IN-CONTEXT CLUSTERING WITH LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose *In-Context Clustering* (ICC), a flexible LLM-based procedure for clustering data from diverse distributions. Unlike traditional clustering algorithms constrained by predefined similarity measures, ICC flexibly captures complex relationships among inputs through an attention mechanism. We show that pretrained LLMs exhibit impressive zero-shot clustering capabilities on text-encoded numeric data, with attention matrices showing salient cluster patterns. Spectral clustering using attention matrices offers surprisingly competitive performance. We further enhance the clustering capabilities of LLMs on numeric and image data through fine-tuning using the Next Token Prediction (NTP) loss. Moreover, the flexibility of LLM prompting enables text-conditioned image clustering, a capability that classical clustering methods lack. Our work extends in-context learning to an unsupervised setting, showcasing the effectiveness and flexibility of LLMs for clustering.

1 INTRODUCTION

Central to any clustering procedure is a similarity measure that makes it possible to separate data into meaningful groups. Classical methods often rely on predefined measures, such as k-means with Euclidean distance, and therefore impose strong assumptions on the underlying data distributions. As a result, these approaches often struggle with high-dimensional and semantically complex data such as text (Liu et al., 2003; Shah & Mahajan, 2012), images (Wazarkar & Keshavamurthy, 2018; Chang et al., 2017; Guérin & Boots, 2018), and audio (Meinedo & Neto, 2003; Alwassel et al., 2020), where similarity is context-dependent and cannot be easily captured by a rigid predefined function.

Recent advances in Large Language Models (LLMs) offer a promising alternative through in-context learning (ICL) (Vaswani et al., 2017; Brown et al., 2020), which has been proven effective across a variety of data distributions (Tsimpoukelli et al., 2021; Garg et al., 2022; Gruver et al., 2023; Vacareanu et al., 2024). Instead of using a predefined similarity function, LLMs capture context-dependent relations through an attention mechanism with query and key projections learned from large-scale pretraining. The ability to recognize contextual relationships among in-context examples provides a foundation for flexible clustering that can adapt to diverse data and different criteria. This LLM-based approach particularly excels in *few-shot scenarios involving semantically rich, naturalistic data*, complementing classical methods optimized for structured large-scale datasets.

In this work, we propose *In-Context Clustering* (ICC), extending in-context learning to an unsupervised setting (Figure 1). Different from previous in-context supervised learning that requires multiple input-output pairs in the prompt (Brown et al., 2020), ICC utilizes only unlabeled input data in the context. Given a natural language instruction specifying the clustering objective and a sequence of inputs, the LLM generates cluster labels autoregressively. When the clustering condition changes (e.g., grouping by color instead of class as shown in Figure 5), one can simply modify the prompt without updating model weights or features. We evaluate ICC on numerical data and image data using a variety of synthetic and real-world datasets to demonstrate the effectiveness and flexibility of ICC.

Our paper is structured as follows:

- We demonstrate that LLMs can provide surprisingly strong zero-shot in-context clustering capabilities (Section 3.1).
- We find attention matrices in intermediate layers show salient cluster structures. Moreover, spectral clustering using these attention matrices yields impressive performance (Section 3.2).

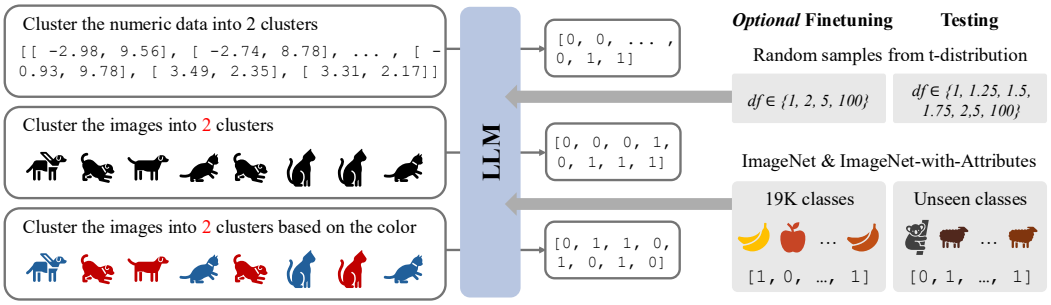

Figure 1: *In-Context Clustering* (ICC). LLMs can flexibly handle diverse modalities and perform text-conditioned clustering. We show the zero-shot clustering capability in pretrained LLMs and further strengthen it through finetuning.

- With lightweight LoRA fine-tuning (Hu et al., 2021) using NTP loss on generated clustering data, we find ICC significantly improves on numeric (Section 4.1) and image data (Section 4.2), especially under heavy-tailed distributions and for images with rich semantics.
- We show that ICC has the relatively distinct ability to do text-conditional image clustering, demonstrating flexibility beyond classical methods. For example, "cluster based on color", or "cluster based on foreground". We believe that this ability to change the way clustering is done based on different prompts makes ICC, and this research direction, particularly compelling. Finally, we show ICC outperforms recent caption-based LLM clustering (Kwon et al., 2024) (Section 5).

2 Related Work

Classical Clustering Algorithms. Classical clustering methods can be classified into hierarchical, partitional, and density-based methods (Jain et al., 1999; Wazarkar & Keshavamurthy, 2018). Hierarchical methods continuously merge data points into clusters based on their similarity with others, resulting in a dendrogram of the data (Ward Jr, 1963; Murtagh & Contreras, 2012). By contrast, partitional clustering algorithms output a single partition of the data instead of a clustering hierarchy (Ikotun et al., 2023). K-means is one of the most widely used partitional clustering methods based on Euclidean distance and works well for spherical Gaussian clusters. Density-based methods can find arbitrarily shaped clusters by detecting the dense regions in the given dataset (Ester et al., 1996). Although widely used, classical methods lack the ability to do representation learning, instead relying on predefined similarity measures that make strong or often unrealistic assumptions about the data. These drawbacks motivate a more flexible clustering algorithm effective for diverse distributions.

LLMs for Text Clustering. LLMs have demonstrated their excellent ability to understand and reason with natural language (Bubeck et al., 2023; Huang & Chang, 2023; Zhang et al., 2024). Recent studies have demonstrated the effectiveness of LLMs in text clustering (Zhang et al., 2023; Viswanathan et al., 2024; Nakshatri et al., 2023; Tipirneni et al., 2024). Various strategies have been explored to enhance clustering performance, including LLM-generated embeddings (Zhang et al., 2023) and few-shot prompting (Viswanathan et al., 2024). However, these practices are limited to text, where the success is somewhat expected, given that the input aligns closely with the pre-training data of the LLMs. In this paper, we extend LLM clustering to non-textual modalities. We find that language pretaining provides a strong foundation for clustering numeric and imagery data.

Multimodal Clustering. Multimodal data introduces challenges in aligning heterogeneous information across modalities. Clustering can be performed jointly across modalities using a shared embedding space, or conditionally where one modality guides the clustering of another. As an example for joint multimodal clustering, Su et al. (2024) propose Multimodal Generalized Category Discovery (Multimodal GCD) that focuses on partitioning a shared multimodal embedding space into known and novel categories. As for conditional multimodal clustering, IC|TC (Kwon et al., 2024) and SSD-LLM (Luo et al., 2025) both leverage LLMs for text-conditioned image clustering by converting images to captions. IC|TC distills image captions into one-word labels using an LLM, which are clustered according to the given textual criteria, and the final assignment is made by prompting the

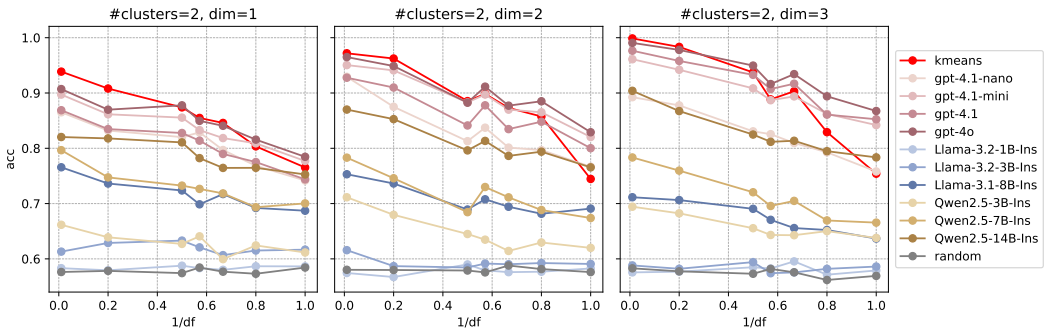

Figure 2: Zero-shot Clustering Accuracy on t-Distribution with Different Degrees of Freedom. When df is small, the data distribution has a heavy tail, which violates the Gaussian assumption of k-means. LLMs show impressive zero-shot clustering capabilities on heavy-tailed data.

LLM to match image captions to the cluster labels. SSD-LLM uses LLMs iteratively to refine and produce subpopulation structures based on image captions, and then utilizes the subpopulation structures for clustering. While the task of text-conditioned image clustering is similar to ours in Section 5, these caption-based approaches are highly constrained by the caption quality, failing to generalize when the data has complicated or nuanced relationships that the captioner is unable to capture.

3 Zero-shot Clustering

In this section, we show that LLMs pre-trained on large text corpus are capable of zero-shot clustering. LLMs outperform k-means on non-Gaussian data, demonstrating their potential to perform in-context clustering. We also observe that a cluster-like pattern emerges in the self-attention of pretrained LLMs and using the attention matrices for spectral clustering results in competitive performance.

3.1 Zero-shot In-Context Clustering

Experimental Setup. To understand the zero-shot clustering capabilities of different model families and model sizes, we test pre-trained Llama 3.1&3.2 (AI@Meta, 2024), Qwen 2.5 (Bai et al., 2023) with different sizes, and various closed-source GPT models (Achiam et al., 2023) including GPT-4o and GPT-4.1 series. We round all numbers to two decimal places and use text to represent the input numeric data as a double list where the inner list represents one data point. Our prompt is as follows:

> *Cluster the following data into {#clusters} clusters. Only output the cluster labels for each point as a list of integers. Data: {input data} Labels:*

Data. We sample data from a t-distribution to evaluate ICC under diverse conditions: When df are large, it approximates the Gaussian distribution; when df are small, it exhibits a heavy tail. We first sample the cluster centroids by drawing each dimension uniformly from $[-10, 10]$, and then generate data points within each cluster by sampling from a t-distribution with the specified df. For each combination of the number of clusters $c \in \{2, 3, 4, 5\}$, dimensions $d \in \{1, 2, 3, 4\}$, and different degrees of freedom $df \in \{1, 1.25, 1.5, 1.75, 2, 5, 100\}$, we generate 100 samples with length randomly drawn from $[10, 50]$. The size of each cluster is also random but forced to be nonempty.

Results. We report zero-shot accuracy[1] in Figure 2 and include more results of different numbers of clusters and dimensions in Figure 6 of Appendix A. LLMs show impressive zero-shot clustering capabilities, outperforming k-means when the data has heavy tails. When df is small, the Gaussian assumption of k-means is violated, leading to a significant drop in performance. GPT-4 and GPT-4.1 outperform k-means when data is heavy-tailed and high-dimensional, demonstrating the potential of applying LLMs for clustering high-dimensional non-Gaussian data.

[1]Since clustering is invariant to label permutation, we adopt the Hungarian Algorithm to find the optimal assignment before computing the accuracy.

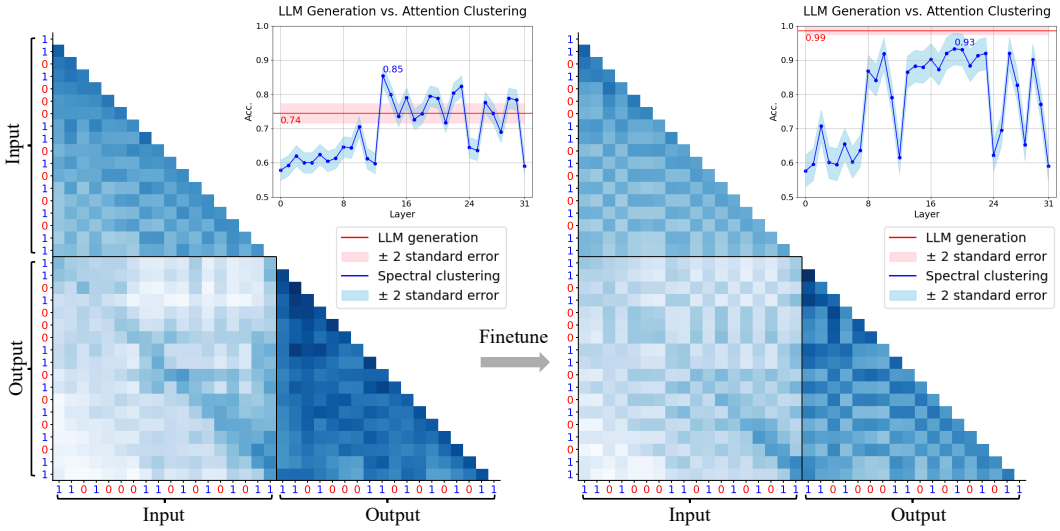

Figure 3: Visualization of Attention Allocation of Input Data and Generated Cluster Labels at an Intermediate Layer. The x-axis and y-axis are the ground-truth cluster labels. The left figure is for the pretrained LLAMA-3.1-8B-INSTRUCT, and the right is after fine-tuning(details in Section 4.1). The top right curves are the average accuracy of spectral clustering using the input-input attention score matrices (top-left) across different layers, compared with the average accuracy of LLM generation.

The performance of LLMs is correlated with the model size and training choices. Small LLMs with 3B or 8B parameters can produce non-trivial answers when the clustering data is simple (with lower dimensions and fewer clusters, shown in Figure 6). When the data becomes more complicated, these small LLMs are either unable to follow the instruction of generating the correct number of clusters or produce answers that are close to random guesses. We also observe that instruction tuning improves the overall accuracy, without which the model is unable to follow the instructions of the clustering task (Figure 7). There is still a gap between the performance of small open-source models and GPT models, probably due to the difference in the model size and pretraining. In Section 4, we show that finetuning Llama models on synthetic clustering data helps close the gap.

3.2 EMERGENCE OF CLUSTERS IN ATTENTION

To better understand the inner mechanism of ICC, we visualize the attention scores across different transformer layers. All LLMs considered here are causal transformers with multi-head self-attention. Given a textual prompt as described in Section 3, the model autoregressively generates cluster labels conditioned on the input data and previous generation. At each layer, we extract the self-attention matrix $A \in \mathbb{R}^{n\times n}$, a lower-triangular matrix due to causality, where n is the total number of tokens. For multi-head attention, we use average attention scores across heads in this section.

To focus on input data and output cluster label tokens, we discard instruction and system prompt tokens. Since each input data point may span multiple tokens, we aggregate token-level attention scores to obtain data-level attention scores. Let m denote the number of input data points. From the full matrix A, we construct an aggregated attention matrix with the following block structure:

$$A = \begin{bmatrix} A^{II} & 0 \\ A^{OI} & A^{OO} \end{bmatrix}. \tag{1}$$

Here, $A^{II} \in \mathbb{R}^{m\times m}$ represents the input-input matrix capturing attention scores among input data points, $A^{OI} \in \mathbb{R}^{m\times m}$ represents the output-input matrix that reflects how generated cluster labels attend to input data, and $A^{OO} \in \mathbb{R}^{m\times m}$ represents the output-output matrix containing attention scores among output tokens. Each input data point d_i may span multiple tokens, indexed from s_i to e_i. We compute A^{II} by averaging attention scores across all token pairs between d_i and d_j:

$$A^{II}_{ij} := \frac{1}{(e_i - s_i + 1)(e_j - s_j + 1)} \sum_{p=s_i}^{e_i} \sum_{q=s_j}^{e_j} A_{pq}. \tag{2}$$

Each output cluster label is represented by a single token, indexed as t_i for the label of d_i. The remaining attention blocks are defined as:

$$A_{ij}^{OI} := \frac{1}{e_j - s_j + 1} \sum_{p=s_j}^{e_j} A_{t_i p}, \quad A_{ij}^{OO} := A_{t_i t_j}. \tag{3}$$

Figure 3 visualizes this block matrix , with A^{II} in the top-left, A^{OI} in the bottom-left, and A^{OO} in the bottom-right. Here, we take one clustering example generated from Gaussian distribution with two clusters. We observe that *attention matrices in intermediate layers show block structures that align with cluster identities*. The transformer assigns higher attention scores to similar data within the same cluster that has been seen in the past. We provide more examples across different layers in Appendix B.1. This cluster pattern is consistent and salient in most middle layers. In contrast, the final layer typically shows a vertical-slash pattern, as also observed by Jiang et al. (2024). We also observe that most attention heads show similar cluster patterns in Figure 10.

Although the pretrained model (left in Figure 3) has a clear cluster pattern in the input-input matrix, clusters are not observed in attention related to outputs. This suggests that the model learns similarity among input data during pretraining, but is not optimized for generating cluster labels as explicit clustering tasks are very likely rare in pretraining.[2] After fine-tuning on ICC data, the cluster structure in the input-input matrix becomes stronger, and similar clusters also emerge in output-input and output-output matrices.

To quantify how well the attention captures the similarity among the input data, we use these input-input attention score matrices for spectral clustering (Ng et al., 2001; von Luxburg, 2007) (more details and results are in Appendix B.2). Although the zero-shot accuracy of prompting pretrained LLAMA-3.1-8B-INSTRUCT to cluster is 74%, the spectral clustering using attention with the optimal choice of layers achieves 85% before fine-tuning. This surprising result suggests that attention of LLMs already encodes rich structural information beyond what is directly generated. In addition to prompting the LLM for generation, directly using attention can be an alternative to leverage pretrained LLM for in-context clustering in zero shot.

4 LEARNING CLUSTERING WITH NEXT TOKEN PREDICTION

While pretrained LLMs show promising zero-shot clustering capabilities, small open-source models lag behind classical methods and proprietary LLMs. In this section, we show that the clusterng capabilities of pretrained LLMs can be further enhanced through LoRA fine-tuning using NTP loss. Inspired by the meta learning literature (Ravi & Larochelle, 2017; Min et al., 2022; Najdenkoska et al., 2023), we construct various clustering episodes to make pretrained (multimodal) LLM learn to cluster in context and then test it on unseen classes. We experiment on both numeric and image data.

4.1 NUMERIC DATA CLUSTERING

Experiment Setup. We follow the standard Supervised Fine-Tuning (SFT) procedure to fine-tune pre-trained Llama models with different sizes (LLAMA-3.2-1B-INSTRUCT, LLAMA-3.2-3B-INSTRUCT, LLAMA-3.1-8B-INSTRUCT) using NTP loss. Similarly to how we construct the clustering data in Section 3, we construct the data by randomly sampling data from a t-distribution with different degrees of freedom $df \in \{1, 2, 5, 100\}$, the number of clusters $c \in \{2, 3, 4, 5\}$, and dimensions of each point $d \in \{1, 2, 3, 4\}$. We generate around 100k input-label pairs, where each sample has a length randomly drawn from $[10, 50]$. We use LoRA (Hu et al., 2021) to fine-tune the pre-trained Llama model for one epoch with an effective batch size of 32 and a learning rate of 5e-4.

Results. We use the test data in Section 3 ($df \in \{1, 1.25, 1.5, 1.75, 2, 5, 100\}$) with $df \in \{1.25, 1.5, 1.75\}$ to test the robustness of the fine-tuned model. During fine-tuning, the LLM exhibits a two-phase learning pattern where it first learns the correct format and then gradually develops a clustering mechanism. Initially, the LLM (especially smaller models with 1B or 3B parameters) struggles with instruction following and produces repetitive outputs. These poorly formatted predictions are heavily penalized by the NTP loss. As training progresses, the model learns to effectively differentiate among cluster labels based on the input data and achieves a high accuracy.

[2]Llama 3 models are claimed to be trained on "15T tokens that were all collected from publicly available sources"(AI@Meta, 2024), but details are not disclosed.

Table 1: Effect of Finetuning on t-Distributed Data with Different Degrees of Freedom. Input $dim = 3$ and number of clusters $c = 3$. We report average accuracy (%) and one standard error.

	DF=1	DF=1.25	DF=1.5	DF=1.75	DF=2	DF=5	DF=100
KMEANS	67.95±1.46	75.43±1.52	85.57±1.20	87.55±1.32	89.05±1.27	95.29±1.00	97.08±0.82
GPT-4O	77.75±1.31	80.60±1.20	86.99±1.15	87.08±1.26	89.56±1.10	93.84±1.03	96.25±0.86
(A) LLAMA-3.2-1B-INSTRUCT	45.40±0.64	47.09±0.71	46.77±0.66	46.63±0.67	46.54±0.69	45.73±0.64	47.36±0.77
(A) + FINETUNE	82.66±1.30	86.45±1.23	91.10±0.90	89.46±1.18	88.76±1.20	95.09±0.93	96.28±0.88
(B) LLAMA-3.2-3B-INSTRUCT	46.71±0.67	46.09±0.72	46.35±0.62	46.85±0.76	46.05±0.82	46.84±0.72	46.35±0.86
(B) + FINETUNE	88.54±1.03	91.05±1.00	94.31±0.77	93.33±0.90	94.51±0.90	98.08±0.49	97.64±0.78
(C) LLAMA-3.1-8B-INSTRUCT	55.29±1.34	55.38±1.44	59.80±1.57	61.09±1.55	61.21±1.47	64.73±1.66	64.42±1.73
(C) + FINETUNE	**90.66**±0.95	**92.20**±0.93	**95.25**±0.54	**94.57**±0.86	**95.44**±0.71	**98.90**±0.31	**97.85**±0.76

As shown in Table 1, all fine-tuned models show superior performance compared to k-means and GPT-4O (the complete results are in Figure 7 of Appendix A). Although these LLMs are fine-tuned on t-distributed data with $df \in \{1, 2, 5, 100\}$, they show generalization capability to more df and different distributions. All fine-tuned models perform consistently well on t-distributed data with new $df \in \{1.25, 1.5, 1.75\}$. While these models are fine-tuned on a symmetric distribution, they also significantly outperform k-means and GPT-4O on a skewed distribution (lognormal) as shown in Table 4 in Appendix A. We also observe that models with higher accuracy tend to be more invariant to permutation in input data, and data augmentation is effective in improving consistency, as shown in Table 5.

We study the effect of fine-tuning by analyzing the attention pattern as visualized in Figure 3. The cluster pattern in the attention score matrix of the input data is significantly more salient after fine-tuning, indicating that the model learns a better similarity function among the data through its attention mechanism during fine-tuning. The accuracy of spectral clustering using attention scores increases as well. More visualization and results are in Appendix B.

4.2 IMAGE CLUSTERING

Here, we extend ICC to multimodal LLMs and present results of image clustering. Given a set of images, the goal is to cluster based on their semantic meanings. By projecting image embeddings obtained from a pretrained visual encoder, LLMs can learn to produce meaningful groupings that outperform an LLM-based method that relies on image captions.

Model. We use `llava-interleave-qwen-7b-hf` (Li et al., 2024a), a multimodal LLM pretrained with multi-image inputs, as our base model. In the LLaVA framework, each image is segmented into 729 patches encoded by a pre-trained ViT, namely the SigLIP's visual encoder (Zhai et al., 2023), then projected through an MLP layer into the embedding space of the base LLM (Bai et al., 2023). While such a high-granularity representation may benefit downstream tasks like object detection, we argue that it is not optimal for clustering tasks. Clustering typically involves a large number of images; thus, using hundreds of tokens per image can quickly exceed context length limitations and significantly increase computational costs during fine-tuning. Additionally, high granularity might be unnecessary for some clustering tasks that only rely on global features.

To address these efficiency concerns, we implement average pooling after the projection layer to reduce per-image token lengths, as illustrated in Figure 4 (left). Each input image is divided into patches, which are preprocessed and flattened (omitted from the figure for clarity), and then encoded by a vision transformer. We reshape the flattened image features back to 2D and then apply average pooling to reduce dimensionality. The pooled features are then flattened, projected into the LLM's embedding space, and concatenated with text token embeddings. We experiment with various pooling kernel sizes in Appendix C.1. No padding is applied and the stride is the same as the kernel width.

Data. We collect images from ImageNet21k (Ridnik et al., 2021) where images sharing the same label are considered part of the same cluster. We reserve the 384 image classes covered in ImageNet-with-Attributes (Russakovsky & Fei-Fei, 2010) for testing and the remaining 18K classes for training. For training, we construct 192K image clustering episodes of various numbers of clusters $c \in \{2, 3, 4\}$, with random length $l \in [10, 30]$ and random cluster proportion. For testing, we use the reserved test classes to construct 100 clustering episodes for each number of clusters. To test generalization on out-of-domain data, we include Plant Disease and EuroSAT datasets from the Cross-Domain Few-Shot Learning (CD-FSL) Benchmark (Guo et al., 2020) with details in Appendix C.2.

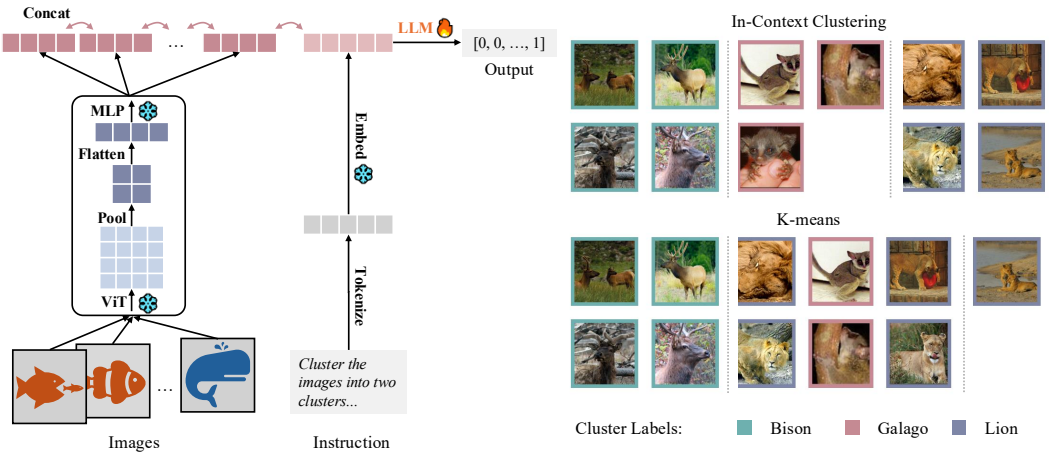

Figure 4: Left: Multimodal LLM Architecture with Average Pooling for Image Features. Right: Qualitative Comparison of Models on Image Clustering — ICC outperforms k-means when the data has rich semantic information.

Table 2: Image Clustering Accuracy (%) with Standard Error. ICC(GPT-4O) is zero-shot ICC using gpt-4o and the shaded rows represent models finetuned on ImageNet data with numbers of clusters $c \in \{2, 3, 4\}$, where SMALL, MEDIUM, LARGE refer to the per-image token length in Appendix C.1. Our finetuned models can generalize to unseen $c = 5$ and other datasets that deviate from ImageNet.

	IMAGENET				PLANT	EUROSAT
NUMBER OF CLUSTERS	C=2	C=3	C=4	C=5	C=2	C=2
K-MEANS	89.43±1.57	82.09±1.44	79.07±1.31	77.96±1.08	**93.70**±1.40	**85.52**±1.43
IC|TC(KWON ET AL., 2024)	90.20±1.54	78.86±1.41	76.49±1.50	73.99±1.58	67.40±1.23	72.97±1.42
ICC (GPT-4O)	82.46±1.40	80.25±1.73	75.91±1.73	78.08±1.50	84.74±1.25	79.08±1.41
ICC (SMALL)	96.81±0.83	91.94±1.03	89.83±1.19	82.08±1.01	73.03±1.58	78.17±1.53
ICC (MEDIUM)	98.26±0.71	**95.92**±0.90	91.62±1.16	84.92±0.95	82.28±1.85	78.64±1.61
ICC (LARGE)	**99.12**±0.41	91.95±0.96	**92.92**±1.06	**84.96**±0.89	85.09±1.80	77.35±1.70

Experiment Setup. Similarly to previous numerical experiments, we use LoRA to fine-tune the LLM with NTP loss. The visual encoder and projection layer are frozen during training. We fine-tune for one epoch with an effective batch size of 32 and a learning rate of 5e-4.

Baselines. To ensure a fair comparison, we use average-pooled image features from the vision encoder of the base model (Li et al., 2024a) as the inputs to k-means. We also compare ICC against IC|TC (Kwon et al., 2024), a recent LLM-based image clustering method. We use the same model (Li et al., 2024a) to generate image captions for IC|TC then use GPT-3.5-TURBO to distill and cluster the captions according to the given number of clusters and the clustering condition. Although converting images to short captions facilitates clustering via LLMs, IC|TC experiences information loss during the captioning and summarization stage, limiting its performance on challenging data.

Results. The performance of different models is summarized in Table 2. While zero-shot ICC using GPT-4O achieves competitive performance, it is less effective than on text-encoded data. This is likely due to the current limitations of multimodal LLMs on long sequences of complex images. Our proposed finetuning method significantly closes this gap, achieving strong performance across all datasets. Despite being only fine-tuned on ImageNet data with the number of clusters less than five, our model can generalize to within-domain data of five clusters and out-of-domain data including plant leaves and satellite images.

With good image features, k-means is effective on datasets with limited semantic complexity, such as Plant Disease and EuroSAT. However, it loses its competence on ImageNet, where images often depict complex scenes involving multiple objects. The caption-based method, IC|TC, performs poorly on Plant Disease or EuroSAT, as its captioning model lacks domain-specific knowledge. This observation highlights a key weakness of caption-based clustering: its dependence on accurate and relevant captions limits its applicability to novel domains. Our model avoids these pitfalls, demonstrating superior flexibility and performance across both general and specialized domains.

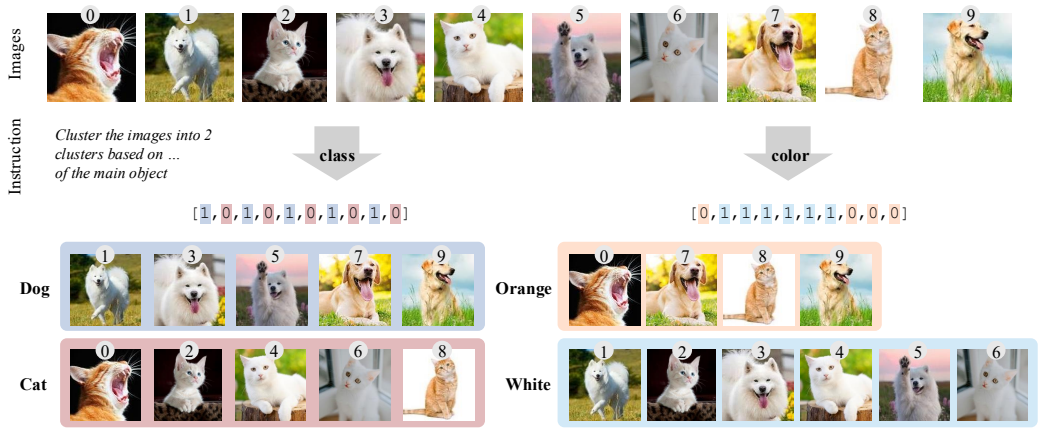

Figure 5: LLMs are able to produce different clusterings according to the condition in the prompt.

5 TEXT-CONDITIONED CLUSTERING

While the experiments in the previous section assume a single, fixed clustering objective, real-world data admits multiple plausible clusterings depending on the objective. For example, the same set of animal images can be clustered by visual properties like colors (orange vs. white) or semantic categories like species (dog vs. cat), as shown in Figure 5. When the clustering condition changes, classical methods typically require retraining or re-engineering features. In contrast, LLMs can easily adapt to new conditions through prompting thanks to their powerful contextual understanding capability. In this section, we perform text-conditioned image clustering by fine-tuning multimodal LLMs with the NTP loss.

Data. We construct conditional clustering using ImageNet-with-Attributes (Russakovsky & Fei-Fei, 2010), which includes 384 classes with 4 categories of attributes (COLOR, SHAPE, PATTERN, TEXTURE). We split the data into 80% training classes and 20% testing classes. We treat the category name as the clustering condition that will be specified in the prompt and use the attribute value as cluster labels. In addition, we include an OBJECT category that is similar to Section 4.2, where we use the class name of the images as cluster labels. Images with ambiguous annotations are filtered out. For training, we construct around 280K image conditional clustering episodes of various numbers of clusters $c \in \{2, 3, 4\}$,[3] with random length $l \in [10, 30]$ and random cluster proportion.

To test the performance of the model on different conditions, we use the reserved test classes of ImageNet-with-Attributes and also include the Stanford 40 Action dataset (Yao et al., 2011) with annotations on the LOCATION of the scene, the ACTION and MOOD of the people in the image provided by (Kwon et al., 2024). For each dataset and clustering condition, we sample 100 clustering data from two random classes of each attribute category, with random size $l \in [10, 30]$ and random cluster proportion.

Experiment Setup. Following the SFT procedure in Section 4.2, we use LoRA to fine-tune `llava-interleave-qwen-7b-hf` with different pooling ratios. We keep the visual encoder and projection layer frozen during training. We use NTP loss to fine-tune for one epoch with an effective batch size of 32 and a learning rate of 5e-4.

Baselines. We test both unconditional and conditional clustering methods. K-means is a unconditional baseline as it does not allow injecting clustering criteria. For conditional clustering methods, we test IC|TC explicitly specifying conditions in the prompts for all the summarization and clustering stages, with GPT-3.5-TURBO as the LLM to save costs.

[3]The pattern category only has two available values, so we don't have $c \in \{2, 3\}$ for this category.

Table 3: Conditional Image Clustering Accuracy (%) with Standard Error. Here, ICC (MEDIUM:4.2) represents the model finetuned on unconditional image clustering data in Section 4.2, while others use conditional image clustering data in Section 5. Our method outperforms all baselines on ImageNet and Stanford 40 Action. SMALL, MEDIAN, LARGE refer to the per-image token length in Appendix C.1.

	IMAGENET					STANFORD 40 ACTION		
	OBJECT	COLOR	PATTERN	SHAPE	TEXTURE	ACTION	MOOD	LOCATION
Unconditional Methods								
K-MEANS	89.96±1.44	66.40±1.16	62.36±0.98	75.76±1.78	78.53±1.65	79.90±1.76	70.93±1.43	78.11±1.50
Conditional Methods								
IC|TC(KWON ET AL., 2024)	91.93±1.38	69.70±1.35	76.12±1.53	70.15±1.34	68.74±1.34	93.74±1.25	75.65±1.35	75.49±1.64
ICC(GPT-4O)	67.58±1.30	66.36±1.22	65.61±1.12	70.15±1.72	73.54±1.54	80.59±1.28	68.61±1.61	67.75±1.33
ICC (SMALL)	98.25±0.71	76.31±1.38	85.50±0.78	81.75±1.69	82.82±1.62	89.60±1.52	67.89±1.27	83.84±1.53
ICC (MEDIUM)	98.64±0.58	81.02±1.31	93.28±0.56	83.02±1.69	86.04±1.52	95.98±1.04	76.77±1.39	77.18±1.67
ICC (MEDIUM:4.2)	98.88±0.55	71.39±1.31	65.04±1.01	72.72±1.37	83.04±1.55	**96.47**±0.95	**78.46**±1.46	**86.19**±1.53
ICC (LARGE)	**99.52**±0.22	**84.29**±1.26	**94.43**±0.40	**83.72**±1.71	**87.27**±1.44	94.14±1.26	73.42±1.47	81.72±1.62

Results. The quantitative evaluation of different models is summarized in Table 3 and qualitative examples are shown in Appendix D. Similar to results in Section 4.2, zero-shot performance of GPT-4O is promising but ultimately falls short of our finetuned approach. Our finetuned models outperform all baselines on ImageNet and Stanford 40 Action. In general, our method with higher per-image token lengths performs better in this conditional clustering task. Unlike experiments in Section 4.2 where the difference between different granularity is small, this task requires more fine-grained information and thus using more tokens to represent images is preferred. K-means and caption-based IC|TC often fail to capture such details, particularly for attributes like COLOR, SHAPE, and PATTERN, where our method is more than 10% higher than all baselines.

Our method generalizes to unseen data and conditions from the Stanford 40 Action dataset. Surprisingly, our model trained solely on clustering objects in ImageNet, achieves the highest accuracy. This suggests that the inductive bias from image-based clustering and the visual-language pretraining enables the model to infer clustering objectives implicitly. We notice that the finetuned models are less competitive on MOOD and LOCATION. We attribute this to the training data (ImageNet-with-Attributes), which emphasizes prominent foreground objects (typically non-human), causing the model to overlook cues from human facial expressions or the background. Scaling our approach to more diverse datasets and clustering conditions could mitigate this bias and further strengthen the model's generalization capabilities.

6 CONCLUSION

In-Context Clustering (ICC) generalizes in-context learning to the unsupervised setting. ICC does not make restrictive assumptions on the input data and enables flexible, text-conditioned clustering objectives through prompting. We find that large LLMs exhibit strong zero-shot performance on text-encoded numeric data, and further show that this capability can be significantly strengthened for smaller and multimodal models through simple fine-tuning using the NTP loss. Multimodal LLMs enhanced by our proposed finetuning achieve impressive performance on image clustering and text-conditioned image clustering. These findings highlight that LLMs can be effectively used to solve clustering tasks that involve complex semantics and contextual understanding.

While we demonstrate ICC's effectiveness and flexibility, ICC is complementary to classical clustering methods, and has certain limitations. For application to larger datasets, it would be particularly promising to scale ICC to longer contexts, which can be computationally expensive for LLMs (Li et al., 2024b; Liu et al., 2024). Our experiments with average pooling for image features show promise in reducing token usage, and recent advances such as dynamic context selection (Hao et al., 2025) and token pruning (Chen et al., 2024; Jianjian et al., 2024) can further address the long-context challenge in future work. Moreover, while visualizing attention provides some insights into the way ICC performs clustering, a theoretical understanding of ICC would be particularly valuable. Emergence of clusters in self-attention have been theoretically studied by Geshkovski et al. (2023), but under a simplified setting (without multi-head attention, feed-forward layers, and layer normalization). Developing theoretical frameworks to explain and exploit these attention structures remains an important open direction.

7 REPRODUCIBILITY STATEMENT

We provide the code for our experiments at `https://anonymous.4open.science/r/ICC-B4CA`. We have included necessary details for reproducing our results in this paper.

- Numeric Data Clustering: The experimental setup for zero-shot experiments, including the data generation process and prompts, is detailed in Section 3 [Experimental Setup & Data]. The fine-tuning procedure is described in Section 4.1 [Experimental Setup]. Additional implementation details are listed in Appendix E.1.
- Image Clustering: The model architecture is described in Section 4.2 [Model]. Data and experiment setup for unconditional and text-conditioned image clustering is in Section 4.2 [Model & Data] and Section 5 [Experimental Setup & Data] respectively. Additional implementation details are provided in Appendix E.2.
- Attention Analysis: Details for processing attention matrices for visualization and spectral clustering is explained in Section 3.2 and Appendix B.

REFERENCES

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

AI@Meta. Llama 3 Model Card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-Supervised Learning by Cross-Modal Audio-Video Clustering. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep Adaptive Image Clustering. *International Conference on Computer Vision (ICCV)*, 2017.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models. *European Conference on Computer Vision (ECCV)*, 2024.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *International Conference on Knowledge Discovery and Data Mining*, 1996.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large Language Models Are Zero Shot Time Series Forecasters. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Joris Guérin and Byron Boots. Improving image clustering with multiple pretrained cnn feature extractors. *arXiv preprint arXiv:1807.07760*, 2018.

Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. *European Conference on Computer Vision (ECCV)*, 2020.

Jitai Hao, Yuke Zhu, Tian Wang, Jun Yu, Xin Xin, Bo Zheng, Zhaochun Ren, and Sheng Guo. OmniKV: Dynamic Context Selection for Efficient Long-Context LLMs. *International Conference on Learning Representations (ICLR)*, 2025.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *Findings of the Association for Computational Linguistics*, 2023.

Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 2023.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 1999.

Huiqiang Jiang, YUCHENG LI, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Cao Jianjian, Ye Peng, Li Shengze, Yu Chong, Tang Yansong, Lu Jiwen, and Chen Tao. MADTP: Multimodal Alignment-Guided Dynamic Token Pruning for Accelerating Vision-Language Transformer. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Sehyun Kwon, Jaeseung Park, Minkyu Kim, Jaewoong Cho, Ernest K. Ryu, and Kangwook Lee. Image clustering conditioned on text criteria. *International Conference on Learning Representations (ICLR)*, 2024.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024a.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024b.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (ACL)*, 2024.

Tao Liu, Shengping Liu, Zheng Chen, and Wei-Ying Ma. An evaluation on feature selection for text clustering. *International Conference on Machine Learning (ICML)*, 2003.

Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as dataset analyst: Subpopulation structure discovery with large language model. 2025.

H. Meinedo and J. Neto. Audio segmentation, classification and clustering in a broadcast news task. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. 2022.

Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 2016.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2012.

Ivona Najdenkoska, Xiantong Zhen, and Marcel Worring. Meta learning to bridge vision and language models for multimodal few-shot learning. 2023.

Nishanth Nakshatri, Siyi Liu, Sihao Chen, Dan Roth, Dan Goldwasser, and Daniel Hopkins. Using LLM for Improving Key Event Discovery: Temporal-Guided News Stream Clustering with Event Summaries. *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Andrew Ng, Michael Jordan, and Yair Weiss. On Spectral Clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems (NeurIPS)*, 2001.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2017.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K Pretraining for the Masses. *arXiv preprint arXiv:2104.10972*, 2021.

Olga Russakovsky and Li Fei-Fei. Attribute Learning in Large-scale Datasets. *ECCV, International Workshop on Parts and Attributes*, 2010.

Neepa Shah and Sunita Mahajan. Document clustering: a detailed review. *International Journal of Applied Information Systems*, 2012.

Yuchang Su, Renping Zhou, Siyu Huang, Xingjian Li, Tianyang Wang, Ziyue Wang, and Min Xu. Multimodal Generalized Category Discovery. *arXiv preprint arXiv:2409.11624*, 2024.

Sindhu Tipirneni, Ravinarayana Adkathimar, Nurendra Choudhary, Gaurush Hiranandani, Rana Ali Amjad, Vassilis N. Ioannidis, Changhe Yuan, and Chandan K. Reddy. Context-Aware Clustering using Large Language Models. *arXiv preprint arXiv:2405.00988*, 2024.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal Few-Shot Learning with Frozen Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Robert Vacareanu, Vlad-Andrei Negru, Vasile Suciu, and Mihai Surdeanu. From Words to Numbers: Your Large Language Model Is Secretly A Capable Regressor When Given In-Context Examples. *Conference on Language Modeling (COLM)*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Vijay Viswanathan, Kiril Gashteovski, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. Large Language Models Enable Few-Shot Clustering. *Transactions of the Association for Computational Linguistics (ACL)*, 2024.

Ulrike von Luxburg. A Tutorial on Spectral Clustering. *arXiv preprint arXiv:0711.0189*, 2007.

Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 1963.

Seema Wazarkar and Bettahally N. Keshavamurthy. A survey on image data analysis through clustering techniques for real world applications. *Journal of Visual Communication and Image Representation*, 2018.

Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. *International Conference on Computer Vision (ICCV)*, 2011.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *International Conference on Computer Vision (ICCV)*, 2023.

Jingyang Zhang and Yueqian Lin. lmms-finetune. URL `https://github.com/zjysteven/lmms-finetune`.

Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Adrian de Wynter, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. LLM as a Mastermind: A Survey of Strategic Reasoning with Large Language Models. *arXiv preprint arXiv:2404.01230*, 2024.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. ClusterLLM: Large Language Models as a Guide for Text Clustering. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Appendix

A Additional Results of Numeric Data Clustering

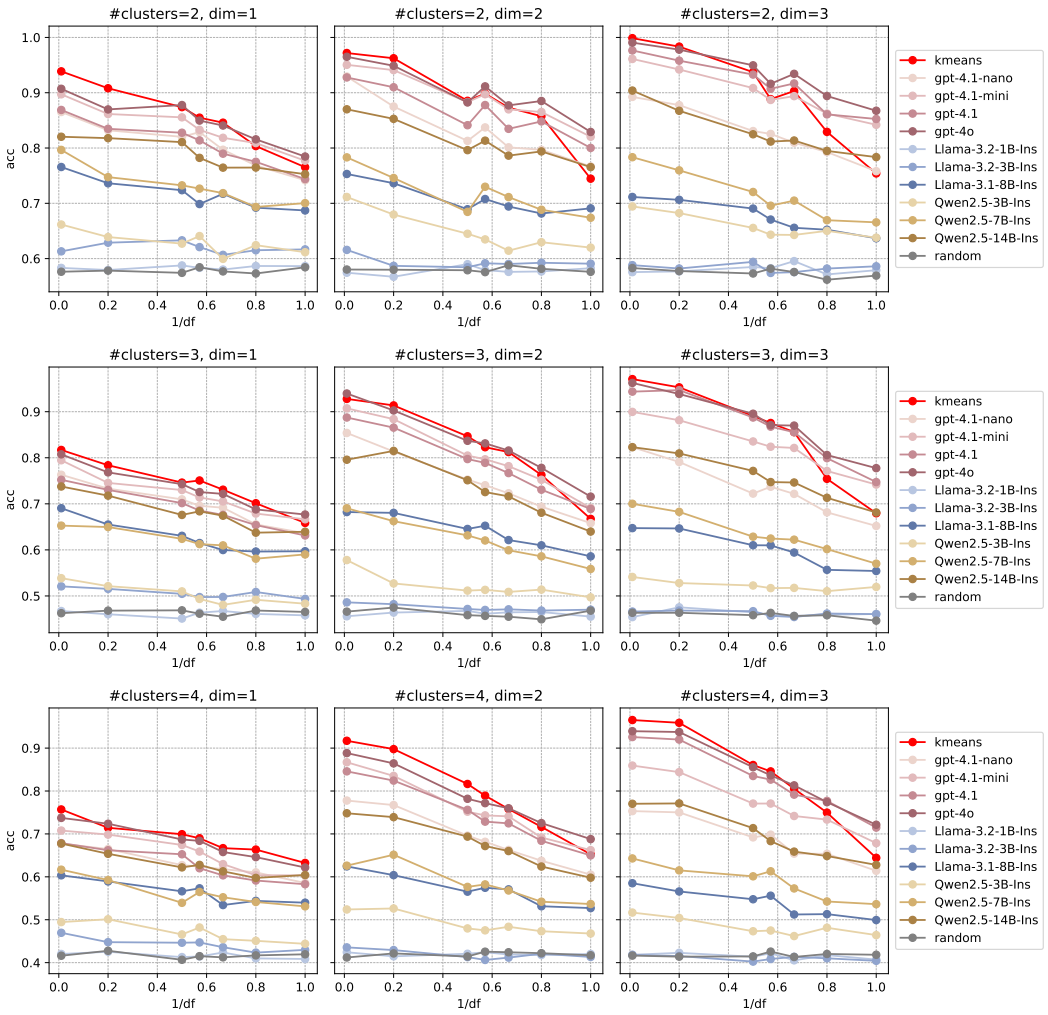

Figure 6: Zero-shot Clustering Accuracy. Test data is t-distributed with different degrees of freedom, number of clusters and dimensions. Note that "Ins" represents "Instruct" in the legend.

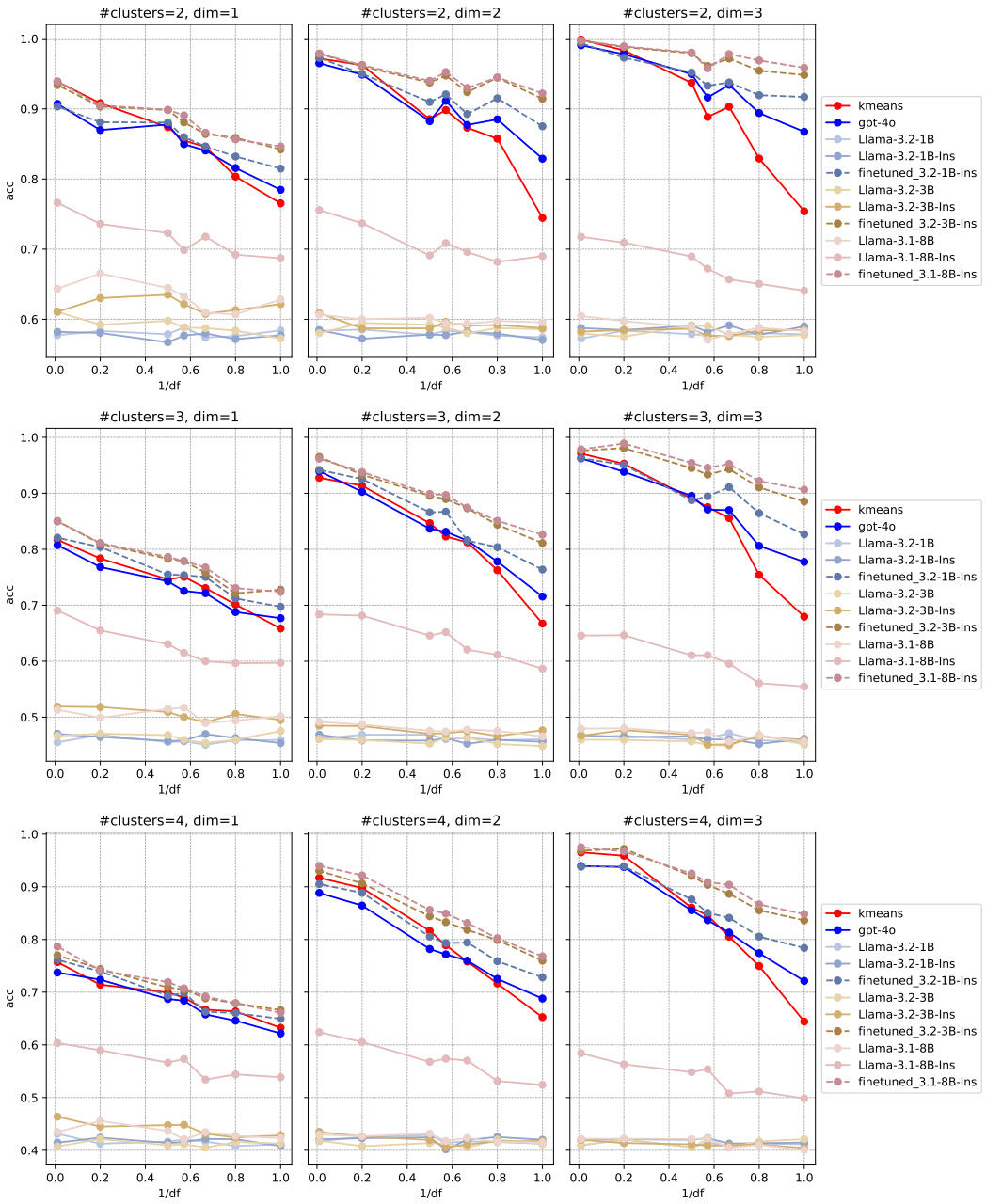

Figure 7: Impact of Instruction Tuning and Clustering-Specific Fine-tuning on Clustering Accuracy. Test data is t-distributed with different degrees of freedom, number of clusters and dimensions. Note that "Ins" represents "Instruct", and "finetune" refers to the fine-tuning on t-distributed clustering data with $df \in \{1, 2, 5, 100\}$ as in Section 4.1.

Table 4: Average Clustering Accuracy with One Standard Error on Lognormal Data. FINETUNED represents the fine-tuned LLAMA-3.1-8B model on t-distributed clustering data with $df \in \{1, 2, 5, 100\}$ as in Section 4.1. Although the model is not fine-tuned on lognormal data, it still outperforms other models in almost all settings.

		$c = 2$	$c = 3$	$c = 4$
$dim = 1$	KMEANS	$0.86_{\pm 0.03}$	$0.77_{\pm 0.02}$	$0.74_{\pm 0.02}$
	GPT-4O	$0.87_{\pm 0.02}$	$0.75_{\pm 0.02}$	$0.73_{\pm 0.02}$
	FINETUNED	$\mathbf{0.89}_{\pm 0.02}$	$\mathbf{0.79}_{\pm 0.02}$	$\mathbf{0.76}_{\pm 0.02}$
$dim = 2$	KMEANS	$0.91_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.82_{\pm 0.02}$
	GPT-4O	$0.91_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.80_{\pm 0.02}$
	FINETUNED	$\mathbf{0.94}_{\pm 0.02}$	$\mathbf{0.91}_{\pm 0.02}$	$\mathbf{0.86}_{\pm 0.02}$
$dim = 3$	KMEANS	$\mathbf{0.98}_{\pm 0.01}$	$0.92_{\pm 0.02}$	$0.91_{\pm 0.02}$
	GPT-4O	$0.94_{\pm 0.01}$	$0.86_{\pm 0.02}$	$0.88_{\pm 0.02}$
	FINETUNED	$0.94_{\pm 0.02}$	$\mathbf{0.94}_{\pm 0.02}$	$\mathbf{0.92}_{\pm 0.02}$

Table 5: Sensitivity to Input Order. The reported values are average accuracy on t-distributed (c=2, dim=3) data, with average standard deviation over five runs of permuted input data in parentheses. We use the standard deviation to reflect the consistency of clustering methods given permutations of input data. FINETUNED denotes the LLAMA-3.1-8B model finetuned on t-distributed clustering data in Section 4.1, and FINETUNED-AUG denotes finetuning on augmented data with 3 times of permutation. We notice that the model with higher clustering accuracy tends to be more invariant to permutation in input data. Data augmentation is also effective in improving the consistency.

	DF=1	DF=2	DF=5	DF=100
K-MEANS	0.75(0.04)	0.95(0.03)	**0.99(0.00)**	**0.99(0.00)**
GPT-4O	0.83(0.08)	0.95(0.03)	0.97(0.02)	0.98(0.01)
FINETUNED	0.92(0.04)	0.97(0.02)	0.98(0.01)	0.99(0.01)
FINETUNED-AUG	**0.93(0.03)**	**0.98(0.01)**	0.98(0.01)	**0.99(0.00)**

B EMERGENCE OF CLUSTERS IN ATTENTION

B.1 ATTENTION OF DIFFERENT LAYERS AND ATTENTION HEADS

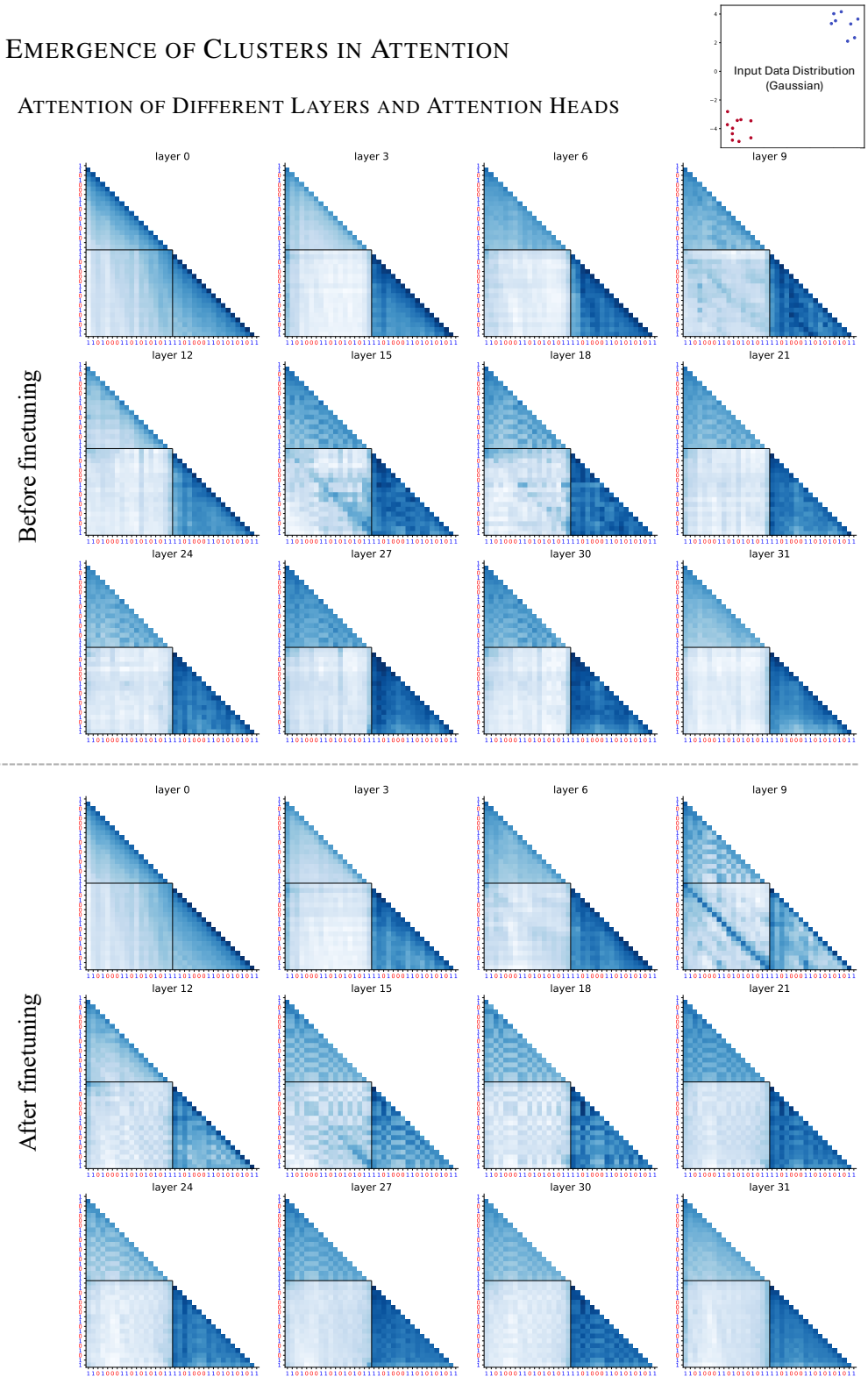

Figure 8: Attention Allocation of LLAMA-3.1-8B-INSTRUCT across Layers. The attention scores are logarithmized for better visualization. Each cluster is generated from a Gaussian distribution, as shown in top right. Figure 3 is a zoom-in view of layer 15 here.

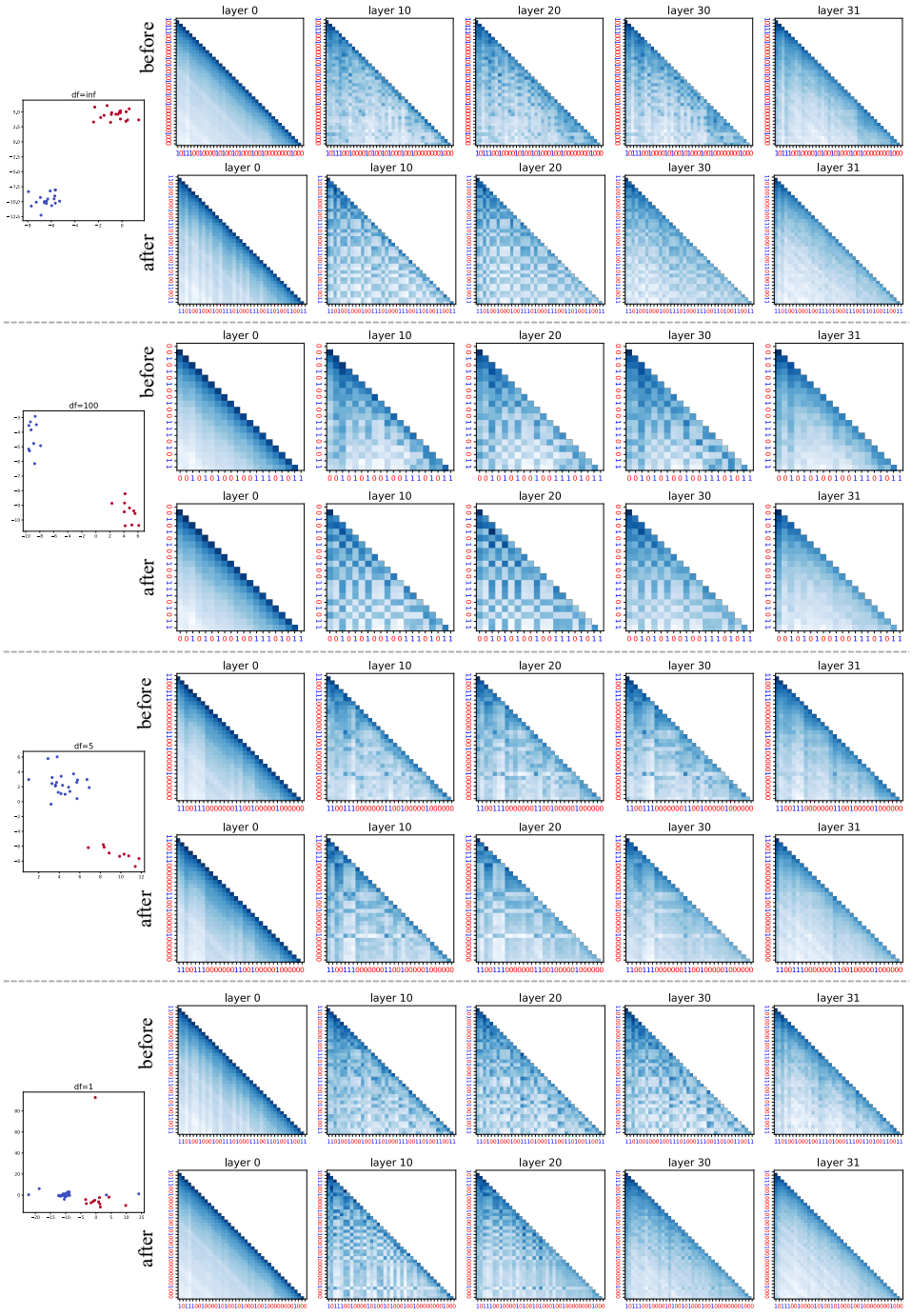

Figure 9: Attention Allocation of LLAMA-3.1-8B-INSTRUCT on t-Distributed Data with Different df, before and after Finetuning. Note that t-distribution with $df = inf$ is Gaussian. The attention scores are logarithmized for better visualization.

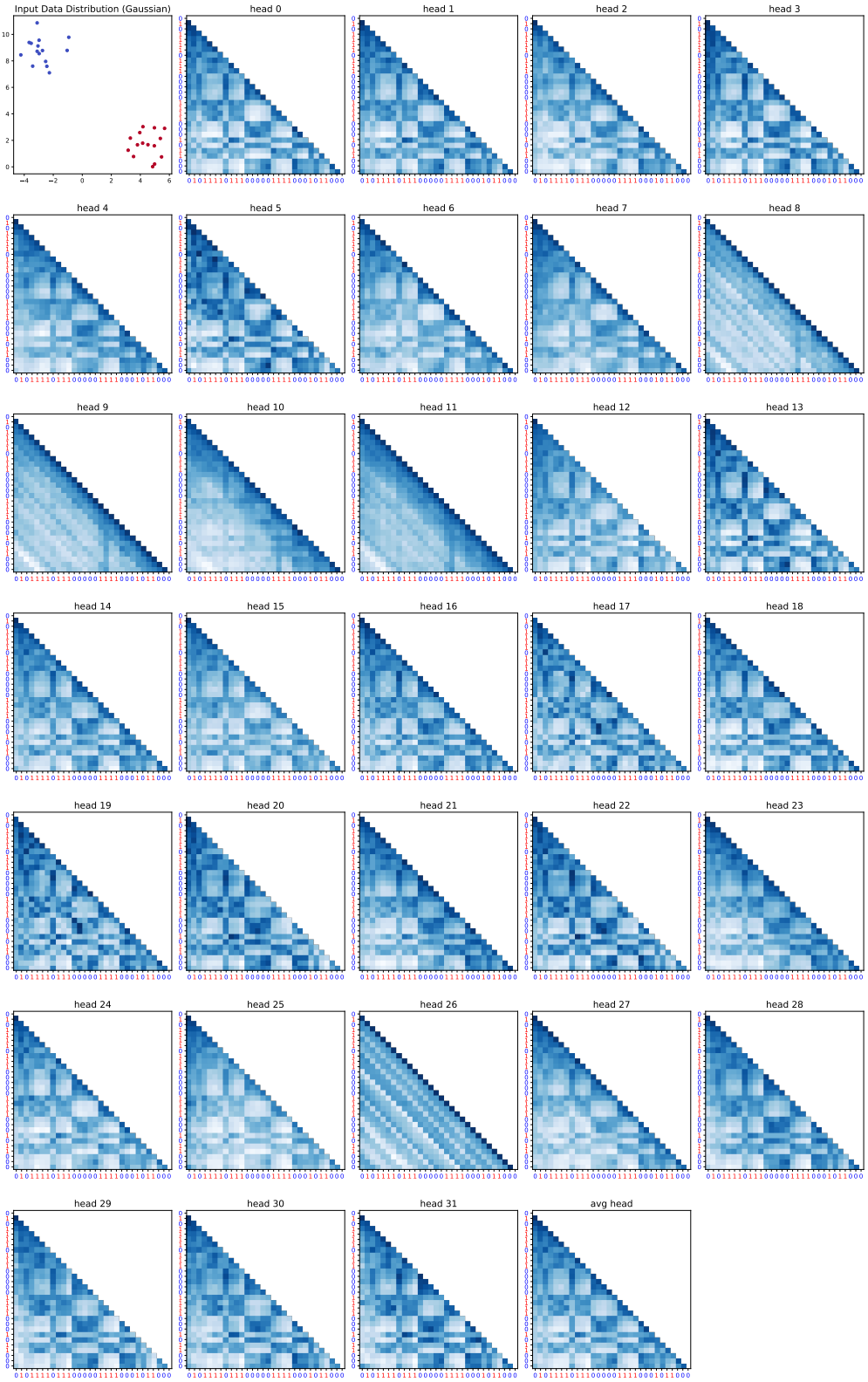

Figure 10: Attention Allocation of LLAMA-3.1-8B-INSTRUCT across attention heads at layer 15. The attention scores are logarithmized for better visualization. Each cluster is generated from a Gaussian distribution, as shown in top left.

B.2 SPECTRAL CLUSTERING

As described in Section 3.2, we perform spectral clustering using the input-input attention score matrix A^{II}. We first standardize A^{II} so that each row sums to one. Due to causality, early tokens cannot attend to later tokens, making the attention scores scale uneven across rows. For example, the second data point always allocates very high attention to the first one regardless of its semantic similarity. To mitigate this imbalance, we further rescale each row by the number of non-zero entries in the row. Finally, we symmetrize the matrix and the resulting matrix is used as the precomputed affinity matrix for spectral clustering. The complete preprocessing procedure is visualized in Figure 11. We use the `sklearn.cluster.SpectralClustering` implementation.

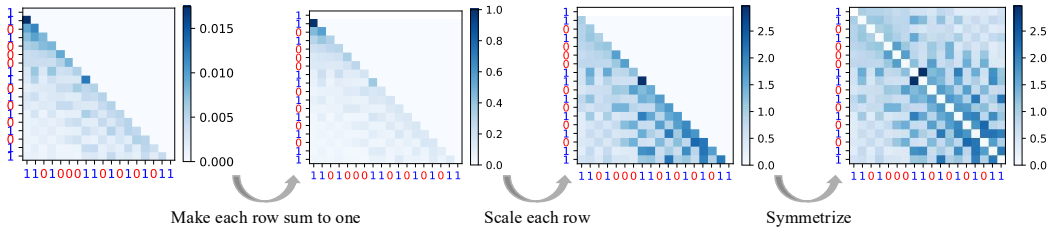

Figure 11: Preprocessing Attention Matrix for Spectral Clustering.

Table 6: Spectral Clustering using Attention Scores. Reported values are average accuracy on t-distributed test data as in Section 3, with one standard error. Models used here are pretrained LLAMA-3.1-8B-INSTRUCT and its fine-tuned checkpoint as in Section 4.1. SC represents spectral clustering using attention scores with OPT denoting the highest accuracy across all layers and L23 denoting the accuracy using a fixed layer 23 (indexing from 0). GEN represents generation using direct LLM prompting. Spectral clustering using attention achieves surprisingly competitive performance that outperforms the raw generation before finetuning.

MODEL	METHOD	DF=1	DF=1.25	DF=1.5	DF=1.75	DF=2	DF=5	DF=100
num of clusters = 2, dim = 1								
PRETRAINED	SC(OPT)	0.68±0.01	0.70±0.01	0.73±0.01	0.73±0.02	0.71±0.01	0.79±0.02	0.79±0.02
	SC(L23)	0.68±0.01	0.68±0.01	0.72±0.01	0.73±0.02	0.71±0.02	0.79±0.02	0.79±0.02
	GEN	0.69±0.01	0.69±0.01	0.72±0.01	0.70±0.01	0.72±0.01	0.74±0.02	0.77±0.01
FINETUNED	SC(OPT)	0.70±0.01	0.72±0.01	0.73±0.01	0.74±0.02	0.74±0.02	0.79±0.02	0.79±0.02
	SC(L23)	0.67±0.01	0.70±0.02	0.72±0.02	0.72±0.02	0.72±0.02	0.76±0.02	0.75±0.02
	GEN	0.85±0.01	0.86±0.01	0.87±0.01	0.89±0.01	0.90±0.01	0.91±0.01	0.94±0.01
num of clusters = 2, dim = 2								
PRETRAINED	SC(OPT)	0.75±0.01	0.76±0.02	0.79±0.02	0.78±0.02	0.81±0.02	0.82±0.02	0.88±0.02
	SC(L23)	0.71±0.01	0.74±0.02	0.73±0.02	0.76±0.02	0.78±0.02	0.80±0.02	0.87±0.02
	GEN	0.69±0.01	0.68±0.01	0.69±0.01	0.71±0.01	0.69±0.01	0.74±0.02	0.75±0.01
FINETUNED	SC(OPT)	0.84±0.01	0.84±0.02	0.85±0.02	0.87±0.01	0.87±0.01	0.89±0.02	0.96±0.01
	SC(L23)	0.77±0.02	0.81±0.02	0.80±0.02	0.82±0.02	0.83±0.02	0.87±0.02	0.94±0.01
	GEN	0.92±0.01	0.94±0.01	0.93±0.01	0.95±0.01	0.94±0.01	0.96±0.01	0.98±0.01
num of clusters = 2, dim = 3								
PRETRAINED	SC(OPT)	0.77±0.02	0.79±0.02	0.78±0.02	0.80±0.02	0.83±0.02	0.85±0.02	0.88±0.02
	SC(L23)	0.68±0.01	0.71±0.02	0.73±0.02	0.74±0.02	0.76±0.02	0.81±0.02	0.85±0.02
	GEN	0.64±0.01	0.65±0.01	0.66±0.01	0.67±0.01	0.69±0.01	0.70±0.02	0.71±0.02
FINETUNED	SC(OPT)	0.90±0.01	0.91±0.01	0.93±0.01	0.91±0.01	0.93±0.01	0.96±0.01	0.99±0.00
	SC(L23)	0.83±0.02	0.86±0.02	0.89±0.02	0.87±0.02	0.91±0.01	0.95±0.01	0.97±0.01
	GEN	0.96±0.01	0.97±0.01	0.98±0.00	0.96±0.01	0.98±0.00	0.99±0.00	1.00±0.00

C Additional Experiment Details and Results of Image Clustering

C.1 Pooling

Table 7: Pooling kernel size and corresponding per-image token length. The original pixel size is 384x384 with a patch size of 14, resulting in 27x27(729) image tokens.

	POOLING KERNEL	TOKEN LENGTH
DEFAULT	1X1	27 X 27 (729)
LARGE	2X2	13 X 13 (169)
MEDIUM	3X3	9 X 9 (81)
SMALL	9X9	3 X 3 (9)

C.2 Out-of-Domain Image datasets

To test the generalization capability of the model, we include two more image datasets from Cross-Domain Few-Shot Learning (CD-FSL) Benchmark Guo et al. (2020).

- Plant Disease Mohanty et al. (2016): Leaves of different trees that are healthy or have different crop diseases. We construct 100 clustering samples based on the plant names, where each sample contains 10-30 images from 3 random classes.
- EuroSAT Helber et al. (2019): Satellite images of different land use and land cover classes. We construct 100 clustering samples where each sample contains 10-30 images from 3 random classes.

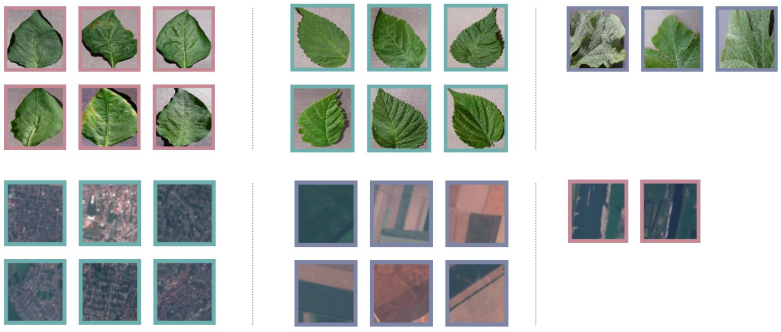

Figure 12: Example of Plant Disease and EuroSAT datasets. The color of frame represents different clusters predicted by our model. Our model can generalize to these images that are quite different from ImageNet.

C.3 Attention

Similar as the numeric experiments in Section 3.2, we visualize the attention allocation for image clustering below (Figure 13). The model used here is fine-tuned model (medium) as in Section 4.2. The attention scores have block structures that roughly align with the ground-truth identities in intermediate layers. We notice that the allocation of attention weights can be uneven within one cluster, where representative samples are assigned with higher weights. The attention patterns for images are generally more complicated than those for synthetic low-dimensional data due to the semantically rich information in images.

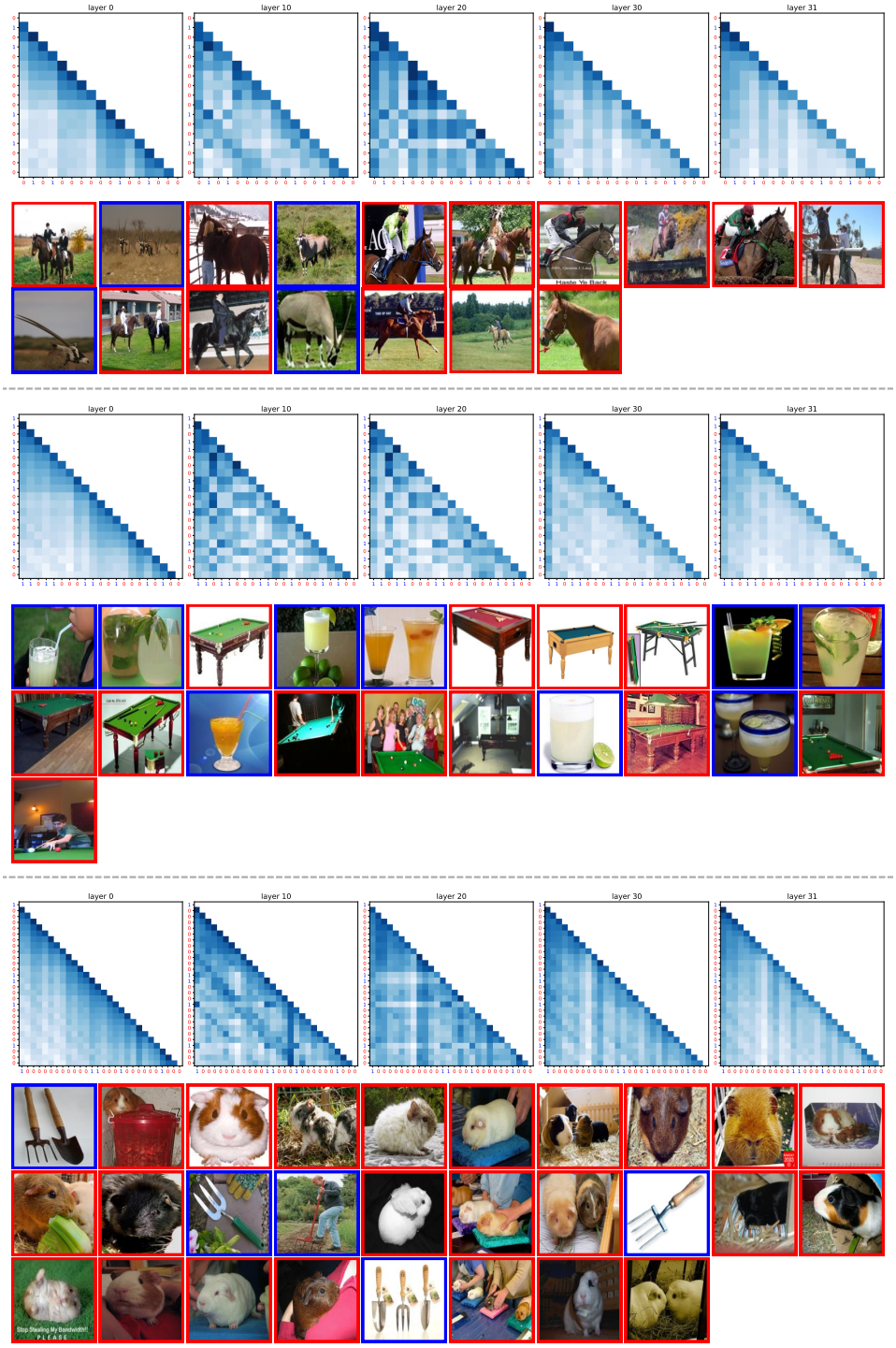

Figure 13: Attention Allocation of Image Clustering. Different colors represent different clusters.

D ADDITIONAL RESULTS FOR CONDITIONAL IMAGE CLUSTERING

Figure 14: Examples of ICC on ImageNet-with-Attributes. The color of the frame indicates different clusters predicted by our model. Most of the images contain multiple objects, making the task more challenging.

E Additional Experiments Details

E.1 Numeric Clustering

For all Llama AI@Meta (2024) and Qwen Bai et al. (2023) models, we use the implementation and checkpoints from HuggingFace. Specifically, we test the following models in Section 3.1.

- `meta-llama/Llama-3.2-1B`, `meta-llama/Llama-3.2-1B-Instruct`, `meta-llama/Llama-3.2-3B`, `meta-llama/Llama-3.2-3B-Instruct`, `meta-llama/Llama-3.1-8B`, `meta-llama/Llama-3.1-8B-Instruct`;
- `Qwen/Qwen2.5-7B-Instruct`, `Qwen/Qwen2.5-3B-Instruct`, `Qwen/Qwen2.5-14B-Instruct`

We then fine-tune Llama models in Section 4.1. We use LoRA Hu et al. (2021) with $r = 64$ and $alpha = 16$ for finetuning. We use an initial learning rate of $lr = 5e - 4$ with a cosine learning rate scheduler. We use one A100 and a batch size of 32. We save the checkpoint with the lowest validation loss.

E.2 Image Clustering

For both unconditional (Section 4.2) and conditional image clustering (Section 5), we use `llava-interleave-qwen-7b-hf` as our base model and use the implementation and checkpoints from HuggingFace. We use LoRA Hu et al. (2021) with $r = 64$ and $alpha = 16$. We use an initial learning rate of $lr = 5e - 4$ with cosine learning rate scheduler. We use two A100s and an effective total batch size of 32. For models with smaller pooling kernels (and thus higher per-image token length), we use gradient accumulation (Table 8). Our finetuning code is adapted from Zhang & Lin. We save the checkpoint with the lowest validation loss.

Table 8: Pooling Kernel Size and Per-Device Batch Size.

	Pooling Kernel	Token Length	Per-Device Batch Size	Gradient Accumulation
Large	2x2	13 x 13 (169)	4	4
Medium	3x3	9 x 9 (81)	8	2
Small	9x9	3 x 3 (9)	8	2

F The Use of Large Language Models (LLMs)

We only use an LLM for proofreading. We have carefully reviewed all suggestions and take full responsibility for the final content of the paper.