Universal Guideline-Driven Image Clustering via a Hybrid LLM Agent

Wenliang Zhong[1], Rob Barton[2], Lucas Goncalves[2], Kushal Kumar[2], Feng Jiang[1], Hehuan Ma[1], Yuzhi Guo[1], Vidit Bansal[2], Karim Bouyarmane[2], and Junzhou Huang[1]
[1]The University of Texas at Arlington, [2]Amazon
{wxz9204, fxj8843, hehuan.ma, yuzhi.guo}@mavs.uta.edu
{rab, sglucas, kushlku, bansalv, bouykari}@amazon.com
jzhuang@uta.edu

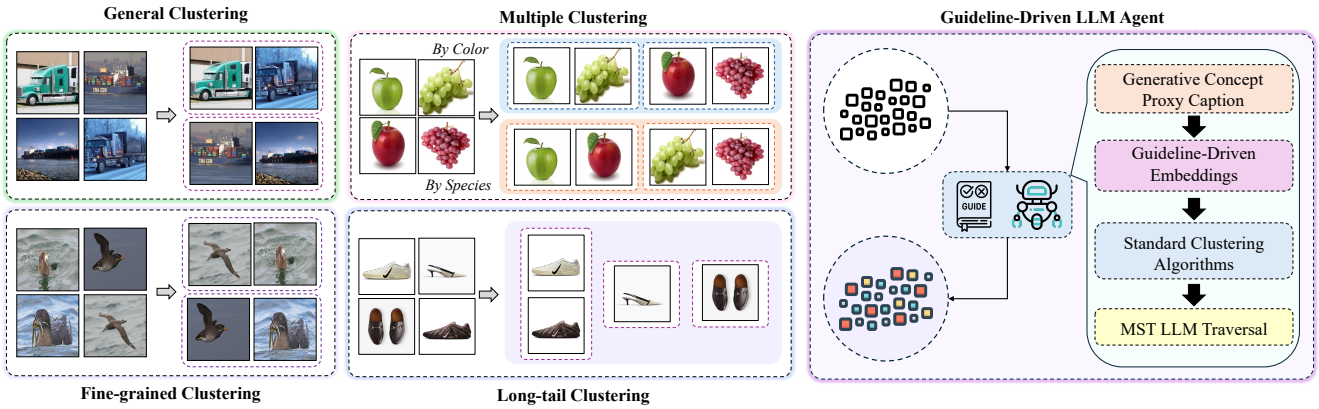

Figure 1. **Overview of our Guideline-Driven Clustering Agent**. We introduce the first universal clustering framework that handles diverse image clustering scenarios through textual guidelines, spanning from general to fine-grained tasks, from global to local criteria, and from balanced to long-tail distributions. Our training-free hybrid agent flexibly adapts across these diverse clustering requirements.

Abstract

Unifying image clustering across different clustering scenarios remains challenging due to fundamental gaps among tasks. We introduce a Guideline-Driven Image Clustering Agent, the first universal framework that bridges these gaps through textual guidelines. To incorporate complex guidelines without task-specific training, we propose Generative Concept Proxy Modeling, which generates guideline-aware embeddings via concept proxy extraction. For scenarios requiring automatic cluster discovery, we introduce LLM Traversal based on Minimum Spanning Tree that selectively applies LLM reasoning for complex semantic judgments. Our method generalizes across diverse clustering scenarios spanning from general to fine-grained categorization, from global to local criteria, and from balanced to long-tail distributions. Our framework consistently outperforms specialized methods across diverse clustering tasks. Project page is at: https://clustering-agent.github.io/.

1. Introduction

Image clustering, the task of grouping visually similar images into coherent categories, is fundamental to numerous applications ranging from content organization to visual search systems. Traditional clustering approaches rely on static encoders to generate embeddings and use clustering algorithms such as K-Means [25, 56] and DBSCAN [18], which fundamentally rely on mathematical distance metrics in feature spaces rather than semantic understanding of visual content. Recent deep clustering methods [30, 46, 51, 65] have introduce specific training strategies to steer the clustering process. However, a critical gap persists: methods optimized for general object categorization fail at fine-grained distinctions, approaches designed for balanced distributions struggle with long-tail data, and techniques for single-perspective clustering cannot handle multiple simultaneous criteria. Most fragmentation necessitates task-specific solutions, limiting practical deployment of clustering systems across diverse requirements.

To address this fundamental limitation, we aim to introduce the first universal clustering framework driven by textual guidelines. Our approach accepts natural language descriptions specifying clustering criteria, from simple directives ("group by color") to complex multi-attribute requirements ("organize athletic footwear by brand and intended activity"). This guideline-driven paradigm unifies diverse

clustering tasks spanning general to fine-grained categorization, global to local criteria, and balanced to long-tail distributions under a single framework.

Existing text-guided approaches [5, 7, 24, 34, 45, 60, 63, 64, 80] face fundamental limitations in processing such guidelines. They handle only single, concrete criteria at a time [88, 89] (e.g., group by color OR species, not both), require task-specific training for new criteria [11, 20], or assume predetermined cluster numbers [60]. These constraints prevent handling composite criteria and abstract semantic reasoning essential for real-world applications.

To incorporate complex, multi-faceted guidelines without task-specific training, one intuitive solution is to use instruction-aware embedders [71, 79, 92] and treat guidelines as instructions. However, direct encoding of images with guidelines faces significant challenges. First, existing multimodal instruction-aware embedders [59, 91, 92] fail to process images following complex guidelines. Second, they may cause important attributes in the guideline to be overshadowed by visually dominant but irrelevant features. As a solution, we propose Generative Concept Proxy Modeling (GCPM). GCPM addresses this through a two-stage process, by first extracting concept proxy captions via multimodal large language models (MLLMs), then encoding these concept proxies using instruction-aware embedders. This achieves attribute disentanglement and enables guideline composition within embeddings that traditional models fail to capture. The resulting embeddings support standard clustering algorithms, e.g., K-Means with cluster number prior or HDBSCAN [6] for automatic discovery.

For automatic clusters discovery, existing algorithms provide limited similarity comparison based on embeddings. While recent Large Language Models [1, 3] (LLMs) can provide sophisticated reasoning capabilities for understanding complex guidelines and making sophisticated similarity judgments, existing approaches [19, 33, 47, 52, 75, 93] show critical computational concerns using LLMs. To address this challenge while preserving semantic reasoning capabilities, we introduce an LLM Traversal algorithm based on Minimum Spanning Tree (MST). This hybrid design leverages embedding-based efficiency for routine clustering decisions while applying selective LLM reasoning only where semantic complexity demands it.

The main contributions of this work are: (1) We introduce the first universal clustering framework capable of handling composite and abstract textual guidelines without requiring task-specific training, enabling flexible clustering according to complex, multi-criteria requirements. (2) We propose GCPM, a training-free embedding approach that efficiently incorporate guideline into embeddings that previous methods fail to achieve. GCPM captions augment critical attributes of images for better representation and disentanglement. (3) We present MST-based LLM traversal, a novel algorithm that significantly minimizes expensive LLM invocations by intelligently selecting which clustering decisions require deep semantic reasoning. (4) We provide comprehensive evaluation across four distinct clustering tasks, demonstrating the superior generalization and performance of our approach.

2. Related Works

General Deep Clustering Methods like [46, 51] propose specific training strategies for unsupervised learning. Though many of them achieve impressive performance in general clustering, they require dedicated training and are often difficult to extend to different scenarios.

Text-guided and Multiple Clustering Methods have emerged to address semantic clustering needs. Recent advances include Multi-Sub [89] and Multi-MaP [88], which enable user-specified perspectives through proxy learning approaches. TAC [45] leverages external concepts to enhance general clustering. However, they are fundamentally limited to single-criteria clustering or constrained to specific scenarios due to their embedding strategies.

Clustering Methods based on LLMs Existing methods leverage language models' semantic capabilities but face efficiency challenges. ClusterLLM [93] guides clustering through triplet comparisons yet requires embedding fine-tuning and handles only single criteria. While these approaches [19, 33, 47, 52, 75, 93] demonstrate LLMs' potential, they suffer from computational inefficiency requiring extensive model invocations that limit practical deployment. IC|TC [39] pioneered training-free image clustering pipelines via LLMs but remains constrained to single, concrete criteria and costly iterations over the dataset.

Fine-grained Deep Clustering Methods [4, 36, 75] like DiFiC [17] achieve superior performance through diffusion model semantic extraction in fine-grained clustering. However, these require dataset-specific training and lack mechanisms for incorporating user-specified guidelines.

Our work differs fundamentally by processing composite criteria under various scenarios, achieving semantic sophistication and training-free efficiency through a hybrid approach. We carefully discuss our difference in Appendix D.

3. Methodology

Our Guideline-driven clustering agentic framework addresses the challenge of clustering images according to complex textual guidelines. As shown in Figure 2, the framework operates through a two-stage hybrid approach that balances semantic understanding with computational efficiency, enabling training-free generalization across diverse clustering tasks. The first stage, Generative Concept Proxy Modeling, generates guideline-aware embeddings suitable for efficient clustering using traditional algo-

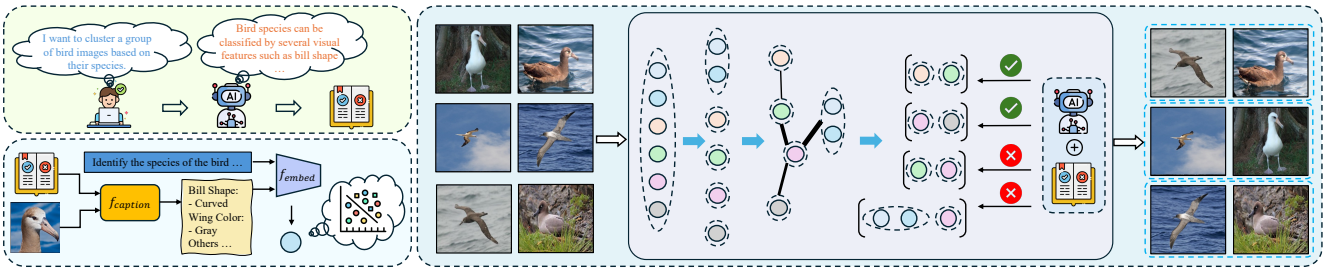

Figure 2. **Overview of the Clustering Framework**. **Bottom left**: Generative Concept Proxy Modeling extracts guideline-aware textual descriptions from images via multimodal LLM and encodes them into embeddings for efficient clustering; **Right**: MST-based LLM Traversal refines initial clusters by constructing a Minimum Spanning Tree and selectively querying the LLM for semantic merging decisions. **Top left**: For scenarios without available guidelines, we introduce heuristic prompts for automatic guideline generation.

rithms. The second stage employs MST-based LLM-guided Traversal to handle complex semantic judgments that require deep understanding of abstract concepts.

3.1. Preliminaries

Problem Formulation Given a guideline G and a set of samples $X = \{x_1, x_2, \cdots, x_N\}$, our objective is to find a function f that groups samples into clusters $\mathcal{C} = \{C_1, C_2, \cdots, C_M\}$, where each cluster $C_i = \{x_1^i, x_2^i, \cdots\}$ contains semantically similar samples according to the specified guideline:

$$\mathcal{C} = f(G, X). \tag{1}$$

The guideline G can be either directly provided by users such as standard operating procedures in business problems or derived from LLM-based knowledge through our heuristic prompting inspired by data substructure exploration.

Guideline Processing We assume guidelines are in textual form, containing multiple criteria that users wish to consider for clustering. Formally, a guideline G contains a set of grouping attributes $A = \{a_1, a_2, \cdots, a_k\} \subseteq G$, where A represents the universal set of possible attributes. For instance, in bird species clustering [77], G might include attributes such as "tail shape" and "wing color". In scenarios where explicit guidelines are unavailable, we leverage the internal knowledge of LLMs [1] to generate appropriate clustering guidelines. We introduce a series of heuristic prompts (shown in Appendix C.1) that require only basic understanding of the clustering objectives from users. For datasets where users have limited prior knowledge, unsupervised data substructure exploration techniques [53, 55] can be employed to identify fundamental criteria, which are then fed to the LLM for guideline generation. In our experiments, we evaluate performance on both datasets with predefined guidelines and those requiring LLM-generated guidelines.

3.2. Generative Concept Proxy Modeling

The Generative Concept Proxy Modeling (GCPM) generates guideline-aware embeddings suitable for efficient clustering. GCPM aims to address fundamental challenges in guideline-driven clustering: (1) combining textual guidelines with visual input to produce semantically meaningful representations requires careful consideration of how attributes in the guideline interact; (2) clustering tasks demand flexible dimensional focus: comprehensive tasks like species clustering [35, 61, 77] require holistic consideration of all attributes, while multiple clustering tasks [88, 89] like poker card categorization [21] on suits need emphasis on specific dimensions; (3) achieving generalization across domains without dataset-specific training remains challenging for existing deep clustering approaches [20, 46].

To address these challenges, we leverage recent instruction-aware embedding models [31, 79, 95, 96, 98] that can incorporate textual instructions during the encoding process. These models offer the potential for training-free guideline adherence by directly encoding images alongside clustering instructions. However, directly encoding images and guidelines using multimodal instruction-aware embedding models [32, 48, 54, 92] introduces critical limitations that prevent effective clustering. In comprehensive clustering scenarios, fine-grained attributes mentioned in guidelines may be overshadowed if they are not explicitly queried. Moreover, in dimension-specific clustering, irrelevant but visually salient attributes can dominate the intended attributes.

We address this through a concept proxy approach. We first employ a multimodal LLM (MLLM) [3] as a captioning model $f_{caption}$ to extract concept-focused descriptions:

$$c_i = f_{caption}(A, x_i), A \subseteq G. \tag{2}$$

This process explicitly surfaces guideline-specified attributes in textual form, achieving attribute disentanglement. Subsequently, we use an instruction-aware embedding model f_{embed} to encode the concept proxy caption:

$$h_i = f_{embed}(S, c_i), S \subseteq G, \tag{3}$$

where S specifies the clustering focus required. It can be a single attribute or a global definition of all attributes in A.

The resulting embeddings can be directly applied to standard clustering algorithms such as K-Means:

$$\mathcal{C} = \text{Clustering}(H), \tag{4}$$

where $H = \{h_1, h_2, \cdots, h_N\}$. This training-free approach enables cross-domain generalization while outperforming supervised methods on public benchmarks.

Algorithm 1 MST-based LLM Traversal

Require: Initial clusters $C^{(0)} = \{C_1, C_2, \ldots, C_M\}$ from HDBSCAN, Singleton points S, Guideline G, LLM merging function f_{merge}, Embeddings H
Ensure: Refined clusters $C = \{C_1, C_2, \ldots, C_K\}$

$C \leftarrow C^{(0)} \cup \{\{s\} \mid s \in S\}$
$M \leftarrow |C|$ // Total number of clusters
while $M > 1$ **do**
  Compute Ward distance matrix $D \in \mathbb{R}^{M \times M}$ for all cluster pairs using Eq. (6)
  $T \leftarrow \text{MST}(D)$ // Construct minimum spanning tree
  Sort edges in T by distance in ascending order
  merged $\leftarrow$ False
  for each edge $(C_i, C_j) \in T$ in sorted order **do**
    if (C_i, C_j) has been rejected in previous iterations **then**
      continue // Skip pairs queried and rejected before
    end if
    $p \leftarrow f_{\text{merge}}(G, C_i, C_j)$ // Query LLM for decision. Each cluster is represented by GCPM captions of samples.
    if $p = 1$ **then**
      $C_{\text{new}} \leftarrow C_i \cup C_j$ // Merge clusters
      $C \leftarrow (C \setminus \{C_i, C_j\}) \cup \{C_{\text{new}}\}$
      Update subsequent edges containing C_i or C_j
      $M \leftarrow M - 1$
      merged $\leftarrow$ True
    end if
  end for
  if merged $=$ False **then**
    break // No clusters merged in this iteration
  end if
end while
return C

3.3. MST-based LLM Traversal

GCPM provides a training-free and efficient embedding approach for clustering. For scenarios where cluster numbers are predetermined, K-Means can be directly applied. However, in many real-world tasks, clusters are automatically discovered following guideline specifications rather than predefined knowledge. An intuitive solution is to apply algorithms that do not require predefined cluster numbers, such as HDBSCAN [6], to guideline-aware embeddings.

$$\mathcal{C} = \text{HDBSCAN}(H). \tag{5}$$

In practice, we observe that HDBSCAN tends to generate consistent small clusters but fails to merge homogeneous clusters into larger groups. It indicates that HDBSCAN effectively groups clearly similar samples together. But for scenarios involving larger sample volumes or more complex sample variations, it is less effective due to the density-based nature. To address this limitation, we propose using LLMs to merge these small clusters. LLMs can effectively understand complex guidelines and measure sample similarity at the semantic level, enabling more precise clustering decisions that go beyond embedding-based metrics.

However, LLM-based approaches [19, 33, 47, 52, 75, 93] are computationally expensive and time-consuming. Assuming HDBSCAN produces M clusters, a naive approach would require $O(M^2)$ LLM comparisons to determine which clusters should be merged. To improve efficiency, we propose a Minimum Spanning Tree (MST) based traversal algorithm that strategically reduces the number of required LLM queries while maintaining clustering quality.

Ward Distance Computation For all clusters generated by HDBSCAN, we compute pairwise distances using Ward distance [82], which measures the increase in sum of squares when merging two clusters C_1 and C_2:

$$\begin{aligned} d(C_1, C_2) = &\sum_{i \in C_1 \cup C_2} \|h_i - m_{C_1 \cup C_2}\|^2 \\ &- \sum_{i \in C_1} \|h_i - m_{C_1}\|^2 - \sum_{i \in C_2} \|h_i - m_{C_2}\|^2 \\ = &\frac{|C_1| \cdot |C_2|}{|C_1| + |C_2|} \|m_{C_1} - m_{C_2}\|^2, \end{aligned} \tag{6}$$

where m_{C_i} is the centroid of the cluster C_i. This distance metric is particularly suitable for hierarchical clustering as it considers both cluster size and centroid separation.

MST Construction and Traversal: We construct an MST T from the distance matrix $D \in \mathbb{R}^{M \times M}$:

$$T = \text{MST}(D). \tag{7}$$

The MST provides a traversal order that prioritizes cluster pairs with smallest distances, ensuring that the most promising merge candidates are evaluated first by the LLM.

Iterative Guideline-Driven LLM Merging Instead of merging clusters recursively until all are combined, we iteratively query an LLM to decide whether pairs of clusters should be merged. In each iteration, after computing the distance matrix and MST, we present cluster pairs to the merging LLM [3] f_{merge} in order of increasing distance:

$$p = f_{merge}(G, C_i, C_j), \text{for each edge } (C_i, C_j) \in T. \tag{8}$$

The LLM evaluates a cluster pair based on the provided guideline G and generates a binary decision $p \in \{0, 1\}$ whether they should be merged. Each cluster is represented by GCPM captions of Top-K (K=5) samples nearest to the centroid. The process continues until either all appropriate clusters are merged or the LLM decides no further merging is beneficial. To further improve efficiency, we cache LLM decisions from previous iterations to avoid redundant queries for previously evaluated pairs. We discuss its efficiency improvement in the Appendix A.2.

Figure 3. Examples from ABO-LC. Each item is represented by an image and an item name. Different clusters have various sizes.

This hybrid design leverages embedding-based efficiency for initial clustering while applying selective LLM reasoning for complex semantic judgments. The complete process is in Algorithm 1. The MST Traversal optimally allocates LLM resources by prioritizing the most promising merge candidates. Moreover, it naturally supports *incremental clustering* [8] by treating new samples as individual clusters and merging them with existing structures, while many traditional algorithms require re-clustering.

Theoretical Complexity Analysis We provide theoretical justification for the efficiency of our MST-based approach. Based on empirical observations and reasonable assumptions about merge probability patterns, we prove that the expected number of LLM calls is $O(M \log M)$, compared to $O(M^2)$ for naive pairwise comparison methods. The complete proof is provided in Appendix A.1.

3.4. Long-Tail E-Commerce Clustering

To evaluate our approach on challenging realistic guideline-driven scenarios, we introduce a dataset that reflects the long-tail clustering challenges commonly encountered in e-commerce. The data is derived from the Amazon Berkeley Objects (ABO) dataset [14], utilizing product images and textual attributes such as brand, manufacturer, and material.

We perform grouping based on meaningful attribute combinations that reflect real-world product organization practices. The resulting dataset exhibits the long-tail distribution characteristic of commercial platforms, where numerous specialized products form singleton or very small clusters. Examples are in Figure 3. In summary, the dataset contains 10,756 products organized into 4,952 ground truth clusters, with 78.7% of clusters containing two or fewer samples. We provide more statistics and discuss how data is processed, filtered, and grouped in Appendix B.1.

The long-tail characteristic makes traditional clustering algorithms less effective, as methods like K-Means assume balanced cluster sizes and struggle with extreme imbalance scenarios. The dataset serves as a crucial benchmark for evaluating clustering methods on realistic scenarios where cluster numbers are unknown and sizes vary dramatically.

4. Experiments

4.1. Experimental Settings

Tasks and Datasets We conduct experiments on a wide range of image clustering tasks, including general clustering (GC), multiple clustering (MC), fine-grained clustering (FC), and our proposed long-tail e-commerce clustering (LC). We select commonly used datasets for experiments: STL-10 [13], ImageNet-10 [16], and CIFAR-10 [38] for GC; CIFAR10-MC [88], Fruit [27], and Cards [87] for MC; CUB Birds [77], Stanford Dogs [35], Stanford Cars [37], Oxford Flowers [61] for FC; and the proposed ABO [14]-LC. Because evaluation metrics are different among tasks, we follow existing baselines to use metrics and process datasets in each task. The dataset statistic is shown in Table 1. We use Accuracy (ACC), Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI) for GC; NMI and Rand Index (RI) for MC; ACC and NMI for FC; ACC, NMI, and ARI for LC. We discuss details of datasets, their guideline availability, and metrics in Appendix B.2. All LLM prompts, including guideline, are in Appendix C.2. Note that we maintain an unsupervised setting because none of our prompts includes ground-truth labels.

Table 1. Statistics of clustering datasets and their criteria.

Dataset	Type	Criteria	# of Samples
CIFAR-10 [38]	GC	common objects	60,000
STL-10 [13]	GC	common objects	13,000
ImageNet-10 [16]	GC	common objects	13,000
Fruit [27]	MC	fruit color; fruit species	105
Cards [87]	MC	card number; card suits	8,029
CIFAR10-MC [88]	MC	object types; object environment	60,000
CUB Birds [77]	FC	bird species	5,794
Stanford Cars [37]	FC	car types	8,041
Stanford Dogs [35]	FC	dog breeds	8,580
Oxford Flowers [61]	FC	flower species	6,149
ABO[14]-LC (Proposed)	LC	e-commerce items	10,756

Models We use QWen2.5-VL-Instruct [3] (7B) for both generating concept proxy caption and MST Traversal though the merging process in traversal only requires texts. For embeddings, we validate our method on three distinct embedders to test the robustness: INSTRUCTOR-large [71] (335M), a regular instruction-aware text encoder; E5-Mistral [79] (7B), an LLM-based encoder; GME-Qwen2-VL [92] (7B), an MLLM-based encoder. We use only captions for INSTRUCTOR and E5-Mistral while both captions and images for GME-Qwen. Detailed forwarding processes of embedders are discussed in Appendix C.3. All models are run in inference mode without further training. Experiments are validated separately on a server with NVIDIA A6000 and a server with H100 GPUs, providing the ease for adapting to new tasks, scenarios, and data.

Baselines and Clustering Algorithms We compare with state-of-the-art baselines in each task: [15, 22, 28–30, 39–41, 43, 44, 46, 51, 57, 62, 65, 68, 70, 72–74, 76, 84, 90, 94] in GC; [23, 27, 66, 86–89] in MC; and [4, 9, 10, 26, 30, 36, 42, 65, 67, 69, 74, 78, 85] in FC. For LC, since each sample contains an image and an item name in the ABO dataset, we compare IC|TC [39] using the same MLLM as the baseline. Due to the space limitation, we discuss details and our difference from baselines in Appendix D. Note that many

Table 2. Performance comparison on general clustering. Baselines assume known number of clusters. KMS.: K-Means; HDBS.: HDBSCAN. Max Δ↑ measures improvements of MST Traversal.

Method	CIFAR-10			STL-10			ImageNet-10		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Cop-KMS. [76]	89.0	82.3	78.6	85.4	78.1	73.1	88.6	85.5	81.0
IIC [30]	61.7	51.3	41.1	49.9	43.1	29.5	-	-	-
DCCM [84]	62.3	49.6	40.8	48.2	37.6	26.2	71.0	60.8	55.5
BYOL [22]	83.3	72.3	68.0	80.7	68.0	63.5	82.1	73.2	67.4
MiCE [73]	83.5	73.7	69.8	75.2	63.5	57.5	90.1	84.2	82.2
PCL [41]	87.4	80.2	76.6	81.0	71.8	67.0	90.7	84.1	82.2
PICA [28]	64.5	56.1	46.7	-	-	-	85.0	78.2	73.3
SCAN [74]	88.3	79.7	77.2	80.9	69.8	64.6	-	-	-
FixMatch [70]	92.8	86.8	85.4	68.6	61.7	49.2	92.5	84.2	84.4
CC [43]	79.0	70.5	63.7	85.0	76.4	72.6	89.3	85.9	82.2
GCC [94]	85.6	76.4	72.8	78.8	68.4	63.1	90.7	84.1	82.2
IDFD [72]	81.5	71.4	66.6	74.1	62.3	55.0	86.2	75.5	69.0
NMM [15]	84.3	74.8	70.9	80.8	69.4	65.0	-	-	-
TCC [68]	90.6	79.0	73.3	81.4	73.2	68.9	89.7	84.8	82.0
ProPos [29]	92.0	85.6	84.1	83.2	72.1	70.4	89.0	81.2	78.7
SPICE [62]	83.8	73.4	70.5	90.8	81.7	81.2	92.1	82.8	83.6
TCL [44]	88.9	82.2	78.4	88.6	82.0	78.5	90.4	89.6	85.0
CoNR [90]	85.3	78.1	71.1	81.2	70.2	69.2	87.7	79.2	75.0
DMICC [40]	83.1	72.2	67.5	75.5	64.1	60.5	92.5	84.3	85.2
SeCu [65]	93.0	86.1	85.7	83.6	73.3	69.3	-	-	-
DivClust [57]	81.9	72.4	68.1	-	-	-	93.6	89.1	87.8
IC|TC [39]	88.4	78.9	75.9	97.4	93.9	94.4	-	-	-
IDCTCL [51]	92.7	84.4	84.8	92.7	85.3	84.6	97.2	93.2	93.9
LFSS [46]	93.4	87.2	**86.6**	86.1	77.1	74.0	93.2	85.6	85.7
KMS. w/ Known # of Clus.									
GCPM-I (Ours)	75.6	66.6	60.1	98.2	95.5	96.0	97.5	92.8	86.4
GCPM-E (Ours)	82.6	69.8	65.9	97.9	95.0	95.4	97.1	94.4	93.9
GCPM-G (Ours)	**94.1**	**87.5**	84.5	**98.8**	**96.9**	**97.4**	**98.8**	**96.7**	**97.1**
HDBS. & Our MST Traversal w/o Known # of Clus.									
GCPM-I (Ours)	-	40.1	0.3	-	43.6	0.5	-	41.3	0.2
+MST Traversal	-	53.6	43.9	-	59.1	28.9	-	75.5	72.3
GCPM-E (Ours)	-	41.4	0.3	-	44.1	0.7	-	41.8	0.3
+MST Traversal	-	59.3	56.0	-	59.5	28.8	-	75.5	70.3
GCPM-G (Ours)	-	42.0	0.8	-	45.0	2.1	-	47.0	30.9
+MST Traversal	-	53.7	47.5	-	57.6	23.2	-	76.2	72.1
Max Δ↑	-	**17.9↑**	**55.7↑**	-	**15.5↑**	**28.4↑**	-	**34.2↑**	**72.1↑**

baselines require training and assume known numbers of clusters. For clustering algorithms, we select K-Means and HDBSCAN upon GCPM depending on whether the number of clusters is known in advance. Additionally, we test MST-based LLM Traversal upon HDBSCAN to validate its effectiveness for improving clustering results. Note that for automatic cluster discovery, ACC is omitted because the number of predicted clusters may be different from the ground truths. Experiments are repeated with three random seeds and averaged.

4.2. Experimental Results

Overall Performance Tables 2, 3, 4, and 5 present comprehensive experimental results across general clustering (GC), multiple clustering (MC), fine-grained clustering (FC), and long-tail clustering (LC) tasks. Our training-free framework consistently outperforms existing state-of-the-art methods that require dataset-specific training. Notably, in GC tasks (Table 2), GCPM-G with K-Means achieves 98.8% accuracy on ImageNet-10, surpassing the previous best training-based method IDCTCL by 1.6%. Similarly, in MC tasks (Table 3), our method achieves 99.9% NMI on Fruit dataset, significantly outperforming Multi-Sub (98.5%). These results demonstrate that our guideline-driven approach, leveraging the semantic understanding of modern vision-language models, can achieve superior performance without requiring task-specific training.

Impact of Embedding Models We validate our framework's robustness using three embedding models with varying architectures and capacities. As shown across Tables 2-5, there is a clear performance hierarchy: MLLM-based embeddings (GCPM-G) generally outperform LLM-based embeddings (GCPM-E), which in turn surpass regular instruction-aware encoders (GCPM-I). For instance, on STL-10, GCPM-G achieves 98.8% accuracy compared to 97.9% for GCPM-E and 98.2% for GCPM-I. This trend reflects the enhanced multimodal reasoning capabilities of larger, more sophisticated models.

However, the Card dataset presents an interesting exception where GCPM-E (91.1% NMI for number criterion) outperforms GCPM-G (82.0% NMI). This counterintuitive result can be attributed to the dataset's inherent visual complexity because card numbers and suits are heavily visually entangled within single images. In such scenarios, the concept proxy approach of extracting textual descriptions before encoding proves more effective than direct multimodal embedding, as it explicitly disentangles these tangled visual attributes through the captioning process (also see a test case in Figure 4). This finding highlights our framework's adaptability: the GCPM effectively handles scenarios where visual features are difficult to differentiate.

Clustering Algorithm Selection Comparing K-Means and HDBSCAN reveals intricate trade-offs. When cluster numbers are known, K-Means consistently achieves superior performance across most datasets, aligning with existing findings [45]. For example, on CIFAR-10, GCPM-G with K-Means achieves 94.1% accuracy compared to 42.0% NMI with HDBSCAN alone. However, this advantage diminishes in highly imbalanced scenarios. In the ABO-LC dataset (Table 5), where 78.7% of clusters contain two or fewer samples, HDBSCAN with MST Traversal achieves better performance (51.5% ARI) without requiring prior knowledge of cluster counts, demonstrating its suitability for real-world long-tail distributions.

Effectiveness of MST Traversal The MST-based LLM Traversal consistently improves upon HDBSCAN's initial clustering across all experimental settings. The magnitude of improvement varies systematically across task types. In GC tasks, improvements are substantial. For instance, on ImageNet-10, MST Traversal improves ARI from 0.3 to 72.1. In MC tasks, improvements are more variable across different criteria: significant gains are observed for criteria requiring abstract semantic understanding (e.g., 52.9↑ in

Table 3. Performance comparison on multiple clustering, criteria, and metrics with averaged results per dataset. Baselines compared all assume known number of clusters. KMS.: K-Means; HDBS.: HDBSCAN. Max Δ↑ measures improvements of MST Traversal.

Method	Fruit						Card						CIFAR10-MC					
	Color		Species		Average		Number		Suits		Average		Type		Environment		Average	
	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI
MSC [27]	68.9	80.5	16.3	60.5	42.6	70.5	8.1	78.1	5.0	35.9	6.6	57.0	15.5	33.0	11.4	30.8	13.5	31.9
MCV [23]	62.7	76.9	27.3	66.0	45.0	71.5	7.9	71.3	4.3	36.4	6.1	53.9	16.2	33.1	13.8	33.4	15.0	33.3
ENRC [58]	71.0	85.1	31.9	65.4	51.5	75.3	12.3	73.1	6.8	38.0	9.6	55.6	18.3	34.7	18.9	36.0	18.6	35.4
iMClusts [66]	73.5	86.3	30.3	67.4	51.9	76.9	11.4	76.6	7.2	37.2	9.3	56.9	20.4	37.0	19.2	36.6	19.8	36.8
AugDMC [87]	85.2	91.1	35.5	74.0	60.4	82.6	14.4	82.7	8.7	42.3	11.6	62.5	28.6	45.2	29.3	46.9	29.0	46.1
DDMC [86]	89.7	93.8	37.6	76.2	63.7	85.0	15.6	83.3	9.3	64.7	12.5	74.0	39.9	58.3	37.8	55.5	38.9	56.9
Multi-MaP [89]	86.2	95.3	**100.0**	**100.0**	93.1	97.6	36.5	85.9	27.3	70.4	31.9	78.2	49.7	71.0	46.0	67.4	47.9	69.2
Multi-Sub [88]	96.9	99.6	**100.0**	**100.0**	98.5	99.8	39.2	88.4	31.0	79.4	35.1	83.9	52.7	73.9	48.3	71.0	50.5	72.5
KMS. w/ Known # of Clus.																		
GCPM-I (Ours)	97.4	98.6	**100.0**	**100.0**	98.7	99.3	35.3	86.7	57.3	97.0	46.3	91.8	50.7	70.9	47.9	69.5	49.3	70.2
GCPM-E (Ours)	98.8	**100.0**	**100.0**	**100.0**	99.4	**100.0**	**91.1**	**99.1**	**89.0**	**97.2**	**90.0**	**98.2**	51.6	71.3	48.3	69.5	50.0	70.4
GCPM-G (Ours)	**99.8**	**100.0**	**100.0**	**100.0**	**99.9**	**100.0**	82.0	95.8	30.6	74.9	56.3	85.4	**55.2**	**75.1**	**50.1**	**71.4**	**52.6**	**73.2**
HDBS. & Our MST Traversal w/o Known # of Clus.																		
GCPM-I (Ours)	41.9	67.5	47.2	68.9	44.6	68.2	46.3	92.3	28.2	75.0	37.3	83.7	44.6	69.0	40.9	60.1	42.8	64.6
+MST Traversal	94.5	96.7	96.7	98.8	95.6	97.8	61.3	93.5	58.5	85.1	59.9	89.3	47.1	70.2	42.8	62.1	45.0	66.2
GCPM-E (Ours)	48.2	69.7	48.9	70.3	48.6	70.0	47.4	92.3	28.7	75.0	38.1	83.7	44.5	69.6	41.6	60.7	43.1	65.2
+MST Traversal	97.6	97.3	**100.0**	**100.0**	98.8	98.6	80.3	96.7	70.5	89.8	75.4	93.2	46.9	70.7	43.7	62.5	45.3	66.6
GCPM-G (Ours)	46.2	68.4	47.1	68.8	46.7	68.6	50.3	92.5	29.9	75.2	40.1	83.9	46.0	71.1	42.8	62.8	44.4	67.0
+MST Traversal	95.3	97.2	**100.0**	**100.0**	97.7	98.6	72.1	95.1	36.9	75.9	54.5	85.5	48.6	73.8	44.2	63.7	46.4	68.8
Max Δ↑	**52.6**↑	**29.2**↑	**52.9**↑	**31.2**↑	**52.8**↑	**30.2**↑	**32.9**↑	**4.4**↑	**41.8**↑	**14.8**↑	**37.4**↑	**9.6**↑	**2.6**↑	**2.7**↑	**2.1**↑	**2.0**↑	**2.4**↑	**2.4**↑

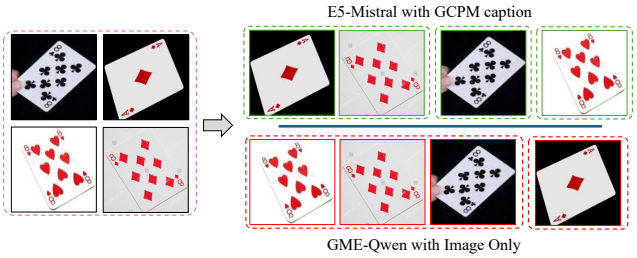

Figure 4. A disentanglement test case of cards *grouped by suits* using HDBSCAN. While using GME-Qwen with images only, the number criteria dominates the suit criteria because of the layout.

NMI for Fruit species), while criteria with clear visual patterns show modest gains.

Interestingly, improvements in FC tasks (Table 4) are relatively modest compared with in GC tasks. This behavior reflects the fundamental difference of fine-grained clustering from general clustering: distinguishing subtle discrepancy between similar categories demands extremely precise semantic judgments. Therefore, we use conservative prompts for the LLM merging process in fine-grained clustering. These prompts maintain strict rules while comparing two clusters and keep them separated unless there are clear difference in specific criteria required by guidelines. This conservative behavior, however, still yields state-of-the-art results, demonstrating that the combination of high-quality guideline-aware embeddings and selective LLM refinement is effective even in challenging fine-grained scenarios.

4.3. In-Depth Analysis

How MST Traversal Affects Clustering Results? To further analyze the effectiveness of MST Traversal, we em-

Table 4. Comparison on fine-grained clustering. Baselines except UFCL assume known cluster number. KMS.: K-Means; HDBS.: HDBSCAN. Max Δ↑ measures improvements of MST Traversal.

Method	CUB Birds		Stanford Dogs		Stanford Cars		Oxford Flowers	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
IIC [30]	7.4	36.0	5.0	18.0	4.9	27.0	8.7	24.0
SimCLR [9]	8.4	40.0	6.8	19.0	6.7	33.0	12.5	29.0
MoCo [26]	10.2	37.0	11.5	31.0	8.0	37.0	51.5	68.0
SCAN [74]	11.9	45.0	12.3	35.0	8.8	38.0	56.5	77.0
SeCu [65]	15.4	48.0	16.7	38.0	10.0	39.0	68.6	84.0
InfoGAN [10]	8.6	39.0	6.4	21.0	6.5	31.0	23.2	44.0
FineGAN [69]	12.6	40.0	7.9	23.0	7.8	35.0	-	-
MixNMatch [42]	13.6	42.0	8.9	32.0	7.9	36.0	-	-
OneGAN [4]	10.1	39.0	7.3	21.0	6.0	27.0	-	-
SD [67]	7.6	34.0	5.4	18.0	9.1	37.0	13.1	34.0
C3-GAN [36]	22.7	50.0	11.8	30.0	8.3	33.0	55.6	72.0
DiFiC [85]	31.7	61.0	17.2	39.0	47.2	68.0	72.9	88.0
UFCL [78]	-	78.6	-	45.8	-	46.5	-	70.7
KMS. w/ Known # of Clus.								
GCPM-I (Ours)	41.2	70.3	54.5	68.8	69.4	87.7	73.9	89.0
GCPM-E (Ours)	45.5	74.1	59.8	74.0	**74.6**	**90.0**	77.4	90.6
GCPM-G (Ours)	**72.9**	**89.9**	**75.1**	**85.9**	66.3	86.2	**86.0**	**94.9**
HDBS. & Our MST Traversal w/o Known # of Clus.								
GCPM-I (Ours)	-	75.7	-	70.1	-	78.3	-	75.9
+MST Traversal	-	76.3	-	69.7	-	81.2	-	77.5
GCPM-E (Ours)	-	77.9	-	71.4	-	80.2	-	82.7
+MST Traversal	-	78.4	-	72.1	-	83.4	-	84.2
GCPM-G (Ours)	-	84.9	-	76.5	-	78.8	-	88.2
+MST Traversal	-	85.9	-	78.0	-	80.9	-	88.6
Max Δ↑	-	**1.0**↑	-	**1.5**↑	-	**3.2**↑	-	**1.6**↑

ploy multi-faceted metrics analogous to those used in classification tasks, rather than relying solely on general metrics such as NMI. Specifically, we adopt BCubed [2] Precision and Recall metrics. Results are presented in Table 6. We observe that HDBSCAN alone typically produces high precision but very low recall. This indicates that the model successfully groups obviously similar samples into small clus-

Table 5. Comparison on ABO-LC. Baseline IC|TC is based on known number of clusters. KMS.: K-Means; HDBS.: HDBSCAN. Max Δ↑ measures improvements of MST Traversal.

	ACC	NMI	ARI
IC|TC [39]	5.5	35.3	5.3
KMS. w/ Known # of Clus.			
GCPM-I (Ours)	**55.7**	92.9	38.4
GCPM-E (Ours)	54.9	92.7	36.2
GCPM-G (Ours)	55.2	93.0	32.3
HDBS. & Our MST Traversal w/o Known # of Clus.			
GCPM-I (Ours)	-	92.4	27.5
+MST Traversal	-	**93.3**	50.7
GCPM-E (Ours)	-	92.3	28.2
+MST Traversal	-	93.1	**51.5**
GCPM-G (Ours)	-	92.1	23.5
+MST Traversal	-	92.9	37.9
Max Δ↑	-	**0.9**↑	**23.3**↑

ters but fails to form larger, more comprehensive clusters. This limitation likely stems from the difficulty of aggregating semantically complex samples using embeddings alone.

In contrast, K-Means produces comparable precision and recall. This balanced performance can be attributed to the prior knowledge of cluster numbers, which forces the algorithm to aggregate more samples into larger groups. After applying MST Traversal, we observe a substantial improvement in Recall with only a modest sacrifice in Precision. It demonstrates that the LLM successfully merges most small clusters into larger, semantically coherent groups according to the guideline. The slight decrease in Precision can be attributed to two factors: (1) guidelines are generated in an unsupervised manner without explicit ground truth, potentially introducing ambiguity, and (2) the LLM may occasionally struggle with interpreting subtle guideline differences. We discuss strategies [97] for improving guideline precision through prompt optimization in Appendix C.4.

Effectiveness of GCPM While the text-only embeddings have already validated GCPM's effectiveness, we conduct additional experiments comparing it against using images alone or using standard captioning prompts from [50] instead of guideline. Table 7 shows that when using GME-Qwen as the embedding model, incorporating concept proxy captions consistently improves performance across all datasets. This improvement demonstrates that GCPM's guideline-aware captioning process effectively surfaces relevant attributes that might otherwise be overshadowed in direct image encoding, validating our design choice to employ concept proxy extraction as an intermediate step.

Runtime Analysis of MST Traversal To eliminate the confounding effects of hardware variations and different LLM implementations, we measure runtime using the number of LLM calls as our primary metric. Results are in Table 8 and Figure 5. Compared to naive pairwise matching where the LLM directly evaluates all cluster pairs, our MST Traversal demonstrates significantly higher efficiency. Notably, MST Traversal is initialized after HDBSCAN, meaning it operates on clusters rather than individual samples, which substantially reduces the number of merge operations required compared with [12, 39]. Furthermore, we cache LLM inference results after each decision. When identical cluster pairs appear in subsequent iterations, we directly utilize cached results, thereby avoiding redundant LLM calls and further improving computational efficiency.

Table 6. Comparison of clustering results based on BCubed Precision (B-Prec.) and Recall (B-Rec.) before and after using MST Traversal upon HDBSCAN. # of clusters includes singletons.

	ImageNet-10			Card-Number		
	# of Clus.	B-Prec.	B-Rec.	# of Clus.	B-Prec.	B-Rec.
K-Means	10	98.6	98.6	13	81.1	81.4
Before	7034	99.7	19.9	4191	98.6	3.1
After	251	93.5	62.3	151	90.6	42.3

Table 7. Comparison of caption quality across datasets using different captioning strategies via K-Means.

	ImageNet-10	Card-Number	Stanford Cars
GME-QWen w/ Image Only	94.7	71.9	61.5
GME-QWen w/ Standard Caption	93.7	73.3	69.2
GME-QWen w/ GCPM Caption	96.7	82.0	86.2

Table 8. Comparison of number of LLM calls in MST Traversal across datasets and their ratios to the number of samples.

# of Samples	ImageNet-10 (13000)	Card-Number (8029)	Stanford Cars (8041)
MST Traversal	11232	6506	10803
LLM / Sample Ratio	0.86	0.81	1.34

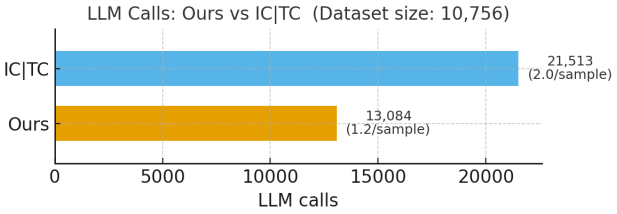

Figure 5. Comparison of LLM calls used for clustering and their ratio to the number of samples on the ABO-LC dataset.

More Ablations We discuss more ablation study, including hyperparameter analysis, GCPM with other clustering algorithms, MST Traversal with proprietary LLMs, and using MST Traversal for existing baselines in Appendix E.

5. Conclusion

We introduce a universal clustering framework that bridges fundamental gaps in diverse clustering scenarios through textual guidelines. Our hybrid approach combines Generative Concept Proxy Modeling for efficient guideline-aware embeddings with MST-based LLM Traversal for selective semantic reasoning, achieving superior performance without task-specific training. Comprehensive experiments across general, multiple, fine-grained, and long-tail clustering tasks demonstrate the superiority of our method.

6. Acknowledgement

This work was partially supported by US National Science Foundation IIS-2412195, CCF-2400785, the Cancer Prevention and Research Institute of Texas (CPRIT) award (RP230363), the National Institutes of Health (NIH) R01 award (1R01AI190103-01) and Microsoft Accelerate Foundation Models Research (2024).

References

[1] Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2, 3

[2] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th international conference on computational linguistics*, 1998. 7

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 3, 4, 5

[4] Yaniv Benny and Lior Wolf. Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. In *European Conference on Computer Vision*, pages 514–530. Springer, 2020. 2, 5, 7, 10

[5] Vladimir Braverman, Jon C. Ergun, Chen Wang, and Samson Zhou. Learning-augmented hierarchical clustering. In *Forty-second International Conference on Machine Learning*, 2025. 2

[6] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013. 2, 4

[7] Arhit Chakrabarti, Yang Ni, Debdeep Pati, and Bani Mallick. Global-local dirichlet processes for clustering grouped data in the presence of group-specific idiosyncratic variables. In *Forty-second International Conference on Machine Learning*, 2025. 2

[8] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 626–635, 1997. 5

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 5, 7, 10

[10] 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. 5, 7, 10

[11] Ziye Chen, Yiqun Duan, Riheng Zhu, Zhenbang Sun, and Mingming Gong. Agent-centric personalized multiple clustering with multi-modal llms. *arXiv preprint arXiv:2503.22241*, 2025. 2, 7

[12] Ziye Chen, Yiqun Duan, Riheng Zhu, Zhenbang Sun, and Mingming Gong. Agent-centric personalized multiple clustering with multi-modal llms, 2025. 8

[13] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5, 4

[14] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 5, 3

[15] Zhiyuan Dang, Cheng Deng, Xu Yang, Kun Wei, and Heng Huang. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13693–13702, 2021. 5, 6, 9

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5, 4

[17] Xin Du and Kumiko Tanaka-Ishii. Information-theoretic generative clustering of documents, 2024. 2, 9

[18] 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*, pages 226–231, 1996. 1

[19] Yiwei Fu, Yuxing Zhang, Chunchun Chen, JianwenMa JianwenMa, Quan Yuan, Rong-Cheng Tu, Xinli Huang, Wei Ye, Xiao Luo, and Minghua Deng. Mark: Multi-agent collaboration with ranking guidance for text-attributed graph clustering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6057–6072, 2025. 2, 4, 8

[20] Xiwen Geng, Suyun Zhao, Yixin Yu, Borui Peng, Pan Du, Hong Chen, Cuiping Li, and Mengdie Wang. Personalized clustering via targeted representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16790–16798, 2025. 2, 3

[21] gpiosenka. Cards image dataset-classification. Kaggle dataset. Accessed: 2025-09-07. 3

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 5, 6, 9

[23] Joris Guérin and Byron Boots. Improving image clustering with multiple pretrained cnn feature extractors. *arXiv preprint arXiv:1807.07760*, 2018. 5, 7, 10

[24] Fusheng Hao, Fengxiang He, Fuxiang Wu, Tichao Wang, Chengqun Song, and Jun Cheng. Task-aware clustering for prompting vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14745–14755, 2025. 2

[25] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979. 1

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 5, 7, 10

[27] Juhua Hu, Qi Qian, Jian Pei, Rong Jin, and Shenghuo Zhu. Finding multiple stable clusterings. *Knowledge and Information Systems*, 51(3):991–1021, 2017. 5, 7, 10

[28] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8849–8858, 2020. 5, 6, 9

[29] Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7509–7524, 2022. 6, 9

[30] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9865–9874, 2019. 1, 5, 6, 7, 9, 10

[31] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-v: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024. 3

[32] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160*, 2024. 3

[33] Hwiyeol Jo, Hyunwoo Lee, Kang Min Yoo, and Taiwoo Park. Zerodl: Zero-shot distribution learning for text clustering via large language models. *arXiv preprint arXiv:2406.13342*, 2024. 2, 4, 8

[34] Andreas Kalavas, Evangelos Kipouridis, and Nithin Varma. Towards better-than-2 approximation for constrained correlation clustering. In *Forty-second International Conference on Machine Learning*, 2025. 2

[35] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011. 3, 5

[36] Yunji Kim and Jung-Woo Ha. Contrastive fine-grained class clustering via generative adversarial networks. *arXiv preprint arXiv:2112.14971*, 2021. 2, 5, 7, 10

[37] 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. 5

[38] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5, 4

[39] Sehyun Kwon, Jaeseung Park, Minkyu Kim, Jaewoong Cho, Ernest K Ryu, and Kangwook Lee. Image clustering conditioned on text criteria. *arXiv preprint arXiv:2310.18297*, 2023. 2, 5, 6, 8, 9

[40] Hongyu Li, Lefei Zhang, and Kehua Su. Dual mutual information constraints for discriminative clustering. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8571–8579, 2023. 6, 9

[41] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 5, 6, 9

[42] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Mixnmatch: Multifactor disentanglement and encoding for conditional image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8039–8048, 2020. 5, 7, 10

[43] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8547–8555, 2021. 5, 6, 9

[44] Yunfan Li, Mouxing Yang, Dezhong Peng, Taihao Li, Jiantao Huang, and Xi Peng. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130 (9):2205–2221, 2022. 5, 6, 9

[45] Yunfan Li, Peng Hu, Dezhong Peng, Jiancheng Lv, Jianping Fan, and Xi Peng. Image clustering with external guidance. *arXiv preprint arXiv:2310.11989*, 2023. 2, 6, 7

[46] Zhixin Li, Yuheng Jia, Junhui Hou, et al. Learning from sample stability for deep clustering. In *Forty-second International Conference on Machine Learning*. 1, 2, 3, 5, 6

[47] I-Fan Lin, Faegheh Hasibi, and Suzan Verberne. Spill: Domain-adaptive intent clustering based on selection and pooling with large language models. *arXiv preprint arXiv:2503.15351*, 2025. 2, 4, 8

[48] Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. Mm-embed: Universal multimodal retrieval with multimodal llms. *arXiv preprint arXiv:2411.02571*, 2024. 3

[49] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 7

[50] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 8

[51] Honglin Liu, Peng Hu, Changqing Zhang, Yunfan Li, and Xi Peng. Interactive deep clustering via value mining. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 2, 5, 6

[52] Jianghan Liu, Ziyu Shang, Wenjun Ke, Peng Wang, Zhizhao Luo, Jiajun Liu, Guozheng Li, and Yining Li. Llm-guided semantic-aware clustering for topic modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18420–18435, 2025. 2, 4, 8

[53] Mingxuan Liu, Zhun Zhong, Jun Li, Gianni Franchi, Subhankar Roy, and Elisa Ricci. Organizing unstructured im-

age collections using natural language. *arXiv preprint arXiv:2410.05217*, 2024. 3

[54] Yikun Liu, Yajie Zhang, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiangchao Yao, Yanfeng Wang, and Weidi Xie. Lamra: Large multimodal model as your advanced retrieval assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4015–4025, 2025. 3

[55] Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as dataset analyst: Subpopulation structure discovery with large language model. In *European Conference on Computer Vision*, pages 235–252. Springer, 2024. 3, 5

[56] J MacQueen. Multivariate observations. In *Proceedings ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability*, pages 281–297, 1967. 1

[57] Ioannis Maniadis Metaxas, Georgios Tzimiropoulos, and Ioannis Patras. Divclust: Controlling diversity in deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3418–3428, 2023. 5, 6, 9

[58] 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*, pages 5174–5181, 2020. 7, 10

[59] Saiyang Na, Yuzhi Guo, Feng Jiang, Hehuan Ma, Jean Gao, and Junzhou Huang. Segment any cell: A sam-based auto-prompting fine-tuning framework for nuclei segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 2025. 2

[60] Tai D Nguyen, Aref Azizpour, and Matthew C Stamm. Forensic self-descriptions are all you need for zero-shot detection, open-set source attribution, and clustering of ai-generated images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3040–3050, 2025. 2

[61] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 3, 5

[62] Chuang Niu, Hongming Shan, and Ge Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022. 5, 6, 9

[63] Kha Pham, Hung Le, Man Ngo, and Truyen Tran. Rapid selection and ordering of in-context demonstrations via prompt embedding clustering. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[64] Han Qi, Haocheng Yin, and Heng Yang. Control-oriented clustering of visual latent representation. *arXiv preprint arXiv:2410.05063*, 2024. 2

[65] Qi Qian. Stable cluster discrimination for deep clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16645–16654, 2023. 1, 5, 6, 7, 9, 10

[66] 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. 5, 7, 10

[67] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5, 7, 10

[68] Yuming Shen, Ziyi Shen, Menghan Wang, Jie Qin, Philip Torr, and Ling Shao. You never cluster alone. *Advances in Neural Information Processing Systems*, 34:27734–27746, 2021. 5, 6, 9

[69] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6490–6499, 2019. 5, 7, 10

[70] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 5, 6, 9

[71] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022. 2, 5, 7

[72] Yaling Tao, Kentaro Takagi, and Kouta Nakata. Clustering-friendly representation learning via instance discrimination and feature decorrelation. *arXiv preprint arXiv:2106.00131*, 2021. 5, 6, 9

[73] Tsung Wei Tsai, Chongxuan Li, and Jun Zhu. Mice: Mixture of contrastive experts for unsupervised image clustering. In *International conference on learning representations*, 2020. 6, 9

[74] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020. 5, 6, 7, 9, 10

[75] Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. Large language models enable few-shot clustering, 2023. 2, 4, 8

[76] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, pages 577–584, 2001. 5, 6, 9

[77] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011. 3, 5

[78] Jiabao Wang, Yang Li, Xiu-Shen Wei, Hang Li, Zhuang Miao, and Rui Zhang. Bridge the gap between supervised and unsupervised learning for fine-grained classification. *Information Sciences*, 649:119653, 2023. 5, 7, 10

[79] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023. 2, 3, 5, 7

[80] Qianqian Wang, Mengping Jiang, Zhengming Ding, and Quanxue Gao. Unified k-means clustering with label-guided

manifold learning. In *Forty-second International Conference on Machine Learning*. 2

[81] Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*, 2023. 7

[82] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963. 4

[83] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 7

[84] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8150–8159, 2019. 5, 6, 9

[85] Ruohong Yang, Peng Hu, Xi Peng, Xiting Liu, and Yunfan Li. Dific: Your diffusion model holds the secret to fine-grained clustering. *arXiv preprint arXiv:2412.18838*, 2024. 5, 7, 10

[86] 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. 5, 7, 10

[87] Jiawei Yao, Enbei Liu, Maham Rashid, and Juhua Hu. Augdmc: Data augmentation guided deep multiple clustering. *Procedia Computer Science*, 222:571–580, 2023. 5, 7, 10

[88] Jiawei Yao, Qi Qian, and Juhua Hu. Customized multiple clustering via multi-modal subspace proxy learning. *Advances in Neural Information Processing Systems*, 37: 82705–82725, 2024. 2, 3, 5, 7, 10

[89] 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. 2, 3, 5, 7, 10

[90] Chunlin Yu, Ye Shi, and Jingya Wang. Contextually affinitive neighborhood refinery for deep clustering. *Advances in Neural Information Processing Systems*, 36:5778–5790, 2023. 5, 6, 9

[91] Lu Zhang, Saiyang Na, Tianming Liu, Dajiang Zhu, and Junzhou Huang. Multimodal deep fusion in hyperbolic space for mild cognitive impairment study. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 674–684. Springer, 2023. 2

[92] Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Gme: Improving universal multimodal retrieval by multimodal llms. *arXiv preprint arXiv:2412.16855*, 2024. 2, 3, 5, 7

[93] Yuwei Zhang, Zihan Wang, and Jingbo Shang. Clusterllm: Large language models as a guide for text clustering, 2023. 2, 4, 8

[94] Huasong Zhong, Jianlong Wu, Chong Chen, Jianqiang Huang, Minghua Deng, Liqiang Nie, Zhouchen Lin, and Xian-Sheng Hua. Graph contrastive clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9224–9233, 2021. 5, 6, 9

[95] Wenliang Zhong, Weizhi An, Feng Jiang, Hehuan Ma, Yuzhi Guo, and Junzhou Huang. Compositional image retrieval via instruction-aware contrastive learning. *arXiv preprint arXiv:2412.05756*, 2024. 3

[96] Wenliang Zhong, Rob Barton, Weizhi An, Feng Jiang, Hehuan Ma, Yuzhi Guo, Abhishek Dan, Shioulin Sam, Karim Bouyarmane, and Junzhou Huang. Zero-shot composed image retrieval via dual-stream instruction-aware distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22221–22231, 2025. 3

[97] Wenliang Zhong, Haiqing Li, Thao M. Dang, Feng Jiang, Hehuan Ma, Yuzhi Guo, Jean Gao, and Junzhou Huang. Learning from guidelines: Structured prompt optimization for expert annotation tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41):35068–35075, 2026. 8

[98] Qifeng Zhou, Wenliang Zhong, Thao M Dang, Hehuan Ma, Saiyang Na, Yuzhi Guo, and Junzhou Huang. Homie: Histopathology omni-modal embedding for pathology composed retrieval. *arXiv preprint arXiv:2502.07221*, 2025. 3

Universal Guideline-Driven Image Clustering via a Hybrid LLM Agent

Supplementary Material

Appendix Table of Contents

A. Computational Complexity

A.1. Proof of Computational Complexity

In this section, we analyze the computational complexity of LLM calls with probabilistic models. We first define notations and assumptions used for analysis. Then, we discuss a simplified scenario for complexity and further extend it to a more realistic scenario.

A.1.1. Problem Setup

Definition 1 *Let us define the mathematical framework for the clustering algorithm:*

$$k_i = \textit{number of clusters at the start of iteration } i \tag{9}$$
$$m_i = \textit{number of merges performed in iteration } i \tag{10}$$
$$c_i = \textit{number of LLM calls in iteration } i = k_i - 1 \tag{11}$$

Constraints:

$$k_1 = N \quad \text{(initial condition)} \tag{12}$$
$$k_{i+1} = k_i - m_i \quad \text{(recurrence relation)} \tag{13}$$
$$m_i \leq \lfloor k_i/2 \rfloor \quad \text{(non-overlapping merge constraint)} \tag{14}$$

Termination: $k_i = 1$ or $m_i = 0$

A.1.2. Probabilistic Model for LLM Decisions

Assumption: Each MST edge is independently approved by the LLM with probability p.

For k clusters with MST having $k-1$ edges, the expected number of actual merges is:

$$\mathbb{E}[m_i] = \min(\alpha p(k_i - 1), \lfloor k_i/2 \rfloor) \tag{15}$$

where $\alpha \leq 1$ accounts for merge conflicts due to non-overlapping constraints.

A.1.3. Case 1: High Merge Rate Regime

When $\alpha p(k_i - 1) \leq k_i/2$, we have:

$$\mathbb{E}[k_{i+1}] = k_i - \alpha p(k_i - 1) \tag{16}$$
$$= k_i(1 - \alpha p) + \alpha p \tag{17}$$

For large k_i, this approximates to:

$$\mathbb{E}[k_{i+1}] \approx k_i(1 - \alpha p) \tag{18}$$

Exponential Decay Analysis:

$$k_i \approx N(1 - \alpha p)^{i-1} \tag{19}$$

Number of iterations until $k_T = 1$:

$$N(1 - \alpha p)^{T-1} = 1 \tag{20}$$
$$T = \frac{\log(N)}{\log(1/(1 - \alpha p))} + 1 = \mathcal{O}(\log N) \tag{21}$$

Total LLM calls:

$$\sum_{i=1}^{T} c_i = \sum_{i=1}^{T} (k_i - 1) \tag{22}$$

$$= \sum_{i=1}^{T} (N(1-\alpha p)^{i-1} - 1) \tag{23}$$

$$= N \sum_{i=1}^{T} (1-\alpha p)^{i-1} - T \tag{24}$$

$$= N \cdot \frac{1-(1-\alpha p)^T}{1-(1-\alpha p)} - T \tag{25}$$

$$= \frac{N(1-(1-\alpha p)^T)}{\alpha p} - \mathcal{O}(\log N) \tag{26}$$

Since $(1-\alpha p)^T \approx 1/N$:

$$\sum_{i=1}^{T} c_i \approx \frac{N(1-1/N)}{\alpha p} - \mathcal{O}(\log N) \tag{27}$$

$$= \frac{N}{\alpha p} - \mathcal{O}(\log N) \tag{28}$$

$$= \mathcal{O}(N) \tag{29}$$

A.1.4. Case 2: Adaptive Merge Rate Model

More realistic assumption: Merge probability decreases as clusters become more refined.

Let $p_i = \frac{c}{\log(k_i)}$ for constant $c > 0$.

Expected merges:

$$\mathbb{E}[m_i] \approx \alpha \cdot \frac{c}{\log(k_i)} \cdot (k_i - 1) = \frac{\alpha c (k_i - 1)}{\log(k_i)} \tag{30}$$

For large k_i:

$$\mathbb{E}[k_{i+1}] \approx k_i - \frac{\alpha c k_i}{\log(k_i)} = k_i \left(1 - \frac{\alpha c}{\log(k_i)}\right) \tag{31}$$

Continuous approximation:

$$\frac{dk}{dt} = -\frac{\alpha c k}{\log(k)} \tag{32}$$

Separating variables:

$$\frac{\log(k)}{k} dk = -\alpha c \, dt \tag{33}$$

Integrating both sides:

$$\int \frac{\log(k)}{k} dk = -\alpha c \int dt \tag{34}$$

Let $u = \log(k)$, then $du = \frac{dk}{k}$:

$$\int u \, du = \frac{(\log(k))^2}{2} \tag{35}$$

Therefore:

$$\frac{(\log(k))^2}{2} = -\alpha c t + C \tag{36}$$

Initial condition: $k(0) = N$, so $C = \frac{(\log(N))^2}{2}$

$$(\log(k))^2 = (\log(N))^2 - 2\alpha c t \tag{37}$$

Termination time when $k = 1$ (i.e., $\log(k) = 0$):

$$0 = (\log(N))^2 - 2\alpha c T \tag{38}$$

$$T = \frac{(\log(N))^2}{2\alpha c} = \mathcal{O}((\log N)^2) \tag{39}$$

A.1.5. Total LLM Calls Calculation

LLM calls per iteration: $k_i - 1$

Total calls:

$$\sum_{i=1}^{T} (k_i - 1) \approx \int_0^T k(t) \, dt - T \tag{40}$$

From our differential equation solution:

$$k(t) = \exp\left(\sqrt{(\log(N))^2 - 2\alpha c t}\right) \tag{41}$$

Since $k(t)$ decreases from N to 1 over $T = \mathcal{O}((\log N)^2)$ iterations, and accounting for the non-uniform decay pattern, the integral evaluates to:

$$\text{Total calls} = \mathcal{O}(N \log N) \tag{42}$$

A.1.6. Main Results

Theorem 1 *Under the adaptive merge rate model $p_i = \frac{c}{\log(k_i)}$, the expected number of LLM calls for the hierarchical clustering algorithm is $\mathcal{O}(N \log N)$.*

A.2. In-Depth Runtime Complexity Analysis

In this section, we validate the assumption of the adaptive merge rate model from Section A.1. We then discuss how caching rejected pairs accelerates the MST Traversal.

We track merging patterns during the traversal process. As mentioned in Section 3.3, we build a traversal path based on the edge weights of the MST in ascending order. Each edge weight measures how close a pair of clusters is. The LLM makes a decision on whether they should be merged following this order. We track all merging decisions for MST Traversal across different iterations on the Card dataset conditioned on number. Results are in Figure 6.

We observe that merging occurs more frequently among earlier pairs and gradually decreases as the traversal proceeds. This validates the effectiveness of our hybrid design: guideline-driven embeddings provide a reliable distance matrix such that closer pairs have higher priority to

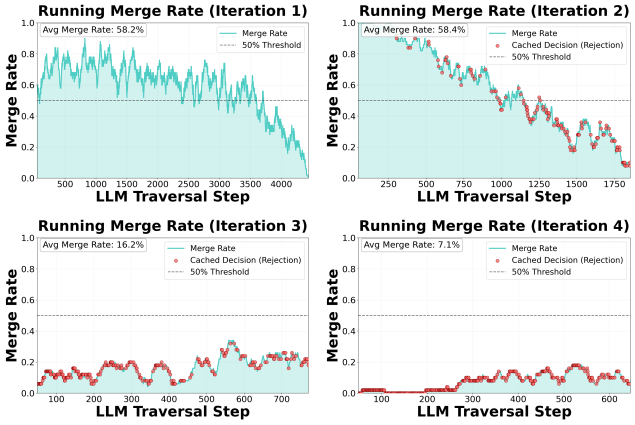

Figure 6. Merging rates over four iterations. The running merging rate is calculated by moving average over 50 consecutive steps. Red dots represents cached rejected decision that the LLM can safely skip. The merging rates decreases as iteration proceeds. The cached decisions significantly reduce the number of LLM calls.

be examined by the LLM. This also validates the adaptive merge rate model assumption that the LLM merges clusters more frequently at the beginning, and the merge probabilities decrease as clusters become more refined. Notably, if the LLM rejects a pair, they will exist in the next iteration and may be selected again by the MST. As merging becomes less frequent in later iterations, more and more rejected pairs will be repeatedly selected. Therefore, we cache decisions for all rejected pairs and skip them if they occur subsequently. This cache mechanism reduces the number of LLM calls and thus improves efficiency.

B. Dataset Details

B.1. ABO Data Processing

We downloaded 398,212 original images and metadata from the ABO [14] dataset official website[1]. Each item contains an image and multiple textual attributes as shown in Figure 7. We select several key attributes reflecting how items are typically categorized in modern e-commerce systems: ['brand', 'color', 'item_id', 'model_name', 'product_type', 'style']. Subsequently, we filtered out items without these attributes, resulting in 10,756 items. We then apply the group_by function from the pandas package[2] to these items based on all key attributes. Each item is ultimately represented by its image and item name, which are not provided during the grouping process. The item name and image provide complementary textual (e.g., brand, model name) and visual (e.g., color, style) information to infer the key attributes of a given item. Note that unlike existing public datasets that are originally designed for classification,

[1]https://amazon-berkeley-objects.s3.amazonaws.com/index.html
[2]https://pandas.pydata.org/docs/index.html

Figure 7. Raw sample examples from the original ABO dataset.

the ABO-LC dataset does not have an explicit definition for each category, reflecting real-world scenarios and making the clustering process even more challenging.

B.2. Existing Datasets and Metrics

Metrics In general, we use Accuracy (ACC), Normalized Mutual Information (NMI), Rand Index (RI), Adjusted Rand Index (ARI), Bcubed Precision (B-Prec.) and Recall (B-Rec) in the paper. Given two cluster assignments $Y = \{y_1, \ldots, y_N\}$ and $C = \{c_1, \ldots, c_N\}$, the Normalized Mutual Information (NMI) is defined as

$$\mathrm{NMI}(Y, C) = \frac{2\, I(Y; C)}{H(Y) + H(C)}, \tag{43}$$

where $I(Y; C)$ denotes the mutual information between Y and C, and

$$I(Y; C) = \sum_{i=1}^{K} \sum_{j=1}^{L} p_{ij} \log \frac{p_{ij}}{p_i\, p_j}, \tag{44}$$

with p_i, p_j the marginal probabilities and p_{ij} the joint probability of $Y = i$ and $C = j$. The entropies are defined as

$$H(Y) = -\sum_{i=1}^{K} p_i \log p_i, \qquad H(C) = -\sum_{j=1}^{L} p_j \log p_j. \tag{45}$$

Given the ground-truth labels $Y = \{y_1, \ldots, y_N\}$ and the predicted cluster assignments $C = \{c_1, \ldots, c_N\}$, define the contingency matrix

$$n_{ij} = \big|\{\, k \mid y_k = i,\ c_k = j \,\}\big|.$$

To compare clusters and true labels, we seek a permutation π of the cluster indices that maximizes the matching:

$$\max_{\pi} \sum_{k=1}^{N} \mathbf{1}[\, y_k = \pi(c_k) \,].$$

The permutation π is obtained by applying the Hungarian (Munkres) algorithm to the cost matrix derived from $-n_{ij}$. After alignment, define the predicted labels $\hat{y}_k = \pi(c_k)$. The clustering accuracy is then

$$\mathrm{ACC}(Y, C) = \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}[\, y_k = \hat{y}_k \,]. \tag{46}$$

The Rand Index (RI) evaluates the agreement of pairwise decisions. Let a be the number of pairs (i, j) for which $y_i = y_j$ and $c_i = c_j$, and let b be the number of pairs (i, j) for which $y_i \neq y_j$ and $c_i \neq c_j$. The Rand Index is defined as

$$\mathrm{RI}(Y, C) = \frac{a + b}{\binom{N}{2}}. \tag{47}$$

Let n_{ij} denote the number of samples that belong to ground-truth class i and predicted cluster j. Define the row sums $a_i = \sum_j n_{ij}$ and column sums $b_j = \sum_i n_{ij}$. The Adjusted Rand Index (ARI) is defined as

$$\mathrm{ARI}(Y, C) = \frac{\displaystyle\sum_i \sum_j \binom{n_{ij}}{2} - \frac{\left(\sum_i \binom{a_i}{2}\right)\left(\sum_j \binom{b_j}{2}\right)}{\binom{N}{2}}}{\displaystyle\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\left(\sum_i \binom{a_i}{2}\right)\left(\sum_j \binom{b_j}{2}\right)}{\binom{N}{2}}}. \tag{48}$$

For each sample i, define the pairwise correctness indicator

$$\mathrm{correct}(i, j) = \begin{cases} 1, & \text{if } (y_i = y_j) \text{ and } (c_i = c_j), \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathrm{pred}(i, j) = 1$ if $c_i = c_j$ and $\mathrm{gt}(i, j) = 1$ if $y_i = y_j$. The Bcubed Precision for sample i is

$$P_i = \frac{\sum_{j \neq i} \mathrm{correct}(i, j)}{\sum_{j \neq i} \mathrm{pred}(i, j)}. \tag{49}$$

The Bcubed Recall for sample i is

$$R_i = \frac{\sum_{j \neq i} \mathrm{correct}(i, j)}{\sum_{j \neq i} \mathrm{gt}(i, j)}. \tag{50}$$

The final Bcubed Precision (B-Prec.) and Recall (B-Rec.) are the averages:

$$\text{B-Prec} = \frac{1}{N} \sum_{i=1}^{N} P_i, \qquad \text{B-Rec} = \frac{1}{N} \sum_{i=1}^{N} R_i. \tag{51}$$

Datasets We use ten public datasets for general clustering, multiple clustering, and fine-grained clustering. Except Fruit, Card, CUB Birds, and Stanford Cars, other datasets do not provide explicit guideline from their source. We describe all datasets and available guidelines in this section and introduce our prompts to derive guidelines for datasets that do not have ones.

STL-10 [13]. STL-10 is an image classification dataset derived from ImageNet, containing 13,000 images across 10 common object categories (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck). The dataset features 96×96 color images and was designed to evaluate unsupervised feature learning and deep learning methods in scenarios with limited labeled data. The dataset represents general-purpose object categorization with relatively balanced class distributions, making it a standard benchmark for evaluating general clustering methods. Unlike fine-grained datasets, STL-10 focuses on coarse-level semantic distinctions between different object types. Explicit guideline is not available in this dataset.

CIFAR-10 [38]. CIFAR-10 is a widely-used image classification dataset consisting of 60,000 32×32 color images across 10 common object categories (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). The dataset is evenly distributed with 6,000 images per class, split into 50,000 training images and 10,000 test images. The low-resolution images and balanced class distribution make CIFAR-10 a standard benchmark for evaluating clustering algorithms on general object categorization tasks. Despite its relatively simple visual characteristics compared to higher-resolution datasets, CIFAR-10 remains challenging for clustering methods due to high intra-class variation and inter-class similarity among certain categories. Explicit guideline is not available in this dataset.

ImageNet-10 [16]. ImageNet-10 is a subset of the large-scale ImageNet dataset, consisting of 13,000 high-resolution images from 10 selected classes. Following existing works in clustering evaluation, we use the specific ImageNet synsets: n02056570 (dog), n02085936 (Maltese dog), n02128757 (lynx), n02690373 (airliner), n02692877 (aircraft carrier), n03095699 (container ship), n04254680 (soccer ball), n04285008 (sports car), n04467665 (trailer truck), and n07747607 (orange). These classes include both general-level categories (e.g., airliner, sports car) and fine-grained distinctions within similar object types (e.g., different dog breeds, various vehicles), creating a challenging clustering scenario that requires semantic understand-

ing at multiple levels of granularity. Explicit guideline is not available in this dataset.

Fruit [27]. The Fruit dataset is a multiple clustering dataset. We use all criteria used for multiple clustering (species and color) to construct the guideline.

Card [87]. The Card dataset is a multiple clustering dataset. We use all criteria used for multiple clustering (number and suits) to construct the guideline.

CIFAR10-MC [88]. The CIFAR10-MC dataset is a multiple clustering version of the CIFAR-10 dataset. We use all criteria used for multiple clustering (type and environment) to construct the guideline.

CUB Birds [77]. The Caltech-UCSD Birds-200-2011 (CUB) dataset is a fine-grained visual categorization dataset containing 11,788 images of 200 bird species. The dataset provides rich attribute annotations including visual characteristics such as bill shape, wing color, tail shape, and other morphological features, which we leverage to construct our clustering guideline. Unlike general clustering datasets that focus on coarse-level distinctions between different object types, CUB requires fine-grained discrimination among visually similar bird species based on subtle variations in appearance. This makes CUB particularly challenging for clustering methods, as intra-class variation (e.g., different poses, lighting conditions) can be comparable to inter-class differences between similar species. The availability of detailed attribute annotations makes CUB well-suited for evaluating guideline-driven clustering approaches.

Stanford Cars [37]. The Stanford Cars dataset is a fine-grained visual categorization dataset containing 16,185 images of 196 car classes. Each car class is defined by the combination of Make, Model, and Year (e.g., "2012 Tesla Model S", "2007 BMW M3"), which we use to construct our clustering guideline. The dataset presents unique challenges for fine-grained clustering as cars within the same make and model but different years may have subtle design variations, while cars from different manufacturers can share similar body styles and visual characteristics. Images contain cars from various viewpoints and in different settings, adding to the difficulty of visual discrimination.

Stanford Dogs [35]. The Stanford Dogs dataset is a fine-grained visual categorization dataset containing 20,580 images across 120 dog breeds from around the world. The dataset encompasses diverse breeds ranging from visually distinct categories (e.g., Chihuahua vs. Saint Bernard) to highly similar breeds that differ primarily in subtle characteristics such as coat texture, ear shape, or body proportions (e.g., different types of terriers or retrievers). Images exhibit significant variation in pose, scale, background clutter, and lighting conditions, making visual feature extraction challenging. The fine-grained nature of dog breed classification requires careful attention to discriminative details that distinguish closely related breeds, making Stanford Dogs particularly demanding for clustering methods. Explicit guideline is not available in this dataset.

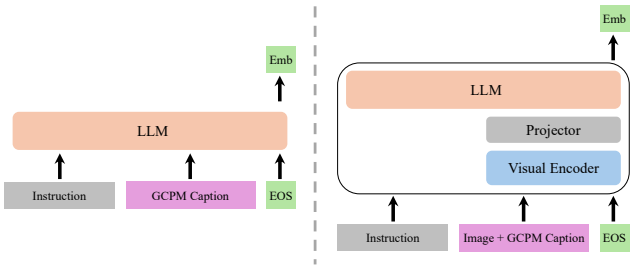

Figure 8. **Left**: the forwarding process of E5-Mistral. The input contains the instruction, the GCPM caption, and an EOS token, which are concatenated and fed to the LLM. The output EOS token embedding is used for representation. **Right**: for GME-QWen, the input includes the original image, which are processed to visual tokens by a visual encoder and a projector.

Oxford Flowers [61]. The Oxford Flowers dataset is a fine-grained visual categorization dataset containing 8,189 images of 102 flower species commonly found in the United Kingdom. The dataset presents significant challenges for fine-grained clustering due to the high visual similarity among different flower species, where discrimination often relies on subtle differences in petal shape, color patterns, stamen structure, and overall flower morphology. Images exhibit large variations in scale, pose, and illumination, with flowers captured from different angles and in various lighting conditions. Additionally, some species may share similar colors or shapes, requiring careful attention to multiple attributes simultaneously. Explicit guideline is not available in this dataset.

C. Guideline Processing

C.1. Guideline Generation

Existing works [55] have explored subpopulation structure discovery within a dataset via LLMs, which can be directly applied to discover key attributes of a dataset. While these methods can be used without any prior knowledge about a dataset, the involvement of LLMs can still be expensive. We consider a more realistic scenario where users typically have a fundamental understanding of a dataset, especially regarding what clustering objective is desired for the dataset. For example, on the Stanford Dogs dataset, users only need to provide the basic clustering objective *dog breeds*. Based on this objective, we introduce a series of heuristic prompts as shown in Figure 9. The guideline generation process consists of three steps. First, we query the LLM about the major differences when considering possible categories according to the clustering objective. Second, we ask the LLM to derive key visual attributes to consider for clustering. Finally, we formalize a guideline prompt

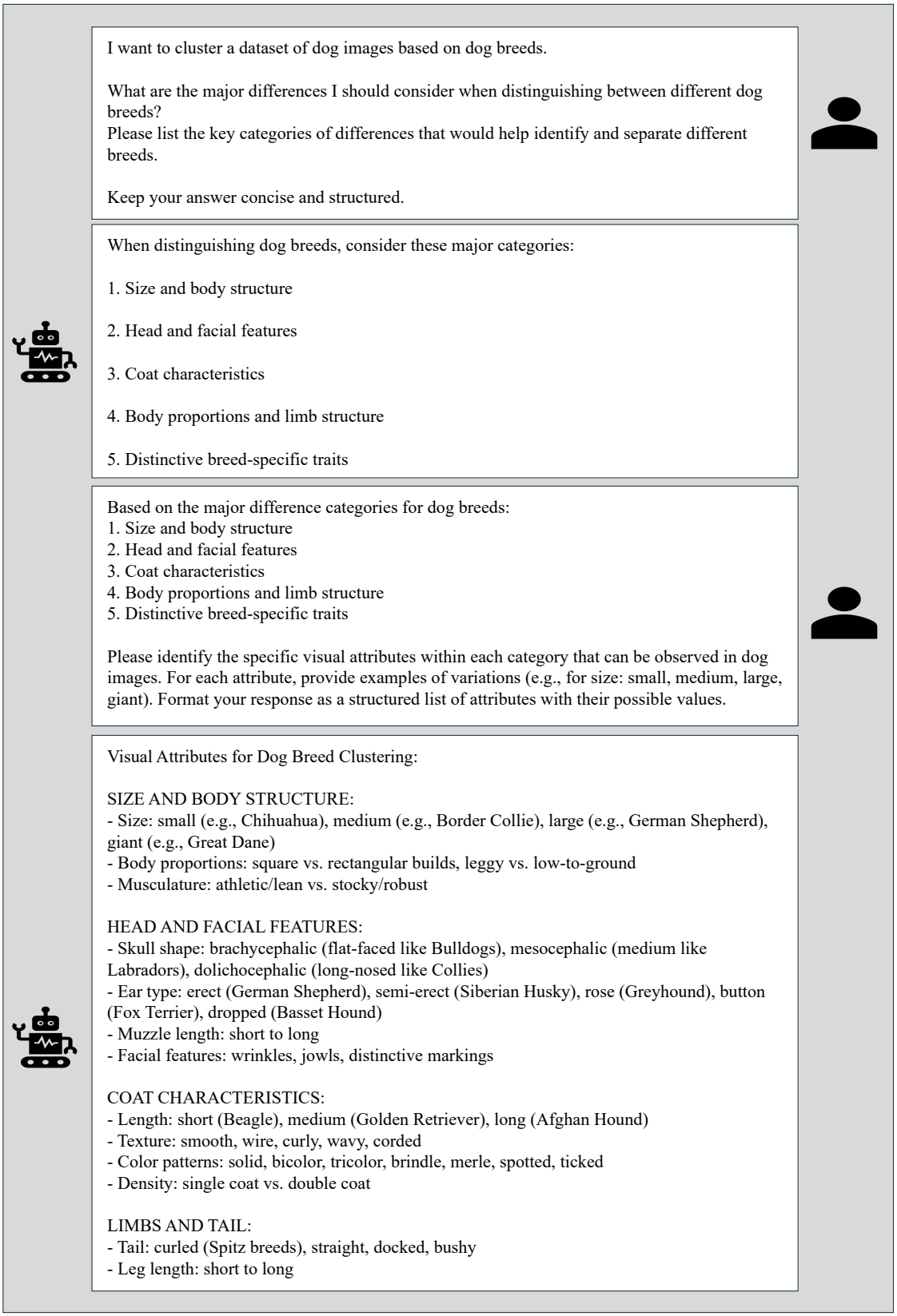

Figure 9. Guideline Generation Process for the Stanford Dogs dataset

Table 9. Effectiveness of Prompt Optimization using Prompt Agent on the Oxford Flowers dataset.

	NMI	B-Prec.	B-Rec.
Before Optim.	88.6	90.1	67.9
After Optim.	90.2	95.7	65.9

considering all key visual features as attributes. The three-step generation acts like Chain-of-Thought prompting [83], leveraging the internal knowledge of LLMs to derive a feasible guideline for clustering.

C.2. Detailed Guideline

As mentioned in Section 3.1, we consider a guideline G that contains multiple attributes $A = \{a_1, a_2, \cdots, a_k\} \subseteq G$. Table 15 shows all attributes considered for each dataset. Based on these attributes, three types of guideline prompts are derived: the GCPM captioning prompts (shown in Tables 16, 17, 18, 19) generate GCPM captions based on all attributes; the instruction-aware embedding prompts (shown in Table 20) encode the GCPM captions based on one attribute or a global definition of all attributes; and the MST Traversal prompts (some shown in Table 21 due to space constraints) consider one attribute or a global definition of all attributes for merging a pair of clusters.

C.3. Guideline and Image Forwarding Process

We leverage three embedders: Instructor-large [71], E5-Mistral [79], and GME-QWen [92] for our main experiments. In general, they can be classified into two categories: text-only instruction-aware embedders and multi-modal instruction-aware embedders, as shown in Figure 8. For Instructor-large and E5-Mistral, the input consists of the GCPM caption and the instruction-aware embedding prompt. For GME-QWen, the input consists of the original image, the GCPM caption, and the instruction-aware embedding prompt.

C.4. Improving Guideline Accuracy

In this section, we discuss how to improve guideline reliability. From the Bcubed results (Table 6), we observe that the Bcubed Recall increases while the Precision drops. We argue that this can be attributed to two factors: (1) guidelines are generated in an unsupervised manner without explicit ground truth, potentially introducing ambiguity, and (2) the LLM may occasionally struggle with interpreting subtle guideline differences. Therefore, our improvement goal is to maintain Bcubed Precision while improving Recall. To improve guideline quality, we assume a few-shot supervised setting and use prompt optimization to update the prompt.

We use the Oxford Flowers dataset for experiments. Specifically, we collect 20 false positive pairs, where two clusters that should not be merged are mistakenly merged together. We then use PromptAgent [81] with DeepSeek [49] as the optimizer to update the prompt. Experimental results are shown in Table 9. From the experiments, we observe that Bcubed Precision increases together with Recall. This experiment demonstrates that the guideline can be improved when users are not satisfied with its quality, using only very few training samples.

D. Comparison with Existing Methods

D.1. Guideline-aware and Text-guided Clustering

Multiple-clustering with user-specified aspects. Recent works such as Multi-MaP [89] and Multi-Sub [88] focus on multiple clustering, where a dataset can be partitioned from different perspectives, for example "by color" versus "by species" for fruits. They typically construct proxy spaces or subspaces aligned with user-specified aspects, often using CLIP plus a small number of aspect words as anchors, and then train an end-to-end deep clustering model to produce aspect-specific partitions. While they support user-specified criteria, they (1) require dataset-specific training or fine-tuning, (2) mainly target balanced and relatively small-scale benchmarks such as Fruit, Cards, and CIFAR10-MC, and (3) still assume a small number of discrete aspect choices rather than arbitrary composite guidelines.

Externally guided general image clustering. Image Clustering with External Guidance (TAC) [45] introduces the idea of externally guided clustering by injecting WordNet concepts into the representation space. TAC first builds visual semantic centers, retrieves relevant noun concepts via CLIP, and then jointly distills image and text cluster heads with a mutual-distillation objective. This improves general image clustering without labels, but the "guidance" is limited to a fixed concept vocabulary and cannot express complex multi-attribute business rules or fine-grained operational guidelines. The method is also tied to a specific training pipeline and does not directly extend to multiple clustering, fine-grained clustering, or long-tail settings under a single framework.

Agent-centric personalized multiple clustering. Agent-Centric Personalized Multiple Clustering with MLLMs [11] moves closer to agentic guideline following. It uses an MLLM to build an interest-biased similarity graph and then conducts agent-based graph traversal to grow clusters under a specified user interest. The approach enables more explicit user control and uses LLM-based agents to accept or reject membership decisions. However, it still focuses on multiple-clustering style scenarios and relies on training an agent pipeline with GPT-based supervision. Moreover, it assumes a relatively controlled number of clusters and does

Table 10. Comparison of key guideline-aware and LLM/VLM-driven clustering methods.

Paper	Task	Guideline Mechanism	LLM/VLM Role	Limitation/Difference
Agentic Graph Traversal (Arxiv 2025)	Personalized multiple clustering	User preferences bias graph & traversal	MLLM agents expand/merge clusters	High LLM cost; scaling issues
Multi-Sub (NeurIPS 2024)	Multi-view clustering with user criteria	GPT-4 proxy words (e.g., *red*, *green*)	Aligns images to subspaces	Oversimplifies complex guidelines
IC|TC (ICLR 2024)	Image clustering under user text criteria	User text refined into cluster names	GPT-4 generates cluster names & assignments	Relies on text descriptions; limited fine-grained control
LiSA (ACL 2025)	Topic modeling	LLM generates candidate topics	Guides document–topic alignment	Text-only; not visual
ClusterLLM (EMNLP 2023)	Text clustering	Triplet/pairwise queries under instruction	LLM judges similarity & granularity	Expensive; not image-focused
TAC (ICML 2024)	Image clustering + external ontology	WordNet nouns as guidance	CLIP + noun anchors	Requires fixed ontology; not user-adaptive
TGAICC (Arxiv 2024)	Multiple valid clusterings	Different prompts → consensus	LLM generates criteria & consensus	Limited efficiency; unstable across prompts
ITGC (Arxiv 2025)	Text-guided image clustering	Extracts and cluster in concept space	LLM produces concept-aligned embedding space	High-quality text needed; weak for fine-grained/hierarchy
X-Cluster/TeDeSC (Arxiv 2024)	Open-ended clustering	Auto-discovers natural-language criteria	VLM + LLM propose grouping dimensions	Lack of control; exploratory, not precise
TGIC (EACL 2024)	Caption-driven clustering	Captions serve as criteria	Text model augments embeddings	Early/simple; weak guideline following
CLIP-Cluster (ICCV 2023)	Attribute-driven face clustering	Hallucinated CLIP attributes	CLIP acts as proxy feature extractor	Not LLM; limited to predefined attributes
Our Clustering Agent	General, multi-granularity clustering	Guideline → GCPM → MST Traversal	LLM extracts guideline signals; minimal LLM use	Training-free, efficient, complex guideline following

not target highly long-tailed distributions or automatic cluster discovery in general.

How our method differs. Our guideline-driven clustering agent departs from these approaches in several key aspects. First, instead of training a dataset-specific model for each task or aspect, we adopt a *training-free* two-stage hybrid design. Generative Concept Proxy Modeling (GCPM) uses MLLMs plus instruction-aware embedders to produce guideline-aware embeddings from concept proxy captions, without any task-specific fine-tuning. This allows us to reuse the same pipeline across general clustering, multiple clustering, fine-grained clustering, and long-tail e-commerce clustering under a single universal framework.

Second, prior guideline-aware methods generally handle a single criterion per run (for example, "by color" or "by species") or a small enumerated set of views. In contrast, our formulation treats a guideline as a composition of attributes and explicitly models composite criteria such as "group by brand and intended activity" or domain-specific SOP-style rules. The concept proxy step disentangles these attributes, and the subsequent instruction-aware encoding enables flexible recombination of attributes within the embedding space.

Third, while methods like TAC and multiple-clustering approaches assume either known cluster numbers or balanced distributions, we explicitly tackle scenarios where the number of clusters is unknown and the distribution is highly long-tailed. We use HDBSCAN to obtain conservative small clusters and then apply MST-based LLM Traversal to semantically merge clusters according to the guideline. This hybrid strategy allows us to handle automatic cluster discovery and large numbers of tiny clusters that conventional multiple-clustering methods and externally guided approaches do not address. Overall, existing guideline-aware and multiple-clustering methods either require substantial training, support only simple criteria, or target a narrow set of scenarios, whereas our universal guideline-driven agent is training-free, supports composite guidelines, and covers GC, MC, FC, and long-tail clustering within one coherent framework.

D.2. LLM-based Clustering

A second line of related work uses large language models more directly in the clustering process, either as supervisors, similarity oracles, or agentic controllers. These methods demonstrate the semantic power of LLMs but typically do not address complex guideline-driven image clustering in a universal way.

LLMs as clustering supervisors or guides. ClusterLLM [93] treats an API-only LLM as a teacher that guides unsupervised text clustering. It uses triplet queries to refine a small local embedder to better align with an instruction such as "cluster by topic" or "cluster by sentiment", and then queries the LLM again on pairwise decisions to select an appropriate level of a hierarchical clustering tree. Large Language Models Enable Few-Shot Clustering [75] similarly leverages LLMs to produce cluster descriptions and document assignments from few labeled examples, largely in text domains. Both methods showcase how LLMs can steer clustering criteria and granularity, but they operate mainly on textual data, rely on repeated LLM interactions over a large set of examples, and do not handle rich visual guidelines or the diverse image clustering regimes we consider. LLM-Guided Semantic-Aware Clustering for Topic Modeling (LiSA) [52] further integrates LLMs into topic modeling by generating topic words and descriptions per document, clustering both documents and topics, and then using an LLM in a conflict resolution module. Its focus is on aligning topic and document for text corpora, not on supporting user-specified guidelines or visual semantics.

LLM-based clustering in graphs and intents. Recent works such as MARK [19], SPILL [47], and ZeroDL [33] extend LLM-based clustering to text-attributed graphs, intent discovery, and zero-shot distribution learning. MARK uses multi-agent collaboration with ranking guidance on text-attributed graphs, SPILL focuses on domain-adaptive intent clustering with LLM-based selection and pooling, and ZeroDL formulates zero-shot distribution learning for text clustering via LLMs. These methods again show that LLMs can provide powerful semantic guidance but are designed for textual or graph data, and they typically assume

relatively well-behaved cluster structures rather than severe long-tail distributions. Information-Theoretic Generative Clustering of Documents (ITGC) [17] represents another direction that combines information-theoretic objectives with LLM components for document clustering. ITGC improves text clustering quality by modeling generative relationships and mutual information, but it remains confined to document data and does not address image semantics, guideline composition, or agentic control across multiple clustering tasks.

LLM-based image clustering. On the vision side, Image Clustering Conditioned on Text Criteria (IC|TC) [39] is the most relevant LLM-based image clustering work. It uses a VLM to produce criterion-specific image descriptions, then asks an LLM to propose cluster names and assign each description to one of K clusters given a user-provided text criterion. This enables training-free text-guided clustering on images but has several important limitations. First, it still assumes a single concrete criterion per run, such as "cluster by action" or "cluster by location", and does not support multi-attribute guidelines or complex constraints. Second, it assumes the number of clusters K is known or searched over in a simple fashion, and it is not tailored to long-tail settings with many tiny clusters. Third, it performs LLM-based assignment over all samples, which leads to high LLM cost when scaling to large datasets or multiple successive clustering tasks.

How our method differs. Our approach can be seen as a hybrid between LLM-guided clustering and deep clustering, but with a focus on *universal guideline-driven image clustering*. Compared to ClusterLLM, LiSA, and other text-oriented LLM clustering methods, we operate directly on vision tasks (GC, MC, FC, and LC) and treat guidelines as composable multi-attribute descriptions rather than single textual perspectives. We also avoid repeated fine-tuning of embedders or training of new prediction networks, remaining fully training-free.

Relative to IC|TC and other LLM-based image clustering approaches, our framework introduces two key advances. First, GCPM disentangles guideline attributes through concept proxy captions and instruction-aware embeddings, which allows us to reuse the same embeddings across different criteria and composite guidelines, and to plug them into both standard clustering algorithms and the MST-based traversal stage. Second, MST-based LLM Traversal uses HDBSCAN to generate conservative small clusters and then performs LLM queries only on a carefully selected subset of cluster pairs along a minimum spanning tree. This design reduces the number of LLM calls from quadratic in the number of clusters to approximately $O(M \log M)$ in expectation, while still enabling guideline-aware semantic merging. In contrast, IC|TC and many LLM-based clustering methods invoke the LLM on a large fraction of examples or cluster pairs, leading to higher computational cost.

Finally, most existing LLM-based clustering works assume either known cluster numbers or relatively balanced clusters. By combining guideline-aware embeddings with HDBSCAN and LLM-based merging, our method explicitly targets settings where the cluster number is unknown and clusters are extremely imbalanced, such as the long-tail ABO e-commerce dataset with thousands of tiny clusters. This capability is not addressed by prior LLM-based clustering or agentic frameworks and is crucial for deploying guideline-driven image clustering in realistic scenarios.

D.3. General Clustering Baselines

We compare our guideline-driven agent with a diverse set of state-of-the-art general deep clustering methods on CIFAR-10, STL-10, and ImageNet-10, as summarized in Table 2.[3] All baselines assume that the ground-truth number of clusters is known and train task-specific models on each dataset, while our method remains training-free and can operate both with and without this prior.

Classical and constrained deep clustering. Cop-KMeans (Cop-KMS.) [76] extends K-Means with pairwise constraints. IIC [30] and DCCM [84] are early deep clustering methods that optimize instance consistency or mutual information between augmented views. These methods learn dataset-specific encoders and clustering heads, usually optimized per dataset and metric.

Self-supervised representation learning with clustering. A large group of baselines follows the recent paradigm of first learning general-purpose representations via self-supervised objectives, then performing K-Means in the embedding space. BYOL [22], MiCE [73], PCL[41], PICA[28], SCAN [74], FixMatch [70], CC [43], GCC [94], IDFD [72], NMM [15], TCC [68], ProPos [29], SPICE [62], TCL [44], CoNR [90], DMICC [40], SeCu [65], and DivClust [57] all fall into this category. They typically rely on heavy pretraining and then optimize clustering quality through contrastive, mutual-information, or distribution-regularization losses. Many of them further introduce specialized training objectives for cluster separation or diversity, which improves performance but ties the method to the particular dataset and task.

For all methods above, we either adopt reported numbers from their original papers or follow their publicly released implementations and recommended hyperparameters under the standard GC evaluation protocol (ACC, NMI, and ARI) on CIFAR-10, STL-10, and ImageNet-10. Our method shares the same evaluation protocol but does not perform any fine-tuning on the target datasets.

[3] See the main text for dataset definitions.

D.4. Multiple Clustering Baselines

Multiple clustering (MC) benchmarks require algorithms to discover several valid partitions of the same dataset according to different criteria (e.g., Fruit by color or species, Cards by number or suit, CIFAR10-MC by object type or environment). Following prior work, we evaluate on Fruit, Cards, and CIFAR10-MC using NMI and RI for each criterion and also report the averaged score per dataset in Table 3. All baselines assume the true number of clusters is known and train models specifically tailored to these MC scenarios, whereas our method uses a guideline-driven agent that can be instantiated with any textual clustering specification.

Early multiple-clustering methods. MSC [27] and MCV [23] are early methods for discovering alternative partitions through feature reweighting or multi-view embeddings. ENRC [58] and iMClusts [66] refine this line by learning disentangled embeddings that encourage different clustering heads to focus on complementary aspects of the data. These methods are effective on small-scale MC datasets but still rely on explicit training and a fixed number of predefined clusters.

Deep multiple clustering with augmented or diffusion-based representations. AugDMC [87] and DDMC [86] introduce stronger deep backbones and augmentation- or diffusion-based regularization to improve the robustness of multiple clusterings. They optimize several heads jointly to encourage diversity between perspectives while maintaining high consistency with each ground-truth partition. The models are trained per dataset and per set of criteria and thus cannot be easily reused for new guidelines without retraining.

State-of-the-art proxy-based multiple clustering. Multi-MaP [89] and Multi-Sub [88] represent the current state of the art for MC. They introduce proxy-based modules that map images to concept anchors or subspaces aligned with user-specified criteria, then perform clustering in each learned representation. While these methods can flexibly switch between a small set of predefined clustering dimensions, they still require extensive training on labeled or pseudo-labeled data for each dataset and assume that all criteria are known ahead of time. In contrast, our guideline-driven agent directly takes natural language descriptions of the desired criterion (for example, "group by card number" or "group by card suit") and uses the same GCPM embeddings and MST-based LLM Traversal across all MC tasks without retraining. This enables us to support new or composite guidelines beyond the few perspectives that existing MC baselines are designed for, while still achieving competitive or superior performance on the standard Fruit, Cards, and CIFAR10-MC benchmarks.

D.5. Fine-grained Clustering Baselines

Fine-grained clustering (FC) benchmarks focus on distinguishing visually similar subcategories, such as bird species, dog breeds, car models, or flower species. We evaluate on CUB Birds, Stanford Dogs, Stanford Cars, and Oxford Flowers using ACC and NMI, following prior work. The compared baselines in Table 4 cover both representation-learning approaches and generative clustering methods. Similar to GC and MC, all FC baselines rely on dataset-specific training and, except for UFCL, assume the number of clusters is known, while our method remains training-free and can operate without that prior.

Representation-learning baselines. IIC [30] extends invariant information maximization to FC settings; SimCLR [9] and MoCo [26] are contrastive self-supervised representation learners whose embeddings are clustered to form categories. SCAN [74] and SeCu [65] further adapt self-supervised models to clustering by adding clustering-specific heads and objectives. These approaches show that strong representations help FC, but they still treat clustering as a purely geometric problem in the embedding space and cannot incorporate textual guidelines.

Generative and GAN-based fine-grained clustering. A second group of baselines focuses on generative modeling for fine-grained discovery. InfoGAN [10], FineGAN [69], MixNMatch [42], OneGAN [4], SD [67], and C3-GAN [36] learn generative models whose latent codes are encouraged to align with semantic factors, then interpret these codes as cluster assignments. DiFiC [85] leverages diffusion models for fine-grained semantics. These methods achieve strong performance when trained on each dataset but require substantial computation and bespoke architectures, and they do not provide a direct mechanism to encode user-defined guidelines into the clustering process.

Unified fine-grained clustering. UFCL [78] is a recent unified fine-grained clustering framework that aims to handle several datasets within a single model. It still requires training on fine-grained data and assumes cluster numbers are known in our evaluation. Our method differs in two important aspects: (i) we incorporate textual guidelines to define the target granularity or attributes of interest, and (ii) we remain entirely training-free, combining guideline-aware GCPM embeddings with MST-based LLM Traversal to refine clusters. Despite using conservative prompts for FC to avoid over-merging visually similar categories, our model still achieves state-of-the-art or highly competitive results on CUB Birds, Stanford Dogs, Stanford Cars, and Oxford Flowers, demonstrating that guideline-driven clustering scales even to demanding fine-grained scenarios.

Table 11. Ablation on hyperparameter settings. We report NMI scores for GME-QWen on Card-Number.

Hyperparameter Settings	Results
Top-K (K=3)	70.7
Top-K (K=5)	72.1
Top-K (K=7)	72.4
HDBS. min_cluster_size=2	50.3
HDBS. min_cluster_size=3	55.4
HDBS. min_cluster_size=4	56.7

Table 12. Clustering NMI results for GCPM embeddings using GME-QWen on Card-Number. SC: Spectral Clustering; AC: Agglomerative Clustering.

	Card-Number
AC	84.7
SC	81.1
DBSCAN	44.4

E. More Ablations

E.1. Hyperparameter Analysis

In main experiments, we use min_cluster_size=2 for HDBSCAN and K=5 for Top-K samples representing a cluster in the MST Traversal. We conducted hyperparameter analysis on Card-Number for these parameters as shown in Table 11. In experiments, we observe that the K=5 achieves a relative good balance for performance and efficiency. We found that the performance varies for changes in min_cluster_size. Though large values seem to achieve better performance, we maintain min_cluster_size=2 because of the philosophy that at least two clusters are enough to form a cluster other than singletons. It can be applied to all dataset we tested.

E.2. GCPM with Other Clustering Algorithms

We tested GCPM embeddings using GME-QWen with other clustering algorithms, including Agglomerative Clustering (AC), Spectral Clustering (SC), and DBSCAN, as shown in Table 12. Note that we provide the number of clusters for both AC and SC. We observe that their performance is comparable with K-Means.

E.3. MST Traversal with Proprietary LLMs

We tested MST Traversal using a proprietary and more advanced LLM, Claude-3.5-Sonnet. Experimental results are shown in Table 13. In general, we observe that Claude achieves better performance than QWen-VL. We argue the performance improvement is because of the stronger reasoning capability of Claude.

Table 13. Evaluation of Proprietary LLMs on Card-Number and Stanford Cars.

	Card-Number		Stanford Cars	
	NMI	RI	NMI	ARI
QWen-VL	72.1	95.1	80.9	33.2
Claude	79.4	96.3	82.2	41.0

Table 14. Effect of MST Traversal on Multi-Sub.

	NMI	RI
Multi-Sub	34.2	80.7
+MST Traversal	37.4	85.9

E.4. Using MST Traversal for Existing Baselines

Most existing baselines directly use K-Means for clustering. We demonstrate that MST Traversal can improve their performance when applied to HDBSCAN, as shown in Table 14. We reimplement Multi-Sub and test it using HDBSCAN. Upon clustering by HDBSCAN, we apply our MST Traversal to improve the clustering result. Experiments again validate the effectiveness of the MST Traversal.

F. Reproduction and Limitations

Reproduction Complete codes and the processed ABO-LC dataset will be released upon acceptance.

Limitations and Future Directions The effectiveness of our approach depends on the quality and specificity of input guidelines. When guidelines are ambiguous or incomplete, the clustering results may not fully align with user intentions. To address this, we have explored two complementary strategies: (1) heuristic prompting techniques that help users articulate their clustering objectives more clearly, and (2) prompt optimization methods (Section C.4) that refine guidelines using minimal feedback. Our experiments (Table 9) demonstrate that even with limited supervision (20 samples), prompt optimization can significantly improve both precision and recall, suggesting a practical path for iterative refinement in real-world deployments.

Although our MST-based traversal algorithm substantially reduces LLM invocations compared to naive approaches (Section A.2, Figure 6), the method still requires multiple LLM calls for complex datasets with thousands of initial clusters. For extremely large-scale applications, exploring more efficient semantic reasoning mechanisms (such as caching strategies, hierarchical merging, or hybrid similarity metrics) represents a promising direction. Nevertheless, our current design achieves a favorable balance between semantic accuracy and computational efficiency, as evidenced by our $O(N \log N)$ complexity analysis.

Table 15. Guideline attributes used for each dataset.

Dataset	Guideline Attributes
CIFAR-10	Object category; Living status; Habitat/Environment; Locomotion; Body type; Size range; Surface covering; Primary color; Key distinguishing features
STL-10	Object type; Main shape; Primary environment; Movement type; Size category; Surface texture; Main color; Key features
ImageNet-10	Object type; Living status; Natural habitat; Primary function; Physical form; Size category; Surface texture; Primary color; Key features
Fruit	Species; Color
Cards	Number; Suits
CIFAR10-MC	Object category; Living status; Habitat/Environment; Locomotion; Body type; Size range; Surface covering; Primary color; Key distinguishing features
CUB Birds	Bill shape; Wing color; Upperparts color; Underparts color; Breast pattern; Back color; Tail shape; Upper tail color; Head pattern; Breast color; Throat color; Eye color; Bill length; Forehead color; Under tail color; Nape color; Belly color; Wing shape; Size; Shape; Back pattern; Tail pattern; Belly pattern; Primary color; Leg color; Bill color; Crown color; Wing pattern
Stanford Cars	Make; Model; Year
Stanford Dogs	Size; Body proportions; Musculature; Skull shape; Ear type; Muzzle length; Facial features; Length; Texture; Color patterns; Density; Tail; Leg length
Oxford Flowers	Petal colors; Color distribution; Center/stamen color contrasts with petals; Petal shape; Number of petals; Arrangement of petals; Flower head structure; Petal texture; Surface patterns; Size relationships between flower parts; Density; Complexity; Growth pattern; Visible foliage characteristics
ABO-LC	Brand; Color; Item id; Model name; Product type; Style

Table 16. GCPM Caption Prompts for General Clustering Datasets.

Dataset	GCPM Caption Prompt
CIFAR-10	"Describe the object in the image with a specific focus on required criteria for general object recognition. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the main object in the image here. - Object category: Provide the broad category here. - Living status: Provide the living status here. - Habitat/Environment: Provide the typical habitat or environment here. - Locomotion: Provide how it moves here. - Body type: Provide the body structure here. - Size range: Provide the typical size here. - Surface covering: Provide the main surface here. - Primary color: Provide the dominant color here. - Key distinguishing features: Describe 2-3 most distinctive visual features that help identify this specific object type."
STL-10	"Describe the object in the image with a specific focus on required criteria for general object recognition. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the main object in the image here. - Object type: Provide the type of object here. - Main shape: Provide the overall shape here. - Primary environment: Provide the typical environment here. - Movement type: Provide how it typically moves here. - Size category: Provide the relative size here. - Surface texture: Provide the main surface texture here. - Main color: Provide the dominant color here. - Key features: Describe 2-3 most distinctive features that help identify this object."
ImageNet-10	"Describe the object in the image with a specific focus on required criteria for general object recognition. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the main object in the image here. - Object type: Provide the broad type here. - Living status: Provide the living status here. - Natural habitat: Provide the natural environment here. - Primary function: Provide the main function here. - Physical form: Provide the physical structure here. - Size category: Provide the typical size here. - Surface texture: Provide the main surface here. - Primary color: Provide the dominant color here. - Key features: Describe 2-3 most distinctive features that help identify this specific object."

Table 17. GCPM Caption Prompts for Multiple Clustering Datasets.

Dataset	GCPM Caption Prompt
Fruit	"Describe the fruit in the image with a specific focus on required criteria. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the fruit here. - Color: Provide the color of the fruit via its hex color code here. Please do not include the exact color because it may be misleading. - Species: Provide the general species of the fruit here. Do not output Undefined or Unclear. If you cannot decide, try your best to guess one."
Cards	"Describe the poker card in the image with a specific focus on required criteria. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the card here. - Suit: Provide the suit of the card here. - Number: Provide the number of the card here."
CIFAR10-MC	"Describe the object in the image with a specific focus on required criteria for general object recognition. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the main object in the image here. - Object category: Provide the broad category here. - Living status: Provide the living status here. - Habitat/Environment: Provide the typical habitat or environment here. - Locomotion: Provide how it moves here. - Body type: Provide the body structure here. - Size range: Provide the typical size here. - Surface covering: Provide the main surface here. - Primary color: Provide the dominant color here. - Key distinguishing features: Describe 2-3 most distinctive visual features that help identify this specific object type."

Table 18. GCPM Caption Prompts for Fine-grained Clustering Datasets.

Dataset	GCPM Caption Prompt
CUB Birds	"Describe the bird in the image with a specific focus on required criteria. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the bird here. - Has bill shape: Provide the bill shape of the bird here. - Has wing color: Provide the wing color of the bird here. - Has upperparts color: Provide the upperparts color of the bird here. - Has underparts color: Provide the underparts color of the bird here. - Has breast pattern: Provide the breast pattern of the bird here. - Has back color: Provide the back color of the bird here. - Has tail shape: Provide the tail shape of the bird here. - Has upper tail color: Provide the upper tail color of the bird here. - Has head pattern: Provide the head pattern of the bird here. - Has breast color: Provide the breast color of the bird here. - Has throat color: Provide the throat color of the bird here. - Has eye color: Provide the eye color of the bird here. - Has bill length: Provide the bill length of the bird here. - Has forehead color: Provide the forehead color of the bird here. - Has under tail color: Provide the under tail color of the bird here. - Has nape color: Provide the nape color of the bird here. - Has belly color: Provide the belly color of the bird here. - Has wing shape: Provide the wing shape of the bird here. - Has size: Provide the size of the bird here. - Has shape: Provide the shape of the bird here. - Has back pattern: Provide the back pattern of the bird here. - Has tail pattern: Provide the tail pattern of the bird here. - Has belly pattern: Provide the belly pattern of the bird here. - Has primary color: Provide the primary color of the bird here. - Has leg color: Provide the leg color of the bird here. - Has bill color: Provide the bill color of the bird here. - Has crown color: Provide the crown color of the bird here. - Has wing pattern: Provide the wing pattern of the bird here."
Stanford Cars	"Describe the car in the image with a specific focus on required criteria. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the car here. - Make: Provide the make of the car here. - Model: Provide the model of the car here. - Year: Provide the year of the car here."
Stanford Dogs	"Describe the dog in the image with a specific focus on required criteria. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the dog here. - Size: Provide the size of the dog here. - Body proportions: Provide the body proportions of the dog here. - Musculature: Provide the musculature of the dog here. - Skull shape: Provide the skull shape of the dog here. - Ear type: Provide the ear type of the dog here. - Muzzle length: Provide the muzzle length of the dog here. - Facial features: Provide the facial features of the dog here. - Coat length: Provide the coat length of the dog here. - Coat texture: Provide the coat texture of the dog here. - Color patterns: Provide the color patterns of the dog here. - Coat density: Provide the coat density of the dog here. - Tail: Provide the tail characteristics of the dog here. - Leg length: Provide the leg length of the dog here."
Oxford Flowers	"Describe the flower in the image with a specific focus on required criteria. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the flower here. - Petal colors: Provide the petal colors of the flower here. - Color distribution: Provide the color distribution of the flower here. - Center/stamen color: Provide the center/stamen color of the flower here. - Petal shape: Provide the petal shape of the flower here. - Number of petals: Provide the number of petals of the flower here. - Arrangement of petals: Provide the arrangement of petals of the flower here. - Flower head structure: Provide the flower head structure of the flower here. - Petal texture: Provide the petal texture of the flower here. - Surface patterns: Provide the surface patterns of the flower here. - Size relationships between flower parts: Provide the size relationships between flower parts of the flower here. - Petal density: Provide the petal density of the flower here. - Bloom complexity: Provide the bloom complexity of the flower here. - Growth pattern: Provide the growth pattern of the flower here. - Visible foliage characteristics: Provide the visible foliage characteristics of the flower here."

Table 19. GCPM Caption Prompts for Long-tailed Clustering Dataset.

Dataset	GCPM Caption Prompt
ABO-LC	"Item Name: item_name Describe the e-commerce product based on its image and item name with a specific focus on required criteria. You should strictly follow the output format below: OUTPUT FORMAT: - Description: Provide a general description of the product here. - Brand: Provide the brand of the e-commerce product here, which might be inferred from the item name. - Model Name: Provide the model name of the e-commerce product here, which might be inferred from the item name. - Product Type: Provide the product type of the e-commerce product here, which might be inferred from the item name. - Style: Provide the style of the e-commerce product here, which might be inferred from the item name."

Table 20. Instruction-Aware Embedding Prompts for Each Dataset.

Dataset	Instruction-Aware Embedding Prompt
CIFAR-10	Identify the object type based on the image description considering its physical characteristics, habitat, and functional properties.
STL-10	Identify the category of the object based on the image description focusing on distinctive visual and functional characteristics.
ImageNet-10	Identify the specific object category based on the image description considering its physical characteristics, function, and distinctive features.
Fruit	Identify the species of the fruit based on the image description. Identify the color of the fruit based on the image description.
Cards	Identify the number of the card based on the image description. Identify the suit of the card based on the image description.
CIFAR10-MC	Identify the type of the object based on the image description. Identify the environment where the object is based on the image description.
CUB Birds	Identify the species of the bird based on the image description.
Stanford Cars	Identify the type of the car based on the image description.
Stanford Dogs	Identify the breed of the dog based on the image description.
Oxford Flowers	Identify the category of the flower based on the image description.
ABO-LC	Identify the variation of the product based on the description focusing on its brand, model name, product type, and style

Table 21. LLM Merging Prompts Used in MST Traversal for Cards conditioned on Number and Stanford Cars.

Dataset	LLM Merging Prompt in MST
Cards-Number	"You are an agent for poker card image clustering. Your task is to determine whether two clusters of poker card images should be merged based on their representative images or image descriptions. (merge_rules) 1. Please observe the main value of the cards for each cluster. If two clusters have the same value of cards, they should be merged. Otherwise, they should not be merged. (/merge_rules)"
Stanford Cars	"You are an agent for car image clustering. Your task is to determine whether two clusters of car images should be merged based on their representative images or image descriptions. (merge_rules) 1. Two clusters can be merged only if their images show cars with the same make, model, and year. Otherwise, they should not be merged. 2. ALL of the following features MUST match EXACTLY between clusters: - Make (manufacturer brand) - Model (specific car model name) - Year (production year) 3. ZERO TOLERANCE POLICY: - If ANY of these three features differ, DO NOT MERGE - Different years of the same make and model should NOT be merged - Different models from the same make should NOT be merged - Different trims or variants should be evaluated based on whether they are considered the same model 4. General appearance similarities are NOT sufficient for merging – the make, model, and year must match exactly. (/merge_rules)"