering by gaussian mixture variational autoencoders with graph embedding," in *CVPR*, 2019, pp. 6440–6449.
[18] J. Xu, Y. Ren, G. Li, L. Pan, C. Zhu, and Z. Xu, "Deep embedded multi-view clustering with collaborative training," *Inf. Sci*, pp. 279–290, 2021.
[19] H. Tang and Y. Liu, "Deep safe multi-view clustering: Reducing the risk of clustering performance degradation caused by view increase," in *CVPR*, 2022, pp. 202–211.
[20] M. R. Khan and J. E. Blumenstock, "Multi-gcn: Graph convolutional networks for multi-view networks, with applications to global poverty," in *AAAI*, no. 01, 2019, pp. 606–613.
[21] J. Xu, Y. Ren, X. Shi, H. T. Shen, and X. Zhu, "Untie: Clustering analysis with disentanglement in multi-view information fusion," *Information Fusion*, vol. 100, p. 101937, 2023.
[22] X. Chen, J. Xu, Y. Ren, X. Pu, C. Zhu, X. Zhu, Z. Hao, and L. He, "Federated deep multi-view clustering with global self-supervision," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 3498–3506.
[23] C. Cui, Y. Ren, J. Pu, J. Li, X. Pu, T. Wu, Y. Shi, and L. He, "A novel approach for effective multi-view clustering with information-theoretic perspective," *arXiv preprint arXiv:2309.13989*, 2023.
[24] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, P. S. Yu, and L. He, "Deep clustering: A comprehensive survey," 2022.
[25] J. Zhang, P. Zhang, B. Kong, J. Wei, and X. Jiang, "Continuous self-attention models with neural ode networks," in *AAAI*, no. 16, 2021, pp. 14 393–14 401.
[26] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Occupancy flow: 4d reconstruction by learning particle dynamics," in *ICCV*, Feb 2020.
[27] B. Jiang, Y. Zhang, X. Wei, X. Xue, and Y. Fu, "Learning compositional representation for 4d captures with neural ode," in *CVPR*, 2021, pp. 5340–5350.
[28] M. Bai, S. Choy, J. Zhang, and J. Gao, "Neural ordinary differential equation model for evolutionary subspace clustering and its applications." Jul 2021.
[29] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *JMLR*, pp. 2579–2605, 2008.
[30] X. Cai, H. Wang, H. Huang, and C. Ding, "Joint stage recognition and anatomical annotation of drosophila gene expression patterns," *Bioinformatics*, pp. i16–24, 2012.
[31] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *CVPR*, 2004, pp. 178–178.
[32] D. Shen and H. H. Ip, "Discriminative wavelet shape descriptors for recognition of 2-d patterns," *Pattern Recognition*, p. 151–165, Jul 2002.
[33] J. Wu and J. M. Rehg, "Centrist: A visual descriptor for scene categorization," *TPAMI*, no. 8, p. 1489–1501, Dec 2010.
[34] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *ICPR*. IEEE, 1994, pp. 582–585.
[35] A. Friedman, "Framing pictures: the role of knowledge in automatized encoding and memory for gist." *Journal of experimental psychology: General*, no. 3, p. 316, 1979.
[36] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, Jul 2005.