

Conceptual Learning via Embedding Approximations for Reinforcing Interpretability and Transparency

Maor Dikter , Tsachi Blau , Chaim Baskin

Technion – Israeli Institute of Technology

{maor.dikter, tsachiblau}@campus.technion.ac.il

{chaimbaskin}@technion.ac.il

Abstract

Concept bottleneck models (CBMs) have emerged as critical tools in domains where interpretability is paramount. These models rely on predefined textual descriptions, referred to as concepts, to inform their decision-making process and offer more accurate reasoning. As a result, the selection of concepts used in the model is of utmost significance. This study proposes **Conceptual Learning via Embedding Approximations for Reinforcing Interpretability and Transparency**, abbreviated as CLEAR, a framework for constructing a CBM for image classification. Using score matching and Langevin sampling, we approximate the embedding of concepts within the latent space of a vision-language model (VLM) by learning the scores associated with the joint distribution of images and concepts. A concept selection process is then employed to optimize the similarity between the learned embeddings and the predefined ones. The derived bottleneck offers insights into the CBM’s decision-making process, enabling more comprehensive interpretations. Our approach was evaluated through extensive experiments and achieved state-of-the-art performance on various benchmarks. The code for our experiments is available at <https://github.com/clearProject/CLEAR/tree/main>.

1 Introduction

The unprecedented increase in the utilization of neural networks across diverse fields has highlighted the need for broader insight into how decisions are made. As deep networks grow in complexity, focusing on explaining the decision using a post-hoc method [3, 31, 51], which performs an analysis of the model after its training, is the best path to understanding. Nevertheless, in sensitive fields such as healthcare, relying solely on such explanations is insufficient, and understanding the elements that shape the decision and the reasoning behind it is essential.

To address the need for a thorough understanding of the inner workings of models, interpretable-by-design models [42] are being used. These inherently interpretable models integrate explanations within their architecture, thereby ensuring transparency and offering a more reliable form of reasoning. Concept bottleneck models (CBMs) [23] are one type of interpretable-by-design models. The idea underlying such models is, given an input, first predict an intermediate set of specified concepts and then use this set to predict the target directly. These models enable interpretation in terms of high-level concepts and allow for human interaction. Like many contemporary challenges, these frameworks often employ multi-modal models. Aiming to quantify the relationship between an image and its corresponding textual representations, leveraging VLMs such as CLIP [38] becomes a natural choice. The embedding space of these models enables us to gauge their alignment.

Today, methods for identifying concepts in the bottleneck commonly rely on Large Language Models (LLMs), for example, GPT-3 [6]. These techniques revolve around employing diverse prompts to guide the LLMs in understanding and extracting meaningful concepts. A recent work by Yang et

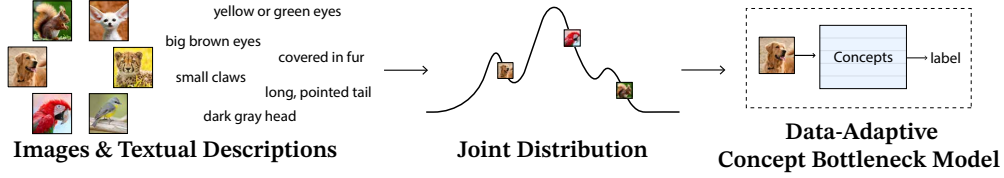


Figure 1: The core components of CLEAR, our proposed paradigm for constructing a data-adaptive CBM by modeling the joint distribution of images and concepts.

al. [61] proposed LaBo, a framework for constructing a CBM. By generating descriptions and employing submodular optimization, LaBo selects relevant concepts for each class. The framework then uses cosine similarity between the encoded image and concepts to make its predictions. Following this path, Yan et al. [60] demonstrate that relevant concepts can be derived from an approximation of the embedding space of VLMs, which then enable the textual descriptions to be located using nearest neighbor search. They anchored the learning of concepts using the Mahalanobis distance [32], a statistical measure that evaluates the distance of a point from a distribution.

While these methods present notable strengths, they also include certain limitations that merit consideration. LaBo [61], for example, requires extensive annotations to accurately represent the data. An overly expansive bottleneck is incomprehensible to humans, could compromise the quality of concepts, and, as demonstrated in Yan et al. [60], often achieves comparable results to the use of random annotations. On the other hand, while it is essential to formulate a small and accurate set of descriptors that efficiently represents our data, achieving so with a rough estimation from a limited number of descriptors falls short. In Yan et al. [60], the set of concepts is relatively small, which limits the framework’s ability to achieve a comprehensive representation. A relatively large set of prior concepts is necessary to accurately estimate the distribution of textual embeddings. Once this distribution is established, reducing the number of concepts can improve interpretability. Interestingly, their approach sometimes showed improved performance without using the Mahalanobis distance, suggesting that the embedding space representation of the VLM may be suboptimal. Furthermore, the method’s concept selection process, being greedy, may lead to less-than-optimal outcomes.

Our approach addresses the aforementioned limitations effectively, as presented in Figure 1, which illustrates the main ideas of our framework. To gain accurate approximations, we use a large pool of prior concepts, a descriptor pool. This strategy enables the learning of a set of embeddings for our textual concepts by training them as a single linear layer within our model. This training process dynamically refines the embeddings based on the model’s learning, ensuring they are fine-tuned to represent the concepts accurately. To guide the learning of concepts appropriately toward meaningful embeddings, we use Langevin sampling to generate concept approximations from the probability density function (PDF) of the joint distribution of images and descriptors within the VLM’s embedding space. Estimating a PDF can be challenging due to an intractable normalizing constant. In score matching, the density of the data is estimated by learning the gradient of the data distribution’s log-likelihood. Langevin dynamics then provides a method for sampling from a distribution using only its score function. By obtaining the score function, our technique directs our learned embeddings toward areas of higher density within the joint image–concept distribution—crucial for effective concept learning. Following this, we construct a similarity matrix by calculating the cosine similarity between the learned embeddings and the descriptor pool. Subsequently, employing the Hungarian method [25], we establish a maximum perfect matching, which identifies the optimal set of textual representations based on their estimations. This approach ensures that our model’s embeddings are both accurate and representative of the underlying conceptual structure.

Overall, our framework offers three main contributions:

- We model the joint distribution density of images and concepts using a score-matching based method, which enables us to develop a novel approach for concept embedding learning via sampling from this distribution.
- We introduce a concept selection methodology that achieves optimal allocation by optimizing joint similarity and develop an interpretable, data-adaptive CBM that surpasses existing models in performance.

- We provide a comprehensive analysis of our framework’s components and demonstrate its interpretability capabilities.

2 Related work

Explainability methods have gained prominence for their utility in demystifying model decisions. These methods often employ saliency-based explanations to highlight input features that significantly influence predictions, with common techniques including feature importance scores [3, 31, 41, 47] and visual heat maps [8, 45, 57, 62]. Such methods are critical in applications where understanding the influence of specific inputs is paramount.

Inherently interpretable models [42] are designed to provide direct insights into the causality within model predictions. These models facilitate a straightforward tracing of cause and effect in decision-making processes [4, 17, 34, 40, 43]. Recent advancements have leveraged VLMs [2, 27, 30, 38] such as CLIP to enhance interpretability, exploiting the model’s ability to measure the similarity between images and descriptive features directly [10, 11, 26, 33, 60, 61].

Extending interpretability in image classification tasks, the use of language descriptions to explain model predictions has been applied across various tasks including object recognition [28, 34, 36], visual question answering [36, 43], text classification [52], and image generation [14]. This approach often involves prompting LLMs [1, 6, 53, 54] to extract and generate relevant content and candidates for explanations [7, 10, 11, 14, 26, 28, 33, 60, 61], bridging the gap between the need for extensive data annotations and human-understandable interpretations.

CBMs [23] represent a structured approach to interpretability, where models explicitly predict a set of intermediary concepts before arriving at a final decision. Many studies have embraced this model structure [9, 10, 11, 15, 44, 46, 59, 60, 61]. To enhance the trustworthiness of the concepts, Kim et al. [20] introduces a visual activation score to assess whether a concept is visually represented. Another approach by Lockhart et al. [29] allows CBMs to abstain from predictions about concepts when there is insufficient confidence, thus focusing on enhancing model reliability when explanations lack certainty.

Some studies have applied a probabilistic approach for constructing CBMs, focusing on modeling the space of concepts within these models. One such study by Kazmierczak et al. [18] represents the distribution of CLIP similarity scores by a mixture of Gaussians and employs statistical tools to explain the classifier. They integrate post-hoc methods such as LIME [40] and SHAP [31] to ascertain the importance scores of each concept. Another research by Kim et al. [19] develops a CBM that predicts concept embeddings by sampling from a normal distribution with learned parameters. Merely predicting embeddings, however, can be inadequate for robust interpretation, as predictions often lack the necessary textual descriptions that anchor these embeddings.

The accurate modeling of the bottleneck involves grasping the distribution density of concepts. Estimating the PDF is a complex yet vital task, tackled by various probabilistic methods [12, 22, 37, 39, 58]. Techniques such as flow networks utilize sequences of invertible transformations to refine a simple initial density into a more complex, desired distribution [39]. In score matching [16], by minimizing the discrepancy between the empirical and model-based score functions, one can learn the density function indirectly. This idea has gained attention across various applications and underpins the theoretical framework of diffusion-based generative models [13, 48, 49].

3 Method

3.1 Problem formulation

We outline the problem we aim to address and detail the approach in developing CLEAR. We consider a dataset D consisting of images and their corresponding labels, denoted as $D = \{(i, c)\}$, where each class $c \in C$ is associated with a set of attributes $A_c = \{a_{c_1}, \dots, a_{c_l}\}$. We define the union of all these attribute sets as our descriptor pool $A = \bigcup_{c \in C} A_c$. Using a VLM, equipped with text and image encoders E_T and E_I , respectively, we project our dataset into the VLM’s embedding space \mathbb{R}^d . Through these encoders, we denote the obtained embeddings for our images and for our descriptor pool $I = \{E_I(i) | (i, c) \in D\}$ and $P = \{E_T(a) | a \in A\}$.

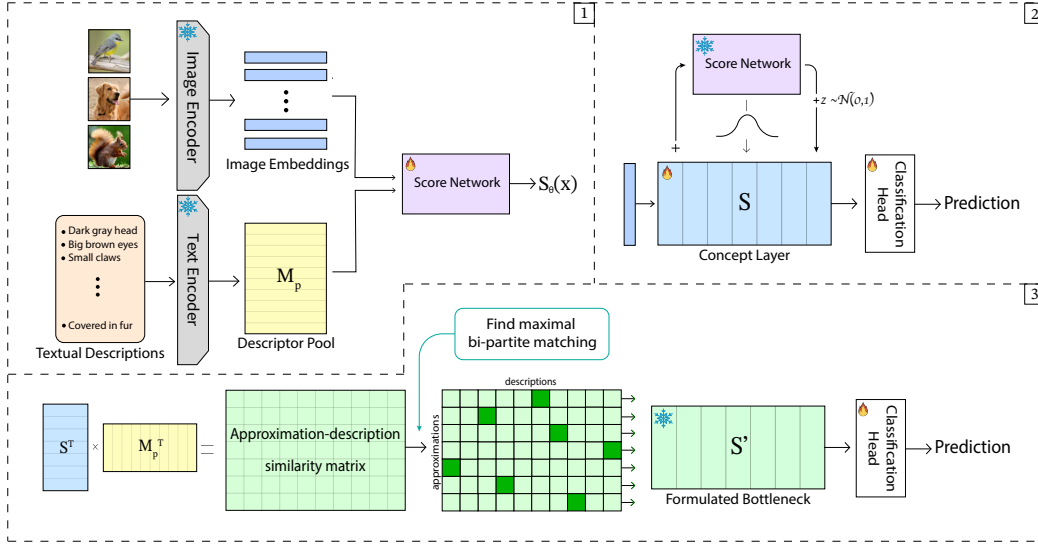


Figure 2: An overview of CLEAR. In step 1 we obtain the image and descriptor embeddings and train the score network. In step 2 we learn the concept approximations and in step 3 we obtain the approximation-description similarity matrix, select the concepts by finding the optimal allocation and integrate our bottleneck.

Our method explores CBMs, which are models of the form $f(g(x))$ and comprise two functions: $g : \mathbb{R}^d \rightarrow \mathbb{R}^k$, mapping an input x to a concept space (the model’s bottleneck), and $f : \mathbb{R}^k \rightarrow \mathbb{R}$, mapping data from the concept space to make the final prediction. The process of learning a CBM can follow one of three approaches: an independent bottleneck approach, where the functions f and g are learned separately with their respective loss functions minimized independently; a sequential bottleneck strategy, which involves training f and g separately by learning g first, fixing it and then learning f ; or a joint bottleneck technique, where f and g are trained simultaneously with the objective to minimize the combined sum of their losses.

Our research adopts a sequential-bottleneck approach to CBM construction. Initially, we learn an approximation to the function g (see Section 3.2). Following this, we move on to formally construct the function g , in Section 3.3. Finally, once g is established, we fix it and train a function f , detailed in Section 3.4. This methodology, illustrated in Figure 2, ensures that each component of the model is optimally trained for its specific role in the overall architecture.

3.2 Embedding approximation learning

Our initial goal is to construct an approximation of the conceptual bottleneck. We focus on learning a linear function, $S : \mathbb{R}^d \rightarrow \mathbb{R}^k$, characterized by its weight matrix, $[S] \in \mathbb{R}^{d \times k}$. This matrix serves as a collection of k unique d -dimensional embeddings. Since S transforms an image embedding into the concept space, training S appropriately results in optimal embeddings that accurately embody our concepts. To facilitate the learning of S , we train it along a function $W : \mathbb{R}^k \rightarrow \mathbb{R}^{|C|}$, responsible for making the model’s classification. Although these functions structurally resemble a CBM, this model is not yet a CBM as it lacks integration with the prior conceptual knowledge we aim to incorporate.

To facilitate the learning of meaningful embeddings, the joint image-descriptor distribution $p(x)$ is modeled into another distribution $p_\theta(x)$ using score matching. The scores of the joint image-descriptor distribution $s_\theta(x)$ are learned such that $s_\theta(x) = \nabla_x \log p_\theta(x) \approx \nabla_x \log p(x)$, aiming to minimize the Fisher Divergence $D(p||p_\theta) = \frac{1}{2} \mathbb{E}_{p(x)} [||\nabla_x \log p(x) - s_\theta(x)||_2^2]$. As demonstrated by Hyvärinen and Dayan [16], this is equivalent to minimizing $\mathbb{E}_{p(x)} [\frac{1}{2} ||s_\theta(x)||_2^2 + \text{tr}(\nabla_x s_\theta(x))]$. Due to the computational complexity of the Hessian term $\nabla_x s_\theta(x)$, sliced score matching (SSM), where scores are projected onto a random direction $v \sim p_v$, is employed. As shown by Song et al.

[50], minimizing the following provides a computationally viable alternative to minimizing the Fisher Divergence.

$$\mathbb{E}_p \mathbb{E}_{p_v} \left[v^T \nabla_x s_\theta(x) v + \frac{1}{2} \|s_\theta(x)\|_2^2 \right] \quad (1)$$

We obtain the score function and integrate it into the embedding learning process using Langevin dynamics. This approach steers the concept embeddings toward the high-density areas of $p(x)$, thereby better aligning with the data’s images and descriptions. The learning of the embeddings in $[S]$ is constrained using the L_2 norm toward the transformed embeddings. The Langevin dynamics’ sampling procedure iteratively updates an example x_0 by $x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p(x_i) + \sqrt{2\epsilon} z_i$ where $z_i \sim \mathcal{N}(0, 1)$ for $i = 0, \dots, t$. This procedure is applied to the columns of $[S]$, transforming a learned embedding $[S]_j^T$, as follows:

$$[S]_{j(i+1)}^T \leftarrow [S]_{j(i)}^T + \frac{\epsilon}{2} \cdot s_\theta([S]_{j(i)}^T) + \sqrt{\epsilon} z_i \quad (2)$$

The overall term guiding the learning of $[S]$ toward $p(x)$ is:

$$\mathcal{L}_{SM}(S) = \frac{1}{k} \sum_{j=1}^k \|[S]_{j(t)}^T - [S]_{j(t)}^T\|_2^2 \quad (3)$$

To ensure accurate classification, the cross-entropy loss function is employed, defined as $\mathcal{L}_{CE}(x, c) = \sum_{j=1}^{|C|} \delta_{j,c} \cdot \log(W(S(x))_j)$, where $W(S(x))_j$ denotes the model’s predicted probability for class j and $\delta_{j,c}$ is the indicator of whether class c is the correct classification for image x . The composite loss function that guides the optimization of our model is formulated as follows:

$$\mathcal{L}(S, x, c) = \lambda \cdot \mathcal{L}_{SM}(S) + \mathcal{L}_{CE}(x, c) = \frac{\lambda}{k} \sum_{j=1}^k \|[S]_{j(t)}^T - [S]_{j(t)}^T\|_2^2 + \sum_{j=1}^{|C|} \delta_{j,c} \cdot \log(W(S(x))_j) \quad (4)$$

Thus, for a given image embedding x , our model determines its prediction by:

$$\hat{c} = \operatorname{argmax}(W(S(x)))$$

3.3 Concept selection

The essence of interpretability in a CBM is rooted in textual concepts, meaning merely deriving the bottleneck approximation S is insufficient. Thus, we venture further, identifying from P a subset of descriptors that closely align with S . Using a matrix $M_p \in \mathbb{R}^{|P| \times d}$ whose rows are the elements of P , we construct an approximation-description similarity matrix $Sim \in \mathbb{R}^{k \times |P|}$ by:

$$Sim = [S]^T \cdot M_p^T \quad (5)$$

Here, $Sim_{i,j}$ measures the similarity between the i -th approximation in $[S]$ and the j -th descriptor in P . Our aim is to uniquely pair each of the k conceptual approximations with a descriptor, maximizing their joint similarity. To achieve this, we employ the Hungarian method [25], an algorithm that finds the optimal assignment that minimizes a total cost in a bipartite matching scenario. Inverting this, we turn the goal of maximizing similarity into one of minimizing cost by subtracting each matrix entry from its highest value, thereby aiming to minimize the total deviation from the maximum similarity.

The runtime complexity of the Hungarian algorithm on Sim is $O(|P|^3)$, which becomes computationally challenging when the descriptor pool is large. To enhance the efficiency, we refine the process by retaining only the top m -most similar concepts in the similarity matrix for each learnable concept. This approach reduces the pool size significantly while still preserving the most promising candidates for matching. Applying this approach enables us to identify an optimal set of embeddings and generate a new matrix, $[S]' \in \mathbb{R}^{d \times k}$, which includes the selected descriptors from P .

Table 1: The comparison of the proposed model on various benchmarks compared to baseline state-of-the-art methods.

Bottleneck Size	Datasets								
	8	CIFAR-10			CIFAR-100		32	Flower 102	204
		10	20	64	100	200			
LaBo [61]	-	78.11	84.84	-	75.10	76.94	-	80.98	86.76
Yan et al. [60]	77.47	80.09	87.99	73.31	75.12	77.29	80.88	87.26	89.02
CLEAR	81.17±0.58	84.19±1.17	89.16±0.73	73.75±0.09	76.07±0.03	77.32±0.00	87.11±0.28	90.19±0.19	91.10±0.05

Bottleneck Size	Datasets					
	32	CUB		64	Food 101	202
LaBo [61]	-	60.93	62.61	-	79.95	81.33
Yan et al. [60]	60.27	63.88	64.05	78.41	80.22	81.85
CLEAR	65.42±0.17	70.18±0.14	69.94±0.13	79.79±0.07	81.61±0.15	82.77±0.18

3.4 Bottleneck integration

Building on the foundation laid in Sections 3.1 to 3.3, where we derived the matrix $[S]'$, we now establish the function $S' : \mathbb{R}^d \rightarrow \mathbb{R}^k$. This function transforms an image embedding x into the concept space through $S'(x) = [S]' \cdot x$. This function S' is exactly the function $g : \mathbb{R}^d \rightarrow \mathbb{R}^k$ of our CBM. To determine the label prediction function $f : \mathbb{R}^k \rightarrow \mathbb{R}$, we fix g and proceed to learn a linear function $W' : \mathbb{R}^k \rightarrow \mathbb{R}^{|C|}$, in line with the approach in Section 3.2. In this instance, however, we streamline the learning process by using the cross-entropy loss $CE(x, c)$ alone, focusing on guiding the transformation of data from the concept space to a classification. The prediction function f is then defined as $f(x') = \text{argmax}(W'(x'))$. By integrating g and f , we synthesize the complete CBM:

$$f(g(x)) = \text{argmax}(W'(S'(x)))$$

4 Experiments

4.1 Experimental setup

Implementation details In our research, we used the CLIP [38] package developed by OpenAI, specifically adopting the base architecture of ViT-B/32 for image encoding. The descriptor pool employed is the textual descriptions provided in LaBo [61], generated by GPT-3 [6] and filtered per class. The score function was constructed using a network of three linear layers. Model training leveraged the PyTorch library and the Adam [21] optimizer for efficient optimization.

For learning the embedding approximations detailed in Section 3.2, we selected model checkpoints that achieved the highest validation accuracy for use in the concept selection phase outlined in Section 3.3. Similarly, we chose the model with the highest validation accuracy, as seen in Section 3.4, and reported its results. All experiments were conducted using an Nvidia GeForce RTX 3090, and the hyperparameters for all datasets and configurations are provided in the supplementary material in Section A.2.

Baselines We compared our method with two other interpretable CBM strategies—LaBo [61] and the framework developed by Yan et al. [60], elaborated upon in Section 1. LaBo [61] requires that the bottleneck size be a multiple of the dataset’s class count. Therefore, for each dataset containing $|C|$ classes, we assessed our model using bottleneck sizes of $|C|$ and $2|C|$. Additionally, we included an experiment with a smaller bottleneck in line with Yan et al. [60], validating our findings and demonstrating the efficacy of identifying a precise set of concepts. This approach ensures a thorough evaluation and showcases conceptual clarity and efficiency improvement.

Datasets Our framework for image classification was evaluated across five datasets: CIFAR-10 [24], CIFAR-100 [24], CUB-200-2011 [56], Flower-102 [35] and Food-101 [5]. Training was conducted on the full datasets, with performance assessed based on the accuracy obtained from the test sets.

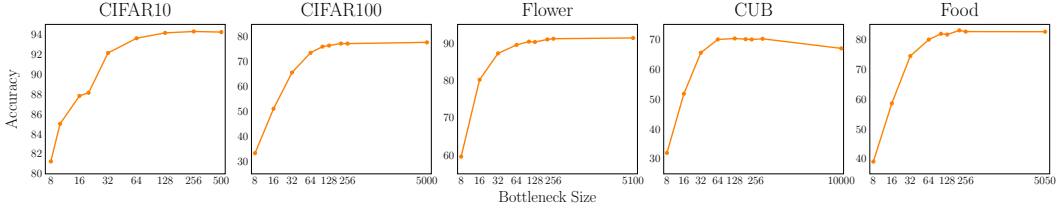


Figure 3: Test accuracy comparison on different bottleneck sizes across all datasets.

4.2 Main results

The results of the aforementioned experiments are presented in Table 1. We conducted three independent runs for each experimental condition to ensure the robustness of our findings. We report the mean accuracy and the standard error of the mean of these runs, providing a view of the variability and reliability of our method. Across the benchmarks, our method demonstrates an average performance increase of 2.84%, with specific gains in accuracy noted across different datasets: a 2.99% increase in CIFAR-10, 0.47% in CIFAR-100, 3.75% in Flower, 5.78% in CUB, and 1.23% in Food. These improvements underscore our method’s effectiveness and highlight our approach’s adaptability. The full results of the independent runs are listed in Section A.1 in the appendix.

Bottleneck size To explore the influence of the number of concepts selected, we conducted experiments with varying concept quantities: 8, 16, 32, 64, 128, 256, and the entire descriptor pool (findings are presented in Figure 3). The findings generally indicate that using more concepts tends to enhance data representation. More specifically, in CIFAR-10, our model with 256 concepts outperforms the model using the full set of 500 descriptors. Similarly, in CUB, using 64 concepts yields better results than the full pool of 10,000 descriptors, and in the Food dataset, selecting 202 concepts surpassed the performance of the full pool containing 5,050 descriptors. These observations suggest that our method effectively identifies the most relevant concepts for each dataset and excludes less pertinent descriptors. Considering the CUB dataset as an example, the superior performance observed at intermediate concept sizes can potentially be explained by the homogeneity of the classes, with all focusing on various bird species. This conceptual similarity among the classes means fewer annotations are necessary for adequate interpretation.

Interpretability analysis To assess the interpretability of our model during its construction and at inference time, we examined how it forms and uses its concepts at different bottlenecks across four datasets: CIFAR-10, CIFAR-100, Flower, and Food. For each dataset, we identified the bottleneck formed and reported some of the key concepts presented in it. At inference, we present the normalized concept scores for a selected image from each dataset and pinpoint the most significant concept, offering a clear and effective way to interpret the model’s decisions. For instance, an image of a truck from CIFAR-10 shows the highest similarity with the concept “*large, red vehicle with four wheels*”, and an image of a Caesar salad from Food shows the highest similarity with “*green salad with romaine lettuce, croutons, and Parmesan cheese*”. The selected concepts presented in Figure 4 offer informative interpretations. These examples highlight how our model’s interpretability not only aids in understanding its internal mechanisms but also enhances trust in its outputs by aligning its decision pathways with comprehensible and relevant human-understandable concepts. Additional visualizations reflecting the diversity of the bottleneck are presented in Section B in the appendix.

4.3 Ablation study

To dissect the contribution of individual components within our model, an ablation study was conducted, focusing on four key aspects:

Regularization In our approach, the regularization function steers the learning process toward the joint image–descriptor distribution, calculated by Equation 3 and weighted into the complete loss function in Equation 4. We evaluate its significance by contrasting it with alternative methods. One such method involves the Mahalanobis distance [32], a statistical measure that evaluates the distance from a point to a distribution. The Mahalanobis distance between a point $x \in \mathbb{R}^n$ and a distribution








Bottleneck construction	<div>CIFAR-10</div> <div></div> <div>CLEAR</div> <div><div>white breast</div><div>large, red vehicle with four wheels</div><div>big ears and large eyes</div><div>sailing on a blue sea</div><div>⋮</div></div>		<div>CIFAR-100</div> <div></div> <div>CLEAR</div> <div><div>round fruit with a thin green skin</div><div>tall, slim tree with long, curved leaves</div><div>shells and rocks scattered along the shore</div><div>tall plant with a large, yellow flower</div><div>⋮</div></div>	
	Inference	<div>Image</div> <div></div> <div><div><div>0.142</div><div>0.233</div><div>0.172</div><div>0.184</div><div>0.156</div><div>⋮</div></div><div>large, red vehicle with four wheels</div></div>	<div>Image</div> <div></div> <div><div><div>0.195</div><div>0.093</div><div>0.179</div><div>0.260</div><div>0.140</div><div>⋮</div></div><div>tall plant with a large, yellow flower</div></div>	
		Bottleneck construction	<div>Flower</div> <div></div> <div>CLEAR</div> <div><div>shades of pink, purple, and blue</div><div>red, orange, and yellow petals</div><div>member of the protea family</div><div>gigantic white arum lily flower</div><div>⋮</div></div>	
Inference	<div>Image</div> <div></div> <div><div><div>0.199</div><div>0.236</div><div>0.307</div><div>0.150</div><div>0.218</div><div>⋮</div></div><div>shades of pink, purple, and blue</div></div>		<div>Image</div> <div></div> <div><div><div>0.382</div><div>0.221</div><div>0.159</div><div>0.254</div><div>0.193</div><div>⋮</div></div><div>green salad with romaine lettuce, croutons, and parmesan cheese</div></div>	

Figure 4: Interpretability analysis of our proposed framework during inference

Q with mean and covariance matrix $\mu \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n \times n}$ is $\sqrt{(x - \mu)^T \cdot \Sigma^{-1} \cdot (x - \mu)}$. In our context, this distance between each row in $[S]^T$ and the embedding space distribution approximated by P is defined as $\mathcal{L}_{MA}(S) = \frac{1}{k} \sum_{j=1}^k \sqrt{([S]_j^T - \mu)^T \cdot \Sigma^{-1} \cdot ([S]_j^T - \mu)}$, where μ and Σ are the mean and covariance of P . Additionally, we compare this to using the Euclidean distance (L_2 norm), which computes the distance between each learned embedding and the descriptor embeddings as $\mathcal{L}_{EU}(S) = \sum_{j=1}^k \frac{1}{|P|} \sum_{h=1}^{|P|} \|[S]_j^T - e_h\|_2^2$ for every $e_h \in P$. Furthermore, we assess the impact of omitting any regularization, focusing learning solely on classification via cross-entropy.

We evaluate the effectiveness of each regularization approach by conducting a comparison among these methods. The results, as outlined in Table 2, demonstrate the superiority of our proposed learning method over the alternatives.

Pool size We evaluated the impact of the size of the descriptor pool by comparing the pool used in our method with the one used in Yan et al. [60]. The findings in Table 3 support the idea that having a more extensive pool of descriptors enhances our ability to match the descriptor distribution closely.

Concept selection We assessed the effectiveness of our concept selection strategy by evaluating the Hungarian algorithm [25] that we implemented in our approach, as outlined in Section 3.3. This

Table 2: An examination of the impact of the regularization function.

Regularization	No. of concepts in Food							
	8	16	32	64	128	256	101	202
None	35.09	56.06	71.41	78.03	80.55	81.38	80.42	81.00
\mathcal{L}_{EU}	23.41	44.07	64.45	76.50	80.97	82.51	80.18	82.35
\mathcal{L}_{MA}	33.46	55.13	71.18	77.93	80.48	81.32	80.12	81.35
\mathcal{L}_{SM}	39.10	58.58	74.42	79.91	81.61	82.59	81.86	83.01

Table 3: A study on the impact of the size of the descriptor pool

Pool size	No. of concepts in Flower								
	8	16	32	64	128	256	102	204	full
503	54.41	75.88	80.09	85.29	88.72	89.90	87.84	90.09	91.17
5100	59.60	80.19	87.25	89.51	90.29	91.17	90.39	90.98	91.37

method is initially compared to the nearest neighbor (NN) algorithm, similar to the approach taken by Yan et al. [60], which matches each learned embedding with the nearest concept based on the L_2 distance. We also contrasted this with a strategy involving randomly selecting descriptors from our available pool. This comparative analysis ensures that our method’s performance enhancements are meaningful and substantiated. It also justifies the complexity added by the Hungarian algorithm over simpler or more random approaches. The figures in Table 4 show that choosing concepts by maximizing the embeddings’ joint similarity yields better results compared to other approaches.

Table 4: An analysis of the effects of the concept selection method

Selection method	No. of concepts in Flower							
	8	16	32	64	128	256	102	204
Random	20.19	51.96	70.19	79.90	86.76	88.72	85.39	88.52
NN	59.60	80.19	87.15	88.92	89.70	90.88	89.70	90.68
Our method	59.60	80.19	87.25	89.51	90.29	91.17	90.39	90.98

5 Limitation

The utilization of a sufficiently large descriptor pool is important for achieving accurate approximations, as well as for ensuring diversity of descriptions and adaptability to the data. It is, nevertheless, important to recognize that relying on a predefined descriptor pool can limit the applicability of the concept selection process to the specific dataset being used. While our work provides an optimal concept assignment, future research could enhance this field by developing methods to generate new concepts, further improving the descriptive power of datasets. In addition, it should be considered that using LLM-generated concepts may introduce biases, which can reflect and perpetuate existing societal biases. This underscores the importance of being mindful of the broader societal impacts when leveraging LLMs for concept generation.

6 Conclusion

In this study, we developed a data-adaptive CBM by learning a precise set of concepts that accurately represents the data, thereby enabling clear interpretations and facilitating human interaction with the model. Our framework, CLEAR, was compared across five different image classification datasets, consistently achieving state-of-the-art results. To thoroughly assess the contributions of our paradigm, we conducted an in-depth analysis of our model’s components. We demonstrated the advantages of the individual components, each of which plays a crucial role in enhancing the performance and interpretability of our CBM.

While our method is evaluated on image classification tasks, we note that our approach directly applies to any computer vision task. This broad applicability stems from the versatility of VLMs when handling a wide range of tasks within this domain. Furthermore, adapting our framework with modifications suitable for different data types can broaden its application to various other tasks. For example, incorporating pre-trained language models as text encoders allows our framework to seamlessly adapt to text classification tasks.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736, 2022.
- [3] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, pages 63–71. Springer, 2016.
- [4] Moritz Böhle, Mario Fritz, and Bernt Schiele. B-cos networks: Alignment is all we need for interpretability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10329–10338, 2022.
- [5] 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] 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.
- [7] Soravit Changpinyo, Doron Kukliansky, Idan Szepes, Xi Chen, Nan Ding, and Radu Soricut. All you may need for vqa are image captions. *arXiv preprint arXiv:2205.01883*, 2022.
- [8] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.
- [9] Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5948–5955, 2023.
- [10] Yan Cui, Shuhong Liu, Liuzhuozheng Li, and Zhiyuan Yuan. Ceir: Concept-based explainable image representation learning. *arXiv preprint arXiv:2312.10747*, 2023.
- [11] Zhili Feng, Anna Bair, and J Zico Kolter. Text descriptions are compressive and invariant representations for visual learning. 2023.
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

- [14] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20406–20417, 2023.
- [15] Qihan Huang, Jie Song, Jingwen Hu, Haofei Zhang, Yong Wang, and Mingli Song. On the concept trustworthiness in concept bottleneck models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21161–21168, 2024.
- [16] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.
- [17] Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva. On explaining decision trees. *arXiv preprint arXiv:2010.11034*, 2020.
- [18] Rémi Kazmierczak, Eloïse Berthier, Goran Frehse, and Gianni Franchi. Clip-qda: An explainable concept bottleneck model. *arXiv preprint arXiv:2312.00110*, 2023.
- [19] Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. Probabilistic concept bottleneck models. *arXiv preprint arXiv:2306.01574*, 2023.
- [20] Injae Kim, Jongha Kim, Joonmyung Choi, and Hyunwoo J Kim. Concept bottleneck with visual concept filtering for explainable medical image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 225–233. Springer, 2023.
- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [23] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.
- [24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [25] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [26] Kathleen M Lewis, Emily Mu, Adrian V Dalca, and John Guttag. Gist: Generating image-specific text for fine-grained object classification. *arXiv e-prints*, pages arXiv–2307, 2023.
- [27] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.
- [28] Liunian Li, Zi-Yi Dou, Nanyun Peng, and Kai-Wei Chang. Desco: Learning object recognition with rich language descriptions. *Advances in Neural Information Processing Systems*, 36, 2024.
- [29] Joshua Lockhart, Daniele Magazzeni, and Manuela Veloso. Learn to explain yourself, when you can: Equipping concept bottleneck models with the ability to abstain on their concept predictions. *arXiv preprint arXiv:2211.11690*, 2022.
- [30] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.
- [31] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.
- [32] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, 80:S1–S7, 2018.

- [33] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022.
- [34] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14933–14943, 2021.
- [35] 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.
- [36] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8779–8788, 2018.
- [37] P Wayne Power and Johann A Schoonees. Understanding background mixture models for foreground segmentation. In *Proceedings image and vision computing New Zealand*, volume 2002, pages 10–11, 2002.
- [38] 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.
- [39] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.
- [40] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.
- [41] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [42] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *nat mach intell* 1: 206–215, 2019.
- [43] Fawaz Sammani, Tanmoy Mukherjee, and Nikos Deligiannis. Nlx-gpt: A model for natural language explanations in vision and vision-language tasks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8322–8332, 2022.
- [44] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.
- [45] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [46] Andrei Semenov, Vladimir Ivanov, Aleksandr Beznosikov, and Alexander Gasnikov. Sparse concept bottleneck models: Gumbel tricks in contrastive learning. *arXiv preprint arXiv:2404.03323*, 2024.
- [47] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.
- [48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

- [49] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- [50] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020.
- [51] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310, 2018.
- [52] Zhen Tan, Lu Cheng, Song Wang, Bo Yuan, Jundong Li, and Huan Liu. Interpreting pretrained language models via concept bottlenecks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 56–74. Springer, 2024.
- [53] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.
- [54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models (2023). *arXiv preprint arXiv:2302.13971*, 2023.
- [55] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [56] Catherine Wah, Branson Steve, Welinder Peter, Perona Pietro, and Belongie Serge. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [57] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.
- [58] Stanisław Węglarczyk. Kernel density estimation and its application. In *ITM web of conferences*, volume 23, page 00037. EDP Sciences, 2018.
- [59] Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. Energy-based concept bottleneck models: unifying prediction, concept intervention, and conditional interpretations. *arXiv preprint arXiv:2401.14142*, 2024.
- [60] 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.
- [61] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023.
- [62] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

A Experiments

A.1 Additional results

We present comprehensive results that expand on those reported in the paper. Initially, we list the hyperparameter values used in our model configuration and present the test accuracy results for varying numbers of concepts: 8, 16, 32, 64, 128, 256, one per class (1-*pc*), two per class (2-*pc*), and the entire descriptor pool. We detail the hyperparameters used and their selection in Section A.2 and report their results in Table 6.

Additionally, Table 5 includes the complete results of the three runs that form the basis for the mean and standard error reported for each dataset in Table 1.

Table 5: Complete results of the three runs for each dataset

Dataset	Accuracy when varying no. of concepts		
	8	10	20
CIFAR-10	81.25	85.02	88.14
	80.11	85.68	88.73
	82.16	81.87	90.61
CIFAR-100	64	100	200
	73.6	76.08	77.32
	73.71	76.12	77.31
Flower	32	102	204
	87.25	90.39	90.98
	86.56	90.39	91.17
CUB	32	200	400
	65.53	70.05	70.19
	65.08	70.48	69.71
Food	64	101	202
	79.91	81.86	83.01
	79.83	81.64	82.40
	79.64	81.33	82.92

A.2 Implementation details

Hyperparameters In Table 6, we provide an overview of the hyperparameters that configure our model for each dataset, along with their values and the empirical results. During the sampling procedure from the joint image–descriptor distribution, the transformation is calculated as formulated in Equation 2. For each dataset, we search for suitable values of $\epsilon \in \{1, 0.1, 0.01\}$ and $t \in \{1, 3, 5, 7, 10\}$. To balance the two terms in the loss function, we determine the optimal λ value from $\lambda \in \{1, 0.1, 0.01, 0.001\}$.

We also fine-tune the batch size, learning rate, random seed values, and the number of epochs for training the model during the embedding approximation learning phase ($epochs_1$), as detailed in Section 3.2, and during the training of the linear layer ($epochs_2$), as described in Section 3.4.

We obtain our score model by minimizing the objective presented in Equation 1. Training the score model involves a network with three linear layers, each having hidden dimensions of 1024. For all datasets, training is performed on the image and descriptor embeddings for 1000 epochs using the

Adam optimizer with a fixed learning rate of $1e - 4$. The batch size for the images remains the same as before, while the batch size for the descriptors is set to 32.

Table 6: Hyperparameter values and full results on varying numbers of concepts.

CIFAR-10										
Dataset										
ϵ	1									
t	7									
λ	0.01									
batch size	4096									
learning rate	0.01									
seed	4									
$epochs_1$	1000									
$epochs_2$	2000									
no. of concepts	8	16	32	64	128	256	1- <i>pc</i>	2- <i>pc</i>	full	
m	5	5	5	5	5	10	5	5	-	
accuracy	81.25	87.82	92.13	93.61	94.15	94.29	85.02	88.14	94.23	
CIFAR-100										
Dataset										
ϵ	0.1									
t	5									
λ	0.1									
batch size	4096									
learning rate	0.01									
seed	0									
$epochs_1$	1000									
$epochs_2$	4000									
no. of concepts	8	16	32	64	128	256	1- <i>pc</i>	2- <i>pc</i>	full	
m	5	5	5	5	5	5	5	5	-	
accuracy	33.30	51.13	65.7	73.6	76.51	77.29	76.08	77.32	77.79	
Flower										
Dataset										
ϵ	0.1									
t	5									
λ	0.01									
batch size	4096									
learning rate	0.001									
seed	1									
$epochs_1$	2000									
$epochs_2$	20000									
no. of concepts	8	16	32	64	128	256	1- <i>pc</i>	2- <i>pc</i>	full	
m	5	5	5	5	5	5	5	5	-	
accuracy	59.60	80.19	87.25	89.51	90.29	91.17	90.39	90.98	91.37	
CUB										
Dataset										
ϵ	1									
t	10									
λ	1									
batch size	32									
learning rate	0.01									
seed	0									
$epochs_1$	5000									
$epochs_2$	8000									
no. of concepts	8	16	32	64	128	256	1- <i>pc</i>	2- <i>pc</i>	full	
m	5	5	5	5	5	5	5	5	-	
accuracy	32.01	51.81	65.53	69.96	70.29	69.95	70.05	70.19	66.98	
Food										
Dataset										
ϵ	1									
t	1									
λ	1									
batch size	4096									
learning rate	0.01									
seed	0									
$epochs_1$	200									
$epochs_2$	4000									
no. of concepts	8	16	32	64	128	256	1- <i>pc</i>	2- <i>pc</i>	full	
m	5	5	5	5	5	5	5	5	-	
accuracy	39.10	58.58	74.42	79.91	81.61	82.59	81.86	83.01	82.55	

Descriptors pool filtering During the concept selection phase described in Section 3.3, we construct Sim by employing Equation 5 and retain only the top m -most similar concepts in Sim for each learnable concept. Initially, we set m to 5 and define $TopDes = \bigcup_{i=1}^k \{sort(Sim_i)^{(1)}, \dots, sort(Sim_i)^{(m)}\}$, where we sort Sim_i and select the top m -most similar embeddings. If the resulting pool size is greater than k , we proceed with concept selection; otherwise, we iteratively find $TopDes$ for $m_{i+1} = 2m_i$ until this condition is met. Generally, a low value of m indicates diverse learned embeddings. The reader is referred to Table 6 for the obtained values.

Citations and rights We have thoroughly cited all datasets and research papers used in our experiments throughout our paper. The CLIP model [38] is available under the MIT license.

B Descriptor visualizations

To gain insights into the structure of the textual descriptions, we visualize the descriptor pool along with the selected concepts that form our bottleneck. This visualization allows us to understand the diversity in the selection of the concepts.

By lowering the dimension of each embedding, we use t-SNE [55] to visualize both the embeddings of the descriptor pool and the embeddings of the selected concepts. The visualizations for the CIFAR-10, CIFAR-100, Flower, CUB, and Food datasets are presented in Figures 5 to 9. In these visualizations, each green point represents a concept from the descriptor pool, while each blue point represents a concept in the CLEAR bottleneck.

These visualizations illustrate how well the selected concepts represent the broader pool, which is vital for ensuring the robustness and generalizability of our approach. They demonstrate that our method’s concept selections effectively distinguish between different conceptual areas and provide a diverse set of concepts.

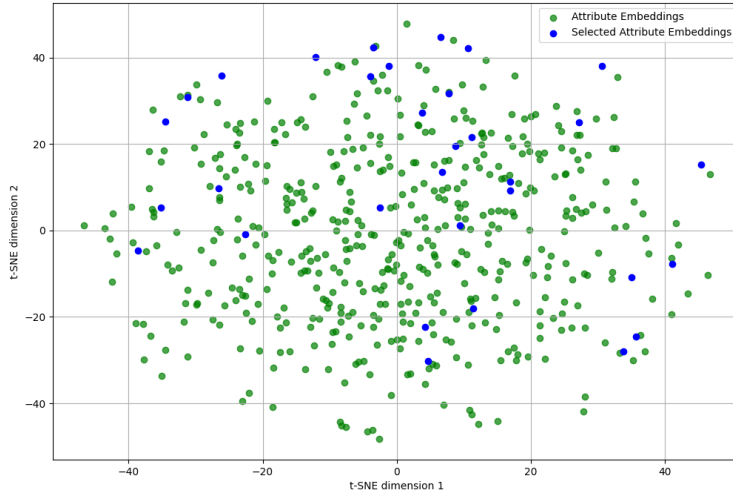


Figure 5: t-SNE visualization of CIFAR-10 descriptors

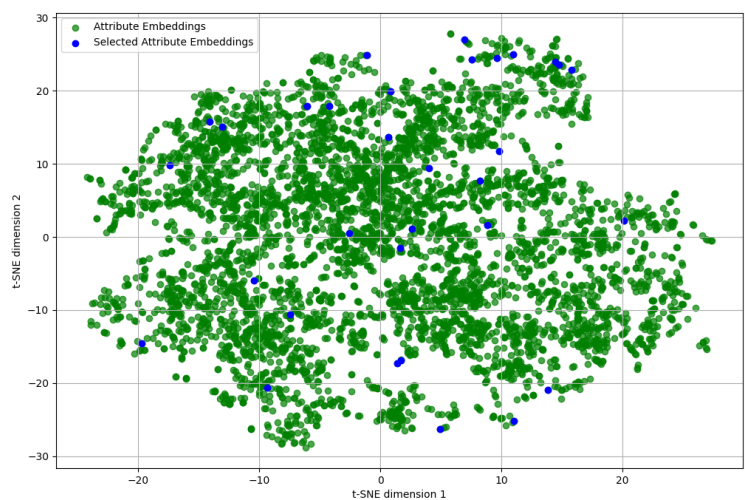


Figure 6: t-SNE visualization of CIFAR-100 descriptors

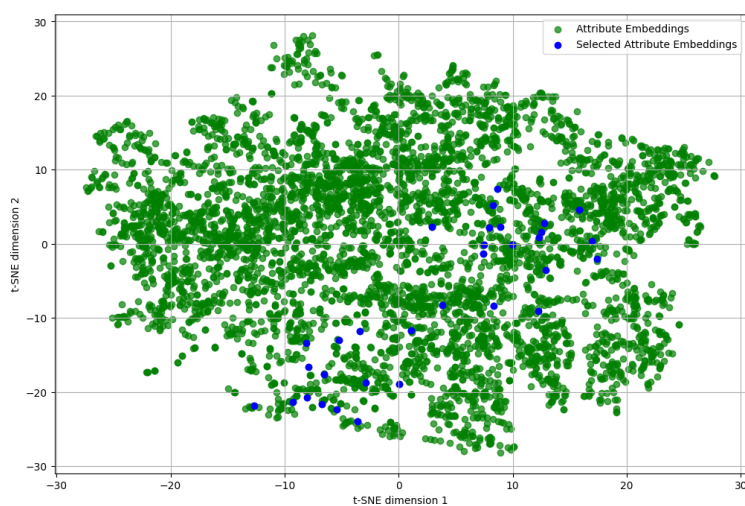


Figure 7: t-SNE visualization of Flower descriptors

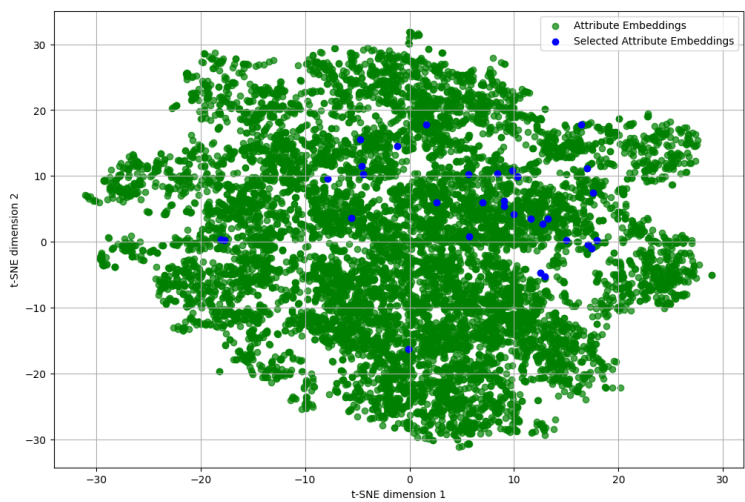


Figure 8: t-SNE visualization of CUB descriptors

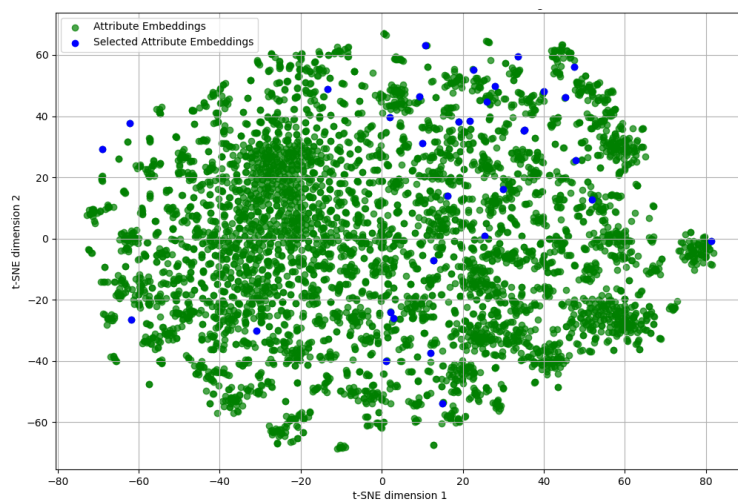


Figure 9: t-SNE visualization of Food descriptors