User-Guided Deep Multiple Clustering

Jiawei Yao

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Juhua Hu, Chair

Ankur Teredesai

Qi Qian

Program Authorized to Offer Degree:

Computer Science and Systems

University of Washington

Abstract

User-Guided Deep Multiple Clustering

Jiawei Yao

Chair of the Supervisory Committee:
Juhua Hu
Department of Computer Science and Systems

Multiple clustering is based on the observation that a dataset can often be partitioned in more than one meaningful way (for example by color or by shape). However, most existing deep methods still optimize a single partition or produce several partitions without making clear which underlying factors they capture, and they often separate representation learning from the clustering objective. This can lead to results that do not match the aspects users care about. This dissertation proposes a user-preference guided framework for deep multiple clustering that aims to obtain partitions that are both diverse and aligned with user interests, and is organized into four contributions that start from data-driven ways of identifying relevant factors and progress to methods that explicitly incorporate user intent and practical system considerations.

The first contribution, AugDMC, uses targeted data augmentations as aspect selectors together with a self-supervised, prototype-based objective with stabilization, to learn representations that preserve distinct factors of variation and support multiple interpretable partitions without manual feature engineering. The second contribution, DDMC, introduces dual-level disentanglement tailored to clustering: a variational EM procedure links coarse and fine grained factor discovery (E-step) with a clustering-aware objective (M-step), narrowing the gap between learning "good features" and obtaining "good partitions". The third contribution, Multi-MaP, aligns frozen CLIP encoders with a user's high-level concept by

introducing learnable textual proxies and constraining them with concept-level and LLM-derived reference-word signals. Building on this, the fourth contribution, Multi-Sub, is a framework for concept conditioned subspace proxies. It first builds a low dimensional subspace that is guided by text, using reference words suggested by an LLM, and then learns a proxy for each image inside this subspace. Representation learning and clustering are optimized together, so the method no longer needs contrastive concepts specified by the user and it also avoids the extra cost of a two stage pipeline.

On publicly available visual multiple-clustering benchmarks such as ALOI, Stanford Cars, CMUface, Flowers, Fruit/Fruit360, and Cards, these methods consistently improve NMI and RI and yield partitions that better reflect user intent, with ablation studies validating each design choice. Taken together, the results illustrate how incorporating user preferences, structuring the representation space, and jointly optimizing representations and clusters can make multiple clustering systems better aligned with users' actual goals in practice.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to begin by expressing my deepest gratitude to my advisor, Dr. Juhua Hu. From the very first meeting to the final stages of this dissertation, she has been a constant source of guidance, challenge, and support. She pushed me to think more deeply, to write more clearly, and to never settle for the "first reasonable answer." Many times, when I felt lost in details or overwhelmed by setbacks, her sharp questions and calm perspective helped me rediscover the core of the problem and move forward. I am especially thankful for her patience with my imperfections, her honesty when something was not good enough, and her belief that I could grow into someone who could eventually "own" the work. The way she approaches research—with rigor, integrity, and quiet persistence—has profoundly shaped how I hope to conduct myself as a scientist in the future.

I am also sincerely grateful to my committee members, Dr. Ankur Teredesai and Dr. Qi Qian, for their time, thoughtful feedback, and encouragement. Their comments during my proposal, practice talks, and defense helped me see my work from different angles and forced me to articulate not just what I did, but why it matters. They asked the questions I did not always want to hear but needed to answer, and their suggestions significantly improved both the technical soundness and the clarity of this dissertation. I deeply appreciate their willingness to invest their time and energy into my growth as a researcher.

This work would not have been possible without generous institutional and financial support. I gratefully acknowledge support from J.P. Morgan Chase & Co., and partial support from NSF Grant No. 2104270 and Advata Gift funding, which enabled me to pursue long-term, high-risk ideas during my PhD and to explore research directions that required significant time, computation, and iteration.

Finally, I would like to thank my family. Although they are far away in distance, their love, trust, and quiet pride have always been close to me. Their understanding when I missed family events, their concern during difficult periods, and their constant reassurance that "everything will be okay" carried me through more moments than they know.

Above all, my deepest thanks go to my partner, Tong. Thank you for walking beside me through every stage of this journey—for sharing the stress of deadlines, listening to half-formed ideas late at night, and accepting the countless evenings and weekends when my attention was somewhere inside a paper or an experiment. Thank you for your patience when I was exhausted, for your strength when I was discouraged, and for your gentle reminders that life is more than results and rejection letters. This dissertation is not just a record of my work, but also of your support, sacrifices, and faith in me. I could not have done this without you, and I am endlessly grateful to have you by my side.

Chapter 1

INTRODUCTION

Clustering, which groups data points based on their similarities, has been extensively researched in the fields of data mining and machine learning, since huge amounts of unlabeled data are becoming more and more available. Traditional clustering algorithms, such as k-means [99], spectral clustering [108], DBSCAN [35], and Gaussian mixture model [14], group data into distinct collections with handcrafted features. However, these features are designed for general purposes and unsuitable for specific tasks. With the development of deep learning, deep clustering algorithms [160, 116, 48, 56, 130, 135] adopt Deep Neural Networks (DNNs) to perform clustering, showing dramatic performance improvement in different applications such as bioinformatics with representation learning [61, 113], computer vision [115, 2], speech processing [7, 144], and text mining [98, 4].

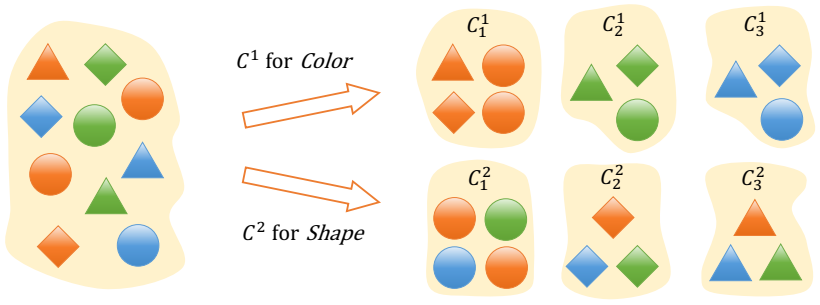


Figure 1.1: An example of multiple clustering. Multiple clustering methods can reveal two or more distinct clusterings (i.e., C^1 for color and C^2 for shape).

1.1 Background

Most clustering algorithms produce a single partition of data [99]. However, in real applications, there might have different orthogonal ways to partition a given dataset [182]. For example, as shown in Fig. 1.1, the geometric figures have two orthogonal ways of data partitions: color and shape. To address this problem, some researchers propose multiple clustering algorithms that aim to find more than one way to partition a given dataset, where different ways can be used for different application purposes [69]. The multiple clustering algorithms can be roughly categorized into shallow and deep models. For shallow models, some of these methods use constraints to generate alternative clusterings. For example, COALA [8] treats objects within an established clustering as constraints for generating an alternative clustering. Qi et al. [117] regard multiple clustering as a constrained optimization problem to obtain alternative clustering. Some other ones rely on different feature subspaces, e.g., [70] proved the relation between Laplacian eigengap and stability of clustering, and discovered multiple clusterings via maximizing the eigengap within different feature subspaces. Recently, some researchers leverage deep learning to generate multiple clusterings and achieve better results. Wei et al. [153] proposed a deep matrix factorization based method to discover multiple clusterings using multi-view data. ENRC [102] exploits an autoencoder to learn the object features and generates multiple clustering via optimizing a clustering objective function and iMClusts [124] makes use of autoencoders and multi-head attention to generate multiple clusterings.

1.2 Motivation

Although existing multiple clustering methods have achieved very promising results in producing various features for clusterings, not all of these features may be relevant or interesting to the users. We are thus interested in user-guided multiple clustering, where the partitions should be both interpretable and aligned with the user's needs, rather than only numerically accurate. Here, user-guided multiple clustering refers to settings where the user provides

high-level semantic preferences (e.g., "color", "species", "pose") or brief textual descriptions, and the algorithm learns representations and partitions that are explicitly aligned with these intents rather than exploring diverse clusterings in a purely unsupervised way.

However, a key problem for user-guided multiple clustering is how to find and select the features that match the users' preferences. In this dissertation, we view this as an effective feature-learning problem: we seek representations that capture diverse but relevant aspects of the data and can be steered by a given preference. This dissertation studies how to design user-guided deep multiple clustering algorithms from the perspective of effective feature learning. In this dissertation, we primarily instantiate and evaluate our methods on visual (image) datasets, while the underlying methodologies are general and can be extended to other data modalities and application domains in future work. Concretely, we consider the following three scenarios.

Scenario I: Capture users' interest by perturbing data Existing methods all input the original images into the algorithm and directly obtain the multiple features of the data from the original images. However, a very promising scenario is: can we perturb the data before it is fed into the model, so that the model can better capture the multiple features from the data? For example, the images of fruit in Fig. 1.2 have two main concepts: color and species. If a user wants to obtain the feature w.r.t. species, the user can perturb the color of these data. Therefore, the features of data color will be disrupted, and the features of species will be easier to capture.

Data augmentation [143] has been widely used to improve the generalization performance of models, which also provides an effective way to perturb the images. However, its potential to capture diverse aspects of the data for multiple clustering purposes has been largely overlooked. In this scenario, we use different types of data augmentation to preserve different aspects of the images and then generate clusterings that follow the user's preferred aspect.

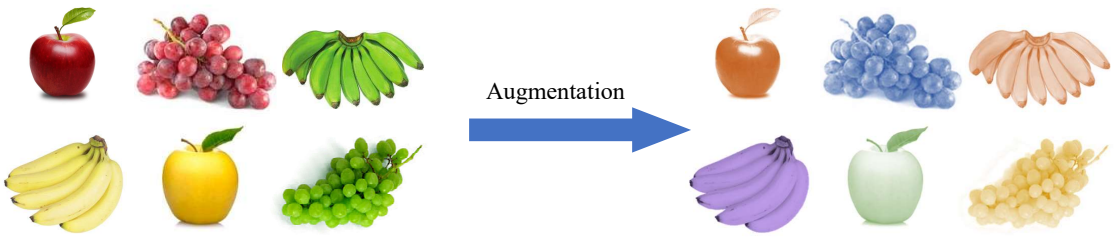


Figure 1.2: An example of data augmentation.

Scenario II: Learning different aspects of features end to end to capture users' interests Although the scenario I can more effectively capture users' interests for multiple clustering, it still follows a common paradigm of multiple clustering: most existing methods first obtain feature representations that maximize the dissimilarity among different clusterings, and the learned representations are directly fed to standard clustering methods, e.g., k-means, to produce the final multiple clustering outcomes. However, merely controlling the dissimilarity of representations cannot directly capture the diversity features, but may instead result in too much noise in the learned representations. Also, the learned representations are often not explicitly optimized for clustering. Therefore, how to effectively separate different features from data and generate corresponding multiple embeddings for clustering is the main challenge in multiple clustering.

Disentangled representation learning [86, 194, 50, 63, 12, 29] offers a natural way to learn separate factors of variation and is therefore a good fit for multiple clustering. Disentangled representation learning aims to learn factorized representations that can discover and isolate the latent factors in data. Such factorized representations can support diverse clusterings: disentangled representation learning separates latent factors and assigns them to different latent variables in the representation space [65, 194], so that, for example, the latent variable for shape varies only with the object's shape but not with other factors. Since both disentangled representation learning and multiple clustering seek to represent latent factors separately, it is natural to combine them.

Scenario III: Generating user-specific clusterings In practice, however, users rarely need all clusterings produced by an algorithm. Inspecting each clustering and deciding which ones are useful can be time-consuming, so a more practical goal is to generate only those clusterings that match a given user preference. However, since the dataset usually only contains image information, and the users' interests are text information, traditional multiple clustering methods have difficulty establishing a connection between these data. Thus, the existing methods cannot directly generate clusters based on the users' interests. Besides, it is hard to choose the right clustering without knowing what each result means. These issues make it difficult to generate multiple clusterings that truly match users' interests.

Users typically express their interests through concise keywords (e.g., color or species), and aligning these with different visual components precisely is challenging. Fortunately, multi-modal and large language models have made it easier to bridge this gap. Multi-modal models such as CLIP [121], which jointly model images and text, provide a natural tool for this scenario. In this dissertation we ask whether such models can be used to uncover different aspects of images and to drive user-specific multiple clusterings.

1.3 Contributions

This dissertation studies user-preference-guided representation learning for deep multiple clustering. We focus on three ideas: (i) using data perturbations to reveal complementary factors, (ii) combining disentanglement with clustering-aware training, and (iii) conditioning vision models on natural-language preferences. Based on these ideas, we develop four methods, ranging from unsupervised discovery of salient aspects to explicitly user-conditioned clustering. Our main contributions are:

- **Data Augmentation Guided Deep Multiple Clustering**. We propose AugDMC, which uses data augmentation to learn image representations for multiple clustering. AugDMC applies different augmentations that preserve different aspects of the data and learns prototype-based representations in a self-supervised way, so that each aug-

mentation emphasizes a specific aspect. We further introduce a simple stabilization strategy to handle the optimization difficulties caused by heterogeneous augmentations. Using the learned representations, we obtain multiple clusterings that correspond to different aspects of the data.

- **Dual-disentangled Deep Multiple Clustering**. We propose DDMC, which learns disentangled representations for multiple clusterings within a variational Expectation–Maximization (EM) framework. In the E-step, a disentanglement module learns coarse-grained and fine-grained latent factors from the data. In the M-step, a clustering module updates cluster assignments by optimizing a clustering objective. DDMC alternates between these two steps to jointly refine the latent factors and the cluster assignments.

- **Multi-Modal Proxy Learning Towards Personalized Visual Multiple Clustering**. We propose Multi-MaP, which uses multi-modal proxy learning to obtain multiple clusterings. Multi-MaP uses CLIP encoders to extract text and image embeddings, and GPT-4 to turn a user's keyword into a short textual context. We then design reference-word and concept-level constraints to learn a text proxy that reflects the user's interest. This proxy allows Multi-MaP to capture the requested aspect and to identify the clusterings that are most relevant to it.

- **Customized Multiple Clustering via Multi-Modal Subspace Proxy Learning**. We propose Multi-Sub, an end-to-end framework that turns a user's high-level textual preference into a task-specific subspace for clustering. The method obtains reference words from a large language model and treats them as bases of this subspace. For each image, it learns a proxy-word embedding as a weighted combination of these bases and aligns it with the image embedding using a partially trainable CLIP image head. Multi-Sub alternates proxy learning with a clustering objective, jointly optimizing the representations and the cluster assignments to produce user-specific partitions.

1.4 List of Publications

This dissertation is based on the material from the following published works:

Conference Papers

- Jiawei Yao and Juhua Hu. "Dual-disentangled Deep Multiple Clustering." In *Proceedings of the 24th SIAM International Conference on Data Mining (SDM'24)*, 2024.

- Jiawei Yao, Qi Qian, and Juhua Hu. "Multi-modal Proxy Learning Towards Personalized Visual Multiple Clustering." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'24)*, 2024.

- Jiawei Yao, Qi Qian, and Juhua Hu. "Customized Multiple Clustering via Multi-modal Subspace Proxy Learning." In *Advances in Neural Information Processing Systems 37 (NeurIPS'24)*, 2024.

Journal Paper

- Jiawei Yao, Enbei Liu, Maham Rashid, and Juhua Hu. "AugDMC: Data Augmentation Guided Deep Multiple Clustering." *Procedia Computer Science*, 222, 2023.

Chapter 2

RELATED WORK

2.1 Definitions and Preliminaries

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a dataset with N instances, where each instance $\mathbf{x}_i \in \mathbb{R}^d$ is represented by a d-dimensional feature vector. Unlike traditional *single-partition* clustering, which seeks a single partition of $\mathbf{X}$ into K clusters, *multiple clustering* aims to generate multiple distinct yet valid partitions of the same dataset. Formally, let $C^{(p)} = \{\mathcal{C}_1^{(p)}, \ldots, \mathcal{C}_{K_p}^{(p)}\}$ denote the p-th clustering solution (or partition), where $\mathcal{C}_j^{(p)}$ is the j-th cluster in the p-th partition, and $p \in \{1, \ldots, M\}$ indexes the total number of clustering solutions to be produced. Each partition $C^{(p)}$ should satisfy common clustering properties, such as non-overlapping clusters (i.e., $\mathcal{C}_j^{(p)} \cap \mathcal{C}_{j'}^{(p)} = \varnothing$ for $j \neq j'$) and covering the entire dataset (i.e., $\bigcup_{j=1}^{K_p} \mathcal{C}_j^{(p)} = \mathbf{X}$).

A key challenge in multiple clustering is the requirement of *diversity* among the solutions. This means that the partitions $C^{(p)}$ and $C^{(q)}$ (for $p \neq q$) should provide *complementary* or *alternative* groupings rather than near-duplicates of each other. The exact criterion for measuring diversity may vary across different methods (e.g., mutual information, orthogonality of cluster indicators, or other divergence measures). By generating multiple clustering solutions that highlight different structural patterns, practitioners can explore the data from various perspectives and gain deeper insights into the underlying distributions.

Non-Deep Learning Baselines. A common starting point for generating multiple clusterings is to vary the initialization of *K-means* [20], although these solutions often lack substantial diversity. To address this limitation, *AKM*^{3}*C* [71] and *Nr-Kmeans* [100] introduce additional constraints or diversification strategies, ensuring that each partition offers

distinct insights.

Another line of work applies *spectral clustering* [111], which manipulates the affinity matrix or eigenvector selection to produce alternative solutions in a low-dimensional spectral space. By contrast, *density-based methods*, such as DBSCAN [37], vary density thresholds to capture multiple partitions with differing granularities and shapes [107].

While effective in certain scenarios, these baselines often rely on shallow or hand-crafted features, limiting their ability to handle high-dimensional and complex data. Consequently, the following sections pivot toward *deep learning-based* paradigms, which learn robust feature representations to uncover richer and more flexible multiple clustering solutions.

2.2 Single Clustering

2.2.1 Shallow models

Over the past decades, a large number of clustering methods with shallow models have been proposed [75]. Some methods are based on density, DBSCAN [35] is a clustering method relying on a density-based notion of clusters that is designed to discover clusters of arbitrary shape. BMSC [126] is a nonparametric clustering method that overcomes the limitations of the mean shift problem. Some researchers leverage ensemble to achieve better cluster results [125]. For example, Strehl and Ghosh [136] studied the cluster ensemble problem and provided a mutual information-based method to solve it. These shallow models are effective only when the features are representative, while their performance on complex data is usually limited due to the poor power of feature learning.

2.2.2 Deep models

Deep clustering aims at effectively extracting more clustering-friendly features from data and performing clustering with learned features simultaneously. Yang et al. [168] proposed a joint unsupervised learning method named JULE, which applies agglomerative clustering magic to train the feature extractor. Chang et al. [25] proposed deep adaptive image clustering

(DAC) to tackle the combination of feature learning and clustering. Zhong et al. [191] introduced deep robust clustering (DRC), where two contrastive loss terms are introduced to decrease intra-class variance and increase inter-class variance. Cao et al. [21] proposed a simple, scalable, and stable variational deep clustering algorithm, which introduces generic improvements for variational deep clustering. ClusterFormer [92] is a universal vision model that is based on the clustering paradigm with Transformer, which uses the updated cluster centers to redistribute image features through similarity-based metrics. Gray et al. [54] offered a new perspective on the well established agglomerative clustering algorithm, focusing on recovery of hierarchical structure.

2.3 Multiple Clustering

2.3.1 Deep Representation Learning-based

Unlike traditional clustering techniques that operate directly in the *original feature space*, deep representation learning-based approaches first embed data into a *latent space* (often lower-dimensional). By learning more expressive representations, these methods can disentangle complex data structures and produce multiple clustering solutions with higher fidelity and diversity.

Representation Forms. A large class of deep clustering methods utilizes *autoencoders* [141] to learn compact, non-linear embeddings. Concretely, an encoder $\mathbf{z}_i = E_{\boldsymbol{\theta}_E}(\mathbf{x}_i)$, and a decoder $\hat{\mathbf{x}}_i = D_{\boldsymbol{\theta}_D}(\mathbf{z}_i)$ are jointly trained to minimize a reconstruction loss, $\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$. Deep Embedded Clustering (DEC) [160] and its improved variants [58] demonstrate that such an embedding can greatly facilitate downstream clustering. While early methods produce only a *single* solution, more recent extensions incorporate diversity regularizers in a multi-head or multi-branch structure [178, 91] to yield multiple distinct partitions.

Beyond *autoencoders*, low-order tensor factorization has also been explored to facilitate multiple clustering by modeling higher-order correlations among heterogeneous data. Zhao *et al.* [189] proposed *TMC*, a flexible framework that represents cyber-physical-social systems

as low-order tensors and introduces a weight tensor construction approach to measure the importance of various attribute combinations. A selective weighted tensor distance is then utilized to cluster tensorized objects according to different application needs, yielding high-quality partitions with minimal redundancy.

Graph neural networks (GNNs) provide another powerful representation when relational or structural links are available. Let $\mathbf{A}$ be a graph adjacency matrix. A GNN layer can be viewed as

$$\mathbf{H}^{(\ell+1)} = \sigma\Big(\widetilde{\mathbf{A}}\,\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}\Big), \tag{2.1}$$

where $\widetilde{\mathbf{A}}$ is a normalized version of $\mathbf{A}$, $\mathbf{H}^{(\ell)}$ is the node embedding at layer ℓ, and σ denotes a non-linear activation. Multiple partition heads or variable hyperparameters in GNN-based approaches can then induce *diverse* clustering solutions that leverage the underlying topological structure [170, 171].

Subspace-Based Approaches Rather than producing a single unified embedding, *subspace-based* methods discover multiple low-dimensional subspaces, each corresponding to a distinct partitioning scheme. This is particularly beneficial when different clustering solutions are embedded in different manifolds or subspace structures. Formally, one might learn subspace projections $\mathbf{W}_p \in \mathbb{R}^{d\times r_p}$ for $p \in \{1,\ldots,M\}$, mapping $\mathbf{x}_i$ to subspace $\mathbf{z}_i^{(p)} = \mathbf{W}_p^\top \mathbf{x}_i$. Cluster assignments are then obtained (e.g., via k-means) within each subspace.

We classify subspace-based methods into the following three categories based on how they learn and combine subspace representations:

- **Multi-Stage Methods.** *ISAAC* [180] and *MISC* [151] both build upon the idea of *independent subspace analysis* to discover subspaces that minimize redundancy across clustering solutions. Specifically, ISAAC applies Independent Subspace Analysis (ISA) to locate arbitrarily oriented subspaces that contain different object groupings, using the Minimum Description Length (MDL) principle to automatically select parameters. The identified subspaces are then clustered, yielding multiple distinct partitions. MISC

extends this by first finding statistically independent subspaces via ISA and determining their number again through the MDL principle. In a second stage, it adopts a graph-regularized semi-nonnegative matrix factorization, further enhanced with kernel functions to handle nonlinear separations. To improve on subspace quality and interpretability, *iMClusts* [124] integrates multi-head attention modules into deep autoencoders. This design not only controls the diversity between subspaces but also enforces *salience*, generating multiple distinct embeddings tailored to discover alternative meaningful clusterings.

- **Simultaneous Subspace Learning.** Unlike multi-stage or iterative procedures, some methods jointly learn multiple subspaces in a single optimization framework. *ENRC* [102] (Embedded Non-Redundant Clustering) introduces a neural network-based representation learning module and then (softly) assigns dimensions of the embedded space to different clusterings, thereby promoting non-redundancy among solutions. This approach is especially effective for complex data, such as images, where distinct subspaces may capture attributes like color, material, or shape. *scMCs* [123], proposed for single-cell multi-omics data integration, similarly learns multiple subspace embeddings in parallel. Each subspace captures either omics-specific or cross-omics features, reducing redundancy among subspaces and thus yielding multiple diverse clusterings. Through fusing transcriptomic and epigenetic representations, scMCs highlights alternative biological perspectives (e.g., different cell types or states) without sacrificing clustering quality.

2.3.2 Multi-view/Multi-source Clustering

To exploit data collected from multiple sources or views, multi-view (or multi-source) clustering techniques seek to learn a consensus (or complementary) partition by integrating information from each view. Below, we group related approaches into three categories according to how they leverage multi-view data:

Exploring the Complementary Information. One prominent theme in multi-view clustering is to explicitly enhance *complementarity* across views so that redundant information is reduced. *DiMSC* [22] leverages the Hilbert Schmidt Independence Criterion to boost multi-view subspace clustering, promoting diversity while constructing an affinity matrix across different views. *LMSC* [183] seeks a latent representation jointly learned from multiple views, making subspace reconstruction more robust. Similarly, the approach in [18] balances agreement across views while encouraging low-rankness and sparsity in a shared affinity matrix. By reducing view redundancy, these methods more effectively harness the unique perspectives each view provides.

Correlation-Based (Common Low-Dimensional Space). Another line of work aims to identify a *common low-dimensional representation* in which correlations between views are maximized. *Deep Canonical Correlation Analysis* (DCCA) [6] and its variants [150] learn deep networks that transform each view to maximize linear correlation in the shared latent space. *DMVSSC* [139] incorporates convolutional autoencoders and Canonical Correlation Analysis (CCA) to capture complementary features across multiple views, while [152] and [154] further extend these ideas to the incomplete-data or heterogeneous-network settings, showing that a carefully designed correlation-based embedding can reveal multiple consistent cluster structures across the views.

View-Specific Subspace Representations. Rather than merging views into a single common space, another strategy is to maintain *view-specific subspaces* and then integrate (or regularize) them to produce multiple complementary partitions. *RMSL* [89] learns hierarchical self-representations linked by a latent embedding to capture each view's subspace, while ensuring global consistency via backward encoding. *DMClusts* [153] iteratively factorizes multi-view data matrices into layered subspaces, generating multiple clusterings and minimizing redundancy through a novel distance-based regularizer. *scMCs* [123] applies this principle to single-cell multi-omics data, projecting each omic view into salient subspaces

that collectively uncover different but meaningful cellular groupings. *Fu et al.* [41] propose SCMC, a subspace-contrastive approach that uses autoencoders and contrastive strategies to enhance discriminability while uncovering complementary structures in heterogeneous views. *Xiong et al.* [162] tackle large-scale multi-view data with an anchor-graph technique and deep matrix factorization, decomposing data into multiple orthogonal subspaces to yield diverse high-quality clusterings efficiently. Similarly, *Xiong* [161] introduces an inverse optimization-based framework to strike a nuanced balance between clustering quality and diversity via a reverse encoder network, aligning shared latent representations with view-specific subspace features.

Chapter 3

DATA AUGMENTATION GUIDED DEEP MULTIPLE CLUSTERING

3.1 Background and Overview

Data augmentation [143] has been widely used as a technique to enhance the generalization performance. However, its ability to capture different aspects of the data for multiple clustering has been ignored. For example, no matter how we change the color of an image, the shape of the interested object is always preserved. Based on this observation, we propose to use different data augmentation methods to preserve different aspects of the data, which can also be used to capture a user's interest to guide the generation of different clusterings. Concretely, we propose a novel deep multiple clustering method called data Augmentation guided Deep Multiple Clustering (AugDMC), in which we leverage prototype-based self-supervised learning to obtain different data representations guided by different augmentation methods. Thereafter, representations from each data perspective can be fed to any single clustering method to obtain a clustering. We also propose a stable optimization strategy to ensure that the learned representations are robust when multiple data augmentations are applied. The main contributions of this chapter are highlighted as follows:

- We study a novel problem of multiple clustering to capture users' interests efficiently, which aims to control the aspect of clusterings via data augmentation. To the best of our knowledge, we are the first to study this problem.

- We propose AugDMC, a novel deep multiple clustering method guided by data augmentations. The proposed method uses a prototype-based self-supervised representation learning to obtain image representations for clustering and control the aspect of

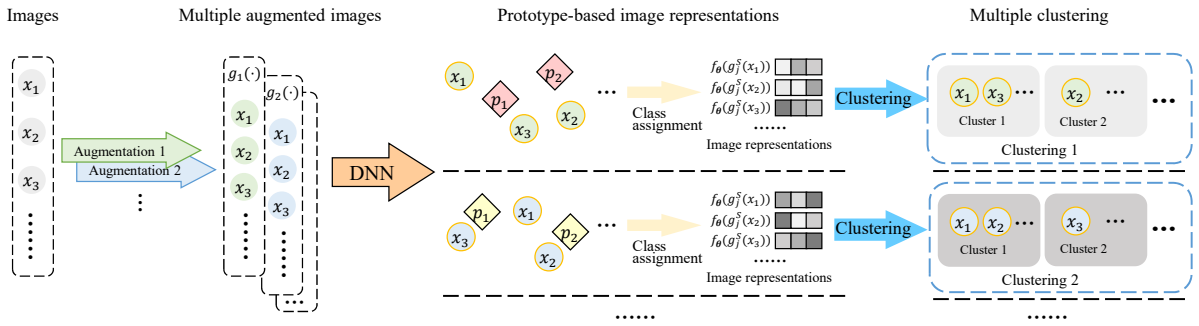


Figure 3.1: The framework of AugDMC. AugDMC uses multiple augmentation methods to obtain augmented images with desired characteristics. The representations of the augmented images are learned via a self-supervised prototype-based representation learning method. The final multiple clusterings can be obtained by employing any single clustering algorithm on the learned representations.

clusterings through data augmentations.

- The experimental results on three real-world datasets demonstrate the effectiveness of the proposed method, compared with the state-of-the-art methods.

3.2 Preliminaries

3.2.1 Problem Formulation

Given a dataset $\boldsymbol{X}$ with N samples and an augmentation method $g(\cdot)$, data augmentation guided multiple clustering aims to divide samples into $K(K > 1)$ clusterings $\{C^k\}_{k=1}^{K}$ with high quality and diversity. All the images in the datasets are augmented by the augmentation method $g(\cdot)$, denoted as $g(X)$. The augmentation guided multiple clustering is to generate the multiple clusterings through the augmented images.

In this chapter we focus on the setting where the user specifies in advance both (i) the desired number K of clustering aspects and (ii) a semantic description for each aspect (e.g., "color", "species", "rank", or "suit"). AugDMC does not attempt to discover these

aspects completely unsupervised. Instead, it uses user-specified augmentation families that are aligned with the target aspects and learns one clustering C^k per aspect. Although our experiments instantiate $K = 2$ on all benchmarks (e.g., color vs. species on fruit images, rank vs. suit on playing cards), the formulation is not restricted to two clusterings. In principle, any number $K > 2$ can be handled by AugDMC as long as the corresponding semantic aspects and augmentation families are provided. For example, on a face dataset such as CMU Face, one can simultaneously define clusterings by identity, pose, and illumination, and obtain three clusterings $\{C^k\}_{k=1}^{3}$ from a single run of AugDMC.

3.2.2 Augmentation

Shallow models

Data augmentation is essential to overcome the limitation of data samples [143]. Image augmentation has performed well in downstream tasks such as image classification [39, 104], image segmentation [105, 187], object detection [78, 196]. Traditional image data augmentation involves geometric transformation and photometric shifting. Flipping [146] reflects an image around its vertical or horizontal axis to double the number of images in a dataset. Rotation [134] rotates the image around an axis in either direction to generate images. Cropping [134] cuts and scales images to zoom in the original image. Color space shifting [157] shifts color space, e.g., RGB, CMY, YIQ, HSL, to generate images. Image filters [42] apply image processing techniques to augment images, e.g., histogram equalization, brightness adjustment, sharpening, blurring, and filters.

Deep models

Some data augmentation methods are based on deep learning. Deep learning image data augmentation comprises three main categories, i.e., generative adversarial networks (GAN), neural style transfer (NST), and meta metric learning. GAN-based methods use artificial images generated from the initial dataset to predict features of the images [181]. For the NST-

based methods, they separate and recombine images using neural representations of content and style, demonstrating a way to construct creative images computationally [46]. Meta metric learning methods generate images using models with meta-learning architecture [197]. A comprehensive review of image augmentation can be found in the surveys [132, 79].

Recently, some researchers attempted to combine clustering with augmentation. Guo et al. [60] introduced a clustering with a data augmentation method named DEC-DA, which optimizes an auto-encoder with augmented data with a clustering loss. ASPC-DA [59] fine-tunes an auto-encoder with augmented data with a self-paced strategy. Abavisani et al. [1] proposed a subspace clustering method using the augmented images and designed some efficient data augmentation policies. Although image augmentation has demonstrated its powerful ability in clustering, these methods are not designed for multiple clustering.

3.3 The Proposed Method

To find multiple clustering structures hidden in the data with the flexibility and efficiency to capture a user's interest, we propose a novel data augmentation guided deep multiple clustering method, named AugDMC.

3.3.1 Self-supervised prototype-based image representation

The first step of AugDMC is to learn image representations, which aims to learn a map $f_{\boldsymbol{\theta}}(x_i)$ without supervision, where $f_{\boldsymbol{\theta}}(\cdot)$ is a deep neural network, mapping image x_i to a d-dimensional feature representation $f_{\boldsymbol{\theta}}(x_i) \in \mathbb{R}^d$.

In this chapter, we propose a prototype-based latent class assignment strategy to learn the data representations. Specifically, let $\{\boldsymbol{p}_k\}_{k=1}^K$ indicate K prototypes that describe K anchors of the latent classes in the dataset. $\boldsymbol{p}_k \in \mathbb{R}^d$ is a d-dimensional prototype, corresponding to latent class k in the dataset. The similarity between image x_i and prototype k can be measured by the inner product between $f_{\boldsymbol{\theta}}(x_i)$ and $\boldsymbol{p}_k$ as $s_{ij} = \boldsymbol{p}_k^T \cdot f_{\boldsymbol{\theta}}(x_i)$. Thus, the

probability of image x_i belonging to latent class k can be described as

$$P(k|x_i;\boldsymbol{\theta}) = \frac{\exp\left(\boldsymbol{p}_k^T \cdot f_{\boldsymbol{\theta}}(x_i)/\tau\right)}{\sum_{k=1}^{K}\exp(\boldsymbol{p}_k^T \cdot f_{\boldsymbol{\theta}}(x_i)/\tau)}, \tag{3.1}$$

where τ is a temperature parameter that controls the scale of values, so as to control the concentration level of the probability distribution [66].

Therefore, considering all images in a dataset, the objective function of the proposed method is to maximize the joint probability as $\prod_{i=1}^{n} P(k|x_i;\boldsymbol{\theta})$, where n is the number of images in the dataset and $\boldsymbol{\theta}$ indicates the parameters of the deep neural network $f_{\boldsymbol{\theta}}(\cdot)$. The deep neural network used in AugDMC consists of multiple convolutional layers and a single fully-connected (FC) layer. Therefore, AugDMC is very flexible to employ these neural networks, such as ResNet [62], MobileNet [68], EfficientNet [137], etc.

3.3.2 Augmentation

AugDMC leverages augmentation to obtain images that could reflect different characteristics. Given an image x_i, it can be augmented by a function $g_j(\cdot)$, so the representation of augmented image $g_j(x_i)$ can be denoted as $f_{\boldsymbol{\theta}}\big(g_j(x_i)\big)$. Considering the prototype-based representation learning, Eqn. (3.1) can be rewritten as

$$P(k|x_i;\boldsymbol{\theta}) = \frac{\exp\left(\boldsymbol{p}_k^T \cdot f_{\boldsymbol{\theta}}\big(g_j(x_i)\big)/\tau\right)}{\sum_{k=1}^{K}\exp\left(\boldsymbol{p}_k^T \cdot f_{\boldsymbol{\theta}}\big(g_j(x_i)\big)/\tau\right)}. \tag{3.2}$$

Therefore, the joint probability of all the images has the formulation $\prod_{i=1}^{n} P(k|g_j(x_i);\boldsymbol{\theta})$. To sum up, prototype-based representation learning aims to learn discriminative representations, while augmentation provides the invariant property for a given perturbation. AugDMC could discover different aspects of representations from the combination.

3.3.3 Multiple Clusterings from Multiple Augmentations

Discrimination is essential for effective representation learning, which can be captured by our prototype-based image representation learning. However, images can be separated in

different ways. Therefore, we propose to do clustering that can aggregate similar ones according to the invariant property based on the augmentation. Specifically, given a set of augmentations $\{g_1, \ldots, g_J\}$, the invariant property corresponding to a certain augmentation is

$$\min_f \|f(g_j(x_i)) - f(x_i)\|_2. \tag{3.3}$$

With appropriate augmentations, we can learn multiple aspects of the data under different invariant properties. Thereafter, multiple clusterings can be realized by employing multiple augmentations.

However, one major challenge is to identify effective augmentations for multiple clusterings. Note that there are some prevalent properties for multiple clusterings, e.g., color, shape, etc [132, 79]. To identify which augmentations should be employed, one straightforward approach is to directly leverage standard augmentation for color invariant (e.g., color jitter), shape invariant (e.g., crop, rotation), etc. This is however very inefficient, since it is hard to choose the color and angle that should be used. AugDMC aims to learn augmentation setups simultaneously. Specifically, for color-invariant property, given a set of images, AugDMC extracts dominating colors from each image and then perturbs with extracted colors for a more effective color jitter augmentation. For shape-invariant property, which can be obtained by crop and rotation. AugDMC computes the pixel difference between original images from different angles of rotation and keeps those angles with the largest variance.

3.3.4 Stable Optimization Strategy

Although AugDMC can achieve multiple clusterings via multiple augmentations, it can suffer from an unstable learning procedure caused by augmentation [30]. In other words, multiple augmentations applied to a single image may result in unstable representations without label information. To address this problem, inspired by RandAug [30] that randomly selects a subset of augmentations at each iteration in the training process, we further design a stable optimization strategy for AugDMC. Specifically, we randomly draw a subset $\{g_j\}_{j=1}^{S}$ from

$\{g_j\}_{j=1}^{J}$, and then learn the prototype-based representations with the selected augmentations. Thus, the final joint probability can be written as

$$\prod_{i=1}^{n} P(k|g_j^S(x_i);\boldsymbol{\theta}) = \prod_{i=1}^{n} P(k|g_1(g_2(\ldots(g_S(x_i))));\boldsymbol{\theta}). \tag{3.4}$$

Since a stable clustering can be obtained when images can be separated well, we set the number of latent classes $K = n$, i.e., the number of images in the prototype-based representation learning process, to maximize representation discrimination. In addition, we repeat the process with a fixed number of epochs and only keep the feature extractors that achieve a satisfied performance, i.e., the accuracy or loss of the prototype-based learning is not changed in a fixed number of epochs. Then, multiple clusterings will be obtained using different feature extractors. It is worth noting that although AugDMC adopts a random strategy to select augmentation methods, it can still use different augmentation candidate sets to make AugDMC capture specific features. For example, using rotation, flip, or crop can capture color-related features, while using colorjitter or grey can capture shape-related features.

Note that the objective of AugDMC in Eqn. (3.4) is equivalent to minimizing the negative log-likelihood, so the final objective function of the proposed method is

$$l(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \log P(k|g_j^S(x_i);\boldsymbol{\theta}). \tag{3.5}$$

Thereafter, we can feed learned representations from each feature extractor to a clustering algorithm, such as k-means [97], to obtain a clustering result, where multiple feature extractors provide multiple clusterings. Fig. 3.1 summarizes the procedure of our proposal.

3.4 Experiments

In this section, we conduct experiments to demonstrate our proposal AugDMC. We first introduce our experimental setup, and then present empirical results in comparison to state-of-the-art baseline methods.

3.4.1 Experimental Setup

Datasets We conduct the experiments on three image datasets. First, the Fruit [70] dataset consists of 105 images and has two clusterings, i.e., species and color. Specifically, it contains three species (i.e., apple, banana, and grape) and three colors (i.e., green, red, and yellow). Second, Fruit360[1] dataset contains 4856 images and also has two clusterings, i.e., species (apple, banana, cherry, and grape) and color (red, green, yellow, and maroon). Different from Fruit [70], images in Fruit360 are with more classes, and thus more complex. Third, Card[2] is a dataset of playing card images, which consists of 8,029 images with two clusterings, i.e., rank (Ace, King, Queen, etc.) and suits (clubs, diamonds, hearts, spades).

Baselines The proposed AugDMC is compared with the following state-of-the-art methods: (1) MSC [70] is a traditional multiple clustering method, which considers the stability of clusterings and finds multiple clusterings by maximizing the Laplacian eigengap; (2) MCV [56] considers multiple different pre-trained feature extractors as different "views" of the same data, and designs a multi-input neural network to obtain a better clustering result; (3) ENRC [102] is a deep multiple clustering method, which combines auto-encoder and clustering objective function to obtain alternative clusterings; and (4) iMClusts [124] makes use of the expressive representational power of deep autoencoders and multi-head attention to achieve multiple clusterings.

Implementation Details AugDMC uses a common and efficient backbone, that is ResNet-18 [62], to do self-supervised representation learning. Several data augmentation methods such as 'RandomRotation' and 'RandomHorizontalFlip' that will not change the data's perspective are included in all experiments for the effectiveness of representation learning. For the clustering on color, we add data augmentation of 'RandomCrop' with a minimum size of half, in which the color perspective should be preserved. For the clustering on species, we

[1]https://www.kaggle.com/moltean/fruits

[2]https://www.kaggle.com/datasets/gpiosenka/cards-image-datasetclassification

Table 3.1: The multiple clusterings performance comparison. The best results are in bold.

Dataset	Clustering Type	MSC		MCV		ENRC		iMClusts		AugDMC	
		NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI
Fruit	Color	0.6886	0.8051	0.6266	0.7685	0.7103	0.8511	0.7351	0.8632	**0.8517**	**0.9108**
	Species	0.1627	0.6045	0.2733	0.6597	0.3187	0.6536	0.3029	0.6743	**0.3546**	**0.7399**
Fruit360	Color	0.2544	0.6054	0.3776	0.6791	0.4264	0.6868	0.4097	0.6841	**0.4594**	**0.7392**
	Species	0.2184	0.5805	0.2985	0.6176	0.4142	0.6984	0.3861	0.6732	**0.5139**	**0.7430**
Card	Order	0.0807	0.7805	0.0792	0.7128	0.1225	0.7313	0.1144	0.7658	**0.1440**	**0.8267**
	Suits	0.0497	0.3587	0.0430	0.3638	0.0676	0.3801	0.0716	0.3715	**0.0873**	**0.4228**

add data augmentation of 'ColorJitter', in which the species shape should be preserved.

The training process is optimized using Stochastic Gradient Descent (SGD). All hyperparameters are searched according to the loss of self-supervised representation learning, where learning rate is searched in $\{0.2, 0.1, 0.05, 0.01, 0.005, 0.0001\}$, weight decay is from$\{0.001, 0.0005, 0.0001, 0.00005\}$, and that of temperature τ is from $\{0.8, 0.85, 0.9, 0.95, 1.0\}$. We also set momentum as 0.9, and training epoch as 1000. Furthermore, AugDMC uses early stopping based on the accuracy of the prototype-based classification in the training process. After that, we can obtain image representations using the input of the last fully connected layer for clustering, which adopts k-means [97] in the following results. We evaluate the clustering performance compared to the ground truth using two commonly used metrics, that is, Normalized Mutual Information (NMI) [156] and Rand index (RI) [122]. We conduct the experiments with a GPU NVIDIA GeForce RTX 2080 Ti.

3.4.2 Performance Comparison

Table 3.1 compares the clustering performance between AugDMC and all other baselines. The best results are in bold. We can observe that the proposed method achieves the best results in all cases. These results demonstrate the effectiveness of the proposed method by capturing the concept of interest using a corresponding data augmentation method. Also, we

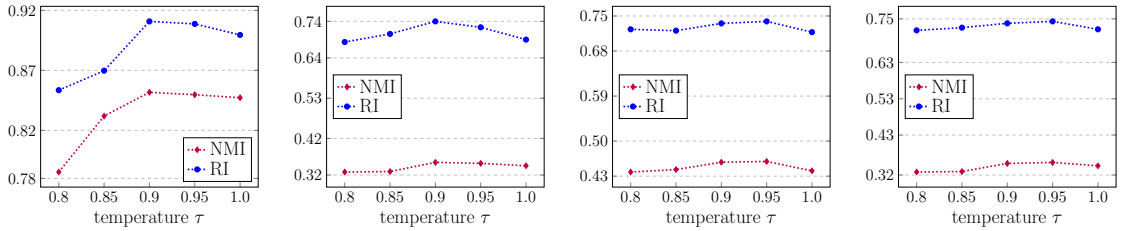
Table 3.2: Performance contribution of each component in AugDMC.

Method	Clustering Type	Fruit		Fruit360		Card	
		NMI	RI	NMI	RI	NMI	RI
AugDMC	Color	**0.8517**	**0.9108**	**0.4594**	**0.7392**	**0.1440**	**0.8267**
	Species	**0.3546**	**0.7399**	**0.5139**	**0.7430**	**0.0873**	**0.4228**
$\text{AugDMC}_{\text{wo}\tau}$	Color	0.8472	0.8995	0.4407	0.7177	0.1391	0.8003
	Species	0.3453	0.6901	0.5042	0.7212	0.0810	0.4029
$\text{AugDMC}_{\text{woS}}$	Color	0.8361	0.8979	0.4387	0.7119	0.1326	0.7892
	Species	0.3409	0.7017	0.4907	0.7285	0.0726	0.3849
$\text{AugDMC}_{\text{woS}\tau}$	Color	0.8273	0.8873	0.4302	0.7091	0.1261	0.7624
	Species	0.3389	0.6817	0.4850	0.6995	0.0687	0.3796
$\text{AugDMC}_{\text{woA}}$	Color	0.7172	0.8549	0.4064	0.6828	0.1057	0.7028
	Species	0.3084	0.6194	0.4249	0.6806	0.0642	0.3623
$\text{AugDMC}_{\text{woA}\tau}$	Color	0.7030	0.8456	0.3964	0.6842	0.0993	0.7256
	Species	0.3035	0.5881	0.4131	0.6775	0.0601	0.3609

can find that the deep multiple clustering models, i.e., MCV, ENRC, iMClusts and AugDMC, achieve better results than the shallow model, i.e., MSC, in most cases. This further confirms that deep multiple clustering methods have a more powerful ability in learning image representations to discover multiple clusterings. Besides, for the deep multiple clustering models, AugDMC achieves 7% to 24% improvement compared with the baselines, suggesting the effectiveness of the proposed method.

3.4.3 Ablation Study

To study the contribution of each component in AugDMC (i.e., the temperature parameter τ in prototype-based representation learning, data augmentation, and stable optimization strategy), we conduct an ablation study in this subsection. Specifically, we remove the above components from AugDMC and obtain three variants named $\text{AugDMC}_{\text{wo}\tau}$, $\text{AugDMC}_{\text{woA}}$, and $\text{AugDMC}_{\text{woS}}$, respectively. Note that we set $\tau = 1$ for $\text{AugDMC}_{\text{wo}\tau}$. Besides, we further re-



(a) Clustering by color on Fruit (b) Clustering by species on Fruit (c) Clustering by color on Fruit360 (d) Clustering by species on Fruit360

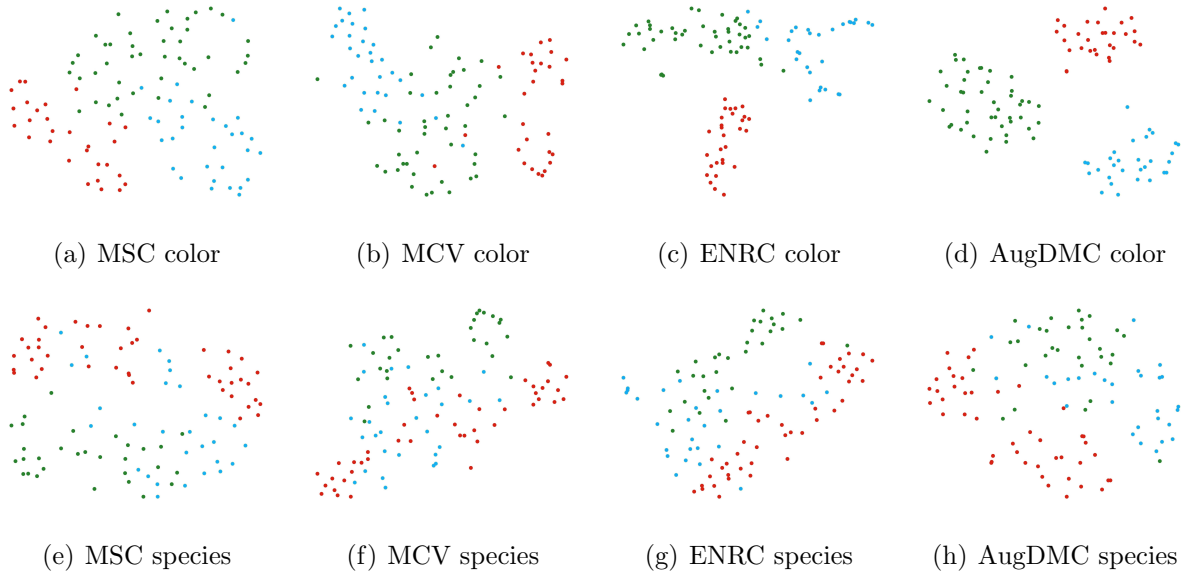
Figure 3.2: Results of parameter sensitivity of AugDMC.

move the combination of the temperature in prototype-based representation learning and stable optimization strategy, as well as the combination of the temperature in prototype-based representation learning and data augmentation, namely AugDMC$_{\text{woS}\tau}$ and AugDMC$_{\text{woA}\tau}$, respectively.

The results are shown in Table 3.2. We can find that AugDMC always achieves the best performance, indicating the effectiveness of prototype-based representation learning, data augmentation, and stable optimization strategy. Also, AugDMC$_{\text{woS}\tau}$ and AugDMC$_{\text{woA}\tau}$ perform worse than AugDMC$_{\text{wo}\tau}$, AugDMC$_{\text{woS}}$, and AugDMC$_{\text{woA}}$. This indicates that the combination of these components is useful for the proposed method.

3.4.4 Parameter Sensitivity

Moreover, we investigate the effect of the temperature τ in AugDMC on Fruit and Fruit360 datasets. The results of AugDMC under varying τ are shown in Fig. 3.2. With the increase of the value τ, the performance of AugDMC first improves and then drops on both datasets. This suggests that a better choice of temperature can further improve the performance, however the change is not big in most cases.



(a) MSC color (b) MCV color (c) ENRC color (d) AugDMC color

(e) MSC species (f) MCV species (g) ENRC species (h) AugDMC species

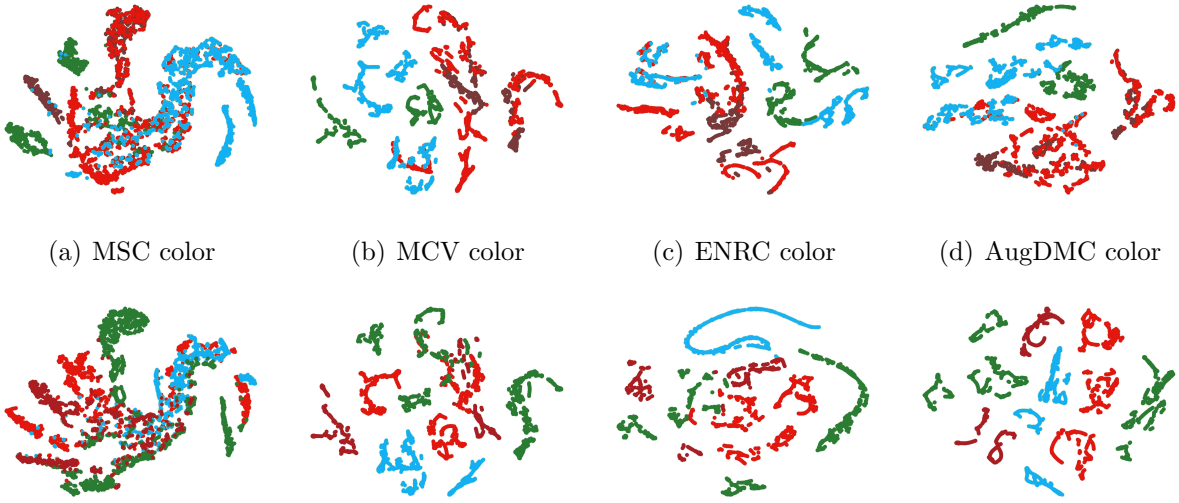
Figure 3.3: Visualization of image representations on the Fruit dataset. For the results of color clusterings, the red, blue, and green points indicate images with red, yellow, and green labels, respectively. For the results of species clusterings, the red, blue, and green points correspond to images with apple, banana, and grapes labels, respectively.

3.4.5 Visualization

To further demonstrate the effectiveness of the proposed method, in this subsection, we visualize the learned representations on the Fruit and Fruit360 datasets using t-SNE [142] to compare different methods. The results on the Fruit and Fruit360 datasets are shown in Figs. 3.3 and 3.4, respectively. Comparing these results, we can find that the representations with different labels learned by MSC, MCV, and ENRC are mixed with each other, while AugDMC distinguishes all the categories with a clearer boundary.

3.5 Summary

In this chapter, we study the problem of deep multiple clusterings for images and propose a novel augmentation guided method, named AugDMC, to flexibly and efficiently capture



(a) MSC color (b) MCV color (c) ENRC color (d) AugDMC color

(e) MSC species (f) MCV species (g) ENRC species (h) AugDMC species

Figure 3.4: Visualization of image representations on Fruit360 dataset. For the results of color clusterings, the red, blue, green, and maroon points signify images with red, yellow, green, and maroon labels, respectively. For the results of species clusterings, the images with apple, banana, cherry, and grape labels are marked by red, blue, green, and maroon points, respectively.

users' interests. Specifically, we perturb given images using augmentations to control the aspect to be clustered, where the corresponding image representations can be obtained through prototype-based representation learning with a stable optimization strategy. Experiments on three real-world datasets demonstrate the effectiveness of the proposed method. Our representation learning is independent from any clustering constraints, which makes the learned representations of different aspects flexible for other downstream tasks as a future direction.

Chapter 4

DUAL-DISENTANGLED DEEP MULTIPLE CLUSTERING

4.1 Background and Overview

Although deep multiple clustering methods have yielded impressive outcomes, they still confront two main challenges. Firstly, the relevance between the learned representations and the ultimate goal of distinct clusterings is weak. This issue arises because the diversity of clusterings is indirectly enforced by limiting the overlap of learned representations. However, this does not guarantee a direct correlation between the dissimilarity of feature representations and the clustering diversity, potentially leading to redundant clusterings. Secondly, most existing methods simply feed the learned representations into traditional clustering algorithms, such as k-means, to obtain multiple clusterings [124, 174]. However, the representations are often learned without involvement of the clustering objective, thereby undermining the final clustering outcomes. Although some efforts, like ENRC [102], have aimed to optimize the clustering performance, they have not yet achieved satisfactory results [124, 174].

Fortunately, disentangled representation learning, aiming to learn factorized representations that uncover and separate the latent factors hidden in data [86, 194, 50, 63, 12, 29], provides a very promising way to learn the diverse representations effectively for multiple clustering. Considering Fig. 6.1 as an example, the objects have two distinct factors, that is, shape and color. Disentangled representation learning can segregate these factors and encode them into independent and distinct latent variables within the representation space [65, 194]. Consequently, the latent variable of shape/color changes exclusively with the variation of the object's shape/color and remains constant relative to other factors. Interestingly, both disentangled representation learning and multiple clustering seek to learn distinct representations for latent factors, which prompts us to explore the application of

disentangled representation learning for multiple clustering. Nonetheless, despite the success of disentangled representation learning, no study to date has examined its use in achieving multiple clustering.

While disentangled representation learning provides a promising pathway towards multiple clustering, it still grapples with two substantial challenges. Firstly, the question arises of how to effectively learn diverse disentangled representations for multiple clustering. Datasets contain a multitude of latent factors. If these factors could be captured more efficiently, it would significantly aid in achieving multiple clustering. However, disentangled representation learning, despite its remarkable success, was not initially designed for multiple clustering. Therefore, the design of a disentangled representation learning framework specifically intended for multiple clustering becomes crucial. Secondly, the effectiveness in the purpose of clustering needs to be ensured. Most existing deep multiple clustering methods emphasize primarily on capturing features at the clustering level, thereby neglecting the effectiveness at the cluster level within each individual clustering. Once multiple clustering is accomplished, a proficient method should have the capability to simultaneously ensure clustering-level and cluster-level performance through an end-to-end approach, thereby leading to superior performance.

In this chapter, we study the problem of multiple clustering on image data and propose a novel Dual-Disentangled deep Multiple Clustering method named DDMC. Our method focuses on dealing with image data and consists of two main modules: disentanglement learning and cluster assignment, designed to tackle the two aforementioned challenges, respectively. Specifically, the disentanglement learning module leverages coarse-grained and fine-grained disentangled representations *to learn more diverse disentangled representations*. Within the coarse-grained disentanglement, we utilize deep augmentation to generate diverse augmented images. To further foster diversity, a Hilbert-Schmidt Independence Criterion (HSIC) constraint is applied among these augmented images. During the fine-grained disentanglement, these augmented images are processed through multiple variational autoencoders (VAEs) to discern the disentangled representations, thereby uncovering the multiple clusterings hidden

within the images. We also derive the Evidence Lower Bound (ELBO) of the fine-grained disentanglement to ensure the optimization of the proposed method in a principal manner. Simultaneously, the cluster assignment module is designed *to boost the effectiveness of the proposed method in cluster-level performance*. A cluster objective function is utilized during this phase to group similar images together, thereby forming distinct clusters of images with clear boundaries.

Moreover, we structure our approach as a variational Expectation-Maximization (EM) framework. In the Expectation (E) step, we decipher unique disentangled representations to reveal potential multiple clusterings, while the cluster assignment component is fixed. During the Maximization (M) step, the disentangled representations obtained from the E-step are fixed and then leveraged in the cluster assignment learning process. Therefore, DDMC is optimized by alternating between the E and M steps. Specifically, the contributions of this chapter can be summarized as follows.

- We propose a novel dual-disentangled deep multiple clustering method named DDMC. To the best of our knowledge, we are the first to introduce disentanglement learning for multiple clustering.

- DDMC consists of two parts, that is, disentanglement learning and cluster assignment, and can be optimized by utilizing a variational EM framework. During the E-step, the disentangled representation is learned, enabling the achievement of multiple clustering. In the M-step, cluster assignment is optimized, enhancing the cluster-level performance.

- Extensive experiments are conducted on seven commonly used tasks. The experimental results demonstrate the superiority of DDMC compared with state-of-the-art methods.

4.2 Preliminaries

4.2.1 Problem Formulation

Given a dataset $\boldsymbol{X}$ with N samples, disentangled multiple clustering aims to learn K set of representations for the dataset. The k-th set of representations is corresponding to the $K(K > 1)$ clusterings $\{C^k\}_{k=1}^K$. The k-th clustering, $C^k = (C_1^k, C_2^k, \ldots, C_{M_k}^k)$, is a partition of $\boldsymbol{X}$ into M_k groups (clusters), where $\cup_{m=1}^{M_k} C_m^k = \boldsymbol{X}$, and $C_{m_1}^k \cap C_{m_1}^k = \emptyset$ for $0 < m_1, m_2 < M_k$.

Throughout this chapter we assume that the user specifies in advance both (i) the number M of semantic clustering aspects of interest and (ii) a brief description of each aspect (e.g., "color", "species", "rank", "suit", "pose", or "identity"). DDMC does not try to automatically infer how many clusterings exist or name them. Instead, it learns K disentangled latent factors and aligns them with these M user-specified aspects via the assignment variables $\{\boldsymbol{c}^k\}_{k=1}^K$. While several of our benchmarks only provide two clusterings per image (e.g., Fruit, Fruit360, Card), the formulation is not restricted to $M = 2$. In particular, DDMC naturally handles datasets with three or more clusterings, such as CMUface, where we jointly model four aspects (identity, pose, glasses, and emotion) within the same framework.

4.2.2 Disentanglement learning

Disentangled representation learning seeks to learn factorized representations that uncover and separate the latent factors underlying data. Some methods concentrate on learning disentangled representations from image data. Some of these methods are based on VAE. For instance, Higgins et al. [65] proposed β-VAE, which learns interpretable factorized latent representations from raw image data without supervision. InfoGAN [28] uses the GAN framework and incorporates an extra variational regularization of mutual information to learn disentangled representations. InfoSwap [44] isolates identity-relevant and identity-irrelevant information by optimizing an information bottleneck to generate more identity-discriminative swapped faces. Disentangled representation learning is also widely used in natural language processing tasks. Hu et al. [72] integrated VAE with an attribute discriminator to disen-

tangle content and attributes of textual data, for generating texts with desired attributes of sentiment and tenses. Bao et al. [12] generated sentences from disentangled syntactic and semantic spaces by modeling syntactic information in the latent space of VAE and regularizing syntactic and semantic spaces via an adversarial reconstruction loss. Wang et al. [149] propose Iterative Partition based Invariant Risk Minimization (IP-IRM), an iterative algorithm based on the self-supervised learning fashion, to specifically learn disentangled representation.

Some methods are based on Generative Adversarial Networks (GAN). Jeon et al. [76] introduce IB-GAN, which compresses the representation by maximizing the mutual information between latent representation and input. This is an application of information bottleneck. Lin et al. [94] develop InfoGAN-CR, a self-supervised version of InfoGAN with a contrastive regularizer. This regularizer makes different dimensions in the latent representation more distinct, which helps with disentanglement. Zhu et al. [195] present PS-SC GAN, an extension of InfoGAN that uses Spatial Constriction (SC) to identify the focused areas of each latent dimension and Perceptual Simplicity (PS) to make the factors of variation in the latent representations simpler and purer. Wei et al. [155] propose an orthogonal Jacobian regularization (OroJaR) to enforce disentanglement for generative models. They use the Jacobian matrix of the output with respect to the input (i.e., latent variables for representation) to measure how the output changes when the input varies.

4.3 The Proposed Method

To simultaneously learn representations for distinct clusterings and achieve good cluster-level performance, we leverage disentangled representation learning and cluster assignment learning within a variational EM framework. The disentangled representation learning process has two main components, i.e., coarse-grained and fine-grained disentanglement. The learned disentangled representations are then fed into the cluster assignment module to improve the cluster-level performance as illustrated in Fig. 4.1. The details are described as follows.

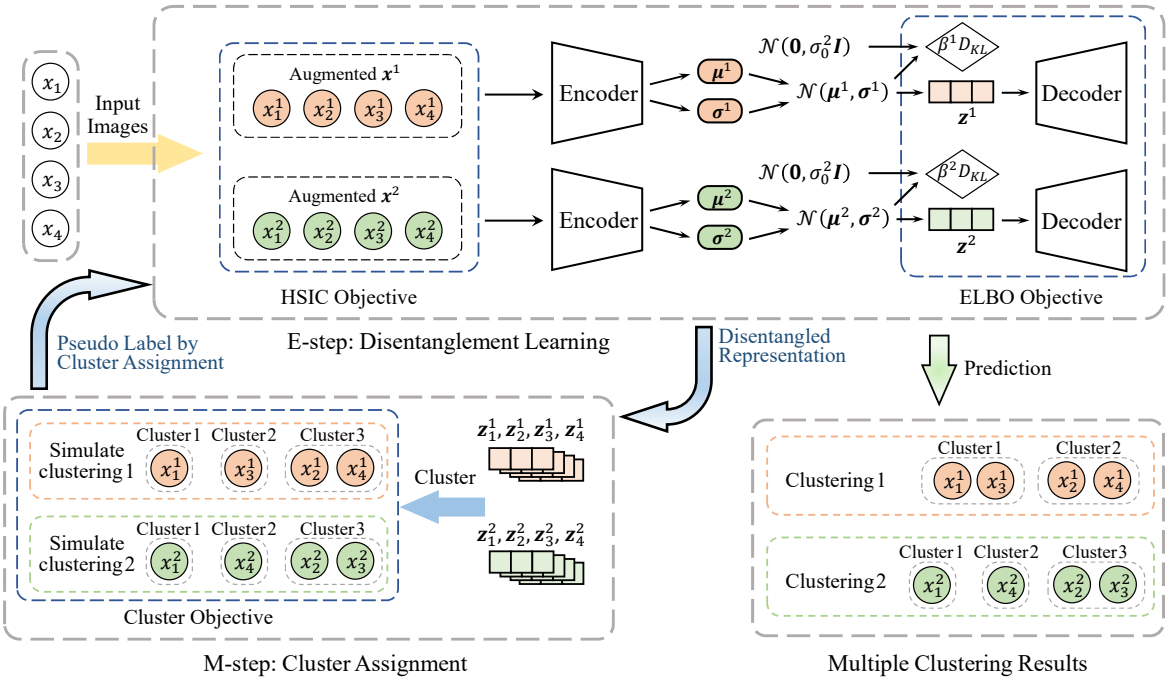


Figure 4.1: DDMC framework. The DDMC framework trains disentanglement learning and cluster assignment in an EM framework. During the E-step, the disentangled representation is learned through both coarse-grained and fine-grained disentangled representation learning. The learned disentangled representations can be applied to multiple clustering tasks. In the M-step, cluster assignment is optimized, enhancing the cluster-level performance.

4.3.1 Task Statement

Given an image $\boldsymbol{x}_i \in \{\boldsymbol{x}_i\}_{i=1}^N$, our aim in disentangled multiple clustering is to derive K distinct image representations $\{\boldsymbol{z}_i^1, \ldots, \boldsymbol{z}_i^K\}$, which describe various facets of the image through achieving both coarse and fine-grained disentanglement. We assume that the user specifies M semantic aspects of interest, and that one clustering is produced for each aspect. Consequently, these images can be classified into M distinct clusterings, each reflecting a unique aspect of the original image, where K can be larger than M. This is because the real-world data may have more aspects than the desired number of aspects. We need to disentangle all

the aspects to obtain the needed representations.

Coarse-grained disentanglement

An image encapsulates diverse aspects, each of which can correspond to a clustering perspective. To effectively uncover these latent facets, we strive for coarse-grained disentanglement achieved through augmentation. Consequently, employing a variety of augmentation methods generates variant images that each mirror a distinct facet of the original image, thereby highlighting the inherent diversity of its elements.

Fine-grained disentanglement

To better obtain diverse representations, we further employ fine-grained disentanglement to decipher the factorized representation of the augmented images, denoted as $\{\boldsymbol{z}_i^k\}_{k=1}^K$ with $\boldsymbol{z}_i^k$ indicating the k-th high-level aspect for image $\boldsymbol{x}_i$. Additionally, we infer a set of one-hot vectors, $\{\boldsymbol{c}^k\}_{k=1}^K$, for each high-level aspect. Assuming the images can be grouped into M clusterings, the dimension of $\boldsymbol{c}^k$ is equal to the number of clusterings, i.e., $\boldsymbol{c}^k = [c_1^k, c_2^k, \ldots, c_M^k]$. If the k-th disentangled representation falls under the clustering m, $c_m^k = 1$ and $c_{m'}^k = 0$ for all $m' \neq m$, ensuring each representation is associated with a single clustering. In the evaluation, if $K > M$, we can select the disentangled representation with the highest correlation according to $\{c_m^k\}_{k=1}^K$ as the feature representations for the m-th clustering, while the remaining can be disregarded.

4.3.2 The Coarse-Grained Disentanglement

Uncovering multiple independent structures hidden in data is a fundamental challenge for multiple clustering. Fortunately, data augmentation can facilitate the extraction of diverse image features. Consequently, we propose a coarse-grained disentanglement as an integral component of our DDMC. This is specifically designed to glean insights from augmented images, enabling us to better characterize and capture the various desired facets inherent

within the image data.

The coarse-grained disentanglement process operates via two primary stages: sampling and mixing. In the sampling stage, augmentation operations are stochastically sampled from a defined set of augmentation operations. This sampling procedure is repeated K times, each corresponding to a unique disentangled representation set. During the k-th sampling iteration, the chosen augmentation methods are applied to the image set $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ to produce the augmented images, denoted as $\boldsymbol{X}_{aug}^k$.

A prevalent issue in deep learning with corrupted data is that applying multiple augmentations to an image can make its representation unstable if there is no label information [30]. To alleviate this issue, we leverage the mixing process that generates a variety of transformations, which are instrumental for robustness. Specifically, for the k-th coarse-grained disentanglement, the augmented images can be generated by

$$\boldsymbol{X}^k = w^k \boldsymbol{X} + (1 - w^k)\boldsymbol{X}_{aug}^k \tag{4.1}$$

where $w^k \in (0, 1)$ is the mixing weight. To further distinguish the K augmented image sets, we set w^k as trainable parameters and use Hilbert Schmidt Independence Criterion (HSIC) [55] to measure the dependency between two augmented image sets. For two augmented image sets $\boldsymbol{X}^k$ and $\boldsymbol{X}^{k'}$, the empirical HSIC is computed as

$$\text{HSIC}(\boldsymbol{X}^k, \boldsymbol{X}^{k'}) = (N-1)^2 \,\text{tr}\left(\boldsymbol{G}_k \boldsymbol{H} \boldsymbol{G}_{k'} \boldsymbol{H}\right) \tag{4.2}$$

where $\text{tr}(\cdot)$ is the matrix trace operator and $\boldsymbol{H} = \boldsymbol{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ is the centering matrix. $\boldsymbol{G}_k \in \mathbb{R}^{N \times N}$ is the kernel matrix to measure the similarity within the image set $\boldsymbol{X}^k$. Here we adopt the inner product kernel for simplicity, so we have

$$\boldsymbol{G}_k = (\boldsymbol{X}^k)^\top \boldsymbol{X}^k \tag{4.3}$$

A smaller HSIC value signifies greater independence between the two augmented image sets. Hence, we can maximize the negative HSIC value across the K augmented image sets to learn a more diverse augmentation manner. This diversity is beneficial for learning embeddings

that effectively capture various aspects of images. Therefore, the objective function for the coarse-grained disentanglement can be defined as

$$\begin{aligned}\mathcal{L}_{\text{coarse}} &= -\sum_{k=1,k\neq k'}^{K} \text{HSIC}\left(\boldsymbol{X}^k, \boldsymbol{X}^{k'}\right) \\ &= -(N-1)^2 \sum_{k=1,k\neq k'}^{K} \text{tr}\left(\boldsymbol{X}_k^\top \boldsymbol{X}_k \boldsymbol{H} (\boldsymbol{X}^{k'})^\top \boldsymbol{X}^{k'} \boldsymbol{H}\right)\end{aligned} \tag{4.4}$$

4.3.3 The Fine-Grained Disentanglement

The coarse-grained disentanglement increases the diversity of images. But only relying on these could not obtain effective image representations for multiple clustering, so we propose fine-grained disentanglement that can directly obtain feature representations to represent different aspects based on VAE, so that the latent factors in dataset can be discovered in this process. Here, we introduce the aspect assignment relations for images: $\boldsymbol{C} = \{\boldsymbol{c}^k\}_{k=1}^K$, where $\boldsymbol{c}^k = [c_1^k, c_2^k, \ldots, c_M^k]$ is a M dimensional vector. Each element $c_m^k \in \boldsymbol{c}^k$ is described by a distribution $p(c_m^k)$, which denotes the probability of the disentangled representation $\boldsymbol{Z}^k = \{\boldsymbol{z}_i^k\}_{i=1}^N$ corresponding to the m-th clustering. Thereafter, we consider the following joint probability:

$$\begin{aligned}p(\boldsymbol{X}^k, \boldsymbol{Z}^k, \boldsymbol{c}^k) &= p(\boldsymbol{X}^k|\boldsymbol{Z}^k, \boldsymbol{c}^k) p(\boldsymbol{Z}^k, \boldsymbol{c}^k) \\ &= p(\boldsymbol{X}^k|\boldsymbol{Z}^k, \boldsymbol{c}^k) p(\boldsymbol{Z}^k) p(\boldsymbol{c}^k)\end{aligned} \tag{4.5}$$

Given that both $\boldsymbol{c}^k$ and $\boldsymbol{Z}^k$ are dependent on the augmented images $\boldsymbol{X}^k$, the posterior of $\boldsymbol{c}^k$ and $\boldsymbol{Z}^k$ are written as $p(\boldsymbol{c}^k|\boldsymbol{X}^k)$ and $p(\boldsymbol{Z}^k|\boldsymbol{X}^k)$, respectively. The integral of the posterior in VAE is intractable, so we employ $q_{\phi^k}(\boldsymbol{c}^k|\boldsymbol{X}^k)$ and $q_{\phi^k}(\boldsymbol{Z}^k|\boldsymbol{X}^k)$, which are parameterized by ϕ^k, as approximations for the true posterior. Since the aspect assignment $\boldsymbol{c}^k$ is a one-hot representation, which is non-differentiable during the training process, we set its prior as a product of independent uniform Gumbel Softmax distributions, i.e., $p(\boldsymbol{c}^k) = p(c_1^k)p(c_2^k)\ldots p(c_M^k)$, where $p(c_m^k) \sim Gumbel(0, 1)$. Hence, the approximate posterior $q_{\phi^k}(\boldsymbol{c}^k|\boldsymbol{X}^k)$ can be described

as

$$\begin{aligned} q_{\phi^k}(\boldsymbol{c}^k|\boldsymbol{X}^k) &= \prod_{m=1}^{M} q_{\phi^k}(c_m^k|\boldsymbol{X}^k) \\ &= \prod_{m=1}^{M} \frac{\exp\big((\log s_k + g_m^k)/\tau\big)}{\sum_{i=1}^{K} \exp\big((\log s_i + g_m^k)/\tau\big)} \end{aligned} \tag{4.6}$$

where $g_m^k \sim Gumbel(0,1)$ and τ is the temperature parameter exploited to control the scale of values. The shared trainable parameter $s_k \in \{s_1, s_2, \ldots, s_K\}$ is employed to generate all the aspect assignment parameters, namely $\{\boldsymbol{c}^1, \ldots, \boldsymbol{c}^K\}$, in a principle manner. Conversely, the posterior $q_{\phi^k}(\boldsymbol{Z}^k|\boldsymbol{X}^k)$ can be parameterized by a factorized Gaussian as

$$q_{\phi^k}(\boldsymbol{Z}^k|\boldsymbol{X}^k) = \prod_{i=1}^{N} q_{\phi^k}(\boldsymbol{z}_i^k|\boldsymbol{X}^k) \tag{4.7}$$

Assuming that the prior of the disentangled representation $\boldsymbol{Z}^k$ follows a normal distribution, that is, $p(\boldsymbol{Z}^k) \sim \mathcal{N}(\boldsymbol{0}, (\sigma_0^k)^2\mathbf{I})$. According to reparameterization trick [80], the encoder $q_{\phi^k}(\boldsymbol{z}_i^k|\boldsymbol{X}^k) = \mathcal{N}(\boldsymbol{\mu}^k, (\boldsymbol{\sigma}^k)^2)$ can be computed utilizing a neural network $f_{nn} : \mathbb{R}^d \to \mathbb{R}^{2d}$ as

$$\boldsymbol{a}^k, \boldsymbol{b}^k = f_{nn}(\boldsymbol{x}^k),\ \boldsymbol{\mu}^k = \boldsymbol{a}^k,\ \boldsymbol{\sigma}^k \leftarrow \sigma_0^k \cdot \exp(-\frac{1}{2}\boldsymbol{b}^k) \tag{4.8}$$

The neural network $f_{nn}(\cdot)$ captures nonlinearity from the data, and is shared across the K disentangled representations. In practical applications, we typically set σ^k to a relatively small value, such as around 0.2 to prevent the inferred values from becoming excessively large.

The objective function of fine-grained disentanglement is to maximize the likelihood function of the augmented images. According to Jensen's inequality, the log-likelihood of our proposed model has the following formulation:

$$\begin{aligned} \sum_{k=1}^{K} \log p(\boldsymbol{X}^k) &= \sum_{k=1}^{K} \log \int_{\boldsymbol{Z}^k} \sum_{\boldsymbol{c}} p(\boldsymbol{X}^k, \boldsymbol{Z}^k, \boldsymbol{c}^k)\, \mathrm{d}\boldsymbol{Z}^k \\ &\geq \sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{Z}^k, \boldsymbol{c}^k|\boldsymbol{X}^k)} \left[\log \frac{p(\boldsymbol{X}^k, \boldsymbol{Z}^k, \boldsymbol{c}^k)}{q(\boldsymbol{Z}^k, \boldsymbol{c}^k|\boldsymbol{X}^k)} \right] \\ &= \sum_{k=1}^{K} \mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{X}^k) \end{aligned} \tag{4.9}$$

where $\mathcal{L}_{\text{ELBO}}(\boldsymbol{X}^k)$ is the Evidence Lower Bound (ELBO) of the k-th disentangled representation. Maximizing the ELBO is equivalent to maximizing the likelihood in the variational inference process. Given $p(\boldsymbol{X}^k, \boldsymbol{Z}^k, \boldsymbol{c}^k) = p(\boldsymbol{X}^k|\boldsymbol{Z}^k, \boldsymbol{c}^k)p(\boldsymbol{Z}^k, \boldsymbol{c}^k)$, the ELBO of the k-th disentangled representation is formulated as:

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{X}^k) = \mathbb{E}_{q(\boldsymbol{Z}^k, \boldsymbol{c}^k|\boldsymbol{X}^k)}[\log p(\boldsymbol{X}^k|\boldsymbol{Z}^k, \boldsymbol{c}^k)] - D_{KL}(q(\boldsymbol{Z}^k, \boldsymbol{c}^k|\boldsymbol{X}^k)\|p(\boldsymbol{Z}^k, \boldsymbol{c}^k)) \tag{4.10}$$

Assuming that the aspect assignment and disentangled representation are conditionally independent, i.e., $q(\boldsymbol{Z}^k, \boldsymbol{c}^k|\boldsymbol{X}^k) = q(\boldsymbol{Z}^k|\boldsymbol{X}^k)q(\boldsymbol{c}^k|\boldsymbol{X}^k)$ and the prior $p(\boldsymbol{Z}^k, \boldsymbol{c}^k) = p(\boldsymbol{Z}^k)p(\boldsymbol{c}^k)$. The KL divergence can be factored into two parts: $D_{KL}(q(\boldsymbol{Z}^k|\boldsymbol{X}^k)\|p(\boldsymbol{Z}^k))$ as well as $D_{KL}(q(\boldsymbol{c}^k|\boldsymbol{X}^k)\|p(\boldsymbol{c}^k))$. In this way, the KL divergence terms of $\boldsymbol{c}^k$ and $\boldsymbol{z}^k$ are separated, which correspond to aspect assignment and disentangled representations, respectively. For the k-th disentangled representation, the objective function is to maximize the following formulation:

$$\mathcal{L}_{ELBO}(\boldsymbol{X}^k) = \mathbb{E}_{q(\boldsymbol{Z}^k, \boldsymbol{c}^k|\boldsymbol{X}^k)}[\log p(\boldsymbol{X}^k|\boldsymbol{Z}^k, \boldsymbol{c}^k)] - D_{KL}\big(q(\boldsymbol{Z}^k|\boldsymbol{X}^k)\|p(\boldsymbol{Z}^k)\big) - D_{KL}\big(q(\boldsymbol{c}^k|\boldsymbol{X}^k)\|p(\boldsymbol{c}^k)\big) \tag{4.11}$$

4.3.4 Expectation (E) step: Disentanglement Optimization

Optimizing the disentangled representations primarily involves fine-tuning the fine-grained disentanglement. To make our model capture more variation information, the channel capacity of the KL divergence terms in (4.11) should increase gradually. We define controlled capacities U_c and U_z for the KL divergence terms of the aspect assignment variable $\boldsymbol{c}^k$ and the disentangled representation $\boldsymbol{Z}^k$ for the k-th disentangled representation, respectively. Therefore, the ELBO can be reformulated as

$$\begin{aligned}\mathcal{L}_{ELBO}(\boldsymbol{X}^k) =& \mathbb{E}_{q(\boldsymbol{Z}^k, \boldsymbol{c}^k|\boldsymbol{X}^k)}[\log p(\boldsymbol{X}^k|\boldsymbol{Z}^k, \boldsymbol{c}^k)] \\ & -\beta^k|D_{KL}\big(q(\boldsymbol{Z}^k|\boldsymbol{X}^k)\|p(\boldsymbol{Z}^k)\big) - U_z| \\ & -\beta^k|D_{KL}\big(q(\boldsymbol{c}^k|\boldsymbol{X}^k)\|p(\boldsymbol{c}^k)\big) - U_c|\end{aligned} \tag{4.12}$$

where β^k is a trade-off coefficient. Considering that different disentangled representation sets have different scales of data reconstruction loss, the β^k can be calculated by:

$$\beta^k = \beta \frac{\mathbb{E}_{q(\boldsymbol{Z}^k, \boldsymbol{c}^k | \boldsymbol{X}^k)}[\log p(\boldsymbol{X}^k | \boldsymbol{Z}^k, \boldsymbol{c}^k)]}{\max_k \mathbb{E}_{q(\boldsymbol{Z}^k \boldsymbol{c}^k | \boldsymbol{X}^k)}[\log p(\boldsymbol{X}^k | \boldsymbol{Z}^k, \boldsymbol{c}^k)]} \tag{4.13}$$

Eventually, the fine-grained disentangled objective function can be written as:

$$\begin{aligned} \mathcal{L}_{\text{fine}} =& \sum_{k=1}^{K} \mathcal{L}_{\text{ELBO}}(\boldsymbol{X}^k) \\ =& \sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{Z}^k, \boldsymbol{c}^k | \boldsymbol{X}^k)} \left[\log p \left(\boldsymbol{X}^k \mid \boldsymbol{Z}^k, \boldsymbol{c}^k \right) \right] \\ & - \sum_{k=1}^{K} \beta^k \left| D_{KL} \left(q \left(\boldsymbol{Z}^k \mid \boldsymbol{X}^k \right) || p \left(\boldsymbol{Z}^k \right) \right) - U_z \right| \\ & - \sum_{k=1}^{K} \beta^k \left| D_{KL} \left(q \left(\boldsymbol{c}^k \mid \boldsymbol{X}^k \right) || p(\boldsymbol{c}^k) \right) - U_c \right| \end{aligned} \tag{4.14}$$

4.3.5 Maximization (M) step: Cluster Assignment

Following the optimization of disentanglement, we further enhance the clustering performance of our method by optimizing the following objective function [167]:

$$\mathcal{L}_{\text{cluster}} = - \max_{\mathbf{s}} \frac{1}{N} \sum_{i=1}^{N} \left\| \boldsymbol{z}_i^k - \mathbf{W}^k s_i \right\|_2^2 \text{ s.t. } \boldsymbol{s}_i \in \{0,1\}^T, \mathbf{1}^\top \boldsymbol{s}_i = 1 \tag{4.15}$$

where $\boldsymbol{W}^k \in \mathbb{R}^{d \times T}$ represents a matrix consisting of all cluster centers and we assume that all the clusterings have T clusters. $\boldsymbol{s}_i$ is the assignment vector of the i-th example which has only one non-zero element. $\mathbf{1}$ signifies a column vector with every element to 1. In the training process, we employ k-means to initialize the cluster centers $\boldsymbol{W}^k$ and optimize s_i, and the $\boldsymbol{W}^k$ is held constant to prevent a degenerate solution where all examples converge to a single point, resulting in the objective equaling zero. As a result of $\boldsymbol{W}^k$ being constant, decision boundaries are also established. This is because the decision boundaries act as perpendicular bisectors of adjacent cluster centers, rendering it impossible to aggregate all examples together by optimizing the given Eq. (4.15). Therefore, when $\boldsymbol{z}_i^k$ and $\boldsymbol{W}^k$ in

Eq. (4.15) are fixed, s_i can be updated as follows

$$s_{ij} \leftarrow \begin{cases} 1, & \text{if} \quad j = \arg\min_t \|\boldsymbol{z}_i^k - \boldsymbol{w}_t^k\|_2 \\ 0, & \text{otherwise} \end{cases} \tag{4.16}$$

where s_{ij} is the j-th element of $\boldsymbol{s}_i$ and $\boldsymbol{w}_t^k$ denotes the centroid of the t-th cluster for the k-th disentangled represenetation.

The training process will cease if the change in predicted cluster labels between two consecutive iterations is less than a specified threshold, δ. Formally, the stopping criterion can be described as follows

$$1 - \frac{1}{n} \sum_{i,j} s_{ij}^e \cdot s_{ij}^{e-1} < \delta \tag{4.17}$$

where s_{ij}^{e-1} and s_{ij}^e are indicators for whether example x_i is assigned to the j-th cluster at the $(e-1)$-th and e-th iteration, respectively. We empirically set $\delta = 0.0005$ in our experiments. Thereafter, the cluster objective can make the learned representation more suitable for downstream tasks. Thereafter, the cluster assignment objective will be a constraint in the training process, so the learned image representations will be more suitable for k-means tasks.

To summarize, the total loss function of our method is

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{coarse}} + \mathcal{L}_{\text{fine}}}_{\text{E-step}} + \underbrace{\mathcal{L}_{\text{cluster}}}_{\text{M-step}} \tag{4.18}$$

In the optimization process, we iteratively perform the E-step and the M-step. During the E-step, we learn K distinct disentangled representations to uncover potential multiple clusterings. In the M-step, the disentangled representations obtained in the E-step are utilized in the cluster assignment learning process, where each set of representations is grouped into T clusters. The knowledge acquired from the cluster assignments serves as a constraint in the subsequent E-step training, ensuring that the learning process benefits from the clustering information.

Table 4.1: Dataset Statistics.

Datasets	# Samples	# Dimensions	# Clusters
ALOI	288	287	2;2
Card	8,029	50,176	13;4
CMUface	640	15,360	4;20;2;4
Fruit	105	119,025	3;3
Fruit360	4,856	10,000	4;4
StickFig	900	400	3;3
C-MNIST	10,000	1,568	10;10

4.4 Experiments

4.4.1 Experimental Setup

Datasets

To demonstrate our proposed method, we evaluate the clustering performance of our DDMC on seven image benchmark datasets in multiple clustering as follows, whose detailed statistics are also summarized in Table 4.1. (1) ALOI [47] (Amsterdam Library of Object Images) contains images of 1000 common objects. Following setting in [124], we sample 288 images of four objects with two clusterings: color (yellow and red) and shape (circle and cylinder). (2) Card[1] is a dataset of playing card images, which consists of 8,029 images with two clusterings, i.e., rank (Ace, King, Queen, etc.) and suits (clubs, diamonds, hearts, spades). (3) CMUface [57] contains 640 32×30 gray images, which can be grouped according to pose (left, right, straight and up), identity (20 individuals), glass (with or without glass), and emotions (angry, happy, neutral and sad). (4) Fruit [70] consists of 105 images and has two clusterings, i.e., species (apples, bananas, and grapes) and color (green, red, and yellow).

[1]https://www.kaggle.com/datasets/gpiosenka/cards-image-datasetclassification

(5) Fruit360[2] contains 4,856 images and has two clusterings as well, i.e., species (apples, bananas, cherries, and grapes) and color (red, green, yellow, and maroon). (6) StickFig [57] has 900 20×20 images and two clusterings, i.e., upper body and lower body. Each clustering has three clusters according to the body postures. (7) MNIST [84] is a well known dataset. Here, we extend it through concatenating two digits side by side, leading to a total of 100 possible combinations, named C-MNIST. As a result, the dataset can be seen as containing two clusterings, i.e., left and right. Each of the clusterings has 10 clusters.

Baselines

We compare DDMC against the following state-of-the-art methods, including two single clustering methods and six multiple clustering methods: (1) DAC [25] is a deep single clustering method that adopts a pairwise classification framework to learn image representations for clustering. (2) DCN [167] learns feature representations through a clustering constraint, making the representations more suitable for single clustering tasks. (3) MSC [70] is a traditional multiple clustering method for finding multiple clusterings, which aims to maximize the Laplacian eigengap and ensure the stability of the clusterings. (4) MCV [56] leverages multiple pre-trained feature extractors to represent different "views" of the same data, and employs a multi-input neural network to enhance clustering outcome. (5) ENRC [102] is a deep multiple clustering method, which integrates auto-encoder and clustering objective function to generate different clusterings. (6) iMClusts [124] makes use of the expressive representational power of deep autoencoders and multi-head attention to accomplish multiple clusterings. (7) AugDMC [174] is a deep multiple clustering method, which leverages augmentations to learn different image representations to achieve multiple clustering. (8) β-VAE [65] is a disentangled representation method based on VAE to learn distinct representation in an unsupervised manner. Here we employ this method to acquire disentangled representations directly and apply it to multiple clustering tasks.

[2]https://www.kaggle.com/moltean/fruits

Table 4.2: Quantitative comparison. The best results are in bold.

		Single Clustering				**Multiple Clustering**													
Datasets	Clusterings	DAC		DCN		MSC		MCV		ENRC		iMClusts		AugDMC		β-VAE		DDMC	
		NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑
Fruit	Color	0.6579	0.7893	0.6724	0.7935	0.6886	0.8051	0.6266	0.7685	0.7103	0.8511	0.7351	0.8632	0.8517	0.9108	0.8329	0.8611	**0.8973**	**0.9383**
	Species	0.2154	0.6206	0.2058	0.6127	0.1627	0.6045	0.2733	0.6597	0.3187	0.6536	0.3029	0.6743	0.3546	0.7399	0.3287	0.6562	**0.3764**	**0.7621**
Fruit360	Color	0.1967	0.5568	0.2197	0.5899	0.2544	0.6054	0.3776	0.6791	0.4264	0.6868	0.4097	0.6841	0.4594	0.7392	0.4354	0.7043	**0.4981**	**0.7472**
	Species	0.1533	0.5439	0.1685	0.5583	0.2184	0.5805	0.2985	0.6176	0.4142	0.6984	0.3861	0.6732	0.5139	0.7430	0.4289	0.6982	**0.5292**	**0.7703**
Card	Order	0.0532	0.6392	0.0612	0.6893	0.0807	0.7805	0.0792	0.7128	0.1225	0.7313	0.1144	0.7658	0.1440	0.8267	0.1205	0.7329	**0.1563**	**0.8326**
	Suits	0.0269	0.3350	0.0416	0.3604	0.0497	0.3587	0.0430	0.3638	0.0676	0.3801	0.0716	0.3715	0.0873	0.4228	0.0728	0.5536	**0.0933**	**0.6469**
StickFig	Upper	0.3781	0.5964	0.3873	0.6390	0.6293	0.7293	0.5387	0.6896	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8226	0.9033	1.0000	1.0000
	Lower	0.3695	0.6009	0.3651	0.6338	0.6431	0.7149	0.5160	0.6524	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.8261	0.9084	1.0000	1.0000
ALOI	Shape	0.2172	0.5153	0.3839	0.7391	0.2968	0.5199	0.7359	0.8261	0.9732	0.9861	0.9963	0.9989	1.0000	1.0000	0.9706	0.9887	1.0000	1.0000
	Color	0.1520	0.3893	0.2083	0.5820	0.1563	0.3428	0.6982	0.7439	0.9833	0.9892	1.0000	1.0000	1.0000	1.0000	0.9754	0.9892	1.0000	1.0000
CMUface	Emotion	0.0612	0.5157	0.0811	0.5107	0.1284	0.6736	0.1433	0.5268	0.1592	0.6630	0.0422	0.5932	0.0161	0.5367	0.1549	0.6580	**0.1726**	**0.7593**
	Glass	0.0439	0.4692	0.0535	0.4791	0.1420	0.5745	0.1201	0.4905	0.1493	0.6209	0.1929	0.5627	0.1039	0.5361	0.1897	0.6225	**0.2261**	**0.7663**
	Identity	0.4196	0.7653	0.4912	0.7932	0.3892	0.7326	0.4637	0.6247	0.5607	0.7635	0.5109	0.8260	0.5876	0.8334	0.4535	0.6873	**0.6360**	**0.8907**
	Pose	0.2184	0.5524	0.2306	0.5596	0.3687	0.6322	0.3254	0.6028	0.2290	0.5029	0.4437	0.6114	0.1320	0.5517	0.3882	0.6831	**0.4526**	**0.7904**
C-MNIST	Left	0.0857	0.5235	0.1038	0.5283	0.0167	0.6273	0.1326	0.5603	0.7263	0.7882	0.7736	0.8250	0.9364	0.9569	0.8085	0.9105	**1.0000**	**1.0000**
	Right	0.0828	0.5185	0.0985	0.5534	0.0542	0.6003	0.1103	0.5938	0.7277	0.7926	0.7698	0.8119	0.9277	0.9208	0.8123	0.8907	**1.0000**	**1.0000**

Implementation Details

ResNet-18 [62] is adopted as the encoder and decoder in our implementation. Several data augmentation methods are utilized in the experiments, such as "RandomRotation", "RandomHorizontalFlip", "RandomCrop", "ColorJitter", and so on. We employ Adam and set momentum as 0.9 to train the model for 1000 epochs. All hyperparameters are searched according to the loss score of DDMC, where learning rate $\in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005\}$, weight decay $\in \{0.0005, 0.0001, 0.00005, 0.00001, 0\}$, temperature $\tau \in \{0.8, 0.85, 0.9, 0.95, 1.0\}$.

We perform k-means [97] and evaluate the clustering performance using two quantitative metrics, including Normalized Mutual Information (NMI) [156] and Rand index (RI) [122]. These measures range in $[0, 1]$, and higher scores imply more accurate clustering results. The experiments are conducted with a single GPU NVIDIA GeForce RTX 2080 Ti.

4.4.2 Performance Comparison

We compare our proposed DDMC with state-of-the-art multiple clustering baselines. Table 4.2 presents the average clustering results, highlighting the best results in bold. Our method, DDMC, consistently outperforms the state-of-the-art methods in all cases, thereby indicating the superiority of the proposed method. Based on the clustering results, we obtain the following conclusions:

The single clustering methods, namely DAC and DCN, typically underperform in comparison to their multiple clustering counterparts. This is an expected outcome as single clustering methods struggle to discern distinct clusters within datasets. Furthermore, DCN often outperforms DAC. This can be attributed to the fact that DCN leverages an optimized cluster objective function, which aids DCN in identifying a more suitable approach to data grouping. Besides, From our observations, disentangled-based methods, specifically β-VAE and DDMC, generally yield superior results. Despite β-VAE not being tailored for multiple clustering, it delivers strong performance due to its capability to disentangle latent factors within datasets. This underscores the efficacy of disentangled representation learning within multiple clustering scenarios. Furthermore, DDMC outperforms β-VAE, lending credibility to the effectiveness of DDMC. Moreover, Both AugDMC and DDMC are methods that employ augmentation to learn diverse representations. These methods outperform other techniques in most cases, suggesting that data augmentation can preserve distinct aspects of the data and subsequently aid in discovering diverse features in multiple clustering tasks. Furthermore, The DDMC method surpasses AugDMC in most cases, largely owing to its more sophisticated and robust design that incorporates fine-grained disentangled learning and cluster assignment. Notably, AugDMC does not perform as well on the CMUface dataset. This is likely because the clusters in this dataset are related to detailed features such as emotion and glasses, which are challenging to capture with augmentation methods. Despite this, the proposed DDMC method continues to deliver the best results, thanks to its fine-grained disentangled representation and cluster assignment components.

Table 4.3: Components ablation. The best results are in bold.

Datasets	Clusterings	DDMC$_{wo\text{mix}}$		DDMC$_{w=0.5}$		DDMC$_{wo\text{CD}}$		DDMC$_{wo\text{CA}}$		DDMC$_{wo\text{CD\&CA}}$		DDMC	
		NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑
Fruit	Color	0.8854	0.9013	0.8912	0.9010	0.8796	0.8993	0.8869	0.8981	0.8621	0.8824	**0.8973**	**0.9383**
	Species	0.3610	0.7389	0.3695	0.7429	0.3592	0.7268	0.3557	0.7369	0.3428	0.7007	**0.3764**	**0.7621**
Fruit360	Color	0.4887	0.7315	0.4923	0.7399	0.4754	0.7209	0.4862	0.7277	0.4633	0.7203	**0.4981**	**0.7472**
	Species	0.5218	0.7494	0.5254	0.7584	0.5037	0.7353	0.5138	0.7508	0.4909	0.7285	**0.5292**	**0.7703**
Card	Order	0.1482	0.7927	0.1527	0.8217	0.1308	0.7671	0.1513	0.8285	0.1253	0.7523	**0.1563**	**0.8326**
	Suits	0.0899	0.6326	0.0918	0.6382	0.0779	0.5982	0.0896	0.6297	0.0730	0.5508	**0.0933**	**0.6469**
StickFig	Upper	1.0000	1.0000	1.0000	1.0000	0.8568	0.9226	1.0000	1.0000	0.8364	0.9186	1.0000	1.0000
	Lower	1.0000	1.0000	1.0000	1.0000	0.8490	0.9094	1.0000	1.0000	0.8436	0.8922	1.0000	1.0000
ALOI	Shape	0.9912	0.9956	1.0000	1.0000	0.9891	0.9921	1.0000	1.0000	0.9885	0.9931	1.0000	1.0000
	Color	0.9927	0.9952	1.0000	1.0000	0.9853	0.9956	1.0000	1.0000	0.9798	0.9889	1.0000	1.0000
CMUface	Emotion	0.1698	0.7452	0.1709	0.7483	0.1707	0.7382	0.1698	0.7436	0.1674	0.7045	**0.1726**	**0.7593**
	Glass	0.2205	0.7598	0.2249	0.7621	0.2253	0.7568	0.2243	0.7514	0.2262	0.7208	**0.2261**	**0.7663**
	Identity	0.5806	0.8570	0.6014	0.8792	0.5294	0.7932	0.6317	0.8823	0.4736	0.7617	**0.636**	**0.8907**
	Pose	0.4492	0.7723	0.4510	0.7855	0.4381	0.7360	0.4427	0.7806	0.4022	0.7183	**0.4526**	**0.7904**
C-MNIST	Left	1.0000	1.0000	1.0000	1.0000	0.8562	0.9253	1.0000	1.0000	0.8259	0.9020	1.0000	1.0000
	Right	1.0000	1.0000	1.0000	1.0000	0.8495	0.9190	1.0000	1.0000	0.8305	0.8994	1.0000	1.0000

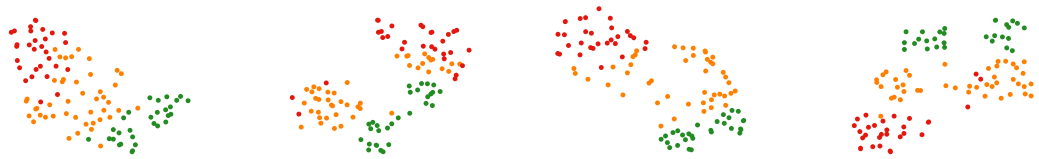
4.4.3 Ablation Study

We conduct the ablation study to assess the contribution of different components within DDMC, i.e., the coarse-grained disentangled learning and the cluster assignment. Specifically, to verify the effect of the mixing process in the coarse-grained disentangled learning, we remove the mixing process and set $w = 0.5$ to validate the effect of the trainable parameter w. Consequently, we can obtain two related variants referred to as DDMC$_{wo\text{mix}}$ and DDMC$_{w=0.5}$, respectively. Moreover, we also generate three additional variants by removing the coarse-grained disentangled representation, the cluster assignment component, or both. These variants are referred to as DDMC$_{wo\text{CD}}$, DDMC$_{wo\text{CA}}$, and DDMC$_{wo\text{CD\&CA}}$, respectively.

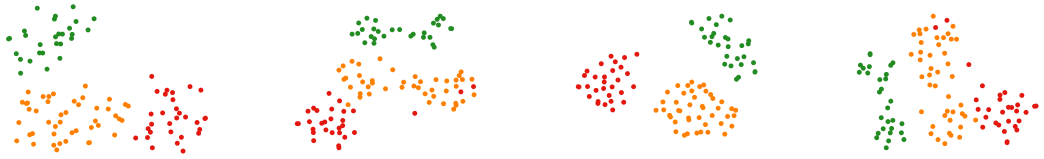
The results are illustrated in Table 4.3. Across all cases, DDMC demonstrates superior performance, underscoring the effectiveness of both the coarse-grained disentangled representation and cluster assignment components. Additional observations include: DDMC$_{w=0.5}$

outperforms DDMC$_{wo\text{mix}}$, which indicates the efficacy of the mixing process within the coarse-grained disentanglement. Also, the only difference between DDMC$_{wo\text{mix}}$ and DDMC$_{wo\text{CD}}$ is the augmentation process. DDMC$_{wo\text{mix}}$ achieves better results than DDMC$_{wo\text{CD}}$ in most cases, implying that augmentation is advantageous in learning diverse representations. However, DDMC$_{wo\text{CD}}$ outperforms both DDMC$_{wo\text{mix}}$ and DDMC$_{w=0.5}$ on the emotion and glass clustering within the CMUface dataset. This can be attributed to the difficulty in capturing intricate information through augmentation. Moreover, DDMC outperforms both DDMC$_{wo\text{CD}}$ and DDMC$_{wo\text{CA}}$, confirming the efficacy of coarse-grained disentanglement and cluster assignment, respectively. This finding is echoed in the comparison between DDMC and DDMC$_{wo\text{CD\&CA}}$. Consequently, both the coarse-grained disentanglement and cluster assignment components enhance the effectiveness of the proposed method.

4.4.4 Visualization



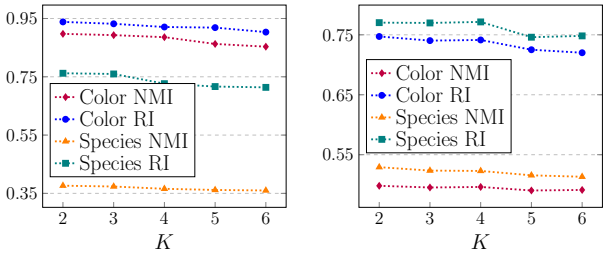
(a) DDMC epoch 50 (b) DDMC$_{wo\text{CA}}$ epoch 50 (c) DDMC epoch 300 (d) DDMC$_{wo\text{CA}}$ epoch 300



(e) DDMC epoch 800 (f) DDMC$_{wo\text{CA}}$ epoch 800 (g) DDMC epoch 1000 (h) DDMC$_{wo\text{CA}}$ epoch 1000

Figure 4.2: Visualization of DDMC and DDMC$_{wo\text{CA}}$ color representations on the Fruit dataset.

We also visualize the disentangled representations in different training epochs for DDMC



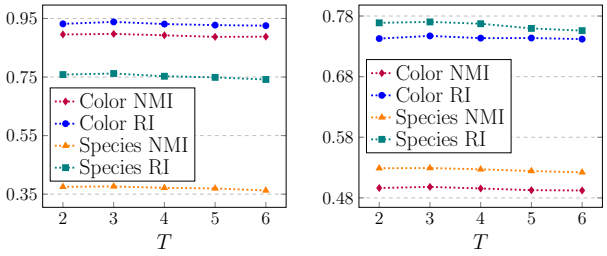
(a) Clusterings on Fruit (b) Clusterings on Fruit360

Figure 4.3: Results of parameter sensitivity of K.

and its variant DDMC$_{woCA}$ to compare their training process on Fruit using t-SNE [142]. The visualizations of color clustering of DDMC and DDMC$_{woCA}$ are shown in Fig. 4.2. We use red, yellow, and green points to denote images with red, yellow, and green labels, respectively. As the number of epochs increases, data points from different categories progressively segregate from the mixture, consequently forming more distinct boundaries. Notably, compared to DDMC$_{woCA}$, DDMC can establish clearer demarcations between data points from different categories and foster a more compact distribution within the same category. These results further attest to the effectiveness of cluster assignment.

4.4.5 Parameter Analysis

We further investigate the effect of the number of disentangled representations K and the number of clusters T of DDMC. We change K and T from 2 to 6. The results of DDMC on Fruit and Fruit360 datasets under varying K and T are shown in Fig. 4.3 and Fig. 4.4, respectively. Regarding parameter K, DDMC performs best when $K = 2$ for both Fruit and Fruit360 datasets, and the performance decreases as K increases, indicating that $K = 2$ is the optimal choice for the datasets. Specifically, for the color clustering of the Fruit dataset, performance experiences a significant downturn when K reaches 5 or 6. This can be attributed to the introduction of excessive noise with an overly large K, thereby adversely



(a) Clusterings on Fruit (b) Clusterings on Fruit360

Figure 4.4: Results of parameter sensitivity of T.

affecting performance. Furthermore, for the parameter T, the performance of DDMC initially improves before declining as T augments, with $T = 3$ generating the best outcomes. These findings imply that an appropriate selection of the K or T value can enhance the effectiveness of DDMC.

4.4.6 Efficiency Analysis

Here we analyze the efficiency of the deep multiple clustering methods. The experiments are conducted on a server with a GPU GeForece RTX 2080Ti. We test the running time on the Fruit dataset and the running time and related performance of color clustering are shown in Fig. 4.5. The two fastest methods are DAC and DCN, that is because both of them are single clustering learning methods and their module structure are simpler than the other multiple clustering methods. For the other methods, we can find methods with longer running time usually have better performance. DDMC can learn more effective image representations with acceptable efficiency.

4.5 Summary

In this chapter, we present DDMC, a novel Dual-Disentangled deep Multiple Clustering method that leverages disentangled representations for multiple clustering. DDMC employs

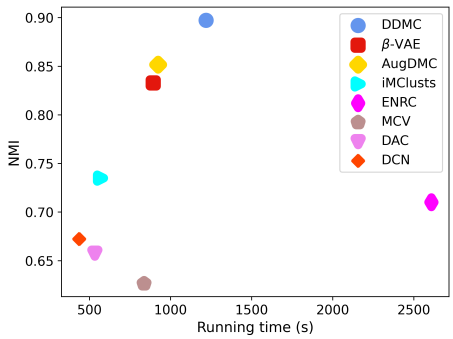


Figure 4.5: Performance v.s. the running time (s) on Fruit dataset.

coarse-grained and fine-grained disentangled representations to reveal and disentangle the latent factors hidden in data. In addition, it incorporates a cluster assignment module to further enhance the effectiveness and robustness of multiple clusterings in cluster-level performance. Furthermore, we formulate our method as a variational Expectation-Maximization framework and derive the Evidence Lower Bound of the fine-grained disentanglement. Extensive experiments on seven benchmark datasets demonstrate that DDMC attains state-of-the-art performance in terms of multiple clustering performance and each individual clustering's performance. In future work, we intend to extend our methodology to manage more complex data types and scenarios, such as multi-modal data.

Chapter 5

MULTI-MODAL PROXY LEARNING TOWARDS PERSONALIZED VISUAL MULTIPLE CLUSTERING

5.1 Background and Overview

Multiple clustering [8, 70] algorithms have been developed to generate different partitions for varying applications, demonstrating the ability to identify multiple distinct clusterings from a dataset. Contemporary advancements in the field reveal a growing inclination among researchers to integrate deep learning methodologies for facilitating multiple clustering outcomes. Predominantly, such techniques capitalize on auto-encoders and data augmentation processes to capture a broad spectrum of feature dimensions, thereby enhancing the performance of multiple clustering [102, 124, 174]. However, a common issue arises as users often do not require all the clusterings generated by the algorithm, and identifying the relevant ones necessitates a substantial understanding of each clustering result. Therefore, in this work, we initiate an exploration into a method that is adept at accurately capturing and reflecting a user's interests. Users typically express their interests through concise keywords (e.g., color or species), and aligning these with different visual components precisely is challenging. Fortunately, the advent of multi-modal models like CLIP [121] that aligns images to their corresponding text descriptions, can be helpful to fill this gap. However, unlike methods that employ labeled data to fine-tune pre-trained models [45, 148], multiple clustering frequently deals with environments marked by vague or undefined label categories and amounts. Consequently, given only a high-level concept from the user, it is infeasible to fine-tune the pre-trained models to capture a specific aspect of the data, without the detailed labels corresponding to the user's concept.

An intuitive strategy to integrate pre-trained models into clustering is the application of

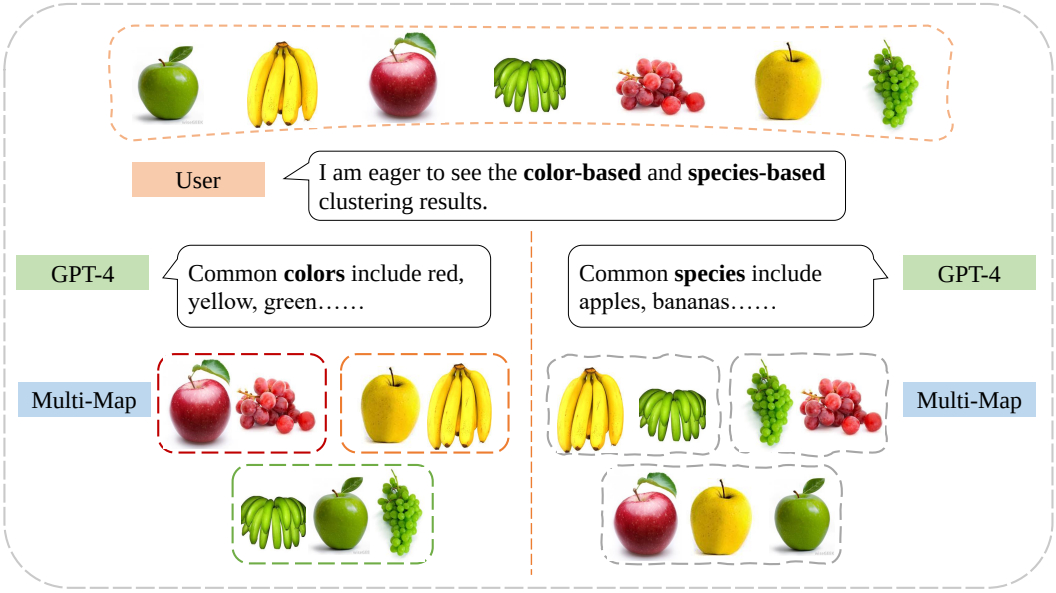


Figure 5.1: The flow chart of Multi-MaP. Multi-MaP obtains multiple clustering results based on the high-level concepts from users and the reference words from GPT-4.

zero-shot feature extractions, followed by clustering of the resultant embeddings. However, this approach exhibits limitations, particularly in capturing the interests of users within the dataset. Taking the multi-modal model CLIP [121] as an example, when feeding image data into CLIP, regardless of what aspects of clustering the user expects, CLIP can only produce the same embeddings. Even considering the scenario that different pre-trained models can capture different aspects of the same data as in [56], it is hard to tell which one matches a user's preference. Fortunately, given CLIP's ability to model image-text pairs collaboratively, we can use a user's high-level concept to trigger the corresponding feature extraction from the pre-trained encoders from CLIP. However, no previous work has studied if CLIP has the potential to uncover different aspects of images, which is the focus of this work.

Specifically, we propose to integrate a user's high-level concept describing the preference using a personalized textual prompt. For example, if a user's focus pertains to the color dimension of fruit, a prospective prompt might be formulated as "a fruit with the color of

", wherein the "" placeholder represents the proxy word awaiting determination using the knowledge in CLIP. Thereafter, we can learn the proxy word embedding by maximizing the similarity between the image and text embedding. However, the proxy word embedding is now searched in a continuous space while the original CLIP uses discrete tokens, which can downgrade the performance. We prove that the performance can be well guaranteed by selecting the nearest token as the reference, which is however unavailable in a clustering task.

Fortunately, we can use the user's high-level concept as a reference, which however covers a broad range of tokens under its scope. Therefore, we propose to leverage large language models like GPT-4 to generate candidate tokens using the user's high-level concept, in which the closest can be used as a closer reference token. We formalize these queries through a simple prompt template that takes the dataset-level object name and the user-specified concept as inputs, as detailed in Sec. 5.3.1. Furthermore, in some scenarios, users may provide multiple concepts to obtain multiple clusterings simultaneously as shown in Fig. 5.1, we can also introduce a negative loss with these contrastive concepts to further enhance the learning. Therefore, to capture a user's specific interest and discover personalized clustering structure hidden in the data, we propose a multi-modal proxy learning method (Multi-MaP). Multi-MaP incorporates both text prompts and unlabeled images from the clustering task, and leverages CLIP to acquire their respective personalized representations using both reference word and concept-level constraints.

The contributions of this work can be summarized as

- We are the first to explore a deep multiple clustering method that can precisely capture a user's high-level interest(s) and generate personalized clustering(s) accordingly.

- We propose a novel multi-modal proxy learning method, Multi-MaP, by leveraging the text and image encoders pre-trained by CLIP, where the user's interests can be captured by the personalized text prompts.

- Considering the challenge of learning a word proxy in a continuous space while tokens in CLIP were discrete, we theoretically prove that a close reference token can help constrain the search, which motivates the proposed reference word constraint and concept-level constraint.

- Finally, to the best of our knowledge, we are the first who demonstrate that CLIP can uncover different semantic aspects of images.

5.2 Preliminaries

5.2.1 Problem Formulation

Given a dataset $\boldsymbol{X}$ with N samples and users' interests $\{\boldsymbol{u}\}_{k=1}^{K}$, multi-modal proxy learning for multiple clustering aims to learn the representations for the images w.r.t. the users' interests. According to the learned representations, the dataset can be divided into $K(K > 1)$ clusterings $\{C^k\}_{k=1}^{K}$ with high quality and diversity. The k-th clustering, $C^k = (C_1^k, C_2^k, \ldots, C_{M_k}^k)$, is a partition of $\boldsymbol{X}$ into M_k clusters corresponding to k-th users' interests $\boldsymbol{u}_k$.

5.2.2 Multi-modal Model

Multi-modal learning refers to the process of learning representations from different types of input modalities, such as image data, text, or speech. Here we introduce two main parts of multimodal deep learning, i.e, images supporting language models and models for both modalities. Images supporting language models explain how CV models can benefit from natural language as an extra source of supervision. These models are expected to be more powerful than models that only use manual labels because they have more information in the training data. A notable example of this is the CLIP model [121], which uses a new dataset called WIT that contains 400 million text-image pairs from the internet. Besides, Data2vec [10] is a multimodal self-supervised model that uses a single framework to process speech, natural language, or visual information. This is different from earlier models that

used different algorithms for different modalities. The main idea of data2vec is to predict latent representations of the input data based on a masked view of the input, using a self-distillation setup and a standard transformer architecture. The main improvement is in the framework itself, not the architectures used. For example, the transformer architecture [145] follows Vaswani et al. Transformers have several advantages over RNNs/CNNs, especially when using large amounts of data, making them the standard approach in vision-language modeling. Flamingo [5] is a state-of-the-art few-shot learning model with 80B parameters, which is much more than the other two models. It has a large language model in its architecture, which gives it the ability to generate text for open-ended tasks.

As related, we focus on how vision models can benefit from natural language supervision. CLIP [121] is a notable model, which is trained with a dataset containing 400 million text-image pairs from the internet. The objective is to align images to their corresponding text using contrastive learning. Fine-tuning approaches adapt vision-language models like CLIP to specific downstream image recognition tasks. CoOp [192] and CLIP-Adapter [45] exemplify this, with the latter integrating residual style feature blending to enhance performance on various visual classification tasks. Additionally, insights from TeS [148], further elucidate the effectiveness of fine-tuning strategies in leveraging natural language supervision for enhanced visual understanding. Recognizing the scarcity of labeled data for various tasks, significant research efforts have been dedicated to enhancing zero-shot learning. Some approaches extend beyond CLIP by incorporating other large pre-trained models. For instance, VisDesc [101] harnesses the power of GPT-3 to generate comprehensive contextual descriptions corresponding to given class names, thereby demonstrating superior performance compared to CLIP's basic prompts. UPL [73] and TPT [133] leverages unlabeled data to optimize learnable input text prompts. InMaP [119] recovers the proxy of each class in the vision space with the help of the text proxy. All of these methods aim to improve the performance of vision classification tasks, while clustering is a different scenario in that we do not have class names that we can exploit to extract useful information from CLIP. Despite CLIP has shown its powerful performance in many tasks, it is unknown how to apply it to

multiple clustering tasks.

Here we briefly review the training objective in CLIP. Given a set of image-text pairs as $\{x_i, t_i\}_{i=1}^n$, where x_i is an image and t_i is the corresponding text description, their vision and text representations can be obtained by two encoders as $\mathbf{x}_i = f(x_i)$ and $\mathbf{t}_i = h(t_i)$. $f(\cdot)$ and $h(\cdot)$ are vision and text encoders for optimization, where $\mathbf{x}_i$ and $\mathbf{t}_i$ have the unit norm. Then, these two encoders can be learned by minimizing the contrastive loss as

$$\min_{f,h} \sum_i -\log \frac{\exp(\mathbf{x}_i^\top \mathbf{t}_i/\tau)}{\sum_j \exp(\mathbf{x}_i^\top \mathbf{t}_j)/\tau)} - \log \frac{\exp(\mathbf{t}_i^\top \mathbf{x}_i/\tau)}{\sum_j \exp(\mathbf{t}_i^\top \mathbf{x}_j)/\tau)}$$

where τ is the temperature. This contrastive loss aims to pull the image and its description together while pushing away the irrelevant text [118], which enables the emerging multi-modal applications, e.g., zero-shot transfer [121, 119], text-to-image generation [129], etc.

5.3 The Proposed Method

5.3.1 Multi-modal Proxy Learning

Given the pre-trained vision and text encoders from CLIP, this work takes one step further to investigate if we can extract user-specific information from the alignment between images and text.

Concretely, given an image of fruit [70] as shown in Fig. 6.1, some users may be interested in only one specific property of the object, e.g., color. In this scenario, applying the vision encoder to extract the representation for the whole image can miss the preferences of users. To mitigate the problem, we propose to explore the proxy representation from the image with the guidance from the text using users' preference, named Multi-Modal Proxy learning (Multi-MaP).

Recall that CLIP is pre-trained by images and text descriptions, where the text prompt is "a photo of a fruit" for an image containing "fruit". Now given a user's preference (e.g., color), we can rewrite the prompt as "fruit with the color of *" denoted by t_i^* for image x_i, where "*" is the proxy word and its text embedding is $\mathbf{w}_i$ that is learnable. More generally,

for each dataset we first define a short noun phrase describing the object category (e.g., "fruit", "flower", "car", "playing card", "face"). Given a user-specified high-level concept u (e.g., "color", "species", "model", "rank", "suit", "pose", "emotion"), we instantiate a natural-language prompt template that links the object and the concept and contains a single slot "*" for the proxy word. For attribute-style concepts (such as color, pose, or suit) we use prompts of the form

$$t_i^* = \text{"a photo of a } \texttt{[object]} \text{ with the } \texttt{[concept]} \text{ of *"},$$

where `[object]` and `[concept]` are replaced by the dataset object name and the concept name, respectively. For category-style concepts (such as species, identity, or model) we instead use

$$t_i^* = \text{"a photo of a * } \texttt{[object]}\text{"}.$$

Then, we can align image and text representations to obtain the appropriate proxy embedding for users' interests. Since there are no negative pairs, only the similarity between positive pairs can be optimized as

$$\mathcal{L}(\mathbf{w}_i) = -\langle f(x_i), h(t_i^*) \rangle \tag{5.1}$$

where the vision and text encoders are frozen and $\mathbf{w}_i$ is the only variable for learning as the representation of the proxy word. By maximizing the similarity to the image representation, Multi-MaP aims to learn the optimal text proxy according to the users' interests.

However, it should be noted that the text encoder was pre-trained with discrete text tokens, while the domain of $\mathbf{w}_i$ in Eqn. 5.1 is unconstrained. Therefore, the text representation extracted from the frozen text encoder can be inaccurate for $\mathbf{w}_i$ that degenerates the performance, which is demonstrated in the following theory.

For the sake of simplicity, we assume $h'(t) \in \mathbb{R}$ is defined on the whole set but only has the right estimation on a discrete set as $T = \{t_i\}$ and the counterpart with the unconstrained set is denoted as $H(w)$. According to the definition, we have $\forall t \in T, h'(t) = H(t)$. The

gap between the estimation from h' and H on unconstrained variable w can be depicted as follows.

Theorem 5.1 *Given $w \notin T$ and $t \in T$, if assuming h' and H are L_h and L_H-Lipschitz continuous, we have*

$$\|h'(w) - H(w)\|_2 \leq (L_h + L_H)\|t - w\|_2$$

Proof 5.1 *According to the definition, we have*

$$\begin{aligned}
&\|h'(w) - H(w)\|_2 = \|h'(w) - h'(t) + h'(t) - H(w)\|_2 \\
&\leq \|h'(w) - h'(t)\|_2 + \|h'(t) - H(w)\|_2 \\
&= \|h'(w) - h'(t)\|_2 + \|H(t) - H(w)\|_2 \\
&\leq (L_h + L_H)\|t - w\|_2
\end{aligned}$$

Remark Theorem 5.1 implies that the distance of the estimation $h'(w)$ to the ground-truth result $H(w)$ is bounded by that of w to an arbitrary discrete token t. Therefore, by selecting the nearest token as the reference, the bound can be improved as shown in the following corollary.

Corollary 5.2 *With the assumptions in Theorem 5.1 and letting $t' = \arg\min_i \|t_i - w\|_2$, we have*

$$\|h'(w) - H(w)\|_2 \leq (L_h + L_H)\|t' - w\|_2$$

Concept-level Constraint

According to the above analysis, a good reference t' can help significantly guarantee the performance. Fortunately, the input concept (e.g., color) from the user can be leveraged as the reference to constrain the freedom of the proxy word. Therefore, given the target concept word u, we can obtain its embedding as $\mathbf{u} = h(u)$. Then, to learn appropriate representations from the proxy embedding, the original problem can be rewritten with the constraint as

$$\mathcal{L}(\mathbf{w}_i) = -\langle f(x_i), h(t_i^*)\rangle \quad s.t. \quad \|\mathbf{w}_i - \mathbf{u}\|_2^2 \leq \lambda$$

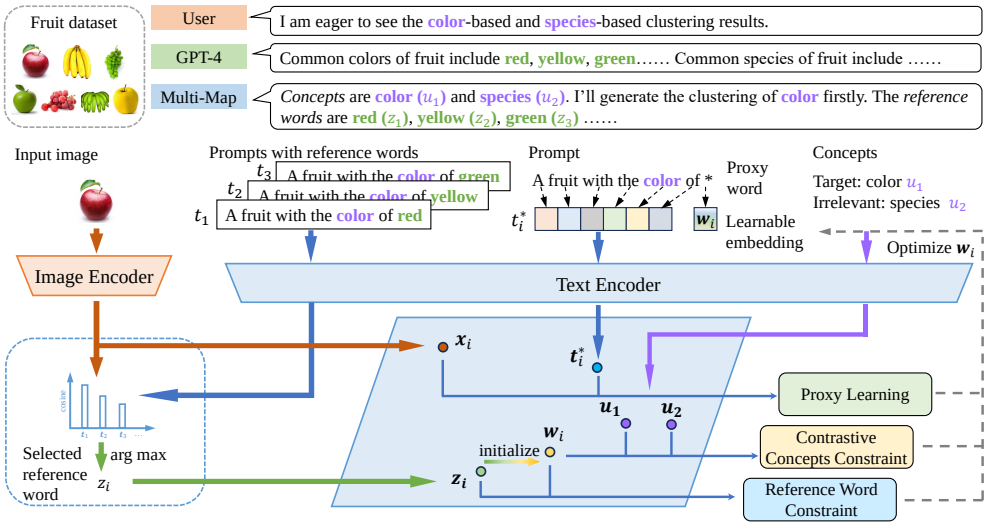


Figure 5.2: Multi-MaP framework. In the training process of Multi-MaP, the vision and text encoders are frozen and the proxy word embeddings $\mathbf{w}_i$ are learnable. Specifically, it first constructs the prompt embeddings based on the reference words provided by GPT-4 using a user's high-level concept, and then selects a reference word z_i for each image according to the similarity between the prompt embeddings $\mathbf{t}_i$ and the image embeddings $\mathbf{x}_i$. Then, it combines the prompt and the reference words to form the new prompt embeddings $\mathbf{t}_i^*$ and maximizes the similarity to the image representation, so the proxy word embeddings $\mathbf{w}_i$ can capture the desired image features.

The constrained problem is equivalent to

$$\mathcal{L}(\mathbf{w}_i) = -\langle f(x_i), h(t_i^*)\rangle + \alpha\|\mathbf{w}_i - \mathbf{u}\|_2^2 \tag{5.2}$$

following [16], which can be optimized effectively by gradient descent.

Constrained Optimization with Reference Word

However, it is well known that the user concept is often with a large scope covering a broad range of words (e.g., color covers all including but not limited to 'red', 'blue', 'green', etc.). As suggested by our above theoretical analysis, it is desired if the reference is as close as possible. In a clustering scenario and given only the user's high-level concept, it is challenging

to find a closer reference word to further constrain the proxy learning. Fortunately, with the development of large language models (LLMs), we can leverage them (e.g., GPT-4) to provide relevant words according to a user's high-level concept as the candidate set and develop a selection strategy to obtain a closer reference word for each image. While the responses gathered from GPT-4 might not always precisely align with the data's ground truth, they indisputably furnish valuable candidate features, enriching the capabilities of Multi-MaP.

To elucidate, considering the task of clustering a fruit dataset based on the concept of color, we construct a natural-language instruction for GPT-4 using the following template: "You are given an image clustering task. Each image contains a `[object]`. I am eager to obtain a `[concept]`-based clustering result for `[object]`. Please list the most common `[concept]`s of `[object]` as a comma-separated list of single English words." For the Fruit dataset with the concept "color", the actual prompt becomes: "You are given an image clustering task. Each image contains a fruit. I am eager to obtain a color-based clustering result for fruit. Please list the most common colors of fruit as a comma-separated list of single English words." GPT-4 then typically returns an answer such as "red, yellow, green, orange, purple, blue", which we treat as reference-word candidates. The same template is reused for all datasets: we only replace `[object]` with the dataset-level noun phrase (e.g., "flower", "playing card", "car", "face") and `[concept]` with the user-specified high-level concept (e.g., "species", "pose", "model", "emotion"). In practice, small changes in wording (for example, replacing "I am eager to obtain" with "I would like to get") do not affect the resulting candidate set, as long as the instruction explicitly asks GPT-4 to enumerate typical `[concept]` values for the given `[object]`. All words returned by GPT-4 are collected into the candidate reference set $\{z_k\}_k$, where, for example, z_1 = "red", z_2 = "yellow", etc. Their text representations are then obtained from prompts t_k such as "fruit with the color of z_k". The shorter input fragment shown in Fig. 5.2 (e.g., "I am eager to get a color-based clustering result for fruit") is a part of this full prompt and is displayed only for illustration.

Given the image x_i, the closest reference can be observed as

$$z_i = \arg\max_k \langle \mathbf{x}_i, \mathbf{t}_k \rangle \tag{5.3}$$

where $\mathbf{t}_k = h(t_k)$. After that, $\mathbf{w}_i$ can be initialized with the embedding of z_i as $\mathbf{z}_i = h(z_i)$. Moreover, we change the regularization using a closer reference word compared to the high-level concept as

$$\mathcal{L}(\mathbf{w}_i) = -\langle f(x_i), h(t_i^*) \rangle + \alpha \|\mathbf{w}_i - \mathbf{z}_i\|_2^2. \tag{5.4}$$

After that, $\mathbf{w}_i$ can be initialized with the embedding of z_i as $\mathbf{z}_i = h(z_i)$. Moreover, we change the regularization using a closer reference word compared to the high-level concept as

$$\mathcal{L}(\mathbf{w}_i) = -\langle f(x_i), h(t_i^*) \rangle + \alpha \|\mathbf{w}_i - \mathbf{z}_i\|_2^2 \tag{5.5}$$

Contrastive Concepts

In some application scenarios, one user may need more than one clustering and provide high-level concepts as $\{u_j\}$, e.g., u_1: "color", u_2: "species", etc. For the concept "color", the irrelevant concept "species" can be leveraged as the negative constraint for the learning of proxy words. Concretely, let u_w denote the target concept word, a contrastive loss can be adopted as regularization

$$R(\mathbf{w}_i) = -\log \frac{\exp(\mathbf{w}_i^\top \mathbf{u}_w)}{\sum_j \exp(\mathbf{w}_i^\top \mathbf{u}_j)}$$

and the final objective becomes

$$\mathcal{L}(\mathbf{w}_i) = -\langle f(x_i), h(t_i^*) \rangle + \alpha \|\mathbf{w}_i - \mathbf{z}_i\|_2^2 - \beta \log \frac{\exp(\mathbf{w}_i^\top \mathbf{u}_w)}{\sum_j \exp(\mathbf{w}_i^\top \mathbf{u}_j)} \tag{5.6}$$

where the first term is to infer the user-specific feature, while the latter two terms constrain the proxy word to the reference words for the appropriate representation extraction from the pre-trained text encoder. The overall framework of Multi-MaP is illustrated in Fig. 5.2.

Table 5.1: Dataset Statistics.

Datasets	# Samples	# Clusters
ALOI	288	2;2
Card	8,029	13;4
CMUface	640	4;20;2;4
Fruit	105	3;3
Fruit360	4,856	4;4
Stanford Cars	1,200	4;3
Flowers	1,600	4;4

5.4 Experiments

To demonstrate our proposed method, we evaluate MultiMap on all publicly available image datasets in multiple clustering, including ALOI [47], Stanford Cars [81], Card [174], CMUface [57], Flowers [110], Fruit [70], and Fruit360 [174] as summarized in Table 5.1.

We compare MultiMap against five state-of-the-art methods: **MSC** [70] is a traditional multiple clustering method that uses hand-crafted features; **MCV** [56] leverages multiple feature extractors to represent different "views" of the same data and employs a multi-input neural network to enhance clustering outcomes; **ENRC** [102] is a deep multiple clustering method that integrates auto-encoder and clustering objective to generate different clusterings; **iMClusts** [124] makes use of the expressive representational power of deep autoencoders and multi-head attention to accomplish multiple clusterings; **AugDMC** [174] leverages augmentations to learn different image representations to achieve multiple clustering.

5.4.1 Experiment Setup

We employ Adam and set momentum as 0.9 to train the model for 1000 epochs. All hyperparameters are searched according to the loss score of MultiMap, where the learning rate is searched in $\{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005\}$, weight decay is in $\{0.0005, 0.0001, 0.00005,$

Table 5.2: Quantitative comparison. The significantly best results with 95% confidence are in bold.

Dataset	Clustering	MSC		MCV		ENRC		iMClusts		AugDMC		MultiMap	
		NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI
ALOI	Color	0.1563	0.3428	0.6982	0.7439	0.9833	0.9892	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Shape	0.2968	0.5199	0.7359	0.8261	0.9732	0.9861	0.9963	0.9989	1.0000	1.0000	1.0000	1.0000
Fruit	Color	0.6886	0.8051	0.6266	0.7685	0.7103	0.8511	0.7351	0.8632	0.8517	0.9108	**0.8619**	**0.9526**
	Species	0.1627	0.6045	0.2733	0.6597	0.3187	0.6536	0.3029	0.6743	0.3546	0.7399	**1.0000**	**1.0000**
Fruit360	Color	0.2544	0.6054	0.3776	0.6791	0.4264	0.6868	0.4097	0.6841	0.4594	0.7392	**0.6239**	**0.8243**
	Species	0.2184	0.5805	0.2985	0.6176	0.4142	0.6984	0.3861	0.6732	0.5139	0.7430	**0.5284**	**0.7582**
Card	Order	0.0807	0.7805	0.0792	0.7128	0.1225	0.7313	0.1144	0.7658	0.1440	0.8267	**0.3653**	**0.8587**
	Suits	0.0497	0.3587	0.0430	0.3638	0.0676	0.3801	0.0716	0.3715	0.0873	0.4228	**0.2734**	**0.7039**
CMUface	Emotion	0.1284	0.6736	0.1433	0.5268	0.1592	0.6630	0.0422	0.5932	0.0161	0.5367	**0.1786**	**0.7105**
	Glass	0.1420	0.5745	0.1201	0.4905	0.1493	0.6209	0.1929	0.5627	0.1039	0.5361	**0.3402**	**0.7068**
	Identity	0.3892	0.7326	0.4637	0.6247	0.5607	0.7635	0.5109	0.8260	0.5875	0.8334	**0.6625**	**0.9496**
	Pose	0.3687	0.6322	0.3254	0.6028	0.2290	0.5029	0.4437	0.6114	0.1320	0.5517	**0.4693**	**0.6624**
Stanford Cars	Color	0.2331	0.6158	0.2103	0.5802	0.2465	0.6779	0.2336	0.6552	0.2736	0.7525	**0.7360**	**0.9193**
	Type	0.1325	0.5336	0.1650	0.5634	0.2063	0.6217	0.1963	0.5643	0.2364	0.7356	**0.6355**	**0.8399**
Flowers	Color	0.2561	0.5965	0.2938	0.5860	0.3329	0.6214	0.3169	0.6127	0.3556	0.6931	**0.6426**	**0.7984**
	Species	0.1326	0.5273	0.1561	0.6065	0.1894	0.6195	0.1887	0.6077	0.1996	0.6227	**0.6013**	**0.8103**

$0.00001, 0\}$, α, β are in $\{0.0, 0.1, 0.2, \ldots, 1.0\}$, and λ is fixed as 1 for all the experiments. For the non pre-trained methods, we perform k-means [97] 10 times due to its randomness and evaluate the average clustering performance using two quantitative metrics, that is, Normalized Mutual Information (NMI) [156] and Rand index (RI) [122]. These measures range in $[0, 1]$, and higher scores imply more accurate clustering results. The experiments are conducted with GPU NVIDIA GeForce RTX 2080 Ti.

It should also be noted that some data are difficult to obtain corresponding candidate labels from GPT-4 or the labels do not provide semantic features, such as names. For example, for the identity clustering for the CMUface dataset [57], different identities represent different people and the semantic meaning of names should not affect the clustering results. In this case, we randomly extract 10 words from WordNet [40] as reference words, in order to make the candidate labels more distinctive. For instance, we randomly choose "abstain,

candid, function, haphazard, knot, luxury, nonchalance, pension, resilience, taciturn" for the above scenario. Furthermore, all publicly available multiple clustering datasets provide each ground-truth clustering a high-level concept, e.g., 'shape', 'pose', etc. Therefore, in the experiment, we directly use them as users' preferences for our evaluation purposes.

5.4.2 Performance Comparison

In our experiments, after we obtain the proxy word embedding of each image for a desired concept, we feed them to k-means to obtain the corresponding clustering. Since k-means is random, we repeat ten times and the average performance is reported in Table 5.2. The best results are marked by bold numbers.

We can observe that MultiMap outperforms the baselines in all the cases, indicating the superiority of the proposed method. This also shows a strong generalization ability of the pre-trained model by CLIP, which can capture the features of data in different aspects.

Since our method uses the CLIP encoder and GPT-4 to obtain clustering results, a natural question arises how the performance would be if we directly use them in a zero-short manner. Therefore, we provide two zero-shot variants of CLIP, that is, CLIP_{GPT} that uses GPT-4 to obtain the candidate labels and predicts labels through zero-shot classification with all candidate labels as class names, and $\text{CLIP}_{\text{label}}$ that performs zero-shot classification with all ground truth labels. It should be noted that $\text{CLIP}_{\text{label}}$ uses an unfair setting with a ground-truth label set known in advance, which is expected to provide the best performance using CLIP in a zero-short manner. The results are shown in Table 5.3.

As expected, $\text{CLIP}_{\text{label}}$ achieves better performance than CLIP_{GPT} in almost all cases, since $\text{CLIP}_{\text{label}}$ uses a fixed ground truth label set, while CLIP_{GPT} uses candidate labels that may not match the ground truth exactly, introducing noise. Note that CLIP_{GPT} and $\text{CLIP}_{\text{label}}$ achieve the same results on Cards, because the candidate labels provided by GPT-4 are exactly the same as the true labels.

Besides, MultiMap performs better than CLIP_{GPT} in almost all cases, indicating that the proposed method can learn more effective features through its training process. Moreover,

Table 5.3: Variants of CLIP. The significantly best results with 95% confidence are in bold.

Dataset	Clustering	CLIP_{GPT}		$\text{CLIP}_{\text{label}}$		MultiMap	
		NMI	RI	NMI	RI	NMI	RI
ALOI	Color	0.8581	0.9407	1.0000	1.0000	1.0000	1.0000
	Shape	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Fruit	Color	0.7912	0.9075	**0.8629**	**0.9780**	0.8619	0.9526
	Species	0.9793	0.9919	1.0000	1.0000	1.0000	1.0000
Fruit360	Color	0.5613	0.7305	0.5746	0.7673	**0.6239**	**0.8243**
	Species	0.4370	0.7552	**0.5364**	**0.7631**	0.5284	0.7582
Card	Order	0.3518	0.8458	0.3518	0.8458	**0.3653**	**0.8587**
	Suits	0.2711	0.6123	0.2711	0.6123	**0.2734**	**0.7039**
CMUface	Emotion	0.1576	0.6532	0.1590	0.6619	**0.1786**	**0.7105**
	Glass	0.2905	0.6869	**0.4686**	**0.7505**	0.3402	0.7068
	Identity	0.1998	0.6388	0.2677	0.7545	**0.6625**	**0.9496**
	Pose	0.4088	0.6473	0.4691	0.6409	**0.4693**	**0.6624**
Stanford Cars	Color	0.6539	0.8237	0.6830	0.8642	**0.7360**	**0.9193**
	Type	0.6207	0.7931	**0.6429**	**0.8456**	0.6355	0.8399
Flowers	Color	0.5653	0.7629	0.5828	0.7836	**0.6426**	**0.7984**
	Species	0.5620	0.7553	**0.6019**	0.7996	0.6013	**0.8103**

although $\text{CLIP}_{\text{label}}$ uses the ground truth, which is expected to be the best, MultiMap still outperforms $\text{CLIP}_{\text{label}}$ in some cases, such as the clustering of color for Fruit360 dataset. This is because CLIP is more inclined to capture the features of one aspect of the data, while MultiMap learns better embedding of different aspects by training with the supervision of users' interests. Furthermore, MultiMap also achieves very competitive results as $\text{CLIP}_{\text{label}}$ in the remaining cases, which further demonstrates the effectiveness of the proposed method.

5.4.3 Ablation Study

To validate the effectiveness of the proposed method, we show the gain from four components in MultiMap in Table 5.4. Let "MultiMap_p" denote the proxy learning without concept-level constraint, "MultiMap_c" denote the variant optimized by solely applying concept word

Table 5.4: Components ablation. All of our components boost performance consistently in all benchmark multi-clustering vision tasks.

		MultiMap$_p$		MultiMap$_c$		MultiMap$_r$		MultiMap$_{cr}$		MultiMap	
Modules	Proxy Learning	✓		✓		✓		✓		✓	
	Concept Word	×		✓		×		✓		✓	
	Reference Word	×		×		✓		✓		✓	
	Contrastive Concepts	×		×		×		×		✓	
		NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑
ALOI [47]	Color	0.9619	0.9826	1.0000	1.0000	0.9795	0.9869	1.0000	1.0000	1.0000	1.0000
	Shape	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Fruit [70]	Color	0.7642	0.8439	0.8215	0.9283	0.8136	0.9073	0.8484	0.9308	**0.8619**	**0.9526**
	Species	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Fruit360 [174]	Color	0.5643	0.7665	0.6217	0.7836	0.5910	0.7746	0.6089	0.7965	**0.6239**	**0.8243**
	Species	0.5077	0.7368	0.5137	0.7436	0.5094	0.7425	0.5199	0.7428	**0.5284**	**0.7582**
Card [174]	Order	0.1932	0.8152	0.3568	0.8472	0.3113	0.8229	0.3616	0.8094	**0.3653**	**0.8587**
	Suits	0.2375	0.6282	0.2696	0.6641	0.2498	0.6365	0.2562	0.6599	**0.2734**	**0.7039**
CMUface [57]	Emotion	0.1690	0.6170	0.1714	0.6229	0.1697	0.6360	0.1713	0.6843	**0.1786**	**0.7105**
	Glass	0.3112	0.6911	0.3269	0.7136	0.3162	0.6917	0.3370	0.7108	**0.3402**	**0.7068**
	Identity	0.5617	0.8234	0.6243	0.8359	0.5839	0.8263	0.6391	0.8946	**0.6625**	**0.9496**
	Pose	0.4361	0.6386	0.4550	0.6499	0.4381	0.6429	0.4387	0.6489	**0.4693**	**0.6624**
Stanford cars [81]	Color	0.5939	0.7835	0.6836	0.8659	0.6729	0.8638	0.7112	0.9117	**0.7360**	**0.9193**
	Type	0.5569	0.7996	0.6383	0.8271	0.6091	0.8046	0.6289	0.8181	**0.6355**	**0.8399**
Flowers [110]	Color	0.5783	0.7723	0.5830	0.7833	0.5987	0.7849	0.6216	0.7941	**0.6426**	**0.7984**
	Species	0.5704	0.7608	0.5744	0.7842	0.5723	0.7811	0.5846	0.7892	**0.6013**	**0.8103**

to constrain the freedom of the proxy word, "MultiMap$_r$" denote the variant optimized by reference word provided by GPT-4 to find a closer reference word to further constrain the proxy learning, and "MultiMap$_{cr}$" denote the variant leveraged both concept word and reference word.

We can observe that the proxy learning only with concept word or reference word, i.e., MultiMap$_c$ and MultiMap$_r$, performs better than MultiMap$_p$. This shows that the proposed concept-level constraint and constrained optimization with reference words play an

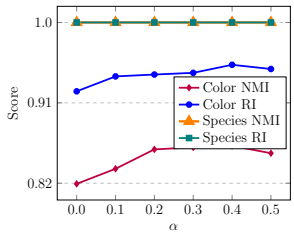


Figure 5.3: Parameter sensitivity of α

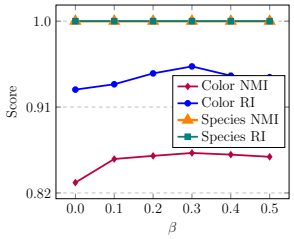


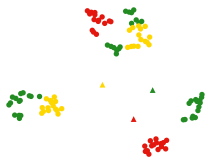
Figure 5.4: Parameter sensitivity of β

Figure 5.5: Parameter analysis of α and β on Fruit [70].

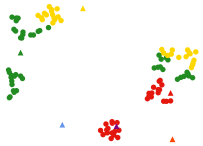
important role in the model as demonstrated by our theoretical analysis. Moreover, the model with combined components, i.e, MultiMap$_{cr}$, achieves better results than MultiMap$_r$ and MultiMap$_c$. This indicates the effectiveness of the combination of reference words and reference concepts. Finally, our proposal using all including the contrastive concepts can further improve the performance, and thus provide the best results on all cases. This further demonstrates our proposal.

5.4.4 Parameter Analysis

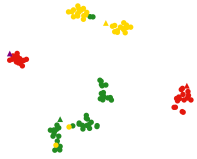
We further investigate the effect of the reference word constraint weight α and concept-level constraint weight β varying from 0.0 to 0.5. The results of MultiMap on Fruit datasets are shown in Fig. 5.5 (a) and Fig. 5.5 (b), respectively. As α and β change, the proposed method keeps a species score of 1, since the image encoder can capture very effective species features. The above results also show that the proposed method can effectively capture useful information from images, reference words, and target concepts. As α increases, the proposed method first increases and then decreases, and reaches the maximum values at α=0.4. Similar results can be observed for β that the performance of the proposed method first increases and then decreases as β increases, and reaches the maximum values at β=0.3. Indicating a suitable value of α or β is helpful for MultiMap to obtain more effective embeddings for multiple clustering tasks. More studies such as efficiency analysis can be found in the



(a) Color using ground truth

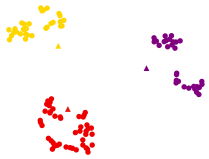


(b) Color using reference words

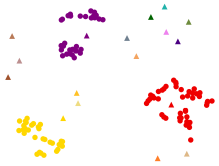


(c) Color of MultiMap

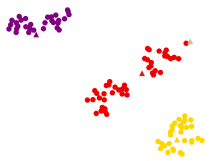
(d) Legend of color



(e) Species using ground truth



(f) Species using reference words



(g) Species of MultiMap

(h) Legend of species

Figure 5.6: Visualization of feature embeddings and related labels. The points represent the image or pseudo-word embeddings, and the triangles represent the prompt or label embeddings. Different colors represent different labels, which are indicated by the text next to the triangles.

supplementary.

5.4.5 Visualization

To further demonstrate the effectiveness of the proposed method, we visualize the representations obtained in $\text{CLIP}_{\text{label}}$, CLIP_{GPT}, and MultiMap. Specifically, for $\text{CLIP}_{\text{label}}$ and CLIP_{GPT}, we visualize the image representations as well as prompts generated with real labels and candidate labels, respectively. For the MultiMap, we visualize the word embedding $\boldsymbol{w}_*$ and the candidate labels selected for initialization. The results are shown in Fig. 5.6. For species clustering, we can see that image embeddings show very clear boundaries and

correspond well to the prompt for $\text{CLIP}_{\text{label}}$, which indicates that CLIP can effectively capture the features of the species in the data. CLIP_{GPT} uses the candidate labels to generate prompts, which introduces more noise, but benefits from the CLIP text encoder, the image embedding can keep a relatively far distance from most of the irrelevant prompts. However, since there are still a few images that are labeled as peaches (i.e., a noisy label), it performs slightly worse than $\text{CLIP}_{\text{label}}$, as shown in Fig. 5.6(e). Besides, MultiMap can capture the image and users' interests in the training process, therefore it compensates for the shortcomings of CLIP_{GPT} and achieves better results. On the other hand, for color clustering, the prompts are farther away from $\text{CLIP}_{\text{label}}$ and CLIP_{GPT}, that indicates the image embeddings mainly capture the features of species, which have no direct connection with color. CLIP_{GPT} generates the prompt from the candidate label, which has more noise than the ground truth label, resulting in worse performance than $\text{CLIP}_{\text{label}}$. The proposed method can distinguish different colors more clearly, because it can learn from the user's interest and capture the color-related features. However, some red color embeddings are closer to purple, because some images in the datasets are actually purple, but labeled as red. To sum up, the proposed method can learn more effective embeddings based on the users' interests for multiple clustering tasks.

5.5 Summary

In the chapter, we investigates the significant challenges that current advanced deep learning techniques face in multiple clustering. A key issue is that users often do not need every clustering result produced by an algorithm, and selecting the most relevant one requires an in-depth understanding of each outcome. To overcome this challenge, the proposed method introduces a novel multi-modal proxy learning process, which effectively aligns a user's brief keyword describing the interest with the corresponding vision components. By integrating a multi-modal model and GPT-4 to precisely capture a user's interest using a keyword, the proposed approach uses both reference word constraint and concept-level constraint to discover personalized clustering result(s), which can also lead to enhanced performance.

Experiments on diverse datasets demonstrate the superiority of the proposed method in multiple clustering tasks with a precise capture of a user's interest. The proposed method is limited by data with semantic meaningful labels, although we can use WordNet to help, whose comprehensive study will be our future work.

Chapter 6

CUSTOMIZED MULTIPLE CLUSTERING VIA MULTI-MODALSUBSPACE PROXY LEARNING

6.1 Background and Overview

Although methods like Multi-MaP leverage models such as CLIP to pull textual and visual embeddings toward a user's high-level concept, they face two real-world limitations: (i) they often require users to provide a contrastive concept distinct from the desired one, which is neither feasible nor user-guided in many settings; and (ii) their clustering quality is capped by a decoupled pipeline in which representation learning and clustering are performed separately (e.g., proxy learning followed by k-means). This separation introduces inefficiencies, prevents mutual refinement between representation and partition, and reduces the system's ability to adapt dynamically to user-specific requirements—ultimately yielding a less intuitive, less user-guided clustering experience.

This chapter builds on the previous chapter's Multi-MaP framework, we propose a multi-modal subspace proxy learning method that turns a user's high-level concept into a low-dimensional text-conditioned subspace. Using LLMs (e.g., GPT-4) to surface reference words as subspace bases, the method learns per-image proxy representations as combinations within this subspace and jointly optimizes them with a clustering objective. By performing clustering directly in the learned feature space, the approach removes the need for explicit contrastive concepts, improves efficiency, and enhances clustering quality while better honoring user intent.

Concretely, Multi-Sub assumes that image and text embeddings relevant to a user's concept lie in (or are well-approximated by) a concept-conditioned subspace spanned by LLM-proposed reference words. Each image learns a soft proxy in this subspace that is aligned

with the visual encoder and regularized by the textual bases; in parallel, a clustering-aware loss optimizes both representation and partition. This end-to-end scheme couples *what to represent* (concept-aligned features) with *how to partition* (cluster assignments), thereby eliminating the sub-optimality of two-stage pipelines.

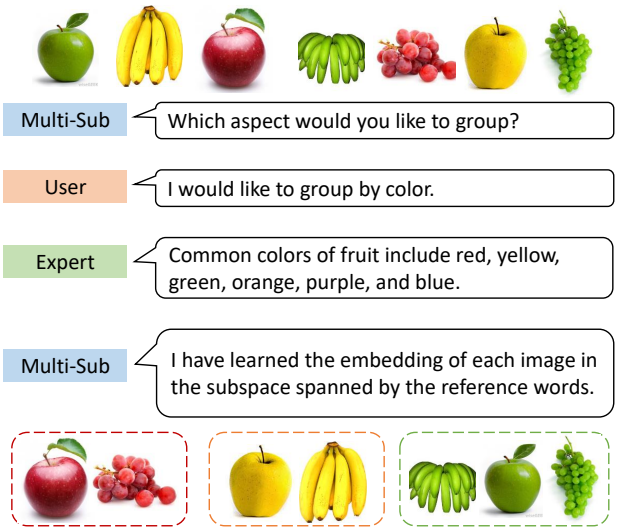


Figure 6.1: The workflow of Multi-Sub, which derives a desired clustering by learning a concept-conditioned subspace spanned by reference words (from GPT-4) and jointly optimizing representations and cluster assignments.

The contributions of this work can be summarized as

- We propose Multi-Sub, a concept-conditioned multi-modal subspace proxy framework that turns a user's high-level textual preference into a low-dimensional text-conditioned subspace spanned by LLM-derived reference words, and learns per-image proxies within this subspace without requiring user-specified contrastive concepts.

- We design an end-to-end optimization scheme that couples proxy learning with clustering: a partially trainable CLIP image head aligns visual features with subspace proxies, while a clustering-aware objective (with intra- and inter-cluster terms) refines

the projection layer, thereby eliminating the sub-optimality of decoupled "proxy-then-k-means" pipelines.

- We conduct extensive experiments on all publicly available visual multiple-clustering benchmarks (and an additional CIFAR-10 variant), together with ablations on subspace construction and text encoders, demonstrating that Multi-Sub precisely captures user interests and achieves state-of-the-art or highly competitive performance across diverse settings.

6.2 Preliminaries

In this chapter we focus on user–conditioned multiple clustering of images under high–level textual concepts. This section introduces the basic setting, notation, and multi–modal components that later sections build upon.

6.2.1 User-Conditioned Multiple Clustering

Let $\mathcal{X} = \{x_i\}_{i=1}^{N}$ denote a collection of images. Classical multiple clustering aims to construct several distinct partitions of $\mathcal{X}$, each reflecting a different latent aspect of the data (e.g., shape, texture, pose) without direct user guidance. In contrast, we assume that a user expresses interests through a small set of high–level textual concepts

$$\mathcal{U} = \{u^{(1)}, u^{(2)}, \ldots, u^{(K)}\}, \tag{6.1}$$

where each $u^{(k)}$ is a short phrase such as "color", "species", "rank", or "emotion". For each concept $u^{(k)}$, our goal is to obtain a corresponding clustering

$$\mathcal{C}^{(k)} = \{C_1^{(k)}, \ldots, C_{M_k}^{(k)}\}, \tag{6.2}$$

where $\{C_m^{(k)}\}_{m=1}^{M_k}$ forms a partition of $\mathcal{X}$ and groups images that are similar under concept $u^{(k)}$. Crucially, we do not assume that the user provides explicit class names, the number of clusters, or contrastive concepts; the only supervision is the concept itself.

From a representation–learning perspective, this setting can be viewed as follows: for each concept $u^{(k)}$, we wish to learn a concept–aligned representation

$$\mathbf{v}_i^{(k)} \in \mathbb{R}^d \tag{6.3}$$

for every image x_i, such that clustering in $\{\mathbf{v}_i^{(k)}\}_{i=1}^N$ yields $\mathcal{C}^{(k)}$ that faithfully reflects the intended aspect. Different concepts should induce different representations and, consequently, diverse clusterings on the same underlying dataset.

6.2.2 Vision–Language Encoders

To connect user–provided textual concepts with image data, we build on large–scale vision–language models such as CLIP, ALIGN, or BLIP [121, 10, 5]. These models consist of an image encoder $f(\cdot)$ and a text encoder $h(\cdot)$ that map images and texts into a shared embedding space:

$$\mathbf{x}_i = f(x_i) \in \mathbb{R}^{d_v}, \qquad \mathbf{t} = h(t) \in \mathbb{R}^{d_t}, \tag{6.4}$$

where cosine similarity between $\mathbf{x}_i$ and $\mathbf{t}$ reflects cross–modal semantic affinity. The encoders are typically pre-trained with a contrastive objective on large collections of image–text pairs, enabling zero-shot transfer to downstream tasks using only textual descriptions.

In our setting, short text fragments play multiple roles: they describe the dataset–level object category (e.g., “fruit”, “flower”, “playing card”, “car”), express the user’s high–level concept (e.g., “color”, “species”), and serve as anchors for more fine–grained reference words. We denote by $\phi(\cdot)$ the token–embedding function associated with the text encoder, so that $\phi(w)$ is the embedding of a single word token w. The combination of $f(\cdot)$, $h(\cdot)$, and $\phi(\cdot)$ gives us a unified space in which both images and concept–related words can be compared and combined.

6.3 The Proposed Method

Given a dataset of images $\{x_i\}_{i=1}^n$ and user-defined preferences for data grouping (such as color and species), our goal is to generate clustering results that are specifically tailored to

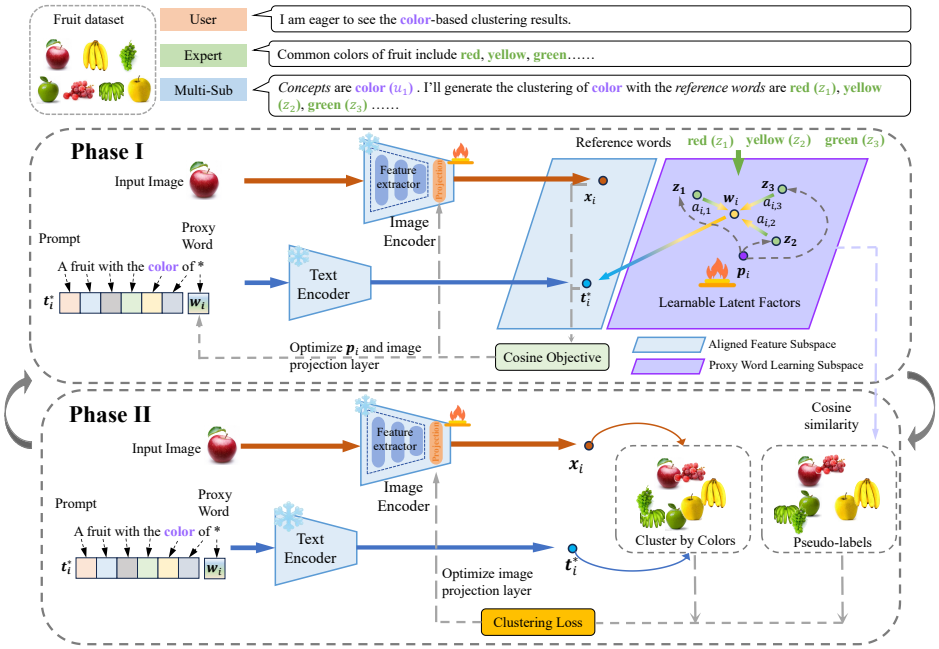


Figure 6.2: In Multi-Sub framework, Phase I (Proxy Learning and Alignment) processes each image x_i with user-defined textual prompts through a partially learnable image encoder (with a learnable projection layer) and a frozen text encoder. The latent factor $\mathbf{p}_i$ calculates weights $\{a_{i,k}\}_{k=1}^{K}$ based on the similarity to reference word embeddings $\{\mathbf{z}_i\}_{k=1}^{K}$, which are then aggregated to form the proxy word embedding $\mathbf{w}_i$. This proxy word embedding, combined with the image representation $\mathbf{x}_i$, establishes the Aligned Feature Subspace for better alignment between the text and image under the user's interest. In Phase II (Clustering), given the learned proxy word embeddings $\{\mathbf{w}_i\}$ from Phase I to form pseudo-labels, the projection layer of the image encoder is further refined using the clustering loss.

each preference. Thereafter, end users can directly use them for different application purposes without additional manual selection efforts. This process poses significant challenges, as it requires accurately aligning the complex, multi-dimensional data of images with the subjective and varied textual preferences of users. Traditional clustering methods often fail to capture these nuances, leading to a generic and less informative categorization for specific user applications.

Recently, the CLIP model [121] facilitated a more natural alignment between textual in-

terests and visual representations. Our method, Multi-Sub, extends this alignment through a novel multi-modal subspace proxy learning approach. Fig. 6.2 outlines the overall framework of Multi-Sub, which is tailored to capture and respond to the diverse interests of users in clustering tasks. Multi-Sub employs a two-phase iterative approach to align and cluster images based on user-defined preferences such as color and species as described below.

6.3.1 Background: Multi-Modal Pre-Training in CLIP

Let $\{x_i, t_i\}_{i=1}^n$ be a set of image-text pairs, where x_i denotes an image and t_i denotes its corresponding text description. We can obtain the vision and text representations of each pair by applying two encoders, $f(\cdot)$ and $h(\cdot)$, as $\mathbf{x}_i = f(x_i)$ and $\mathbf{t}_i = h(t_i)$. Both $f(\cdot)$ and $h(\cdot)$ are encoders that optimize the vision and text representations, respectively, such that $\mathbf{x}_i$ and $\mathbf{t}_i$ are unit vectors. The primary goal during this pre-training phase is to minimize the contrastive loss, formulated as

$$\min_{f,h} \sum_i -\log \frac{\exp(\mathbf{x}_i^\top \mathbf{t}_i/\tau)}{\sum_j \exp(\mathbf{x}_i^\top \mathbf{t}_j/\tau)} - \log \frac{\exp(\mathbf{t}_i^\top \mathbf{x}_i/\tau)}{\sum_j \exp(\mathbf{t}_i^\top \mathbf{x}_j/\tau)} \tag{6.5}$$

where τ is a temperature parameter. The contrastive loss encourages the alignment of the image and its description while penalizing the similarity of the image with irrelevant texts [118]. The efficacy of this contrastive approach is vital for the subsequent phases of proxy word learning and fine-grained clustering, as it ensures that the foundational embeddings accurately reflect the inherent content and context of each modality.

6.3.2 Subspace Proxy Word Representation

We build upon the pre-trained image and text encoders from CLIP and investigate whether we can leverage the image-text alignment to extract user-specific information. Specifically, given a fruit image [70] as illustrated in Fig. 6.2, different users may have different interests of its attributes, such as color, species, etc. However, the pre-trained image encoder in CLIP can only produce a single image embedding, which may not capture a user's interest exactly, not

mentioning capturing different aspects. Furthermore, unlike classification tasks, clustering tasks do not come with concrete cluster names or numbers. Therefore, we cannot directly use the pre-trained text encoder of CLIP to generate the corresponding text embedding.

To address these challenges, we propose a subspace proxy word learning method to learn new embedding under the preferred aspect provided by the user. Thereafter, the main challenge is, given only a high-level concept like 'color' as in Fig. 6.2, how to effectively represent its subspace. Since the high-level concept itself cannot reflect different details under this concept in different images, it is difficult to do effective alignment between the high-level concept and images to figure out the corresponding vision subspace. Therefore, we propose to figure out the text subspace at first. Concretely, given pre-trained large language models like GPT-4 as low-cost experts, we can quickly gather common categories under a high-level concept using only one query like 'what are the common fruit colors' in Fig. 6.2. However, we cannot directly use the returned categories to do grouping, since they may not cover all existing categories in the data. Instead, we consider that most categories in the data under this concept are residing in the same subspace as the returned ones. Therefore, we can apply suggested categories as basis or reference words in the subspace. Then, each image's category under the desired concept can be represented by a linear combination of these reference words.

Assuming GPT-4 provides K reference words as $\{z_k\}_{k=1}^{K}$, the proxy word of image x_i can be calculated as

$$\mathbf{w}_i = \sum_{k=1}^{K} a_{i,k}\phi(z_k) \tag{6.6}$$

where $\phi(z_k)$ is the token embedding of reference word z_k and $\{a_{i,k}\}_{k=1}^{K}$ are weights corresponding to each reference word as a basis. A higher weight $a_{i,k}$ indicates that the image x_i's category is closer to the reference word z_k. Here, we introduce trainable latent factor $\mathbf{p}_i$ to learn the weight $a_{i,k}$, and it can be calculated as

$$a_{i,k} = \frac{\exp\left(\mathbf{p}_i \mathbf{z}_k\right)}{\sum_j \exp\left(\mathbf{p}_i \mathbf{z}_j\right)} \tag{6.7}$$

where $\mathbf{z}_k = \phi(z_k)$. Thereafter, $\mathbf{w}_i$ is representing the token embedding of image x_i's proxy word under the preferred user concept. Once $\mathbf{p}_i$ is well obtained, the image's proxy word representation under the preferred user concept is also obtained. Next, we discuss how to learn $\mathbf{p}_i$ using CLIP.

6.3.3 Multi-Modal Subspace Proxy Learning

As mentioned above, CLIP's text and image encoders were learned by aligning the text prompt with its corresponding image. The standard text prompt of CLIP is designed as "a photo of a fruit" for an image containing "fruit". Now, given a user's preference (e.g., color), we can rewrite the prompt as "a fruit with the color of *" denoted by t_i^* for image x_i, where "*" is the placeholder for the unknown proxy word of image x_i under concept 'color' and its token embedding $\mathbf{w}_i$ can be formulated as the linear superposition of reference words' token embeddings as discussed above.

Thereafter, the prompt text embedding after the text encoder can be formulated as

$$\mathbf{t}_i^* = h(\phi(t_i^*) \| \phi(w_i)) \tag{6.8}$$

To effectively learn $\mathbf{p}_i$, the trainable latent factors, we utilize the alignment capabilities of CLIP by adjusting these factors so that the weighted sum of reference word embeddings closely aligns with the visual representation of the image. This process involves iteratively adjusting $\mathbf{p}_i$ to maximize the cosine similarity between the image's representation $\mathbf{x}_i$ and its corresponding proxy word embedding $\mathbf{w}_i$. The optimization is conducted with the following loss function:

$$\mathcal{L}(\mathbf{w}_i) = -\langle f(x_i), h(\phi(t_i^*) \| \phi(w_i)) \rangle \tag{6.9}$$

It should be noted that this optimization procedure can be conducted with both the text encoder and image encoder frozen, which is very efficient. However, the image embedding extracted directly from the pre-trained image encoder may not reflect its representation under the desired user interest. Therefore, during the optimization procedure, we do freeze the text encoder but open the image encoder. Nevertheless, to preserve the strong capacity of the pre-trained image encoder in CLIP, we open only the projection layer of the image encoder, while its remaining parameters are frozen as shown in the 'Phase I' of Fig. 6.2.

6.3.4 Clustering Loss

To enhance the clustering performance of Multi-Sub, in 'Phase II', we leverage pseudo-labels assigned using the currently learned proxy word embeddings $\{\mathbf{w}_i\}$ and image embeddings $\{\mathbf{x}_i\}$ from 'Phase I'. Concretely, each image x_i can be represented by the concatenation of its currently learned proxy word embedding $\mathbf{w}_i$ and image embedding $\mathbf{x}_i$, denoted as $\mathbf{v}_i = [\mathbf{w}_i, \mathbf{x}_i]$. The pseudo-labels can be obtained by an offline k-means on $\{\mathbf{v}_i\}$, which is however not efficient. Considering that proxy words for data points within the same cluster should show similar relationships to reference words, we obtain the pseudo-labels using the highest cosine similarity between the currently learned proxy word embeddings $\{\mathbf{w}_i\}$ and the reference word embeddings $\{\mathbf{z}_k\}$.

Given the pseudo-labels, the image embeddings can be further optimized by opening only the projection layer of the image encoder for improved compactness and separability in clusters. This loss consists of two primary components: intra-cluster loss and inter-cluster loss, aimed at refining cluster cohesion and separation, respectively. It should be noted that to better represent each image under the desired user concept, we define the clustering loss over $\mathbf{v}_i$ containing both textual and visual information.

Intra-cluster Loss: The intra-cluster loss is designed to minimize the distances between embeddings within the same cluster, encouraging cluster compactness. It is calculated using

the following formula:

$$\mathcal{L}_{\text{intra}} = \frac{1}{N_{\text{intra}}} \sum_{i,j \in \text{intra}} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \tag{6.10}$$

Here, $\|\mathbf{v}_i - \mathbf{v}_j\|^2$ is the squared Euclidean distance between embeddings $\mathbf{x}_i$ and $\mathbf{x}_j$ of data points i and j within the same cluster, and N_{intra} denotes the number of intra-cluster pairs.

Inter-cluster Loss: This component aims to maximize the distances between embeddings from different clusters, thus enhancing separability. The inter-cluster loss is defined by a margin-based hinge loss as follows:

$$\mathcal{L}_{\text{inter}} = \frac{1}{N_{\text{inter}}} \sum_{i,j \in \text{inter}} \max(0, m - \|\mathbf{v}_i - \mathbf{v}_j\|) \tag{6.11}$$

where $\max(0, m - \|\mathbf{v}_i - \mathbf{v}_j\|)$ computes the hinge loss for each pair of embeddings from different clusters, ensuring a minimum margin m between them. N_{inter} is the count of inter-cluster pairs.

Total Loss: The overall clustering loss combines the intra- and inter-cluster losses, moderated by a balancing factor λ:

$$\mathcal{L}_{\text{total}} = \lambda \cdot \mathcal{L}_{\text{intra}} + (1 - \lambda) \cdot \mathcal{L}_{\text{inter}} \tag{6.12}$$

Optimizing this loss function in 'Phase II' helps regularize the embedding space where clusters are both internally dense and well-separated from each other. It should be noted that in this phase we aim to learn a better projection layer only for the image encoder, while all others are fixed as shown in 'Phase II' of Fig. 6.2.

Previous methods often use a two-stage strategy that separates representation learning and clustering to simplify the optimization process. This separation, however, can lead to sub-optimal clustering results, since the learned representations may not be fully aligned with the clustering objective without refinement. In this work, we obtain both the proxy word and the clustering alternatively and simultaneously. Concretely, we first learn the proxy word in a user-preferred subspace. Then, we fix the proxy word and refine the image encoder further to obtain better image representations using the clustering objective. These

Table 6.1: Dataset Statistics.

Datasets	# Samples	# Hand-crafted features	# Clusters
Standford Cars	1,200	wheelbase length; body shape; color histogram	4;3
Card	8,029	symbol shapes; color distribution	13;4
CMUface	640	HOG; edge maps	4;20;2;4
Fruit	105	shape descriptors; color histogram	3;3
Fruit360	4,856	shape descriptors; color histogram	4;4
Flowers	1,600	petal shape; color histogram	4;4
CIFAR-10	60,000	edge detection; color histograms; shape descriptors	2;3

two phases are repeated alternatively until convergence, where 'Phase I' learns 100 epochs and 'Phase II' learns 10 epochs in each alternating according to the empirical experience as summarized in Fig. 6.2.

6.4 Experiments

Datasets To demonstrate the effectiveness of Multi-Sub, we evaluate the proposed method on almost all publicly available visual datasets commonly used in multiple clustering tasks [182], including Stanford Cars [177], Card [174], CMUface [57], Flowers [177], Fruit [70] and Fruit360 [174]. **Stanford Cars** contains two different clustering types, one for car color (e.g., red, blue, black) and one for car type (e.g., sedan, SUV, convertible), comprising 1,200 annotated car images. **Card** includes 8,029 images of playing cards, with two clustering types: one based on rank (e.g., Ace, King, Queen) and another on suit (e.g., clubs, diamonds, hearts, spades). **CMUface** provides 640 facial images with clustering options for pose (e.g., front-facing, side-facing), identity, glasses (with/without), and emotion (e.g., happy, neutral, sad). **Flowers** comprises 1,600 flower images with two clustering types: one for color (e.g., red, blue, yellow) and another for species (e.g., iris, aster). **Fruit** includes 105 images of fruits with two clustering criteria: species (e.g., apples, bananas, grapes) and

Table 6.2: Quantitative comparison. The significantly best results with 95% confidence are in bold.

Dataset	Clustering	MSC		MCV		ENRC		iMClusts		AugDMC		DDMC		Multi-MaP		Multi-Sub	
		NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑
Fruit	Color	0.6886	0.8051	0.6266	0.7685	0.7103	0.8511	0.7351	0.8632	0.8517	0.9108	0.8973	0.9383	0.8619	0.9526	**0.9693**	**0.9964**
	Species	0.1627	0.6045	0.2733	0.6597	0.3187	0.6536	0.3029	0.6743	0.3546	0.7399	0.3764	0.7621	1.0000	1.0000	**1.0000**	**1.0000**
Fruit360	Color	0.2544	0.6054	0.3776	0.6791	0.4264	0.6868	0.4097	0.6841	0.4594	0.7392	0.4981	0.7472	0.6239	0.8243	**0.6654**	**0.8821**
	Species	0.2184	0.5805	0.2985	0.6176	0.4142	0.6984	0.3861	0.6732	0.5139	0.7430	0.5292	0.7703	0.5284	0.7582	**0.6123**	**0.8504**
Card	Order	0.0807	0.7805	0.0792	0.7128	0.1225	0.7313	0.1144	0.7658	0.1440	0.8267	0.1563	0.8326	0.3653	0.8587	**0.3921**	**0.8842**
	Suits	0.0497	0.3587	0.0430	0.3638	0.0676	0.3801	0.0716	0.3715	0.0873	0.4228	0.0933	0.6469	0.2734	0.7039	**0.3104**	**0.7941**
CMUface	Emotion	0.1284	0.6736	0.1433	0.5268	0.1592	0.6630	0.0422	0.5932	0.0161	0.5367	0.1726	0.7593	0.1786	0.7105	**0.2053**	**0.8527**
	Glass	0.1420	0.5745	0.1201	0.4905	0.1493	0.6209	0.1929	0.5627	0.1039	0.5361	0.2261	0.7663	0.3402	0.7068	**0.4870**	**0.8324**
	Identity	0.3892	0.7326	0.4637	0.6247	0.5607	0.7635	0.5109	0.8260	0.5875	0.8334	0.6360	0.8907	0.6625	0.9496	**0.7441**	**0.9834**
	Pose	0.3687	0.6322	0.3254	0.6028	0.2290	0.5029	0.4437	0.6114	0.1320	0.5517	0.4526	0.7904	0.4693	0.6624	**0.5923**	**0.8736**
Stanford Cars	Color	0.2331	0.6158	0.2103	0.5802	0.2465	0.6779	0.2336	0.6552	0.2736	0.7525	0.6899	0.8765	0.7360	0.9193	**0.7533**	**0.9387**
	Type	0.1325	0.5336	0.1650	0.5634	0.2063	0.6217	0.1963	0.5643	0.2364	0.7356	0.6045	0.7957	0.6355	0.8399	**0.6616**	**0.8792**
Flowers	Color	0.2561	0.5965	0.2938	0.5860	0.3329	0.6214	0.3169	0.6127	0.3556	0.6931	0.6327	0.7887	0.6426	0.7984	**0.6940**	**0.8843**
	Species	0.1326	0.5273	0.1561	0.6065	0.1894	0.6195	0.1887	0.6077	0.1996	0.6227	0.6148	0.8321	0.6013	0.8103	**0.6724**	**0.8719**
CIFAR-10	Type	0.1547	0.3296	0.1618	0.3305	0.1826	0.3469	0.2040	0.3695	0.2855	0.4516	0.3991	0.5827	0.4969	0.7104	**0.5271**	**0.7394**
	Environment	0.1136	0.3082	0.1379	0.3344	0.1892	0.3599	0.1920	0.3664	0.2927	0.4689	0.3782	0.5547	0.4598	0.6737	**0.4828**	**0.7096**

color (e.g., green, red, yellow). **Fruit360**, similar to the Fruit dataset, contains 4,856 images annotated for species (e.g., apple, banana, cherry) and color.

Additionally, we created a multiple clustering dataset from **CIFAR-10** [82] by organizing the images into clusters based on type and environment. For type, the clusters are transportation and animals. For environment, the clusters are land, air, and water. The dataset characteristics about data size, handcrafted features, and cluster information are also summarized in Table 6.1.

It should be noted that some data may face challenges in extraction of meaningful candidate categories from GPT-4, or their labels lack semantic features. Taking the identity clustering on the CMUface dataset [57] as an example, different identities correspond to different individuals, and the names' semantic meanings should not affect clustering outcomes. In such cases, following the Multi-Map setting [177], we randomly select 10 words from WordNet [40] as reference categories.

Baselines We compare our Multi-Sub with seven state-of-the-art multiple clustering methods. These methods are: **MSC** [70] is a traditional multiple clustering method that uses hand-crafted features to automatically find different feature subspace for different clusterings; **MCV** [56] leverages multiple pre-trained feature extractors as different views of the same data; **ENRC** [102] integrates auto-encoder and clustering objective to generate different clusterings; **iMClusts** [124] is a deep multiple clustering method that leverages the expressive representational power of deep autoencoders and multi-head attention to generate multiple salient embedding matrices and multiple clusterings therein; **AugDMC** [174] leverages data augmentations to automatically extract features related to different aspects of the data using a self-supervised prototype-based representation learning method; **DDMC** [173] combines disentangled representation learning with a variational Expectation-Maximization (EM) framework; **Multi-MaP** [177] relies on a contrastive user-defined concept to learn a proxy better tailored to a user's interest. It is worth noting that, in our experiments, we apply both traditional and deep learning baselines. Traditional methods rely on hand-crafted features, while deep learning methods directly utilize the original images as input.

Hyperparameter For each user's preference, we train the model for 1000 epochs using Adam optimizer with a momentum of 0.9. We tune all the hyper-parameters based on the loss score of Multi-Sub, where the learning rate is selected from {1e-1,5e-2,1e-2,5e-3,1e-3,5e-4}, weight decay is chosen from $\{5\text{e-}4,1\text{e-}4,5\text{e-}5,1\text{e-}5,0\}$ for all the experiments. Most methods obtain each clustering by applying k-means [97] to the newly learned representations, while ours is end-to-end. The experiments are performed on four NVIDIA GeForce RTX 2080 Ti GPUs.

Evaluation metrics Considering the randomness of k-means for those applicable baselines, we run k-means 10 times and report the average clustering performance using two metrics, namely, Normalized Mutual Information (NMI) [156] and Rand index (RI) [122]. These metrics range from 0 to 1 with higher value indicating better performance compared

Table 6.3: Variants of CLIP. The significantly best results with 95% confidence are in bold.

Dataset	Clustering	CLIP_{GPT}		$\text{CLIP}_{\text{label}}$		Multi-Sub	
		NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑
Fruit	Color	0.7912	0.9075	0.8629	0.9780	**0.9693**	**0.9964**
	Species	0.9793	0.9919	1.0000	1.0000	1.0000	1.0000
Fruit360	Color	0.5613	0.7305	0.5746	0.7673	**0.6654**	**0.8821**
	Species	0.4370	0.7552	0.5364	0.7631	**0.6123**	**0.8504**
Card	Order	0.3518	0.8458	0.3518	0.8458	**0.3921**	**0.8842**
	Suits	0.2711	0.6123	0.2711	0.6123	**0.3104**	**0.7941**
CMUface	Emotion	0.1576	0.6532	0.1590	0.6619	**0.2053**	**0.8527**
	Glass	0.2905	0.6869	0.4686	0.7505	**0.4870**	**0.8324**
	Identity	0.1998	0.6388	0.2677	0.7545	**0.7441**	**0.9834**
	Pose	0.4088	0.6473	0.4691	0.6409	**0.5923**	**0.8736**
Stanford Cars	Color	0.6539	0.8237	0.6830	0.8642	**0.7533**	**0.9387**
	Type	0.6207	0.7931	0.6429	0.8456	**0.6616**	**0.8792**
Flowers	Color	0.5653	0.7629	0.5828	0.7836	**0.6940**	**0.8843**
	Species	0.5620	0.7553	0.6019	0.7996	**0.6724**	**0.8719**
CIFAR-10	Type	0.4935	0.6741	0.5087	0.7102	**0.5271**	**0.7394**
	Environment	0.4302	0.6507	0.4643	0.6801	**0.4828**	**0.7096**

to the groundtruth.

6.4.1 Performance Comparison

Table 6.2 reports the clustering results. During the clustering stage, after we obtain the proxy word embedding of each image for a desired concept, we can concatenate the image embedding and the token embedding of proxy word. The results show that Multi-Sub consistently outperforms the baselines, demonstrating the superiority of the proposed method. This also indicates a strong generalization ability of the pre-trained model by CLIP, which can capture the features of data from different perspectives.

Our methodology uses the CLIP encoder and GPT-4 to derive clustering results, prompting an evaluation of their performance in a zero-shot manner. We introduce two zero-shot variants of CLIP: CLIP_{GPT} and $\text{CLIP}_{\text{label}}$. CLIP_{GPT} uses GPT-4 to generate candidate

Table 6.4: Comparison of differenttext encoders. The significantly best results with 95% confidence are in bold.

Dataset	Clustering	CLIP		ALIGN		BLIP	
		NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑
Fruit360	Color	0.6654	0.8821	**0.7031**	**0.8925**	0.6522	0.8814
	Species	0.6123	0.8504	**0.6426**	**0.8565**	0.6254	0.8536
Card	Order	0.3921	0.8842	**0.4316**	**0.9023**	0.3845	0.8359
	Suits	0.3104	0.7941	**0.3226**	**0.8006**	0.3151	0.7956
CMUface	Emotion	0.2053	0.8527	**0.2148**	**0.8553**	0.2081	0.8535
	Glass	0.4870	0.8324	**0.4951**	0.8351	**0.4951**	**0.8353**
	Identity	0.7441	**0.9834**	**0.7514**	0.9828	0.6853	0.8321
	Pose	0.5923	0.8736	**0.6137**	**0.8942**	0.5732	0.8427
Standford Cars	Color	0.7533	**0.9387**	**0.7624**	0.8942	0.5732	0.8427
	Type	0.6616	0.8792	**0.6712**	**0.8865**	0.6581	0.8731
Flowers	Color	**0.694**	**0.8843**	0.6925	0.8812	0.6843	0.8789
	Species	**0.6724**	**0.8719**	0.6693	0.8691	0.6627	0.8654
CIFAR-10	Type	0.5271	0.7394	**0.5342**	**0.7456**	0.5221	0.7381
	Environment	**0.4828**	**0.7096**	0.4793	0.7064	0.4752	0.7038

labels and performs zero-shot classification, while $\text{CLIP}_{\text{label}}$ uses ground truth labels directly, providing an optimal setting. As shown in Table 6.3, $\text{CLIP}_{\text{label}}$ generally outperforms CLIP_{GPT} due to its use of accurate labels, while CLIP_{GPT} introduces noise. Both variants perform equally on the Card dataset as GPT-4's labels match the groundtruth. Multi-Sub surpasses CLIP_{GPT} and even outperforms $\text{CLIP}_{\text{label}}$ in all cases, demonstrating its ability to capture user-interest-based data aspects and confirming its efficacy. This superiority can be attributed to Multi-Sub's proxy word learning mechanism, which automatically adjusts textual embeddings based on user-defined interests, creating more accurate proxy word embeddings. This approach reduces noise compared to CLIP_{GPT}, which suffers from label mismatches. Additionally, Multi-Sub's iterative learning process refines these embeddings, optimizing alignment between text and image representations.

Table 6.5: Ablation study of Multi-Sub. The results that achieved the highest and second highest performance for each clustering are indicated by boldface and underlined numerals, respectively.

Dataset	Clustering	Subspace	clustering with $h(\phi(w_i))$						clustering with $\mathbf{t}_i^*$				clustering with $\phi(w_i)$			
			image		text		concatenate		text		concatenate		text		concatenate	
			NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑	NMI↑	RI↑
CIFAR-10	Type	$h(\phi(w_i))$	0.3649	0.6546	0.4789	0.6607	<u>0.5208</u>	<u>0.7281</u>	0.4586	0.6331	0.4987	0.6933	0.4438	0.6282	0.4996	0.7069
		$\mathbf{t}_i^*$	0.3581	0.6378	0.4634	0.6439	0.5114	0.7189	0.4704	0.6586	0.5136	0.7196	0.4672	0.6524	0.5013	0.7136
		$\phi(w_i)$	0.3715	0.6589	0.4737	0.6563	0.5185	0.7211	0.4601	0.6420	0.5033	0.6989	0.4821	0.6638	**0.5271**	**0.7394**
	Envrionment	$h(\phi(w_i))$	0.4271	0.6764	0.4533	0.6813	<u>0.4737</u>	<u>0.6905</u>	0.4249	0.6537	0.4149	0.6662	0.4336	0.6691	0.4569	0.6836
		$\mathbf{t}_i^*$	0.4216	0.6677	0.4229	0.6533	0.4496	0.6630	0.4336	0.6689	0.4563	0.6781	0.4264	0.6596	0.4514	0.6695
		$\phi(w_i)$	0.4320	0.6837	0.4507	0.6762	0.4686	0.6834	0.4218	0.6541	0.4432	0.6631	0.4586	0.6876	**0.4828**	**0.7096**

6.4.2 Ablation study

Different ways of constructing subspace The subspace of the proposed method can be expanded by different embeddings, i.e., the token embedding of the proxy word $\phi(w_i)$, the text embedding of the proxy word $h(\phi(w_i))$, and the text embedding of the prompt $\mathbf{t}_i^* = h(\phi(t_i^*)\|\phi(w_i))$. These three kinds of embeddings can also be used to evaluate the clustering results in each case. In addition, we can use different combinations of learned embeddings (e.g., different concatenations of text and image embedding) as the final embedding for clustering. The results are shown in Table 6.5. It can be seen that using word token embedding usually achieves better results. This is expected since the word proxy directly reflects the image's category under the desired concept. The token word embedding subspace is also aligning well with CLIP's training method. In contrast, prompt embedding performs the worst as it introduces noise from user interest, dataset, and reference words, which are unnecessary for clustering. Additionally, most methods perform better when the same approach is used for constructing subspace and evaluating clustering results. Combining text and image embeddings generally enhances performance, capturing user interests from both aspects effectively.

Effect of text encoder Table 6.4 compares the performance of three text encoders—CLIP, ALIGN, and BLIP—across various datasets. The results indicate that ALIGN generally outperforms CLIP and BLIP in most tasks. This suggests that ALIGN's text encoder effectively captures and aligns textual and visual representations, enhancing clustering performance. ALIGN tends to excel in tasks that require distinguishing subtle visual differences influenced by textual descriptions, such as emotions and accessories in the CMUface dataset, and colors in the Fruit360 dataset. CLIP shows a strong tendency in identity-related tasks and complex object categorization, as evidenced by its performance in the CMUface identity task and Standford Cars type clustering. BLIP, while competitive, seems to perform better in categorical distinctions rather than abstract attributes, performing relatively well in species-related tasks across various datasets. These findings underscore the importance of effective text embeddings in multi-modal clustering frameworks.

We conducted an additional analysis using the Maximum Mean Discrepancy (MMD) metric to quantify the differences in the feature spaces generated by different text encoders (i.e., CLIP, ALIGN, and BLIP) in Table 6.6. The MMD results indicate that although our text prompts are simple, the feature spaces generated by different text encoders exhibit significant distributional differences. The effectiveness of a text encoder can vary depending on the specific clustering task. For example, ALIGN tends to excel in tasks with more abstract attributes, such as colors and emotions, while CLIP shows strong performance in identity-related tasks. This variability underscores the importance of selecting an appropriate text encoder based on the specific application requirements. The difference between text encoders may come from the different corresponding pre-training tasks and this will be an interesting future direction.

Visualization To further demonstrate the effectiveness of Multi-Sub, we visualize the representations from $\text{CLIP}_{\text{label}}$, CLIP_{GPT}, and Multi-Sub for color and species clustering tasks (Figure 6.3). In species clustering, $\text{CLIP}_{\text{label}}$ shows clear boundaries using ground truth labels, while CLIP_{GPT} introduces noise from reference words. Multi-Sub outperforms

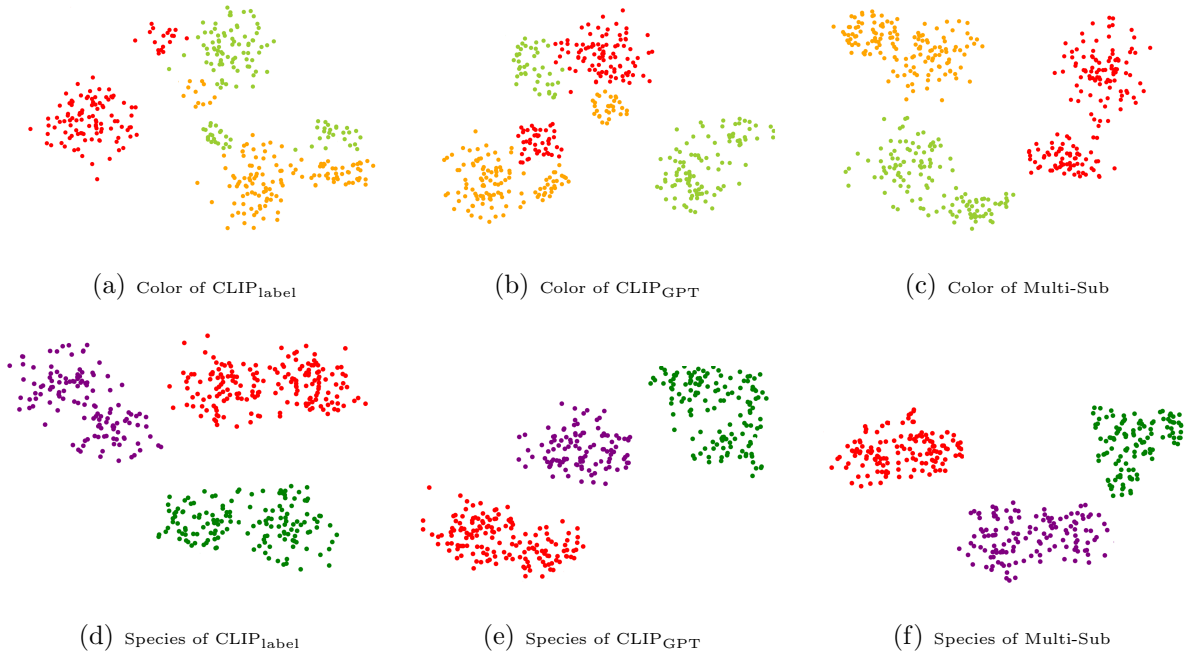
Table 6.6: MMD between different text encoders across datasets.

Dataset	Clustering	CLIP vs. ALIGN	CLIP vs. BLIP	ALIGN vs. BLIP
Fruit360	Color	0.234	0.198	0.211
	Species	0.189	0.172	0.183
Card	Order	0.215	0.202	0.219
	Suits	0.198	0.184	0.192
CMUface	Emotion	0.276	0.245	0.263
	Glass	0.231	0.217	0.225
	Identity	0.263	0.249	0.258
	Pose	0.245	0.228	0.239
Stanford Cars	Color	0.238	0.223	0.231
	Type	0.212	0.198	0.205
Flowers	Color	0.257	0.244	0.252
	Species	0.248	0.231	0.242
CIFAR-10	Type	0.193	0.178	0.186
	Environment	0.178	0.162	0.174

both by effectively capturing image features and user interests with proxy word embeddings. In color clustering, both $\text{CLIP}_{\text{label}}$ and CLIP_{GPT} focus on species features, resulting in less distinct clusters. Multi-Sub excels by clearly distinguishing colors, leveraging user-specific interests for improved alignment. Overall, Multi-Sub consistently aligns embeddings with user interests, surpassing $\text{CLIP}_{\text{label}}$ and CLIP_{GPT}, demonstrating its robust multi-modal subspace proxy learning.

6.5 Summary

In this chapter, we mitigates an important challenge in multiple clustering: effectively identifying desired clustering results based on user interests or application purposes. We introduce Multi-Sub, a novel approach that integrates user-defined preferences into a customized multi-modal subspace proxy learning framework. By leveraging the synergy between CLIP and GPT-4, Multi-Sub automatically aligns textual prompts expressing user interests with



(a) Color of $\text{CLIP}_{\text{label}}$ (b) Color of CLIP_{GPT} (c) Color of Multi-Sub

(d) Species of $\text{CLIP}_{\text{label}}$ (e) Species of CLIP_{GPT} (f) Species of Multi-Sub

Figure 6.3: Visualization of feature embeddings and related labels on Fruit dataset. For the visualization of color, red, green, and yellow points indicate the color of red, green, and yellow, respectively. For the visualization of species, red, yellow, and purple points indicate the species of apple, banana, and grapes, respectively.

corresponding visual representations. First, we observe reference words for user's interests from large language models. Given the absence of concrete class names in clustering tasks, our method uses these reference words to learn both text and vision embeddings tailored to user preferences. Extensive experiments across various visual multiple clustering tasks demonstrate that Multi-Sub consistently outperforms state-of-the-art techniques.

However, our approach has certain limitations. The reliance on large language models like GPT-4 can introduce biases inherent in these models, potentially affecting the clustering outcomes. Additionally, the field of multiple clustering lacks large, diverse datasets, which limits comprehensive evaluation. Although we have annotated CIFAR-10, more extensive datasets are needed.

Chapter 7

CONCLUSIONS

7.1 Summary

This dissertation advances *user-preference–guided representation learning* for deep multiple clustering. The four studies in this dissertation approach the problem from different angles: (i) data-centric elicitation of complementary factors, (ii) model-centric disentanglement tailored for clustering, (iii) intent-centric multimodal conditioning that grounds user concepts in vision embeddings, and (iv) coupled optimization that learns preference-aligned representations and cluster assignments jointly. Our goal is to make multiple clustering both accurate and diverse, while still being guided by the user: the resulting partitions should reflect the factors the user actually cares about and require only limited interaction. My work mainly includes the following four aspects:

(1) **From data perturbations to aspect-aligned features (AugDMC).** We introduced AugDMC, which leverages targeted data augmentations to preserve distinct aspects (e.g., color vs. shape) and a prototype-based self-supervised objective with a stabilization strategy. In this view, different augmentations act as selectors for different aspects, so we can obtain several interpretable clusterings without hand-crafted features.

(2) **From representation diversity to clustering-aware disentanglement (DDMC).** We proposed DDMC, a dual-disentangled framework optimized via a variational EM procedure. A coarse–fine factorization module learns latent factors at two levels of detail, and a clustering-aware M-step refines the decision boundaries. In this chapter, we explicitly link disentanglement and clustering objectives so that the learned features are more directly aligned with the final partitions.

(3) **From preference capture to multimodal proxy learning (Multi-MaP).** We

then bridged user intent and vision features via Multi-MaP, which aligns frozen CLIP encoders with user-specified high-level concepts through learnable text proxies. To stabilize proxy learning in CLIP's discrete token space, we introduced concept-level and reference-word constraints, operationalized with LLMs (e.g., GPT-4) to surface candidate reference vocabularies. Our experiments with Multi-MaP show that CLIP embeddings contain multiple semantic aspects that can be used for personalized clustering.

(4) **From decoupled pipelines to joint subspace–clustering optimization (Multi-Sub).** Finally, we addressed two practical limitations—requiring user-provided contrastive concepts and decoupling representation learning from clustering—by proposing a concept-conditioned subspace proxy framework. Reference words from LLMs span a low-dimensional subspace in which image-aligned proxies are learned. Crucially, Multi-Sub jointly optimizes concept-aligned representations and cluster assignments, improving both performance and efficiency while reducing interaction friction.

7.2 Limitations and Future Work

Despite the progress, several limitations remain and open up promising directions for future work:

- **User-guided but not yet fully user-friendly.** All methods in this dissertation are user-guided: they assume that the user can articulate high-level concepts as textual prompts, which then steer representation learning and clustering. This is still some distance from a fully user-friendly system, because it does not automatically propose what the user might care about without explicit input. An important future direction is to move toward proactively assistive systems that: (i) mine candidate clustering aspects from the data itself, (ii) summarize these aspects in natural language via LLMs, and (iii) present a small set of candidate clusterings that the user can inspect, accept, or refine, so that the required interaction remains light.

- **Dependence on foundation models and limited domain coverage.** Multi-MaP

and Multi-Sub rely on CLIP and GPT-style models that are primarily trained on natural images and generic web text. As a result, they inherit the biases, vocabulary gaps, and failure modes of these foundation models, and their behavior in specialized domains such as medical imaging, radiology, remote sensing, or non-visual data (e.g., time series, tabular EHR, scientific text) is not yet validated. A natural future direction is to instantiate the same user-guided multiple clustering principles on top of domain-specific encoders (e.g., radiology-focused vision transformers, biomedical language models), and to study how augmentation strategies, disentanglement modules, and multimodal proxies need to be adapted for safety-critical settings such as radiology and broader medical applications. From an application standpoint, radiology and the broader medical field are particularly promising yet challenging testbeds for user-guided multiple clustering. Radiologists and clinicians routinely reason along multiple axes—for example, anatomy, imaging modality, acquisition protocol, lesion type, disease stage, and response to treatment—and often need to explore patient cohorts under different combinations of these aspects. In principle, the mechanisms developed in this dissertation could be transferred to such settings by (i) replacing CLIP with encoders pre-trained on medical images and reports (e.g., chest X-ray or CT transformers coupled with radiology-report language models), (ii) constraining the LLM components to clinically validated, domain-specific models, and (iii) keeping domain experts in the loop to define and refine high-level concepts (e.g., "primary tumor burden", "treatment-related change" versus "disease progression") and to audit failure cases. However, any deployment in radiology or medicine would require careful validation for safety, fairness, and robustness, as well as explicit assessment of whether the discovered aspects correspond to clinically meaningful and actionable groupings. Carefully designed studies along these lines are an important next step.

- **Scalability on truly large-scale datasets and systems.** While the proposed methods scale reasonably well to the benchmark datasets used in this dissertation, they have

not been rigorously evaluated on web-scale or industry-scale corpora with millions of items and many concurrent users. Joint optimization and per-image proxy learning introduce nontrivial computational overheads. From an engineering perspective, making multiple clustering practical at scale will require: (i) distributed and parallel training of the representation and clustering modules, (ii) efficient approximate nearest neighbor indexing in the learned subspaces, (iii) caching and batching strategies for LLM calls, and (iv) streaming and incremental updates so that clusterings can be maintained under evolving data. We leave a more systematic study of these systems issues to future work.

- **Remaining quality gaps even on moderate benchmarks.** Although Multi-Sub achieves state-of-the-art performance on all public visual multiple clustering benchmarks, the absolute NMI/RI scores for some clusterings remain far from perfect, even on datasets that are not particularly challenging. This suggests that there are still modeling gaps, for example due to ambiguous ground-truth semantics, misalignment between CLIP's embedding geometry and the clustering labels, limited expressivity of the current subspace proxy family, or suboptimal coupling between the proxy learning and clustering losses. Future work could test richer proxy parameterizations (e.g., mixtures of subspaces, nonlinear subspace heads, or multiple proxies per aspect), stronger task-specific backbones, and active or human-in-the-loop refinement to reduce these gaps.

- **Granularity and compositionality of intent.** The methods in this dissertation mostly handle one concept at a time: the user asks for a clustering with respect to color or species (or pose, identity, etc.), and the model returns one aspect-specific partition per concept. In practice, however, users often care about higher-order, compositional intents such as "color and species" simultaneously—for example, grouping fruits by species while further separating them by color, or directly discovering clusters corresponding to joint configurations like "red apples", "green apples", "yellow ba-

nanas", and so on. To support such multi-aspect intents, we need explicit mechanisms for combining proxies across concepts, for example through hierarchical clusterings (first species, then color), product or intersection subspaces over multiple concept-specific proxies, or simple logical operators (AND/OR/NOT) defined in proxy space. Extending our methods so that users can flexibly specify, compose, and interact with multi-concept clusterings is a natural direction for future work.

- **Evaluation beyond NMI/RI.** Standard clustering metrics such as NMI and RI are useful but do not fully capture whether a user's preferences have been satisfied, how interpretable the discovered aspects are, or how much effort the user must invest to arrive at a useful clustering. Future work should also include human-centric evaluation, for example measuring preference satisfaction, explanation quality, time needed to reach a decision, or downstream task performance in realistic workflows (e.g., diagnosis support, dataset curation, retrieval), to get a more complete picture of usefulness.

- **Multiple clustering as a component in LLM-based agents.** The current work treats multiple clustering as a standalone algorithmic objective rather than as a tool that can be orchestrated by an LLM agent. A promising direction is to integrate user-guided multiple clustering into agentic pipelines, where an LLM decides when to call a clustering tool, which aspects to request (possibly inferred from conversation context), and how to use the resulting partitions—for example, to structure long-term memory, to personalize retrieval and recommendation, or to organize complex workspaces for downstream reasoning. Designing interfaces, APIs, and feedback loops that allow agents to leverage multiple clustering as a reusable component is an exciting avenue for future work.

7.3 Closing Remarks

This dissertation demonstrates that who the clustering is for should inform what is represented and how partitions are formed. We first use augmentation-driven aspect elicitation

(AugDMC), then introduce clustering-aware disentanglement (DDMC), then ground user intent via multimodal proxies (Multi-MaP), and finally couple subspace learning with clustering (Multi-Sub). Together, these steps move us toward user-guided deep multiple clustering. Concretely, the proposed techniques can help practitioners in real-world settings—such as clinicians exploring patient cohorts along clinically meaningful axes, scientists organizing multi-faceted experimental data, or product teams curating and segmenting large content and user collections—to obtain clusterings that reflect their actual analytical questions rather than opaque model biases. As LLM-based agents and interactive analytics systems become more prevalent, user-guided multiple clustering can also serve as a mechanism for structuring long-term memory, personalizing retrieval, and organizing complex information spaces in a controllable way.

We hope these ideas—preference conditioning, disentangled and subspace-aware representations, and joint optimization—serve as building blocks for systems that are not only statistically sound, but also deployable in high-stakes, data-rich environments and genuinely aligned with human goals, constraints, and values.

BIBLIOGRAPHY

[1] Mahdi Abavisani, Alireza Naghizadeh, Dimitris Metaxas, and Vishal Patel. Deep subspace clustering with data augmentation. *Advances in Neural Information Processing Systems*, 33:10360–10370, 2020.

[2] Mohamed Abd Elaziz, Mohammed AA Al-Qaness, Esraa Osama Abo Zaid, Songfeng Lu, Rehab Ali Ibrahim, and Ahmed A. Ewees. Automatic clustering method to segment covid-19 ct images. *PLoS One*, 16(1):e0244416, 2021.

[3] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, Massachusetts, 1985.

[4] Laith Abualigah, Amir H Gandomi, Mohamed Abd Elaziz, Abdelazim G Hussien, Ahmad M Khasawneh, Mohammad Alshinwan, and Essam H Houssein. Nature-inspired optimization algorithms for text document clustering—a comprehensive analysis. *Algorithms*, 13(12):345, 2020.

[5] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[6] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.

[7] Francis R Bach and Michael I Jordan. Learning spectral clustering, with application to speech separation. *The Journal of Machine Learning Research*, 7:1963–2001, 2006.

[8] Eric Bae and James Bailey. Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *ICDM*, pages 53–62. IEEE, 2006.

[9] Eric Bae, James Bailey, and Guozhu Dong. A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings. *Data Mining and Knowledge Discovery*, 21(3):427–471, 2010.

[10] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR, 2022.

[11] James Bailey. Alternative clustering analysis: A review. *Data Clustering*, pages 535–550, 2018.

[12] Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. *arXiv preprint arXiv:1907.05789*, 2019.

[13] Robert Baumgartner, Georg Gottlob, and Sergio Flesca. Visual information extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Databases*, pages 119–128, Rome, Italy, September 2001. Morgan Kaufmann.

[14] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[15] Christos Bouras and Vassilis Tsogkas. Clustering user preferences using w-kmeans. In *2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems*, pages 75–82. IEEE, 2011.

[16] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014.

[17] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April–June 1985.

[18] Maria Brbić and Ivica Kopriva. Multi-view low-rank sparse subspace clustering. *Pattern recognition*, 73:247–258, 2018.

[19] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[20] Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *Twenty-Third International Joint conference on artificial intelligence*, 2013.

[21] Lele Cao, Sahar Asadi, Wenfei Zhu, Christian Schmidli, and Michael Sjöberg. Simple, scalable, and stable variational deep clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 108–124. Springer, 2020.

[22] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–594, 2015.

[23] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[24] Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. Meta clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 107–118. IEEE, 2006.

[25] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887, 2017.

[26] Yale Chang, Junxiang Chen, Michael H Cho, Peter J Castaldi, Edwin K Silverman, and Jennifer G Dy. Multiple clustering views from multiple uncertain experts. In *International Conference on Machine Learning*, pages 674–683. PMLR, 2017.

[27] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A survey on multiview clustering. *IEEE transactions on artificial intelligence*, 2(2):146–168, 2021.

[28] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[29] Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. Improving disentangled text representation learning with information-theoretic guidance. *arXiv preprint arXiv:2006.00693*, 2020.

[30] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[31] Xuan Hong Dang and James Bailey. Generation of alternative clusterings using the cami approach. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 118–129. SIAM, 2010.

[32] Xuan-Hong Dang and James Bailey. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *Proceedings of the 16th ACM*

SIGKDD international conference on Knowledge discovery and data mining, pages 573–582, 2010.

[33] Xuan Hong Dang and James Bailey. A framework to uncover multiple alternative clusterings. *Machine Learning*, 98(1-2):7–30, 2015.

[34] Sajib Dasgupta and Vincent Ng. Mining clustering dimensions. In *ICML*, 2010.

[35] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[36] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, 4(1):65–75, 2002.

[37] Edi Faizal, Sri Hartati, and Aina Musdholifah. Multi-cluster dbscan for analysing tourism data. *International Journal of Intelligent Engineering & Systems*, 18(1), 2025.

[38] Uno Fang, Man Li, Jianxin Li, Longxiang Gao, Tao Jia, and Yanchun Zhang. A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12350–12368, 2023.

[39] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Adaptive data augmentation for image classification. In *2016 IEEE international conference on image processing (ICIP)*, pages 3688–3692. Ieee, 2016.

[40] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.

[41] Lele Fu, Sheng Huang, Lei Zhang, Jinghua Yang, Zibin Zheng, Chuanfu Zhang, and Chuan Chen. Subspace-contrastive multi-view clustering. *ACM Transactions on Knowledge Discovery from Data*, 18(9):1–35, 2024.

[42] Adrian Galdran, Aitor Alvarez-Gila, Maria Ines Meyer, Cristina L Saratxaga, Teresa Araújo, Estibaliz Garrote, Guilherme Aresta, Pedro Costa, Ana Maria Mendonça, and Aurélio Campilho. Data-driven color augmentation techniques for deep skin image analysis. *arXiv preprint arXiv:1703.03702*, 2017.

[43] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. SIAM, 2020.

[44] Gege Gao, Huaibo Huang, Chaoyou Fu, Zhaoyang Li, and Ran He. Information bottleneck disentanglement for identity swapping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3404–3413, 2021.

[45] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023.

[46] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[47] Jan-Mark Geusebroek, Gertjan J Burghouts, and Arnold WM Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61:103–112, 2005.

[48] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE international conference on computer vision*, pages 5736–5745, 2017.

[49] David Gondek and Thomas Hofmann. Conditional information bottleneck clustering. In *3rd ieee international conference on data mining, workshop on clustering large data sets*, pages 36–42, 2003.

[50] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. *Advances in neural information processing systems*, 31, 2018.

[51] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[52] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[53] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64(3):579–627, May 2002.

[54] Annie Gray, Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. Hierarchical clustering with dot products recovers hidden tree structure. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[55] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory: 16th International Conference, ALT 2005, Singapore, October 8-11, 2005. Proceedings 16*, pages 63–77. Springer, 2005.

[56] Joris Guérin and Byron Boots. Improving image clustering with multiple pretrained cnn feature extractors. In *British Machine Vision Conference 2018, BMVC 2018*, 2018.

[57] Stephan Günnemann, Ines Färber, Matthias Rüdiger, and Thomas Seidl. Smvc: semi-supervised multi-view clustering in subspace projections. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 253–262, 2014.

[58] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Ijcai*, volume 17, pages 1753–1759, 2017.

[59] Xifeng Guo, Xinwang Liu, En Zhu, Xinzhong Zhu, Miaomiao Li, Xin Xu, and Jianping Yin. Adaptive self-paced deep clustering with data augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 32(9):1680–1693, 2019.

[60] Xifeng Guo, En Zhu, Xinwang Liu, and Jianping Yin. Deep embedded clustering with data augmentation. In *Asian conference on machine learning*, pages 550–565. PMLR, 2018.

[61] Pietro Hiram Guzzi, Elio Masciari, Giuseppe Massimiliano Mazzeo, and Carlo Zaniolo. A discussion on the biological relevance of clustering results. In *Information Technology in Bio-and Medical Informatics: 5th International Conference*, pages 30–44. Springer, 2014.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[63] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, 2017.

[64] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International conference on learning representations*, volume 1, page 6, 2020.

[65] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

[66] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[67] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.

[68] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[69] Juhua Hu and Jian Pei. Subspace multi-clustering: a review. *Knowledge and information systems*, 56:257–284, 2018.

[70] Juhua Hu, Qi Qian, Jian Pei, Rong Jin, and Shenghuo Zhu. Finding multiple stable clusterings. *Knowledge and Information Systems*, 51(3):991–1021, 2017.

[71] Yongli Hu, Zuolong Song, Boyue Wang, Junbin Gao, Yanfeng Sun, and Baocai Yin. Akm 3 c: Adaptive k-multiple-means for multi-view clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(11):4214–4226, 2021.

[72] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR, 2017.

[73] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022.

[74] IJCAI Proceedings. IJCAI camera ready submission. `https://proceedings.ijcai.org/info`.

[75] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[76] Insu Jeon, Wonkwang Lee, Myeongjang Pyeon, and Gunhee Kim. Ib-gan: Disentangled representation learning with information bottleneck generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7926–7934, 2021.

[77] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

[78] HyunJun Jo, Yong-Ho Na, and Jae-Bok Song. Data augmentation using synthesized images for object detection. In *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, pages 1035–1038. IEEE, 2017.

[79] Nour Eldeen Khalifa, Mohamed Loey, and Seyedali Mirjalili. A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artificial Intelligence Review*, 55(3):2351–2377, 2022.

[80] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[81] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[82] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[83] Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 393–400, 2011.

[84] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[85] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[86] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018.

[87] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, July 1984.

[88] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, pages 198–202, Austin, Texas, August 1984. American Association for Artificial Intelligence.

[89] Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, and Qinghua Hu. Reciprocal multi-layer subspace learning for multi-view clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8172–8180, 2019.

[90] Zhaoyang Li, Qianqian Wang, Zhiqiang Tao, Quanxue Gao, Zhaohua Yang, et al. Deep adversarial multi-view clustering network. In *IJCAI*, volume 2, page 4, 2019.

[91] Ziyue Li, Hao Yan, Chen Zhang, and Fugee Tsung. Individualized passenger travel pattern multi-clustering based on graph regularized tensor latent dirichlet allocation. *Data Mining and Knowledge Discovery*, 36(4):1247–1278, 2022.

[92] James Chenhao Liang, Yiming Cui, Qifan Wang, Tong Geng, Wenguan Wang, and Dongfang Liu. Clusterfomer: Clustering as a universal visual learner. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[93] Fangfei Lin, Bing Bai, Kun Bai, Yazhou Ren, Peng Zhao, and Zenglin Xu. Contrastive multi-view hyperbolic hierarchical clustering. *arXiv preprint arXiv:2205.02618*, 2022.

[94] Zinan Lin, Kiran Koshy Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr: Disentangling generative adversarial networks with contrastive regularizers. *arXiv preprint arXiv:1906.06034*, page 60, 2019.

[95] Robert F Ling. On the theory and construction of k-clusters. *The computer journal*, 15(4):326–332, 1972.

[96] Liangchen Liu, Feiping Nie, Arnold Wiliem, Zhihui Li, Teng Zhang, and Brian C Lovell. Multi-modal joint clustering with application for unsupervised attribute discovery. *IEEE Transactions on Image Processing*, 27(9):4345–4356, 2018.

[97] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[98] E Laxmi Lydia, P Govindaswamy, S Lakshmanaprabu, and D Ramya. Document clustering based on text mining k-means algorithm using euclidean distance similarity. *Journal of Advanced Research in Dynamical & Control Systems*, 10(02-Special Issue), 2018.

[99] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[100] Dominik Mautz, Wei Ye, Claudia Plant, and Christian Böhm. Discovering non-redundant k-means clusterings in optimal subspaces. In *Proceedings of the 24th ACM*

SIGKDD international conference on knowledge discovery & data mining, pages 1973–1982, 2018.

[101] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022.

[102] Lukas Miklautz, Dominik Mautz, Muzaffer Can Altinigneli, Christian Böhm, and Claudia Plant. Deep embedded non-redundant clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5174–5181, 2020.

[103] Lukas Miklautz, Martin Teuffenbach, Pascal Weber, Rona Perjuci, Walid Durani, Christian Böhm, and Claudia Plant. Deep clustering with consensus representations. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1119–1124. IEEE, 2022.

[104] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[105] Jakub Nalepa, Michal Marcinkiewicz, and Michal Kawulok. Data augmentation for brain-tumor segmentation: a review. *Frontiers in computational neuroscience*, 13:83, 2019.

[106] Bernhard Nebel. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research*, 12:271–315, 2000.

[107] Antonio Cavalcante Araujo Neto, Jörg Sander, Ricardo JGB Campello, and Mario A Nascimento. Efficient computation and visualization of multiple density-based clustering hierarchies. *IEEE Transactions on Knowledge and Data Engineering*, 33(8):3075–3089, 2019.

[108] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[109] Hoang Vu Nguyen, Emmanuel Müller, Jilles Vreeken, Fabian Keller, and Klemens Böhm. Cmi: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 198–206. SIAM, 2013.

[110] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[111] Donglin Niu, Jennifer G Dy, and Michael I Jordan. Multiple non-redundant spectral clustering views. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 831–838, 2010.

[112] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9359–9367, 2018.

[113] Jelili Oyelade, Itunuoluwa Isewon, Funke Oladipupo, Olufemi Aromolaran, Efosa Uwoghiren, Faridah Ameh, Moses Achas, and Ezekiel Adebiyi. Clustering algorithms: their application to gene expression data. *Bioinformatics and Biology insights*, 10:BBI–S38316, 2016.

[114] Divya Pandove, Shivan Goel, and Rinkl Rani. Systematic review of clustering high-dimensional and large datasets. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):1–68, 2018.

[115] Priyadarsan Parida and Nilamani Bhoi. Fuzzy clustering based transition region extraction for image segmentation. *Engineering Science and Technology, an International Journal*, 21(4):547–563, 2018.

[116] Xi Peng, Jiashi Feng, Jiwen Lu, Wei-Yun Yau, and Zhang Yi. Cascade subspace

clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[117] ZiJie Qi and Ian Davidson. A principled and flexible framework for finding alternative clusterings. In *SIGKDD*, pages 717–726, 2009.

[118] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Tacoma Tacoma, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*, pages 6449–6457. IEEE, 2019.

[119] Qi Qian, Yuanhong Xu, and Juhua Hu. Intra-modal proxy learning for zero-shot visual categorization with clip. In *Thirty-seventh Conference on Neural Information Processing Systems, NeurIPS 2023*, 2023.

[120] Qi Qian, Yuanhong Xu, Juhua Hu, Hao Li, and Rong Jin. Unsupervised visual representation learning by online constrained k-means. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16640–16649, 2022.

[121] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[122] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[123] Liangrui Ren, Jun Wang, Zhao Li, Qingzhong Li, and Guoxian Yu. scmcs: a framework for single-cell multi-omics data integration and multiple clusterings. *Bioinformatics*, 39(4):btad133, 2023.

[124] Liangrui Ren, Guoxian Yu, Jun Wang, Lei Liu, Carlotta Domeniconi, and Xiangliang Zhang. A diversified attention model for interpretable multiple clusterings. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8852–8864, 2022.

[125] Yazhou Ren, Carlotta Domeniconi, Guoji Zhang, and Guoxian Yu. Weighted-object ensemble clustering: methods and analysis. *Knowledge and Information Systems*, 51:661–689, 2017.

[126] Yazhou Ren, Uday Kamath, Carlotta Domeniconi, and Guoji Zhang. Boosted mean shift clustering. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pages 646–661. Springer, 2014.

[127] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, S Yu Philip, and Lifang He. Deep clustering: A comprehensive survey. *IEEE transactions on neural networks and learning systems*, 2024.

[128] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[129] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE, 2022.

[130] Meitar Ronen, Shahaf E Finder, and Oren Freifeld. Deepdpm: Deep clustering with an unknown number of clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9861–9870, 2022.

[131] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520. IEEE Computer Society, 2018.

[132] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[133] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022.

[134] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1233–1240, 2013.

[135] Kihyuk Sohn, Jinsung Yoon, Chun-Liang Li, Chen-Yu Lee, and Tomas Pfister. Anomaly clustering: Grouping images into coherent clusters of anomaly types. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5479–5490, 2023.

[136] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[137] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[138] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019.

[139] Xiaoliang Tang, Xuan Tang, Wanli Wang, Li Fang, and Xian Wei. Deep multi-view sparse subspace clustering. In *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, pages 115–119, 2018.

[140] Tomoki Tokuda, Okito Yamashita, and Junichiro Yoshimoto. Multiple clustering for identifying subject clusters and brain sub-networks using functional connectivity matrices without vectorization. *Neural Networks*, 142:269–287, 2021.

[141] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[142] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[143] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

[144] H Vani, MA Anusuya, and ML Chayadevi. Fuzzy clustering algorithms-comparative studies for noisy speech signals. *Ictact J. Soft Comput.*, 9(3):1920–1926, 2019.

[145] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[146] Aparna Vyas, Soohwan Yu, and Joonki Paik. Fundamentals of digital image processing. In *Multiscale Transforms with Application to Image Processing*, pages 3–11. Springer, 2018.

[147] Jun Wang, Huiling Zhang, Wei Ren, Maozu Guo, and Guoxian Yu. Epimc: Detecting epistatic interactions using multiple clusterings. *IEEE Transactions on Computational Biology and Bioinformatics*, 19(1):243–254, 2021.

[148] Junyang Wang, Yuanhong Xu, Juhua Hu, Ming Yan, Jitao Sang, and Qi Qian. Improved visual fine-tuning with natural language supervision. *arXiv preprint arXiv:2304.01489*, 2023.

[149] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021.

[150] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International conference on machine learning*, pages 1083–1092. PMLR, 2015.

[151] Xing Wang, Jun Wang, Carlotta Domeniconi, Guoxian Yu, Guoqiang Xiao, and Maozu Guo. Multiple independent subspace clusterings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5353–5360, 2019.

[152] Shaowei Wei, Jun Wang, Guoxian Yu, Carlotta Domeniconi, and Xiangliang Zhang. Deep incomplete multi-view multiple clusterings. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 651–660. IEEE, 2020.

[153] Shaowei Wei, Jun Wang, Guoxian Yu, Carlotta Domeniconi, and Xiangliang Zhang. Multi-view multiple clusterings using deep matrix factorization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6348–6355, 2020.

[154] Shaowei Wei, Guoxian Yu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Multiple clusterings of heterogeneous information networks. *Machine Learning*, 110(6):1505–1526, 2021.

[155] Yuxiang Wei, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo. Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6721–6730, 2021.

[156] JV White, Sam Steingold, and CG Fournelle. Performance metrics for group-detection algorithms. *Proceedings of Interface*, 2004, 2004.

[157] Stefan Winkler. *Color space conversions*. Digital Video Qual, 2005.

[158] Tina Wong, Randy Katz, and Steven McCanne. An evaluation of preference clustering in large-scale multicast applications. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE*

Computer and Communications Societies (Cat. No. 00CH37064), volume 2, pages 451–460. IEEE, 2000.

[159] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[160] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[161] Xiaolong Xiong, Jinhan Cui, Jiaxiong Liu, Shuzhan Guo, and Jun Zhou. Inverse optimization for multi-view multiple clustering. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.

[162] Xiaolong Xiong, Jinhan Cui, Rui Xie, Shuzhan Guo, and Jun Zhou. Large-scale multi-view multiple clustering. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6205–6209. IEEE, 2024.

[163] Jie Xu, Chao Li, Yazhou Ren, Liang Peng, Yujie Mo, Xiaoshuang Shi, and Xiaofeng Zhu. Deep incomplete multi-view clustering via mining cluster complementarity. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8761–8769, 2022.

[164] Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, and Zenglin Xu. Deep embedded multi-view clustering with collaborative training. *Information Sciences*, 573:279–290, 2021.

[165] Jie Xu, Yazhou Ren, Huayi Tang, Xiaorong Pu, Xiaofeng Zhu, Ming Zeng, and Lifang He. Multi-vae: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9234–9243, 2021.

[166] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. Multi-level feature learning for contrastive multi-view clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16051–16060, 2022.

[167] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017.

[168] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.

[169] Sen Yang and Lijun Zhang. Non-redundant multiple clustering by nonnegative matrix factorization. *Machine Learning*, 106(5):695–712, 2017.

[170] Yaming Yang, Ziyu Guan, Jianxin Li, Wei Zhao, Jiangtao Cui, and Quan Wang. Interpretable and efficient heterogeneous graph convolutional network. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1637–1650, 2021.

[171] Yaming Yang, Ziyu Guan, Wei Zhao, Weigang Lu, and Bo Zong. Graph substructure assembling network with soft sequence and context attention. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4894–4907, 2022.

[172] Yan Yang and Hao Wang. Multi-view clustering: A survey. *Big data mining and analytics*, 1(2):83–107, 2018.

[173] Jiawei Yao and Juhua Hu. Dual-disentangled deep multiple clustering. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 679–687. SIAM, 2024.

[174] Jiawei Yao, Enbei Liu, Maham Rashid, and Juhua Hu. Augdmc: Data augmentation guided deep multiple clustering. In *INNS DLIA@IJCNN*, 2023.

[175] Jiawei Yao, Enbei Liu, Maham Rashid, and Juhua Hu. Augdmc: Data augmentation guided deep multiple clustering. *Procedia Computer Science*, 222:571–580, 2023.

[176] Jiawei Yao, Qi Qian, and Juhua Hu. Customized multiple clustering via multi-modal subspace proxy learning. *arXiv preprint arXiv:2411.03978*, 2024.

[177] Jiawei Yao, Qi Qian, and Juhua Hu. Multi-modal proxy learning towards personalized visual multiple clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14066–14075, 2024.

[178] Shihong Yao, Chuli Hu, Tao Wang, and Xinyou Cui. Autoencoder-like semi-nmf multiple clustering. *Information Sciences*, 572:331–342, 2021.

[179] Shixin Yao, Guoxian Yu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Multi-view multiple clustering. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4121–4127, 2019.

[180] Wei Ye, Samuel Maurus, Nina Hubig, and Claudia Plant. Generalized independent subspace clustering. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 569–578. IEEE, 2016.

[181] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical image analysis*, 58:101552, 2019.

[182] Guoxian Yu, Liangrui Ren, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Multiple clusterings: Recent advances and perspectives. *Computer Science Review*, 52:100621, 2024.

[183] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. Latent multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4279–4287, 2017.

[184] Hongjun Zhang, Ruoyan Xia, Hao Ye, Desheng Shi, Peng Li, and Weibei Fan. Multi-cluster high performance computing method based on multimodal tensor in enterprise resource planning system. *Physical Communication*, 62:102231, 2024.

[185] Huiling Zhang, Guoxian Yu, Wei Ren, Maozu Guo, and Jun Wang. Epintmc: Detecting epistatic interactions using multiple clusterings. In *International Symposium on Bioinformatics Research and Applications*, pages 56–67. Springer, 2020.

[186] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer, 2022.

[187] Amy Zhao, Guha Balakrishnan, Fredo Durand, John V Guttag, and Adrian V Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8543–8553, 2019.

[188] Yaliang Zhao, Laurence T Yang, and Jiayu Sun. Privacy-preserving tensor-based multiple clusterings on cloud for industrial iot. *IEEE Transactions on Industrial Informatics*, 15(4):2372–2381, 2018.

[189] Yaliang Zhao, Laurence T Yang, and Ronghao Zhang. A tensor-based multiple clustering approach with its applications in automation systems. *IEEE Transactions on Industrial Informatics*, 14(1):283–291, 2017.

[190] Yaliang Zhao, Laurence T Yang, and Ronghao Zhang. Tensor-based multiple clustering approaches for cyber-physical-social applications. *IEEE Transactions on Emerging Topics in Computing*, 8(1):69–81, 2018.

[191] Huasong Zhong, Chong Chen, Zhongming Jin, and Xian-Sheng Hua. Deep robust clustering by contrastive learning. *arXiv preprint arXiv:2008.03030*, 2020.

[192] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[193] Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3):1–38, 2024.

[194] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. *Advances in neural information processing systems*, 31, 2018.

[195] Xinqi Zhu, Chang Xu, and Dacheng Tao. Where and what? examining interpretable disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5861–5870, 2021.

[196] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European conference on computer vision*, pages 566–583. Springer, 2020.

[197] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.