

Politecnico di Torino

Politecnico di Torino

Data Science

a.y. 2025/2026

Graduation Session September 2026

Hierarchical Concept Representation in Concept Bottleneck Models via Box Embeddings

Relators:

Eliana Pastor

Elonora Poeta

Candidate:

Stefano Gamba

Abstract

As Deep Neural Networks are increasingly deployed in safety-critical domains, the need for Explainable Artificial Intelligence (XAI) continues to grow. Concept Bottleneck Models (CBMs) offer an interpretable-by-design architecture by mapping raw inputs to intermediate, human-readable semantic concepts before producing final predictions. However, standard CBM paradigms predominantly rely on flat linear classifiers, making them vulnerable to shortcut learning and concept leakage, while also overlooking hierarchical relationships among concepts. Sequential optimization can lead to algebraic compensation and reliance on spurious concept co-occurrences, while joint end-to-end training allows classification gradients to propagate through the bottleneck, potentially encoding task-specific visual information that is not aligned with the intended concept semantics.

To address these limitations, this thesis introduces a novel decoupled architecture that explicitly models hierarchical relationships among attributes using Box Embeddings. Complex ontologies and logical hierarchies are represented as geometric spatial constraints, replacing flexible algebraic class weights with structured topological operations. In particular, the proposed Dynamic Box inference strategy performs classification through spatial intersections and containment relations, reducing the model's ability to exploit arbitrary combinations of concepts. Because the underlying hierarchical containment rules remain strictly frozen according to the predefined ontology, during joint trainign classification gradients refine the visual extractor's representation while remaining structurally incapable of altering concept semantics, thereby definitively resolving concept leakage by design.

Empirical evaluations on the Animals with Attributes 2 (AwA2) dataset show that the proposed geometric framework improves the consistency of the learned concept representation with the underlying attribute hierarchy and reduces reliance on shortcut relationships between concepts. We also evaluate the resulting representations in a Zero-Shot Learning setting, where the hierarchical approach shows improved transfer performance compared with standard similarity-based baselines. Overall, this work demonstrates how explicit geometric structure can be incorporated into Concept Bottleneck Models to obtain representations that are more consistent with predefined semantic hierarchies.

Acknowledgements

I would like to express my gratitude to my supervisors, Eliana Pastor and Elonora Poeta, for proposing this thesis topic and for their invaluable guidance. Their continuous supervision, expertise, and clear research direction were fundamental throughout the development of this work. Additionally, I would like to acknowledge that artificial intelligence tools were employed during the drafting of this manuscript. These tools were utilized exclusively as writing assistants to help formalize the concepts, refine the language, and express the conducted research as clearly and effectively as possible.

Table of Contents

List of Figures VI

1 Introduction 1
- 1.1 The Need for Explainable AI (XAI) in Safety-Critical Domains . . . 1
- 1.2 Concept Bottleneck Models (CBMs) – Promises and Limitations . . 1
- 1.3 The Shortcut Learning Problem in Linear Classifiers 2
- 1.4 Thesis Objectives and Contributions 3

2 Related Works 5
- 2.1 Concept Bottleneck Models and Interpretable AI 5
- 2.2 The Shortcut Learning Dilemma in Flat Classifiers 6
- 2.3 Hierarchical Representations in C-XAI 7
- 2.4 Zero-Shot Generalization through Concepts 9

3 Background 11
- 3.1 Concept-Based Interpretable Architectures 11
  - 3.1.1 Formal Definition of Concept Bottleneck Models (CBMs) . . 11
  - 3.1.2 Interpretability, Faithfulness and Test-Time Intervention (TTI) 12
  - 3.1.3 Concept Embedding Models (CEM) 12
- 3.2 Training Paradigms and Bottleneck Vulnerabilities 12
  - 3.2.1 Independent and Sequential Training 12
  - 3.2.2 Joint Training and Concept Leakage 14
- 3.3 Geometric Representation of Logical Hierarchies (Box Embeddings) 14
  - 3.3.1 Limitations of Point-Vector Spaces 14
  - 3.3.2 Mathematical Formulation of Boxes 15
  - 3.3.3 Intersection and Volume Functions 15
  - 3.3.4 Differentiable Relaxations (Gumbel / Bessel) 16
  - 3.3.5 Modeling Probabilities and Logical Entailment 17

4 Methodology 18
4.1 Architectural Overview . . . 18
4.2 Phase 1: Embedding the Logical Hierarchy . . . 19
4.2.1 Implementation of the Latent Space . . . 19
4.2.2 Translating Ontologies into Spatial Constraints . . . 19
4.3 Inference Strategies: Defining the Classifiers . . . 21
4.3.1 Combining Box Embeddings and Linear Layers . . . 21
4.3.2 Strict Logical Containment (The Dynamic Box) . . . 21
4.4 Phase 2: Visual Concept Extractor and Training Paradigms . . . 24
4.4.1 Independent and Sequential Training . . . 24
4.4.2 Joint Training for Linear Classifier . . . 24
4.4.3 Frozen Geometric Joint Training . . . 24
4.4.4 Addressing Concept Leakage . . . 26

5 Experiments and Results 27
5.1 Experimental Setup . . . 27
5.1.1 Datasets and Concept Ontologies . . . 27
5.1.2 Evaluation Metrics . . . 29
5.1.3 Training Details and Hyperparameters . . . 32
5.2 Linear Layer Classification and Shortcut . . . 34
5.2.1 Performance Linear Layer Classifier . . . 34
5.2.2 Exposing Shortcut Learning: Global Flaws and Local Manifestations . . . 34
5.3 Evaluating Hierarchical Inference (Seen Classes) . . . 41
5.3.1 Linear Layer vs. Dynamic Box . . . 41
5.4 Zero-Shot Learning (ZSL) Transfer . . . 42
5.4.1 ZSL Protocol and Baselines . . . 42
5.4.2 Zero-Shot Performance Comparison . . . 43
5.4.3 Topological Insights on Unseen Classes . . . 44
5.5 Faithfulness and Test-Time Intervention (TTI) . . . 52
5.5.1 Intervention Protocols . . . 52
5.5.2 TTI Trajectories (Linear vs. Geometric) . . . 53

6 Conclusion and Future Work 55
6.1 Summary of Contributions . . . 55
6.2 Limitations of the Proposed Approach . . . 56
6.3 Future Work . . . 57

Bibliography 59

List of Figures

4.1 Architectural overview of Hybrid Approaches. Visual concept probabilities act as scaling factors for the pre-trained geometric features. However, the resulting representations are aggregated by an unconstrained flat linear layer, functionally reintroducing the vulnerability to shortcut learning. 22

4.2 Phases 1 & 2: The CNN predicts the animal's traits with high confidence. The probabilities act as "sculptors" on the latent space. Since the probabilities for "c1," "c2," and "c3" are very high, the image box (B_{img}) physically contracts to the exact intersection region (the purple volume) of these three pre-trained concepts. Absent concepts (e.g., wings) push their boundaries to infinity and are ignored. Phase 3: The final class boxes are frozen (from Phase 1). The Class 2 box is far from B_{img} because, while Class 2 has concept 1, it has not concept 2 for example (it does not share the volume). Class 1 box, conversely, is perfectly enclosed within B_{img}. The volume calculation will identify Class 1 as the predicted class in a geometrically rigorous manner. 23

5.1 DAG representing the AWA2 concept hierarchy after the manipulation described in Section 5.1.1. Due to the extensive scale of the full concept hierarchy DAG, displaying all node labels simultaneously renders the text illegible. To ensure clarity while preserving the overall structural context of the graph, a representative root-to-leaf path has been visually highlighted 29

5.2 Comparison between the Ground Truth topological constraints and the learned spatial probabilities. Left: The target conditional probability matrix $P(i|j)$ generated from the AwA2 ontology, where values of 1.0 indicate strict containment and 0.0 indicate spatial disjunction. Right: The actual spatial intersections predicted by the model after Phase 1 optimization, demonstrating near-perfect alignment with the predefined hierarchy (the diagonal elements represent $P(i|i)$ and because of the nature of the boxes these probabilities will be always equal to 1). 30

5.3 Geometric representation of a parent-child structural inclusion. The optimization process strictly forces the bounding box of the child concept (flys, in green) to be contained entirely within the spatial volume of the parent concept (wings, in yellow), thereby guaranteeing the logical entailment $P(wings|flys) = 1.0$ by design. 30

5.4 Geometric representation of sibling disjunctions. Mutually exclusive semantic traits—such as the colors red (yellow box) and orange (green box)—are subjected to a target intersection probability of 0.0. The optimizer correctly carves the latent space to ensure these concepts are spatially disjoint, structurally preventing impossible attribute combinations. 31

5.5 Logical Violations per classes proves that linear layer uses shortcut learning. 37

5.6 Example algebraic compensation made by the Linear Classifier. To overcome this negative evidence given by 'meatteath' and to force the prediction, the linear layer mathematically compensates by relying on the incorrect features, like 'buckteeth'. 38

5.7 Example of taxonomic violation produced by the Linear Classifier. The presence of mouth, head, and leg are utilized as negative algebraic weights to penalize the logit, despite being perfectly predicted with a 1.00 probability 39

5.8 Example of inability of the Linear Classifier to impose logical vetoes. The model confidently identifies the fundamental attribute blue (probability 0.97), and the linear classifier correctly applies a negative weight to it, Yet, the linear model overrides this penalty. 40

5.9 Standard ZSL Confusion Matrix for Cosine Similarity Baseline . . . 45

5.10 Standard ZSL Confusion Matrix for Dynamic Box Model with joint training 46

5.11 Concept Error Rate for the Cosine Similarity baseline utilizing Independent Training. The visual extractor produces scattered binary mispredictions across various concepts and unseen classes. Due to the additive nature of the flat vector space, the model algebraically compensates for these errors by summing the positive signals of correctly identified attributes, allowing for graceful degradation rather than categorical failure. 47
5.12 Concept Error Rate for the Dynamic Box architecture utilizing Frozen Geometric Joint Training. The matrix exposes severe, localized high-confidence errors on specific unseen classes, such as the Giant Panda. Governed by strict geometric AND-logic, the model cannot algebraically "outvote" these confident errors. Instead, erroneous traits physically drag the dynamic image volume into logically incompatible regions of the latent space, resulting in feature fragmentation. 48
5.13 Uncertainty Map for the Cosine Similarity baseline utilizing Independent Training. The heatmap reveals polarized predictions, characterized by a lack of high uncertainty bands ($p \approx 0.5$). Because the visual extractor is optimized in isolation using solely Binary Cross-Entropy, it maximizes predictive certainty without facing contradictory gradients from the downstream classification task, resulting in confident visual features that lack global topological awareness. 49
5.14 Uncertainty Map for the Dynamic Box architecture utilizing Frozen Geometric Joint Training. Prominent dark bands indicate high uncertainty ($p \approx 0.5$), illustrating the concept predictor's gradient oscillation between raw visual evidence and the strict topological penalties backpropagated from the frozen latent space. This structural uncertainty geometrically dilates the relaxed boundaries of the image box, driving the localized semantic clustering of hierarchically ambiguous classes like the Rat and Raccoon 50
5.15 TTI curves comparison. Intervention is simulated correcting the k most uncertain concept predictions 54

Chapter 1

Introduction

1.1 The Need for Explainable AI (XAI) in Safety-Critical Domains

In recent years, the development of deep neural networks has improved significantly across various tasks, yet these models often function as "black boxes." Composed of a vast number of parameters combined non-linearly, their learning and reasoning processes lack transparency and are difficult for humans to interpret.

While this opacity may be acceptable in low-stakes applications, it becomes difficult to accept in safety-critical domains such as healthcare, autonomous navigation, and defense where an erroneous algorithmic decision can lead to catastrophic physical, financial, or ethical consequences.

Furthermore, these models could produce erroneous or flawed decisions. In such cases, the ability to interpret how those conclusions were reached is crucial for ensuring trust and awareness regarding the decisions made using them. This social necessity is reflected in recent regulatory frameworks, such as the European Union's AI Act and the GDPR's "right to explanation," which mandate transparency for high-risk AI systems [1]. These arguments have driven a significant paradigm shift in machine learning research, expanding the focus from raw predictive accuracy to the field of Explainable AI (XAI), which aim at making algorithmic reasoning more understandable to humans.

1.2 Concept Bottleneck Models (CBMs) – Promises and Limitations

XAI methodologies can be divided into two main categories: post-hoc explanations and inherently interpretable-by-design architectures [2]. Post-hoc methods attempt

to approximate and explain the behavior of a black-box model after it has been trained, often acting as an external probe. Instead, interpretable-by-design architectures impose transparency directly building or modifying the internal structure of a model, making it explainable by design and eliminating the need for external explanations.

Among the interpretable-by-design models, Concept Bottleneck Models (CBMs) have emerged as a highly promising architecture. In a standard CBM pipeline, the model is decomposed into two distinct stages: a concept predictor that maps raw inputs (e.g., images) to intermediate, human-readable semantic concepts, and a label predictor that maps these concept activations to the final target class. This structural design theoretically guarantees that the final prediction is based exclusively on understandable features, allowing human experts to verify the model's logic and dynamically intervene to correct mispredicted concepts.

Despite these theoretical promises, the current CBM paradigm suffers from methodological limitations. The mapping from the concept bottleneck to the final prediction is handled by a standard linear classifier. While computationally efficient, this flat architecture could fail to capture the complex, hierarchical, and logically asymmetric relationships inherent in real-world taxonomies. Furthermore, standard optimization paradigms struggle to maintain the semantic purity of the bottleneck, leading to vulnerabilities that can compromise the model's true interpretability.

1.3 The Shortcut Learning Problem in Linear Classifiers

The vulnerabilities of standard CBMs manifest differently depending on the chosen training paradigm, but both ultimately severely undermines the model's faithfulness. When CBMs are trained sequentially (i.e., optimizing the concept extractor first, followed by the linear classifier), the architecture frequently falls prey to the phenomenon of shortcut learning. Instead of learning robust, logically sound rules that align with human reasoning, the final linear layer exploits algebraic compensation to minimize the training loss. By leveraging negative weights and spurious statistical correlations among concept activations, the linear classifier mathematically compensates for incorrect or noisy intermediate concept predictions. Consequently, the model achieves high classification accuracy while relying on a flawed understanding of the visual concepts. Conversely, when CBMs are trained jointly to maximize overall task accuracy, they suffer from a phenomenon known as concept leakage. During joint optimization, the gradients from the final classification loss backpropagate through the bottleneck, encouraging the neural network to encode task-relevant but semantically unrelated visual information into the concept logits. As a result, the concepts lose their strict semantic purity, becoming a conduit

for uninterpretable signals that bypass the intended meaning of the bottleneck entirely. In both scenarios, the bottleneck's activations no longer represent the true causal reasons behind the model's output, rendering the purported interpretability of the CBM an illusion. This lack of structural rigor not only invalidates the primary objective of explainable AI but also limits the model's capacity to generalize to out-of-distribution data.

1.4 Thesis Objectives and Contributions

The primary motivation of this thesis is to investigate how injecting prior structural knowledge regarding hierarchical relationships among concepts can enhance a model's ability to generalize, improve its faithfulness, and elevate its overall interpretability. By embedding strict ontological structures directly into the learning process, we hypothesize that the model can develop more robust and human-aligned reasoning pathways. A significant and highly beneficial consequence of enforcing this structural prior is the natural mitigation of the shortcut learning flaws inherent in standard flat architectures. To achieve this, this thesis proposes replacing the vulnerable linear classification layer with a hierarchical concept representation framework utilizing Box Embeddings. By translating structural ontologies and logical hierarchies into geometric spatial constraints, Box Embeddings provide a mathematically rigorous medium for inference. Instead of relying on learned algebraic weights, the proposed architecture performs classification through exact spatial intersections and strict logical containment. The primary contributions of this thesis are as follows:

1. **Geometric Representation of Hierarchies:** We formulate and implement a novel decoupled architecture that maps visual concepts into a highly structured geometric latent space, embedding complex logical rules (such as parent-child entailment) directly into the inference mechanism

2. **Eradication of Shortcut Learning as a Structural Consequence:** By replacing flat algebraic classifiers with geometric containment strategies (specifically, Template Matching and the Dynamic Box approach), we demonstrate that the model naturally avoids algebraic compensation, thereby structurally preserving the faithfulness of the concept bottleneck.

3. **Facilitating Joint Training without Concept Leakage:** By utilizing a strictly geometric classifier that remains frozen based on the predefined ontology, our architecture prevents arbitrary task gradients from polluting the concept space. This enables the exploration of joint optimization paradigms while avoiding the concept leakage that traditionally plagues standard CBMs

4. **Model Evaluation:** We empirically evaluate the proposed hierarchical architecture, measuring its performance on both standard classification tasks and Zero-Shot Learning (ZSL) protocols. The results suggests that enforcing logical reasoning through Box Embeddings could help the model to generalize its learning process and make the model able to recognize better unseen classes.

Chapter 2

Related Works

2.1 Concept Bottleneck Models and Interpretable AI

The pursuit of inherently interpretable deep learning has led to the development of Concept Bottleneck Models (CBMs), [3], which fundamentally restructure the decision-making pipeline. Instead of mapping raw inputs directly to final predictions, CBMs decompose the classification task into two distinct stages: an encoder that maps raw inputs to a set of predefined, human-understandable concepts, and a subsequent label predictor that maps these intermediate concepts to the final target class. This architectural design was primarily motivated by three core desiderata: interpretability, predictability, and intervenability. By exposing the intermediate reasoning steps, CBMs theoretically allow domain experts to understand which features drive a prediction and enable test-time interventions, where humans can correct mispredicted concepts to improve the final model output. CBMs can be optimized through independent, sequential, or joint training paradigms, balancing the trade-off between concept accuracy and overall task performance.

To address the performance degradation often observed in standard scalar-based CBMs—especially in real-world scenarios where concept annotations are incomplete—Concept Embedding Models (CEMs) were introduced, [4]. CEMs replace single scalar probabilities with high-dimensional supervised vectors, learning a mixture of two distinct embeddings for each concept representing its active and inactive states. While this additional capacity allows CEMs to match or exceed the task accuracy of standard black-box models and supports effective test-time interventions, it fundamentally alters the nature of the information passing through the bottleneck.

2.2 The Shortcut Learning Dilemma in Flat Classifiers

Despite their theoretical promises, extensive literature has demonstrated that standard concept-based architectures frequently do not learn as intended, [5] and [6] and [7].

The vulnerabilities of Concept Bottleneck Models (CBMs) become structurally evident when evaluating the standard paradigm used to map intermediate concepts to the final target class: the flat linear classifier. While computationally efficient, linear layers are fundamentally prone to a pervasive deep learning failure mode known as "shortcut learning". As characterized by Geirhos et al., [8], shortcut learning occurs when a model achieves high accuracy on standard, independent and identically distributed (i.i.d.) training sets by exploiting unintended decision rules or spurious correlations, rather than learning the robust, intended solution. These shortcuts give the illusion of high performance but fail catastrophically under out-of-distribution (o.o.d.) testing or when deployed in real-world scenarios.

In standard CBM architectures, the vulnerability to shortcut learning manifests in two distinct but equally detrimental ways, depending on the chosen training paradigm. During sequential optimization, shortcut learning primarily manifests as algebraic compensation. The linear classifier seeks the most mathematically efficient route to minimize the final task loss based on the imperfect, predicted concept probabilities supplied by the visual extractor. Because linear layers lack structural constraints, they frequently resort to utilizing negative weights to artificially separate classes in the latent space. Instead of building a generative, logical definition of a class, the model treats true semantic attributes as algebraic liabilities to neutralize upstream errors.

When CBMs undergo joint training to maximize overall task performance, shortcut learning manifests in a specific and severe form known as concept leakage. In this paradigm, both the concept extractor and the linear classifier are optimized end-to-end.

Post-hoc interpretability analyses reveal that, when jointly optimized with the downstream task, the concepts learned by CBMs,often fail to correspond to semantically meaningful regions in the input space. Instead of acting as a strict semantic bottleneck, the intermediate layer acts as a proxy to pass target information, severely violating the predictability and interpretability desiderata.

Because the flat linear layer acts as a highly parameterized, unconstrained conduit, arbitrary gradients from the final classification loss backpropagate directly into the concept bottleneck. If the predefined concepts are insufficient for perfect separation, or if non-semantic visual artifacts offer an easier shortcut to minimize the loss, the optimizer will exploit this pathway allowing non-concept-related,

task-relevant information to leak through to the final classifier as demonstrated by Mahinpei et al., [6]. Consequently, soft concept representations—the real-valued probabilities predominantly used in standard CBMs—lose their strict semantic purity; they encode much more information about the underlying data distribution than the intended concept itself. This specific genre of shortcut learning renders post-hoc interpretations highly misleading, polluting the bottleneck and fundamentally violating the predictability and interpretability desiderata of CBM architectures.

Recent studies, [7], have established rigorous information-theoretic frameworks to characterize this vulnerability, categorizing it into concepts-task leakage and inter-concept leakage. These frameworks reveal that while architectures like CEMs resolve the accuracy-interpretability trade-off in terms of raw predictive performance, they do so by structurally embedding concepts-task leakage by design. Because the high-dimensional concept vectors in CEMs are only loosely constrained by the concept reconstruction loss, they act as conduits for arbitrary task-relevant information that bypasses the intended semantic meaning altogether. Consequently, while the model achieves high accuracy and responds well to interventions, the semantic purity of the bottleneck is compromised, casting doubt on the true interpretability and safety of these models in high-stakes deployment scenarios.

The pervasive nature of shortcut learning—whether through algebraic compensation or concept leakage—highlights the structural inadequacy of flat, linear classifiers for C-XAI. This critical vulnerability forms the primary motivation for introducing rigid, hierarchical geometric representations, such as Box Embeddings. By structuring the hierarchical relationships explicitly among the attributes themselves, rather than relying on unconstrained weights between attributes and final classes, the model is physically constrained. This geometric approach effectively eradicates the mathematical pathways that enable these unintended shortcuts.

2.3 Hierarchical Representations in C-XAI

The inherent limitations of flat Concept Bottleneck Models have driven recent research to explore hierarchical representations, operating under the hypothesis that injecting structural knowledge into the model enhances both interpretability and generalization. However, current literature reveals a dichotomy in how "hierarchy" is defined, broadly split between architectural (or visual) hierarchies and semantic (or taxonomic) hierarchies.

A significant portion of recent work focuses on architectural hierarchies, extracting concepts at different levels of visual granularity or across different depths of the neural network. For instance, Coarse-to-Fine Concept Bottleneck Models (CF-CBM) introduce a dual-level approach that extracts concepts from both the whole image and localized patch-specific regions using data-driven binary latent

variables, [9]. Similarly, the Concept Prompting and Aggregating Network (CoPA) extracts and aligns concept representations across multiple shallow and deep layers of a visual encoder to capture multiscale features, [10]. Other approaches, such as ConceptFlow, [11], and HINT, [12], operate post-hoc to trace how semantic concepts dynamically propagate, emerge, or transform across the successive layers of a standard Convolutional Neural Network. In the domain of Graph Neural Networks, the HELP framework introduces hierarchical pooling by clustering node embeddings to progressively form coarser conceptual subgraphs, [13].

While these methods successfully capture multiscale visual information and provide valuable insights into network depth, they fundamentally address the spatial or layer-wise distribution of features rather than the logical, ontological relationships among the concepts themselves. Furthermore, CF-CBM and CoPA ultimately aggregate their hierarchical features using linear layers or learnable gated networks for the final task prediction. Consequently, they remain highly vulnerable to algebraic compensation and shortcut learning.

To address semantic relationships, other frameworks attempt to explicitly model taxonomic ontologies. SupCBM constructs a two-level concept set, distinguishing between broad perceptual concepts and granular descriptive concepts, utilizing an intervention matrix for label prediction, [14]. Analogously, Hierarchical Concept Embedding Models (HiCEM) discover finer-grained sub-concepts from a pretrained model's embedding space, structuring top-level concepts as mixtures of these sub-concept embeddings, [15]. HPnet maps feature prototypes to a predefined taxonomy tree, computing joint distributions over fine-grained classes through specialized prototype layers at each parent node, [16]. In a similar vein, Pittino et al. decompose object classification into a hierarchy where low-level 3D shapes and visual categories are first segmented and subsequently fed into a Multi-Layer Perceptron (MLP) to predict a higher-level "Logical category", [17]. Despite explicitly acknowledging semantic taxonomies, these methods share a critical structural flaw: the aggregation of hierarchical information relies heavily on unconstrained algebraic operations. Whether utilizing intervention matrices, fully connected layers at taxonomic nodes, gated fusion mechanisms, or MLPs, these architectures map intermediate hierarchical representations to final predictions using mathematical weighted sums. As demonstrated in Chapter 3, any reliance on continuous algebraic weights allows the optimizer to learn negative penalties and exploit spurious correlations to separate classes in the latent space. They fail to enforce strict ontological rules—such as the absolute geometric impossibility of a mutually exclusive trait—because the additive logic can simply overpower negative evidence if enough spurious features are present.

An alternative to additive logic is proposed by the Hierarchical Concept Memory Reasoner (H-CMR), which learns a Directed Acyclic Graph (DAG) over concepts and tasks, utilizing neural attention to select and symbolically execute logic rules,

[18]. While H-CMR introduces rigorous symbolic execution, the rules and the graph structure are learned dynamically from the data. Learning the ontology directly from potentially noisy visual datasets risks encoding spurious, data-specific correlations into the logic rules themselves, deviating from objective expert knowledge.

The pervasive reliance on algebraic classifiers and dynamically learned graphs in current hierarchical C-XAI literature strongly motivates the transition to the geometric framework proposed in this thesis. To achieve true faithfulness, the hierarchical relationship must be structured explicitly among the concepts themselves—such as enforcing that the concept "claw" is an absolute structural subset of the concept "paw"—and it must be enforced by design. By translating a predefined, expert-curated Directed Acyclic Graph into exact spatial constraints utilizing Box Embeddings, the proposed architecture replaces learnable algebraic weights with pure geometric AND-logic. Inference via the Dynamic Box approach relies on exact volume intersections and strict spatial containment, rendering algebraic compensation mathematically impossible. This structural rigidity ensures that the model cannot bypass the predefined ontology, guaranteeing that the purported hierarchy is not merely a feature representation, but the absolute governing law of the model's reasoning.

2.4 Zero-Shot Generalization through Concepts

Standard classification objectives typically treat all misclassifications as equally incorrect, ignoring the rich taxonomic structure inherent in real-world data. Bertinetto et al. demonstrated that explicitly leveraging class hierarchies during the training of deep neural networks encourages models to "make better mistakes," prioritizing semantically sensible errors over catastrophic ones, which in turn leads to more robust learned representations, [19]. This observation directly inspires the core hypothesis of this thesis: embedding a strict, hierarchical structure into the concept bottleneck should not only resolve interpretability flaws but also fundamentally enhance the model's capacity to generalize.

To rigorously evaluate a model's true generalization capabilities, standard supervised classification is often insufficient, as it primarily tests in-distribution performance and algebraic memorization. Instead, Zero-Shot Learning (ZSL) serves as the natural and ultimate proving ground for this hypothesis. The ZSL paradigm aims to recognize instances of novel classes that were entirely absent during the training phase. To achieve this, ZSL methods must associate observed and non-observed classes through auxiliary side-information, typically by mapping visual features into an intermediate semantic space composed of shared, human-understandable attributes or concepts. If a Concept-Based Model has genuinely learned the causal semantic definitions of its concepts—rather than merely exploiting

spurious algebraic shortcuts—it should be able to logically recombine those known concepts to accurately identify unseen classes.

The evaluations conducted by Xian et al. established rigorous, unified protocols for benchmarking ZSL and Generalized Zero-Shot Learning (GZSL) methods, exposing the flaws in earlier unstandardized metrics and highlighting the necessity of proper dataset splits, [20] and [21]. As part of this comprehensive benchmarking effort, they introduced the Animals with Attributes 2 (AwA2) dataset, a large-scale collection of animal images annotated with an extensive set of shared visual attributes. Because AwA2 provides the precise semantic side-information required to link disjoint training and testing classes, it has become a cornerstone benchmark for concept-based generalization. By evaluating the proposed geometric bottleneck architecture on the AwA2 dataset under standard ZSL protocols, this thesis aims to demonstrate that structurally overcoming shortcut learning through geometric hierarchical constraints is mandatory for effective and robust zero-shot transfer.

Chapter 3

Background

This section introduces the contributions from the literature used to conduct the experiments via the methodology explained in Chapter 4. Specifically, Sections 3.1 and 3.2 introduce the CBM architecture and the various ways it can be trained, while Section 3.3 introduces Box Embeddings—the tool used to represent semantically relevant concepts and their logical and hierarchical relationships.

3.1 Concept-Based Interpretable Architectures

3.1.1 Formal Definition of Concept Bottleneck Models (CBMs)

The standard Concept Bottleneck Model (CBM), presented in [3], introduces a structural shift from traditional end-to-end deep learning architectures by decomposing the decision-making process into two sequential, human-interpretable mappings. Formally, a CBM requires a dataset annotated with both the final target labels y and an intermediate set of human-specified concepts c. The architecture is defined by two distinct functions: a Concept Predictor $g : X \to C$, which maps raw inputs x (e.g., images) to the intermediate semantic concept space, and a Label Predictor $f : C \to Y$, which maps these predicted concepts to the final target class. By construction, the model's final prediction $\hat{y} = f(g(x))$ relies on the raw input entirely through the bottleneck of predicted concepts $\hat{c} = g(x)$, which are explicitly trained to align component-wise with the true concepts c.

3.1.2 Interpretability, Faithfulness and Test-Time Intervention (TTI)

For a CBM to provide genuine transparency, it must be strictly faithful to its bottleneck. Faithfulness dictates that the final target prediction depends exclusively on the intermediate concept representations, guaranteeing that the model relies on understandable features rather than spurious, hidden visual artifacts. This structural guarantee enables the powerful Human-in-the-Loop paradigm known as Test-Time Intervention (TTI). Because the intermediate concepts are semantically meaningful, a domain expert can dynamically interact with the model by editing mispredicted concept values $\hat{c}$ and replacing them with expert ground truth concepts c^{true} to correct misclassifications. Intervening on these concepts and observing the resulting propagation to the final prediction yields actionable counterfactual explanations, such as understanding how a classification would change if a specific feature were removed. However, true interpretability demands that the relationships defining the bottleneck are structurally sound.

In this thesis, the objective is to structure the hierarchical relationship explicitly among the attributes themselves—for example, embedding the ontological relationship between a specific feature like "claw" and a broader anatomical parent like "paw"—rather than focusing the hierarchy on the relationship between the attributes and the final predicted class. Imposing this concept-to-concept structure provides a rigorous foundation that standard flat bottlenecks currently lack.

3.1.3 Concept Embedding Models (CEM)

While scalar concept probabilities provide highly readable intermediate outputs, they can severely restrict the information flow through the network. To regain expressive capacity without abandoning the interpretable bottleneck, Concept Embedding Models (CEM), [4], expand the scalar version of each concept into a high-dimensional vector. This expansion allows the model to capture more complex representations while maintaining a structural tie to the intended semantic meaning, serving as an alternative to the strictly scalar constraints of foundational CBMs.

3.2 Training Paradigms and Bottleneck Vulnerabilities

3.2.1 Independent and Sequential Training

The performance and integrity of a CBM depend heavily on how the functions f and g are optimized. As shown in [3] the foundational formulations rely on two primary loss functions: a concept loss L_{C_j} that measures the discrepancy between

predicted and true concepts, and a task loss L_Y for the final target prediction. In the independent bottleneck paradigm, the two functions are learned entirely separately. The concept predictor is trained to minimize the sum of concept losses across all k concepts:

$$\hat{g} = \arg\min_{g} \sum_{i,j} L_{C_j}(g_j(x^{(i)}); c_j^{(i)})$$

while the label predictor is independently trained to minimize the task loss using the true concept annotations c:

$$\hat{f} = \arg\min_{f} \sum_{i} L_Y(f(c^{(i)}); y^{(i)})$$

.

Conversely, in the sequential training paradigm, the concept predictor $\hat{g}$ is optimized first using the same formulation, but the label predictor is subsequently trained using the model's own intermediate predictions $\hat{g}(x)$:

$$\tilde{f} = \arg\min_{f} \sum_{i} L_Y(f(\hat{g}(x^{(i)})); y^{(i)})$$

.

While sequential training allows the label predictor to adapt to the specific error distributions of the concept extractor, both paradigms expose standard CBMs to the critical flaw of shortcut learning.

Because the mapping from $C \to Y$ is predominantly handled by a flat linear classifier, the network routinely minimizes the training loss via algebraic compensation rather than logical rule formulation.

To understand how this vulnerability manifests structurally, it is necessary to examine the mathematical formulation of the final label predictor when implemented as a standard linear layer. For a target class y, the unnormalized logit z_y is computed as the weighted sum of the predicted concept probabilities $\hat{c}_i$:

$$z_y = \sum_{i=1}^{k} w_{y,i} \cdot \hat{c}_i + b_y$$

where $w_{y,i}$ represents the learned weight connecting the i-th concept to class y, and b_y is the bias term. In a rigorously interpretable and generative model, a concept that is objectively part of a class definition should always contribute positively to this sum. However, during sequential training, the linear layer's sole objective is to minimize $\mathcal{L}_{task}$ given the imperfect intermediate predictions $\hat{c}$.To achieve this, the optimizer frequently resorts to algebraic compensation. If the Concept Predictor systematically hallucinates a specific feature, the linear classifier can learn to assign an unjustified negative weight ($w_{y,i} < 0$) to that concept, effectively treating a true semantic attribute as an algebraic liability to neutralize

the error. By exploiting these negative weights and leveraging spurious statistical correlations among co-occurring concepts, the linear layer mathematically forces an artificial separation between classes in the latent space. Consequently, the model achieves high classification accuracy while completely bypassing the intended semantic ontology, reducing the interpretability of the CBM.

3.2.2 Joint Training and Concept Leakage

To maximize overall task performance, CBMs can be optimized via joint training, where both the concept predictor and the linear classifier are trained end-to-end simultaneously, [3]. In this paradigm, the network is optimized by minimizing a combined total loss function:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda \mathcal{L}_{concept}$$

where the hyperparameter λ dictates the trade-off between prioritizing task accuracy and concept accuracy. While joint training theoretically allows the visual feature extractor to adapt its representations based on the requirements of the final classification task, it introduces a documented vulnerability known as concept leakage, [7]. Because the entire pipeline is differentiable and unconstrained by structural logic, arbitrary task gradients backpropagate directly from the final classification loss through the concept bottleneck and into the visual extractor. Mathematically, the update to the concept extractor is heavily influenced by the derivative of the task loss with respect to the intermediate concept activations, $\frac{\partial \mathcal{L}_{task}}{\partial \hat{c}}$. If the predefined semantic concepts are insufficient to perfectly separate the target classes, or if spurious visual artifacts offer a more mathematically efficient shortcut to minimize $\mathcal{L}_{task}$, the optimizer will exploit this gradient flow. The neural network is actively encouraged to encode task-relevant, non-semantic visual information directly into the concept logits. As a result, the concepts lose their strict semantic purity, becoming an algebraic conduit for uninterpretable signals. The bottleneck's activations no longer represent the true causal reasons behind the model's output. By allowing raw visual signals to "leak" through the semantic layer, joint training severely pollutes the true interpretability of the model.

3.3 Geometric Representation of Logical Hierarchies (Box Embeddings)

3.3.1 Limitations of Point-Vector Spaces

Standard embedding techniques typically represent concepts as point vectors in a high-dimensional continuous space, relying on symmetric distance metrics like

cosine similarity or Euclidean distance to measure semantic relatedness. However, these point-vector spaces are fundamentally ill-equipped to model strict logical hierarchies and asymmetric relationships, such as ontological "is-a" entailments. If a highly specific concept entails a broader concept, the relationship is strictly directional; a symmetric metric cannot geometrically capture that one concept is a subset of another without implying the reverse. While Order Embeddings [22] attempted to solve this asymmetry by representing concepts as infinite cones, they lack the structural capacity to model disjoint concepts or negative correlations. A mathematically rigorous ontology requires a representation capable of modeling asymmetry, exact intersections, and disjointness simultaneously.

3.3.2 Mathematical Formulation of Boxes

To overcome the limitations of point-vectors and infinite cones, Box Embeddings, [23] and [24], map concepts into regions of finite measure, specifically representing them as high-dimensional products of intervals (hyperrectangles) within a geometric latent space. Formally, an n-dimensional box B parameterized by a set of latent variables θ is defined as the Cartesian product of closed intervals:

$$B(\theta) = \prod_{i=1}^{n} [z_i(\theta), Z_i(\theta)]$$

where z_i and Z_i represent the lower-left (minimum) and top-right (maximum) coordinates in the i-th dimension, respectively. To ensure structural validity—meaning that the maximum coordinate is strictly greater than or equal to the minimum coordinate ($z_i \leq Z_i$)—boxes are parameterized using an unconstrained minimum vector alongside a strictly positive dimensional length (often denoted as Δ). For instance, the maximum coordinates can be derived as $Z_i = z_i + \text{softplus}(\Delta_i)$, mathematically guaranteeing non-negative spatial side lengths.

3.3.3 Intersection and Volume Functions

In this geometric framework, logical conjunctions (the logical AND operation) and hierarchical entailments are modeled entirely through spatial overlap. Given two boxes A and B, their geometric intersection is calculated by taking the pointwise maximum of their minimum coordinates and the pointwise minimum of their maximum coordinates:

$$A \cap B = \prod_{i=1}^{n} [\max(z_{A,i}, z_{B,i}), \min(Z_{A,i}, Z_{B,i})]$$

. The probability mass of a concept, or a conjunction of concepts, corresponds directly to the spatial volume of its representative box. The exact, or "hard,"

volume of a box is computed simply as the product of its structural side lengths:

$$\text{Volume}(B) = \prod_{i=1}^{n} \max(Z_i - z_i, 0)$$

3.3.4 Differentiable Relaxations (Gumbel / Bessel)

While the hard intersection and volume functions provide exact geometric calculations for inference, they present severe structural challenges for gradient-based optimization. The primary issue arises because the gradients of rigid bounding boxes are zero almost everywhere, particularly when boxes are entirely disjoint.

This phenomenon is mathematically rooted in the formulation of the hard intersection and hard volume. The intersection of two boxes relies on piecewise functions, specifically $\max(z_{A,i}, z_{B,i})$ and $\min(Z_{A,i}, Z_{B,i})$. Furthermore, the hard volume computes the spatial size by applying a Rectified Linear Unit (ReLU) formulation to the resulting side lengths to prevent negative volumes:

$$\text{Volume}(B) = \prod_{i=1}^{n} \max(Z_i - z_i, 0)$$

If two boxes are completely disjoint in even a single dimension, the term $(Z_i - z_i)$ for their intersection becomes negative. The $\max(\cdot, 0)$ operation flattens this negative value to exactly zero.

Because the derivative of a constant zero is zero, this geometric rigidity effectively halts the learning process during backpropagation. In standard gradient descent, network parameters are updated proportionally to the gradient of the loss function ($\theta_{new} = \theta_{old} - \alpha\nabla\mathcal{L}$). When the gradient evaluates to zero, the parameter update is also zero. Consequently, the optimizer receives no directional signal to indicate how to move the boxes. Disjoint boxes that are theoretically supposed to overlap (e.g., a child concept that must be contained within a parent concept) become permanently stuck, as the neural network cannot "see" which direction to shift the coordinates to bridge the gap.

To enable stable training and resolve this gradient starvation, differentiable relaxations are employed. The rigid max and min operations required for intersections can be replaced with smoothed approximations, such as the Gumbel Intersection. This approach utilizes a temperature-scaled LogSumExp (LSE) function to provide a soft probabilistic boundary:

$$z_{int} = \beta \ \text{LSE}\left(\frac{z_A}{\beta}, \frac{z_B}{\beta}\right)$$

$$Z_{int} = -\beta \ \text{LSE}\left(-\frac{Z_A}{\beta}, -\frac{Z_B}{\beta}\right)$$

where $\text{LSE}(x, y) = \log(\exp(x) + \exp(y))$ and β represents the intersection temperature parameter. Unlike the sharp, non-differentiable corners of the rigid max and min functions, the LSE function is smooth and everywhere differentiable.

Similarly, the volume calculation can be relaxed using a Bessel Approximation, which estimates the expected volume under the assumption that the box coordinates act as location parameters for Gumbel random variables. This formulation replaces the rigid $\max(\cdot, 0)$ threshold with a smooth softplus function, defined as $\text{softplus}(x) = \log(1 + \exp(x))$:

$$\text{Volume} \approx \prod_{i=1}^{n} T \cdot \text{softplus}\left(\frac{Z_i - z_i - 2\gamma\beta}{T}\right)$$

where T is the volume temperature and γ is the Euler-Mascheroni constant. Because the softplus function is infinitely differentiable and asymptotically approaches zero without ever flattening out entirely, it ensures a continuous gradient flow. Even when the theoretical intersection volume is negative (i.e., when boxes are spatially disjoint), the soft relaxation provides a non-zero gradient that smoothly "pulls" the disconnected boxes toward each other, allowing the optimizer to successfully learn the hierarchical topology.

3.3.5 Modeling Probabilities and Logical Entailment

The core mathematical advantage of embedding concepts as boxes of finite volume is the ability to construct a rigorous probabilistic model over the latent space. Under a uniform measure, the marginal probability of a concept is directly proportional to its spatial volume. Crucially, this topology natively supports the calculation of conditional probabilities, which serve as the mechanism for modeling logical entailment and ontological constraints. The conditional probability of concept A given concept B is calculated as the ratio of their intersection volume to the volume of the conditioning box:

$$P(A|B) = \frac{\text{Volume}(A \cap B)}{\text{Volume}(B)}$$

.

This formulation dictates that if the box for child concept A is spatially constrained entirely within the bounds of parent concept B, the intersection volume perfectly equals the volume of A. Consequently, $P(B|A) = 1$, structurally enforcing absolute logical entailment without relying on learned algebraic weights.

Chapter 4

Methodology

This chapter introduces a new extension of CBM called Hierarchical Concept Bottleneck Models. Section 4.1 gives an overview of this new architecture while the following sections explain the details described in the overview. Specifically, Section 4.2 explains how the logical hierarchy is represented and built, Section 4.3 explains the inference strategies provided by the architecture and executed after the concept bottleneck to make the final prediction. Finally Section 4.4 introduces the training details that can be adopted.

4.1 Architectural Overview

To overcome the limitations of standard flat classifiers—namely shortcut learning and concept leakage—this thesis introduces a novel architectural framework. The core hypothesis of this methodology is that spatial topology should not be utilized merely as a distance metric, but as an exact translation of hierarchical and set-theoretic logical rules. By enforcing geometric constraints on the latent space, the model is structurally compelled to respect the underlying ontology of the data.

To achieve this, the proposed architecture employs a decoupled pipeline divided into two distinct phases. Phase 1 focuses exclusively on embedding the logical hierarchy into a geometric latent space, optimizing Box Embeddings to represent concepts and classes as spatial hyper-rectangles without any visual input. Phase 2 trains an independent visual concept extractor to map raw images into concept probabilities. Finally, during inference, these visual probabilities are routed through the frozen, pre-trained geometric space, substituting algebraic linear layers with exact spatial intersections and containment rules.

4.2 Phase 1: Embedding the Logical Hierarchy

4.2.1 Implementation of the Latent Space

The logical hierarchy is modeled by representing each concept and class as an n-dimensional hyper-rectangle (box) within a geometric latent space. To ensure structural validity and prevent negative side lengths during optimization, the boxes are parameterized using a format which defines each box by its minimum coordinate vector (z_{min}) and a strictly positive dimensional length vector (Δ). To break spatial symmetry while ensuring that all concepts initially overlap, the box centers (z_{min}) are initialized using a narrow uniform distribution $\mathcal{U}(-0.01, 0.01)$, whereas the dimensions (Δ) are initialized to a large constant value (e.g., 2.0). This allows the optimizer to smoothly carve out the hierarchical structure by shrinking and shifting the boxes. Because rigid geometric boundaries have zero gradients almost everywhere, the model employs differentiable relaxations to allow gradient-based optimization. Specifically, the spatial overlap between boxes is computed using Gumbel Intersection (with a temperature parameter of $\tau = 0.001$), which provides a smooth, probabilistic approximation of the exact intersection coordinates. Similarly, the volume of these intersections is calculated using a Bessel Approximattion Volume function (with a volume temperature of 0.1 and matching intersection temperature) to maintain stable gradient flow during backpropagation. To prevent boxes from collapsing into infinitely small points or expanding uncontrollably, an L_2 regularization penalty is applied to the side lengths of the boxes during training.

4.2.2 Translating Ontologies into Spatial Constraints

To enforce logical hierarchies within the geometric latent space, the initial unstructured attribute-class matrix must be transformed into a rigorous topological structure. This process serves to construct a structured dictionary of rules—comprising both inclusions and disjunctions—that acts as the primary supervision signal for the geometric optimizer.

Concept Filtering and Graph Construction

The ontology translation begins by pruning the initial concept set to mitigate the risk of the model learning spurious correlations. Concepts that cannot be inferred from the visual characteristics of the image's primary subject are considered noisy and are explicitly filtered out. Some example of these noisy concepts are relational behaviors and environmental backgrounds which often force the network to focus on the image background rather than the primary subject.

Following this refinement, the remaining concepts are organized into a Directed Acyclic Graph (DAG). This hierarchical tree is constructed using a hybrid approach

based on custom rules and WordNet Integration.

Custom rules explicitly maps logical entailment and therefore hierarchical relationship between concepts in order to handle specific domain knowledge.

For concepts lacking manual explicit rules, the structural hierarchy is derived by extracting hypernym paths from WordNet. To maintain the granularity of the domain and prevent the hierarchy from collapsing into overly generic terms, a curated blacklist is utilized to block abstract parent nodes such as "entity," "abstraction," or "physical property". Finally all disjointed branches and top-level concepts are connected to a singular root node to ensure a fully cohesive graph.

Deriving the Supervision Rule Dictionary

The finalized hierarchical tree is then parsed to extract the numerical constraints required to optimize the Box Embeddings. The DAG topology is translated into a ground truth conditional probability matrix, $P(x|y)$, which indicates the spatial intersections in the latent space. The translation from the hierarchical tree to the conditional probability matrix is executed following two rules. The first one represents the relationship between parent and child concepts: given the child concept the conditional probability of its parent is 1. Geometrically, this forces the optimizer to ensure that the hyper-rectangle of a child concept is strictly contained within the spatial volume of its parent. The second rule instead indicates which concepts are mutually exclusive. As an arbitrary rule to model this disjunction we considered mutually exclusive the concepts that share the same direct parent. These mutually exclusive pairs are assigned a target probability of 0.0, acting as negative samples that force the optimizer to physically separate their respective boxes, preventing spatial overlap.

Matrix Expansion via Logical Propagation

Once the topology of the DAG is established, the original binary incidence matrix is expanded horizontally to incorporate the newly discovered parent nodes. The class-concept memberships are updated through a bottom-up topological propagation. By iterating from the leaf nodes up toward the root, a logical OR operation is applied: if a specific class exhibits at least one child concept, it is mathematically enforced to also possess the corresponding parent concept. This matrix is used as ground truth to train the box embeddings simultaneously optimizing two distinct relational axes:

1. **Concept-to-Concept Hierarchy:** Optimizes the conditional probability $P(C_{container}|C_{contained})$, enforcing that child concepts are spatially contained within parent concepts.

2. **Class-to-Concept Membership:** Optimizes the conditional probability $P(Class|Concept)$, enforcing that specific classes are bounded within the intersection of their defining attributes.

The optimization process utilizes Binary Cross-Entropy (BCE) to minimize the divergence between the theoretical ground truth and the predicted spatial probability, calculated as $P(A|B) = \frac{Volume(A \cap B)}{Volume(B)}$. By independently training this hierarchy, the geometric space strictly encodes the predefined ontology prior to any interaction with visual data.

4.3 Inference Strategies: Defining the Classifiers

Once the logical hierarchy is embedded into the geometric latent space (Phase 1), it serves as the foundational structure for the final classification layer. This layer is responsible for mapping the visual concept probabilities—extracted from the image by the visual module detailed in Section 4.4—to the final target class (Y). The Concept Bottleneck Classifier defines multiple classifier variants to evaluate the impact of structural rigidity on the model's reasoning.

4.3.1 Combining Box Embeddings and Linear Layers

Initial hybrid strategies attempt to fuse the geometric features with standard algebraic classifiers as illustrated in Figure 4.1. In these configurations, the pre-trained box parameters or the theoretical probability matrices are scaled by the CNN's predicted concept probabilities. These scaled geometric representations are then flattened and fed into a standard Linear layer. However, empirical observations reveal that utilizing the Box outputs merely as features for a flat linear layer reintroduces the vulnerabilities associated with shortcut learning. The linear layer's capacity to assign negative weights allows it to algebraically compensate for incorrect visual concepts, bypassing the strict geometric topology and severely compromising the model's faithfulness.

4.3.2 Strict Logical Containment (The Dynamic Box)

The most mathematically rigorous inference strategy is the Dynamic Box, which replaces all algebraic weights with a pure geometric AND-logic. In this paradigm, the visual image is transformed into its own dynamic spatial volume (B_{img}) constrained by the established hierarchy. For each concept, the bounds of the image box are dynamically relaxed based on the CNN's certainty (p). If $p \approx 1$, the bounds tightly match the predefined concept box (z_c, Z_c). If $p \approx 0$, the bounds are pushed

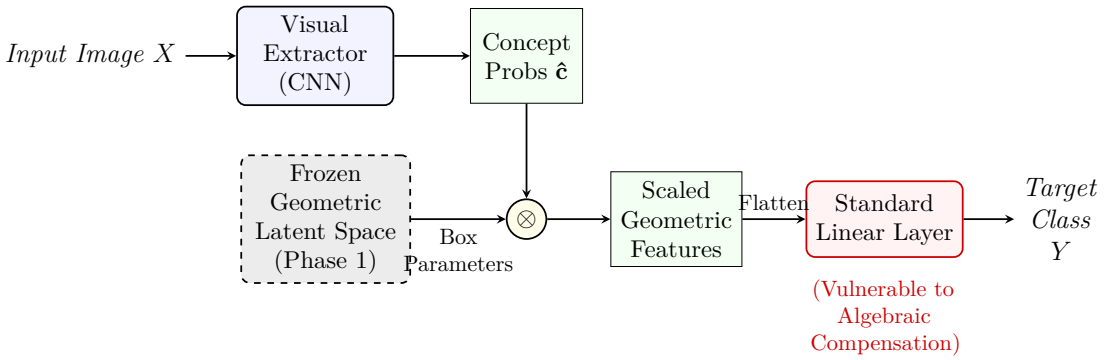


Figure 4.1: Architectural overview of Hybrid Approaches. Visual concept probabilities act as scaling factors for the pre-trained geometric features. However, the resulting representations are aggregated by an unconstrained flat linear layer, functionally reintroducing the vulnerability to shortcut learning.

outward to a predefined universe boundary ($M = 10.0, -M = -10.0$), effectively ignoring the absence of the concept:

$$z_{relaxed} = z_c \cdot p + (-M) \cdot (1 - p)$$

$$Z_{relaxed} = Z_c \cdot p + M \cdot (1 - p)$$

As shown in 4.2 the overall image box (B_{img}) is then computed as the strict intersection of all relaxed concept bounds, achieved by taking the maximum of the minimums (z_{img}) and the minimum of the maximums (Z_{img}) across all dimensions. The final classification logit for each class is derived purely geometrically by computing the conditional probability $P(B_{img}|Class)$, represented as the log-volume of the intersection between the target class box and the dynamic image box, normalized by the class volume. This structural rigidity ensures that the classification is perfectly faithful to the bottleneck, entirely eradicating shortcut learning by making algebraic compensation mathematically impossible.

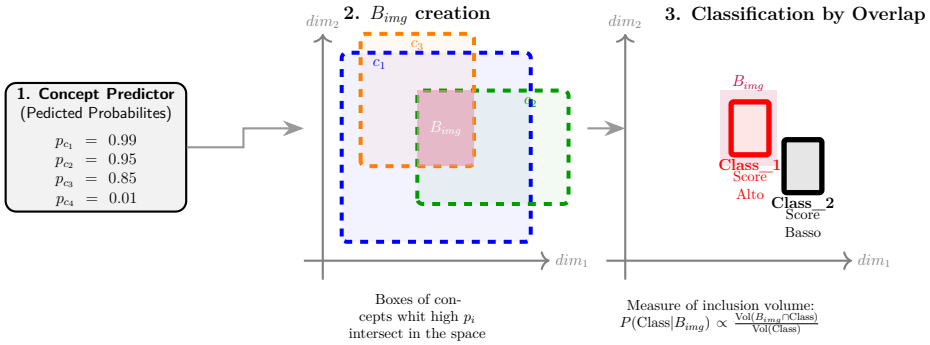


Figure 4.2: Phases 1 & 2: The CNN predicts the animal's traits with high confidence. The probabilities act as "sculptors" on the latent space. Since the probabilities for "c1," "c2," and "c3" are very high, the image box (B_{img}) physically contracts to the exact intersection region (the purple volume) of these three pre-trained concepts. Absent concepts (e.g., wings) push their boundaries to infinity and are ignored. Phase 3: The final class boxes are frozen (from Phase 1). The Class 2 box is far from B_{img} because, while Class 2 has concept 1, it has not concept 2 for example (it does not share the volume). Class 1 box, conversely, is perfectly enclosed within B_{img}. The volume calculation will identify Class 1 as the predicted class in a geometrically rigorous manner.

4.4 Phase 2: Visual Concept Extractor and Training Paradigms

With the classifier structures defined, the second phase of the decoupled pipeline involves training a Convolutional Neural Network (CNN) backbone to map raw input images (X) to the intermediate semantic concepts (C). The Concept Predictor module utilizes established vision architectures, such as ResNet-18 or ResNet-101, to output logits for each concept. Depending on the chosen classifier (Linear or Geometric), the optimization of this visual extractor and its connection to the final layer can follow different training paradigms.

4.4.1 Independent and Sequential Training

The most foundational approach isolates the concept predictor from the final task. During Independent Training, the CNN is optimized using standard Binary Cross-Entropy (BCE) against the ground-truth concept annotations, effectively learning to identify the presence or absence of visual attributes purely from the image features. Once the concept predictor is trained, the final classifier can be introduced. For models utilizing linear layers, this can be done via Sequential Training: the pre-trained concept predictor generates predicted concepts ($\hat{C}$) for the training set, and the linear classifier is subsequently trained to map these imperfect predictions to the final class labels (Y). Conversely, for purely geometric classifiers (which possess no learnable algebraic weights), sequential optimization is not applicable; they strictly rely on the independently trained CNN to supply visual probabilities for inference.

4.4.2 Joint Training for Linear Classifier

To improve overall task accuracy, architectures with linear classifier can be optimized via Joint Training. In this paradigm, both the concept predictor and the linear classifier are trained end-to-end simultaneously. The total loss is a weighted sum of the concept BCE loss and the final classification Cross-Entropy loss. While this allows the CNN to adjust its feature extraction based on task requirements, it introduces the critical vulnerability of Concept Leakage, as arbitrary task gradients backpropagate through the linear layer and pollute the semantic purity of the bottleneck.

4.4.3 Frozen Geometric Joint Training

In this novel setup introduced by this thesis, the CNN is trained jointly based on the final classification loss, but the geometric classifier remains entirely frozen in

evaluation mode. Because the geometric classifier is constructed using differentiable relaxations (such as Gumbel intersections), the classification loss gradients can backpropagate through the rigid spatial logic back to the CNN without altering the underlying ontology. To ensure gradient stability during this process, the geometric logits are scaled by a dedicated temperature parameter before calculating the loss. Furthermore, to guide the CNN in respecting the hierarchical constraints during joint optimization, two specialized loss functions are integrated alongside the task loss:

1. **Weighted Concept Loss:** A dynamic weighting mechanism applies an exponential decay based on the hierarchical height of the concept, forcing the CNN to prioritize accurate predictions for foundational root concepts over granular leaf nodes. Defining h_i as the height of the i-th concept within the Directed Acyclic Graph (DAG) and α as the decay hyperparameter, the weight w_i is calculated as:
$$w_i = e^{\alpha \cdot h_i}$$
This weighting is then applied to the standard Binary Cross-Entropy (BCE). For a batch of size N and K concepts, the weighted conceptual loss is formalized as:
$$\mathcal{L}_{\text{W-BCE}} = \frac{1}{N \cdot K} \sum_{b=1}^{N} \sum_{i=1}^{K} w_i \cdot \text{BCE}(\hat{z}_i^{(b)}, y_i^{(b)})$$
where $\hat{z}_i^{(b)}$ is the logit predicted by the model and $y_i^{(b)}$ represents the ground-truth label for the i-th concept.

2. **Hierarchical Concept Loss:** A topological penalty that actively suppresses logical violations. If the CNN predicts a higher probability for a child concept than its mandatory parent, the squared violation is penalized, scaled by the theoretical conditional probability extracted from the frozen geometric space. Mathematically, this relationship is expressed as:
$$\mathcal{L}_{\text{hier}} = \frac{1}{N} \sum_{b=1}^{N} \sum_{i=1}^{K} \sum_{j=1}^{K} P(c_i|c_j) \cdot \max\left(0, \hat{p}_j^{(b)} - \hat{p}_i^{(b)}\right)^2$$
where $\hat{p}_j^{(b)}$ and $\hat{p}_i^{(b)}$ are the predicted probabilities. The term $P(c_i|c_j)$ acts as a structural mask derived from the underlying ontology, ensuring that penalties are strictly applied only where a genuine topological inclusion constraint exists.

During Frozen Geometric Joint Training, the network is optimized by minimizing a total loss function that is a linear combination of the classification loss (calculated using geometric operations) and the two aforementioned conceptual regularizations:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_c \mathcal{L}_{\text{W-BCE}} + \gamma_{\text{hierarchy}} \mathcal{L}_{\text{hier}}$$

where the hyperparameters λ_c and $\gamma_{\text{hierarchy}}$ control the balance between the predictive accuracy of the final task and strict adherence to the geometric topology.

4.4.4 Addressing Concept Leakage

An architectural advantage of the Dynamic Box inference strategy is that it mitigates concept leakage during joint training regimes. As established in Chapter 3 Section 3.2.2, what happens to standard Concept Bottleneck Models is that during joint optimization, the backpropagation of the final task loss actively alters the intermediate concept representations, frequently sacrificing semantic purity to maximize classification accuracy. In this standard paradigm, the final linear layer acts as a highly parameterized algebraic conduit, allowing task-specific, non-semantic visual artifacts to bypass the intended meaning of the bottleneck and pollute the concept logits.

Conversely, the Dynamic Box approach operates as a strictly geometric, parameter-free classifier. Once the spatial configuration of the Box Embeddings is optimized to satisfy the hierarchical ontology (Phase 1) and subsequently frozen, the final inference process relies exclusively on topological operations—namely, spatial intersections and volume calculations based on rigid logical AND constraints. Because there are no intermediate, trainable algebraic weights bridging the concept bottleneck and the final target class, the structural pathway required for spurious gradient exploitation is effectively severed.

If the proposed architecture is subjected to joint training, the gradients propagating backward from the classification loss cannot alter the logical rules of the frozen latent space. Instead, they are strictly constrained to refine the visual extractor's predictive accuracy while forcing it to adhere to the predefined spatial ontology. Rather than bending the semantic meaning of the concepts to accommodate visual anomalies, the network is forced to learn better, more robust visual representations. Consequently, the Dynamic Box framework guarantees that the semantic integrity of the bottleneck remains strictly faithful to the original knowledge graph, definitively resolving the concept leakage dilemma by design rather than by heuristic approximation.

Chapter 5

Experiments and Results

This chapter shows the experiments that we conducted and the results achieved along with their interpretation[1]. Section 5.1 describes the preprocessing necessary to construct the concept hierarchy, the metrics used to evaluate the model and the training details and hyperparameters. In Section 5.2 we show the results achieved by models with Linear Classifier, we suggest the hypothesis that the Linear Classifier could have benefit of shortcut learning to achieve them and we support this hypothesis with some analysis of the weights of the Linear Classifier. In Section 5.3 we show the results for classification of the model with the Dynamic Box Classifier. Finally in Section 5.4 we show the results for Zero Shot Learning and the analysis made to interpret these results.

5.1 Experimental Setup

5.1.1 Datasets and Concept Ontologies

To evaluate the proposed hierarchical architecture and its zero-shot transfer capabilities, the primary experiments are conducted using the Animals with Attributes 2 (AwA2) dataset. AwA2 is a widely adopted benchmark in both Zero-Shot Learning (ZSL) and interpretable computer vision. The dataset comprises 37,322 images distributed across 50 animal classes. Each class is annotated with a predefined set of 85 semantic attributes (concepts), ranging from anatomical features (e.g., "stripes", "paws") to behavioral traits and habitats.

Crucially, the original dataset provides these 85 concepts in a completely flat, unstructured format via a continuous or binary predicate matrix. In accordance

[1]The code to reproduce the experiments is available at github.com/stefano-gamba/BoxEmbeddingCBM

with the objective of enforcing strict logical rigor, we utilize the provided predicate-matrix-binary rather than the continuous version. However, this flat binary matrix lacks the explicit taxonomic relationships required by the hierarchical Box Embedding architecture. Therefore, to map the dataset into a structured ontology, the AwA2 concepts are systematically manipulated following the methodology outlined abstractly in Section 4.2.2.

This transformation is executed through an automated pipeline that restructures the flat AwA2 attributes into a rigorous topological graph:

- **Semantic Filtering:** The original set of 85 concepts is first subjected to a pruning process to isolate strictly visually grounded attributes. Concepts deemed "noisy"—such as subjective characteristics (e.g., "smart", "fast"), purely behavioral traits (e.g., "nocturnal", "hibernate"), dietary habits, and environmental backgrounds (e.g., "jungle", "ocean")—are removed. This filtering prevents the model from hallucinating features or relying on background pixels rather than the primary subject, thereby mitigating spurious correlations

- **Taxonomic Graph Construction:** The remaining valid concepts are organized into a Directed Acyclic Graph (DAG) as illustrated in Figure 5.1. Following the approach detailed in Section 4.2.2, this tree is built primarily by querying WordNet for hypernym paths, specifically filtering out overly abstract parent nodes to maintain domain relevance. To ensure structural accuracy, this is augmented with custom anatomical mappings specific to AwA2 (for instance, explicitly defining "paws", "hooves", and "claws" as direct descendants of "foot", which in turn is a descendant of "leg"). All distinct branches are unified under a singular "Animal" root node.

- **Incidence Matrix Expansion:** Once the DAG topology is defined, the original AwA2 binary incidence matrix is expanded horizontally to encompass the newly introduced parent concepts. Class-concept memberships are logically propagated bottom-up through the hierarchy. By applying a logical OR operation, any animal class possessing a child attribute (e.g., "flippers") is mathematically forced to also possess its respective parent attributes (e.g., "swims").

- **Structural Constraint Generation:** Finally, the resulting expanded matrix and the DAG are parsed to generate the geometric supervision dictionary. The full set of these generated topological constraints is represented as a target conditional probability matrix as shown in Figure 5.2 left. This dictionary maps the taxonomy into strict spatial constraints: enforcing a target probability of 1.0 for parent-child structural inclusions—requiring the child's bounding box to reside entirely within the parent's box, as visually demonstrated by

the entailment between the concepts flys and wings shown in Figure 5.3. Conversely, it assigns a target probability of 0.0 for sibling disjunctions, ensuring that mutually exclusive features, such as distinct colors like red and orange, are forced to be spatially disjoint, see Figure 5.4.

Through this transformation pipeline, the standard, flat AwA2 dataset is successfully converted into an extended semantic hierarchy, providing the exact topological ground truth required to optimize and evaluate the geometric inference capabilities of the model.

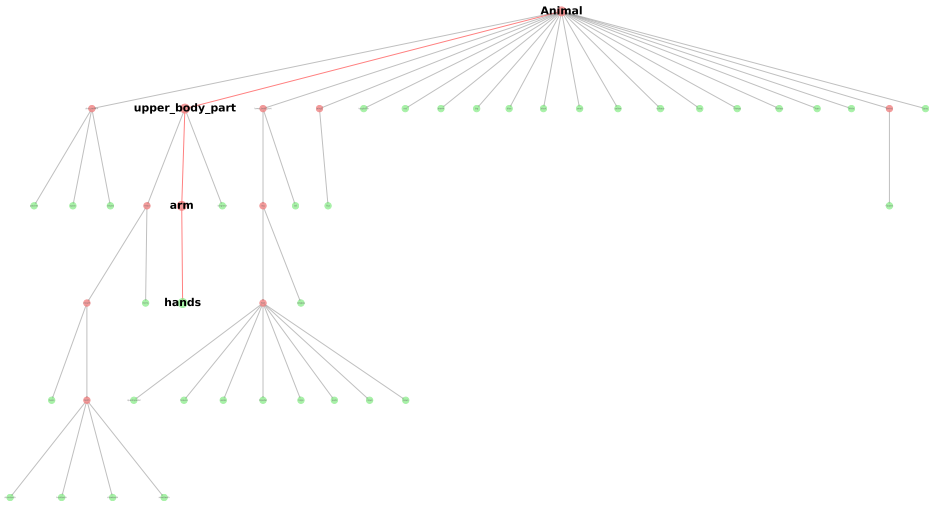


Figure 5.1: DAG representing the AWA2 concept hierarchy after the manipulation described in Section 5.1.1. Due to the extensive scale of the full concept hierarchy DAG, displaying all node labels simultaneously renders the text illegible. To ensure clarity while preserving the overall structural context of the graph, a representative root-to-leaf path has been visually highlighted

5.1.2 Evaluation Metrics

To comprehensively assess the performance of the proposed hierarchical architecture, the evaluation protocol is divided into two primary dimensions: standard classification on seen classes and zero-shot transfer capabilities. All standard classification metrics were computed utilizing the Scikit-learn framework to ensure rigorous and reproducible evaluation.

For the standard in-distribution classification, we report accuracy, precision and recall. For the zero-shot transfer evaluation, standard metrics are often insufficient

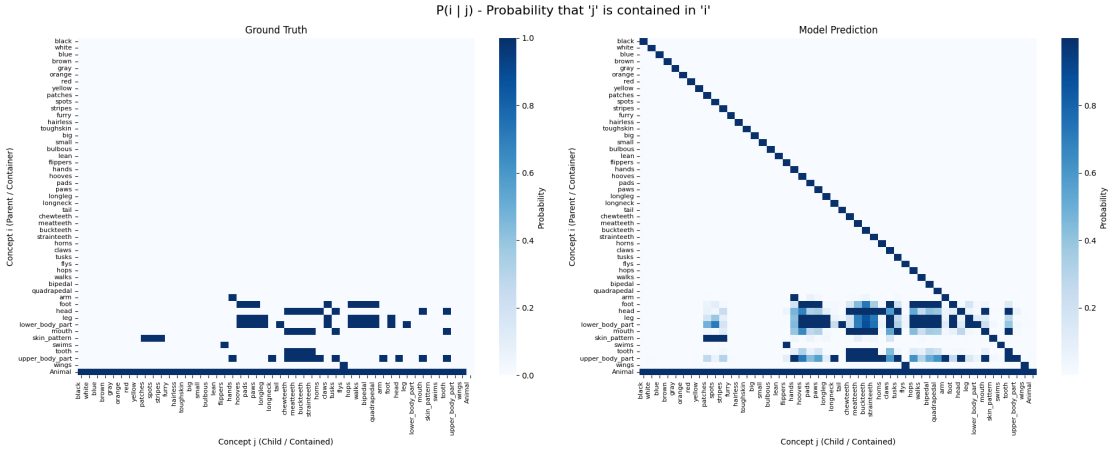


Figure 5.2: Comparison between the Ground Truth topological constraints and the learned spatial probabilities. Left: The target conditional probability matrix $P(i|j)$ generated from the AwA2 ontology, where values of 1.0 indicate strict containment and 0.0 indicate spatial disjunction. Right: The actual spatial intersections predicted by the model after Phase 1 optimization, demonstrating near-perfect alignment with the predefined hierarchy (the diagonal elements represent $P(i|i)$ and because of the nature of the boxes these probabilities will be always equal to 1).

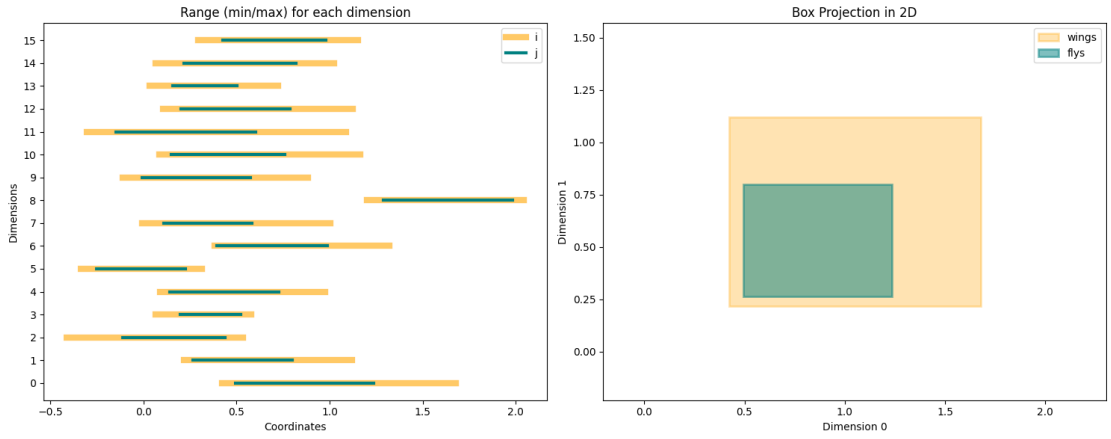


Figure 5.3: Geometric representation of a parent-child structural inclusion. The optimization process strictly forces the bounding box of the child concept (flys, in green) to be contained entirely within the spatial volume of the parent concept (wings, in yellow), thereby guaranteeing the logical entailment $P(wings|flys) = 1.0$ by design.

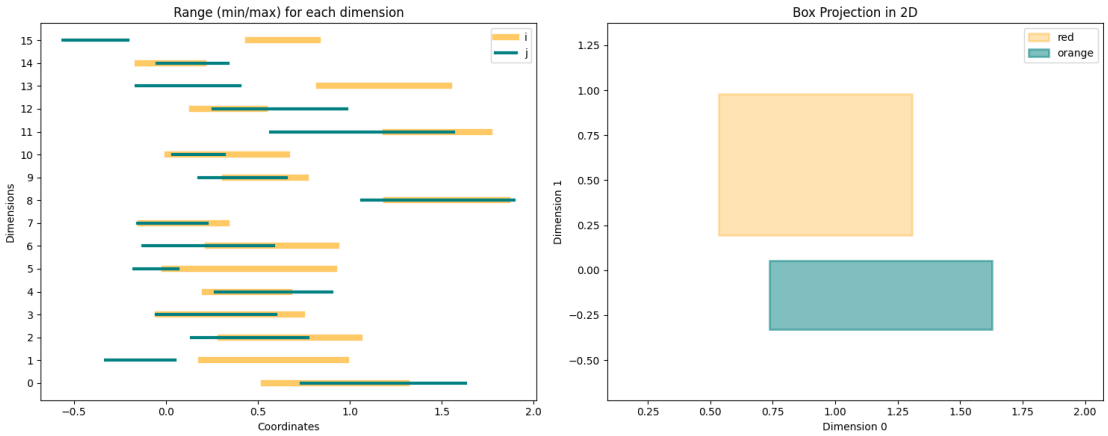


Figure 5.4: Geometric representation of sibling disjunctions. Mutually exclusive semantic traits—such as the colors red (yellow box) and orange (green box)—are subjected to a target intersection probability of 0.0. The optimizer correctly carves the latent space to ensure these concepts are spatially disjoint, structurally preventing impossible attribute combinations.

due to the inherent bias towards seen classes. Therefore, we rely on Standard ZSL Accuracy. Following the benchmarking protocols established by Xian et al. [20], this metric computes the average per-class accuracy exclusively on the unseen test set. By averaging the accuracy independently for each unseen class before computing the final mean, this metric prevents densely populated unseen classes from dominating the evaluation, providing a true reflection of the model's capacity to generalize semantic concepts.

5.1.3 Training Details and Hyperparameters

The entire pipeline, including the custom differentiable box operations and the visual concept extractor, was implemented using the PyTorch framework. The training process followed the decoupled Phase 1 and Phase 2 methodology outlined in Chapter 4.

Phase 1 (Geometric Latent Space): The geometric latent space was implemented using 16-dimensional box embeddings. The topology optimization was performed without visual input using a Binary Cross-Entropy (BCE) loss, driven by the Adam optimizer with a learning rate of 0.05 for a total of 100 epochs. However, the structural scope, the initialization strategies, and the choice of geometric relaxations varied significantly depending on the intended downstream inference strategy.

For Hybrid Approaches the optimization was restricted solely to the concept boxes, enforcing conditional probabilities strictly between concepts (e.g., $P(C_{contained}|C_{container})$) to map the predefined ontology. The model utilized a simplified parameterization: all box parameters (both centers and dimensions) were initialized using a standard uniform distribution $\mathcal{U}(-0.5, 0.5)$. Inference relied on a strict Hard Intersection combined with a Soft Volume formulation (using a volume temperature of 0.1), [24].

For Geometric Classifiers (Dynamic Box): Both the concept boxes and the final class boxes were instantiated and optimized simultaneously. This joint geometric optimization mapped the entire topology at once, enforcing both the concept-to-concept hierarchy and the class-to-concept memberships (where concepts act as topological containers for the classes), which is mandatory for the strict spatial containment required by the Dynamic Box. To ensure smooth gradient flow and structural validity, box centers (z_{min}) were initialized using a narrow uniform distribution $\mathcal{U}(-0.01, 0.01)$, while the initial dimensional lengths (Δ) were set to a large constant value of 3.0 to guarantee initial spatial overlap. Furthermore, this model relied on fully differentiable relaxations: both the Gumbel Intersection and the Bessel Approx Volume were utilized, setting the intersection and volume temperature parameters to 0.1, [24].

In both optimization paradigms, L_2 regularization on the bounding box side lengths was deactivated (weight set to 0.0) to allow the optimizer to freely carve the logical volumes without artificial mathematical penalities.

Phase 2 (Visual Extractor) - Standard Classification on Seen Classes: To establish a fair baseline and evaluate all models (both Hybrid Approaches and Geometric Classifiers) on the standard classification of seen classes, we strictly adhered to the official benchmarking protocols. Instead of training a visual extractor from scratch, we utilized the standardized image embeddings provided with the AwA2 dataset. These features consist of 2048-dimensional vectors extracted from the top-layer pooling units of a 101-layer ResNet pre-trained on ImageNet 1K. As specified in the benchmark guidelines, [21], these embeddings were extracted from the entire image without any additional pre-processing or fine-tuning of the deep neural network. Then the embeddings are mapped into the concept bottleneck by a trainable linear layer.

Phase 2 (Visual Extractor) Zero-Shot Learning (ZSL) & Raw Image Processing: Conversely, to evaluate the true generalization capabilities in the Zero-Shot Learning (ZSL) transfer task, the pipeline operated directly on raw images. The dataset underwent a rigorous data augmentation pipeline during training: images were resized to 256 pixels, randomly cropped to a 224×224 resolution, horizontally flipped, and normalized using standard ImageNet channel statistics (mean $= [0.485, 0.456, 0.406]$, std $= [0.229, 0.224, 0.225]$). During the deterministic testing phase, a standard central crop of 224×224 was applied following the initial resize.

Frozen Geometric Joint Training (Dynamic Box Exclusive): During the ZSL training the Dynamic Box architecture was subjected to an end-to-end joint training paradigm directly on the raw images.

The visual backbone employed was a ResNet-101 initialized with default pre-trained ImageNet weights. To adapt the high-level semantic representations while preserving the stability of the foundational geometric features, we implemented a targeted unfreezing strategy. The network was optimized using the Adam optimizer with a weight decay of 1×10^{-4} and distinct, differential learning rates across the unfrozen blocks: the deeper fully-connected (fc) classification head was updated with a learning rate of 5×10^{-3}, while the earlier convolutional blocks layer 4 and layer 3 were fine-tuned with conservative learning rates of 1×10^{-4} and 1×10^{-5}, respectively. The rest of the backbone remained completely frozen. During this joint training, the visual extractor was trained for 20 total epochs. As detailed in Section 5.4.2, a two-phase curriculum was applied: the first 10 epochs utilized a sharp logit temperature of 0.5 and a hierarchical weight decay of $\alpha = 0.0$, while the subsequent

epochs softened the temperature to 1.0 and increased the topological penalty ($\gamma_{hierarchy}$) to 2.0. The specific hyperparameters utilized during this phase were selected following a rigorous tuning process. To achieve this without compromising the zero-shot evaluation protocol, the hyperparameter search was conducted on a dedicated validation set generated by applying an 80/20 split exclusively to the data instances belonging to the 40 seen training classes. All experiments were accelerated utilizing a T4 GPU.

5.2 Linear Layer Classification Results

5.2.1 Performance Linear Layer Classifier

Before isolating the purely geometric inference strategies, initial experiments evaluated hybrid architectures to establish a baseline for standard CBMs appended with flat classifiers. In these configurations, intermediate representations—either the CNN's predicted concept probabilities (Sequential Training) or the pre-trained box features—were flattened and passed through a standard fully connected linear layer to predict the final target class.

As shown in Table 5.1, the hybrid approaches exhibit ostensibly exceptional performance on the seen training classes. Across all training paradigms (Independent, Sequential, and Joint), the standard CBM and the hybrid variants (CBM + BOX, CBM + MATRIX) consistently achieve raw accuracy scores of approximately 90%.

This high performance on seen classes is a direct result of the linear layer's immense algebraic capacity. Given the rich, high-dimensional intermediate features provided by the concept bottleneck or the geometric space, the final linear classifier easily finds mathematical hyperplanes that perfectly separate the training data. However, while these accuracy metrics appear indicative of a highly effective model, they are structurally deceptive. Because the linear layer is unconstrained by topological rules, it achieves this separation not by respecting the carefully constructed ontology, but by finding the path of least mathematical resistance. This sets a critical trap: the model performs flawlessly on familiar data distributions while silently bypassing the hierarchical logic it was meant to learn.

5.2.2 Exposing Shortcut Learning: Global Flaws and Local Manifestations

While linear classifiers appended to Concept Bottleneck Models (CBMs) may occasionally achieve high classification accuracy on standard test sets, probing their internal decision-making process reveals critical structural flaws. To demonstrate how standard CBM architectures fail to respect the semantic hierarchy and instead

Table 5.1: Performance comparison based on architecture and training mode

Model	**Training**	**Accuracy**	**Precision**	**Recall**
CBM	Independent Training	0.90	0.90	0.87
	Sequential Training	0.91	0.89	0.89
	Joint Training	0.91	0.89	0.89
CBM + BOX	Independent Training	0.90	0.90	0.89
	Sequential Training	0.91	0.89	0.90
	Joint Training	0.90	0.89	0.88
CBM + MATRIX	Independent Training	0.89	0.89	0.86
	Sequential Training	0.91	0.88	0.89
	Joint Training	0.90	0.88	0.89

rely on shortcut learning, it is necessary to inspect the direct algebraic contributions of individual concepts to the final class logits.

Global Logical Violations Analysis

A CBM should assign a positive weight to a concept predicted as belonging to a certain class in order to allow humans to interpret its prediction about the class. To evaluate whether the linear classifier respects this premise globally on all the classes it learned, we analyze its static, learned weight matrix W against the absolute ground-truth ontology (the class-concept matrix). A structural logical violation occurs when the model systematically learns a negative weight (acting as a penalty, $w < 0$) for a concept that is objectively present in the ground-truth definition of a class, effectively creating a "Logical False Negative". As illustrated in Figure 5.5, the linear classifier produces several systemic violations that reduce the likelihood of correctly interpreting its output. The bar chart ranks the classes most heavily affected by these logical shortcuts. The vast majority of these violations are Logical False Negatives (red bars). This proves that across the entire dataset, the linear model systematically learns to penalize true semantic attributes. Instead of learning generative class definitions, the model treats shared hierarchical concepts (e.g., being an animal, having legs) as algebraic liabilities, applying negative weights to force arbitrary mathematical separation between classes in the latent space. Crucially, the geometrically constrained Dynamic Box approach proposed in this thesis strictly enforces structural inclusions, resulting in zero logical violations globally by mathematical definition.

Local Manifestations of Shortcut Learning

In a standard sequential CBM, the logit z_y for a target class y is computed as the weighted sum of the predicted concept probabilities $\hat{c}_i$:

$$z_y = \sum_{i=1}^{k} w_{y,i} \cdot \hat{c}_i + b_y$$

where $w_{y,i}$ represents the learned weight connecting the i-th concept to class y.

To demonstrate how these global structural flaws manifest during actual inference we isolate the specific algebraic contributions, the term $w_{y,i} \cdot \hat{c}_i$ indicating the exact positive or negative force that a specific concept exerts on the final prediction.

An analysis of these contributions across different test samples (Figure 5.6, 5.7, 5.8) exposes three distinct and severe manifestations of shortcut learning: algebraic compensation, taxonomic violation, and the absence of logical vetoes.

Figure 5.6 illustrates a misclassification where a chimpanzee is incorrectly predicted as a horse. The decision is driven by a mathematical balancing of statistical

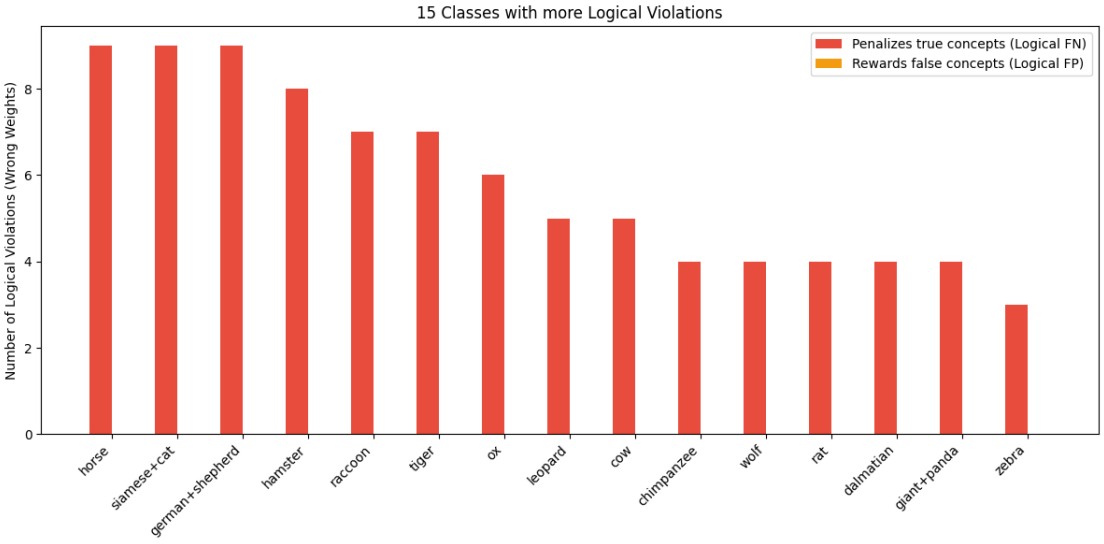


Figure 5.5: Logical Violations per classes proves that linear layer uses shortcut learning.

weights rather than coherent logical reasoning. The underlying visual concept extractor generates severe mispredictions, such as assigning a 0.76 probability to the concept "buckteeth" The flat linear layer blindly accepts this incorrect prediction, applying a positive weight to it. Simultaneously, the model correctly identifies the presence of "meatteeth" (probability 0.54), which correctly applies a strong negative penalty against the "horse" class. To overcome this negative evidence and force the prediction, the linear layer mathematically compensates by relying on the incorrect features. The prediction is effectively the result of an algebraic process that is difficult to be interpreted by humans not experts in mathematichs and machine learning.

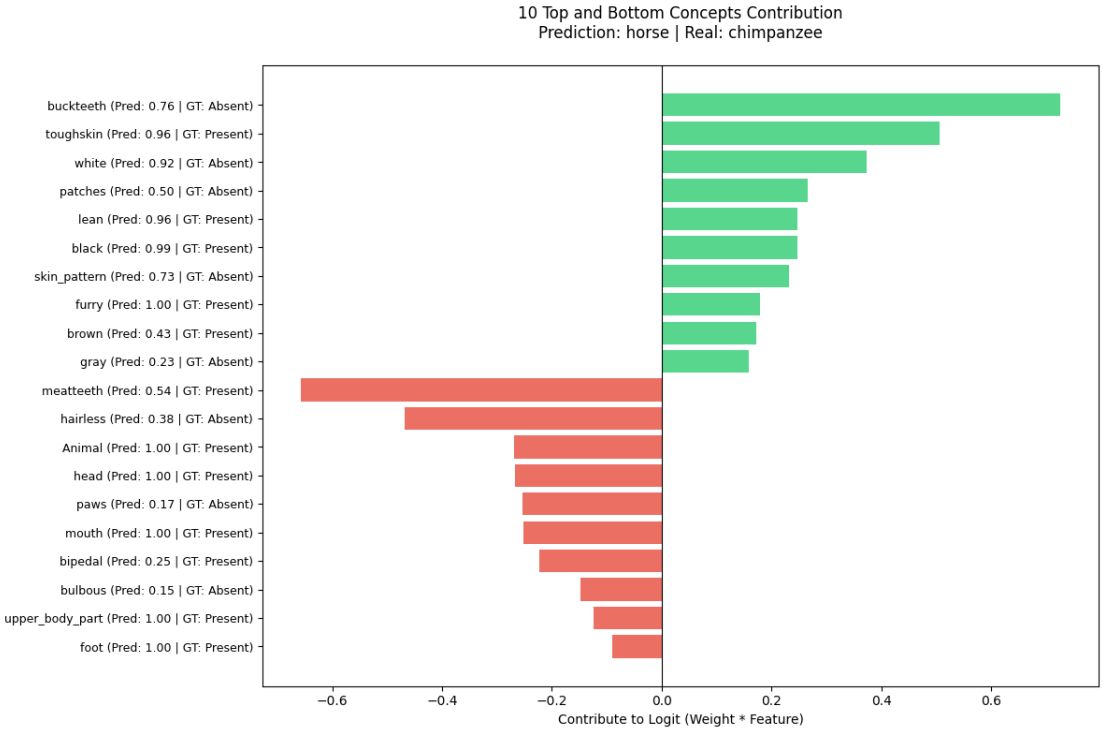


Figure 5.6: Example algebraic compensation made by the Linear Classifier. To overcome this negative evidence given by 'meatteath' and to force the prediction, the linear layer mathematically compensates by relying on the incorrect features, like 'buckteeth'.

The divergence from human logic of linear classifiers becomes even more evident when analyzing correctly predicted samples. Figure 5.7 shows a successful prediction of the giraffe class. However, undeniable structural tautologies such as the presence of mouth, head, and leg are utilized as negative algebraic weights to penalize the logit, despite being perfectly predicted with a 1.00 probability. From a semantic

perspective, this is a logical paradox: the confirmed presence of a defining anatomical trait should strictly support the classification. This contradiction arises because the flat linear layer learns a purely discriminative decision boundary rather than a generative ontological definition. Because traits having a leg or having the mouth are shared among competing classes, the linear layer algebraically penalizes these shared attributes to artificially force a wider mathematical separation in the latent space. By penalizing class membership based on indisputable biological tautologies, the standard CBM violates the taxonomic hierarchy.

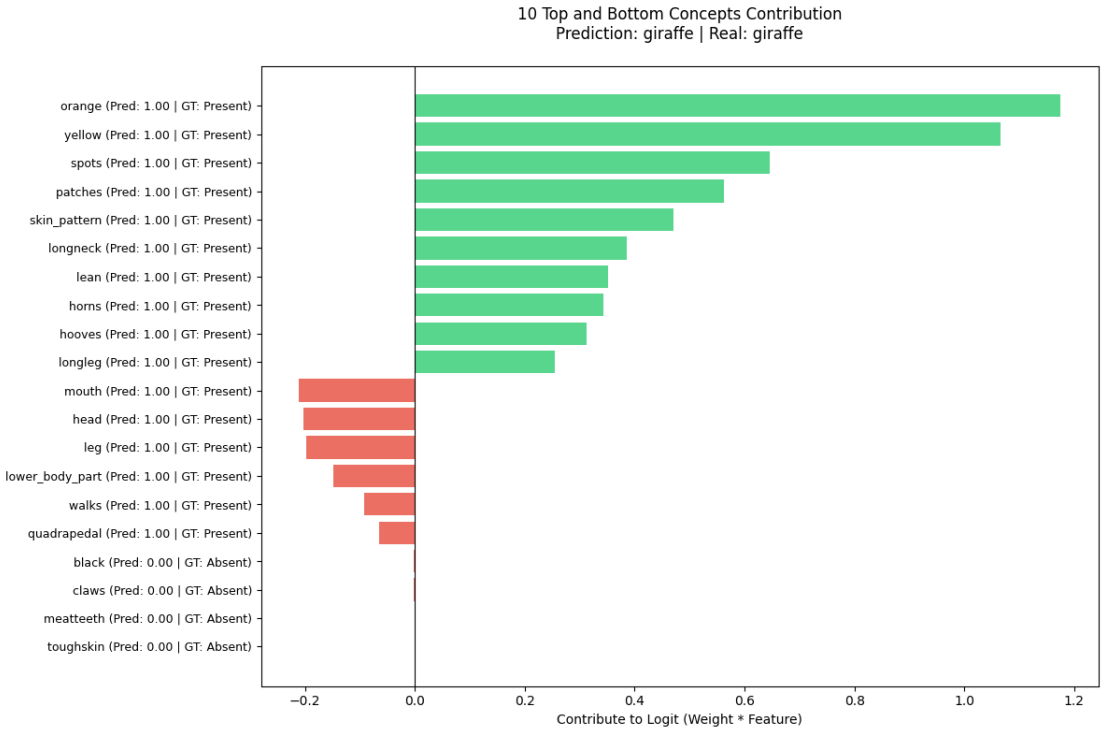


Figure 5.7: Example of taxonomic violation produced by the Linear Classifier. The presence of mouth, head, and leg are utilized as negative algebraic weights to penalize the logit, despite being perfectly predicted with a 1.00 probability

A final, critical vulnerability is the linear classifier's inability to enforce strict logical negations, or "logical vetoes." In formal ontologies, certain semantic attributes act as absolute disqualifiers. Figure 5.8 exemplifies this failure in the misclassification of a blue whale as a killer whale. The model confidently identifies the fundamental attribute blue (probability 0.97), and the linear classifier correctly applies a negative weight to it, knowing killer whales are not blue. Logically, the definitive presence of the color blue should immediately veto the killer whale class. Yet, the linear model overrides this penalty by summing the positive contributions

of shared anatomical features (e.g., swims, strainteeth) and strongly incorrect attributes (e.g., white, lean). Because the linear layer relies on unconstrained additive aggregation, the accumulated positive mass algebraically overpowers the correct negative evidence.

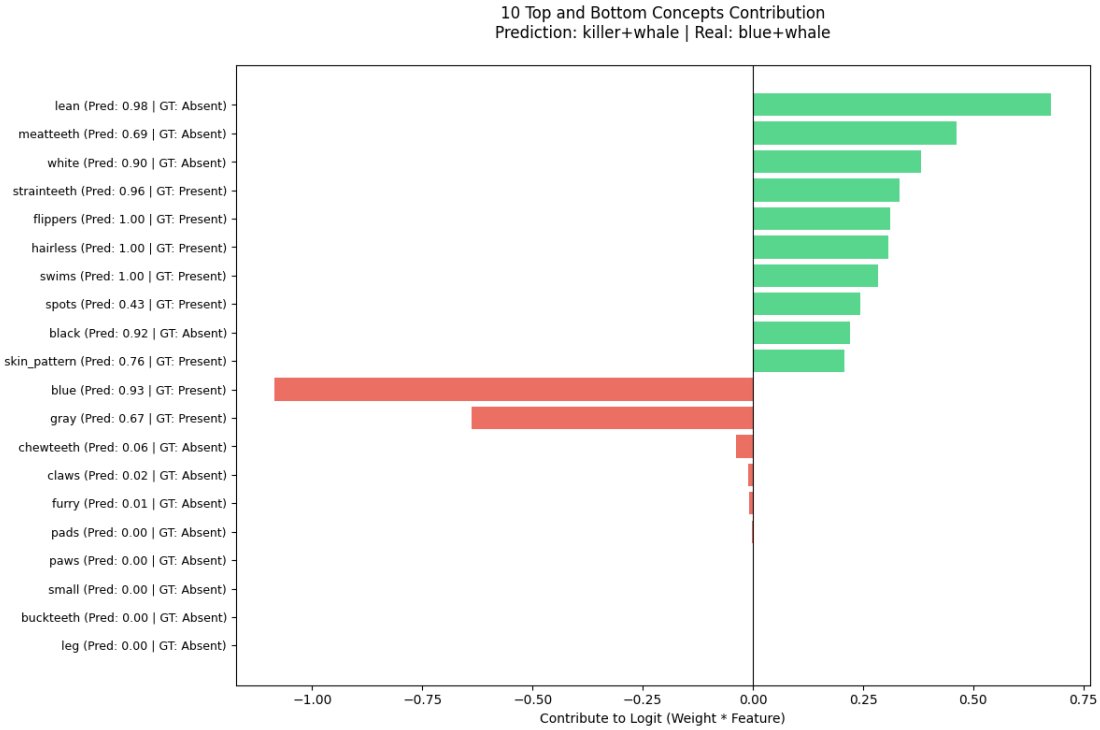


Figure 5.8: Example of inability of the Linear Classifier to impose logical vetoes. The model confidently identifies the fundamental attribute blue (probability 0.97), and the linear classifier correctly applies a negative weight to it, Yet, the linear model overrides this penalty.

These three phenomena suggest that flat linear classifiers are incompatible with strict hierarchical logic. They achieve their task by finding efficient statistical shortcuts, sacrificing the model's faithfulness. This structural limitation directly motivates the transition to the geometric framework proposed in this thesis, where strict logical entailment—enforced through spatial inclusions and zero-volume intersections—inherently prevents algebraic compensation and structurally preserves the concept hierarchy.

5.3 Dynamic Box Classification Results

Before evaluating the zero-shot transfer capabilities of the architecture, it is necessary to benchmark the hierarchical inference strategies on the standard test set containing seen classes. Crucially, because Dynamic Box classifiers relies on a static, geometrically optimized latent space that remains frozen after Phase 1, the distinction between Independent and Sequential training paradigms is effectively neutralized during inference. In both training modes, the visual Concept Predictor generates a vector of probabilities without receiving backpropagated gradients from the final classification loss. Consequently, benchmarking this model directly isolates the impact of the spatial inference logic, allowing for a pure comparison between algebraic learning and strict geometric containment.

5.3.1 Linear Layer vs. Dynamic Box

The empirical evaluation on the seen classes reveals a fundamental trade-off between robustness to visual noise and the enforcement of strict logical rigor. As illustrated in Table 5.2, the Dynamic Box strategy achieves an overall accuracy of 73.38%

Table 5.2: Performance comparison based on architecture and training mode.

Model	Training	Accuracy	Precision	Recall
Best Linear Layer	Independent Training	0.90	0.90	0.89
Dynamic Box	Independent Training	0.73	0.71	0.78

It maintain a strong diagonal in its confusion matrices, demonstrating that it is possible to achieve still decent classification performance without relying on the unconstrained algebraic shortcuts of a flat linear layer.

The Dynamic Box approach replaces additive weights with strict geometric AND-logic. During inference, the model translates the visual probabilities into a dynamic spatial volume representing the image. The final classification relies on the exact intersection volume between this dynamic image box and the predefined class boxes, calculated via Gumbel Intersections and Bessel Approximations. While this strategy entirely eradicates algebraic compensation and guarantees absolute faithfulness to the semantic ontology, its structural rigidity makes it highly sensitive to visual errors. Because it relies on spatial intersections, if the CNN severely misidentifies even a single critical attribute, the dynamic bounds of the image box may shrink or shift inappropriately. This can cause the intersection volume with the correct class box to collapse abruptly, explaining the minor 0.9% decrease in overall accuracy compared to the additive approach.

5.4 Zero-Shot Learning (ZSL) Transfer

5.4.1 ZSL Protocol and Baselines

Zero-Shot Learning (ZSL) aims to classify instances from classes that the model did not see during the training phase. This is accomplished by utilizing an intermediate semantic representation, that could be a set of human-annotated attributes for example, to bridge the knowledge gap between a set of seen classes, used for training, and a disjoint set of unseen classes, used for testing. The Animals with Attributes 2 (AwA2) dataset was specifically constructed to benchmark this capability, providing a rich, 85-attribute predicate matrix that acts as the essential semantic side information connecting 40 training classes to 10 distinct test classes.

The primary motivation for pursuing ZSL is to test the ability of recognizing a new class simply by reading a semantic description of its features. From a practical standpoint, ZSL is crucial for scalability and real-world deployment, as it circumvents the need to collect and annotate massive volumes of visual data for every conceivable class, particularly in cases where some categories are exceptionally rare, [20], [21].

Because CBMs inherently map raw visual inputs into a strictly defined semantic space, their architecture is naturally aligned with the ZSL objective. More importantly, the zero-shot protocol serves as empirical test for generalization and faithfulness. If an interpretable model has genuinely learned the causal semantic definition of the concepts, rather than merely memorizing statistical artifacts, it should be capable of recombining those concepts to identify unseen classes.

However, standard CBMs equipped with flat linear classifiers face a fundamental architectural barrier in zero-shot scenarios: dimensional incompatibility. A linear layer trained to map K concepts to N seen classes inherently learns a weight matrix of size $N \times K$. To classify M entirely novel classes during inference, this matrix is mathematically obsolete; it possesses neither the output dimensions nor the learned weights required for the unseen categories. The linear model cannot project its reasoning onto novel classes without undergoing a new training phase, which fundamentally violates the zero-shot premise. Consequently, standard CBM architectures are forced to entirely discard their learned classification layer during ZSL evaluation. To perform inference, they must fall back on uncalibrated, non-parametric heuristic metrics—such as Cosine Similarity or Nearest Neighbors—to compare the predicted concept vectors against the ground-truth binary signatures of the unseen classes. This reliance on heuristic baselines rather than the model's actual inference engine completely bypasses the classifier's learned behavior, exposing its inability to naturally generalize.

The hierarchical Dynamic Box architecture proposed in this thesis was developed specifically to overcome this limitation. By completely replacing algebraic weights

with strict spatial intersections and containment rules, the geometric latent space is forced to learn absolute structural logic and natively supports zero-shot expansion. If the topological rules embedded in the latent space, such as the parent-child entailment, or the strict spatial disjunction, are universal constants they will hold true regardless of the specific class being observed. Because the classification of an unseen class in the Dynamic Box approach relies purely on calculating the geometric intersection between the dynamically generated image volume and the theoretical unseen class volume, the model requires zero algebraic recalibration. This strict logical rigidity prevents the exploitation of spurious correlations, enabling more robust zero-shot transfer.

5.4.2 Zero-Shot Performance Comparison

To rigorously evaluate the generalization capabilities of the proposed architecture, we benchmark the geometrically constrained inference against the standard Zero-Shot Learning (ZSL) baseline for Concept Bottleneck Models: the Cosine Similarity approach, [25], [26]. In the baseline configuration, the final classification is determined by computing the maximum cosine similarity between the CNN's predicted concept vector and the binary ground-truth attribute vectors of the unseen classes, effectively treating all concepts equally within a flat vector space.

Conversely, our proposed model evaluates the unseen classes by replacing the algebraic vector comparison with the Dynamic Box inference strategy. The Dynamic Box model was trained using a two-phase joint training curriculum.

The first 10 epochs focus on developing a highly accurate visual concept extractor. During this phase, the concept loss is designed in order to outweighs the logical constraints ($\lambda_c = 2.0$, $\gamma_{hierarchy} = 0.1$). To force the CNN to be confident in making the definitive visual decisions, the temperature scaling on the logits is set to a sharp $logit_temp = 0.5$. The hierarchical weight decay is balanced ($\alpha = 0.0$), giving equal learning priority to root and leaf concepts.

In the subsequent 10 epochs, the network is forced to align its visual predictions with the frozen geometric latent space. The logical penalty is increased to $\gamma_{hierarchy} = 2.0$ in order to enforce strict topological compliance. However, to allow the dynamic image box to smoothly intersect with the predefined class boxes without suffering from vanishing gradients, the logit temperature is softened to $logit_temp = 1.0$. This combination of a strict logical constraint with softer probabilistic boundaries enables the model to find the optimal geometric intersections.

The results of this evaluation support the core hypothesis of this thesis. As illustrated in the resulting confusion matrices, the baseline Cosine Similarity approach achieves a standard ZSL accuracy of 46.29% (see Table 5.3).

In contrast, the Dynamic Box Joint Training model achieves a standard ZSL

accuracy of 52.09%, establishing a substantial improvement of +5.8% over the baseline. By substituting unconstrained additive geometry with strict logical containment, the model structurally prevents impossible attribute combinations, forcing the latent representation to respect the global ontology even when evaluating entirely novel classes.

Table 5.3: Performance comparison of zsl based on architecture and training mode.

Model	Training	Mean Class Accuracy
Cosine Similarity	-	0.46
Dynamic Box	Independent	0.49
	Joint	**0.52**

5.4.3 Topological Insights on Unseen Classes

A granular analysis of the normalized confusion matrices (Figure 5.9, Figure 5.10), augmented by the visualization of Concept Error Rates (Figure 5.11, Figure 5.12) and Uncertainty Maps (Figure 5.13, Figure 5.14) provides profound insights into how geometric inference manipulates the decision-making process. The transition from a flat vector space, Figure 5.9, to a hierarchically constrained topological space, Figure 5.10, yields three distinct observable phenomena: the stabilization of intermediate classes, the clustering of ambiguous nodes, and the manifestation of feature fragmentation.

The Stabilization of Intermediate Classes The most significant victories of the Dynamic Box approach occur in classes defined by broad, overlapping anatomical traits. In the Cosine Similarity baseline, the Hippopotamus achieves a mere 39% accuracy, heavily confused with the Rat (32%). Similarly, the Pig achieves only 19%, misclassified across a wide array of mammals.

The rising to 88% of the accuracy for the Hippopotamus, and to 52% for the Pig in the Dynamic box suggests that the enforcing of the hierarchical logic could have helped avoiding these misclassifications. The main argument to support this hypothesis is that the geometric AND-logic ensures that the spatial volume representing these classes acts as an exclusive topological region therefore preventing classes with conflicting parent-nodes from invading their latent space.

The Clustering of Ambiguous Nodes While strict logic resolves macroscopic errors, it highlights the limits of visual extraction on granular details. In the

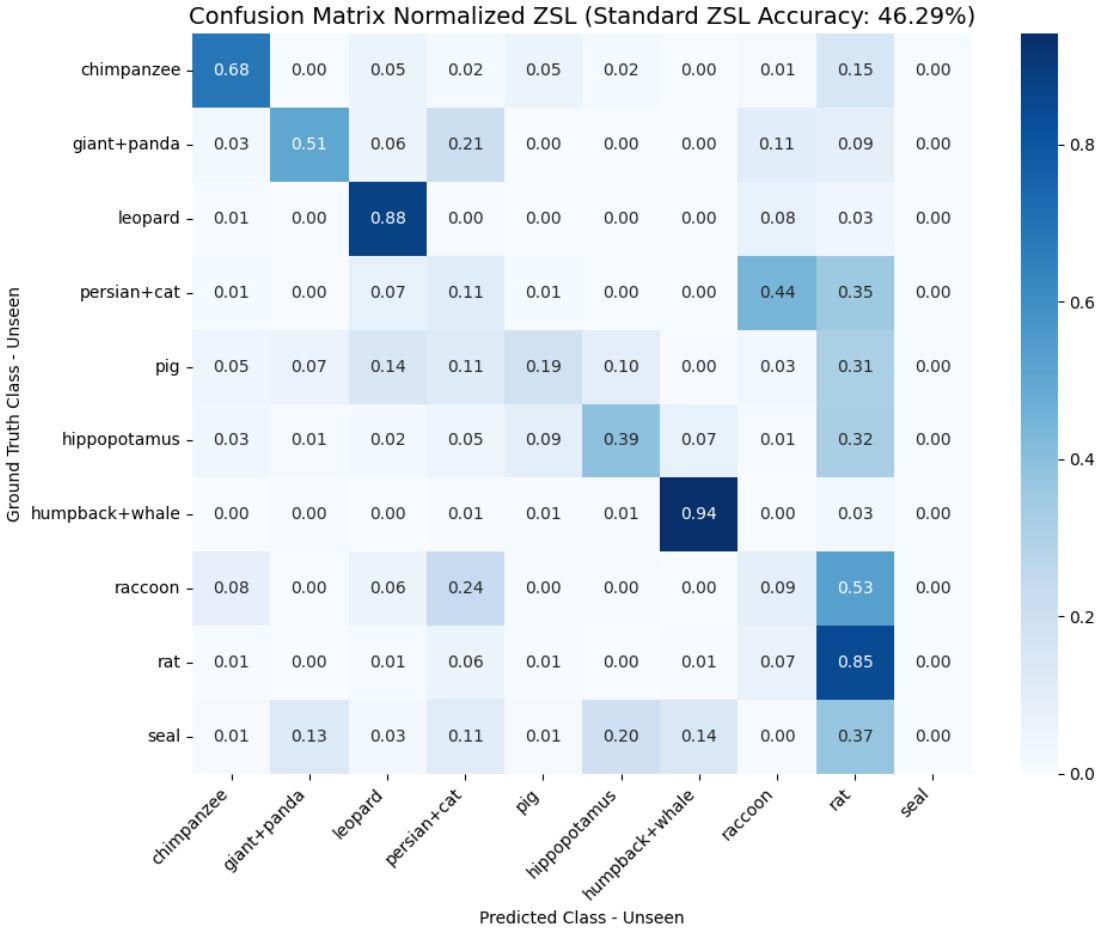


Figure 5.9: Standard ZSL Confusion Matrix for Cosine Similarity Baseline

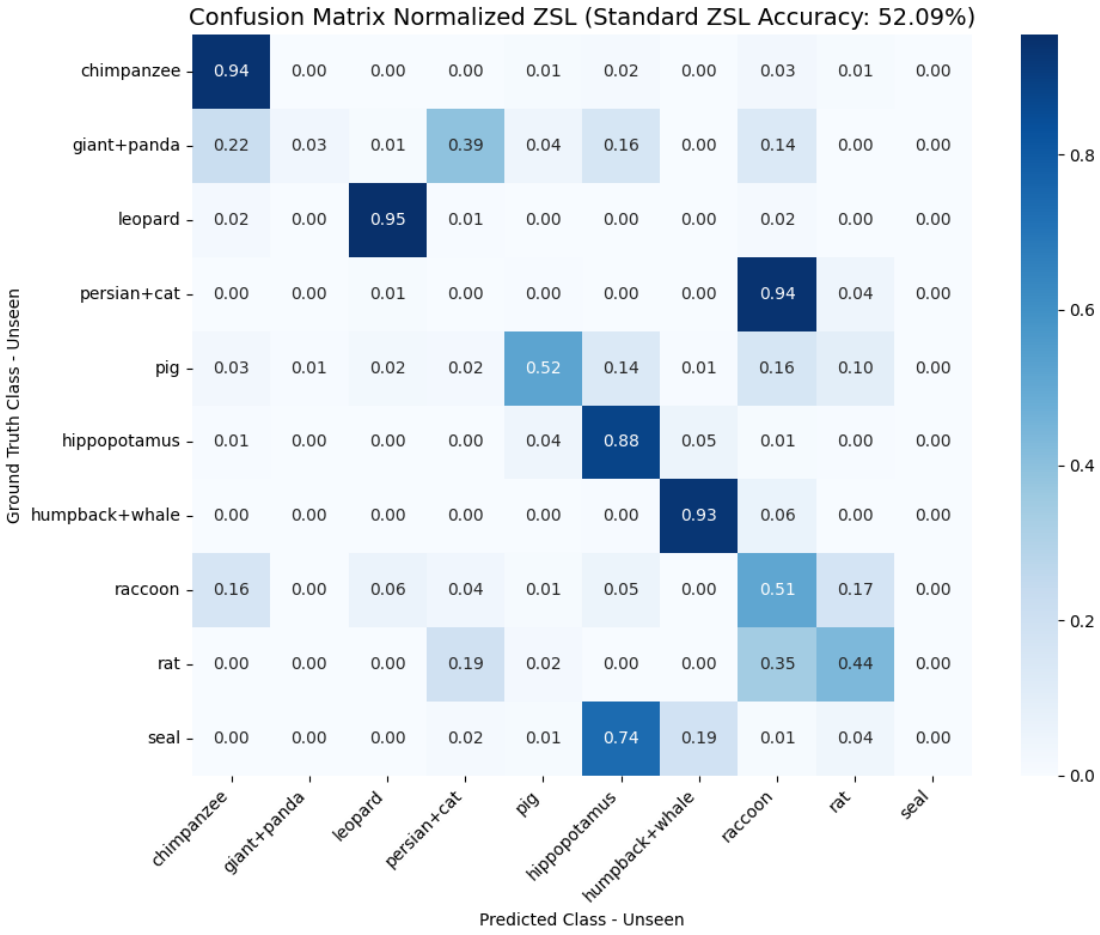


Figure 5.10: Standard ZSL Confusion Matrix for Dynamic Box Model with joint training

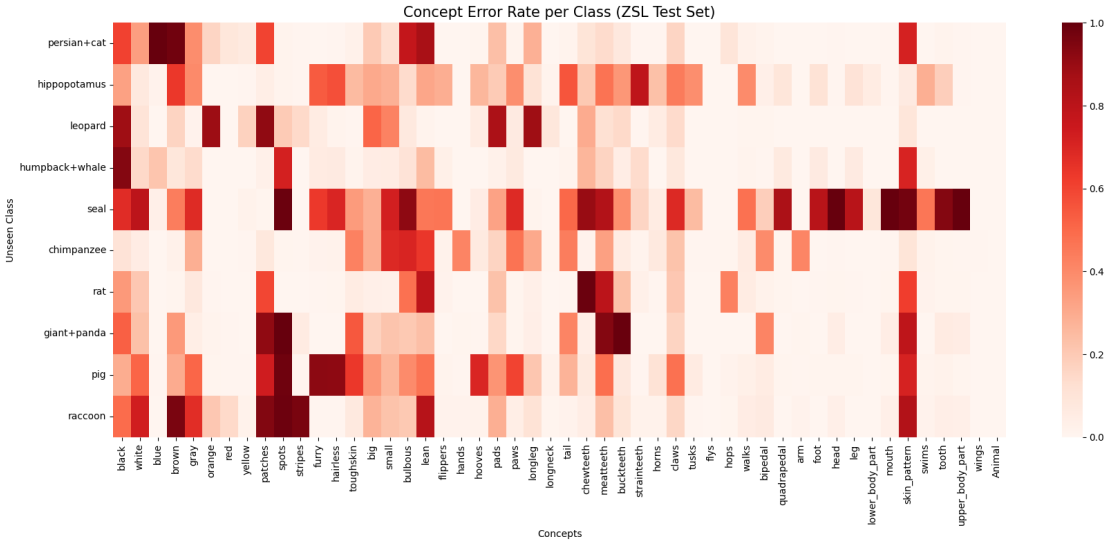


Figure 5.11: Concept Error Rate for the Cosine Similarity baseline utilizing Independent Training. The visual extractor produces scattered binary mispredictions across various concepts and unseen classes. Due to the additive nature of the flat vector space, the model algebraically compensates for these errors by summing the positive signals of correctly identified attributes, allowing for graceful degradation rather than categorical failure.

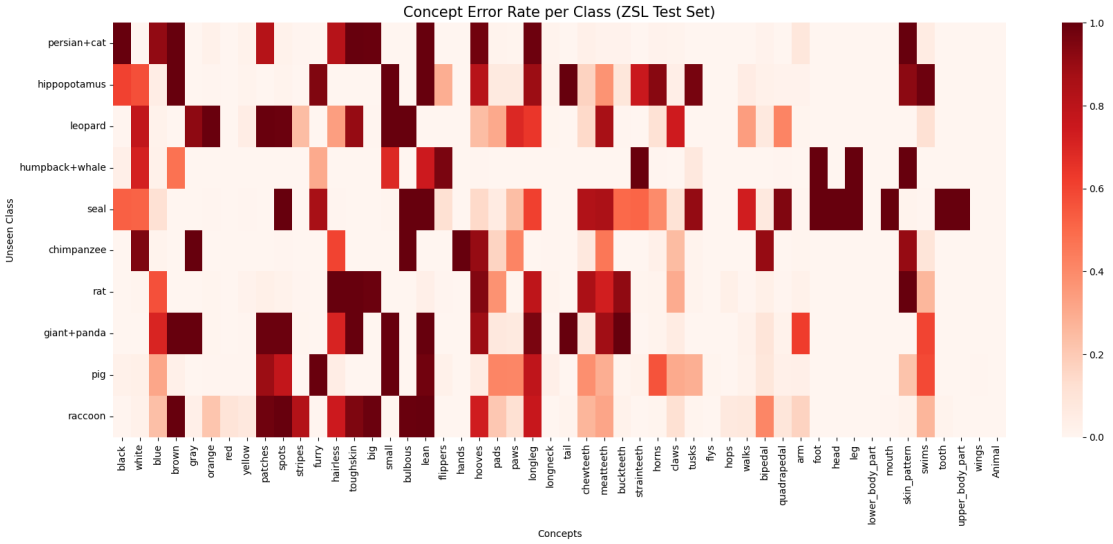


Figure 5.12: Concept Error Rate for the Dynamic Box architecture utilizing Frozen Geometric Joint Training. The matrix exposes severe, localized high-confidence errors on specific unseen classes, such as the Giant Panda. Governed by strict geometric AND-logic, the model cannot algebraically "outvote" these confident errors. Instead, erroneous traits physically drag the dynamic image volume into logically incompatible regions of the latent space, resulting in feature fragmentation.

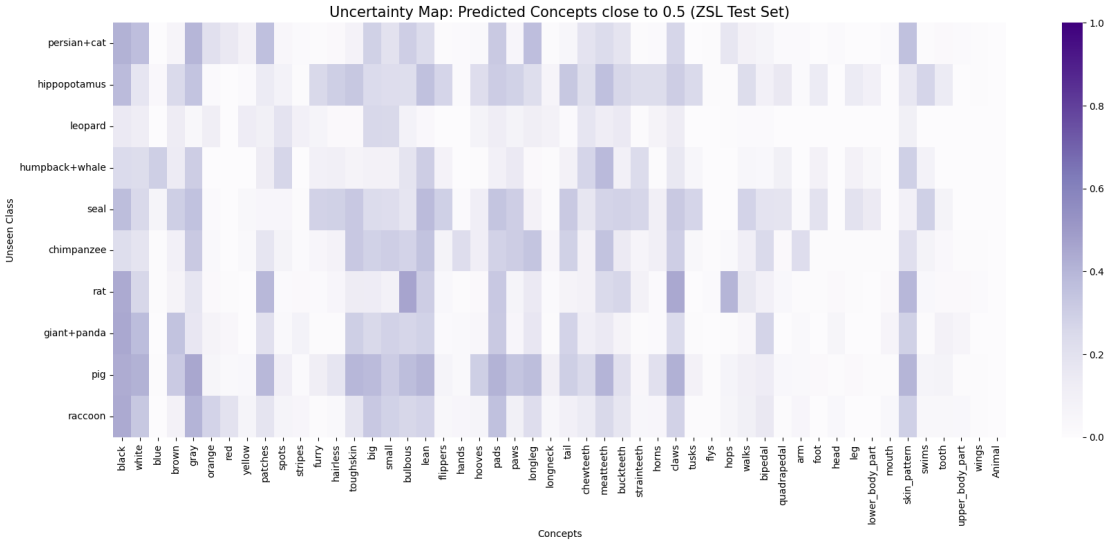


Figure 5.13: Uncertainty Map for the Cosine Similarity baseline utilizing Independent Training. The heatmap reveals polarized predictions, characterized by a lack of high uncertainty bands ($p \approx 0.5$). Because the visual extractor is optimized in isolation using solely Binary Cross-Entropy, it maximizes predictive certainty without facing contradictory gradients from the downstream classification task, resulting in confident visual features that lack global topological awareness.

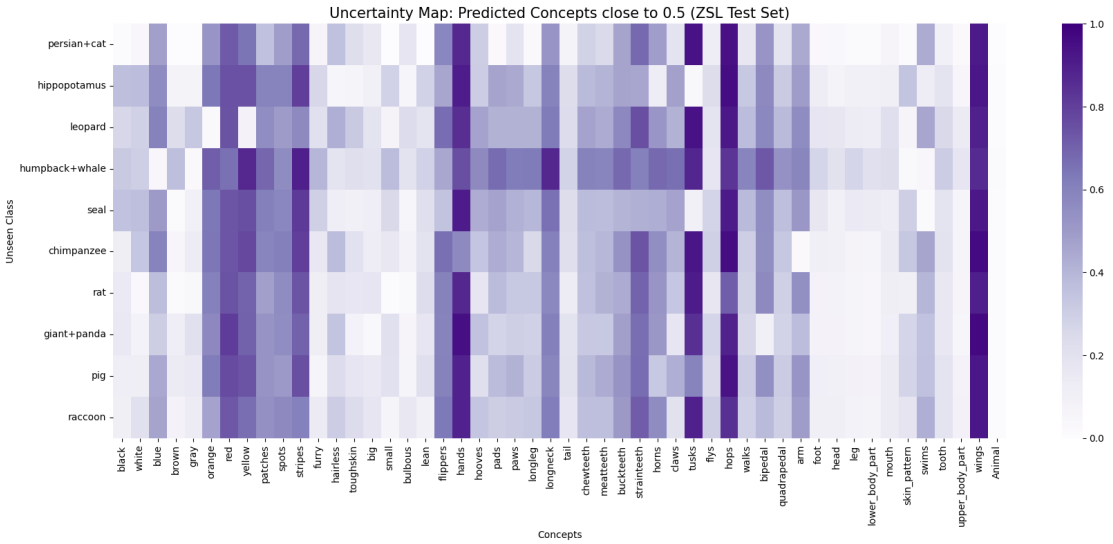


Figure 5.14: Uncertainty Map for the Dynamic Box architecture utilizing Frozen Geometric Joint Training. Prominent dark bands indicate high uncertainty ($p \approx 0.5$), illustrating the concept predictor's gradient oscillation between raw visual evidence and the strict topological penalties backpropagated from the frozen latent space. This structural uncertainty geometrically dilates the relaxed boundaries of the image box, driving the localized semantic clustering of hierarchically ambiguous classes like the Rat and Raccoon

Dynamic Box matrix, the Rat and the Raccoon exhibit a strong semantic overlap. The Rat (44% accuracy) is frequently confused with the Raccoon (35%), while the Raccoon (51%) is confused with the Rat (17%). From a topological perspective, this is a coherent error. Both animals share nearly identical hierarchical lineages (small, furry, quadrupedal, tailed). Because the geometric model prevents them from being misclassified as logically distant animals (e.g., marine life or large felines), their dynamic boxes are physically forced into the same constrained region of the latent space.

The Uncertainty Map in Figure 5.14 suggests the geometric cause for this localized confusion: the visual extractor exhibits several predictions gravitating around $p \approx 0.5$ for highly discriminative leaf-concepts. Because the geometric model computes the relaxed bounds of the image box as $z_{relaxed} = z_c \cdot p + (-M) \cdot (1 - p)$, this predictive uncertainty expands the dynamic boundaries toward infinity. Consequently, the lack of certainty on discriminative attributes creates enormously dilated volumes that overlap in the latent space. Without perfectly predicted leaf-concepts, the geometric intersection relies on minute volume differences, forcing the overlapping dynamic boxes into the same constrained region and leading to semantic clustering

Feature Fragmentation The most fascinating insights arise from the model's categorical failures, specifically the Seal (0.00% accuracy in both models) and the Giant Panda. The latter provides a textbook example of the fundamental trade-off between additive vector spaces and strictly constrained geometric topologies.

In the Cosine Similarity baseline, the Giant Panda is classified correctly 51% of the time. Conversely, in the Dynamic Box model, its accuracy collapses to 0.00%, with the predictions chaotically fragmenting into the Persian Cat (39%), Chimpanzee (22%), Hippopotamus (16%), and Raccoon (14%).

Comparing the Concept Error Rates in Figure 5.11 and Figure 5.12 with the Uncertainty Maps of Figure 5.13 and Figure 5.14 indicates that the main cause of feature fragmentation could be errors in concept predictions made with high confidence. The Error Map displays severe mispredictions on specific concepts for the Giant Panda, while the Uncertainty Map lacks dark bands in those regions, indicating these errors were made with absolute confidence (probability values perilously close to 1.0 or 0.0). The Cosine Similarity baseline operates on an additive logic (dot product). If the visual CNN correctly identifies major attributes like "black," "white," "big," and "furry," these positive signals algebraically outvote any erroneous micro-hallucinations, allowing the model to gracefully degrade and maintain predictive performance. The Dynamic Box, however, replaces additive compensation with strict geometric AND-logic. If the visual extractor hallucinates even a single highly confident, out-of-context attribute, the dynamic bounds of the image box are physically dragged toward logically incompatible regions. Because

the geometric model cannot mathematically "ignore" a highly confident visual error, the true intersection volume for the Giant Panda abruptly collapses to zero.

Ultimately, these topological errors demonstrate that the Dynamic Box model does not fail randomly. Unlike algebraic classifiers that rely on spurious correlations, the geometric model's failures are strictly semantic. When the visual extractor produces conflicting evidence, the strict geometric constraints force the model to seek the nearest logically sound container, exposing the critical necessity of a highly accurate visual backbone in neuro-symbolic architectures.

5.5 Faithfulness and Test-Time Intervention (TTI)

One of the primary desiderata of Concept Bottleneck Models is intervenability, the ability for a human expert in the loop to interact with the system at test time. By reviewing the intermediate semantic bottleneck and correcting mispredicted concepts, the expert should theoretically be able to guide the model toward the correct final classification. This process, known as Test-Time Intervention (TTI), serves not only as a practical tool for deployment in safety-critical domains but also as a rigorous proxy metric for evaluating the true faithfulness of the architecture, [3].

5.5.1 Intervention Protocols

Evaluating TTI at scale with real domain experts is often infeasible. Therefore, standard evaluation protocols simulate human interaction by progressively replacing the predicted concept probabilities $\hat{c}_i$ with their exact ground-truth binary values $c_{true,i}$ and observing the resulting impact on the final classification accuracy. To comprehensively evaluate the models, we employ two distinct intervention simulation strategies:

- **Random Intervention:** In this baseline protocol, the order in which concepts are selected for correction is entirely random. For each step (from 0 to k total concepts), a randomly chosen subset of concepts is overwritten with the ground truth. This approach evaluates the average algebraic or geometric sensitivity of the bottleneck but does not reflect realistic human behavior, as an expert would not waste time correcting concepts the model has already predicted with high certainty.

- **Uncertainty-Based Intervention:** To simulate a more realistic human-in-the-loop scenario, concepts are prioritized based on the model's predictive uncertainty. For each image, the concepts are ranked according to their proximity to the maximum uncertainty threshold (i.e., a predicted probability

$p \approx 0.5$). The simulated human expert intervenes by correcting the most ambiguous concepts first. This protocol evaluates how efficiently the model recovers from its most severe visual ambiguities when provided with precise, targeted semantic corrections.

5.5.2 TTI Trajectories (Linear vs. Geometric)

The efficacy of the interventions is visualized through TTI trajectories, plotting the final classification accuracy (Y-axis) against the number of intervened concepts (X-axis). As illustrated in Figure 5.15, both the standard Linear Classifier and the proposed Dynamic Box strategy exhibit monotonically increasing curves, eventually converging to 100% accuracy when all concepts are corrected (assuming the ground-truth ontology perfectly separates the classes).The monotonic increase of the Dynamic Box trajectory perfectly aligns with its underlying theoretical framework. Because inference is governed by strict spatial containment and geometric AND-logic, correcting a mispredicted concept physically restores the spatial bounds of the dynamic image box (B_{img}). Removing a visual hallucination expands an incorrectly shrunk bounding box, while enforcing a missed attribute correctly restricts it. Consequently, the intersection volume with the true class box monotonically recovers, directly reflecting a restoration of sound causal reasoning.

Conversely, the observation of a monotonically increasing trajectory for the standard Linear Classifier presents an apparent paradox. As established in Section 5.2.2, linear layers rely on shortcut learning and algebraic compensation, exploiting negative weights and spurious correlations to separate classes. Theoretically, intervening on a single concept should break this algebraic balancing act, fixing a mispredicted feature that the model was using to artificially penalize a competing class could cause the change of a right final class prediction to a wrong one, leading to local instability. However, the TTI curve for the linear layer remains monotonically increasing. This could be explained by a phenomenon of algebraic saturation that dominates at a macro-statistical level. When a simulated intervention replaces a predicted probability (often a noisy value like 0.3 or 0.8) with an absolute ground-truth integer (0 or 1), it maximizes the activation magnitude for that specific feature. Even if the linear layer has learned flawed, non-causal weights to compensate for visual noise, the injection of perfect ground-truth vectors acts as a massive algebraic signal boost for the correct class logit. The sheer mathematical mass of the correct 1.0 and 0.0 values eventually overwhelms the spurious negative weights. The TTI curve measures the aggregate accuracy across the entire test dataset. While an individual sample might indeed exhibit unpredictable, non-monotonic behavior due to a broken algebraic shortcut, the statistical average dictates that feeding perfectly clean, in-distribution data into a linear model optimized on similar ground-truth distributions will universally decrease the global error rate.

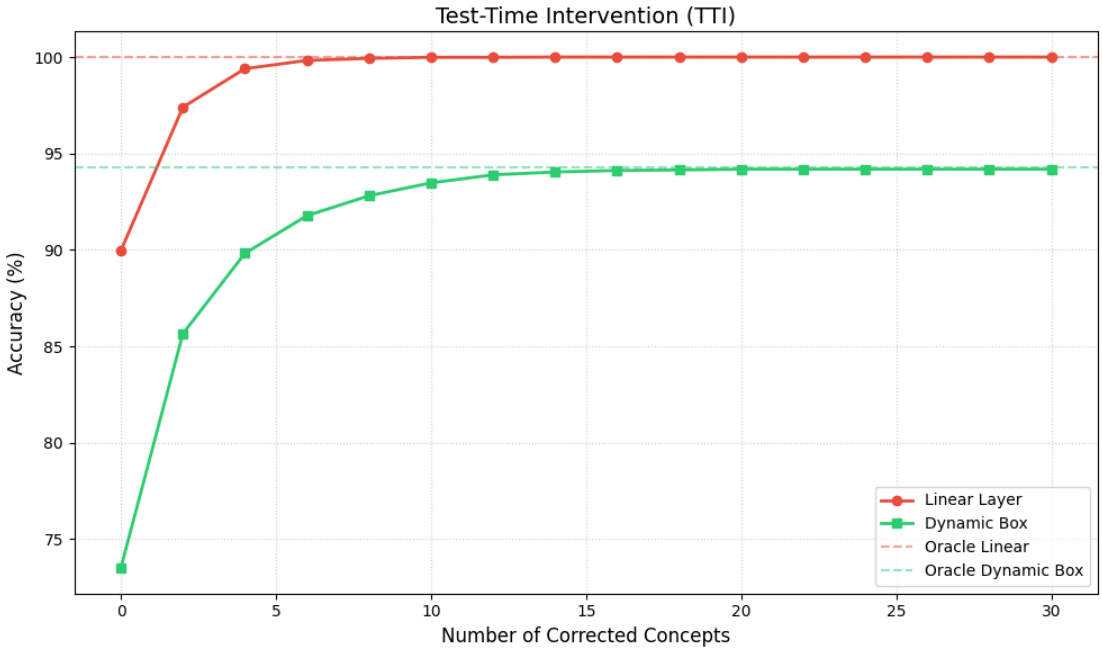


Figure 5.15: TTI curves comparison. Intervention is simulated correcting the k most uncertain concept predictions

Chapter 6

Conclusion and Future Work

6.1 Summary of Contributions

This thesis introduced an interpretable-by-design architecture developed to address the structural vulnerabilities of standard Concept Bottleneck Models (CBMs) and to improve its generalization capability. By transitioning from flat, unconstrained algebraic classifiers to a rigorous hierarchical framework, this work mitigated the shortcut learning and concept leakage phenomenon. The primary contributions of this research are summarized as follows:

- **Geometric Representation of Logical Hierarchies:** We formulated and implemented a decoupled architecture that maps visual concepts into a hierarchical geometric latent space using Box Embeddings. By translating expert-curated ontologies and Directed Acyclic Graphs (DAGs) into exact spatial constraints, the model directly embeds hierarchial rules such as strict parent-child entailment and mutually exclusive disjunctions into the core inference mechanism.

- **Mitigating Shortcut Learning as a Structural Consequence:** We demonstrated that replacing flat linear layers with pure geometric AND-logic (specifically through the Dynamic Box strategy) inherently prevents algebraic compensation. By performing classification via exact spatial intersections and strict logical containment, the architecture addresses the exploitation of negative weights and spurious correlations ensuring greater faithfulness to the concept bottleneck.

- **Facilitating Joint Training without Concept Leakage:** We introduced a new "Frozen Geometric Joint Training" paradigm. By freezing the structurally optimized geometric classifier during the joint training of the visual backbone,

the architecture significantly mitigates the concept leakage. This helps the concept representations to maintain their semantic meaning.

- **Enhancement of Zero-Shot Transfer:** We find results that suggest that enforcing logical rigor through hierarchical topology could improve a model's capacity to generalize to new, unseen classes. Evaluated on the Animals with Attributes 2 (AwA2) dataset under standard Zero-Shot Learning (ZSL) protocols, the proposed hierarchical architecture achieved a 52.09% accuracy, yielding a +5.8% improvement over the standard Cosine Similarity baseline.

6.2 Limitations of the Proposed Approach

The primary vulnerability of the Dynamic Box approach lies in its logical fragility in the presence of visual errors made by the concept extractor before the concept bottleneck. By entirely replacing additive mathematical compensation with strict geometric AND-logic, the model loses the capacity to gracefully degrade when the Convolutional Neural Network (CNN) produces highly confident but erroneous concept predictions. As demonstrated in the Zero-Shot Learning evaluations a single mispredicted, out-of-context attribute can physically drag the dynamic bounds of the image box toward logically incompatible regions of the latent space. Because the rigid geometric framework cannot algebraically "outvote" or ignore these errors as a flat vector space would, the true intersection volume abruptly collapses, leading to severe feature fragmentation and semantic misclassifications.

An additional challenge is given by the fact that certain concepts are defined as mutually exclusive in the ground truth used to train box embeddings could cause issues for dynamic boxes. For instance, while concepts like "white" and "black" are clearly semantically exclusive, they might both belong to the same class—such as "giant panda." Since the classifier expects absolute spatial disjointness for opposing concepts (an intersection probability of 0.0), the spatial and geometric logic struggles to represent a visual instance that simultaneously contains attributes considered mutually exclusive at an ontological level.

Furthermore, the decoupled nature of the architecture relies heavily on a static, pre-defined Directed Acyclic Graph (DAG) to construct the frozen geometric latent space. This approach operates under the assumption that the provided ontology is absolutely exhaustive, unambiguous, and perfectly aligned with the visual reality of the dataset. In many complex, real-world domains, concept annotations are inherently noisy, context-dependent, or incomplete. Because the geometric classifier remains strictly frozen during the joint training phase, the model lacks the structural flexibility to adapt to missing semantic relationships or to adjust flawed ontological premises dynamically.

These structural vulnerabilities highlight the inherent challenge of balancing

strict logical enforcement with adaptive flexibility. As explored in Section 6.3, these very limitations directly motivate the transition toward dynamic, learnable Box Embeddings. Permitting the geometric parameters to update during joint optimization—thereby intentionally harnessing "benign leakage"—could provide the necessary elasticity to overcome this logical brittleness. This would allow the model to gracefully compensate for visual errors and incomplete taxonomic trees while preserving the rigorous, hierarchical foundation of its reasoning.

6.3 Future Work

The methodology proposed in this thesis strictly relies on freezing the geometric box embeddings after the logical hierarchy has been embedded, severing the pathways required for gradient exploitation. This guarantees semantic faithfulness and structurally prevents the concept leakage that typically plagues standard joint training paradigms. During the initial development of this architecture, an alternative configuration was considered: utilizing box embeddings in place of the flat, high-dimensional vectors found in Concept Embedding Models (CEMs), [4]. In this scenario, the geometric boxes would be positioned directly within the bottleneck and remain fully learnable during the joint training of the visual model.

This dynamic approach was initially discarded due to the vulnerabilities associated with concept leakage. As demonstrated by recent critiques, [7], architectures that utilize high-dimensional continuous embeddings, such as CEMs, inherently suffer from both concepts-task and interconcept leakage. By allowing unconstrained classification gradients to backpropagate through the bottleneck, the embeddings act as conduits for uninterpretable, task-specific visual signals, fundamentally compromising the model's true interpretability and safety in high-stakes deployment scenarios.

However, recent literature, [27] argues that the complete eradication of information leakage is often an ill-posed objective that can lead to impractical models when deployed under real-world constraints. Specifically, in scenarios where the annotated concept set is inherently incomplete, allowing a controlled amount of "benign leakage" is often necessary to achieve competitive downstream task accuracy and to maintain effective test-time intervenability.

In light of the potential benefits of harnessing benign leakage, reconsidering learnable Box Embeddings as an active, dynamic bottleneck represents a highly promising direction for future work. By allowing the geometric parameters of the boxes to be updated during the joint training phase, the architecture could exploit benign leakage to mathematically compensate for missing semantic information. The inherent rigidity of geometric intersections and spatial containment rules could act as a natural structural regularizer. This would ensure that, while the model

captures the necessary task-relevant visual signals to maximize performance, the fundamental topological hierarchy of the ontology remains robust and intact.

Bibliography

[1] Cynthia Rudin. «Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead». In: *Nature machine intelligence* 1 (2018), pp. 206–215 (cit. on p. 1).

[2] Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. «Concept-based Explainable Artificial Intelligence: A Survey». In: *ACM Computing Surveys* (2023) (cit. on p. 1).

[3] 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*. Pmlr. 2020, pp. 5338–5348 (cit. on pp. 5, 11, 12, 14, 52).

[4] Mateo Espinosa Zarlenga et al. «Concept embedding models: Beyond the accuracy-explainability trade-off». In: *Advances in neural information processing systems* 35 (2022), pp. 21400–21413 (cit. on pp. 5, 12, 57).

[5] Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. «Do concept bottleneck models learn as intended?» In: *arXiv preprint arXiv:2105.04289* (2021) (cit. on p. 6).

[6] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. «Promises and pitfalls of black-box concept learning models». In: *arXiv preprint arXiv:2106.13314* (2021) (cit. on pp. 6, 7).

[7] Enrico Parisini, Tapabrata Chakraborti, Chris Harbron, Ben D MacArthur, and Christopher RS Banerji. «Leakage and interpretability in concept-based models». In: *arXiv preprint arXiv:2504.14094* (2025) (cit. on pp. 6, 7, 14, 57).

[8] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. «Shortcut learning in deep neural networks». In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673 (cit. on p. 6).

[9] Konstantinos P Panousis, Dino Ienco, and Diego Marcos. «Coarse-to-fine concept bottleneck models». In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 105171–105199 (cit. on p. 8).

[10] Yiheng Dong, Yi Lin, and Xin Yang. «Copa: Hierarchical concept prompting and aggregating network for explainable diagnosis». In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2025, pp. 67–76 (cit. on p. 8).

[11] Xinyu Mu, Hui Dou, Furao Shen, and Jian Zhao. «ConceptFlow: Hierarchical and Fine-grained Concept-Based Explanation for Convolutional Neural Networks». In: *arXiv preprint arXiv:2509.18147* (2025) (cit. on p. 8).

[12] Andong Wang, Wei-Ning Lee, and Xiaojuan Qi. «Hint: Hierarchical neuron concept explainer». In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2022, pp. 10244–10254 (cit. on p. 8).

[13] J Jürß, LC Magister, P Barbiero, P Lio, and N Simidjievski. *Everybody Needs a Little HELP: Explaining Graphs via Hierarchical Concepts (2023)* (cit. on p. 8).

[14] Ao Sun, Yuanyuan Yuan, Pingchuan Ma, and Shuai Wang. «Boosting Concept Bottleneck Models with Supervised, Hierarchical Concept Learning». In: () (cit. on p. 8).

[15] Oscar Hill, Mateo Espinosa Zarlenga, and Mateja Jamnik. «Hierarchical concept-based interpretable models». In: *International Conference on Learning Representations*. Vol. 2026. 2026, pp. 11581–11609 (cit. on p. 8).

[16] Peter Hase, Chaofan Chen, Oscar Li, and Cynthia Rudin. «Interpretable image recognition with hierarchical prototypes». In: *Proceedings of the AAAI conference on human computation and crowdsourcing*. Vol. 7. 2019, pp. 32–40 (cit. on p. 8).

[17] Federico Pittino, Vesna Dimitrievska, and Rudolf Heer. «Hierarchical concept bottleneck models for vision and their application to explainable fine classification and tracking». In: *Engineering Applications of Artificial Intelligence* 118 (2023), p. 105674 (cit. on p. 8).

[18] David Debot, Pietro Barbiero, Gabriele Dominici, and Giuseppe Marra. «Interpretable hierarchical concept reasoning through attention-guided graph learning». In: *arXiv preprint arXiv:2506.21102* (2025) (cit. on p. 9).

[19] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. «Making better mistakes: Leveraging class hierarchies with deep networks». In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2020, pp. 12503–12512 (cit. on p. 9).

[20] Yongqin Xian, Bernt Schiele, and Zeynep Akata. «Zero-shot learning—the good, the bad and the ugly». In: *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE. 2017, pp. 3077–3086 (cit. on pp. 10, 32, 42).

[21] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. «Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly». In: *IEEE transactions on pattern analysis and machine intelligence* 41.9 (2018), pp. 2251–2265 (cit. on pp. 10, 33, 42).

[22] Alice Lai and Julia Hockenmaier. «Learning to predict denotational probabilities for modeling entailment». In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 2017, pp. 721–730 (cit. on p. 15).

[23] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. «Probabilistic embedding of knowledge graphs with box lattice measures». In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 263–272 (cit. on p. 15).

[24] Tejas Chheda, Purujit Goyal, Trang Tran, Dhruvesh Patel, Michael Boratko, Shib Sankar Dasgupta, and Andrew McCallum. «Box embeddings: An open-source library for representation learning using geometric structures». In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2021, pp. 203–211 (cit. on pp. 15, 32).

[25] Christoph H Lampert, Hannes Nickisch, Stefan Harmeling, et al. «Learning to detect unseen object classes by between-class attribute transfer.» In: *Cvpr*. Vol. 1. 2. Miami, FL. 2009, p. 3 (cit. on p. 43).

[26] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. «Attribute-based classification for zero-shot visual object categorization». In: *IEEE transactions on pattern analysis and machine intelligence* 36.3 (2013), pp. 453–465 (cit. on p. 43).

[27] Mateo Espinosa Zarlenga. «In Defense of Information Leakage in Concept-based Models». In: *arXiv preprint arXiv:2606.10669* (2026) (cit. on p. 57).