F$\mathcal{ALC}$ON: Faithful Neural Semantic Entailment over $\mathcal{ALC}$ Ontologies

No Author Given

No Institute Given

1 Supplementary Materials

1.1 Reproducibility

We provide the statistics of the datasets in Table 1 and we list the tuned hyper-parameter settings in Table 2. The Family Ontology we used is defined as Eqn. 1. The detailed computation steps are shown in Algorithm 1.

Table 1. Statistics of datasets.

	Family	**Pizza**	**HPO**	**Yeast**
Individual Symbols	0	5	2504	5586
Concept Symbols	10	99	30893	45003
Relation Symbols	2	3	186	11
Abox C(a) Train	0	5	93270	56934
Abox R(a, b) Train	0	0	9554	171740
Abox R(a, b) Test	N/A	N/A	2400	21560
Tbox Train	25	676	64416	120027
Tbox Test	N/A	115	2000	N/A

$$\begin{aligned}
&\text{Male} \sqsubseteq \text{Person}, \text{Female} \sqsubseteq \text{Person}, \text{Male} \sqcap \text{Female} \sqsubseteq \bot,\\
&\text{Parent} \sqsubseteq \text{Person}, \text{Child} \sqsubseteq \text{Person}, \text{Parent} \sqcap \text{Child} \sqsubseteq \bot,\\
&\text{Father} \sqsubseteq \text{Male}, \text{Boy} \sqsubseteq \text{Male}, \text{Father} \sqcap \text{Boy} \sqsubseteq \bot,\\
&\text{Mother} \sqsubseteq \text{Female}, \text{Girl} \sqsubseteq \text{Female}, \text{Mother} \sqcap \text{Girl} \sqsubseteq \bot,\\
&\text{Father} \sqsubseteq \text{Parent}, \text{Mother} \sqsubseteq \text{Parent}, \text{Father} \sqcap \text{Mother} \sqsubseteq \bot,\\
&\text{Boy} \sqsubseteq \text{Child}, \text{Girl} \sqsubseteq \text{Child}, \text{Boy} \sqcap \text{Girl} \sqsubseteq \bot,\\
&\text{Female} \sqcap \text{Parent} \sqsubseteq \text{Mother}, \text{Male} \sqcap \text{Parent} \sqsubseteq \text{Father},\\
&\text{Female} \sqcap \text{Child} \sqsubseteq \text{Girl}, \text{Male} \sqcap \text{Child} \sqsubseteq \text{Boy},\\
&\exists \text{hasChild}.\text{Person} \sqsubseteq \text{Parent}, \exists \text{hasParent}.\text{Person} \sqsubseteq \text{Child},\\
&\text{Grandma} \sqsubseteq \text{Mother}
\end{aligned} \tag{1}$$

Note that we further add ABox axioms and create one named individual for each concept name, e.g., creating the individual *A child* for concept name *Child*.

Table 2. Tuned hyperparameter settings.

	Family	**Pizza**	**HPO**	**Yeast**
Number of models k	100	1-20	1	1
Contradictory axioms N_{inc}	0	0-1000	N/A	N/A
Learning rate	$1e^{-2}$	$5e^{-3}$	$1e^{-4}$	$1e^{-4}$
Embedding dimension	50	50	128	32
Negatives	N/A	N/A	8	8
Batchsize R(a, b)	N/A	N/A	64	64
Batchsize C(a)	N/A	N/A	64	64
Batchsize TBox	25	256	64	64
Sampled individuals	4	4	4	4
Created ABox axioms	10	99	1000	4000
t-norm	Product	Product	Product	Product

1.2 Additional Experiments

Table 3. Reasoning performance (AUC) with 10 models and improvement of the multi-model semantic entailment described in Section 4.2 across increasing inconsistency on Pizza, where **Avg** denotes the average results (AUC) of independent single models and **Multi** denotes multi-model approximate entailment. N_{inc} denotes the number of explicitly added contradictions.

N_{inc}	**0**	**1**	**10**	**50**
Avg	0.8418	0.8375	0.8500	0.7884
Multi	0.8543	0.8564	0.8706	0.8032
Impr.(%)	**3.66%**	**5.04%**	**5.89%**	**5.13%**

As shown in Table 3, with the introduction of inconsistency, the improvement of **Multi** over **AVG** becomes more significant; as contradictory statements tend to create degenerate models (with empty concepts and relations), aggregating over multiple models generally perform better under inconsistency since some models still tend to preserve the memberships required for entailment.

We also evaluate the effectiveness of approximate entailment under different number of models. As the results show in Figure 1, we observe that the approximate entailment performance improves with a larger number of models and will eventually converge. Although the performance will fluctuate with fewer models, multi-model approximate entailment can consistently improve over the single-model case, i.e., Avg in Figure 1. Such results demonstrate the effectiveness of reasoning with multiple models as well as the rationale for setting the number of models as a hyperparameter in practice.

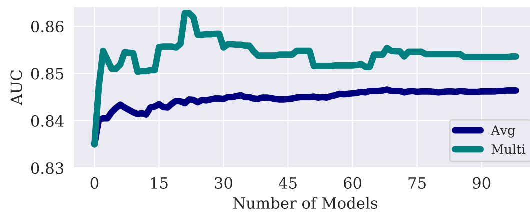

Fig. 1. Experimental results of multi-model reasoning on Pizza. **Avg** denotes the average results (AUC) of independent single models. **Multi** denotes semantic entailment over multiple models described in Section 5.2 of the paper.

Algorithm 1 Generating $C^{\mathcal{I}}$ for a Concept Description C.

Function: Calculate $m(\cdot, C^{\mathcal{I}})$

Require: Embedding function f_e;
Multilayer Perceptron MLP;
Activation function σ;
Sampling size k;
Fuzzy operators θ, κ, ν;
Individuals $\mathbf{I} = \mathbf{I}_n \cup \mathbf{I}_{\mathbb{R}^n}$.

Sample X with $|X| = k$ from $\mathbf{I}$
Compute $m(X, C^{\mathcal{I}}) := \{m(x, C^{\mathcal{I}}) | x \in X\}$:
if C is a concept name **then**
  $m(X, C^{\mathcal{I}}) = \sigma(MLP(f_e(C), f_e(X)))$
else if $C = C_1 \sqcap C_2$ **then**
  $m(X, (C_1 \sqcap C_2)^{\mathcal{I}}) = \theta(m(X, C_1^{\mathcal{I}}), m(x, C_2^{\mathcal{I}}))$
else if $C = C_1 \sqcup C_2$ **then**
  $m(X, (C_1 \sqcup C_2)^{\mathcal{I}}) = \kappa(m(X, C_1^{\mathcal{I}}), m(X, C_2^{\mathcal{I}}))$
else if $C = \neg D$ **then**
  $m(X, (\neg D)^{\mathcal{I}}) = \nu(m(X, D^{\mathcal{I}}))$
else if $C = \exists R.D$ **then**
  Sample Y with $|Y| = k$ from $\mathbf{I}$
  $m(X, (\exists R.D)^{\mathcal{I}}) = \max_{y \in Y} \theta(m(y, D^{\mathcal{I}}), m((X, y), R^{\mathcal{I}}))$
  with $m((x, y), R^{\mathcal{I}}) = \sigma(MLP(f_e(x) + f_e(R), f_e(y)))$
else if $C = \forall R.D$ **then**
  Sample Y with $|Y| = k$ from $\mathbf{I}$
  $m(X, (\forall R.D)^{\mathcal{I}}) = \min_{y \in Y} \kappa(\nu(m(y, D^{\mathcal{I}})), m((X, y), R^{\mathcal{I}}))$
  with $m((x, y), R^{\mathcal{I}}) = \sigma(MLP(f_e(x) + f_e(R), f_e(y)))$
end if
return $m(X, C^{\mathcal{I}})$
