

Self-Evolving Visual Concept Library using Vision-Language Critics

Atharva Sehgal^{1*} Patrick Yuan² Ziniu Hu³ Yisong Yue³ Jennifer J. Sun² Swarat Chaudhuri¹

¹University of Texas at Austin ²Cornell University ³California Institute of Technology

Abstract

We study the problem of building a visual concept library for visual recognition. Building effective visual concept libraries is challenging, as manual definition is labor-intensive, while relying solely on LLMs for concept generation can result in concepts that lack discriminative power or fail to account for the complex interactions between them. Our approach, ESCHER, takes a library learning perspective to iteratively discover and improve visual concepts. ESCHER uses a vision-language model (VLM) as a critic to iteratively refine the concept library, including accounting for interactions between concepts and how they affect downstream classifiers. By leveraging the in-context learning abilities of LLMs and the history of performance using various concepts, ESCHER dynamically improves its concept generation strategy based on the VLM critic’s feedback. Finally, ESCHER does not require any human annotations, and is thus an automated plug-and-play framework. We empirically demonstrate the ability of ESCHER to learn a concept library for zero-shot, few-shot, and fine-tuning visual classification tasks. This work represents, to our knowledge, the first application of concept library learning to real-world visual tasks.

1. Introduction

How do humans recognize different visual categories? Consider the example in Figure 1: while easily recognizable as a pastry, distinguishing between a “donut” and a “beignet” requires understanding visual concepts such as circular shape with a hole for the donut, puffy texture, and the presence of powdered sugar specifically on the beignet. These visual concepts, including shape, texture, and the presence or absence of specific features, enable us to make distinctions between objects. Concept-bottleneck visual recognition [19, 30, 31] aims to leverage these discriminative visual concepts, to enable vision systems to more accurately recognize a wider range of classes. Here, we study general

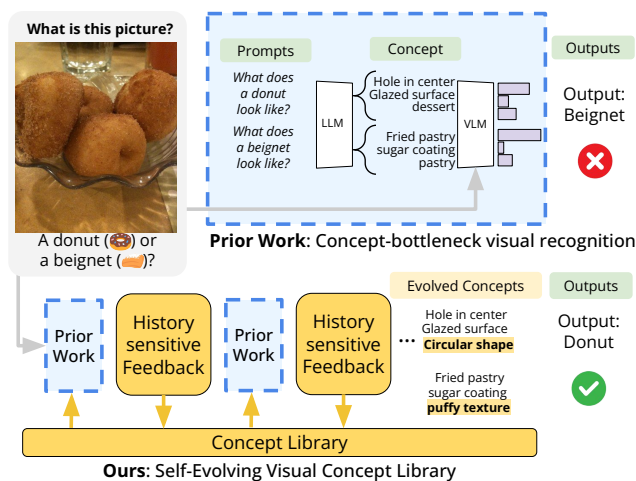


Figure 1. An overview of ESCHER. Prior work: concept-bottleneck visual recognition aims to leverage discriminative visual concepts to enable more accurate object classification. Ours: ESCHER is an approach for iteratively evolving a visual concept library using feedback from a VLM critic, to discover more effective visual concepts.

approaches for improving concept-bottleneck visual recognition systems by evolving visual concept libraries to find more effective concepts.

Existing concept-bottleneck visual recognition systems typically leverage a Large Language Model (LLM) to generate a set of potential visual concepts relevant to the task, then use a Vision-Language Model (VLM) to make predictions from these concepts. This process potentially improves both interpretability and accuracy for classification [19, 30, 31]. However, existing methods face limitations: manually defined concepts are labor-intensive, and LLM-generated concepts can be inaccurate or fail to account for interactions between them. We need more effective methods to construct and refine visual concepts. One promising approach to improve visual concept learning is to leverage library learning [6, 9, 10], which focuses on building a reusable collection of components. While library learning has shown success in domains that are naturally

*Correspondence to atharvas@utexas.edu. Artifacts available at <https://trishullab.github.io/escher-web>

symbolically decomposable (e.g., equation learning [9]), it is not well-explored for visual concept learning. Our key insight is that library learning complements visual concept learning by providing a structured and evolving repository of concepts that is more effective for visual recognition.

To achieve this, we introduce **ESCHER**, a novel self-evolving framework to automatically discover and refine a library of visual concepts. **ESCHER** employs an iterative algorithm where a VLM acts as a critic, providing feedback on the effectiveness of concepts generated by an LLM. Specifically, the VLM evaluates the similarity between each image and its associated concepts compared to other images. This evaluation, captured in a contrastive score, serves as a feedback signal to guide the LLM in refining its generated concepts. Furthermore, **ESCHER** provides the history of concepts and feedbacks to the LLM, enabling the LLM to effectively learn from its past performance and improving its proposals over time. Through this iterative process, **ESCHER** produces a set of concepts that are both accurate and highly informative for the VLM, enabling it to make more effective predictions.

Our approach offers several key advantages. First, it is broadly applicable and complements a range of existing concept-bottleneck visual recognition frameworks, including those designed for zero-shot, few-shot, and fine-tuned settings. This adaptability ensures that as LLMs, VLMs, and visual concept learning frameworks continue to evolve, **ESCHER** remains relevant and applicable to emerging techniques. Second, **ESCHER** requires no human annotations or labeled datasets, making it a plug-and-play solution for various visual recognition tasks. Finally, the iterative approach of **ESCHER** leverages the in-context learning capabilities of LLMs, allowing them to learn from their concept history and generate increasingly effective concept concepts. This iterative refinement process ensures that the concept library continuously adapts and improves, leading to more accurate and discriminative visual representations.

To summarize, our contributions are:

- We present **ESCHER**, a novel VLM- and LLM-based framework for self-evolving visual concept libraries. Our method does not require human-labeled data and can improve the quality of the learned concepts via an open-ended learning loop.
- We develop an iterative concept refinement algorithm, leveraging the both the ability of VLMs to act as a critic and the ability of LLMs to incorporate history, to improve visual concepts based on past performance.
- We demonstrate that **ESCHER** is complementary to a range of different state-of-the-art baselines, and our learned concept library improves performance across zero-shot, few-shot, and fine-tuned image classification settings.

2. Related Work

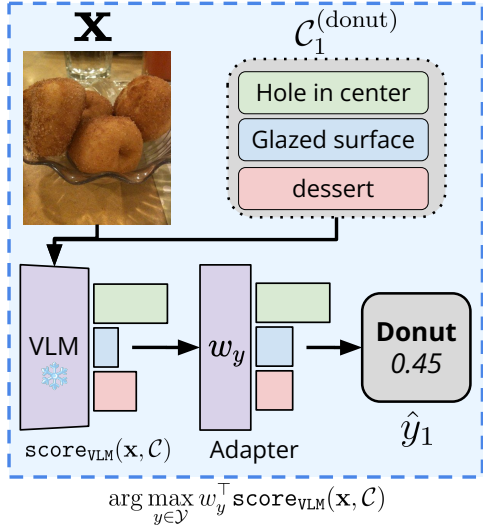
Vision-language models. VLMs have emerged as powerful tools for a wide array of visual tasks, from visual question answering to zero-shot image classification [2, 18, 23]. These models are typically trained on large datasets of image-text pairs, enabling them to reason about the relationship between these modalities. Some VLMs [11, 23, 32] are trained using contrastive learning objectives to learn embedding spaces aligned between image and text representations. These models have shown promising results for zero-shot and few-shot classification.

While VLMs have shown remarkable capabilities, they have limitations that motivate exploring alternative approaches, such as concept-bottleneck models. One limitation is their difficulty in perceiving and reasoning about fine-grained visual concepts. For example, recent work has shown that VLMs can struggle to distinguish between subtle visual differences [26]. As such, despite recent advances, these models can still often fail with accurate visual recognition. The standard zero-shot method does not provide any intermediate understanding or explanation of the model’s reasoning process [19]. These challenges highlight the need for more interpretable and controllable approaches to visual recognition, such as those offered by concept-bottleneck methods.

Concept-bottleneck Visual Recognition Models. Our work builds upon recent work on concept-bottleneck models [19, 22, 24, 30, 31, 33], which first identifies relevant concepts using a vision language model (VLM), and then uses those concepts to make a prediction. Compared to directly querying a VLM, this approach has advantages including interpretability, (if the concepts are interpretable) as well as stronger accuracy (if the concepts are useful in capturing the classification task). Similar ideas predate the rise of VLMs [12, 17], where a so-called concept bottleneck is built into neural network architectures, and concepts are learned via end-to-end training. Recent algorithms for learning concept-bottleneck models with VLMs generally fall within two categories:

Non-parametric Algorithms: These methods focus on improving the visual concepts by employing non-parametric optimization techniques. One increasingly common approach is to use zero-shot queries to an LLM to select a list of concepts for each class that is useful for classification [19]. The aggregated score for a class is the mean over the scores of the selected concepts. [5] follows the LLM induction paradigm to initialize class concepts but, borrowing ideas from genetic mutation, repeatedly queries a finetuned LLM to generate new concepts and new concept selections for each class – using binary classification loss to rate each mutation. This mutation process focuses

Concept-Bottleneck Visual Recognition Learning w_y in isolation



ESCHER: Self-Evolving Visual Concept Library Simultaneously Learning w_y and \mathcal{C}

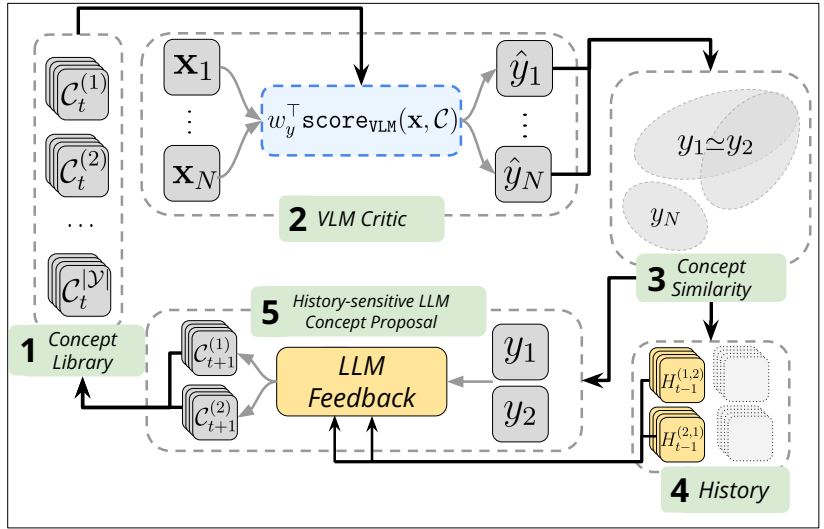


Figure 2. **(Left)** Existing work on concept-bottleneck visual recognition, where a VLM scores a set of concepts to perform classification. The classification is based on the class with the maximum concept scores. **(Right)** ESCHER. (1) ESCHER follows previous work [19] in instantiating a set of concepts for each class using an LLM. (2) It initializes a concept-bottleneck model and collect the predictions for a classification dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ (labels optional). (3) A concept similarity heuristic identifies frequently confused classes. (4) A history bank then stores relevant information to guide (5) the LLM sampling procedure for improved concepts that disambiguate these classes. The new concepts are integrated into the next iteration.

on each class in isolation and must be repeated for each class, which proves to be impractical for datasets with more than 20 classes. ESCHER, instead, reasons jointly about all classes and focuses on only those classes that are underperforming. As ESCHER is agnostic to the choice of VLM and the number of samples needed for training, it is technically possible to integrate the llm-mutate framework within ESCHER to maximize classification performance.

Parametric Algorithms: These methods focus on improving the performance of a visual concept bottleneck classifier by training parametric adapters on top of the scores output by a frozen VLM model. These models may also subsample concepts from the concept library. These architectures generally consist of a ‘concept bottleneck’ [30, 31], with additional learning required such as a linear probing adapter or additional finetuning [24]. ESCHER is agnostic to the choice of concept tuning method, and focuses on using such methods to guide the concept discovery loop. One can ‘plug’ ESCHER into any of these other works and observe a performance improvement while retaining the key characteristics of the type of architecture used.

Library Learning. Library learning is an emerging direction of program synthesis that aims to automate the construction of reusable components (libraries) for program generation. This is often framed as a hierarchical Bayesian

optimization problem, where the goal is to simultaneously learn the library of components and the optimal way to combine them to solve a given task [6, 7, 16, 29]. While library learning has shown promise in domains like equation learning [9], its application to visual concept learning presents unique challenges.

Unlike domains that are naturally symbolically decomposable, visual concepts often exhibit complex and subtle relationships that are difficult to capture with traditional library learning techniques. Moreover, the space of potential visual concepts is vast and diverse, making it challenging to design effective search strategies. Our work addresses these challenges by introducing a novel library learning approach designed for visual concept discovery. By leveraging a VLM as a critic and incorporating class resolution history into the LLM, our method can effectively explore the space of visual concepts and construct a dynamic library that adapts to the specific needs of the visual recognition task.

3. Problem Formulation

Concept-Based Visual Recognition Our work is rooted in the emerging area of concept-bottleneck visual recognition [19, 30, 31] (Figure 2). Given an image \mathbf{x} , a set of concepts \mathcal{C} , and a label y , the basic setup is to use a vi-

sion language model (VLM) to score the likelihood of each concept $c \in \mathcal{C}$ for image \mathbf{x} , denoted as $\text{score}_{\text{VLM}}(\mathbf{x}, c)$. We also write the vector of scores over all concepts as $\text{score}_{\text{VLM}}(\mathbf{x}, \mathcal{C})$. Afterward, the aggregate score of a label y for image \mathbf{x} is a weighted sum over the concept scores: $f(\mathbf{x}, y) = w_y^T \text{score}_{\text{VLM}}(\mathbf{x}, \mathcal{C})$, where w_y can be either a learned or fixed parameter vector.¹ Finally, classification over a fixed label set \mathcal{Y} is performed by choosing the class that maximizes image-concept similarity.

$$y^* = \arg \max_{y \in \mathcal{Y}} w_y^T \text{score}_{\text{VLM}}(\mathbf{x}, \mathcal{C}) \quad (1)$$

This setup encompasses the bulk of recent work in concept-bottleneck models for image classification [19, 30, 31] and helps considerably in fine-grained and out-of-distribution classification scenarios. For instance, while a class $y = \text{“SpaceX Starship”}$ may not have been seen during CLIP training, we can construct a reasonably accurate and interpretable rocket classifier by aggregating the likelihoods over the feature set: $\{f(\mathbf{x}|c_y) \mid c_y \in [\text{‘Stainless Steel Rocket’}, \text{‘Grid Fins’}, \text{‘Space X logo’}]\}$. Selecting this feature set can be automated using a foundation model with access to an external database [25].

Typically, a concept-bottleneck model is developed by predefining a fixed set of concepts \mathcal{C} , and then finetuning the weight matrix w_y for each class. The optimization objective is then chosen to maximize the performance of the model and identify an interpretable set of concepts. A Bayesian formulation is presented in Eq 3.

In contrast, scientists – when confronted with a new domain – rarely rely on a static set of concepts. The first reaction of a scientist is to learn more about the domain and expand their conceptual knowledge base. This newly gained knowledge is subsequently used to structurally discriminate between classes. For instance, even a trained ecologist might struggle to differentiate a Northern Curly-tailed Lizard from a Florida Scrub Lizard due to lack of prior knowledge, while a herpetologist can rely on their knowledge of lizard physiology to identify the correct characteristic feature difference²

Visual Recognition with Latent Concept Libraries. We model this evolving set of concepts as textual descriptions drawn from a latent concept library. We frame the relationship between the latent concept library and the classification model as a Hierarchical Bayesian model consisting of (i) a prior $p(\mathcal{C})$ representing the natural distribution over concept libraries; (ii) a model $p_{\mathcal{C}}(w_y|\mathcal{C})$ that quantifies the likelihood of assigning open-vocabulary classes for a given concept library; and (iii) an evaluation function $p(\mathcal{D}|\mathcal{C}) :=$

$\arg \max_{y \in \mathcal{Y}} w_y^T \text{score}_{\text{VLM}}(\mathbf{x}, \mathcal{C})$ which grounds the performance of a concept library using an image classification dataset (\mathcal{D}) and a VLM-based recognition engine. We assume that the distributions $p(\mathcal{C})$ and $p(w_y|\mathcal{C})$ can be approximated using LLMs. That is, we can prompt an LLM to generate interesting concepts, and we can prompt an LLM to generate and discover new concepts that adhere to an open-vocabulary category. We also assume that the VLM-based visual recognition engine is well calibrated for confidence estimation. We now pose the problem of visual recognition with latent concept libraries as one of simultaneously inducing an optimal set of concepts and an optimal concept-bottleneck visual reasoning model:

$$\begin{aligned} w_y^*, \mathcal{C}^* &= \arg \max_{w_y, \mathcal{C}} p(w_y, \mathcal{C}|\mathcal{D}) \\ &= \arg \max_{w_y, \mathcal{C}} \underbrace{p(\mathcal{D}|w_y)}_{\text{CBM training}} \cdot \underbrace{p(w_y|\mathcal{C})}_{\text{By LLM}} \cdot \underbrace{p(\mathcal{C})}_{\text{By LLM}} \end{aligned} \quad (2)$$

4. Method

ESCHER performs a two stage evolution over the natural-language concepts and the weight matrix assignment. The two stages follow an alternating maximization strategy, as illustrated in Figure 2: (1) *Concept Bottleneck Optimization*: We fix a set of concepts and learn a concept-bottleneck model that maximizes the fitness to the dataset (Fig. 2, Left). (2) *History-sensitive concept evolution*: We leverage the best model to identify classes that appear to be confused and sample new concepts to resolve the confusion (Fig. 2, Right).

The rest of this section first describes the classical concept-bottleneck model maximization strategies. Then, we discuss common heuristics for identifying confused classes. Finally, we show how the classes can be disambiguated by sampling new concepts conditioned on feedback derived from previous evolutions.

Concept Bottleneck Optimization. Concept-bottleneck models [13] generate their predictions by learning to linearly combine the intermediate predictions over a fixed set of interpretable concepts. This yields a high-performing yet interpretable classifier that can be used for downstream classification tasks. We focus on two paradigms within this field for optimizing the adapter weight matrix.

Zero shot maximization. In this setting, the adapter weights are instantiated by an LLM. Intuitively, the adapter will take the form of a block diagonal matrix, where each block represents the concepts selected by the LLM for a particular class and each element in the block is assigned the uniform weight $1/|c_y|$, where c_y is the set of concepts the LLM generates for a particular label y . As no labels are needed, this paradigm generates extremely flexible classifiers. However, the efficacy of the concepts is deeply tied to the backbone VLMs ability to score fine-grained concepts.

¹Some prior work uses a uniform vector for w_y [19].

²<https://www.inaturalist.org/observations/1970016>

Algorithm 1 Pseudocode for ESCHER. ESCHER takes as input a set of open vocabulary categories \mathcal{V} , a dataset of images \mathcal{D} (labels optional), a pretrained vision-language model θ_{VLM} , an optional adapter w (We drop the $\cdot_{\mathcal{V}}$ subscript for readability), and three hyperparameters: the number of iterations T , the decay rate γ for repeated categories, and the number of pairs to evolve K . ESCHER outputs two artifacts: the adapter parameters after T iterations, w_T , and the corresponding evolved library of interpretable concepts \mathcal{C}_T .

```

1: function ESCHER( $\mathcal{V}, \mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N, \theta_{\text{VLM}}, T, \gamma, K$ )
2:    $\mathcal{C}_0 \leftarrow \text{INITCONCEPTS}(\mathcal{V})$   $\triangleright$  Initialize concepts via
   zero-shot LLM queries
3:    $H_0^{(i,j)} \leftarrow \text{INITHISTORY}(\mathcal{V}, T)$   $\triangleright$  Track concepts and
   feedback for each class pair per iteration
4:   for  $t$  in  $\text{range}(T)$  do
5:      $w_t^* \leftarrow \text{fit}(w_t, \mathcal{C}_t, \mathcal{D}, \theta_{\text{VLM}})$ 
6:      $\hat{\mathbf{y}} \leftarrow \text{evaluate}(\mathcal{D}, \theta_{\text{VLM}}, w_t^*)$ 
7:      $\{r_{ij}\}_{i,j=1}^{|\mathcal{V}|^2} \leftarrow \text{CALCULATESIMILARITY}(\hat{\mathbf{y}})$ 
8:      $H_t^{(i,j)} \leftarrow \text{UPDATEHISTORY}(\{r_{ij}\})$   $\triangleright$  Store the
     similarity of  $y_i$  and  $y_j$  for iteration  $t$ 
9:      $s_{ij} \leftarrow \text{COMPUTESAMPLEPROB}(r_{ij}, H_{[1:t]}^{(i,j)}, \gamma)$ 
10:     $\{r_{ij}\}_1^K \leftarrow \text{subsample}(\{r_{ij}\}, s_{ij})$ 
11:    for each  $(i, j) \in \{r_{ij}\}_1^K$  do
12:       $\hat{c}^{(i)}, \hat{c}^{(j)} \leftarrow \text{CONCEPTEVOL}(y_i, y_j, \mathcal{C}_t^{(i)}, \mathcal{C}_t^{(j)}, H_{[1:t]}^{(i,j)})$ 
13:       $\mathcal{C}_{t+1}^{(i)} \leftarrow \mathcal{C}_t^{(i)} \cup \{\hat{c}^{(i)}\}$ 
14:       $\mathcal{C}_{t+1}^{(j)} \leftarrow \mathcal{C}_t^{(j)} \cup \{\hat{c}^{(j)}\}$ 
15:       $H_{t+1}^{(i,j)} \leftarrow \text{UPDATEHISTORY}(\mathcal{C}_{t+1}^{(i)}, \mathcal{C}_{t+1}^{(j)})$   $\triangleright$  Store the
      updated concepts for iteration  $t + 1$ 
16:     $w_{t+1} \leftarrow w_t^*$ 
17:     $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_t$ 
18: return  $w_T, \mathcal{C}_T$ 

```

Few shot / Fine-tuned maximization. Under this paradigm, the adapter weights are instantiated as a learnable linear layer of shape $\mathbb{R}^{|\mathcal{C}| \times |\mathcal{V}|}$. As the set of concepts can grow very large, most approaches subsample the set of concepts as well. The linear layer is trained with cross-entropy loss, often with various regularizers [30]. In the few-shot setting, the number of images per class is fixed while in the fine-tuned setting, no restriction is placed on the number of images. This adapter training approach also overcomes the inherent weakness of the zero shot setting, as concepts that do not add to the performance of the model can be down-weighted or ignored by the linear adapter.

Heuristics for disambiguation. After maximizing the model for the given set of concepts, the CALCULATESIMILARITY function identifies classes that are confused with each other. This function takes as input a matrix of scores for the dataset of images $\hat{\mathbf{y}} \in \mathbb{R}^{N \times |\mathcal{V}|}$. Each value in $\hat{\mathbf{y}}$ indicates the image-class similarity. We leverage this ma-

trix of scores to identify classes that are confused with each other across the images in the dataset. $\{r_{ij}\}_{i,j=1}^{|\mathcal{V}|^2}$ denotes the list of all possible confusion scores. ESCHER’s modular framework allows us to admit any heuristics to model this confusion score. We list several heuristics to measure this disambiguation signal that perform well empirically. Additional studies are presented in § 8.4, Figure 4, and Figure 5.

- *Top-k Confusion* Top-k confusion is computed by sorting each row in descending order of score and keeping the top k class predictions. That is: $r_{ij} = \text{confusion_mat}(\text{sort}(\hat{\mathbf{y}}[:, :k]))$. Consult Figure 4 for more insight into how this heuristic works.
- *Correlation*: We compute Pearson’s correlation between each class $r_{ij} = \text{corrcoef}(\hat{\mathbf{y}}^\top)$. Pearson’s correlation measures linear correlation between two classes (y_i and y_j) with -1 denoting maximal negative correlation and 1 denoting maximal positive correlation. This is visualized in Figure 5. The hyperparameter k differs from the hyperparameter K which tracks the number of classes to disambiguate.
- *Agglomerative Clustering*: Identifying classes with similar response signals can be viewed as an unsupervised clustering problem over the columns of $\hat{\mathbf{y}}$. The number of clusters is kept as a tunable hyperparameter, and choose the values of r_{ij} by greedily bottom-up traversing the dendrogram.
- *Confusion Matrix*: ESCHER assumes no access to labels during evolution. However, if labels are provided, we can construct a confusion matrix to identify classes that are confused for each other.

History-sensitive concept evolution. Each value in $\{r_{ij}\}_1^K$ represents two classes that the model is unable to discriminate between. This suggests that the model lacks the correct discriminative concepts in the concept space \mathcal{C} to conceptually separate y_i and y_j . Now we use the CONCEPTEVOLUTION function, which uses a zero-shot prompt to extract a list of natural language concepts for both classes (\hat{c}_i, \hat{c}_j) that add additional concepts useful for disambiguating y_i and y_j to the concept library.

This task requires good reasoning skills, so we provide the model with a scratchpad to enhance the model’s reasoning abilities [21]. More details are presented in § 8.5.

This process is repeated for each pair of confused concepts. However, the CALCULATESIMILARITY function generates $|\mathcal{V}|^2$ pairs in each iteration, and processing all such pairs is extremely inefficient for real-world datasets. To increase feedback generation efficiency, we only sample the top K pairs from a random distribution weighted by the confusion coefficient score and a penalty for repeat confusions which increases exponentially, controlled by a decay parameter γ . This is implemented as

$\text{COMPUTESAMPLEPROB}(r_{ij}, H_{[1:t]}^{(i,j)}, \gamma)$ where the size of $H_{[1:t]}^{(i,j)}$ indicates the number of times the $(i, j)^{\text{th}}$ classes have been confused for each other. The γ parameter helps guard against classes that are often misclassified due to failures in the backbone ViT model.

Additionally, each pairwise disambiguation can cause collisions with other classes in later iterations, which makes it likely that some class pair will need multiple rounds of feedback. If we generate feedback in isolation, the model is likely to regurgitate previously proposed features. To ensure that each new round of feedback generates novel and interesting concepts, we borrow ideas from the program synthesis literature [1] and append past ‘execution traces’ to the model query. This history-conditioned prompt is presented in Figure 6. We maintain a history of past evolutions for each pair (y_i, y_j) along with the similarity score derived from the VLM critic’s score of the modified concepts $c_i \cup \hat{c}_i$ and $c_j \cup \hat{c}_j$. Concretely, `INITIALIZEHISTORY` instantiates this datastructure, and `UPDATEHISTORY` updates the relevant fields in each iteration (the new concepts added to class i and j in iteration t and the class confusion score in iteration $t + 1$ after incorporating feedback). These functions are explored in more detail in § 8.10.

5. Experiments

ESCHER is a meta-algorithm that aims to enhance the performance of concept-bottlenecked models (CBMs) by learning a library of concepts using an alternating maximization loop. Such CBMs operate in diverse data regimes, with some models requiring no human labels [19] and others requiring a fully annotated classification dataset ([30]). Our experiments focus on studying whether ESCHER can improve the performance of such preexisting algorithms that are characteristic of the data regime they operate in, with no additional modifications (§5.1, §5.2, and §5.3)

In addition, §5.4 presents an ablation of ESCHER’s library learning component, and §5.5 explores ESCHER’s performance under various LLM and VLM backbones. More studies are presented in the Appendix (§ 8).

Datasets. We demonstrate the effectiveness of ESCHER on seven fine-grained and general purpose classification datasets that generally fall within three categories: fine-grained classification datasets with scientific applications (NABirds [27] and CUB [28]), fine-grained classification datasets with general purpose applications (Food-101 [3], Stanford Cars [14], Flowers-102 [20]), and general purpose categorization datasets (CIFAR100 [15]).

Evaluation. We extend three concept-bottleneck visual classification models with publicly available and reproducible codebases at the time of writing: LM4CV (fine-

tuned adapter), LaBO (few shot adapter), and Classify by descriptions (zero-shot adapter) [19, 30, 31]. We follow previous work in using Top-1 accuracy to evaluate the performance of the CBM model on the test set of the classification dataset.

Methodology. All experiments begin with an initial set of concepts generated with a backbone LLM (gpt-3.5-turbo-0125 [4]). A CBM conditioned on these concepts produces logits for the images in the validation set, and a heuristic identifies classes that are confused for each other. The specific heuristic and hyperparameters vary depending on the task and underlying algorithm, and grid-searching for such hyperparameters proves to be practical and effective. We use a generic history-conditioned prompt presented in Fig. 6. We use a ViT-L/14 CLIP model [23] as the vision backbone for all experiments unless otherwise clarified.

5.1. Comparison against fine-tuned baselines

	LM4CV	LM4CV+ESCHER
CIFAR-100	84.48	89.63
CUB-200-2011	63.26	83.17
Food101	94.77	94.90
NABirds	76.58	78.21
Oxford Flowers	94.80	96.86
Oxford IIIT Pets	92.50	92.86
Stanford Cars	86.84	93.76

Table 1. Top-1 accuracy of LM4CV [30] and LM4CV evolved with ESCHER on multiple fine-grained classification problems. ESCHER improves upon LM4CV’s performance in all datasets while utilizing no extra human annotations.

Setup. For each dataset, we generate an initial set of concepts using queries to gpt-3.5-turbo and a generic visual concept learning prompt. We use LM4CV’s suggested hyperparameters wherever possible. We run 60 iterations of ESCHER. Similarity is computed using the Top-k confusion metric ($k = 3$), only the top 50 pairwise confusions are evolved, and decay rate γ is set to $1/30$ (i.e.: after 30 repeated evolution calls, the value drops to half of the original). LM4CV and LM4CV+Escher are trained for the same number of steps per iteration with the same batch size and learning rate.

Observations. Our observations are presented in Table 1. We draw three observations from this experiment. First, LM4CV + ESCHER achieves a higher Top-1 accuracy than vanilla LM4CV on all datasets, suggesting that learning a

concept library is an effective axis of improvement. Second, LM4CV+ESCHER significantly improves the performance of LM4CV on datasets where the initial accuracy is low, and finally, LM4CV+ESCHER did not require any human provided information to achieve this result.

5.2. Comparison against few-shot baselines

Dataset	8 shot		16 shot	
	LaBO	LaBO+ESCHER	LaBO	LaBO+ESCHER
CIFAR100	74.23	73.62	77.67	77.23
CUB	72.78	73.37	78.75	78.79
Food101	87.02	86.10	88.49	88.50
Oxford Flowers	95.66	95.37	97.69	97.80
Stanford Cars	75.07	75.99	81.48	82.56

Table 2. Top-1 accuracy for LaBO and LaBO evolved with ESCHER on multiple fine-grained classification datasets in a few-shot learning setting. LaBO benefits from ESCHER’s library guidance in the 16 shot setting more than in the 8 shot setting. We observe mixed results on ESCHER’s efficacy in the few shot domain.

Setup. Our setup for LaBO follows the same setup as that of LM4CV. We evaluate all datasets for 8 shot and 16 shot. This setup necessitates a balanced set of images for each class. We drop NABirds from this evaluation, as some NABirds classes contain as low as 3 images per class. We keep the same hyperparameters as our LM4CV experiments but do not use a decay rate as the number of repeat classifications is generally unproblematic.

Observations. We observe mixed results in few-shot evaluation, with similar performance compared to the baseline on all datasets except for Stanford Cars, which showed modest improvement, and CIFAR100, which showed very modest deterioration (Table 2). In general, LaBO+ESCHER performs considerably better in the 16-shot setting compared to the 8-shot setting. We hypothesize these mixed results are induced by poor calibration of LaBO’s CBM’s and as a result of model overfitting on the few available labels for each class.

5.3. Comparison against zero-shot baselines

Setup. Our zero shot setup compares against Classify by Descriptions. We sample all concepts from gpt-3.5-turbo. We use the same hyperparameters as LM4CV experiments, except for setting the decay rate to 50. We continue to use the ViT-L-14 backbone mode.

Observations. Our observations are presented in Table 3. Overall, we find that CbD’s performance improves consistently for all datasets when we evolve the concepts with ESCHER. The relative improvement in performance is less

Dataset	CLIP	CbD	CbD+ESCHER
CIFAR-100	73.30	76.20	77.80
CUB-200-2011	64.83	62.00	63.33
Food101	92.51	93.11	93.58
NABirds	53.53	53.61	54.30
Oxford Flowers	74.51	79.41	81.37
Stanford Cars	74.53	75.65	77.14

Table 3. Top-1 accuracy of CLIP (ViT-L/14) [23], Classify by Descriptions (CbD) [19], and CbD evolved with ESCHER on multiple fine-grained classification datasets in a zero-shot learning setting. CbD+ESCHER improves upon CbD’s performance in all datasets.

than that in LM4CV as the backbone (CLIP ViT-L-14) model’s output scores are less calibrated than those produced by the finetuned adapter, which leads to noisier iterations.

5.4. Library Learning Ablation

Dataset	LM4CV	LM4CV	LM4CV
		+ Many Concepts	+ ESCHER
CIFAR-100	84.48	86.91	89.63
CUB-200-2011	63.26	66.09	83.17
Food101	94.77	94.77	94.90
NABirds	76.58	76.28	78.21
Oxford Flowers	94.80	94.51	96.86
Oxford IIIT Pets	92.50	92.02	92.86
Stanford Cars	86.84	86.84	93.76

Table 4. Top-1 accuracy of an ablation of ESCHER’s library learning component. For LM4CV, we replace the concepts learned with library learning with an equal number of concepts sampled from an LLM. We find that concepts evolved with ESCHER still outperform naively sampling more concepts – suggesting that feedback from a VLM critic is essential for LM4CV+ESCHER’s performance.

ESCHER focuses on using zero shot disambiguation queries to maximize the performance of a concept library. To do this, ESCHER makes asymmetrically more calls to a zero-shot LLM model than the baseline (which only samples concepts once). To verify that ESCHER’s performance is not simply a result of more concepts or more number of LLM calls, we rerun LM4CV with three times more concepts per class than the initial set of concepts (for a total of 3810 concepts). These concepts are sampled with the same LLM backbone we use in ESCHER. This represents an ablation on the Library learning component used in ESCHER.

Results are highlighted in Table 4. We find that LM4CV’s performance does not significantly improve even given the same number of LLM sampled concepts as LM4CV+ESCHER. This suggests that (1) the library learning component is essential for concept evolution and (2) sampling concepts without taking into account feedback

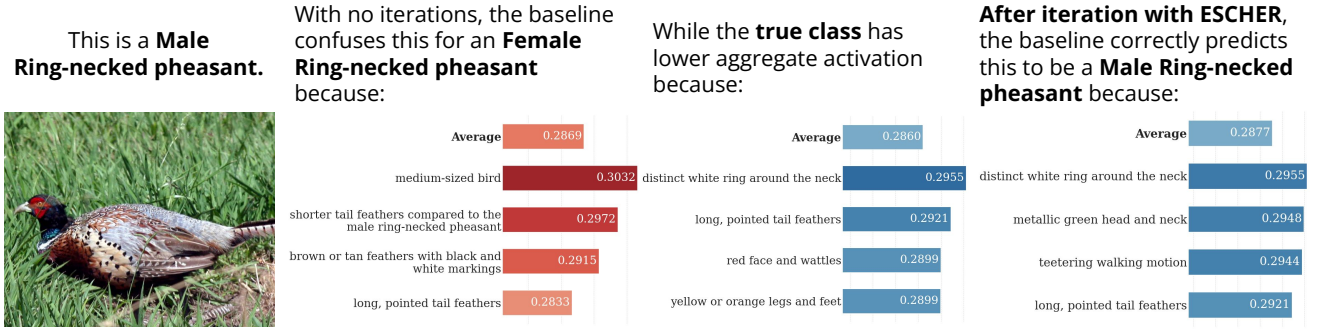


Figure 3. A qualitative example of evolving concepts with CbD+ESCHER in NABirds. Initially, the model is confused between two similar categories with almost the same mean CLIP activation indicating that the concepts provide a coarse categorization signal, but miss subtle nuances. After training with ESCHER, the feedback mechanism identifies new characteristic features (e.g. *metallic green head and neck*) enabling the correct classification. Additional examples are provided in § 8.6.

from the VLM results underperforms integrating a VLM ‘critic’ into the library learning loop.

We visualize qualitative results of concepts for each dataset and corresponding activated images in the supplementary material.

5.5. Backbone VLM/LLM Ablation

ViT-B/16	CLIP	LM4CV	LM4CV +ESCHER	CbD	CbD +ESCHER
CIFAR-100	63.20	81.35	81.72	67.70	69.90
CUB-200-2011	57.06	70.90	77.17	56.16	56.16
Food101	87.24	92.00	92.20	88.51	89.19
NABirds	44.56	66.69	68.71	45.25	45.25
Stanford Cars	61.86	80.87	81.82	65.96	66.34

Table 5. Top-1 accuracy for evolving ESCHER with a weaker LLM (Llama-3.3-70B-4bit) and visual critic (ViT-B/16). ESCHER consistently improves the performance of LM4CV and CbD across datasets.

ESCHER is a meta-algorithm that enhances the performance of existing CBMs. The performance of such models is inherently bottlenecked by the quality of their CLIP-based visual backbone for capturing concept-image relationships and the quality of the GPT-based language backbone for querying relevant concepts. In this experiment, we investigate whether ESCHER can accelerate CBMs using other VLM/LLM backbones. Concretely, we instantiate LM4CV and CbD with a new backbone LLM (4bit quantized Llama-3.3-70B-Instruct [8]) and with new backbone VLMs (ViT-B/16 and ViT-B/32). These backbones are slightly weaker than the base models used in other experiments. As a result, we expect an overall reduction in performance and a weaker learning signal from the visual critic. We maintain the same training and evaluation setup as used in other experiments.

Observations. Results for these experiments are presented in Table 5 (for ViT-B/16) and in the Appendix Table 10 (for ViT-B/32). Overall, we observe that CBMs tend to perform better when paired with stronger backbones. Additionally, in every case, refining CBMs iteratively using ESCHER leads to better performance than relying on a fixed set of concepts.

6. Conclusion

We present ESCHER, a framework for evolving visual concept libraries for visual recognition. ESCHER iteratively updates the library using a VLM as a critic to guide an LLM to generate more effective concepts. This process also enables the LLM to incorporate past histories of the feedback from the VLM. Notably, ESCHER does not require any additional annotations, and is compatible with a range of concept-bottleneck visual recognition systems. We demonstrate our results on concept-bottleneck models in zero-shot, few-shot, as well as fine-tuned settings.

One direction of future work is to further improve the performance of evolving visual libraries on few-shot settings, by incorporating few-shot learning into the library evolution process and allowing the algorithm to leverage limited labeled data. Additionally, we aim to study the application of our approach to more complex visual reasoning tasks, where the learned concept libraries could provide a foundation for higher-level reasoning. By demonstrating the potential of combining library learning and concept-bottleneck visual recognition, we aim to encourage further research at this intersection, towards the development of more robust, interpretable, and intelligent visual recognition systems.

Acknowledgments This research was partially supported by the NSF Expeditions award #CCF-1918651, a DARPA award #HR00112320018, an ARO award

#W911NF2110009, and a DARPA TIAMAT award
#HR0011-24-9-0431.

References

- [1] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. [6](#)
- [2] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. [2](#)
- [3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *Computer vision—ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. [6](#)
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. [6](#)
- [5] Mia Chiquier, Utkarsh Mall, and Carl Vondrick. Evolving interpretable visual classifiers with large language models. *arXiv preprint arXiv:2404.09941*, 2024. [2](#)
- [6] Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Lucas Morales, Luke Hewitt, Luc Cary, Armando Solar-Lezama, and Joshua B Tenenbaum. Dreamcoder: Bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd acm sigplan international conference on programming language design and implementation*, pages 835–850, 2021. [1](#), [3](#)
- [7] Gabriel Grand, Lionel Wong, Maddy Bowers, Theo X. Olausson, Muxin Liu, Joshua B. Tenenbaum, and Jacob Andreas. LILO: Learning interpretable libraries by compressing and documenting code. In *The Twelfth International Conference on Learning Representations*, 2024. [3](#)
- [8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. [8](#)
- [9] Arya Grayeli, Atharva Sehgal, Omar Costilla-Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. In *Neural Information Processing Systems (NeurIPS)*, 2024. [1](#), [2](#), [3](#)
- [10] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scenes as blender code. In *Forty-first International Conference on Machine Learning*, 2024. [1](#)
- [11] 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 *International conference on machine learning*, pages 4904–4916. PMLR, 2021. [2](#)
- [12] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. Xprotonet: diagnosis in chest radiography with global and local explanations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15719–15728, 2021. [2](#)
- [13] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models, 2020. [4](#)
- [14] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. [6](#)
- [15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. [6](#)
- [16] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. [3](#)
- [17] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. [2](#)
- [18] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. [2](#)
- [19] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023*. OpenReview.net, 2023. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#)
- [20] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. [6](#)
- [21] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021. [5](#)
- [22] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15645–15655, 2022. [2](#)
- [23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. [2](#), [6](#), [7](#)
- [24] Oindrila Saha, Grant Van Horn, and Subhansu Maji. Improved zero-shot classification by adapting vlms with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17542–17552, 2024. [2](#), [3](#)

- [25] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. [4](#)
- [26] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024. [2](#)
- [27] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 595–604, 2015. [6](#)
- [28] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset, 2023. [6](#)
- [29] Catherine Wong, Kevin Ellis, Joshua B. Tenenbaum, and Jacob Andreas. Leveraging language to learn program abstractions and search heuristics. In *International Conference on Machine Learning*, 2021. [3](#)
- [30] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3090–3100, 2023. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#)
- [31] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification, 2023. [1](#), [2](#), [3](#), [4](#), [6](#)
- [32] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. [2](#)
- [33] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Hongsheng Li, Yu Qiao, and Peng Gao. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. *arXiv preprint arXiv:2303.02151*, 2023. [2](#)