# Unveiling Ontological Commitment in Multi-Modal Foundation Models

**Mert Keser**[a,b,*,1], **Gesina Schwalbe**[c,**,1], **Niki Amini-Naieni**[d], **Matthias Rottmann**[e] **and Alois Knoll**[b]

[a]Technical University of Munich, Germany
[b]Continental AG, Germany
[c]University of Lübeck, Germany
[d]University of Oxford, UK
[e]University of Wuppertal, Germany

**Abstract.** Ontological commitment, i.e., used concepts, relations, and assumptions, are a corner stone of qualitative reasoning (QR) models. The state-of-the-art for processing raw inputs, though, are deep neural networks (DNNs), nowadays often based off from multimodal foundation models. These automatically learn rich representations of concepts and respective reasoning. Unfortunately, the learned qualitative knowledge is opaque, preventing easy inspection, validation, or adaptation against available QR models. So far, it is possible to associate pre-defined concepts with latent representations of DNNs, but extractable relations are mostly limited to semantic similarity. As a next step towards *QR for validation and verification of DNNs*: Concretely, we propose a method that *extracts the learned superclass hierarchy* from a multimodal DNN for a given set of leaf concepts. Under the hood we (1) obtain leaf concept embeddings using the DNN's *textual input modality*; (2) apply hierarchical clustering to them, using that *DNNs encode semantic similarities via vector distances*; and (3) label the such-obtained parent concepts using search in *available ontologies from QR*. An initial evaluation study shows that meaningful ontological class hierarchies can be extracted from state-of-the-art foundation models. Furthermore, we demonstrate how to validate and verify a DNN's learned representations against given ontologies. Lastly, we discuss potential future applications in the context of QR.

## 1 Introduction

One of the basic ingredients of QR models is an ontology specifying the allowed concepts, relations, and any prior assumption about them; more precisely, the commitment to (a subset of an) ontology with associated semantic meaning of concepts and relations [20]. Thanks to years of research, large and rich ontologies like Cyc [30], SUMO [35], or ConceptNet [53] are readily available for building or verifying QR models.

Meanwhile, however, DNNs have become the de-facto state of the art for many applications that hardly allow a precise input specification [42], such as processing of raw images (*computer vision*), e.g., for object detection [19], or processing of unstructured natural language text [37]. This machine learning approach owes its success to its strong representation learning capabilities: DNNs automatically learn highly non-linear mappings (*encoding*) from inputs to vectorial intermediate representations (*latent representations* or vectors) [11], and reasoning-alike processing rules [3, 23] from these to a desired output. Availability of large text and image datasets have further sparked the development of *multimodal* so-called *foundation models* [10, 28, 45]. These are large general-purpose DNNs trained to develop semantically rich encodings suitable for a variety of tasks [10]. This is oft achieved by training them to map textual descriptions and images onto matching vectorial representations (*text-to-image alignment*) [45], using multimodal inputs of both images and text.

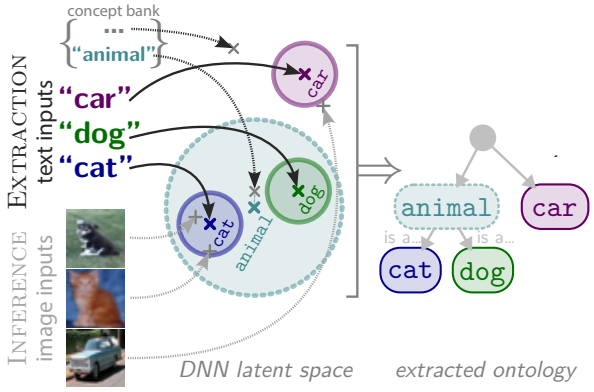


**Figure 1.** Illustration of the approach for ontology extraction from multimodal DNNs: For *extraction*, (1) obtain leaf nodes (`cat`, `dog`, `car`) as the latent representations of their textual descriptions; (2) cluster these to get parent representations (*dotted*); (3) assign parents the closest concept (`animal`) from a *concept bank*. For *inference* check at each level similarity against nodes' latent representations (e.g., first `animal` vs. `car`).

**The prospect.** Foundation models come with some interesting prospects regarding their learned knowledge: (1) One can expect foundation models to **learn a possibly interesting and useful *ontology***, giving insights into *concepts* [27, 29, 49, 62] and concept relations [16, 27] prevalent in the training data; and (2) such sufficiently large models can also **develop sophisticated reasoning chains** on the learned concepts [23, 44]. From the point of perspective of QR, this raises the question, whether this learned knowledge is consistent with the high quality available ontologies and QR models. This opens up well-grounded verification and validation criteria for safety or ethically critical applications. As a first step towards this, this pa-

* Corresponding Author. Email: mert.keser@continental.com
** Corresponding Author. Email: gesina.schwalbe@uni-luebeck.de
[1] Equal contribution.

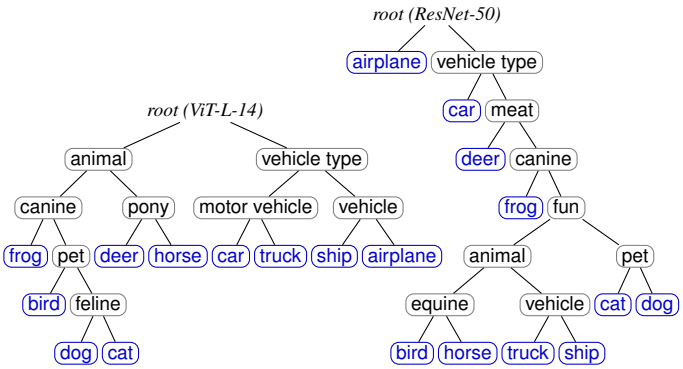


**Figure 2.** Comparison of two superclass hierarchies for given leaf concepts (blue) from CIFAR-10 [4] extracted from the large ViT-L-14 (*left*; with optimized prompt; 92% accuracy) and the smaller ResNet-50 (*right*; 46% accuracy) CLIP backbones with optimal distance metric settings. It shows the positive influence of model quality and prompt optimization (using *"a photo of a* `class` *"* instead of "`class`") on the plausibility of the extracted ontology, and how the human-alignedness accuracy serves as indicator for it.

per defines techniques for extraction and verification of simple class hierarchies. Future prospects encompass to use the extracted knowledge from DNNs for knowledge retrieval, and ultimately gain control over the learned reasoning: This would enable the creation of powerful **hybrid systems** [14, 31] that unite learned encoding of raw inputs like images with QR models.

**The problem.** Unfortunately, the flexibility of DNNs in terms of knowledge representation comes at the cost of *interpretability* [22]; and, being purely statistical models, they may extract *unwanted and even unsafe correlations* [27, 47, 51]. The opaque distributed latent representations of the input do not readily reveal which interpretable concepts have been learned, nor what reasoning is applied to them for obtaining the output. This is a pity, not least because that hinders verification of ethical and safety properties. Take as an example the ontological commitment: Which hierarchical subclass-relations between concepts are considered? An example is shown in Fig. 3. This directly encodes the learned bias, which commonalities between classes are taken into account, and which of these are predominant for differentiating between classes. The same example also nicely illustrates the issue with wrongly learned knowledge: The models may focus on irrelevant but correlated features to solve a task, such as typical background of an object in object detection [47].

(a) $\texttt{mammal} \supseteq \{\texttt{cat}, \texttt{dog}, \texttt{horse}\}, \texttt{amphibian} \supseteq \{\texttt{frog}\}$

(b) $\texttt{indoor} \supseteq \{\texttt{cat}, \texttt{dog}\}, \texttt{outdoor} \supseteq \{\texttt{horse}, \texttt{wet}\}, \texttt{wet} \supseteq \{\texttt{frog}\}$

**Figure 3.** Two exemplary ontological commitments: class hierarchies of the given leaf classes `frog`, `cat`, `dog`, `horse`, differentiating by (a) biology (mammal vs. amphibian), (b) image background (a Clever Hans effect!).

A whole research field, *explainable artificial intelligence* (XAI), has evolved that tries to overcome the lack of DNN interpretability [22, 50]. To date it is possible to partly associate learned representations with interpretable symoblic *concepts* (1-ary predicates) [52], such as whether an image region is a certain object part (e.g., `isLeg`), or of a certain texture (e.g., `isStriped`) [16, 27]. However, extraction of learned relations is so far focused on simple semantic similarity of concepts [16, 48]; hierarchical relations that hold across subsequent layers, i.e., across subsequent encoding steps [27, 59, 60]; or hierarchies obtained when *sub*dividing a root concept [33]. And while first works recently pursued the idea to extract superclass hiearchies from given leaves, these are still limited to simple classifier architectures [59]. A next step must therefore be: Given a set of (hierarchy leaf) concepts, how to extract (1) the **unifying superclasses**, and (2) the resulting **class hierarchy with subclass relationships** from any semantically rich intermediate output of a DNN, preferrably from the embedding space of **foundation models**.

**Approach.** We here propose a simple yet effective means to get hold of these encoded class hierarchies in foundation models; thereby taking another step towards unveiling and verifying the ontological commitment of DNNs against known QR models respectively ontologies. Building on [59] and [62], our approach leverages two intrinsic properties of the considered computer vision models:

(1) Vision DNNs generally encode learned concept similarities via distances in their latent representation vector space [16]. This makes it reasonable to find a hierarchy of superclass representations by means of **hierarchical clustering** [59].
(2) Foundation models accept textual descriptions as inputs, trained for **text-to-image alignment**. This allows to cheaply establish an approximate bijection of textual concept descriptions to representations: A description is mapped by the DNN to a vector representation, and a given representation is assigned to that candidate textual description mapped to the most similar (=close by) vector [62].[2]

**Contributions.** Our main contributions and findings are:

★ An approach to **extract and complete a simple learned ontology**, namely a superclass hierarchy with given desired leaf concepts (Figure 2), from intermediate representations of any multimodal DNN, which allows to manually validate DNN-learned knowledge against QR models (see Figure 1);
★ An approach to **test the consistency of multimodal DNNs against a given class hierarchy**, e.g., from standard ontologies;
★ An initial experimental validation showing that the approach can **extract meaningful ontologies**, and reveal inconsistencies with given ontologies;
★ A thorough discussion of **potential applications** for QR extraction and insertion from / into DNNs.

## 2 Related Work

**Extraction of learned ontologies.** Within the field of XAI [22, 50], the subfield of concept-based XAI (c-XAI) has evolved around the goal to associate semantic concepts with vectors in the latent representations [29, 40, 49]. For analysis purposes, methods here allow to both extract representations which match given concept specifications (supervised approach) [16, 26, 27, 62] as well as mine meanings for the most prevalent representations used by the DNN (unsupervised approach) [18, 63]. Notably, we here utilize the supervised approach by Yuksekgonul et al. [62] which directly utilizes the text-to-image alignment in multimodal DNNs. Such associations have found manifold applications in the inspection of DNNs' learned ontology, such as: Which concepts from a *given* ontology are learned [2, 52]? And how similar are representations of different concepts [16, 48]? This was extended to questions about the QR of the models, such as sensitivity of later concept representations (or outputs) to ones in earlier layers [27], or compliance with pre-defined logical rules [52]. However, very few approaches so far explored more *specific relations* between concept representations within *the same* layer's representation space. In particular, specific relations beyond general semantic similarity, such as class hierarchies. This is a severe gap when

[2] This could be replaced by the mentioned approximate concept extraction techniques for models without decoder and text-to-image alignment.

trying to understand the learned ontological relations between concepts: DNNs develop increasing levels of abstraction across subsequent layers [16], rendering the concepts occurring in their representation spaces hardly comparable. Notably, Wan et al. [59] challenged this gap and applied hierarchical clustering on DNN representations. However, their association of given concepts to latent representations is limited to last layer's output class representations, which we want to resolve. Furthermore, existing work was devoted only to single kinds of relations. We here want to show that these efforts can be unified under the perspective of investigating ontological commitment of DNNs.

## 3 Background

### *3.1 Deep neural network representations*

**DNNs.** Mathematically speaking, deep neural networks are (almost everywhere) differentiable functions $F\colon \mathbb{R}^n \to \mathbb{R}^m$ which can be written in terms of small unit functions, the so-called *neurons* $f\colon \mathbb{R}^n \to \mathbb{R}$, by means of the standard concatenation operation $f \circ g\colon x \mapsto f(g(x))$, linear combination $x \mapsto Wx + b$, and product $a, b \mapsto a \cdot b$. Typically, the linear weights $W$ and biases $b$ serve as trainable parameters, which can be optimized in an iterative manner using, e.g., stochastic gradient descent. Neurons are typically arranged in *layers*, i.e., groups where no neuron receives outputs from the others. Due to this "Lego"-principle, DNNs are theoretically capable of approximating any continuous function (on a compact subspace) up to any desired accuracy [25], and layers can be processed highly parallel. In practice, this is a double-edged sword: DNNs of manageable size show astonishing approximation capabilities for target functions like detection or pixel-wise segmentation of objects in images [28, 56]. However, they also tend to easily extract irrelevant correlations in the data, leading to incorrect [47] or even non-robust [55] generalization respectively "reasoning" on new inputs.

**Latent representations.** In the course of an inference of an input $x$, each layer $L$ of the DNN produces as intermediate output a vector $F_{\to L}(x) \in \mathbb{R}^n$, each entry being the output of one of the $n$ neurons of $L$. This vectorial encoding of the input is called the *latent representation* of the input within $L$, and the vector space $\mathbb{R}^n$ hosting the representations is called the *latent space*. Interestingly, it was shown that DNNs encode semantically meaningful information about the input in their latent representations, with abstraction increasing the more layers are passed (e.g., starting with colors and textures, to later develop notions of shapes and objects) [16, 36].

**Concept embeddings.** An emergent property of these representations is that in some layers, a concept $C$ (e.g., color `Red`, or object part `Leg`), can be encoded as prototypical vector $e(C)$ within this latent space. These are called *concept (activation) vectors* [27] or *concept embeddings* [16]. The mapping $e\colon \mathcal{C} \to \mathbb{R}^n$ from a set of human-interpretable concepts to their embeddings even preserves semantic similarities to some extend: Examples are the reflection of analogical proportions [43] in word vector spaces (DNNs with textual inputs trained for natural language processing), like "$e(\texttt{King}) - e(\texttt{Queen}) = e(\texttt{Man}) - e(\texttt{Woman})$" [32]; and their analogues in standard computer vision architectures trained for object classification or detection: "$e(\texttt{Green}) + e(\texttt{Wood}) = e(\texttt{Tree})$" [16]. Our approach relies on these natural translation of semantic to vector operations/properties. In particular, we assume that the relation `IsSimilarTo`[3] on input instances $x$ is mapped to some distance metric $d$ like Euclidean or cosine distance by the DNN representations: $\forall \texttt{C}, \texttt{C}' : \texttt{IsSimilarTo}(\texttt{C}, \texttt{C}') \Leftrightarrow d(e(\texttt{C}), e(\texttt{C}')) \approx 0$.[4]

Concretely, we use the translation of similarity relations to find a superclass concept representation via interpolation.

**Text-to-image alignment.** In the case of multimodal DNNs that accept both textual and image inputs, the training often encompasses an additional (soft) constraint: Given textual descriptions of an input image, these must be mapped to the same/a similar latent representation as their respective image. While pure language models suffer from the impossibility to learn the true meaning of language concepts without supervision [9], this additional supervision might help the model to develop representations that better match the human understanding of the word/concept. We here leverage this intrinsic mapping to associate textual or graphical descriptions of our concepts with latent representations.

When using textual decriptions, good text-to-image alignment is an important assumption; but, sadly, even with explicit training constraints this is not guaranteed [17] (cf. distance of image and text embeddings in Figure 4). We show both the influence of text-to-image alignment on our method, how it can be reduced, and how to use our method in order to identify issues with the learned meaning of concepts, which opens up options to fix the representations.

### *3.2 Ontologies*

When modeling any problem or world, a basis of the model is to know "what the model is talking about". This is exactly answered by the underlying *ontology*, i.e., a definition of what categories/properties and relations are used in the model. We here adopt the definition from [20].

**Definition 1** (Ontology)**.** *An* ontology *is a pair* $(\mathcal{V}, \mathcal{A})$ *constituted by a vocabulary* $\mathcal{V} = \mathcal{C} \cup \mathcal{R}$ *of a set of unary predicates* $\mathcal{C}$ *(the* concepts *corresponding to class memberships and other properties) and a set of binary predicates* $\mathcal{R}$ *(the instance* relations*) used to describe a certain reality, and which are further constraint by a set* $\mathcal{A}$ *of explicit* assumptions *in the form of a first- (or higher-)order logic theory on the predicates.*

A relation we will use further is $\texttt{IsSimilarTo} \in \mathcal{R}$. Also spatial relations like `IsCloseBy` [52] and `LeftOf`, `TopOf`, etc. [44] have been defined and used in literature for latent space representations of objects. Simple examples of assumptions that relate the concept sets are, e.g., the subclass relationship we investigate in this paper: $\texttt{IsSuperclassOf}(\texttt{C}', \texttt{C}) :\Leftrightarrow (\forall v\colon \texttt{C}(v) \Rightarrow \texttt{C}'(v))$ (cf. Figure 3). This can also be seen as a relation between concepts, by interpreting the unary concept predicates $C$ as sets of objects (e.g., classes) via $v \in C :\Leftrightarrow C(v)$. The validity of concept embeddings also gives rise to assumptions about concepts $(\forall v\colon C(v) \Leftrightarrow \texttt{IsSimilarTo}(v, e(C)))$. Note that, given embeddings, we can formulate relations between *concepts* using *instance* relations $R \in \mathcal{R}$ via $R(\texttt{C}, \texttt{C}') :\Leftrightarrow R(e(\texttt{C}), e(\texttt{C}'))$. An example would be $\texttt{isSimilarTo}(\texttt{cat}, \texttt{dog})$.

The first challenge in extracting learned QR from DNNs is to find/explain the ontology that is used within the reasoning process of the DNN. Unraveling an ontology as done in 1 above breaks this step roughly down into:

(1) Find the concepts $\mathcal{C}$ (and their embeddings) used by the model.

[3] We here assume that `IsSimilarTo` is reflexive and symmetric, following geometrical instead of psychological models of similarity [57].

[4] For optimization, the relative formulation can be more convenient: $\forall \texttt{C}, \texttt{C}', \texttt{C}''\colon$ C more similar to C′ than to C″ $\Rightarrow$ $d(e(\texttt{C}), e(\texttt{C}')) \leq d(e(\texttt{C}), e(\texttt{C}''))$.

(2) Find the relations $\mathcal{R}$ that may be formulated on vector instances.
(3) Simple assumptions $\mathcal{A}_s \subseteq \mathcal{A}$: How are concept related.
(4) Identify further assumptions $\mathcal{A} \setminus \mathcal{A}_s$ that the model applies.

Note that the layer-wise architecture of DNNs partitions the representations into objects (vectors) in the different latent spaces. For a layer $L$ we denote $v$ in the latent space of $L$ as $L(v)$. This gives rise to a partition of the concept, relation, and assumption definitions, allowing to conveniently split up above steps as follows:

(1') What concepts $\mathcal{C}_i \subset \mathcal{C}$ are encoded *within the ith layer* $L_i$ ($\forall C \in \mathcal{C}_i, v\colon \neg L_i(v) \Rightarrow \neg C(v)$)?

(3a') What assumptions $\mathcal{A}_{i,i}$ hold for which items within *the same* $i$th latent space ($\forall A \in \mathcal{A}_i, (v^{(s)})_s\colon \bigvee_s \neg L_i(v^{(s)}) \Rightarrow \neg A(v^{(1)}, \dots)$)?

(3b') What assumptions $\mathcal{A}_{i,j}, i \neq j$, hold between items of *different* latent spaces?

Task (1') is (somewhat) solved by methods from c-XAI, where both learned concepts [16, 27, 62] as well as their distribution over different layer representation spaces [34] are investigated. (3a') and (3b') show the yet-to-be-filled gaps: Investigated relations between items, item groups respectively concepts within the same arbitrary latent space (=(3a')). These so far only concern general semantic similarity, and relations across latent spaces only sensitivity. That falls far behind the richness of natural language; in particular it misses out on concept and instance relations of the kind "$\texttt{C}$ is similar to $\texttt{C}'$ with respect to feature $\texttt{F}$" respectively "$\texttt{C}$, $\texttt{C}'$ both are $\texttt{F}$", and counterpart "$\texttt{C}$ differs from $\texttt{C}'$ with respect to feature $\texttt{F}$"[5]. In other words, the relation $\texttt{IsSuperclassOf}$ is missing, despite known to be learned [59]. This inhibits the expressivity of extracted constraints such as obtained in [44], as this directly relies on the richness of available vocabulary. The method proposed in this paper thus sorts in as follows: **We extend the extraction of relations relevant to point (3a') (relations amongst concepts within the same layer representation space) by allowing to extract the $\texttt{IsSuperclassOf}$ relation between concepts.**

### 3.3 Hierarchical clustering

Hierarchical clustering [46] aims to find for a given set $M$ a chain of partitions $\mathcal{M}_1 \leq \mathcal{M}_2 \leq \dots \leq \{M\}$ connected by inclusion[6], i.e., assign each point in $M$ to a chain of nested clusters $M_{1,i_1} \subseteq M_{2,i_2} \dots \subseteq M$, as illustrated in Figure 1. Such a hierarchy can be depicted using a dendrogram as in Figure 2. There are two regimes for hierarchical clustering: Divisive breaks up clusters top-down, while agglomerative starts from the leaves $\mathcal{M}_1 = \{\{p\} \mid p \in M\}$ and iteratively merges clusters bottom-up [46]. We here employ hierarchical clustering to find a hierarchy of subsets of latent representation vectors. Since we start with given leaf vectors, **this work uses standard agglomerative hierarchical clustering [61]**.[7] This optimizes the partitions for small distance between the *single points* within a cluster (*affinity*) and a large distance between the *sets of points* making up different clusters (*linkage*), typically at a complexity of $\mathcal{O}(|M|^3)$.

[5] "$\texttt{C}$, $\texttt{C}'$ both are $\texttt{F}$" ($\forall x\colon (\texttt{C}(x) \vee \texttt{C}'(x)) \Rightarrow \texttt{F}(x)$) rewrites to $\texttt{IsSuperclassOf}(F, C) \wedge \texttt{IsSuperclassOf}(F, C')$; the "differs"-case to $\neg\texttt{IsSuperclassOf}(F, C) \wedge \texttt{IsSuperclassOf}(F, C')$.

[6] To be precise: $\mathcal{M} \leq \mathcal{M}' \Leftrightarrow \forall M \in \mathcal{M}\colon \exists M' \in \mathcal{M}'\colon M \subseteq M'$

[7] We here use the scikit-learn implementation at https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

## 4 Approach

This section details our approach towards extracting a globally valid approximation of a DNN's learned concept hierarchy, given the hierarchy's desired leaf concepts. The goal is to allow manual validation or verification testing against existing ontologies from QR. Recall that this both requires a guided exploration of the learned concepts (*which parent classes did the model learn?*), as well as an exploration of the applicability of the superclass relation (*which superclasses/features are shared or different amongst given concepts?*). We will start in subsection 4.2 by detailing how to obtain the extracted class hierarchy (here simply referred to as *ontology*). This is followed by an excursion on how to conduct a kind of instance-based inference using the global taxonomy (subsection 4.2, which is then used in subsection 4.3 where we discuss techniques for validation and verification of DNN learned knowledge.

### 4.1 Extracting an ontology

**Overview** The steps to extract our desired ontology are (explained in detail further below): (1) obtain the **embeddings** $e(\texttt{C}_i)$, (2) apply **hierarchical clustering** to obtain superclass representations as superclass cluster centers, (3) **decode** the obtained superclass representations into a human-interpretable description.

**Ingredients.** We need as ingredients our trained **DNN** $F$, some concept encoder $e$ (in our case defined using the DNN, see Step 1 below), the finite set $(\texttt{C}_i)_i = \mathcal{C}_{\text{leaf}}$ of **leaf concepts** for which we want to find parents classes, and the choice of **layer** $L$ in which we search for them. Furthermore, to ensure human interpretability of the results, we constrain both our leaf concepts as well as our solution parent concepts to come from a given **concept bank** $\mathcal{C}$ of human-interpretable concepts[8]. We furthermore need per concept $\texttt{C} \in \mathcal{C}$: A **textual description** $\texttt{toText}(\texttt{C})$ of $\texttt{C}$ as textual specification; optionally a set $\texttt{toImages}(\texttt{C})$ containing the concept as graphical specification (see Step 1), as available, e.g., from many densely labeled image datasets [8, 24]; and optionally a set $\texttt{Parents}(\texttt{C})$ of candidates for parent concepts of $\texttt{C}$ (for more efficient search). The following assumptions must be fulfilled, in order to make our approach applicable:

**Assumptions 1.**

(a) ***Text-to-image alignment:*** *The DNN should accept textual inputs, and be trained for text-to-image alignment, such that for a suitable textual description $T$ of any concept $\texttt{C} \in \mathcal{C}$ one can reasonably assume $e(\texttt{C}) \approx F_{\to L}(T)$. We use this to find embeddings: The embedding of a visual concept $\texttt{C}$ can be set to the DNN's text encoding $F_{\to L}(T)$ of a suitable textual description $T$ of $\texttt{C}$.*

(b) ***Existence of embeddings:*** *For all leaf concepts, embeddings $e(\texttt{C}_i)$ of sufficient quality exist in the latent space of $L$.*

(c) ***Concentric distribution of subconcepts:*** *Representations of subconcepts are distributed in a **concentric manner** around its parent. Generally, this does not hold [33], but so far turned out to be a viable simplification as long as semantic similarities are well preserved by the concept embedding function $e$ [18, 41]. I.e. for a superclass concept $\texttt{Parent}$ with children set $\mathcal{C}_S$ we can choose*

$$e(\texttt{Parent}) \approx \underset{\texttt{Child} \in \mathcal{C}_S}{\text{mean}}\, e(\texttt{Child}) \tag{1}$$

(d) ***Semantic interpolatability:*** *Consider a latent representation $v$ that is close to or inbetween (wrt. linear interpolation) some embeddings $e(C_i)$ and $e(C_j)$. We assume that $v$ can be interpreted*

[8] The concept bank restriction makes this essentially a search problem.

*to correspond to some concept, i.e.,* $\exists \mathtt{c} \in \mathcal{C}\colon \|e(\mathtt{c}) - v\|_2 < \epsilon$ *for some admissible error* $\epsilon$*. This is needed to make the averaging in the parent identification in* (1) *above meaningful.*

Note that Assumption 1(d) is very strong, stating that there is a correspondence between the semantic relations of natural language concepts, and the metric space structure of latent spaces. This is by no means guaranteed, but according to findings in word vector spaces [32] and also image model latent spaces [16] a viable assumption for the structure of learned semantics in DNNs.

**Step 1: Obtain the embeddings** $e(\mathtt{c}_i)$**.** We here leverage the text-to-image alignment to directly define the concept-to-vector mapping $e$: $e(\mathtt{c}) \coloneqq \text{mean}_{x \in \mathtt{toDNNInput(c)}}\, F_{\to L}(x)$. Following [59, 62], the `toDNNInput` function can be a mapping from concept to a single textual description [62] or to a set of representative images [59].

- *Textual concepts:* The naive candidate for a textual description $\mathtt{toDNNInput(c)} \coloneqq \mathtt{toText(c)}$. However, some additional prompt engineering may be necessary, i.e., manual adjustment and finetuning of the formulation [17, 45]. For example, following [45] we replace "c" by "an image of c" for the prompting.
- *Visual concepts:* Here we take the graphical $\mathtt{toImages(c)}$ specification of our concept. One could then employ standard supervised c-XAI techniques to find a common representing vector for the given images, e.g., as the weights of a linear classifier of the concept's presence [16, 27]. We here instead simply feed the DNN with each of the images and capture its respective intermediate latent representations, which is valid due to the concentricity assumption.

If the text-to-image alignment is low, we found image representations of concepts to yield more meaningful results.

**Step 2: Hierarchical clustering.** Employ any standard hierarchical agglomerative clustering technique to find a hierarchy of partitionings of the set of given concept embeddings. Each partitioning level represents one level of superclasses, with one cluster per class (see the simple example in Figure 1). As of (1), the mean of the cluster's embedding vectors is the embedding of its corresponding superclass (the *cluster center*).

Note that the hierarchical clustering in principle allows to: (a) start off with more than one vector per leaf concept, e.g., coming from several image representations or from jointly using embeddings from textual and image representations; (b) weight the contribution of each child to the parent. This, however, is only viable together with means to automatically determine the weights, and not further pursued here.

**Step 3: Decoding of cluster centers.** We here use a two-step search approach to assign each cluster center a concept from the concept bank $\mathcal{C}$. Given a cluster center $p$, the first optional step is to reduce the search space by selecting a subset of candidate concepts from $\mathcal{C}$. Following [62], (1a) we collect for every leaf concept c the set of those concepts that, according to the ConceptNet knowledge graph [53], are related to c by any of the relations in $\mathcal{R}_{\text{concepts}} = \{\mathtt{hasA}, \mathtt{isA}, \mathtt{partOf}, \mathtt{HasProperty}, \mathtt{MadeOf}\}$:

$$\mathtt{Parents(c)} \coloneqq \{\mathtt{p} \mid \bigvee_{R \in \mathcal{R}_{\text{concepts}}} R(\mathtt{p}, \mathtt{c})\}\,. \tag{2}$$

(1b) The union $\mathcal{P} = \bigcup_{\mathtt{c}\text{ leaf in cluster}} \mathtt{Parents(c)}$ of these sets serves as candidate set for $p$. Note that this is a simplification that allows to capture as superclass any best fitting commonality between the leaf concepts (e.g., background context like `indoor` or biological relation like `mammal` for $\{\mathtt{cat}, \mathtt{dog}\}$ as in Figure 3). Generally, there is a trade-off between very specific relation definitions, and fidelity to the learned knowledge of the model. The trade-off can be controlled by the broadening or narrowing of the candidate set. The here chosen broad definition of the `IsSuperClass` relationship between concepts favors fidelity to the model's learned knowledge. Investigating effects of more narrow concept candidate sets is future work. (2) In the second step, the concept for $p$ is then selected from the candidate set $\mathcal{P}$ to be the one with minimum distance embedding (embeddings again obtained as in Step 1): $e^{-1}(p) \coloneqq \text{argmin}_{\mathtt{p} \in \mathcal{P}}\, \|p - e(\mathtt{p})\|_2$.

The final result then is a hierarchy tree, where leaf nodes are the originally provided concepts, inner nodes are the newly extracted superclasses, and the connections represent the `IsSuperclassOf` relation. In the experimental section we will more closely investigate the influence of the proposed variants with/without prompt engineering and with/without finetuning.

## 4.2 Inference of an ontology

The such obtained ontology can be used for outlier-aware inference, i.e., classification of new input samples to one of the leaf concepts. This will be useful not only as an interesting standalone application in safety-relevant classification scenarios, but in particular for the validation.

The baseline of the inference is the $k$-nearest neighbor classifier: It directly compares the latent representation of a new input with each available concept embedding; and then assigns the majority vote of the $k$ nearest concept embeddings. To enrich the inference process with information from the ontology, one instead traverses the ontology tree, at each node branching off towards the closest child node.

*Remark* 1. Note that this allows to easily insert an outlier criterion: If at a parent class p none of the children nodes is closer than a threshold, the sample is considered an outlier *of class* p. This neatly preserves the maximum amount of information available about the properties of the sample, and, thus, eases subsequent handling of the unknown input. For example, an outlier of (parent-)class `StaticObject` should be treated differently than one of (parent-)class `Animal`.

Hyperparameters of this inference procedure are the choice of similarity, including whether to take into account the size (variance/width) of the cluster, e.g., by favoring wide over near-to-point-estimate clusters; and the threshold for being an outlier.

## 4.3 Validating and comparing learned ontologies

We now get to the core goal of this paper: Verify or validate a given DNN using QR. For this we start with validation of an extracted ontology from subsection 4.1, and discuss how to measure its fidelity to DNN learned knowledge, and alignedness to human prior knowledge, which here corresponds to the expected image-to-concept matching. Lastly, we show how one can encode a given ontology as contextualized embeddings to verify a DNN against given prior knowledge from QR.

**Human-alignedness.** One main desirable of a DNN's ontology is that it well aligns with the semantics that humans would expect and apply for the respective task. Any mismatch may either bring insights to the human on alternative solutions, or, more probably, indicates a suboptimal solution or even Clever Hans effect of the learned representations. A straight-forward way to measure the human-alignedness is to test the **prediction accuracy** of the ontology when used for inference (see subsection 4.2) on human-labeled samples. If human labels deviate often from the predictions, this indicates a bad alignment of the semantics the DNN has learned for the

concepts from those a human would expect. Other means to estimate the human-alignedness (not yet investigated in this work) are direct qualitative user studies, where human evaluators **manually check** the consistency of the obtained ontology tree with their own mental model; or automatic checking of consistency against given world knowledge or common sense ontologies like Cyc [30]. Lastly, the improvement in humans' predictions about the behavior of the model, a typical human-grounded XAI metric [50], could quantify in how far humans can make sense of the ontology.

A different aspect of human-alignedness is how well the ontology, in particular the inference scheme it defines, generalizes to novel concepts (semantic outliers) that so far have not occurred in leaves or nodes. The gerenalization can be measured as the performance in assigning a correct parent node. A special case here are blended cases where the novel concept unifies features of very different classes, such as a `cat with wheel as walking support`. The uncertainty of the model in such blended cases can be qualitatively compared against human one, potentially uncovering a bias.

**Text-to-image alignment.** The to-be-expected performance of cross-modal inference of the ontology (i.e., ontology defined using textual concepts, but inference done on images) directly depends on the quality of the text-to-image alignment. This motivates a use as an indicator for suboptimal text-to-image alignment.

**Fidelity.** Fidelity of the ontology, respectively shortcomings in the simplified modeling of the ontology, can be measured by the deviation between the baseline inference directly on the leaves, and the ontology inference. Inference on the leaf concepts $\texttt{c}_i$ means we predict for an image $x$ the output class $\texttt{c}$ for which the textual embedding is closest to the embedding of $x$, proximity measured with respect to some distance $d$ (here: cosine similarity):

$$\texttt{c} := \underset{\texttt{c}\in(\texttt{c}_i)_i}{\operatorname{argmin}}\, d\left(F_{\to L}(\texttt{toText}(\texttt{c}), F_{\to L}(x))\right) \tag{3}$$

This is referred to as naive *zero-shot* approach, following research on using foundation models on specialized tasks without finetuning (=with training on zero samples) [17, 45]. The reason to choose this as a baseline is that the ideal tree should sort samples into the same leaf neighborhood as direct distance measurement would do. Simplifications that may infringe this equality are unequal covariances ($\approx$ widths) of sibling class clusters; the chosen similarity measure; or assuming perfect text-to-image alignment.

**Verification against a given ontology.** The previous extraction techniques yield an inspectable representation of the ontology learned by a model. This allows manual validation of the learned knowledge against models from QR. Alternatively, one could directly verify a multimodal model against consistency with a given ontology: In short, we propose to modify the leaf concept embeddings from Step 1 such that they additionally encode their local part of the ontology, i.e., information about all desired parents of the leaf, as *context*. One can then measure the performance of naive inference (see subsection 4.2) on these contextualized leaf nodes as defined in (3). A higher performance then means a better alignment of the context of a leaf concept with its image representations. This even would allow to narrow down unalignedness to specific concepts (those with bad inference results). We suggest as point of attack for contextualization is the textual encoding: Let $\texttt{c}$ be a leaf concept at depth $d$ in the tree with chain of parents $(\texttt{p}_i)_{i=1}^{d}$ from root to leaf. We can now follow [17] and modify the original $\texttt{tT} = \texttt{toText}$ function of a leaf concept to:

$$\texttt{toText}'(\texttt{c}) := \text{“}\texttt{tT}(\texttt{p}_1), \ldots, \texttt{tT}(\texttt{p}_d), \texttt{tT}(\texttt{c})\text{”} \tag{4}$$

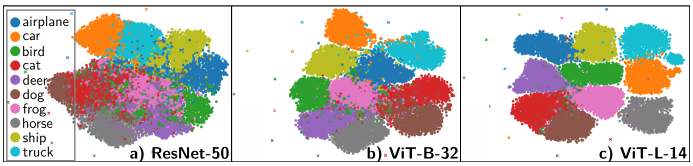


**Figure 4.** Visualization of the latent space representations of CIFAR-10 embeddings in different CLIP model backbones (one color per class), generated using the distance-preserving t-SNE dimensionality reduction method [58]. The better class separation in the transformer-based backbones (**b**), **c**)) are consistent with fidelity and human-alignedness results in Tabs. 1, 2.

E.g., `cat` may turn into "`animal, pet, cat`". The effect is that the obtained embedding (possibly after prompt finetuning as above) is shifted towards including the desired context; and all leaves together encode the complete ontology.

## 5 Experiments

### 5.1 Settings

**Models under test.** In our experiments, we utilized **CLIP** [45], one of the first multimodal foundation model family accepting both text and images [13]. For text-to-image alignment CLIP was trained to map an image and its corresponding text descriptions onto a similar (with respect to cosine similarity) latent space representation. This general-purpose model captures rich semantic information, and achieves impressive performance compared to task-specific models across various applications, including image captioning [7, 12], recognition of novel unseen objects [5], and retrieval tasks [6, 54]. This makes it a common choice as basis for training or distilling more specialized models [12, 13], and thus a highly interesting target for validation and verification of its learned knowledge and internalized QR. In our experiments, we explored various CLIP backbones, including ResNet-50, as well as Vision Transformer (ViT) variants featuring different patch sizes and model capacities (e.g., ViT-B/32, ViT-L/14)[9].

**Dataset.** The **CIFAR-10** dataset [4, Chap. 3] is a benchmark in the field of computer vision, consisting of 60,000 32×32 color images, split into 50,000 training and 10,000 test images. The images are equally distributed onto the 10 diverse classes `airplane`, `ship`, `car`, `truck`, `bird`, `cat`, `dog`, `deer`, `horse`, `frog`. The choice of classes suits our initial study well, as they both exhibit pairs of semantically similar objects (e.g., `car`, `truck`), as well as mostly unrelated ones (e.g., `car`, `cat`), so we can expect a deep class hierarchy. In our study, we conduct inference both of the baseline (naive zero-shot) and the proposed method on the CIFAR-10 test dataset [4].

**Fidelity baseline.** As discussed in subsection 4.3, the inference on the leaf concepts (**naive zero-shot** approach) serves as baseline (maximum performance) for fidelity measurements. The closer the tree inference gets to the naive zero-shot performance, the higher the fidelity. We here choose as distance metric the cosine distance $\text{CosDist}(a, b) := 1 - \frac{a \cdot b}{\|a\| \cdot \|b\|}$ (0 for $a, b$ parallel, 1 for orthogonal, 2 for $a = -b$), going along with the training of CLIP.

**Metrics.** Any quantitative classification performances are measured in terms of **accuracy** of the results on CIFAR-10 test images against their respective ground truth label.

[9] Pre-trained models and weights were obtained from: https://github.com/openai/CLIP

## 5.2 Ablation Study: Influences on Human-Alignedness and Fidelity of Ontology Extraction

As detailed in subsection 4.3, to measure the **human-alignedness** of the given multi-modal encoder model, we evaluated the performance when using our extracted ontology for inference of class labels on new images. And as a **fidelity indicator**, we measure the performance drop between inference on the leaves (naive zero-shot approach) against that of inference on our tree.[10] Both are measured in the course of an ablation study to identify the influence of different settings on the ontology's usefulness and quality.

**Investigated influences.** Both the ontology extraction by means of agglomerative hierarchical clustering (see subsection 3.3, as well as later the inference on new samples (see subsection 4.2) rely on measuring similarities between embedding vectors. However, due to being automatically optimized, the embeddings' optimal similarity metric is unknown. Hence, we treat each choice of similarity metric as a hyperparameter, and investigate their influence on human-alignedness of the extracted ontology:

- **Affinity:** Affinity typically influences which data points are most similar, i.e., closest related, in the final tree structure. In our experiments, we tested the standard Manhattan ($L_1$), Euclidean ($L_2$, and cosine distances.
- **Linkage:** This parameter determines the criterion used to merge clusters during the hierarchical clustering process, and in particular affects the shape and compactness of the clusters. In our experiments, we tested the standard settings of Ward, complete, average, and single linkages. Ward linkage minimizes the variance within clusters, while complete / average / single linkage focuses on the maximum / average / minimum distance between clusters.
- **Inference similarity:** We use use the same choices as for affinity.

Next, we compare different settings for obtaining the leaf embeddings. The following variants are considered:

- **Prompt tuning:** In case text embeddings are to be obtained, CLIP suggests using text prompts in the form *"a photo of a* `classname`*"* rather than simply *"*`classname`*"*, because the model is trained on image captions as text. If applied, this augmentation is done for both leaf and parent node textual embeddings.
- **Text encoding vs. few-shot image encoding:** As described in subsection 4.1, Step 1, the two different approaches to obtain leaf embeddings are text encoding and image encoding. We here only consider few-shot image encoding, i.e., specifying the concept via $< 10$ images, which ensures manageable complexity of the hierarchical clustering algorithm[11].

**Results.** An illustrative example of an ontology extracted from CLIP (ViT-L-14 backbone) using the prompt *"a photo of a* `classname`*"* is provided in Figure 2 for found-to-be-optimal settings according to the ablation study. Consistently optimal hyperparameter settings with respect to human-alignedness and fidelity turned out to be affinity=Manhattan, linkage=complete, and inference similarity=cosine, which were also used to create the remainder of the ablation studies. The accuracy results on CIFAR-10 of inference using the extracted ontology versus the naive-zero shot approach as a baseline for fidelity are given in Tabs. 1 for the prompt engineering settings, and 2 for the comparison of text and image encodings of the leafs.

*Please note that we did not yet conduct a cross-validation, so results should foremostly serve as guide for further investigations.*

[10] Performance against a ground truth is only a proxy; future experiments should directly compare predictions of the two.

[11] Standard implementations have a complexity of $\mathcal{O}(n^3)$ for $n$ leaf samples.

**Table 1.** Comparison of inference accuracy using naive zero-shot (Naive) and our method across different model architectures and textual prompt types. Fidelity calculated as ratio $\frac{\text{ours}}{\text{naive}} \in [0, 1]$; best models marked.

| | **Prompts** | | | | | |
|---|---|---|---|---|---|---|
| | *"*`classname`*"* | | | *"a photo of a* `classname`*"* | | |
| | **Naive** | **Ours** | *ratio* | **Naive** | **Ours** | *ratio* |
| **ResNet-50** | 0.70 | 0.46 | *0.66* | 0.69 | 0.67 | *0.97* |
| **ViT-B-32** | 0.87 | 0.82 | ***0.94*** | 0.89 | 0.85 | ***0.96*** |
| **ViT-L-14** | **0.91** | **0.85** | ***0.93*** | **0.95** | **0.92** | ***0.97*** |

**First findings.** In advance we manually validated the assumption of a good text-to-image alignment (Assumption 1(a)). For this we visualized the distribution and class separability of text and CIFAR-10 test sample embeddings in the latent spaces of the different CLIP backbones, results shown in Figure 4. The dimensionality-reduced visualizations suggest that with increasing parameter number, the clusters of different classes become more distinctly separated; and transformer-based backbones demonstrate superior separation. Notably, across all backbones, the text inputs and images are encoded in separate regions of the latent space, indicating a clear distinction between these two modalities in the model's internal representation.

The *prompt engineering*, i.e., replacing the text prompt "`classname`" with "*a photo of* `classname`" turned out to be have a strong *positive impact on human-alignedness and fidelity* in case of the worse aligned CNN-based CLIP backbone, and still a notable one for the already good transformer backbones.

In contrast, using *few images instead of text to obtain the leaf embedding resulted in worse performance*. However, in our initial tests performance seemed to increase with the number of images: Dropping the few-shot constraint showed competitive results. In the following table, we replaced the leaf node information with the randomly-sampled training images in the respective class.

**Table 2.** Comparison of inference accuracy for different ways to obtain the leaf embeddings: *few-shot* image embeddings vs. textual embeddings (*zero-shot*), with the naive zero-shot approach as baseline. Best model **bold**.

| | **Few-Shot** | | | **Zero-Shot** | |
|---|---|---|---|---|---|
| | **1-shot** | **5-shot** | **10-shot** | **Naive** | **Ours** |
| **ResNet-50** | 0.45 | 0.58 | 0.61 | 0.69 | 0.67 |
| **ViT-B-32** | **0.67** | **0.79** | **0.86** | 0.89 | 0.85 |
| **ViT-L-14** | 0.64 | 0.76 | 0.80 | **0.95** | **0.92** |

It should be noted, that a better performance of the textual embedding could possibly be attributed to a sub-optimal text-to-image alignment. This would be consistent with the insights into the distribution and class separability of image and text embeddings in the latent space in Figure 4 (with respect to Euclidean distance). It should be further investigated, whether this must be attributed to disparity in metrics, the domain shift to CIFAR-10 inputs, or could serve as an indicator for bad text-to-image alignment wrt. the considered classes.

## 5.3 Ontology validation and verification

**Validation: qualitative results.** A manual inspection of the obtained ontologies (see Figure 2 for an example) showed, that *good human-alignedness also coincides with seemingly valid tree structures*. Seemingly valid here means, that a human inspector can easily find convincing arguments for the validity most of the splitting criteria of the nodes. In Figure 2, two trees which are created with different parameters are compared. The tree on the left, which uses ViT-L/14 as a backbone, affinity clustering, and Manhattan linkage, achieves 92% accuracy on the classification task. In contrast, the tree

on the right, created with a ResNet-50 backbone, affinity clustering, and Euclidean linkage, yields an accuracy of 45%. One of the reasons for the low accuracy score in the classification task for the tree on the right is that its decision process does not align well with human-like decision-making. For example, the structure first checks whether an object is a *"vehicle"* and then whether it is *"meat"*. This decision process deviates from human-aligned reasoning, which can also be observed through manual inspection.

Furthermore, we identified the tendency that the *superior vision transformer backbones also showed the seemingly more valid tree structures*. This *possible architectural dependency of good ontological commitment* should be further investigated.

**Verification against a given ontology.** To exemplify the verification of ontological commitment against a given ontology, we chose the simple tree structure provided by [59] for CIFAR-10 dataset. To label the inner nodes of this tree, we utilized two external knowledge sources: WordNet [15] and GPT-4 [1], in each case bottom-to-top queried for a textual description of a parent for sibling nodes. We then used the ontology information to create contextualized leaf embeddings, as described in subsection 4.3, and applied naive zero-shot inference on these contextualized leaves. For WordNet, we labeled each node with the closest matching superclass. For GPT-4, we queried the model to provide the superclass of the given leaf nodes.

Initial verification results for the different given ontologies are shown in Table 3: As expected, using the extracted learned ontology for the contextualization caused no change compared to the baseline of non-contextualized embeddings; this contextualization is supposed to be equivalent to the non-contextualized leaf embeddings from the perspective of the model. However, the contextualization with external ontologies caused a strong drop in inference accuracy. A closer look at the results showed that those leaves with parents mentioning technical terms (e.g., "non-mammalian vertebrate") were mostly misclassified, indicating that the learned knowledge is inconsistent / not aware of these parts of the given ontologies. Further research is needed on practical implications (e.g., thus induced error cases), and how to align the ontologies.

**Table 3.** Verification results of different models against different sources of external ontologies: the NBDT tree structure [59] with *WordNet* [15] or *GPT-4* [1] queried node labels; versus no contextualization (*Naive*) and contextualization against the extracted ontology (*Ours*). Values are measured in inference accuracy on contextualized nodes.

| | WordNet | GPT-4 | Naive | Ours |
|---|---|---|---|---|
| **ResNet-50** | 0.31 | 0.36 | 0.69 | 0.67 |
| **ViT-B-32** | 0.40 | 0.53 | 0.89 | 0.85 |
| **ViT-L-14** | 0.52 | 0.54 | 0.95 | 0.92 |

# 6 Future work: Applications and next steps

## *6.1 Applications of learned ontology extraction*

Our method opens up several further interesting applications for the use of QR in DNN understanding, verification, and improvement.

**Optimal learned reasoning representations.** As discussed above, access to the internal ontology of a DNN is key to understand its internal QR. In particular, an open research question is, *what kind of concept representations are DNNs optimized for*, and, subsequently, *which kinds of reasoning would be supported by this?* For example, qualitative spatial reasoning would most benefit from a region-based representation of concepts, while cone-based reasoning from cones as representations [38]. The quantitative measurement of ontological commitment allows to do ablation studies on different representations of concepts and relations, e.g., different similarity measures.

**DNN inspection.** The obtained ontologies open up new inspection possibilities for DNNs. An interesting one could be to generate **contrastive examples** [21]: Change a given input minimally such that the class/superclass changes, possibly under a constraint to remain within a given superclass. Also, one could globally test the models against biases towards scenerios respectively background. A bias is uncovered, if the commonality of two classes is based on background rather than functionally relevant features; possibly supported on test samples generated by inpainting techniques. Unfortunately, the text-to-image alignment training of foundation models may easily introduce such a bias, as concepts occurring in similar image scenarios additionally will occur in similar textual context. E.g., one may expect `cat` and `dog` to be similar, as both often occur indoors.

**Knowledge insertion.** The final goal of the introspection discussed above should be to not only be able to verify the learned ontological commitment, but also to control both the commitment, and subsequently the learned reasoning. This might be achieved by adding penalties during training, determined by iterative ontology extraction and model finetuning. Thus, a foundation model with acceptable ontological commitment may be obtained. Lastly, to distill this knowledge of the large model into smaller specialized models, standard model distillation techniques could be amended [39]. Concretely, regularization terms can be added to (1) enforce that correspondences to some/most of the concepts, and to (2) enforce respective similarities and other relationships between the concepts.

## *6.2 Next steps*

Our initial experiments are clearly limited in their extend, so immediate next steps should encompass more experiments on measuring **human-alignedness** respectively a larger ablation study on possible influence of the made assumptions. Such can be domain shifts, like text-to-image, and real-to-synthetic image. Experiments should include user studies, and comparison to existing ontologies; Similarly, the **outlier detection and handling** capabilities of ontologies should be further investigated, both for novel as well as novel blended classes. Lastly, it can be investigated how to extend the here proposed approach **from multimodal models to unimodal** ones, allowing to compare the ontologies of large foundation models against that of state-of-practice small and efficient object detectors.

# 7 Conclusion

Altogether, this paper tackles the problem how to validate and verify a multimodal DNN's learned knowledge using QR. Concretely, we take the step to unveil the ontological commitment of DNNs, i.e., the learned concepts and (here: superclass-)relations. For this, we proposed a simple yet effective approach to (1) uncover yet undiscovered superclasses of given subclasses as used by the DNN; and to (2) extract a full hierarchical class tree with the `IsSuperClass`-relationships; together with means to verify and validate the extracted part of the learned ontology. Even though this initial proof-of-concept still relies on some simplifications, our initial experiments could already extract meaningful class hierarchies from concurrent multimodal DNNs, and reveal inconsistencies with existing ontologies. These may serve as a basis to access further insights into the ontological commitment of DNNs, and subsequently validate and

verify its learned QR. We are confident that, eventually, this could allow to control, i.e., correct and integrate, valuable prior knowledge from QR into DNNs, creating powerful yet verifiable and efficient hybrid systems. Thus, we hope to spark further interest into interdisciplinary research of QR for verification of DNNs within the QR community.

## Acknowledgements

The paper was written in the context of the "NXT GEN AI Methods" research project funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK), The authors would like to thank the consortium for the successful cooperation.

## References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] A. Agafonov and A. Ponomarev. An Experiment on Localization of Ontology Concepts in Deep Convolutional Neural Networks. In *Proc. 11th Int. Symposium on Information and Communication Technology*, SoICT '22, pages 82–87. Assoc. for Computing Machinery, 2022. doi: 10.1145/3568562.3568602.

[3] F. Aghaeipoor, M. Sabokrou, and A. Fernández. Fuzzy rule-based explainer systems for deep neural networks: From local explainability to global understanding. *IEEE Transactions on Fuzzy Systems*, 2023.

[4] K. Alex. Learning Multiple Layers of Features from Tiny Images. Master's thesis, University of Toronto, Canada, Apr. 2009.

[5] N. Amini-Naieni, K. Amini-Naieni, T. Han, and A. Zisserman. Open-world text-specifed object counting. In *BMVC*, 2023.

[6] A. Baldrati, M. Bertini, T. Uricchio, and A. Del Bimbo. Effective conditioned and composed image retrieval combining clip-based features. In *Proc. IEEE/CVF Conf. CVPR*, pages 21466–21474, 2022.

[7] M. Barraco, M. Cornia, S. Cascianelli, L. Baraldi, and R. Cucchiara. The unreasonable effectiveness of clip features for image captioning: an experimental analysis. In *Proc. IEEE/CVF Conf. CVPR*, 2022.

[8] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proc. 2017 IEEE Conf. CVPR*, pages 3319–3327. IEEE Comput. Society, 2017. doi: 10.1109/CVPR.2017.354.

[9] E. M. Bender and A. Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proc. 58th Ann. Meeting ACL*, pages 5185–5198. ACL, 2020. doi: 10.18653/v1/2020.acl-main.463.

[10] R. Bommasani and et al. On the Opportunities and Risks of Foundation Models, 2022.

[11] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proc. IEEE/CVF Int. Conf. Comput. vision*, pages 9650–9660, 2021.

[12] J. Cho, S. Yoon, A. Kale, F. Dernoncourt, T. Bui, and M. Bansal. Fine-grained Image Captioning with CLIP Reward. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 517–527. Association for Computational Linguistics, July 2022. doi: 10.18653/v1/2022.findings-naacl.39.

[13] CMA. AI Foundation Models: Initial Report. Technical report, Competition & Markets Authority, UK, Sept. 2023.

[14] I. Donadello, L. Serafini, and A. S. d'Avila Garcez. Logic tensor networks for semantic image interpretation. In *Proc. 26th Int. Joint Conf. Artificial Intelligence*, pages 1596–1602. ijcai.org, 2017. doi: 10.24963/ijcai.2017/221.

[15] C. Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.

[16] R. Fong and A. Vedaldi. Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proc. 2018 IEEE Conf. CVPR*, pages 8730–8738. IEEE Comput. Society, 2018. doi: 10.1109/CVPR.2018.00910.

[17] Y. Ge and et al. Improving Zero-Shot Generalization and Robustness of Multi-Modal Models. In *Proc. IEEE/CVF Conf. CVPR*, pages 11093–11101, 2023.

[18] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems 32*, pages 9273–9282, 2019.

[19] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

[20] N. Guarino. Formal Ontologies and Information Systems. In *Proc. FOIS'98*, pages 3–15. IOS Press, June 1998.

[21] R. Guidotti. Counterfactual explanations and how to find them: Literature review and benchmarking. *Data Mining and Knowledge Discovery*, 2022. doi: 10.1007/s10618-022-00831-6.

[22] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. XAI—Explainable artificial intelligence. *Science Robotics*, 4(37), 2019. doi: 10.1126/scirobotics.aay7120.

[23] C. He, M. Ma, and P. Wang. Extract interpretability-accuracy balanced rules from artificial neural networks: A review. *Neurocomputing*, 387: 346–358, 2020.

[24] J. He, S. Yang, S. Yang, A. Kortylewski, X. Yuan, J.-N. Chen, S. Liu, C. Yang, Q. Yu, and A. Yuille. PartImageNet: A Large, High-Quality Dataset of Parts. In *ECCV 2022*, pages 128–145. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-20074-8_8.

[25] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. doi: 10.1016/0893-6080(91)90009-T.

[26] M. Keser, G. Schwalbe, A. Nowzad, and A. Knoll. Interpretable model-agnostic plausibility verification for 2d object detectors using domain-invariant concept bottleneck models. In *Proc. IEEE/CVF Conf. CVPR*, pages 3890–3899, 2023.

[27] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proc. 35th Int. Conf. Machine Learning*, volume 80 of *Proc. Machine Learning Research*, pages 2668–2677. PMLR, 2018.

[28] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick. Segment Anything. In *Proc. IEEE/CVF Int. Conf. on Comput. Vision*, pages 4015–4026, 2023.

[29] J. H. Lee, S. Lanza, and S. Wermter. From Neural Activations to Concepts: A Survey on Explaining Concepts in Neural Networks, 2024.

[30] D. B. Lenat. *Building Large Knowledge-Based Systems*. Addison-Wesley Pub. Co., 1989. ISBN 978-0-201-51752-1.

[31] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *Int. Conf. Learning Representations*, 2018.

[32] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proc. 2013 Conf. North American Chapter ACL: Human Language Technologies*, pages 746–751. ACL, 2013.

[33] G. Mikriukov, G. Schwalbe, C. Hellert, and K. Bade. GCPV: Guided Concept Projection Vectors for the Explainable Inspection of CNN Feature Spaces, 2023.

[34] G. Mikriukov, G. Schwalbe, C. Hellert, and K. Bade. Revealing similar semantics inside CNNs: An interpretable concept-based comparison of feature spaces. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer, 2023.

[35] I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 2–9. Assoc. for Computing Machinery, 2001. doi: 10.1145/505168.505170.

[36] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. doi: 10.23915/distill.00007.

[37] D. W. Otter, J. R. Medina, and J. K. Kalita. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2021. doi: 10.1109/TNNLS.2020.2979670.

[38] Ö. L. Özcep, M. Leemhuis, and D. Wolter. Embedding Ontologies in the Description Logic ALC by Axis-Aligned Cones. *JAIR*, 78:217–267, 2023. doi: 10.1613/jair.1.13939.

[39] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016. doi: 10.1109/SP.2016.41.

[40] E. Poeta, G. Ciravegna, E. Pastor, T. Cerquitelli, and E. Baralis. Concept-based Explainable Artificial Intelligence: A Survey, 2023.

[41] A. F. Posada-Moreno, N. Surya, and S. Trimpe. ECLAD: Extracting Concepts with Local Aggregated Descriptors. *Pattern Recognition*, 147: 110146, 2024. doi: 10.1016/j.patcog.2023.110146.

[42] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[43] H. Prade and G. Richard. Analogical proportions: Why they are useful in AI. In *Proc. 30th Int. Joint Conf. Artificial Intelligence, IJCAI 2021*, pages 4568–4576. ijcai.org, 2021. doi: 10.24963/IJCAI.2021/621.

[44] J. Rabold, G. Schwalbe, and U. Schmid. Expressive explanations of DNNs by combining concept analysis with ILP. In *KI 2020: Advances in Artificial Intelligence*, Lecture Notes in Comput. Science, pages 148–162. Springer, 2020. doi: 10.1007/978-3-030-58285-2_11.

[45] A. Radford and et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. 38th Int. Conf. on Machine Learning*, pages 8748–8763. PMLR, 2021.

[46] X. Ran, Y. Xi, Y. Lu, X. Wang, and Z. Lu. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *AI Review*, 56(8):8219–8264, 2023. doi: 10.1007/s10462-022-10366-3.

[47] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144. ACM, 2016. doi: 10.1145/2939672.2939778.

[48] G. Schwalbe. Verification of size invariance in DNN activations using concept embeddings. In *Artificial Intelligence Applications and Innovations*, IFIP Advances in Information and Communication Technology, pages 374–386. Springer, 2021. doi: 10.1007/978-3-030-79150-6_30.

[49] G. Schwalbe. Concept Embedding Analysis: A Review. *arXiv:2203.13909 [cs, stat]*, 2022.

[50] G. Schwalbe and B. Finzel. A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 2023. doi: 10.1007/s10618-022-00867-8.

[51] G. Schwalbe, B. Knie, T. Sämann, T. Dobberphul, L. Gauerhof, S. Raafatnia, and V. Rocco. Structuring the safety argumentation for deep neural network based perception in automotive applications. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, pages 383–394. Springer, 2020. doi: 10.1007/978-3-030-55583-2_29.

[52] G. Schwalbe, C. Wirth, and U. Schmid. Enabling verification of deep neural networks in perception tasks using fuzzy logic and concept embeddings, 2022.

[53] R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *Proc. AAAI Conf. Artificial Intelligence*, 31(1), 2017. doi: 10.1609/aaai.v31i1.11164.

[54] M. Sultan, L. Jacobs, A. Stylianou, and R. Pless. Exploring clip for real world, text-based image retrieval. In *2023 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2023.

[55] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[56] M. Tan, R. Pang, and Q. V. Le. EfficientDet: Scalable and efficient object detection. In *Proc. 2020 IEEE/CVF Conf. CVPR*, pages 10781–10790, 2020.

[57] A. Tversky. Features of similarity, 1977. ISSN 1939-1471.

[58] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[59] A. Wan, L. Dunlap, D. Ho, J. Yin, S. Lee, S. Petryk, S. A. Bargal, and J. E. Gonzalez. NBDT: Neural-backed decision tree. In *Posters 2021 Int. Conf. Learning Representations*, 2020.

[60] D. Wang, X. Cui, and Z. J. Wang. CHAIN: Concept-harmonized hierarchical inference interpretation of deep convolutional neural networks. *CoRR*, abs/2002.01660, 2020.

[61] J. H. Ward Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. doi: 10.1080/01621459.1963.10500845.

[62] M. Yuksekgonul, M. Wang, and J. Zou. Post-hoc Concept Bottleneck Models. In *ICLR 2022 Workshop on PAIR2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022. doi: 10.48550/arXiv.2205.15480.

[63] R. Zhang, P. Madumal, T. Miller, K. A. Ehinger, and B. I. P. Rubinstein. Invertible concept-based explanations for CNN models with non-negative concept activation vectors. In *Proc. 35th AAAI Conf. Artificial Intelligence*, volume 35, pages 11682–11690. AAAI Press, 2021.