Interpretable Text-Guided Image Clustering via Iterative Search

Bingchen Zhao
zhaobc.gm@gmail.com
Oisin Mac Aodha
oisin.macaodha@ed.ac.uk
University of Edinburgh
Edinburgh, UK

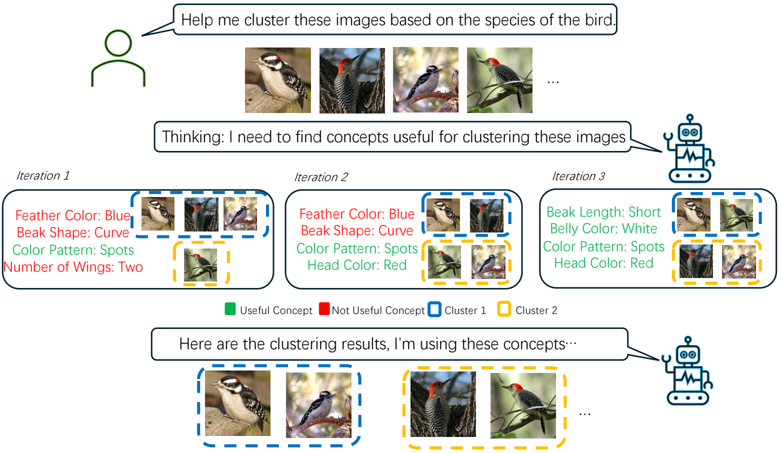

Figure 1: We propose **ITGC**, a new approach for text-guided image clustering that takes a set of unlabeled images, along with a high-level user text query, as input and automatically returns a clustering of the images that is aligned to the user's intent. ITGC conducts an iterative search procedure, guided by an unsupervised objective, that enables it to discover interpretable visual concepts to assist clustering.

Abstract

Traditional clustering methods aim to group unlabeled data points based on their similarity to each other. However, clustering, in the absence of additional information, is an ill-posed problem as there can be many different, yet equally valid, ways to partition a dataset. Distinct users may want to use different criteria to form clusters in the same data, *e.g*. shape *vs*. color. Recently introduced text-guided image clustering methods aim to address this ambiguity by allowing users to specify the clustering criteria of interest using natural language instructions. This instruction provides the necessary context and control needed to obtain clusters that are more aligned with the users' intent. We propose a new text-guided clustering approach named ITGC that uses an iterative discovery process, guided by an unsupervised clustering objective, to generate interpretable visual concepts that better capture the criteria expressed in a user's instructions. We report superior performance compared to existing methods across a wide variety of image clustering and fine-grained classification benchmarks.

1 Introduction

Clustering is a fundamental task in computer vision, enabling the organization of unlabeled visual data into meaningful groups based on inherent patterns contained within. However, traditional clustering methods attempt to address an ill-posed problem, as for any given dataset, multiple equally different groupings/partitionings of the data can be valid. In the fully unsupervised setting, there is no guarantee that a clustering method effectively groups the data based on the visual criteria desired by the user. To obtain groupings that encode structure that are relevant to the visual criteria of interest to users, some form of guidance to direct the clustering process is needed.

Multiple clustering (*i.e.* multi-criteria) methods attempt to provide diverse partitionings of unlabeled data [19, 39, 40, 59, 67], but they cannot be guided based on users' preferences and any structure discovered needs to be explained post-hoc. Category discovery methods aim to group unlabeled data into clusters by leveraging the supervision signal available from additional labeled data [15, 55, 61]. This labeled data guides the model during training to extract the types of visual features deemed important and relevant for defining clusters in the unlabeled data. This dependence on an already available labeled dataset limits the applicability of these methods in scenarios where such labels are unavailable or when the desired clustering criteria is dynamic and user-dependent. Moreover, existing methods can be biased towards fixed semantic labels (*e.g.* semantic clusters [57]), and lack the flexibility to accommodate multiple or nuanced clustering criteria.

Recently a set of text-guided clustering approaches have been proposed that capitalize on advances in pre-trained large language models (LLMs) and vision-language models (VLMs). They enable users to specify the clustering criteria they wish to group images with via a concise high-level natural language instruction. IC|TC [29] uses an VLM to caption images and then performs clustering in text space using an LLM, while Multi-MaP [68] generates a set of language concepts using an LLM and then further refines VLM extracted image embeddings using an optimization approach. While effective, these methods can fail to best use the visual signal available [29] or produce non-interpretable representations [68]. Our work takes a different definition for interpretable clustering. Instead of generating a human readable cluster name like in IC|TC, our work performs clustering in a feature space where each feature is described by a semantic name. This approach enables a more fine-grained manipulation of the clustering by changing the features used for clustering.

We propose a novel Iterative and interpretable Text-Guided image Clustering approach named **ITGC** that is inspired by recent work on learning interpretable image classifiers [8]. We begin with an Concept Generator, which processes a text query to produce an initial list of concepts relevant to a users' query. These language concepts are then mapped to images via a Concept Encoder which results in a set of embeddings that encodes the relevance of each concept for each image. These concept vectors can be clustered to form groups. The quality of the resulting clusters is evaluated using an unsupervised loss which provides a feedback signal to the Concept Generator to guide the generation of a further refined set of concepts. This iterative process of concept generation, encoding, clustering, and evaluation continues until convergence is achieved, ensuring that the model learns to more effectively cluster images based on the user-provided criteria of interest. This allows for dynamic and flexible clustering without the need for labeled examples or predefined distance metrics. Our results show that **ITGC** obtains more accurate and meaningful clusterings, outperforming existing zero-shot, multiple clustering, category discovery, and text-guided clustering methods.

In summary, we make the following contributions: (i) We introduce **ITGC**, a new approach for text-guided image clustering that takes a text query as input and clusters unlabeled data using a learned interpretable feature space. (ii) We show that iterative application of our method, guided by an unsupervised objective, results in improved clusterings for successive time steps. (iii) We demonstrate superior performance over existing methods, providing interpretable and user-aligned results on both existing multi-criteria and fine-grained datasets.

2 Related Work

Clustering. Clustering is an unsupervised task where the goal is to partition data points into discrete groups. There is a large body of literature on classical methods that performing clustering on fixed input features [21], but most success of late has come from deep clustering methods that also learn visual representations [48]. Deep methods can be broadly grouped into those that perform some form of alternating optimization between cluster assignment and representation learning [62, 63] and those that perform end-to-end learning with a clustering specific objective [14, 20, 22, 31]. Clustering objectives have also been shown to be effective for learning visual features in the context of representation learning [1, 6, 7, 54]. The main limitation of conventional clustering methods is that they only produce a single partitioning/clustering of the data.

A single clustering cannot capture the full diversity of most datasets, where multiple valid groupings (e.g., by color or shape) may exist. Multi-criteria clustering addresses this by generating multiple partitions of the data [19]. Note, multi-*criteria* clustering is distinct from multi-*view* clustering [65]. In multi-view clustering each data instance is encoded using different modalities (*e.g*. text and images). Despite being more realistic, multi-criteria clustering remains underexplored, partly due to limited benchmark datasets. Basic methods rely on varying clustering hyperparameters [3], but are constrained by feature expressiveness. Recent deep methods include matrix factorization requiring multi-view data [59], augmentation-based diversity [67], and loss functions promoting distinct clusterings [39, 40]. However, these remain fully unsupervised and lack user control over the clustering criteria.

Label-guided Clustering. Given set of labeled and unlabeled data, category discovery methods aim to partition the unlabeled data into a set of discrete groups. They can can be viewed as a form of label-guided clustering, whereby the labeled data informs they types of concepts/categories that should be discovered in the unlabeled data. For example, if the labeled data contained a set of birds species, it would be reasonable to assume that the unlabeled data should be grouped based on the 'species' criteria. In novel category discovery it is assumed that the unlabeled data only contains instances from novel categories [11, 15, 16, 17, 64, 73, 76], while in the generalized setting the unlabeled data can contain instances from both previously seen and novel categories [4, 10, 44, 47, 55, 57, 61, 71]. Various approaches have been explored in the literature, typically relying on some combination of self-supervised and supervised representation learning. There has also been work in other variants of the discovery problem such as online/incremental [25, 50, 72, 74], federated [45], active [37], and work investigating the impact of different sets of labeled data [75].

With limited exception, most previous methods only investigate discovery in the context of one visual criteria, *i.e*. the semantic category. This is predominately an artifact of how existing category discovery datasets are constructed, *i.e*. the commonly used evaluation datasets are typically generated from existing visual categorization datasets [56] where the categories contained within correspond to semantic concepts (*e.g*. different dog breeds). As a result, there is only one type of visual criteria (*i.e*. the semantic category) present in the data. In practice, image data can be grouped using many different visual criteria. This type of multi-criteria reasoning has been explored in metric learning in cases where the criteria are

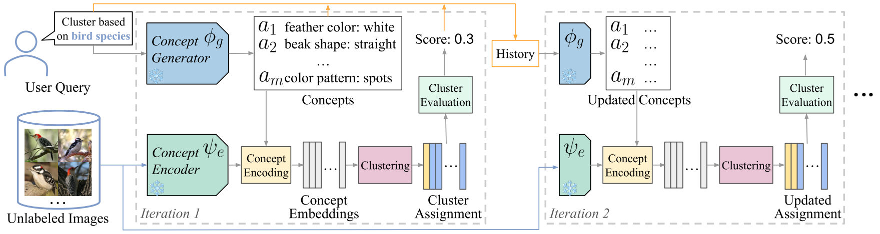

Figure 2: **Overview of ITGC, our proposed text-guided image clustering method.** *(Itr. 1)* The Concept Generator first generates an initial set of concepts based on the user provided text query. The Concept Encoder then encodes the images into the concept space to perform clustering. An unsupervised clustering evaluation score is used to measure how effective the clustering is. *(Itr. 2)* Then the Concept Generator takes the entire previous history, including the concepts used for the previous clusterings and the scores for each clustering, to generate a new set of concepts. This iterative process is repeated until convergence, or a predefined number of iterations.

known in advance [58] or need to be estimated [26, 42]. While these metric learning methods aim to learn representations that encode different visual criteria simultaneously, they do discover novel concepts within each criteria.

Vaze et al. [57] introduced the Clevr-4 dataset to study multi-criteria category discovery, extending CLEVR [24] with four grouping criteria: shape, texture, color, and count. Their experiments show that pre-trained vision models are biased toward certain criteria (e.g., shape). However, their approach assumes access to labeled data for the target criterion. In contrast, we consider a more realistic setting where only the name of the desired visual criterion is known, without any labeled category examples.

Text-guided Clustering. Recent advances in vision-language models (VLMs) enable zero-shot image classification by matching images to a predefined set of text-based concepts [23, 46, 70]. However, this requires the target concepts to be known in advance, limiting its use for clustering. To address this, Li et al. [32] propose matching a large candidate vocabulary (e.g., WordNet nouns [41]) to image clusters using CLIP, followed by filtering and k-means on combined text-image features. While effective for single-criteria clustering, their method is constrained by the vocabulary, produces only one clustering, and offers limited control.

The most relevant work to ours is IC|TC [29], which performs image clustering based on user-specified text criteria. Given a set of unlabeled images, a textual criterion, and the desired number of clusters, it uses a VLM (e.g., LLaVA [34]) to caption each image, then summarizes the captions using an LLM (e.g., GPT-4 [13]) to generate candidate cluster labels. These are clustered via k-means, and the LLM assigns each caption to a cluster. Since the process operates solely on text after captioning, IC|TC is better viewed as a text clustering method grounded in image-generated captions. Related work includes [51], which injects domain knowledge into the captioning step, and [36], which jointly discovers both criteria and clusters from captions. [35] also uses LLMs to generate concepts for fine-grained categorization, but does not address multi-criteria clustering where ITGC is focused on. Additionally, in experiments we show that ITGC can improves the generated concepts via an unsupervised refinement process.

Multi-MaP [68] similarly generates concept words with an LLM and optimizes their embeddings using gradient descent to match VLM features, but loses interpretability in the process of obtaining the continuous embeddings. We also explore the same text-guided clustering setting as [29, 68], but introduce an iterative approach that refines concepts over

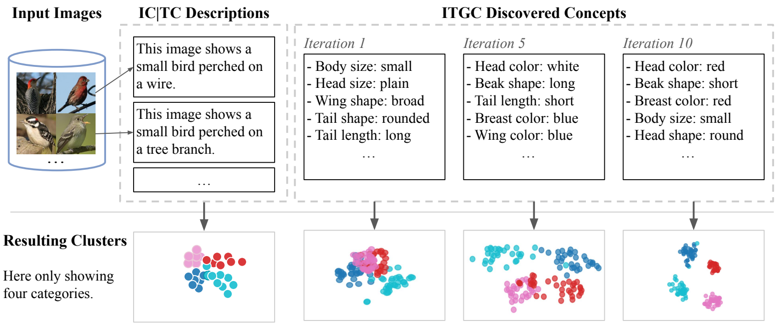

Figure 3: **Examples of generated concepts for CUB.** IC|TC [29] cannot meaningfully separate visual concepts when its captions for them are similar. In contrast, ITGC discovers more interpretable concepts that better fit the clustering task, *i.e*. bird species clustering.

time, resulting in an interpretable concept-bottleneck style [27] representation for clustering.

We take inspiration from [8], which outlines an approach for evolving interpretable classifiers using a pre-trained LLM and VLM to iteratively discover a set of visual concepts which can be used to represent a class. Their goal is classification, and they are provided with images that are already grouped into different categories. However, the semantic category text labels are not provided. They use the labels to train a classifier at each iteration to help find more relevant concepts. In contrast, we begin with an unlabeled set of images and are only provided a high-level criteria from the user which we use to discovery clusters.

3 Method

We first introduce our text-guided image clustering problem setting, and then describe our new **ITGC** approach that leverages recent advances in vision-language and large language models (VLMs and LLMs) in an iterative framework.

3.1 Problem Formulation

Our goal is to cluster unlabeled images based on a user-defined criteria provided via a single natural language query/instruction. Our approach diverges from traditional clustering and category discovery methods by eliminating the need for labeled datasets or predefined distance metrics. Given a dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, *i.e*. a collection of images, and a user provided natural language query $\mathcal{T}$, the objective is to partition $\mathcal{D}$ into k clusters $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ such that the clustering aligns with the visual criteria specified in the query. $C_k = \{\mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \dots, \mathbf{x}_{k_n}\}$ contains the images in the k-th cluster. Here, each image $\mathbf{x}_i$ is assigned to only one cluster, but more than one image can be assigned to each cluster as $k << n$. Thus we denote the clustering results as $\mathcal{D}_{\mathcal{T}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, where $y_i \in \{1, 2, \dots, k\}$ is the assignment for each image to a cluster.

3.2 ITGC Overview

The core idea of ITGC is to perform clustering in a language concept space, where each dimension represents a score indicating the presence of a single language-based concept in a given image. As a result, the clustering performed in this space can yield interpretable results. By leveraging the knowledge encoded in LLMs, we can extract suitable concepts for the given query criteria to guide the clustering process. In this context, a language concept refers to a specific value of a visual attribute present in an image. For example, instead of using a general concept like *"bird wing colors"*, we specify it as *"bird wing color: red"*.

Specifically, we utilize an LLM as a *Concept Generator* ϕ_g to interpret the user provided query $\mathcal{T}$ and generate a set of relevant concepts/attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$. The Concept Generator is explicitly prompted to generate concepts with specified values as mentioned previously. These concepts serve as the basis for defining the clustering criteria. Next, a *Concept Encoder* ψ_e is employed to project each data point $\mathbf{x}_i \in \mathcal{D}$ into the language concept space. This is achieved by encoding $\mathbf{x}_i$ into an embedding vector $\mathbf{z}_i$, where each dimension corresponds to the significance of a concept in $\mathcal{A}$. The resulting embeddings capture the semantic alignment of each data point with the user-defined criteria. From this, an off-the-shelf clustering method such as k-means is applied to these embeddings to partition the dataset into clusters. This clustering process can be iteratively refined using 'supervision' from an unsupervised clustering quality metric such as the silhouette score. As discussed later, this information provide feedback to the Concept Generator to enable it to generate a more relevant set of concepts. This iterative feedback loop continues until convergence of the unsupervised clustering quality metrics, better enabling the clustering process to adhere to the user-defined query. By operating in the language concept space, ITGC provides enhanced flexibility and interpretability via a clustering that is closely aligned with human intent.

Fig. 2 illustrates an overview of ITGC. The design of ITGC focuses on two key components: (i) effectively encoding images into the language concept space to capture the user-defined criteria and (ii) implementing an unsupervised iterative search process that refines clustering through continuous feedback and evaluation. We present the details of these components in the following sections.

3.3 Encoding Images into Concepts

Encoding images into the concept space is a critical step in aligning the clustering process with the user-defined criteria. This involves transforming visual data into an interpretable representation that still reflects the user's criteria. We achieve this by leveraging pre-trained VLMs, *e.g.* CLIP [46] or LLaVA [34], to bridge the gap between vision and language.

For CLIP, the process begins by encoding each image $\mathbf{x}$ into the CLIP embedding space using its image encoder, $\mathbf{e}^v = \text{CLIP}_{\text{image}}(\mathbf{x}) \in \mathbb{R}^d$, here d is the latent dimension of CLIP. Simultaneously, the text descriptions of the LLM extracted concepts, generated by the Concept Generator, are encoded into the same embedding space using CLIP's text encoder $\mathbf{e}^{t_m} = \text{CLIP}_{\text{text}}(a_m) \in \mathbb{R}^d$. The similarity between the image embedding and each text embedding is then calculated, resulting in a vector $\mathbf{z}$, for each image, that denotes the relevance of each concept in an image. Specifically, the m-th dimension of $\mathbf{z}$ equals the dot product of $\mathbf{e}^v$ and $\mathbf{e}^{t_m}$, *i.e.* $z_m = \mathbf{e}^v \cdot \mathbf{e}^{t_m}$. This approach allows the clustering algorithm to operate in an interpretable space, aligned with the user's query, facilitating more interpretable clustering.

In the case of LLaVA, the extraction of each concept is formulated as a question, such as *"Please indicate the level of the presence of concept in the given image in a number between 0-10."* LLaVA processes these questions and provides numerical outputs that represent the presence of each concept in an image. These outputs are used to encode a vector for each image, capturing the semantic alignment with the user-defined criteria. This method enables a nuanced representation of concepts, enhancing the model's ability to adapt to diverse and complex datasets.

3.4 Concept Refinement via Iterative Search

The steps outlined previously result in a single set of language concepts and a set of per-image scores indicating how relevant the concepts are to each image. Next, inspired by [8], we introduce an iterative search process designed to refine the clustering by continuously enhancing the alignment between the generated concepts and the user-defined criteria. A single instruction/prompt to the Concept Generator LLM may not yield sufficiently expressive queries, as the LLM lacks specific knowledge about which types of prompts best align with the Concept Encoder. To address this, we propose an iterative search mechanism. We note, that while our approach is iterative, it still only requires a single text query from the user at the start.

At each iteration, the LLM, serving as the Concept Generator, has access to all previously generated sets of concepts and their corresponding evaluation scores. This historical context allows the LLM to assess the effectiveness of past concept sets and understand how the Concept Encoder responds to different prompts. The LLM is explicitly instructed to mutate and evolve these previous concepts, where the aim is for it to discover a set of concepts that yields improved unsupervised clustering performance. Importantly, the LLM is informed that its output will be used for encoding images, prompting it to generate concepts that correspond to visual features. This ensures that the generated concepts are interpretable and relevant for visual data. The precise prompt used for mutating and evolving the previous concepts is provided in Appendix C.3.

The iterative process is facilitated by a feedback loop that evaluates the quality of the clusters using an supervised clustering metric that does not require label information. This feedback guides the LLM to adjust and refine the concept set $\mathcal{A}$, allowing it to better align with the Concept Encoder ψ_e for the clustering task. Specifically, we denote the concept set used at the j-th iteration as $\mathcal{A}_j$, the feedback at the $j+1$-th iteration contains the concept sets $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_j\}$, and the silhouette scores for each of the concept sets $\{s_1, s_2, \ldots, s_j\}$, where s_j is the silhouette score for the j-th iteration. The silhouette score is an unsupervised metric used to evaluate the quality of a clustering algorithm [49]. By iteratively updating the concepts and re-projecting the images into the updated concept space, the ITGC converges towards a clustering solution that better satisfies the user's intent. This dynamic process not only enhances its adaptability to diverse datasets but also ensures that the clustering remains flexible and responsive to nuanced user preferences. Note, that ITGC does not require any training of the vision backbone model or the language model, as the search iterations are purely done in the language space using prompts. Thus, the search process of our model can be completed much faster compared to other methods that use gradient-based optimization [67, 68]. The overall pseudo-code of our method is provided in the appendix.

4 Experiments

4.1 Implementation Details

Models. We utilize existing pre-trained large models where possible. Specifically, for the Concept Generator, we use the open-sourced LLM from DeepSeek-V2 [33], as it demonstrates strong performance at a relatively low cost. We also include results using the closed-source GPT-4 [13] which further enhances the performance. We evaluate two different ways of encoding images into the concept space for our Concept Encoder; a CLIP or a LLaVA encoder. For the CLIP model, we use the CLIP-L/14 model, and for the LLaVA model, we use the LLaVA-v1.6-34B model. For the clustering process, we perform k-means clustering. Note, here we assume the number of clusters k is known before hand for us to perform k-means clustering. However, it is straightforward to use other clustering methods such as DBSCAN [9] which do not require the number of clusters a-priori (see Appendix).

Datasets. We evaluate our method on a diverse set of datasets to showcase its versatility. Specifically, two different dataset types are used; multi-criteria and fine-grained classification datasets. The multi-criteria clustering datasets each contain between two and four different high-level visual criteria, where each criteria induces a different clustering of the data.

Dataset	Type	Stanford 40 Actions [66]			Clevr-4 [57]				ImageNet-R [18]	
Criteria		Action	Location	Mood	Texture	Shape	Count	Color	Class	Rendition
CLIP [46]*	Zero-shot	78.3	80.4	80.1	34.2	74.5	74.5	33.5	75.8	33.5
SimGCD [61]†	Discovery	78.6	79.4	80.4	35.6	75.0	74.6	34.0	76.3	33.8
SCAN [54]	Clustering	39.7	35.9	25.0	28.5	54.1	30.6	31.5	60.1	18.9
AugDMC [67]	Clustering	65.6	45.6	40.2	28.9	50.2	35.6	27.6	45.3	25.4
IC|TC [29]	Text-guided	77.4	82.2	79.3	37.9	73.5	76.5	30.2	69.3	31.9
Multi-MaP [68]	Text-guided	79.5	83.5	81.5	34.5	75.6	74.8	35.6	76.4	34.0
Ours (w. LLaVA [34])	Text-guided	81.0	81.2	83.5	36.0	76.0	75.2	36.2	77.0	33.6
Ours (w. CLIP [46])	Text-guided	80.2	86.7	85.3	35.3	76.4	75.7	36.0	78.4	36.7

Table 1: **Results on multi-clustering datasets.** Our ITGC approach demonstrates superior performance compared to existing baselines. The zero-shot CLIP performance denoted with * has access to the ground truth class names, and the SimGCD method denoted with † has access to a small set of labeled examples. We can see that our method obtains stronger performance than these two methods despite their use of this additional information, indicating the effectiveness of our iterative search process.

For example, for the criteria 'shape', objects are grouped based on their shape, whereas the criteria 'color' results in a different grouping. We evaluate on the Stanford 40 Actions [66], ImageNet-R [18], Fruits-360 [12], Cards [5], and the larger-scaled dataset Clevr-4 [57]. These datasets present various challenges in discovering multiple independent clusterings structures within the datasets. For fine-grained evaluation, we use CUB-200 [60], Stanford Cars [28], and FGVC Aircraft [38]. The fine-grained nature of these datasets necessitates that a model can understand nuanced visual concepts and thus represent a challenging test case of our ability to generate interpretable spaces for clustering.

Evaluation. The evaluation protocol is similar to that used in category discovery [55], as we need to generate a match between the predicted clusterings from the model and the ground truth clustering. During evaluation, given the ground truth y^* and the predicted label $\hat{y}$, the accuracy (ACC) is calculated as $\text{ACC} = \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}(y_i^* = p(\hat{y}_i))$ where $M = |\mathcal{D}|$, and p is the optimal permutation that matches the predicted cluster assignments to the ground truth.

4.2 Results on Multi-clustering Datasets

In Tab. 1 we evaluate ITGC on several multi-clustering datasets. Compared to the baseline methods of zero-shot CLIP, which receives the cluster names as input, and SimGCD, which is provided with a set of labeled examples as input, we observe that ITGC consistently outperforms them under the different input criteria/queries. This indicates the effectiveness of our method in understanding the clustering criteria. Compared to methods that are specifically designed to handle text-guided and multi-criteria clustering (*e.g.* AugDMC, IC|TC, and Multi-MaP), we obtain an increase in performance. We attribute this performance improvement to the better alignment of the language concepts and the concept encoder. This concept alignment is a result of the iterative search procedure of our method, which enables it to automatically find the concepts more relevant to the user indicated clustering task. Previous methods either use one-off prompting [46], hand-designed augmentation priors [67, 68], or human-in-the-loop optimizations [29] that do not fully utilize the possible space of concepts for the clustering. Fruits360 and Cards are two datasets that have been used in previous literature on multi-criteria clustering. In Tab. 2 we report performance on these two datasets. We consider Fruits360 and Cards as easy datasets as the criteria for clustering is simple and the number of images in the datasets is also small. We can see that all models perform well on these two datasets, and that our proposed approach also obtains strong performance across the different clustering criteria.

Dataset	Type	Fruits-360 [12]		Cards [5]	
Criteria		Species	Color	Rank	Suits
CLIP [46]*	Zero-shot	92.5	91.4	96.7	93.0
SimGCD [61]†	Discovery	90.1	91.9	95.0	93.4
AugDMC [67]	Clustering	89.7	90.5	87.3	80.6
IC|TC [29]	Text-guided	90.4	89.2	93.4	92.5
Multi-MaP [68]	Text-guided	90.8	89.8	93.2	91.4
Ours (w. LLaVA [34])	Text-guided	90.4	90.5	94.6	91.0
Ours (w. CLIP [46])	Text-guided	91.6	90.1	95.8	93.7

Table 2: **Results on additional multi-criteria clustering datasets.** Here, * denotes access to ground truth class name and † denotes access to a small set of labeled examples.

Dataset	Type	CUB	Cars	Aircraft
CLIP [46]*	Zero-shot	51.4	60.1	25.6
SimGCD [61]†	Discovery	53.4	61.3	27.9
AugDMC [67]	Clustering	24.6	35.6	20.5
IC|TC [29]	Text-guided	48.9	62.1	31.4
Multi-MaP [68]	Text-guided	47.8	64.0	32.1
Ours (w. LLaVA [34])	Text-guided	52.7	63.5	28.7
Ours (w. CLIP [46])	Text-guided	57.8	65.4	34.5

Table 3: **Results on fine-grained classification datasets.** These fine-grained datasets require models to optimize for subtle visual details to separate the categories. Our ITGC approach shows significant improvements, discovering sufficient fine-grained structure.

4.3 Results on Fine-grained Datasets

Here we evaluate our method on fine-grained classification datasets which only contain a single clustering criteria (*i.e.* the semantic category). This requires models to be able to capture and compare the subtle visual difference within the image clusters to be able to distinguish fine-grained classes. We report performance on CUB-200 [60], Stanford Cars [28], and FGVC-Aircraft [38] in Tab. 3. We can see that these datasets pose a greater challenge for the models evaluated. This is evident by the fact that the CLIP baseline does not perform well on these datasets, despite being given the class names. SimGCD, a category discovery method, leverages a set of labeled examples, and thus demonstrates improved performance. If the signal for learning the clustering criteria is only applied on the coarse augmentation level, as in the case of AugDMC, we observe that it does learn to perform fine-grained clustering well. IC|TC and Multi-MaP also demonstrate a lower performance as they only optimize the search for the suitable clustering criteria in the text space. Our method demonstrate strong performance on these datasets, as it is able to better align the language concept space to the encoder space. In Fig. 3 we display qualitative results comparing the clustering results of our ITGC approach to the recent IC|TC [29] on CUB. In contrast, ITGC is able to generate more detailed concepts that are relevant to the objects of interest in the images.

4.4 Ablations

Search Iterations. In Tab. 4, we present the results of varying the number of search iterations. At the beginning of the search, the ITGC already results in a good clustering thanks to the image-text alignment ability of the CLIP (or LLaVA) model and the reasoning ability of the LLM. As we increase the number of search iterations, the Concept Generator begins to align more effectively with the Concept Encoder, and therefore results in improved performance. We can also see that for CLIP the performance saturates after a certain number of iterations, this indicates the search converges to a set of useful concepts for the given clustering criteria. The search process only involves prompting an LLM and performing forward passes through a pre-trained model. As a result, the computational cost of our search process is significantly lower than gradient-based optimization methods.

Encoder	# Itrs.	Clevr-4				CUB
Criteria		Texture	Shape	Count	Color	Species
CLIP	1	31.5	70.1	71.5	30.0	46.7
CLIP	5	34.2	72.6	73.8	33.5	50.3
CLIP	10	35.7	75.9	75.4	35.8	57.9
CLIP	15	35.3	76.4	75.7	36.0	57.8
LLaVA	1	30.1	68.2	66.2	27.8	43.5
LLaVA	5	32.4	70.1	68.3	31.2	44.3
LLaVA	10	35.1	74.5	73.5	34.5	48.6
LLaVA	15	36.0	76.0	75.2	36.2	52.7

Table 4: **Impact of search iterations.**

Encoder	Clevr-4				CUB
Criteria	Texture	Shape	Count	Color	Species
CLIP [46]	35.3	76.4	75.7	36.0	57.8
CoCa [69]	36.7	77.8	76.2	36.5	59.3
EVA-CLIP [52]	37.2	78.4	76.8	37.5	61.0
LLaVA-1.6 [34]	36.0	76.0	75.2	36.2	52.7
LLaVA-OneVision [30]	37.6	77.2	75.4	36.8	53.2
GPT-4o [43]	38.9	79.4	77.3	37.5	57.6

Table 5: **Impact of stronger models.**

Encoder	Auxiliary Info.	Clevr-4				CUB
Criteria		Texture	Shape	Count	Color	Species
CLIP	None	35.3	76.4	75.7	36.0	57.8
CLIP	Partial Labels	38.4	80.0	79.4	38.5	59.9
CLIP	Example Class Name	38.6	80.4	79.2	38.7	59.7
LLaVA	None	36.0	76.0	75.2	36.2	52.7
LLaVA	Partial Labels	37.1	77.9	78.4	37.5	53.4
LLaVA	Example Class Name	38.0	78.4	79.0	37.6	53.3

Table 6: **Impact of auxiliary information.**

Stronger Models. In Tab. 5, we explore the impact of using stronger models for both the Concept Generator and Encoder in ITGC. CoCa [69] improves upon CLIP [46] by adding an additional image captioning objective to enable the model to obtain better alignment, and EVA-CLIP [52] incorporates improved representation learning, optimization, and augmentation to achieve superior performance compared to previous CLIP models with a smaller training cost. Using stronger models than CLIP for the Concept Encoder improves performance. This demonstrates that the iterative search process in ITGC is able to utilize the better alignment of the image and text from the stronger models. LLaVA-OneVision [30] and GPT-4o [43] are more performant than the LLaVA-1.6 model used in previous experiments. LLaVA-OneVision utilizes more instruction tuning data to fine-tune the model and supports higher resolution inputs. It has also uses a larger LLM, QWen-2.0 72B [2], as its decoder. GPT-4o [43] is one of the most advanced multi-modal language model currently available for image understanding tasks. Our evaluation utilizing these stronger models demonstrates that ITGC is able to leverage the stronger models' capabilities to perform superior clustering that better matches the user's intent.

Using Auxiliary Information for Search. In our previous experiments, the search process is performed using only the feedback from the silhouette score derived from the current clustering. In Tab. 6, we present results where we use richer feedback signals to further assist the search process. 'Partial Labels' denotes the case where we allow the model to have access to the labels of 10% of all examples for the specified clustering criteria. The model can use the clustering accuracy on these examples as a more direct feedback signal for the iterative search process. 'Example Class Name' adds three example ground truth class names, for the given dataset and criteria, to the prompt for the Concept Generator so it can produce more relevant concepts for clustering. Note that these two forms of auxiliary information can be relatively cheap to obtain in practice, yet from our experiments, can greatly improve the performance of ITGC. This illustrates the flexibility of ITGC, demonstrating that it performs well given varying amounts of supervision.

5 Conclusion

We introduced ITGC, a new approach for text-guided clustering that uses an iterative search procedure to discover interpretable concepts for performing clustering to satisfy a user provided criteria. The search procedure is guided by an unsupervised clustering objective, and can generate interpretable concepts to help the user understand the clustering results. In our experiments, we demonstrate that ITGC is able to outperform existing baselines on the tasks of multi-criteria clustering as well as fine-grained image clustering.

Limitations. Despite our performance improvements over previous methods on the challenging real-world problem of text-guided clustering, limitations still exist. For example not all visual concepts can be conveyed easily via language, thus ITGC will not work as well in such cases. For some fine-grained concepts, visual examples may need to be used, which is not possible with ITGC.

Acknowledgements. This work was in part supported by a Royal Society Research Grant.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv:2309.16609*, 2023.

[3] Tossapon Boongoen and Natthakan Iam-On. Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review*, 2018.

[4] Kaidi Cao, Maria Brbić, and Jure Leskovec. Open-world semi-supervised learning. In *ICLR*, 2022.

[5] Cards Dataset. https://www.kaggle.com/datasets/gpiosenka/cards-image-datasetclassification, 2025. Accessed on 2025-03-03.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

[8] Mia Chiquier, Utkarsh Mall, and Carl Vondrick. Evolving interpretable visual classifiers with large language models. In *ECCV*, 2024.

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.

[10] Yixin Fei, Zhongkai Zhao, Siwei Yang, and Bingchen Zhao. Xcon: Learning with experts for fine-grained category discovery. In *BMVC*, 2022.

[11] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *ICCV*, 2021.

[12] Fruits-360 Dataset. https://www.kaggle.com/moltean/fruits, 2025. Accessed on 2025-03-03.

[13] GPT-4. https://openai.com/index/gpt-4, 2025. Accessed on 2025-03-03.

[14] Philip Haeusser, Johannes Plapp, Vladimir Golkov, Elie Aljalbout, and Daniel Cremers. Associative deep clustering: Training a classification network with no labels. In *GCPR*, 2019.

[15] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *ICCV*, 2019.

[16] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*, 2020.

[17] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *TPAMI*, 2021.

[18] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021.

[19] Juhua Hu and Jian Pei. Subspace multi-clustering: a review. *Knowledge and information systems*, 2018.

[20] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *CVPR*, 2020.

[21] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys*, 1999.

[22] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

[23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[24] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[25] Hyungmin Kim, Sungho Suh, Daehwan Kim, Daun Jeong, Hansang Cho, and Junmo Kim. Proxy anchor-based unsupervised learning for continuous generalized category discovery. In *ICCV*, 2023.

[26] Kun Ho Kim, Oisin Mac Aodha, and Pietro Perona. Context embedding networks. In *CVPR*, 2018.

[27] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *ICML*, 2020.

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV*, 2013.

[29] Sehyun Kwon, Jaeseung Park, Minkyu Kim, Jaewoong Cho, Ernest K Ryu, and Kangwook Lee. Image clustering conditioned on text criteria. In *ICLR*, 2024.

[30] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer. *TMLR*, 2025.

[31] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *AAAI*, 2021.

[32] Yunfan Li, Peng Hu, Dezhong Peng, Jiancheng Lv, Jianping Fan, and Xi Peng. Image clustering with external guidance. In *ICML*, 2024.

[33] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv:2405.04434*, 2024.

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2024.

[35] Mingxuan Liu, Subhankar Roy, Wenjing Li, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Democratizing fine-grained visual recognition with large language models. In *ICLR*, 2024.

[36] Mingxuan Liu, Zhun Zhong, Jun Li, Gianni Franchi, Subhankar Roy, and Elisa Ricci. Organizing unstructured image collections using natural language. *arXiv:2410.05217*, 2024.

[37] Shijie Ma, Fei Zhu, Zhun Zhong, Xu-Yao Zhang, and Cheng-Lin Liu. Active generalized category discovery. In *CVPR*, 2024.

[38] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013.

[39] Ioannis Maniadis Metaxas, Georgios Tzimiropoulos, and Ioannis Patras. Divclust: Controlling diversity in deep clustering. In *CVPR*, 2023.

[40] Lukas Miklautz, Dominik Mautz, Muzaffer Can Altinigneli, Christian Böhm, and Claudia Plant. Deep embedded non-redundant clustering. In *AAAI*, 2020.

[41] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.

[42] Ishan Nigam, Pavel Tokmakov, and Deva Ramanan. Towards latent attribute discovery from triplet similarities. In *ICCV*, 2019.

[43] OpenAI. Gpt-4o system card. *arXiv:2410.21276*, 2024.

[44] Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *CVPR*, 2023.

[45] Nan Pu, Wenjing Li, Xingyuan Ji, Yalan Qin, Nicu Sebe, and Zhun Zhong. Federated generalized category discovery. In *CVPR*, 2024.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[47] Sarah Rastegar, Mohammadreza Salehi, Yuki M Asano, Hazel Doughty, and Cees GM Snoek. Selex: Self-expertise in fine-grained generalized category discovery. In *ECCV*, 2024.

[48] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, S Yu Philip, and Lifang He. Deep clustering: A comprehensive survey. *Transactions on Neural Networks and Learning Systems*, 2024.

[49] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987.

[50] Subhankar Roy, Mingxuan Liu, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Class-incremental novel class discovery. In *ECCV*, 2022.

[51] Andreas Stephan, Lukas Miklautz, Kevin Sidak, Jan Philip Wahle, Bela Gipp, Claudia Plant, and Benjamin Roth. Text-guided image clustering. In *EACL*, 2024.

[52] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv:2303.15389*, 2023.

[53] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *EMNLP*, 2023.

[54] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, 2020.

[55] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *CVPR*, 2022.

[56] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need? In *ICLR*, 2022.

[57] Sagar Vaze, Andrea Vedaldi, and Andrew Zisserman. No representation rules them all in category discovery. In *NeurIPS*, 2023.

[58] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *CVPR*, 2017.

[59] Shaowei Wei, Jun Wang, Guoxian Yu, Carlotta Domeniconi, and Xiangliang Zhang. Multi-view multiple clusterings using deep matrix factorization. In *AAAI*, 2020.

[60] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010.

[61] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *ICCV*, 2023.

[62] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

[63] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016.

[64] Muli Yang, Liancheng Wang, Cheng Deng, and Hanwang Zhang. Bootstrap your own prior: Towards distribution-agnostic novel class discovery. In *CVPR*, 2023.

[65] Yan Yang and Hao Wang. Multi-view clustering: A survey. *Big data mining and analytics*, 2018.

[66] 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. In *ICCV*, 2011.

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

[68] Jiawei Yao, Qi Qian, and Juhua Hu. Multi-modal proxy learning towards personalized visual multiple clustering. In *CVPR*, 2024.

[69] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. 2022.

[70] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[71] Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Khan. Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *CVPR*, 2023.

[72] Xinwei Zhang, Jianwen Jiang, Yutong Feng, Zhi-Fan Wu, Xibin Zhao, Hai Wan, Mingqian Tang, Rong Jin, and Yue Gao. Grow and merge: A unified framework for continuous categories discovery. In *NeurIPS*, 2022.

[73] Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. In *NeurIPS*, 2021.

[74] Bingchen Zhao and Oisin Mac Aodha. Incremental generalized category discovery. In *ICCV*, 2023.

[75] Bingchen Zhao, Nico Lang, Serge Belongie, and Oisin Mac Aodha. Labeled data selection for category discovery. In *ECCV*, 2024.

[76] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, 2021.