

Continual Reasoning: Non-monotonic Reasoning in Neurosymbolic AI using Continual Learning

Sofoklis Kyriakopoulos^{1,*}, Artur S. d'Avila Garcez¹

¹Department of Computer Science, City, University of London, London, UK

Abstract

Despite the extensive investment and impressive recent progress at reasoning by similarity, deep learning continues to struggle with more complex forms of reasoning such as non-monotonic and common-sense reasoning. Non-monotonicity is a property of non-classical reasoning typically seen in common-sense reasoning, whereby a reasoning system is allowed (differently from classical logic) to *jump to conclusions* which may be retracted later, when new information becomes available. Neural-symbolic systems such as Logic Tensor Networks (LTN) have been shown to be effective at enabling deep neural networks to achieve reasoning capabilities. In this paper, we show that by combining a neural-symbolic system with methods from continual learning, LTN can obtain a higher level of accuracy when addressing non-monotonic reasoning tasks. Continual learning is added to LTNs by adopting a curriculum of learning from knowledge and data with recall. We call this process *Continual Reasoning*, a new methodology for the application of neural-symbolic systems to reasoning tasks. Continual Reasoning is applied to a prototypical non-monotonic reasoning problem as well as other reasoning examples. Experimentation is conducted to compare and analyze the effects that different curriculum choices may have on overall learning and reasoning results. Results indicate significant improvement on the prototypical non-monotonic reasoning problem and a promising outlook for the proposed approach on statistical relational learning examples.

Keywords

Neural-Symbolic Systems, Continual Learning, Non-monotonic Reasoning, Logic Tensor Networks

1. Introduction

The combination of machine learning and symbolic reasoning, now embodied by the area known as neurosymbolic AI, has been a developing field of research since the early days of AI. Recent advancements in deep learning allowed for a surge of interest in this particular type of models. Many variations of neural-symbolic (NeSy) models have surfaced in the past few years, showing the advantages of NeSy systems at reasoning and learning with increased explainability, data efficiency and generalization in comparison with other deep learning models [1, 2, 3, 4].

In this paper we propose **Continual Reasoning**, a new paradigm of learning for NeSy models to achieve *non-monotonic reasoning* (NMR). The core principle of Continual Reasoning states that reasoning tasks, especially those of a non-monotonic nature, should be addressed by learning from data and knowledge in a multi-stage curriculum of training. We illustrate this learning paradigm using a combination of Logic Tensor Networks (LTN) [5], a NeSy framework

NeSy 2023: 17th International Workshop on Neural-Symbolic Learning and Reasoning, 3–5 July, 2023, Sienna, Italy

*Corresponding author.

✉ sofoklis.kyriakopoulos@city.ac.uk (S. Kyriakopoulos); a.garcez@city.ac.uk (A. S. d. Garcez)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

capable of simulating First-Order Logic (FOL), and methodologies borrowed from Continual Learning (CL) for deep learning [6]. LTN is chosen for its ability to constrain the loss calculations of a deep learning system based on symbolic knowledge defined in FOL and its effectiveness in dealing with both typical deep learning and reasoning tasks [5, 7, 8]. CL, that is, the sequential learning of knowledge, without forgetting, from data that may no longer be available, will be shown to implement non-monotonicity in LTNs efficiently, when adopting an appropriate learning curriculum. Continual Reasoning combining LTN and CL aims to address the difficulties that many NeSy models have when dealing with non-monotonic tasks.

We apply and evaluate Continual Reasoning on an exemplar NMR task (the birds and penguins example), on the Smokers and Friends statistical relational reasoning task [9], and on a Natural Language Understanding (NLU) task that contains NMR (from the bAbI dataset [10]). Results indicate that a considerable increase in accuracy can be achieved in comparison with a single-stage curriculum of learning.

The remainder of this paper is organised as follows. In Section 2, we discuss the challenges faced by previous approaches to NMR. In Section 3, we introduce the Continual Reasoning methodology and two general approaches to curriculum design. In Section 4, we analyze the experimental results. Section 5 concludes the paper and discusses directions for future work.

2. Background

A common scenario to explain NMR is the *Penguin Exception Task* (PET) [11, 12], which can be defined in simple terms as: *In a group of animals, there exist birds and non-birds. It is known that normally all birds fly, and that all non-birds do not fly. However, it is also known that penguins are animals that are birds, but do not fly.* In First-Order Logic (FOL), the PET can be defined using axioms such as $\forall X (is_bird(X) \rightarrow can_fly(X))$ and $\exists X (is_penguin(X) \wedge is_bird(X))$, etc. The idea is that in the absence of further information, it is reasonable to assume that all birds can fly. Although, when faced with information about penguins as an exception to the rule, one would like to retract the previous conclusion. In monotonic FOL, however, retracting a conclusion is not possible. Thus, in classical logic, the PET becomes unsolvable due to the contradiction that may arise from $can_fly(X)$ and $\neg can_fly(X)$. The PET is unsolvable also in traditional *logic programming languages*, such as PROLOG [13]. In order to address the problem, many non-monotonic approaches have been developed, including Moore’s Autoepistemic logic, McCarthy’s Circumscription, Reiter’s Default Logic and in logic programming with negation by failure [12]. In autoepistemic logic, certain rules can be adjusted to include an exception: $\forall X (is_bird(X) \wedge \neg is_penguin(X) \rightarrow can_fly(X))$. However, the need to be explicit in including all exceptions makes this approach computationally expensive (considering that there are other birds that do not fly, e.g. ostriches). Circumscription and logic programming with negation by failure, on the other hand, find a solution to the problem by introducing the predicate *abnormal*, to indicate an exceptional case. The above rules would be re-written as $\forall X (is_bird(X) \wedge \neg is_abnormal(X) \rightarrow can_fly(X))$ along with a rule to state that penguins are abnormal birds. Other exceptions would then be added as needed without changing the original rule. Unfortunately, this approach does not adapt well to exceptions to the exceptions such as an abnormal penguins (a hypothetical super-penguin that is capable of flying).

At present, there is a tension between the above attempts to formalizing non-monotonicity and large-scale data-driven approaches based on neural networks and natural language that are efficient but lack any formalization. In this paper we seek to investigate approaches to solving PET and other simple examples that can be formalized but that work using the same tools as the large-scale network models. Work has been conducted to formalize NMR in neural networks starting with the Connectionist Inductive Learning and Logic Programming System (CILP) [14], later developed into a system for statistical relational learning. More recently, the Differentiable Inductive Logic Programming (∂ ILP) approach [15] was proposed, addressing cycles through negation. Probabilistic approaches have also been developed which can implement a form of non-monotonicity or at least avoid the problems of classical logic by assigning probabilities to beliefs expressed as Horn clauses, e.g. DeepProbLog [16]. In this paper, rather than mapping symbolic representations into neural networks and vice-versa, we are interested in the interplay between learning and reasoning as part of a curriculum. We focus on the Logic Tensor Network (LTN) [5] because it is a highly modular NeSy framework applicable in principle to any underlying neural network model and based on the canonical, highly expressive FOL language. Additionally, the LTN has shown promise for learning in continual mode [17].

The LTN relies on two main ideas, the *grounding* of predicates and logical axioms into vectors and *Real Logic* which maps the satisfiability of the logical axioms to a real number in the interval $\{0,1\}$ thus enabling viewing satisfiability as optimization. Given a knowledge base of FOL axioms \mathcal{K} , the LTN grounds every variable X to a vector representation $\mathcal{G}(X) = \langle x_1 \dots x_k \rangle \in R^{k-1}$, and every predicate P to a neural network $\mathcal{G}(p) \rightarrow [0, 1]$.² The application of *Real Logic* uses differentiable fuzzy logic to calculate the truth value of any LTN rule in the usual way. The satisfiability (*sat*), i.e. the aggregated truth value of the knowledge base, is then used in the loss function, with $Loss = 1 - sat$. For further details, we point the reader to [5].

3. Method

Continual Reasoning is proposed as a novel methodology, addressing reasoning tasks with a combination of NeSy models and a curriculum of training. In CL, a multi-task dataset is split along the different tasks, so that the model can be trained on each subset of data at each stage of the curriculum, with the aim to learn new tasks without forgetting old ones. In the context of NeSy models where tasks and knowledge are mostly represented at the symbolic level, we treat the aforementioned splitting of data as a division of the symbolic knowledge along a series of stages, which constitutes our curriculum of learning. In doing so, we rely on the neural

¹The value k is a hyperparameter of the framework, and is defined by the developer. For the experiments below, the values $\langle x_1 \dots x_k \rangle$ are initialized randomly and trained along with the predicate neural network.

²A note on terminology: the LTN framework treats FOL axioms in a slightly different way than logic programming. A grounding creates a direct connection with data, mapping a variable to a specific partition of the data. For this reason, we use the term *rules* instead of axioms when referring to the FOL knowledge base defined in LTN. The FOL axiom $\forall_X is_bird(X) \rightarrow can_fly(X)$ is defined in LTN as the rule $\forall_{Animals} is_bird(Animals) \Rightarrow can_fly(Animals)$, where *Animals* is the set of vector groundings for all animals in the data. This makes LTN a typed FOL language. If we wish to declare rules that only apply to a subset of *Animals*, we can do this in LTN using e.g. $\forall_{Norm_Birds} is_bird(Norm_Birds)$, where *Norm_Birds* consists only of the vector representations for birds, which is a subset of *Animals*. This excludes other subsets of animals, e.g. *Penguins* or *Cows*. For the definition of the PET used in LTN, see Appendix A.

networks of the NeSy models to learn new knowledge without forgetting previously learned knowledge, adjusting their beliefs about previously learned knowledge to allow for the new knowledge to be mapped to true without creating an inconsistency. Specifically, when using the LTN as our NeSy model, a knowledge base (KB) of FOL rules is separated into multiple stages for learning. For example, consider a KB consisting of facts $a(X)$, $b(X)$, $c(X)$, and rules $a(X) \Rightarrow d(X)$ and $b(x) \wedge c(X) \Rightarrow d(X)$. A split into three stages might be: (1) train on the facts; (2) train on $a(X) \Rightarrow d(X)$ and recall fact $a(X)$; (3) train on $b(x) \wedge c(X) \Rightarrow d(X)$. All facts and rules are assumed to be universally quantified. Our experiments will show, as one would expect, that the choice of curriculum, i.e the specific sequence in which the rules are learned and the facts are recalled, can affect the outcome. It becomes apparent that while in traditional machine learning all data is treated equally as being i.i.d. (although recent work around out-of-distribution (OOD) learning has started to question this assumption [18]), in reasoning tasks, especially NMR, the order in which knowledge is learned matters (in addition to the data split already identified as important in OOD learning).

Thus, we focus on two core requirements for the choice of curriculum. The first relies on the approach commonly applied in CL where data is split into separate tasks [6]. This can be applied in Continual Reasoning by treating each predicate as an individual task and training any rule aimed at learning about said predicate in a single stage of the curriculum. We call this *Task Separation*. In our previous example, we would split the KB into four stages: (1) learn $a(X)$; (2) $b(X)$; (3) $c(X)$; and (4) learn about $d(_)$, training on both rules. The second requirement takes inspiration from work conducted with knowledge graphs and lifelong learning projects such as NELL [19], in which we aim to “build up” from atomic knowledge (i.e. facts) and augment knowledge by abiding to new rules. In Continual Reasoning, we can accomplish this by giving priority to learning propositional rules, and rules that are directly tied to labelled data. Following this, we aim to use rules that extend the learned domain beyond what is available to more abstract concepts. This is known as *Knowledge Completion*. Using again our previous example, to satisfy both requirements we would split the KB into two stages: (1) train $a(X)$, $b(X)$ and $c(X)$; (2) learn $a(X) \Rightarrow d(X)$ and $b(x) \wedge c(X) \Rightarrow d(X)$.

To be able to do the above using neural networks, we must address the core issue found in CL, often referred to as *catastrophic forgetting*, i.e. when the process of gradient descent leads the neural network to forget previously learned data by conforming entirely to newly provided data. To address this problem, we apply a common CL technique of *rehearsal* [6]. Rehearsal is the process by which previously seen data is sampled and recalled in the current stage of learning. For Continual Reasoning, since our knowledge is represented in FOL, in each stage of learning, we recall a random set of previously learned knowledge, such as $a(X)$ earlier, to be learned along with the set of FOL rules.

For our analysis, we compare the task separation and knowledge completion curricula to a *Baseline*, where all knowledge is learned in a single stage, and a *Random* curriculum, where the KB split is randomly selected for each stage. To allow for effective comparison, all curricula, apart from the baseline, are composed of three stages. These comparisons are applied to the PET as a prototypical NMR task to show their benefits. In addition, to show the effectiveness of Continual Reasoning on other types of reasoning problems, we apply it to the Smokers and Friends task[5, 20, 9] and to Task 1 of the bAbI dataset [10] in what follows.

Table 1

Accuracy for each Curriculum (Curr) choice for the Penguin Exception Task (PET). Baseline: all rules are learned at once. Random: random split of rules along 3 stages. Task Separation (TS): divide rules according to task. Knowledge Completion (KC): divide rules to train facts before general rules. Best performing overall performance can be found in the task separation curriculum.

Curr	Rules	Stage 1	Stage 2	Stage 3
Baseline	is_bird(Norm_Birds)			97.1% \pm 0.11%
	is_bird(Penguins)	-	-	61.8% \pm 0.00%
	can_fly(Birds)			96.2% \pm 0.33%
	\neg can_fly(Penguins)			62.8% \pm 0.00%
Random	is_bird(Norm_Birds)	61.8% \pm 47.8%	88.2% \pm 33.0%	97.7% \pm 0.02%
	is_bird(Penguins)	54.5% \pm 45.9%	58.2% \pm 48.3%	67.1% \pm 21.7%
	can_fly(Birds)	28.5% \pm 43.9%	65.7% \pm 49.1%	90.5% \pm 7.06%
	\neg can_fly(Penguins)	71.2% \pm 43.7%	44.0% \pm 48.4%	79.7% \pm 16.8%
KC	is_bird(Norm_Birds)	99.9% \pm 0.01%	99.9% \pm 0.01%	99.9% \pm 0.01%
	is_bird(Penguins)	22.5% \pm 24.3%	98.9% \pm 2.22%	99.1% \pm 1.16%
	can_fly(Birds)	57.6% \pm 6.21%	99.1% \pm 1.94%	91.9% \pm 7.29%
	\neg can_fly(Penguins)	41.4% \pm 4.81%	2.64% \pm 5.74%	78.7% \pm 43.5%
TS	is_bird(Norm_Birds)	99.9% \pm 0.00%	99.9% \pm 0.00%	99.9% \pm 0.01%
	is_bird(Penguins)	99.8% \pm 0.02%	99.9% \pm 0.01%	99.5% \pm 0.32%
	can_fly(Birds)	53.9% \pm 5.53%	99.9% \pm 0.00%	84.7% \pm 2.21%
	\neg can_fly(Penguins)	53.1% \pm 5.25%	0.01% \pm 0.01%	99.7% \pm 0.25%

4. Results

Penguin Exception Task (PET) : For the PET, we examine the behaviour of the LTN model throughout the curriculum of training, paying particular attention to three distinct types of reasoning that are necessary for success. First, we have knowledge that can be learned through induction with *one-hop reasoning*, such as determining that all normal birds fly: $\forall_{Norm_Birds} can_fly(Norm_Birds)$, and that all penguins are birds: $\forall_{Penguins} is_bird(Penguins)$. Second, we have *two-hop reasoning* when determining that all penguins should be able to fly, $\forall_{Penguins} can_fly(Penguins)$, because they are birds. This is an instance of *jumping to a conclusion* in the absence of further information. Lastly, we contradict this conclusion with our final learning stage for which we expect to conclude non-monotonically that penguins in fact do not fly, $\forall_{Penguins} \neg can_fly(Penguins)$. We use these four FOL statements as queries in the analysis of our curricula of learning by measuring their LTN satisfiability over time (Table 1).

The results indicate that the task separation curriculum performs better than the other curricula, with the LTN able to correctly distinguish between all types of animals, as well as learn that normal birds can fly, while penguins, although still classified as birds, do not fly. The knowledge completion curriculum also performs to a high satisfiability for each of the queries.

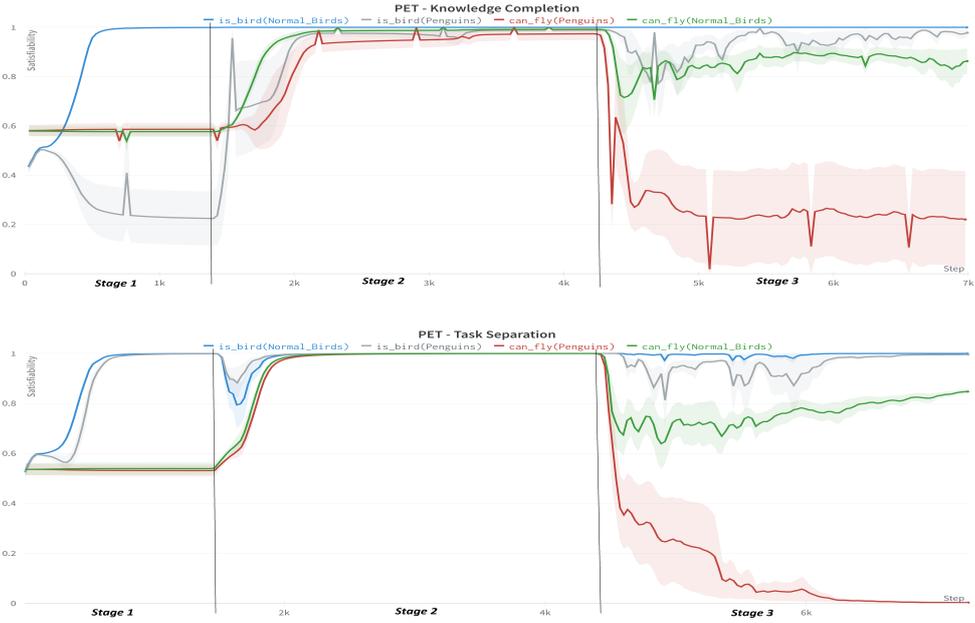


Figure 1: Satisfiability of four LTN queries: (1) Normal birds are birds (blue), (2) Penguins are birds (grey), (3) Normal birds can fly (green), (4) Penguins can fly (red), for the Knowledge Completion curriculum (top) and the Task Separation curriculum (bottom). Overall better performance is seen in the Task Separation curriculum.

However, in comparison with task separation, the knowledge completion curriculum is less robust, and in our experimentation led to one failure case, in which penguins were misclassified as normal birds, and therefore could fly.

When analyzing the queries throughout the training stages, we can identify changes that show that the LTN has the desired behaviour, including *jumping to conclusions* and *belief revision*. Specifically, in the second stage of both curricula, the LTN is trained to infer that penguins are birds, as well as that all birds can fly. Until told otherwise, the LTN jumps to the conclusion that penguins should be able to fly. In the third stage, however, the LTN is trained on the rule that penguins cannot fly. Given this knowledge, $can_fly(Penguins)$ and $can_fly(Normal_Birds)$ take an initial plunge (clearly shown in Figure 1). This, of course, makes sense, as the LTN does not yet have any reason to distinguish between penguins and normal birds, and thus once again jumps to the conclusion that since penguins cannot fly, then normal birds should not fly either. However, we see that the process of recall makes $can_fly(Normal_Birds)$ regain satisfiability, while the satisfiability of $can_fly(Penguins)$ decreases towards zero. It is interesting to note that in stage 3 the apparent contradiction does not lead to a convergence around an uninformative satisfiability of 0.5. With a random curriculum, we see more variance in the final results, which is to be expected given the random choice of rules, but overall, on average, this curriculum performs slightly better than the baseline. This shows that even *without* the benefit of curriculum design, the *method of Continual Reasoning*

Table 2

Accuracy comparison for the Smokers and Friends statistical relational reasoning task. Compares the lower and upper bounds of LNN [20] trained on P_2^5 , and log-probability weights of MLN [9] (provided in [20]), for selected rules, with baseline LTN and LTN with continual learning on task separation (TS) and knowledge completion (KC) curricula.

Rules	LNN- P_2^5	MLN	LTN-Baseline	LTN-TS Stage 3	LTN-KC Stage 3
$\neg F(x, x)$	[0.83,0.98]	0.26	0.998	0.996	0.995
$F(x, y) \Rightarrow F(y, x)$	[0.97,1.00]	-	0.796	0.826	0.954
$\exists_y F(x, y)$	[1.00,1.00]	6.88	0.730	0.748	0.718
$F(x, y) \wedge S(x) \Rightarrow S(y)$	[0.65,1.00]	3.53	0.716	0.615	0.497
$S(x) \Rightarrow C(x)$	[0.58,1.00]	-1.35	0.715	0.806	0.978
$F(x, y)$	-	-	0.865	0.821	0.876
$S(x)$	-	-	0.825	0.731	0.653
$C(x)$	-	-	0.919	0.969	0.999
$\neg S(x) \Rightarrow \neg C(x)$	-	-	0.917	0.910	0.991

leads to better results than attempting to learn the full knowledge base in a single stage. By further analysing the experiments in which the random curricula perform optimally, we see that task separation and knowledge completion curricula are not the only viable option for success (see Appendix A).

Smokers and Friends Task (S&F) : The S&F problem consists of a statistical relational reasoning task. We define the knowledge base in accordance with [5] and compare a baseline curriculum to curricula belonging to knowledge completion and task separation paradigms. The satisfiability of each rule throughout the stages show that a knowledge completion curriculum outperforms the baseline and task separation on identifying that smoking causes cancer (97.8% to 71.5% and 80.6%, respectively). Overall, the knowledge completion curriculum leads to the LTN reaching higher satisfiability in five of the nine FOL rules, in comparison with the baseline which beats the other curriculum in only three of the nine rules (see Appendix B for a table detailing the satisfiability of rules per stage of each curriculum).

In addition to comparison between curricula, we compare the outcome of Continual Reasoning with two other NeSy models that have been applied to S&F, the Logical Neural Network (LNN) and the Markov Logic Network (MLN). The LNN allows for a lower and upper bound truth value, which signifies the lowest possible and highest possible truth value for a given FOL axiom, such that the whole knowledge base holds true. The MLN derives axiom log-probability weights which signify the probability of the axiom’s mapping to true compared to the probability of it mapping to false. In Table 4, we see the results of these models per FOL rule used for training in our experiments. It is important to note that a precise comparison is not possible, as each model defines the set of FOL rules slightly differently in training. However, we see that the application of continual reasoning on LTNs for the S&F task performs comparably to other NeSy approaches.

bAbI - Task 1 : Task 1 of the bAbI dataset contains story lines of given facts and questions about those facts. For example, one instance will provide the sentences "Mary went to the office. Jack travelled to the garden." and ask "Where is Mary?". In order to address such a task with the proposed approach of Continual Reasoning using LTNs, we transform natural language sentences into FOL rules using GPT-3 [21]³. As the task already consists of stories told in stages, separated by questions, for curriculum design, we simply separate the FOL rules along the same stages in the dataset. The reasoning here can be said to be non-monotonic over time in that, later in the story, truth-values may change, e.g. Mary may no longer be in the office. Initial experimentation showed that by applying Continual Reasoning, a LTN model achieves 96.9% accuracy on the testing set of bAbI-Task 1, surpassing the 95% threshold for success. Further experimentation is ongoing.

5. Discussion, Conclusions and Future Work

We have introduced a novel methodology that integrates neurosymbolic AI and continual learning techniques in order to achieve non-monotonic reasoning. We call this Continual Reasoning, and we showed that by using Logic Tensor Networks [5] as our neural-symbolic framework, and training the knowledge base of First-Order Logic rules in a curriculum of multiple stages, we can improve on the traditional approach of learning all rules together. Additionally, we have analysed multiple types of curricula, proposing two general paradigms for curriculum design, and showed that while even random curriculum performs better on average than the baseline, a specific design choice can allow the model to appropriately jump to conclusions and revise its beliefs more effectively.

Experimentation conducted for this paper showed that Continual Reasoning also performs comparably on statistical relational reasoning tasks to a baseline curriculum, and other NeSy models. Continuation of this work could apply Continual Reasoning on larger datasets, such as the dataset used in RuleTaker [24], visual relational question-answering datasets, such as the CLEVR [25], and the remaining tasks in the bAbI dataset [10].

Furthermore, there still remain open questions concerning Continual Reasoning, such as how it might perform in extended non-monotonic reasoning tasks that occur when addressing lifelong learning. Rudimentary exploration of extending the PET to learn about a "super-penguin" which could fly, resulted in the LTN mostly failing to learn the exception to the exception. We believe, however, that utilising more advanced continual learning techniques, such as structural choices for neural network architecture, as well as more sophisticated recall methods like active learning, as suggested in [6, 17], would allow the Continual Reasoning methodology to succeed. This is to be investigated. Additionally, while LTNs proved to be a straightforward NeSy model to apply Continual Reasoning on, it should be possible to apply our methodology to other NeSy models, such as LNNs. Integration with a very recent software framework called PyReason [26] could provide an efficient way to do this.

³This approach is inspired by that used in [22], although FOL parsing of natural language is an evolving field of research which continues to face challenges [23]

References

- [1] A. d'Avila Garcez, L. C. Lamb, Neurosymbolic ai: The 3rd wave (2020). URL: <http://arxiv.org/abs/2012.05876>.
- [2] J. Zhang, B. Chen, L. Zhang, X. Ke, H. Ding, Neural, symbolic and neural-symbolic reasoning on knowledge graphs, arXiv:2010.05446 [cs] (2021). URL: <http://arxiv.org/abs/2010.05446>.
- [3] T. R. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kuehnberger, L. C. Lamb, D. Lowd, P. M. V. Lima, L. de Penning, G. Pinkas, H. Poon, G. Zaverucha, Neural-symbolic learning and reasoning: A survey and interpretation, arXiv:1711.03902 [cs] (2017). URL: <http://arxiv.org/abs/1711.03902>.
- [4] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, J. Wu, The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision, ICLR (2019).
- [5] S. Badreddine, A. d'Avila Garcez, L. Serafini, M. Sparanger, Logic tensor networks (2020).
- [6] M. Mundt, Y. W. Hong, I. Pliushch, V. Ramesh, A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning, arXiv:2009.01797 [cs, stat] (2020). URL: <http://arxiv.org/abs/2009.01797>.
- [7] L. Serafini, A. d'Avila Garcez, Learning and reasoning with logic tensor networks, Proc. Ai*AI (2016) 334–348.
- [8] I. Donadello, L. Serafini, A. d'Avila Garcez, Logic tensor networks for semantic image interpretation, IJCAI-17 (2017) 1596–1602.
- [9] M. Richardson, P. Domingos, Markov logic networks, Machine Learning 62 (2006) 107–136. doi:10.1007/s10994-006-5833-1.
- [10] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, T. Mikolov, Towards ai-complete question answering: A set of prerequisite toy tasks, 2015. arXiv:1502.05698.
- [11] A. d'Avila Garcez, L. Lamb, D. M. Gabbay, Neural-symbolic cognitive reasoning, in: Cognitive Technologies, 2009.
- [12] G. Antoniou, M. A. Williams, Nonmonotonic reasoning / Grigoris Antoniou ; with contributions by Mary-Anne Williams, MIT Press Cambridge, Mass, 1997.
- [13] M. A. Covington, R. Bagnara, R. A. O'Keefe, J. Wielemaker, S. Price, Coding guidelines for prolog, 2009. URL: <https://arxiv.org/abs/0911.2899>. doi:10.48550/ARXIV.0911.2899.
- [14] A. d'Avila Garcez, G. Zaverucha, The connectionist inductive learning and logic programming system, Appl. Intell. 11 (1999) 59–77. doi:10.1023/A:1008328630915.
- [15] R. Evans, E. Grefenstette, Learning explanatory rules from noisy data, 2017. URL: <https://arxiv.org/abs/1711.04574>. doi:10.48550/ARXIV.1711.04574.
- [16] R. Manhaeve, S. Dumančić, A. Kimmig, T. Demeester, L. De Raedt, Deepprolog: Neural probabilistic logic programming, 2018. URL: <https://arxiv.org/abs/1805.10872>. doi:10.48550/ARXIV.1805.10872.
- [17] B. Wagner, A. S. d'Avila Garcez, Neural-symbolic integration for fairness in ai, in: AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), volume 2846, 2021. URL: <https://openaccess.city.ac.uk/id/eprint/26151/>.
- [18] A. Słowik, L. Bottou, Algorithmic bias and data bias: Understanding the relation between

- distributionally robust optimization and data curation, 2021. [arXiv:2106.09467](https://arxiv.org/abs/2106.09467).
- [19] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, J. Welling, Never-ending learning, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), 2015.
 - [20] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, S. Iqbal, H. Karanam, S. Neelam, A. Likhyan, S. Srivastava, Logical neural networks (2020). URL: <http://arxiv.org/abs/2006.13155>.
 - [21] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020). URL: <https://arxiv.org/abs/2005.14165>. [arXiv:2005.14165](https://arxiv.org/abs/2005.14165).
 - [22] M. Nye, M. Tessler, J. Tenenbaum, B. M. Lake, Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning, in: Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 25192–25204. URL: <https://proceedings.neurips.cc/paper/2021/file/d3e2e8f631bd9336ed25b8162aef8782-Paper.pdf>.
 - [23] H. Singh, M. Aggarwal, B. Krishnamurthy, Exploring neural models for parsing natural language into first-order logic, CoRR (2020). URL: <https://arxiv.org/abs/2002.06544>.
 - [24] P. Clark, O. Tafjord, K. Richardson, Transformers as soft reasoners over language, 2020. URL: <https://arxiv.org/abs/2002.05867>. doi:10.48550/ARXIV.2002.05867.
 - [25] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, R. B. Girshick, CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning, CoRR abs/1612.06890 (2016). URL: <http://arxiv.org/abs/1612.06890>.
 - [26] D. Aditya, K. Mukherji, S. Balasubramanian, A. Chaudhary, P. Shakarian, Pyreason: Software for open world temporal logic, 2023. URL: <https://arxiv.org/abs/2302.13482>. doi:10.48550/ARXIV.2302.13482.

A. Penguin Exception Task - Extra Material

PET - LTN Rule and curriculum definition Let us assume the variables *Norm_Birds*, *Cows*, *Penguins*, and *Animals*, which represent groups of normal birds, cows, penguins, and the union of all groups of animals, respectively. Therefore, we define below a knowledgebase of FOL rules that reflect the PET as a prototypical non-monotonic reasoning task.

1. $\forall_{Norm_Birds} is_bird(Norm_Birds)$ (normal birds are birds)
2. $\forall_{Cows} \neg is_bird(Cows)$ (cows are not birds)
3. $\forall_{Animals} is_bird(Animals) \Rightarrow can_fly(Animals)$ (birds can fly)
4. $\forall_{Animals} \neg is_bird(Animals) \Rightarrow \neg can_fly(Animals)$ (non-birds cannot fly)
5. $\forall_{Penguins} is_penguin(Penguins)$ (penguins are penguins)
6. $\forall_{Non_Penguins} \neg is_penguin(Non_Penguins)$ (non-penguins are not penguins)
7. $\forall_{Animals} is_penguin(Animals) \Rightarrow is_bird(Animals)$ (penguins are birds)
8. $\forall_{Animals} is_penguin(Animals) \Rightarrow \neg can_fly(Animals)$ (penguins do not fly)

It is important to note that these rules are defined taking an open-world assumption, hence the need for declaring negations in rules 2 and 6. Additionally, we recognize that the same knowledge task could be defined using other forms of the same rules, to the same end. For example, rules 7 and 8 could be combined into one: $\forall_{Animals} is_penguin(Animals) \Rightarrow is_bird(Animals) \wedge \neg can_fly(Animals)$. However, for the purposes of this paper, we limit the rules to their simplest forms.

Table 3

Rules selection for each stage of the curriculum for the Penguin Exception Task (PET). In the baseline curriculum, all logical rules are learned in a single stage, and in the random curriculum, the full set of rules are randomly split into three stages. Task separation (TS) and Knowledge Completion (KC) are structured according to learning paradigms. See section 3 for a description of each paradigm.

Curriculum	Stage 1	Stage 2	Stage 3
PET-KC	normal birds are Birds cows are not Birds penguins are Penguins non-penguins aren't Penguins	birds can Fly non-birds cannot Fly penguins are Birds	penguins cannot Fly
PET-TS	normal birds are Birds cows are not Birds penguins are Penguins non-penguins aren't Penguins penguins are Birds	birds can Fly non-birds cannot Fly	penguins cannot Fly

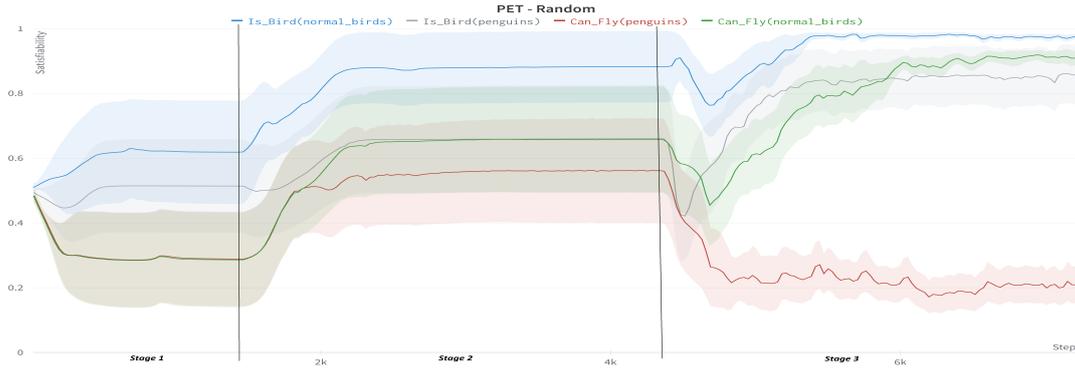


Figure 2: The figures present the satisfiability of four metrics, 1. normal birds are classified as birds (blue), 2. penguins are classified as birds (grey), 3. normal birds can fly (green), and 4. penguins can fly (red), for the Random curriculum in the PET.

Table 4

Random Curriculum choice that leads to well performing LTN in the PET. This curriculum results in $is_bird(\text{Normal_Birds})=99.9\%$, $is_bird(\text{Penguins})=99.8\%$, $can_fly(\text{Normal_Birds})=90.5\%$, and $not(can_fly(\text{Penguins}))=99.8\%$. One likely reason this curriculum succeeds is that the non-monotonic rule, that penguins do not fly, is learned in the final stage.

Stage	FOL rules
Stage 1	$\forall_{Animals} is_bird(Animals) \Rightarrow can_fly(Animals)$ (birds can fly) $\forall_{Animals} \neg is_bird(Animals) \Rightarrow \neg can_fly(Animals)$ (non-birds cannot fly) $\forall_{Non_Penguins} \neg is_penguin(Non_Penguins)$ (non-penguins are not penguins) $\forall_{Cows} \neg is_bird(Cows)$ (cows are not birds)
Stage 2	$\forall_{Penguins} is_penguin(Penguins)$ (penguins are penguins) $\forall_{Norm_Birds} is_bird(Norm_Birds)$ (normal birds are birds)
Stage 3	$\forall_{Animals} is_penguin(Animals) \Rightarrow \neg can_fly(Animals)$ (penguins do not fly) $\forall_{Animals} is_penguin(Animals) \Rightarrow is_bird(Animals)$ (penguins are birds)

B. Smokers and Friends Task - Extra Material

Table 5

Rules selection for each stage of the curriculum for the Smokers and Friends task (S&F). In the baseline curriculum, all logical rules are learning in a single stage. Task separation (TS) and Knowledge Completion (KC) are structured according to learning paradigms.

Curriculum	Stage 1	Stage 2	Stage 3
S&F-Baseline	-	-	identify known friendships identify known smokers identify known cancer friendship is antireflexive friendship is symmetric everyone has a friend friends of smokers smoke smokers have cancer non-smokers don't have cancer
S&F-KC	identify known friendships identify known smokers identify known cancer	friendship is antireflexive friendship is symmetric everyone has a friend friends of smokers smoke	smokers have cancer non-smokers don't have cancer
S&F-TS	identify known friendships friendship is antireflexive friendship is symmetric everyone has a friend	identify known smokers friends of smokers smoke	identify known cancer smokers have cancer non-smokers don't have cancer

Table 6

Accuracy for each Curriculum Choice for the Smokers and Friends (S&F) task. Baseline: all rules are learned at once. Random: random split of rules along 3 stages. Task Separation (TS): divide rules according to "task". Knowledge Completion (KC): divide rules to train facts before general rules. Asterisks signify that the associated FOL rule was trained in the particular stage of the curriculum. Best performing overall performance can be found in the knowledge completion curriculum.

Rules	Baseline	Task Separation			Knowledge Completion		
		Stage 1	Stage 2	Stage 3	Stage 1	Stage 2	Stage 3
$F(x, y)$	0.865	0.836*	0.795	0.821	0.872*	0.831	0.876
$S(x)$	0.825	0.489	0.830*	0.731	0.999*	0.811	0.653
$C(x)$	0.919	0.487	0.487	0.969*	0.999*	0.999	0.999
$\neg F(x, x)$	0.998	0.998*	0.997	0.996	0.116	0.989*	0.995
$F(x, y) \Rightarrow F(y, x)$	0.796	0.852*	0.784	0.826	0.200	0.778*	0.954
$\exists_y F(x, y)$	0.730	0.764*	0.742	0.748	0.753	0.715*	0.718
$F(x, y) \wedge S(x) \Rightarrow S(y)$	0.716	0.792	0.656*	0.615	0.298	0.742*	0.497
$S(x) \Rightarrow C(x)$	0.715	0.741	0.615	0.806*	0.267	0.366	0.978*
$\neg S(x) \Rightarrow \neg C(x)$	0.917	0.755	0.635	0.910*	0.463	0.940	0.991*
SAT of KB	0.813	0.658	0.683	0.780	0.511	0.752	0.732