Learning Visually Grounded Domain Ontologies via Embodied Conversation and Explanation

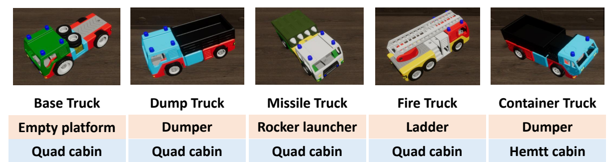
Jonghyuk Park, Alex Lascarides, Subramanian Ramamoorthy

School of Informatics, The University of Edinburgh
{jay.jh.park, a.lascarides, s.ramamoorthy}@ed.ac.uk

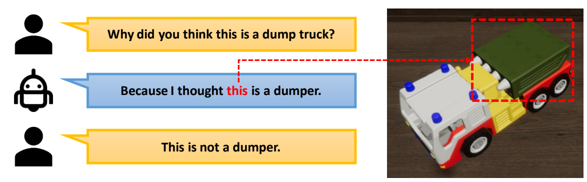
Abstract

In this paper, we offer a learning framework in which the agent's knowledge gaps are overcome through corrective feedback from a teacher whenever the agent explains its (incorrect) predictions. We test it in a low-resource visual processing scenario, in which the agent must learn to recognize distinct types of toy truck. The agent starts the learning process with no ontology about what types of truck exist nor which parts they have, and a deficient model for recognizing those parts from visual input. The teacher's feedback to the agent's explanations addresses its lack of relevant knowledge in the ontology via a generic rule (e.g., "dump trucks have dumpers"), whereas an inaccurate part recognition is corrected by a deictic statement (e.g., "this is not a dumper"). The learner utilizes this feedback not only to improve its estimate of the hypothesis space of possible domain ontologies and probability distributions over them but also to use those estimates to update its visual interpretation of the scene. Our experiments demonstrate that teacher-learner pairs utilizing explanations and corrections are more data-efficient than those without such a faculty.

Code — https://github.com/jpstyle/ns-arch-unity



(a) 3D models of toy trucks characterized by parts.



(b) Example interaction between a teacher and a learner featuring a part-based explanation and corrective feedback.

Figure 1: Explanatory interactive learning in a simulated domain.

Introduction

The field of eXplainable Artificial Intelligence (XAI) aims to develop AI systems that can explain their decisions, such that any identified knowledge gaps can be closed to improve performance on various dimensions: e.g., task success rates, faster convergence, or closer alignment with human knowledge. However, many XAI works focus only on identifying knowledge gaps, not exploiting those gaps in a principled way as learning opportunities (Weber et al. 2023). Furthermore, the currently most popular ML models that require explanations are deep neural networks (DNNs), which generally need many iterations of parameter updates to incur significant behavior changes. Providing human feedback to every single model explanation would be infeasible for DNNs, so most existing work on XAI-based model improvements aims to involve as little human involvement as possible during training.

In the meantime, feedback by a human-in-the-loop in response to XAI explanations could prove much more valuable when the learner has to quickly adapt to an unfamiliar domain with a continuously changing hypothesis space. Consider, for example, a general-purpose robotic agent deployed in an assembly plant where task concepts may be frequently added or modified in unforeseen ways, and supervision by a human domain expert (perhaps without ML expertise) is available via natural dialogues. Concepts in the domain ontology may be niche and specific to this particular plant, making it difficult to address the learner's initial ignorance of the domain ontology with some existing pretrained ML models. In such scenarios, it would be desirable for the expert's corrective feedback to the agent's explanations (as demonstrated in Fig. 1b) to cause immediate behaviour changes without interrupting its operation, so as not to exhaust or irritate the human teacher. This motivation is particularly pertinent when dealing with highly specialized, low-resource domains, where it is difficult to prepare automated feedback to agent explanations.

In this paper, we propose a framework for explanatory interactive learning (XIL; cf. Teso et al. 2023), suited to a family of knowledge-based neurosymbolic AI systems. The primary strength of our approach is that the learner can start out with scarce symbolic knowledge of the domain ontology, completely unaware of many crucial concepts, let alone their relationships to each other. It learns online and incrementally through continued natural interaction with a human supervisor. This contrasts with a nontrivial assumption commonly made in most neurosymbolic architectures proposed so far, that the symbolic domain knowledge is either fully encoded or batch-learned before deployment.

We instantiate the framework in an example testbed scenario where a learner agent must acquire and distinguish among a set of novel visual concepts, namely a range of fine-grained types of toy vehicle. We empirically demonstrate how the learner's explanations can induce tailored feedback from the teacher that addresses the exposed knowledge gaps, leading to two distinct types of knowledge update: 1) acquisition of generic rule-like knowledge about 'have-a' relations between whole and part concepts; and 2) joint refinement of recognition capabilities of visual concepts, for both object wholes and their parts. Our experiments demonstrate that strategies exploiting agent explanations in this way accomplish significantly better performance after the same number of training examples, compared to baseline strategies that do not—especially when the learner's initial model for recognizing object parts is deficient.

Explanatory Interactive Learning

Testbed task: fine-grained visual classification

The main challenge of fine-grained visual classification (FGVC) is that instances of fine-grained subcategories exhibit small interclass variance and large intraclass variance (Wei et al. 2021). Recent development of powerful vision foundation models has enabled strong FGVC performance with only visual features and label annotations, adding a classification head on top of the pretrained models with weights frozen (Oquab et al. 2023), which we will employ as a 'vision-only' baseline in this study.

Formally, each input to our FGVC task is a pair $x_i = (\mathcal{I}_i, m_i)$ with an RGB image $\mathcal{I}_i$ and a binary mask m_i. Thus, x_i essentially refers to an object in an image. The anticipated task output $y_i \in C_{\text{fg\_type}}$ is the correct concept of the object instance x_i, where $C_{\text{fg\_type}}$ is a set of fine-grained visual concepts. One important difference between our problem setting versus traditional FGVC is that for us $C_{\text{fg\_type}}$ is not known to the learner at the start—it must be acquired through interaction with the teacher. The classification target concepts in our experiments are fine-grained types of toy truck as illustrated in Fig. 1a. Some features define the fine-grained types of truck and thus are crucial for their successful classification (types of load and cabin parts), while other irrelevant features may vary across the types and act as distracting signals (e.g., colors of different body parts, numbers and sizes of wheels). The learner starts out knowing that toy trucks have loads and cabins, albeit with imperfect recognition capability.

Dialogue Strategies

Fig. 2 depicts dialogue flows that we aim to cover with our proposed approach and baselines. In effect, each episode of

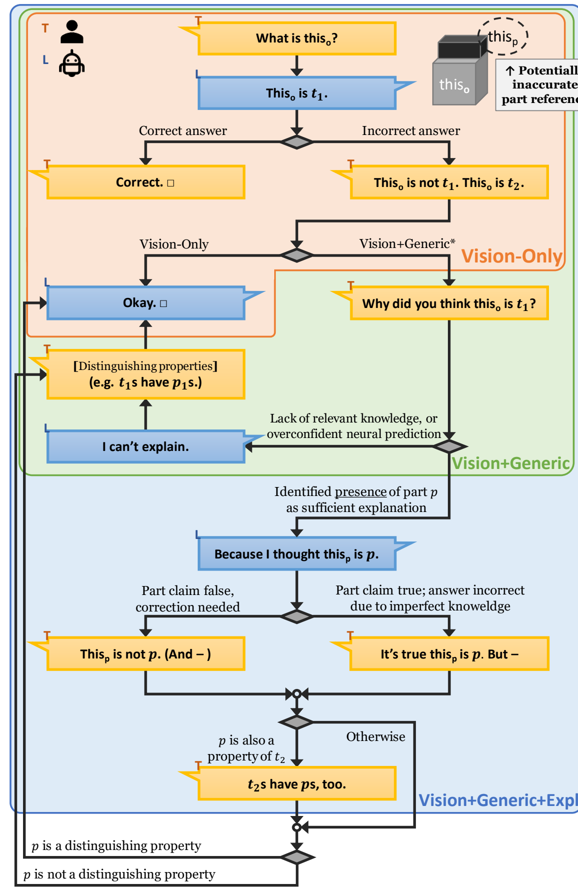


Figure 2: Flowchart covering the range of training dialogues modeled in this study. □ marks termination of an episode.

interaction is a single round of FGVC inference and learning with a training example, unrolled into a sequence of NL utterances exchanged between the teacher and the learner. An interaction episode starts with the teacher presenting an instance o of one of the target visual concepts in $C_{\text{fg\_type}}$. The teacher asks a probing question "What kind of truck is this$_o$?" along with an accurate binary mask specifying the object o demonstratively referenced by "this$_o$". The learner answers with its best estimate of the fine-grained type in $C_{\text{fg\_type}}$; e.g., "This$_o$ is a dump truck". This answer may be right or wrong; if the object is indeed a dump truck, then the episode terminates, and the teacher and learner proceed to the next episode with a new example. If the answer is wrong, then the teacher corrects the learner's answer by uttering, for instance, "This$_o$ is not a dump truck. This$_o$ is a missile truck".

After the teacher's correction, the dialogue branches according to whether the learner relies on direct neural prediction only or utilizes generic knowledge about distinctive part properties as well. The former case is a baseline, which we refer to as **Vis-Only**. No further interactions ensue, and the episode terminates. Otherwise, the teacher investigates the source of the error by asking "Why did you think this$_o$ is a dump truck?". This introduces another branching point that hinges upon whether the learner is able to explain the most relevant reason why it concluded so. We refer to the configuration where the learner lacks such a faculty as the **Vis+Genr** baseline, and the opposite case with the complete toolkit (i.e., our full approach) as **Vis+Genr+Expl**. There are two alternative scenarios in which the learner answers "I cannot explain" to the teacher's *why*-question. Either:

1. The episode takes place under **Vis+Genr** setting; or
2. The episode takes place under **Vis+Genr+Expl** setting, but the learner lacks relevant generic rules about whole-part relations in its KB, or the vision module made an overconfident (but wrong) prediction on the whole in spite of not recognizing its parts.

The teacher responds to "I cannot explain" with rule-like knowledge about properties that separate the truck types: e.g., "Missile trucks have rocket launchers, and dump trucks have dumpers". The episode then terminates.

If the learner has leveraged recognition of some truck part as evidence, the learner reports a part-based explanation such as "Because I thought this$_p$ is a dumper" while accompanying the demonstrative "this$_p$" with a pointing action that corresponds to its estimated binary mask. Note that this explanation implies the learner is aware of the rule "dump trucks have dumpers" and recognizes having dumpers as a distinguishing property.

There are various potential sources of error that can be exposed by this explanation. If the object referenced by "this$_p$" is not a dumper as claimed, or the estimated mask has poor quality and fails to specify an object at all, the explanation itself needs correction, and the teacher says "This$_p$ is not a dumper". The learner can then memorize the quoted part as a negative exemplar of the dumper concept. Alternatively, the part claim may be correct in its own right, but dumpers may not be a distinguishing feature between the learner's estimate concept and the ground truth concept. For instance, in the domain illustrated by Fig. 1a, dump trucks and container trucks both have dumpers but have different type of cabins: quad vs. hemtt. In this case, the teacher first acknowledges "It's true that this$_p$ is a dumper", then corrects the learner's reasoning by stating "But container trucks have dumpers, too", effectively depriving the 'dumper' concept of the role as a distinguishing property. Since the part cited as explanation is not a distinguishing property between the agent's answer vs. the true answer, it is implied that the learner may be ignorant of the true distinguishing feature. So the teacher provides the gap in the learner's knowledge via a further generic utterance: e.g., "Dump trucks have quad cabins while container trucks have hemtt cabins". The episode then terminates.

Neurosymbolic Agent Architecture

This section outlines our neurosymbolic architecture, assuming it is deployed to learn and perform image analysis tasks. However, note that in principle our framework is designed for knowledge-intensive tasks with any input modality (image, video, language, audio, etc.) where dialogue participants can naturally refer to parts of inputs. Fig. 3 illustrates the system components and their coordination in in-

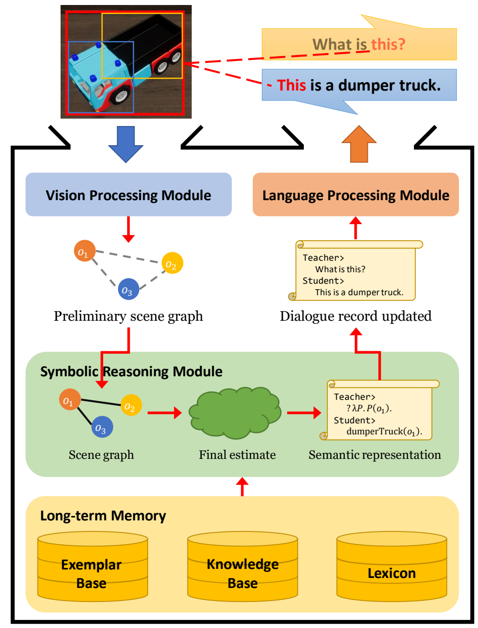


Figure 3: Overview of the proposed neurosymbolic architecture in inference mode, in which the component modules interact to generate an answer to a user question.

ference mode. As depicted, the agent architecture has four components with distinct responsibilities: vision processing, language processing, long-term memory and symbolic reasoning. The architecture is highly modular and can employ any existing methods for each component as long as they satisfy the feature requirements described in the subsections below.

The vision processing module

The vision processing module is responsible for parsing raw pixel-level inputs perceived by the system's vision sensor into structured representations. We use an object-centric graph-like data structure for summarizing visual scenes, which we will refer to as *scene graphs* hereafter. Scene graphs encode a set of salient objects in a visual scene and their conceptual relations (e.g., $dumpTruck(o_i)$, $have(o_i, o_j)$) along with likelihood scores. Each object in a scene graph is also associated with a demonstrative reference to it by segmentation mask. A scene graph serves as an internal, abstract representation of an image that will be later processed by symbolic reasoning methods.

The functional requirement we impose on the vision processing module is twofold. First, given a raw RGB image and a set of n binary masks $\{m_1, \cdots, m_n\}$ referring to scene objects, the vision module should be able to perform few-shot classification on visual concepts it's aware of, returning the estimated probabilities of concept membership as its output. Second, given a raw RGB image and a visual concept γ, the vision module should be able to localize and segment instances of γ in the visual scene in few shot (i.e., γ is specified by a moderately sized set of positive exemplars χ^+_γ), producing a set of corresponding binary masks

$\{m_1, \cdots, m_p\}$ for some p. This 'visual search' functionality is needed for recognizing object parts as evidence that would significantly affect later symbolic reasoning. For instance, if it is known that existence of a dumper on a truck will significantly raise the probability that the truck is a dump truck, instances of dumper should be searched for if not already included in the scene graph due to being less salient.

We will refer to the classification and segmentation functions defined for a concept γ as f^{γ}_{clf} and f^{γ}_{seg} hereafter. It is important that both f^{γ}_{clf} and f^{γ}_{seg} can operate few-shot, so as to enable quick, online acquisition of novel domain concepts during operation; the learner shouldn't require several thousands of annotated examples from the human user to acquire unforeseen domain concepts. We also need to ensure that f^{γ}_{clf} and f^{γ}_{seg} perform better as exemplar sets grow larger via continued interaction with the human user, so as to allow gradual improvement of object recognition from visual data.

Refer to the Technical Appendix in the extended version (Park, Lascarides, and Ramamoorthy 2024) for a more formal, detailed description.

The language processing module

The language processing module handles NL interactions with human users. Its core responsibilities include:

- Parsing free-form NL inputs from users into their semantic contents and pragmatic intents (natural language understanding; NLU).
- Incrementally updating semantic representations of states of ongoing dialogues and deciding which speech acts to perform from current states (dialogue management; DM).
- Generating NL utterances according to the DM's decisions about the agent's next speech act (natural language generation; NLG).

We draw on formal semantic representations for the NLU, DM and NLG components. In the scope of this work, each clause in the dialogues is either a proposition or a question. Suppose we have a standard first-order language $\mathcal{L}$ defined for describing scene graphs, including constants referring to objects and predicates referring to visual concepts. We will represent the semantic content of an indicative NL sentence with an antecedent-consequent pair of $\mathcal{L}$-formulae; we will refer to such a unit as a PROP hereafter. If ψ is a PROP comprising an antecedent $\mathcal{L}$-formula $Ante$ and a consequent $\mathcal{L}$-formula $Cons$, then ψ has the form $Ante \Rightarrow Cons$. We refer to $Ante$ and $Cons$ of ψ as $Ante(\psi)$ and $Cons(\psi)$ respectively. When a PROP ψ is a non-conditional, factual statement, $Ante(\psi)$ is empty and we omit the $Ante \Rightarrow$ part, leaving $Cons$. For example, the NL sentence "o is a dump truck" is represented as:

$$dumpTruck(o) \tag{1}$$

Quantified PROPs are marked with matching quantifiers. In this work, we are primarily concerned with generic statements and will let $\mathbb{G}$ denote the generic quantifier. Thus, for example, the NL sentence "Dump trucks (generally) have dumpers" can be represented as follows:

$$\mathbb{G}x.dumpTruck(x) \Rightarrow (\exists y.have(x,y) \wedge dumper(y)) \tag{2}$$

However, we go further and use a skolem function f in $\mathcal{L}$ that maps an instance of dump truck to its dumper:

$$\mathbb{G}x.dumpTruck(x) \Rightarrow have(x, f(x)) \wedge dumper(f(x)) \tag{3}$$

(3) is more compatible with the logic programming formalism that we adopt in the symbolic reasoning module.

Our semantic representation of questions, QUES hereafter, resembles the notation from Groenendijk and Stokhof (1982). A *wh*-question is represented as $?\lambda x.\psi(x)$ where ψ is a PROP in which the variable x is free. Its (partial) answers provide values a for x such that $\psi[x/a]$ evaluates to true. For example, we represent the NL sentence "What kind of truck is o?" by the following QUES:

$$?\lambda P.P(o) \wedge \vdash^{type}_{KB} (P, truck) \tag{4}$$

by taking the liberty of interpreting the NL question as meaning "Which concept P has o as an instance and entails that o is a truck?". Here, $\vdash^{type}_{KB}$ is a reserved predicate such that $\vdash^{type}_{KB} (p_1, p_2)$ evaluates to true if and only if p_1 is a subtype of p_2 according to the domain ontology represented by a knowledge base KB.

Another important type of question in the interactions from Fig. 2 is *why*-questions. We simply introduce another symbol $?why_{DP}$ for representing *why*-questions that query the source of belief by some dialogue participant DP. For example, we represent the NL question "Why did you think o is a dump truck?" addressed to the learner agent agt as:

$$?why_{agt}.dumpTruck(o) \tag{5}$$

The long-term memory module

The long-term memory module stores interpretable fragments of knowledge, which the learner has acquired from the teacher over the course of interaction. Three types of storage subcomponents are needed: the visual exemplar base (XB), the symbolic knowledge base (KB) and the lexicon.

The visual XB stores collections of positive and negative concept exemplars χ^+_γ and χ^-_γ for each visual concept γ. χ^+_γ and χ^-_γ naturally induce a binary classifier for the concept γ, which is used by f^{γ}_{clf} in the vision module. χ^+_γ and χ^-_γ are updated whenever the learner encounters an object and deems it worth remembering as a positive/negative sample of γ. Whenever the learner encounters an unforeseen concept γ via the teacher's utterance that features a word that's outside the learner's vocabulary (i.e., a neologism), it is registered as a new concept, and new empty sets χ^+_γ and χ^-_γ are created in the XB.

The symbolic KB stores a collection of generic rules about relations between concepts, represented as $\mathbb{G}$-quantified PROPs like (3). Generic rules are acquired via interactions with a teacher and stored in the KB. In our setting, the generic statements express 'have-a' relations between whole vs. part concepts.

The lexicon stores a mapping between visual concepts and NL content word strings. Neologisms referring to novel visual concepts are added to the lexicon when first mentioned by a teacher, accordingly updating the visual concept vocabulary and the XB.

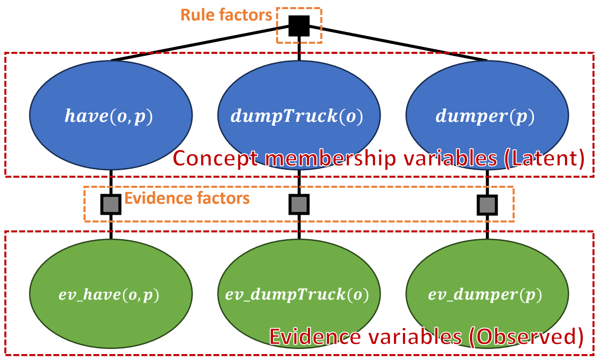


Figure 4: An example truck classification problem represented as a probabilistic factor graph model.

Symbolic reasoning module

The symbolic reasoning module combines subsymbolic perceptual inputs and symbolic relational knowledge. The 'first impressions' formed by the vision module are adjusted by the generic rules in KB to yield final estimations of the current image's interpretation. To cope with estimation of quantitative uncertainties on logical grounds, we use probabilistic graphical models for symbolic reasoning.

Fig. 4 illustrates a small example of how we represent an instance of a visual classification problem as a factor graph model. Our factor graph representations feature two types of variable nodes and two types of factor nodes. *Concept membership variables* represent whether an object (or an ordered tuple of objects) is 'truly' an instance of a concept. *Evidence variables* represent evidence with quantitative uncertainty, obtained through a process external to the scope of the reasoning problem (*virtual evidence*; cf. Pearl 1988), namely visual perception and neural prediction here. Each evidence variable is uniquely defined for and causally linked to the corresponding concept membership variable. *Evidence factors* connect each matching pair of a concept membership variable and an evidence variable. The strengths of observed evidence, which are exactly the probability values returned by f^{γ}_{clf} functions, are entered as input to corresponding evidence factors as likelihood ratios: e.g., $\Pr(ev\_dumper(p) = T | dumper(p) = T) : \Pr(ev\_dumper(p) = T | dumper(p) = F)$. *Rule factors* connect concept membership variables and represent KB entries that encode relational knowledge like (3).

We construct a graphical model for inference by first translating the scene graph and the symbolic KB entries into a probability-weighted normal logic program. The primary reason we employ normal logic programs as an intermediate representation is that we would like the generic PROPs in the KB to receive minimal model interpretations. In other words, we want to infer a rule consequent $Cons$ only when some rule antecedent $Ante$ that entails $Cons$ is proven to hold, as opposed to material implication in propositional logic and Markov logic (Richardson and Domingos 2006). This enables inference in the *abductive* direction, e.g., raising the probability of an object being a dump truck after recognizing a dumper as its part due to the rule "Dump trucks have dumpers". Default negations used in normal logic programs would also facilitate better treatment of the generic quantifier $\mathbb{G}$ through non-monotonic reasoning, though such use is outside the scope of this work.

Likelihood values encoded in a scene graph are compiled into a program Π_V that summarizes visual evidence. For instance, the likelihood that an object o is an instance of a concept γ is converted into a fragment of Π_V as follows:

$$\begin{array}{l} 0.5 \ :: \ \gamma(o)\,\texttt{.} \\ p \ :: \ ev\_\gamma(o) \leftarrow \gamma(o)\,\texttt{.} \\ 1-p \ :: \ ev\_\gamma(o) \leftarrow \texttt{not} \ \ \gamma(o)\,\texttt{.} \end{array}$$

The number annotated to the left of each rule indicates that the rule head can hold with probability equal to that value if and only if the rule body is satisfied. Note how the translation captures a Bayesian viewpoint; we can interpret that 0.5 in the first line represents a uniform prior, and p and $1-p$ (where p is obtained from f^{γ}_{clf}) represent data likelihoods.

The symbolic KB is translated into a program Π_K that implements logical inference in both deductive and abductive directions. For deductive inference, each KB entry κ adds to Π_K the following rule parametrized by $U_d \in [0, 1]$:

$$U_d \ :: \ \leftarrow Ante(\kappa), \ \texttt{not} \ \ Cons(\kappa)\,\texttt{.}$$

which penalizes 'deductive violation' of the entry κ. Here, a rule without a head represents an integrity constraint, as is standard in normal logic programs. For abductive inference, each set of KB entries $\{\kappa_i\}$ that share identical $Cons$ adds to Π_K the following rule parametrized by $U_a \in [0, 1]$:

$$U_a \ :: \ \leftarrow Cons, \bigwedge_{\{\kappa_i | Cons(\kappa_i) = Cons\}} \{\texttt{not} \ \ Ante(\kappa_i)\}\,\texttt{.}$$

which penalizes failure to explain $Cons$ observed. The parameters U_d and U_a can be understood as encoding the extent to which the agent relies on its symbolic knowledge. We fix $U_d = U_a = 0.99$ in our experiments.

After Π_V and Π_K are obtained, the program $\Pi_V \cup \Pi_K$ is translated into an equivalent factor graph. For instance, having (3) as a KB entry will add the subgraph shown in Fig. 4. The symbolic reasoning module needs to be equipped with an inference algorithm for probabilistic graphical models, which should be implemented so as to correctly accommodate the semantics of normal logic programs.

The agent's inference and learning procedures

We now describe how the architecture components coordinate in service to the four cognitive processes integral to our framework: inference, explanation, instance-level learning and rule-level learning.

Inference The agent performs FGVC inference upon parsing a probing QUES like (4) from the teacher. The vision module first prepares a scene graph that includes the classification target object o (referenced as "this$_o$" in Fig. 2). If the symbolic KB is not empty, potential instances of relevant part subtypes as determined by the KB are searched by f^{γ}_{seg} and added to the scene graph as well. Then, each concept membership probability for the scene objects are estimated by f^{γ}_{clf}, completing the scene graph SG. SG and the symbolic KB are each translated into probability-weighted normal logic programs Π_V and Π_K respectively. Note that $\Pi_K = \varnothing$ for the **Vis-Only** baseline. $\Pi_V \cup \Pi_K$ is converted into a factor graph, and the symbolic reasoning module executes its inference algorithm on the graph. Marginal probabilities are queried afterwards, then the candidate concept with the highest probability is selected as answer.

Explanation The explanation process is triggered when the agent parses a *why*-QUES like (5) from the teacher, upon which it invokes a dedicated procedure for reflecting upon its reasoning process to select a *sufficient reason* (Darwiche and Hirth 2020). For example, if the agent knows the generic rule that fire trucks have ladders, then identifying a ladder with high confidence would be a sufficient reason for recognizing the whole object as a fire truck in our domain. We implement a modified version of Koopman and Renooij (2021)'s algorithm for finding sufficient reasons from our factor graphs. We only consider the evidence variables in the factor graph (e.g., $ev\_dumper(p)$), which are observable, as potential explanans candidates. If the algorithm returns 'direct perceptions' as (part of) sufficient reasons, e.g., selecting $ev\_dumpTruck(o)$ as sufficient reason of the agent's answer $dumpTruck(o)$, they are deemed not informative and thus omitted from the explanation. The intuition is that explanations like 'I thought o is a dump truck because it looked like a dump truck' do not reveal a knowledge gap that the teacher can exploit as a learning opportunity.

Instance-level learning Instance-level learning updates the sets χ^+_γ, χ^-_γ stored in the XB and thus contributes to updating f^γ_{clf} and f^γ_{seg}. As the sizes of $\chi^{+/-}_\gamma$ get larger, the accuracy of these functions should increase.

χ^+_γ and χ^-_γ are updated whenever the agent makes an incorrect prediction. Specifically, the teacher's correction "This is not a dump truck. This is a missile truck" will add the referenced instance to $\chi^-_{dumpTruck}$ and $\chi^+_{missileTruck}$. On any occasion where the agent made a correct prediction that an instance is a dump truck, but was not highly confident in that prediction, the instance is added to $\chi^+_{dumpTruck}$. In this case, any recognized instances of relevant part concepts, as determined by the agent's current KB, are added to the XB as well; e.g., χ^+_{dumper} and $\chi^+_{quadCabin}$ would also be expanded in our example. Note that the estimated binary masks for the part instances may be faulty and thus allow introduction of bad exemplars in the XB. In fact, it is precisely these sources of error that we aim to mitigate via the teacher's corrective feedback in **Vis+Genr+Expl** dialogues. That is, the corrective feedback "This is not a quad cabin" to the explanation "Because I thought this is a quad cabin" will add the misclassified non-instance to $\chi^-_{quadCabin}$. This enables joint refinement of whole and part concepts.

Rule-level learning For rule-level learning, the symbolic KB is updated whenever the teacher utters a generic statement after the learner fails to provide some verbal explanation for its incorrect prediction. Each NL generic statement, like "Dump trucks have dumpers", is translated into a corresponding $\mathbb{G}$-quantified PROP like (3) and added to the KB entry if not already included. We assume the rules provided by the teacher are always correct. KB rule addition is an idempotent operation in this work; we do not take any measure when already known rules are provided again.

Experiments

Setup

We conduct a suite of experiments to assess the data-efficiency of the three different interaction strategies during FGVC training: **Vis-Only**, **Vis+Genr** and **Vis+Genr+Expl**. We only compare against ablative baselines because no other existing neurosymbolic approaches are designed to cope with incremental learning combined with a lack of symbolic domain ontology after deployment. We consider two FGVC domains with differing difficulties:

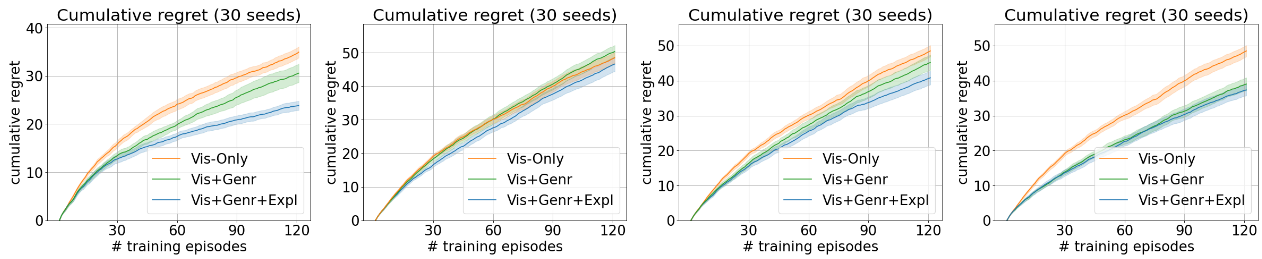
- single_4way: Four types of trucks can be distinguished solely by their load types.
- double_5way: Five types of trucks can be distinguished by two dimensions of part properties (load and cabin), where some parts may be shared between truck type pairs and thus can fail to serve as distinguishing.

We also vary the part-recognition performance of the vision model with which the learner starts each experiment, by controlling initial XB entries, in order to study how the initial recognition capability affects the learning process. We run experiments with three initial part recognition accuracies: LQ/MQ/HQ (low-/medium-/high-quality). Our evaluation metric is *cumulative regret*; i.e., the accumulated number of mistakes made across a series of 120 interaction episodes, where subtypes and visual features are randomly sampled for each training instance. We report cumulative regret curves averaged over 30 random seeds with 95% confidence intervals. Due to space limits, the Appendix (Park, Lascarides, and Ramamoorthy 2024) offers more details on experiment setup and agent implementation.

Results and discussion

Plots in Fig. 5 show how teacher-learner pairs adopting the three strategies acquire the target concepts at different rates. Fig. 5a shows that the order for data efficiency of learning in single_4way setting (starting from LQ) is: **Vis+Genr+Expl**, **Vis+Genr**, **Vis-Only**, from the most to the least efficient. Two-sample t-tests on their final cumulative regret values show these differences are all statistically significant ($p < 10^{-6}$), also attested by the wide separations between the 95% confidence intervals in Fig. 5a. Generic rules allow the learner to generalize better by attending to more important visual features (e.g., part types) while mitigating distractions by irrelevant features (e.g., colors). Teacher feedback to learner explanations provides even further synergies through improving part recognition models.

The results for the double_5way experiments are nuanced. In particular, notice in Fig. 5b how **Vis+Genr** performs worse than **Vis-Only**. This may be due to the fact that the cabin parts (quad vs. hemtt) are more difficult to tell apart: they look more similar to each other and have varying colors as distracting features. This suggests that depending on task difficulty, generic part-whole rules may actually cause an adverse effect on the learning progress if



(a) single_4way, LQ (b) double_5way, LQ (c) double_5way, MQ (d) double_5way, HQ

Figure 5: Averaged cumulative regret curves (with 95% confidence intervals): training example count vs. cumulative regret.

recognition capabilities for auxiliary feature concepts are too inaccurate and not properly refined during learning. **Vis+Genr+Expl** manages to take back the (marginal) comparative advantage in data efficiency by joint refinement of concept boundaries for both object parts and wholes.

Fig. 5c and Fig. 5d show that when we start with better part recognition models—namely, the MQ and HQ models—then generic rules improve data efficiency again. Notice how **Vis+Genr+Expl** consistently outperforms **Vis+Genr** with all LQ, MQ and HQ models, though the confidence intervals start to overlap in Fig. 5d. Two-sample t-tests on final regret values between **Vis+Genr** vs. **Vis+Genr+Expl** with LQ/MQ/HQ as starting model yield p-values of 0.015/0.005/0.187 respectively. This makes sense, considering the fact that the HQ part model already has strong part recognition capability and leaves relatively little room for improvement. Overall, one needs generics combined with explanations to *guarantee* that improving ontological knowledge enhances data efficiency in visual tasks, regardless of the quality of the initial vision model.

Related Work

A line of research investigates a framework often referred to as explanatory interactive learning (XIL), which focuses on the corrective role of explanations in teacher-learner interactions (Teso et al. 2023). CAIPI (Teso and Kersting 2019) is a model-agnostic method that converts local explanations to counterexamples, but it cannot analytically state how agent explanations should be corrected at the feature level. Schramowski et al. (2020) extend CAIPI by allowing users to interact with explanations given as input gradients at the expense of model-agnosticity. ProtoPDebug (Bontempelli et al. 2023) makes explicit references to visual parts for human-in-the-loop debugging of neural FGVC models, but not in the form of logical rules. NeSy XIL (Stammer, Schramowski, and Kersting 2021) bears a strong resemblance to our approach in that it implements a neurosymbolic architecture which admits rule-based correction from local explanations. However, NeSy XIL heavily relies on the traditional DNN training scheme and thus is more suitable for static task domains where human inspections are needed only occasionally. Michael (2019) and Tsamoura, Hospedales, and Michael (2021) are another relevant line of work with a heavy emphasis on neurosymbolic architectures and argumentation-based model coaching, but they are also chiefly concerned with guiding parameter updates of neural models in static domains. HELPER (Sarch et al. 2023) is another closely relevant approach to building customizable agents with memory-augmented large neural models, but it is designed to learn novel user-specific action routines, while our approach focuses on enabling a pretrained neural model to acquire concepts it was completely unaware of.

Interactive task learning (ITL; Laird et al. 2017) is a ML paradigm motivated by scenarios where an AI system needs to cope with unforeseen changes after deployment. Learning in ITL takes place primarily through natural interactions with a human domain expert. Therefore, it is natural for ITL to utilize insights from linguistic phenomena, such as discourse coherence (Appelgren and Lascarides 2020) and complex referential expressions (Rubavicius and Lascarides 2022). We contribute to this body of work by extending the neurosymbolic ITL architecture proposed in Park, Lascarides, and Ramamoorthy (2023), which exploits the truth-conditional semantics of generic statements; our extension adds corrective feedback to the agent's explanations as further learning opportunities.

Conclusion and Future Directions

In this paper, we have proposed an explanatory interactive learning framework for neurosymbolic architectures designed to solve knowledge-intensive tasks. It uses natural language dialogues through which the teacher provides feedback to the learner's explanations of its (incorrect) predictions. This feedback provides piecemeal information of domain ontologies, enabling incremental refinement of imperfect visual concept boundaries as and when needed. Symbolic reasoning enhances generalizability by explicitly highlighting important feature concepts, while correction of inaccurate part-based explanations ensures the referenced visual evidence is of good quality. Our proof-of-concept experiments show that timely exploitation of explanations can significantly boost learning efficiency in tasks where knowledge of part-whole relations is integral, although such improvements may be conditional upon the quality of the part recognition model. Potential future research directions include: in-depth exploration of exception-admitting generics (Pelletier and Asher 1997), extension to continuous regression tasks (Letzgus et al. 2022) and long-horizon planning tasks (Fox, Long, and Magazzeni 2017).

Acknowledgements

This work was supported by Informatics Global PhD Scholarships, funded by the School of Informatics at The University of Edinburgh. Ramamoorthy is supported by a UKRI Turing AI World Leading Researcher Fellowship on AI for Person-Centred and Teachable Autonomy (grant EP/Z534833/1). We thank the anonymous reviewers for their feedback on an earlier draft of this paper, and Rimvydas Rubavicius and Gautier Dagan for continued feedback over the course of this research.

References

Appelgren, M.; and Lascarides, A. 2020. Interactive task learning via embodied corrective feedback. *Autonomous Agents and Multi-Agent Systems*, 34(2): 1–45.

Bontempelli, A.; Teso, S.; Tentori, K.; Giunchiglia, F.; Passerini, A.; et al. 2023. Concept-level debugging of part-prototype networks. In *Proceedings of the The Eleventh International Conference on Learning Representations (ICLR 23)*. ICLR 2023.

Darwiche, A.; and Hirth, A. 2020. On the Reasons Behind Decisions. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325, 712–720.

Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable planning. In *Proceedings of the IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*.

Groenendijk, J.; and Stokhof, M. 1982. Semantic analysis of" wh"-complements. *Linguistics and philosophy*, 175–233.

Koopman, T.; and Renooij, S. 2021. Persuasive contrastive explanations for Bayesian networks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21–24, 2021, Proceedings 16*, 229–242. Springer.

Laird, J. E.; Gluck, K.; Anderson, J.; Forbus, K. D.; Jenkins, O. C.; Lebiere, C.; Salvucci, D.; Scheutz, M.; Thomaz, A.; Trafton, G.; et al. 2017. Interactive task learning. *IEEE Intelligent Systems*, 32(4): 6–21.

Letzgus, S.; Wagner, P.; Lederer, J.; Samek, W.; Müller, K.-R.; and Montavon, G. 2022. Toward explainable artificial intelligence for regression models: A methodological perspective. *IEEE Signal Processing Magazine*, 39(4): 40–58.

Michael, L. 2019. Machine coaching. In *Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI)*, 80–86.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H. V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; HAZIZA, D.; Massa, F.; El-Nouby, A.; et al. 2023. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*.

Park, J.; Lascarides, A.; and Ramamoorthy, S. 2023. Interactive Acquisition of Fine-grained Visual Concepts by Exploiting Semantics of Generic Characterizations in Discourse. In Amblard, M.; and Breitholtz, E., eds., *Proceedings of the 15th International Conference on Computational Semantics*, 318–331. Nancy, France: Association for Computational Linguistics.

Park, J.; Lascarides, A.; and Ramamoorthy, S. 2024. Learning Visually Grounded Domain Ontologies via Embodied Conversation and Explanation (Extended Version). arXiv:2412.09770.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann.

Pelletier, F. J.; and Asher, N. 1997. Generics and defaults. In *Handbook of logic and language*, 1125–1177. Elsevier.

Richardson, M.; and Domingos, P. 2006. Markov logic networks. *Machine learning*, 62: 107–136.

Rubavicius, R.; and Lascarides, A. 2022. Interactive Symbol Grounding with Complex Referential Expressions. In *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Sarch, G.; Wu, Y.; Tarr, M.; and Fragkiadaki, K. 2023. Open-Ended Instructable Embodied Agents with Memory-Augmented Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 3468–3500.

Schramowski, P.; Stammer, W.; Teso, S.; Brugger, A.; Herbert, F.; Shao, X.; Luigs, H.-G.; Mahlein, A.-K.; and Kersting, K. 2020. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8): 476–486.

Stammer, W.; Schramowski, P.; and Kersting, K. 2021. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3619–3629.

Teso, S.; Alkan, Ö.; Stammer, W.; and Daly, E. 2023. Leveraging explanations in interactive machine learning: An overview. *Frontiers in Artificial Intelligence*, 6: 1066049.

Teso, S.; and Kersting, K. 2019. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 239–245.

Tsamoura, E.; Hospedales, T.; and Michael, L. 2021. Neural-symbolic integration: A compositional perspective. *Proceedings of the AAAI conference on artificial intelligence*, 35(6): 5051–5060.

Weber, L.; Lapuschkin, S.; Binder, A.; and Samek, W. 2023. Beyond explaining: Opportunities and challenges of XAI-based model improvement. *Information Fusion*, 92: 154–176.

Wei, X.-S.; Song, Y.-Z.; Mac Aodha, O.; Wu, J.; Peng, Y.; Tang, J.; Yang, J.; and Belongie, S. 2021. Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.