

Neuro-symbolic Natural Logic with Introspective Revision for Natural Language Inference

Yufei Feng*

Ingenuity Labs Research Institute &
ECE, Queen’s University, Canada
feng.yufei@queensu.ca

Xiaoyu Yang*

Ingenuity Labs Research Institute &
ECE, Queen’s University, Canada
xiaoyu.yang@queensu.ca

Xiaodan Zhu

Ingenuity Labs Research Institute &
ECE, Queen’s University, Canada
xiaodan.zhu@queensu.ca

Michael Greenspan

Ingenuity Labs Research Institute &
ECE, Queen’s University, Canada
michael.greenspan@queensu.ca

Abstract

We introduce a neuro-symbolic natural logic framework based on reinforcement learning with introspective revision. The model samples and rewards specific reasoning paths through policy gradient, in which the introspective revision algorithm modifies intermediate symbolic reasoning steps to discover reward-earning operations as well as leverages external knowledge to alleviate spurious reasoning and training inefficiency. The framework is supported by properly designed local relation models to avoid input entangling, which helps ensure the interpretability of the proof paths. The proposed model has built-in interpretability and shows superior capability in monotonicity inference, systematic generalization, and interpretability, compared with previous models on the existing datasets.

1 Introduction

In the past decade, deep neural networks have achieved impressive performance on modeling natural language inference (NLI) (Dagan et al., 2005; MacCartney, 2009; Bowman et al., 2015; Chen et al., 2017a,b), which aims to determine the entailment relations between a *premise* sentence and its corresponding *hypothesis*. Progress in NLI has greatly benefited from the models’ capabilities at approximating complex underlying functions, discovering and utilizing rich (true and/or spurious) patterns, and exhibiting robustness to noise and ambiguity. However, the black-box models inherently lack interpretability, and still fail to capture many aspects of human reasoning, including monotonicity inference (Yanaka et al., 2019a,b, 2020), systematic

compositionality and generalization (Fodor and Pylyshyn, 1988; Aydede, 1997; Yanaka et al., 2020), and negation (Geiger et al., 2020), among others.

A recent research trend has attempted to advance the long-standing problem of bringing together the complementary strengths of neural networks and symbolic models (Garcez et al., 2015; Yang et al., 2017; Rocktäschel and Riedel, 2017; Evans and Grefenstette, 2018; Weber et al., 2019; De Raedt et al., 2019; Mao et al., 2018). Specifically for natural language, *natural logic* has long been studied to model reasoning in human language (Lakoff, 1970; van Benthem, 1988; Valencia, 1991; Van Benthem, 1995; Nairn et al., 2006; MacCartney and Manning, 2009; MacCartney and Manning, 2009; Icard, 2012; Angeli and Manning, 2014). However, the work of investigating the joint advantage of neural networks and natural logic is sparse (Feng et al., 2020) (see Section 2 for more details) and understudied.

In this paper, we present a neuro-symbolic framework that integrates natural logic with neural networks for natural language inference. At the local level, we explore appropriate transformer networks to model the *local* relations between the constituents of a premise and hypothesis, in order to prevent attention from fully entangling the input, which otherwise can seriously impair the interpretability of proof paths built on local relations. We then construct natural logic programs and use reinforcement learning to reward the aggregation of the local relations. When reinforcement learning passes the final reward signals (NLI labels) through the neural natural logic composition network, it faces the challenges of

*Equal contribution.

excessive spurious programs (incorrect programs that lead to correct final NLI labels) as well as training inefficiency; the former is particularly harmful to interpretability. Our framework leverages the proposed Introspective Revision method to discover better reward-earning operations and leverage external knowledge to reduce spurious proofs.

We conducted experiments on six datasets: SNLI (Bowman et al., 2015), HELP (Yanaka et al., 2019b), MED (Yanaka et al., 2019a), MoNLI (Geiger et al., 2020), NatLog-2hop (Feng et al., 2020), and a compositional generalization dataset (Yanaka et al., 2020). The results show the model’s superior capability in monotonicity inferences, systematic generalization, and interpretability, compared with previous models on these existing datasets, while the model remains a competitive performance on the generic SNLI test set.

2 Related Work

Natural Logic: Rather than performing deduction over an abstract logical form, natural logic (Lakoff, 1970; van Benthem, 1988; Valencia, 1991; Van Benthem, 1995; Nairn et al., 2006; MacCartney, 2009; MacCartney and Manning, 2009; Icard, 2012; Angeli and Manning, 2014) models logical inferences in natural language by operating directly on the structure of language. Natural logic allows for a wide range of intuitive inferences in a conceptually clean way (MacCartney, 2009; Angeli and Manning, 2014) and hence provides a good framework for developing explainable neural natural language inference models. Specifically, our work is motivated by the natural logic variant proposed by MacCartney and Manning (2009), for which we will provide more background in Section 3.

Natural Language Inference: Natural language inference (NLI) (Dagan et al., 2005; MacCartney, 2009; Bowman et al., 2015) aims to identify the entailment relations between the *premise-hypothesis* sentence pairs. Benefiting from pre-training on large-scale unlabeled corpora and then fine-tuning on large crowd-sourced datasets like SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018), the pre-trained language models (Devlin et al., 2019; Radford et al., 2019, 2018) have achieved the state-of-the-art performance. However, recent

work revealed several drawbacks of the current deep NLI systems. The research in Gururangan et al. (2018) and Poliak et al. (2018) has shown that deep NLI models learn to utilize dataset biases and label-relevant artifacts for prediction. Yanaka et al. (2019a,b) and Geiger et al. (2020) showed that a dominating proportion of samples in SNLI and MultiNLI are in upward monotone, and models trained on these datasets have limited ability to generalize to downward monotone. More recently, systematically generated datasets have been proposed to evaluate the current models’ ability on compositional generalization and showed that pretrained transformers generalize poorly to unseen combinations of the semantic fragments (Geiger et al., 2019; Richardson et al., 2020; Yanaka et al., 2020; Goodwin et al., 2020).

Neural Network with Logic Components for NLI: Recent work (Kalouli et al., 2020; Hu et al., 2020; Chen et al., 2021; Feng et al., 2020) has started to combine neural networks with logic-based components. The work most related to ours is Feng et al. (2020), which adapts ESIM (Chen et al., 2017b) to predict relations between tokens in a premise and hypothesis, and composes them to predict final inferential labels. Rather than optimizing the likelihood of specific reasoning paths, the model maximizes the sum of the likelihood of all possible paths (i.e., marginal likelihood) that reach the correct final NLI labels. As a result, the model potentially encourages a large set of *spurious* reasoning paths and has to rely on external prior and strong constraints to predict meaningful intermediate local relations.

This paper, instead, proposes a reinforcement learning with introspective revision framework to sample and reward specific reasoning paths through the policy gradient method. The introspective revision leverages external commonsense knowledge to tackle spurious proof paths and training inefficiency, key issues in developing interpretable neuro-symbolic models. To support that, local relation components need to be carefully designed. We will demonstrate that the proposed model substantially outperforms that proposed in Feng et al. (2020) on five datasets.

Policy Gradient: Policy gradient algorithms like REINFORCE (Williams, 1992) have been used in neuro-symbolic models to connect neural

Relation	Relation Name	Example
$x \equiv y$	equivalence	<i>mom</i> \equiv <i>mother</i>
$x \sqsubset y$	forward entailment	<i>cat</i> \sqsubset <i>animal</i>
$x \supset y$	reverse entailment	<i>animal</i> \supset <i>cat</i>
$x \wedge y$	negation	<i>human</i> \wedge <i>nonhuman</i>
$x \mid y$	alternation	<i>cat</i> \mid <i>dog</i>
$x \smile y$	cover	<i>animal</i> \smile <i>nonhuman</i>
$x \# y$	independence	<i>happy</i> $\#$ <i>student</i>

Table 1: A set \mathfrak{B} of seven natural logic relations proposed by MacCartney and Manning (2009).

representation learning and symbolic reasoning (Andreas et al., 2017; Liang et al., 2017; Mascharka et al., 2018; Yi et al., 2018; Mao et al., 2018). The original REINFORCE algorithm suffers from sparse rewards and high variance in the gradient. To overcome these issues, the research presented in Popov et al. (2017), Goyal et al. (2019), and Trott et al. (2019) proposes reward shaping, which leverages domain-specific knowledge to carefully design the reward functions. Instead of learning only from the desired outcomes, some approaches also learn from failed attempts. Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) and Scheduled Auxiliary Control (SAC-X) (Riedmiller et al., 2018) can replay the failed episodes and provide the agent with auxiliary learning goals to enable sample-efficient learning. Li et al. (2020) propose a back-search algorithm, which diagnoses the failed reasoning processes and corrects potential errors to facilitate model training. Based on Li et al. (2020), we propose the introspective revision method, which leverages external knowledge to effectively discover reward-earning reasoning programs and to alleviate spurious reasoning.

3 Background

Our model’s backbone logic framework is based on the MacCartney and Manning (2009) variant of the natural logic formalism. The inference system operates by mutating spans of text in a premise to obtain the corresponding hypothesis sentence, and generates proofs based on the natural logic relations of the mutations. To extend the entailment relations to consider semantic exclusion, MacCartney and Manning (2009) introduced seven set-theoretic relations \mathfrak{B} for modeling entailment relations between two spans of texts (see Table 1 for some examples).

Assuming the availability of the alignment between a premise and hypothesis, the system first infers the relations between aligned pairs of words or phrases. Consider the top-left example in Figure 1: the relation between “*the child*” and “*the kid*” is *equivalence* (\equiv), same as the relation between “*does not love*” and “*doesn’t like*”, while “*sports*” reversely entails (\supset) “*table-tennis*”.

The next step is monotonicity inference. Monotonicity is a pervasive feature of natural language that explains the impact of semantic composition on entailment relations (Van Benthem, 1986; Valencia, 1991; Icard and Moss, 2014). Similar to the monotone functions in calculus, upward monotone keeps the entailment relation when the argument “increases” (e.g., *cat* \sqsubset *animal*). Downward monotone keeps the entailment relation when the argument “decreases” (e.g., in *all animals* \sqsubset *all cats*). The system performs monotonicity inference through a projection function $\rho: \mathfrak{B} \rightarrow \mathfrak{B}$, which is determined by the context and projection rules. Table 2 shows some examples. Consider the last row in the table—it shows how the project function ρ works in the negated context following the negation word *not*. Specifically, this row shows seven relations that $\rho(r)$ will output, given the corresponding input relations r . For example, if the input relation is *forward entailment* (\sqsubset), the function ρ projects it to *reverse entailment* (\supset); that is, $\rho(\sqsubset) = \supset$. As a result, in the example in Figure 1, the *reverse entailment* relation (\supset) between “*sports*” and “*table-tennis*” will be projected to *forward entailment* (\sqsubset) in the negated context.

Built on that, the system aggregates/composes the projected local relations to obtain the inferential relation between a premise and hypothesis. Specifically, Table 3 shows the composition function when a relation (in a row) is composed with another (in a column). In practice, multiple compositions as such are performed in sequential order or from leaves to root along a constituency parse tree. MacCartney (2009) shows that different orders of compositions yield consistent results except in some rare artificial cases. Therefore, many studies, including ours here, perform a sequential (left-to-right) composition. In the example in Figure 1, composing two *equivalence* (\equiv) with *forward entailment* (\sqsubset) yields *forward entailment* (\sqsubset), resulting in a prediction that the premise entails the hypothesis.

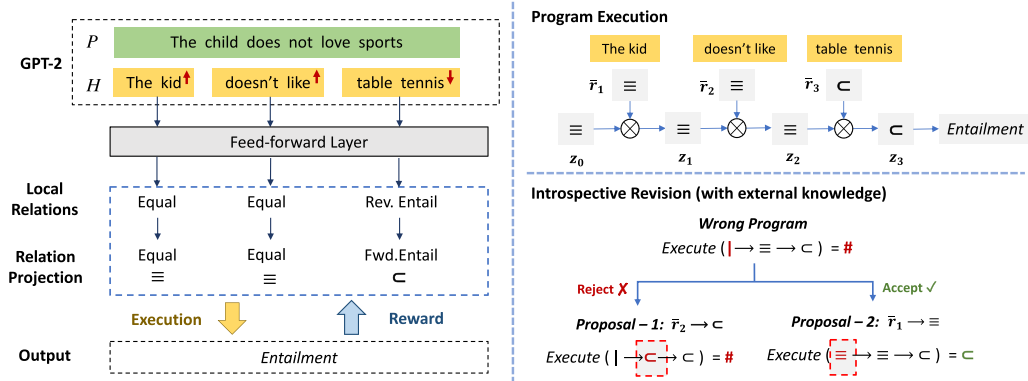


Figure 1: An overview of the proposed neuro-symbolic natural logic framework.

Quantifier & Connective	Proj.	Input Relation r					
		\equiv	\sqsubset	\sqsupset	\wedge	\vee	$\#$
<i>all</i>	$\rho^{arg1}(r)$	\equiv	\sqsubset	\sqsupset	\wedge	\vee	$\#$
	$\rho^{arg2}(r)$	\equiv	\sqsubset	\sqsupset	\wedge	\vee	$\#$
<i>some</i>	$\rho^{arg1}(r)$	\equiv	\sqsubset	\sqsupset	\vee	\wedge	$\#$
	$\rho^{arg2}(r)$	\equiv	\sqsubset	\sqsupset	\vee	\wedge	$\#$
<i>not</i>	$\rho(r)$	\equiv	\sqsubset	\sqsupset	\wedge	\vee	$\#$

Table 2: The projection function ρ maps input relations to output relations under different contexts (here, different surrounding quantifiers).

\bowtie	\equiv	\sqsubset	\sqsupset	\wedge	\vee	$\#$
\equiv	\equiv	\sqsubset	\sqsupset	\wedge	\vee	$\#$
\sqsubset	\sqsubset	\sqsubset	$\#$	\vee	\wedge	$\#$
\sqsupset	\sqsupset	$\#$	\sqsupset	\vee	\wedge	$\#$
\wedge	\wedge	\vee	\vee	\equiv	\sqsupset	\sqsubset
\vee	\vee	$\#$	\vee	\sqsubset	$\#$	\sqsubset
$\#$	$\#$	$\#$	$\#$	\sqsupset	\sqsupset	$\#$
$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$

Table 3: Results (Icard, 2012) of composing one relation (row) with another relation (column).

4 Method

This section introduces our neural natural logic framework based on the proposed Reinforcement Learning with Introspective Revision approach. We start with local relation modeling, in which caution needs to be taken to avoid the input entangling problem, which can seriously harm the model’s interpretability. By viewing the local relation distribution as the stochastic policy, our model then samples and rewards specific reasoning paths through policy gradient, in which the Introspective Revision model can modify intermediate symbolic reasoning steps to discover better reward-earning operations and leverages external

knowledge to alleviate spurious reasoning and training inefficiency.

4.1 Local Relation Modeling

We use phrases/chunks instead of words as the basic reasoning units. The primary motivation for chunking is to shorten the reasoning paths and hence reduce the number of possible paths, both of which make the reasoning process more efficient. Motivated by Ouyang and McKeown (2019), we segment the premise P and the hypothesis H into several phrases/chunks. Specifically, we first extract noun phrases with spaCy (Honnibal et al., 2020) and then group the continuous spans of words between two noun phrases as chunks. As shown in Figure 1, by identifying the noun phrases “the kid” and “table tennis”, the hypothesis sentence H is segmented into three chunks. We denote the number of chunks in the hypothesis as m , and the t -th hypothesis chunk (and its vectorized representation) as s_t . Similarly, the t' -th premise phrase is denoted as $\tilde{s}_{t'}$.

As the first step of the neuro-symbolic natural logic, we use a neural network to model the local natural logic relation between each hypothesis phrase s_t and its associated premise constituents. However, accurately finding the hard alignment between s_t and the corresponding phrase $\tilde{s}_{t'}$ in the premise is a hard problem (MacCartney et al., 2008). Current state-of-the-art NLI systems, like BERT (Devlin et al., 2019), use bi-directional soft attention to model the cross-sentence relationship, however, we observe that it tends to fully *entangle* the input (DeYoung et al., 2020). Consider the top-left example in Figure 1. If we use BERT to encode the input sentences, then the bi-directional attention model can infer the final NLI label solely

based on the last-layer hidden states of the first hypothesis phrase “*the kid*” because the contextualized representation of this phrase *entangles* the information of the whole input through attention. Consequently, the hidden states of the phrase contain global information, thus not being suitable for modeling the local relations.

To alleviate the undesired entangling, we model local relations with uni-directional attention (such as GPT-2). On the one hand, the uni-directional attention prevents entangling future inputs. For example, in Figure 1, the phrase “*table tennis*” will not affect the relation prediction anchored on “*the kid*”. On the other hand, although the last hypothesis phrase attends to all previous inputs, without knowing whether the current phrase is the ending one (the future inputs are not available), the model cannot skip predicting the natural logic relation at the current phrase s_t and postpone all the required reasoning to the last phrase. Specifically, suppose a model always predicts *equivalence* (\equiv) at each step t and postpones its final decision to the last hypothesis phrase. Without knowing that “*table tennis*” is the ending phrase, the model can predict *equivalence* (\equiv) for “*table tennis*” and wait to make a better decision upon seeing the next input phrase, which actually does not exist. Failing to make timely local predictions that lead to the correct label before running out of the hypothesis phrases, the model will receive a negative reward in the end. In this way, the model is encouraged to be more careful in predicting the local relation for each hypothesis phrase. We also develop a model that achieves local relations by masking both the past and future hypothesis chunks. Compared with such a model, we will show later (Table 6) that the uni-directional attention model performs better, partly because it preserves the structure of the pretrained GPT-2 model.

Specifically, we propose to model the local relation between s_t and the premise P , which can be efficiently achieved by the pretrained GPT-2 model (Radford et al., 2019). We concatenate a premise and hypothesis as the input and separate them with a special token $\langle sep \rangle$. The contextualized encoding h_τ for the τ -th hypothesis token is extracted from the GPT-2 last-layer hidden states at the corresponding location:

$$h_\tau = GPT-2(P, H_{1:\tau}) \quad (1)$$

For the t -th phrase in the hypothesis $s_t = H_{\tau_1:\tau_2}$, which starts from position τ_1 and ends at position

τ_2 , we concatenate features of the starting token h_{τ_1} and the ending token h_{τ_2} as the vectorized phrase representation:

$$s_t = Concat(h_{\tau_1}, h_{\tau_2}) \quad (2)$$

We use a feed-forward network f with ReLU activation to model the local natural logic relations between the hypothesis phrase s_t and its potential counterpart in the premise. The feed-forward network outputs 7 logits that correspond to the seven natural logic relations listed in Table 1. The logits are converted with *softmax* to obtain the local relation distribution:

$$p_t = softmax(f(s_t)), \quad (3)$$

Intuitively, the model learns to align each hypothesis phrase s_t with the corresponding premise constituents through attention, and combines information from both sources to model local relations. In practice, the local relation distribution is defined over five relations: we merge relation *negation* (\wedge) and *alternation* (\mid) because they have similar behaviors in Table 3, and we suppress *cover* (\neg), because it is rare in the current NLI datasets. Hence we only need to model five natural logic relation types, following Feng et al. (2020).

4.2 Natural Logic Program

We propose to use reinforcement learning to develop neural natural logic, which views the local relation distribution p_t as the stochastic policy. At each time step t , the model samples a relation $r_t \in \mathfrak{B}$ according to the policy, and we treat the sequence of sampled relations $\{r_t\}_{t=1}^m$ as a symbolic program, which executes to produce the final inferential relation between a premise and hypothesis. To the best of our knowledge, this is the first model that integrates reinforcement learning with natural logic.

Built on the natural logic formalism of MacCartney and Manning (2009), a projection function ρ (Eq. 5) maps r_t to a new relation \bar{r}_t . In our model, the projection function ρ is determined by the projectivity feature from the StanfordCoreNLP *natlog* parser.¹ For each input token, the projectivity feature specifies the projected relation \bar{r}_t for each input relation r_t . In this

¹<https://stanfordnlp.github.io/CoreNLP/natlog.html>.

work, we extend the token-level projectivity to handle phrases: For a phrase with multiple tokens, ρ is determined by the projectivity of the first token in the phrase. In Figure 1, the projectivity of the phrase “*table tennis*” is determined by the first token “*table*”, and ρ projects the predicted *reverse entailment* (\sqsupset) relation to *forward entailment* (\sqsubset).

$$r_t = \text{sampling}(\mathbf{p}_t), \quad (4)$$

$$\bar{r}_t = \rho(r_t) \quad (5)$$

The program then composes the projected relations $\{\bar{r}_t\}_{t=1}^m$ to derive the final relation prediction, as shown in top-right part in Figure 1. Specifically, at time step $t=0$, the executor starts with the default state $z_0 = \text{equivalence}$ (\equiv). For each hypothesis phrase $s_t, t > 0$, the program performs one step update to compose the previous state z_{t-1} with the projected relation \bar{r}_t :

$$z_t = \text{step}(z_{t-1}, \bar{r}_t) \quad (6)$$

The final prediction is yielded from the last state z_m of program execution. Following Angeli and Manning (2014), we group *equivalence* (\equiv) and *forward entailment* (\sqsubset) as **entailment**; *negation* (\wedge) and *alternation* (\mid) as **contradiction**; and *reverse entailment* (\sqsupset), *cover* (\cup), and *independence* ($\#$) as **neutral**.

Rewards and Optimization: During training, we reward the model when the program executes to the correct answer. Given a sequence of local relations $\mathbf{r} = \{r_t\}_{t=1}^m$, at each step t the model receives a reward R_t as follows:

$$R_t = \begin{cases} \mu, & \text{if } \text{Execute}(\mathbf{r}) = y \\ -\gamma^{m-t}\mu, & \text{if } \text{Execute}(\mathbf{r}) \neq y, \end{cases} \quad (7)$$

where μ is the constant reward unit, $\gamma \in (0, 1]$ is the discount factor, and y is the ground-truth label. In addition to Eq. 7, different rewards are applied under two exceptional cases: (1) if at step t there is no chance for the program to get a positive reward, then the execution is terminated and the model receives an immediate reward $R_t = -\mu$; (2) when the true label is *entailment*, the model receives no positive reward if the last state z_m is *equivalence* (\equiv). In this way, we encourage the model to select at least one forward entailment (\sqsubset) relation during prediction, instead of aggregating a sequence of *equivalence* (\equiv) for all entailment

cases. In the current NLI datasets, it is less likely that the premise and hypothesis sentences are semantically equivalent to each other.

We apply the REINFORCE (Williams, 1992) algorithm to optimize the model parameters. During training, the local relations r_t are sampled from the predicted distribution, and we minimize the policy gradient objective:

$$J = - \sum_{t=1}^m \log \mathbf{p}_t[r_t] \cdot R_t \quad (8)$$

where $\mathbf{p}_t[r_t]$ is the probability that corresponds to the sampled relation r_t . During the test, the model picks the relation with largest probability.

Relation Augmentation: It can be hard to learn the *reverse entailment* (\sqsupset) relation from the existing NLI datasets because the relation of a pair of premise and hypothesis is labeled as *neutral*, if H entails P and P does not entails H . To help the model distinguish *reverse entailment* (\sqsupset) from *independence* ($\#$), both of which result in the NLI label *neutral*, we perform relation augmentation to create samples whose hypothesis entails the premise. Specifically, for each sample that is originally labeled as entailment in the training set, we create an augmented sample by exchanging the premise and the hypothesis. Note that we avoid augmenting the case where P and H mutually entail each other because the new premise still entails the hypothesis after the exchange. To achieve this, we exclude an exchanged sample from relation augmentation if it is still identified as entailment by a pretrained model finetuned on SNLI (Bowman et al., 2015). In terms of the augmented samples, the program receives a positive reward during training if and only if it reaches the correct final state *reverse entailment* (\sqsupset).

4.3 Introspective Revision

The key challenges of developing interpretable neural natural logic models include coping with spurious reasoning paths (incorrect paths $\mathbf{r} = \{r_t\}_{t=1}^m$ leading to the correct inferential label for a premise-hypothesis pair) as well as training inefficiency. Finding a correct program that reaches the correct label is challenging because it is inefficient to explore a space of 5^m paths for a reward. A positive reward to the correct path is often sparse.

We propose to use the fail-and-fix approach based on the newly proposed Back-Search algorithm (Li et al., 2020) to mitigate training

inefficiency caused by sparse positive rewards, which, based on a failed program that earns no positive reward, searches for better proof paths in its neighborhood that reaches the correct final prediction. To solve the spurious issue in this fail-and-fix framework, we propose Introspective Revision that leverages external commonsense knowledge (denoted as \mathcal{K}) to control spurious proof paths. We believe unstated commonsense knowledge is important not only for improving prediction accuracy (which, as discussed in Section 2, often results from fitting to spurious correlations), but critical for developing interpretable natural language reasoning models by avoiding spurious proofs.

Without loss of generality, we distinguish a non-spurious program r^* from spurious ones based on the following assumption, whose effectiveness will be shown and discussed in our experiments.

Assumption 4.1 *A program r^* has a larger probability than another program r to be a non-spurious program if r^* has a better agreement with the external knowledge base \mathcal{K} .*

External Knowledge: Previous work (Chen et al., 2017a) queries the knowledge base for each pair of words between a premise and hypothesis exhaustively, which is inefficient and likely to introduce undesired local relations. As a remedy, we found that the lightweight text alignment tool JacanaAligner (Yao et al., 2013), though not accurate enough to align all pairs of associated phrases in the input, can be used to guide the search. For a hypothesis phrase s , we first apply JacanaAligner to obtain its associated premise phrase \tilde{s} , and then query the WordNet (Miller, 1998) database for the possible natural logic relations for the phrase pair $\langle s, \tilde{s} \rangle$:

- ① **Equivalence** (\equiv): $s = \tilde{s}$ or $s \subset \tilde{s}$;
- ② **Forward Entailment** (\sqsubset): $s \subset \tilde{s}$ or $s = \tilde{s}$ or $\exists u \in s, v \in \tilde{s}$ and u is a hypernym of v ;
- ③ **Reverse Entailment** (\supset): $\tilde{s} \subset s$ or $\exists u \in s, v \in \tilde{s}$ and v is a hypernym of u ;
- ④ **Alternation** (\circ): $\exists u \in s, v \in \tilde{s}$ and u is an antonym of v ;

where u, v denote tokens in the phrase and $s \subset \tilde{s}$ means that s is a sub-phrase of \tilde{s} . The local relations suggested by the knowledge base are for-

mulated as a set of triplet proposals $(t, \tilde{r}_t, \mathbf{p}_t[\tilde{r}_t])$, where t is the time step, \tilde{r}_t is the suggested relation, and $\mathbf{p}_t[\tilde{r}_t]$ is the model predicted probability that corresponds to \tilde{r}_t .

Human-curated rules, which are designed to retrieve natural logic relations from the knowledge base, are often imperfect. They inevitably introduce errors due to language variations. For example, intuitively $s \subset \tilde{s}$ indicates *forward entailment* (\sqsubset); for example, “white cat” entails “cat”, while there are cases where the sub-phrase rule indicates *equivalence* (\equiv); for example, “have a chat with” is equivalent to “chat with” in meaning. In rare cases, the relation can be *alternation* (\circ); for example, “fake gun” and “gun” are distinct concepts. While $s = \tilde{s}$ often indicates *equivalence* (\equiv), our rules need to handle cases where the adverbial is posed in separate phrases; for example, “a bike” and “near the park” entails “a bike”.

To deal with this issue, instead of making an intensive effort to design sophisticated rules to pinpoint a single accurate relation, we design relatively coarse rules to narrow down the possibilities and leave the final choice to the model. Specifically, at each step we provide the model with multiple possible candidates, and the proposed introspective revision algorithm introduced in this section decides to accept a useful proposal or reject a misleading one, based on both the reasoning objective (i.e., the label) and the predicted relation distribution.

Algorithm: Given a program $r = \{r_t\}_{t=1}^m$, the goal of the Introspective Revision algorithm is to find a program r^* in the neighbourhood of r that executes to the correct answer y while maintaining a large agreement with the external knowledge \mathcal{K} , as detailed in Algorithm 1. The algorithm starts with knowledge-driven revision (lines 2~15). We arrange the triplet proposals obtained from the knowledge base as a priority queue $\Phi = \{(t, \tilde{r}_t, \mathbf{p}_t[\tilde{r}_t]) \mid 0 < t \leq m, \tilde{r}_t \in \mathcal{B}\}$. In each iteration the queue pops the triplet with the largest probability $\mathbf{p}_t[\tilde{r}_t]$ that specifies a modification to the sampled program $r' = \mathbf{Fix}(r, t, \tilde{r}_t)$. In other words, changing the relation r_t at step t of program r to the proposed relation \tilde{r}_t yields a new program r' . Following Li et al. (2020), the modification is accepted with a probability $1 - \epsilon$ if r' executes to the correct answer y ; otherwise, it is accepted with a probability $\min(1, \mathbf{p}_t[\tilde{r}_t]/\mathbf{p}_t[r_t])$.

Algorithm 1: Introspective Revision

Input: program $r = \{r_1, \dots, r_m\}$, label y , relation proposals Φ
Param: maximum step M , threshold ϵ
Output: r^*

```
1 Init  $r^* = r$ ,  $i = 0$ 
  /* Knowledge-driven revision. */
2 while  $i < M$  and  $\Phi \neq \emptyset$  do
3    $(t, \tilde{r}_t, \mathbf{p}_t[\tilde{r}_t]) = \Phi.pop()$ 
4   sample  $u \sim \mathcal{U}(0, 1)$ 
5    $r' = \mathbf{Fix}(r^*, \tilde{r}_t, t)$ 
  /* Accept or reject  $(t, \tilde{r}_t, \mathbf{p}_t[\tilde{r}_t])$  */
6   if  $\mathbf{Execute}(r') = y$  and  $u > \epsilon$  then
7      $r^* = r'$ 
8   else
9     sample  $u \sim \mathcal{U}(0, 1)$ 
10    if  $u < \min(1, \frac{\mathbf{p}_t[\tilde{r}_t]}{\mathbf{p}_t[r_t]})$  then
11       $r^* = r'$ 
12    end
13  end
14   $i = i + 1$ 
15 end
  /* Answer-driven revision. */
16 if  $\mathbf{Execute}(r^*) \neq y$  then
17    $\Psi = \mathbf{GridSearch}(r^*, \Phi, y)$ 
18   if  $\Psi \neq \emptyset$  then
19      $(t, \tilde{r}_t, \mathbf{p}_t[\tilde{r}_t]) = \Psi.pop()$ 
20      $r^* = \mathbf{Fix}(r^*, \tilde{r}_t, t)$ 
21   end
22 end
Return:  $r^*$ 
```

The hyperparameter ϵ encourages the model to explore low-probability proposals. For each sample, the model accepts or rejects up to M triplets.

The knowledge-driven revision above is conservative because only the top- M proposals are considered. However, there are complex cases where the program still cannot reach the correct answer after M steps, or where the provided proposals are insufficient to solve the problem. In these cases, we apply the answer-driven revision (lines 17~22) by conducting a $5 \times m$ grid search to find modifications that lead to the correct answers. Among the search results Ψ , we accept the triplet with the maximum probability. A detailed description of the grid search is presented in Algorithm 2.

Following the reward in Eq. 7 and the objective function in Eq. 8, we compute a new objective function J' with the modified program r^* and its corresponding reward R^* . The model learns by

Algorithm 2: GridSearch

Input: program $r = \{r_1, \dots, r_m\}$, label y , relation proposals Φ
Output: Ψ

```
1 Init  $\Psi = \mathbf{PriorityQueue}()$ 
2 for  $t \leftarrow 1$  to  $m$  do
3   foreach  $\tilde{r} \in \mathfrak{B}$  do
4      $r' = \mathbf{Fix}(r, \tilde{r}, t)$ 
5     if  $\mathbf{Execute}(r') = y$  then
6        $\Psi.push((t, \tilde{r}, \mathbf{p}_t[\tilde{r}]))$ 
7     end
8   end
9 end
10 if  $\Psi \cap \Phi \neq \emptyset$  then
11    $\Psi = \Psi \cap \Phi$ 
12 end
Return:  $\Psi$ 
```

optimizing the hybrid training objective J_{hybrid} , defined as:

$$J' = - \sum_{t=1}^m \log \mathbf{p}_t[r_t^*] \cdot R_t^* \quad (9)$$

$$J_{\text{hybrid}} = \lambda J + (1 - \lambda) J', \quad (10)$$

where λ is a weight that specifies the importance of the revision. The introspective revision algorithm is only applied during training since the label y is required to determine whether a proposal is accepted or not.

5 Experiments

We evaluate the performance of the proposed model on six NLI tasks from various perspectives: the ability of performing monotonicity inference (Section 5.2), reasoning systematicity (Section 5.3), and model interpretability (Section 5.4).

Our model is trained on Stanford Natural Language Inference (SNLI) (Bowman et al., 2015), in which the relation between a premise and hypothesis is classified to either *entailment*, *contradiction*, or *neutral*. We set the unit reward $\mu = 1.0$, and optimize our model with Adam gradient descent for six epochs with a learning rate of $2e-5$. We compare the models with discount factor $\gamma \in \{0.25, 0.50, 0.75, 1.00\}$ and $\epsilon \in \{0.05, 0.10, 0.20\}$. We found that the test accuracies are not sensitive to γ when $\gamma \geq 0.50$, and we select $\gamma = 0.50$, $\epsilon = 0.20$, which achieved the best validation accuracy on SNLI. For the introspective revision algorithm we set

Phase	Revision (Knowl. / Answ. / Both)	Success Rate of Revision
Start	80.4% (85.2% / 8.1% / 6.7%)	59.4%
End	81.7% (80.3% / 5.9% / 13.8%)	98.4%

Table 4: The percentage of samples being revised and the revision success rate at the start/end of the training.

Relation	Knowledge Available	Knowl.-driven start / end	Answ.-driven start / end
Equivalence	1.035	0.595 / 0.482	0.096 / 0.026
Fwd. Entail	1.087	0.370 / 0.523	0.014 / 0.037
Rev. Entail	0.249	0.097 / 0.191	0.008 / 0.061
Alternation	0.012	0.004 / 0.008	0.001 / 0.037
Sum	2.383	1.066 / 1.204	0.119 / 0.161

Table 5: The average number of triplet proposals obtained from the WordNet per sample and the average number of proposals accepted by knowledge or answer-driven revision at the start / end of the training.

$M=3$ based on the average number of proposals (2.383 proposals/sample) in Table 5. We treat the revised program and the original program as equally informative by setting $\lambda = 0.5$. Our code is available at <https://github.com/feng-yufei/NS-NLI>.

5.1 Statistics for Introspective Revision

In Table 4, we present the statistics for the introspective revision at the start/end of the training, where the natural logic programs are sampled from the predicted distribution. Approximately 80% of the samples perform at least one step of revision, and at the end of the training, there is an increasing chance (98.4% vs. 59.4%) that introspective revision helps the model reach the final correct NLI prediction. In Table 5, we show the statistics of the average number of triplet proposals obtained from WordNet and the average number of proposals accepted by knowledge or answer-driven revision during training. *Equivalence* (\equiv) and *forward entailment* (\sqsubset) make up a large portion of the proposals, while the *alternation* relation is scarce due to the sparsity of the antonym relation obtained from WordNet. As a result, the numbers of proposals accepted in the knowledge-driven revision are imbalanced across different relations. Moreover, we found that the number of accepted answer-driven revisions slightly increased at the end of the training, which is due to the fact that as

the training proceeds, the programs produced by the model are closer to the target labels.

5.2 Performance on Monotonicity Reasoning

We conduct experiments on multiple recently proposed challenging test datasets for monotonicity inference: HELP (Yanaka et al., 2019b), MED (Yanaka et al., 2019a), and Monotonicity NLI (MoNLI) (Geiger et al., 2020). Unlike SNLI, half of the samples in HELP, MED, and MoNLI are in downward monotone, and they are categorized as *entailment* or *non-entailment*. In the above datasets, a premise and the corresponding hypothesis differ by *1-hop*; that is, they are different by either a lexical substitution, insertion, or deletion. In addition, we also evaluated our model on the Natural Logic 2-hop dataset (Feng et al., 2020), which requires a model to perform a 2-hop natural logic composition according to Table 3.

We compare our model with popular natural language inference baselines including ESIM (Chen et al., 2017b), BERT-base (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and Feng et al. (2020). Following Yanaka et al. (2019a) and to ensure a fair comparison, all models are trained on SNLI, and during testing, we regard *contradiction* and *neutral* as *non-entailment* if a binary prediction is required.

Table 6 shows the test accuracy on SNLI and four challenging test datasets. Our model performs consistently and significantly better than previous state-of-the-art models on all challenging datasets while achieving competitive “*in-domain*” performance on SNLI. Manual inspection shows that compared to GPT-2, a significant proportion of the failure cases on SNLI are due to errors from the projectivity parser, and the ambiguity between *contradiction* and *neutral* (Bowman et al., 2015). The introspective revision algorithm significantly boosts the model performance on the monotonicity reasoning test sets (A0 vs. A3). Ablation shows that the knowledge-driven revision improves the performance on MoNLI and the 2-hop dataset (A0 vs. A1), which suggests that without proper constraints, the answer-driven revision can lead to spurious reasoning. We found that removing *equivalence* (\equiv) (knowledge ①) from the knowledge-driven revision lowers the performance, because in this case the knowledge-driven revision mistakenly encourages the model to replace *equivalence* (\equiv) with *forward entailment*

Model	SNLI	HELP	MED	MoNLI	Nature Logic-2hop
ESIM (Chen et al., 2017b)	88.0	55.3	51.8	63.9	45.1
BERT-base (Devlin et al., 2019)	90.1	51.4	45.9	53.0	49.3
GPT-2 (Radford et al., 2019)	89.5	52.1	44.8	57.5	48.3
Feng et al. (2020)	81.2	58.2	52.4	76.8	60.1
Ours – full model (A0)	87.5	65.9	66.7	87.8	62.2
w/o knowledge ①	87.2	62.8	62.2	77.0	61.7
w/o knowledge ②	87.4	65.8	64.2	81.7	51.7
w/o knowledge ③	87.5	65.6	65.9	83.6	61.6
w/o knowledge ④	87.6	65.4	64.7	83.3	58.2
w/o knowledge ①②③④ (A1)	87.6	65.0	64.8	77.3	48.8
w/o answer driven revision (A2)	87.5	65.4	65.5	85.1	60.9
w/o introspective revision (A3)	87.6	62.1	60.7	74.4	53.3
w/o relation augmentation (A4)	87.8	59.6	54.7	74.7	59.9
Ours w/ masked attention (A5)	75.9	61.3	61.6	70.9	54.6

Table 6: Model accuracy on multiple challenging test datasets. All models are trained on SNLI and the results of model (A0) \sim (A5) are the average of 3 models starting from different consistently-seeded initializations.

(\square), which may lead to incorrect prediction under downward monotonicity. Compared to *forward entailment* (\square) (knowledge ②), removing *reverse entailment* (\square) (knowledge ③) and *alternation* (\neg) (knowledge ④) does not significantly affect the results. We deduce that the relative importance of different relations are affected by the frequency of the external knowledge, and without the help of the knowledge-driven revision, the model can still learn the *reverse entailment* (\square) relation from relation augmentation in Section 4.2. The performance drops when the relation augmentation is vacant (A0 vs. A4).

We also include the model that masks both the past and the future hypothesis chunks in the transformer attention layers for local relation prediction (A5). The model with masked attention yields significantly lower performance on SNLI, partly due to the fact that aggressively masking the past hypothesis chunks changes the structure of the pretrained GPT-2 model, and thus the model benefits less from the pretrained representations.

5.3 Systematicity of Monotonicity Inference

Making systematic generalizations from limited data is an essential property of human language (Lake and Baroni, 2018). While finetuning pretrained transformers achieves high NLI accuracy, Yanaka et al. (2020) have recently shown that these models have limited capability of capturing the systematicity of monotonicity inference. We use the dataset proposed by Yanaka et al. (2020) to

evaluate the model’s ability in compositional generalization: The model is exposed to all primitive types of quantifiers Q and predicate replacements R , but samples in the training set and test set contain different combinations of quantifiers and predicate replacements. Specifically, with an arbitrarily selected set of quantifiers $\{q\}$ and predicate replacement $\{r\}$, the training set contains data $D^{\{q\},R} \cup D^{Q,\{r\}}$ while the test data only includes the complementary set $D^{Q \setminus \{q\}, R \setminus \{r\}}$. An example of compositional generalization is shown below:

- (1) P : Some dogs run \Rightarrow H : Some animals run
- (2) P : No animals run \Rightarrow H : No dogs runs
- (3) P : Some small dogs run \Rightarrow H : Some dogs run

An ideal model can learn from the training samples (1), (2), and (3) the entailment relations between concepts *small dog* \square *dog* \square *animal*, as well as the fact that the quantifier *some* indicates the upward monotonicity and *no* indicates the downward. During testing, the model needs to compose the entailment relations and the monotonicity signatures to make inference over unseen combinations, for example, sample (4):

- (4) P : No dogs run \Rightarrow H : No small dogs run
- (5) P : Near the shore, no dogs run \Rightarrow
 H : Near the shore, no small dogs run

To test the model stability, Yanaka et al. (2020) also added adverbs or prepositional phrases as test-only noise to the beginning of both the premise and the hypothesis, for example, sample (5).

Model	Train	Test	Test _{adv}	Test _{pp}
BERT-base	100.0	69.2	50.8	49.3
GPT-2	100.0	25.6	35.6	35.4
BERT-base ^{↑↓}	100.0	65.4	51.4	52.7
GPT-2 ^{↑↓}	100.0	28.1	35.1	39.6
Ours w/o. IR	91.3	79.3	57.1	54.0
Ours	98.4	95.1	61.0	61.5

Table 7: Results for compositional generalization; $\uparrow\downarrow$ marks the models with polarity features.

In Table 7, all models are trained with 3,270 samples and tested on the complementary test set with about 9,112 examples, exactly following the data split in Yanaka et al. (2020). While all baseline models achieved high training accuracy, BERT has limited performance on the test set. For our model, there is only a 3% gap between the training and test performance, which demonstrates that our model successfully learns to identify and compose the natural logic relations of the predicate replacements with limited training examples.

We also compare our model to variants of BERT and GPT-2 models that are aware of the token projectivity (models with $\uparrow\downarrow$ in Table 7). Specifically, for each token, we concatenate the hidden states in the final layer of transformer with its projectivity feature. We aggregated the concatenated features with multiple feed-forward layers and applied average pooling before sending them to the classification layer. Results show that BERT and GPT-2 do not benefit from the projectivity features. The test accuracy drops with additional adverbs and preposition phrases, leaving space for future research on the robustness to unseen perturbations.

5.4 Evaluation of Model Explainability

The proposed model provides built-in interpretability following natural logic—the execution of programs $\{z_t\}_{t=1}^m$ (Eq. 6) provides explanation along with the model’s decision making process, namely giving a *faithful* explanation (Jacovi and Goldberg, 2020). To evaluate the model interpretability, we derive the predicted rationales from the natural logic programs and compare it with human annotations in e-SNLI (Camburu et al., 2018). Specifically, our model regards as rationales the hypothesis phrases s_t that satisfies: (1) z_t points to the final prediction according to the grouping described at the

end of Section 4.2; (2) $z_t \neq z_{t-1}$. Following DeYoung et al. (2020), we use Intersection Over Union (IOU) formulated in Eq. (11) as the evaluation metric: the numerator is the number of shared tokens between the model generated rationales and the gold rationales, and the denominator is the number of tokens in the union. We also compute *precision*, *recall* and *F1 score* to measure the phrasal agreement between the predicted rationales and human annotations. We also invited 3 graduate students (not the authors of this paper) to evaluate the quality of the predicted rationales on the first 100 test samples in e-SNLI. Given the premise-hypothesis pair and the golden label, the evaluators judged the explanation as plausible if the predicted rationale (1) alone is sufficient to justify the label, and; (2) does not include the whole hypothesis sentence.

$$IOU = \frac{\text{num-tokens}\{R_{pred} \cap R_{truth}\}}{\text{num-tokens}\{R_{pred} \cup R_{truth}\}} \quad (11)$$

From the perspective of natural logic, we follow Feng et al. (2020) to evaluate the quality of the natural logic programs. For each sample, the Natural Logic 2-hop dataset provides the gold program execution states, and we evaluated the accuracy of our predicted states \hat{z}_t against the ground-truth. We compare our model with representative neural rationalization models proposed by Lei et al. (2016), which learns to extract rationales without direct supervision, and Feng et al. (2020), which explains its prediction by generating natural logic reasoning paths. The summary statistics in Table 8 shows that our model matches Lei et al. (2016) on the IOU score, and that it produces rationales with significantly higher precision and F1-scores on the e-SNLI test set. The superior rationalization performance is also supported by the human evaluation mentioned above (the 4th column in Table 8). Compared to Feng et al. (2020), our model produces intermediate natural logic states that better agree with the ground truth. The results in Table 8 show that the model explanation significantly benefits from the external knowledge (B0 vs. B1), and the answer-driven revision alone does not improve the quality of the generated rationales (B1 vs. B2). We also compare our model to the system that replaces the uni-directional attention model GPT-2

Model	e-SNLI	e-SNLI	e-SNLI	Natural Logic 2-hop Acc.
	IOU	Precision / Recall / F1	Human Eval.	
Lei et al. (2016)	0.42	0.37 / 0.46 / 0.41	56 / 100	–
Feng et al. (2020)	0.27	0.21 / 0.35 / 0.26	52 / 100	0.44
Ours – full model (B0)	0.44	0.58 / 0.49 / 0.53	80 / 100	0.52
w/o. external knowledge (B1)	0.41	0.53 / 0.45 / 0.48	67 / 100	0.44
w/o. introspective revision (B2)	0.40	0.52 / 0.43 / 0.47	68 / 100	0.43
w/o. relation augmentation (B3)	0.44	0.57 / 0.48 / 0.52	75 / 100	0.51
Ours – BERT encoder (B4)	0.14	0.20 / 0.15 / 0.17	29 / 100	0.28

Table 8: Evaluation for the model generated explanation.

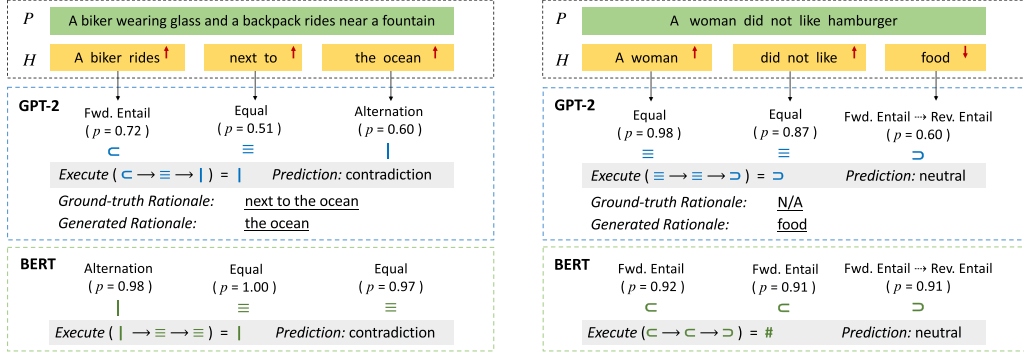


Figure 2: Examples for predictions and explanation for some cases from SNLI (left) and MoNLI (right).

with the bi-directional attention model BERT. The model with BERT encoder yields significantly lower scores on interpretability (B0 vs. B4).

5.5 Case Study

The upper part of the Figure 2 shows how our natural logic model makes predictions during testing. The left example involves upward monotone. Upon seeing the premise and the first hypothesis phrase *A biker rides*, the model predicts the local relation as *forward entailment* ($r_1 = \sqsubset$) at time step $t=1$. The predicted relation stays unchanged after applying the projection function $\rho(\sqsubset) = \sqsubset$ because it is in the context of upward monotone. According to Table 3 we have $z_1 = z_0 \otimes r_1 = \sqsubset$. Similarly, as the second prediction for the phrase *next to*, relation *equivalence* ($r_2 = \equiv$) does not change the reasoning states because $z_2 = z_1 \otimes r_2 = \sqsubset$. The third hypothesis phrase *the ocean* is a distinct concept against *a fountain* in the premise, our model outputs relation *alternation* ($r_3 = \mid$) and we have $z_3 = z_2 \otimes r_3 = \mid$. The model runs out of the hypothesis phrases after 3 steps, and reaches *contradiction* according to the final state z_3 .

An additional example with downward monotone is illustrated on the right of Figure 2. Our

model predicts the relation *forward entailment* ($r_3 = \sqsubset$) at the third time step since *food* includes *hamburger*. The projection function flips the relation to *reverse entailment* (\supset) because according to the projectivity in Table 2, the first argument that follows negation *did not* is in downward monotone, i.e., $\rho(\sqsubset) = \supset$.

At the bottom of Figure 2, we provide examples for the reasoning processes produced by the natural logic model that is built upon the bi-directional attention model BERT. Although it produces the same final labels as our proposed model, the model based on BERT can predict wrong local relations due to its entangling effect. Specifically, the model with bi-directional attention is prone to make its final decision in the first place (82% of the cases in the human evaluation), and then predict local relations that can keep the initial decision during the program execution (according to the composition rules in Table 3). In the first example in Figure 2, to keep the first predicted relation *alternation* (\mid) unchanged during execution, the model subsequently predicts a series of *equivalence* (\equiv) relations. In the second example, the model predicts local relation *forward entailment* (\sqsubset) for each hypothesis phrase, and at the last step, the *forward entailment* (\sqsubset) relation is

projected to *reverse entailment* (\sqsupset) according to the projectivity.

6 Summary

The proposed neuro-symbolic framework integrates the long-studied natural logic with reinforcement learning and introspective revision, effectively rewarding the intermediate proof paths and leveraging external knowledge to alleviate spurious reasoning. The model has built-in interpretability following natural logic, which allows for a wide range of intuitive inferences easily understandable by humans. Experimental results show the model’s superior capability in monotonicity-based inferences and systematic generalization, compared to previous models on the existing datasets, while the model keeps competitive performance on the generic SNLI test set.

Acknowledgments

This research was supported by NSERC Discovery Grants. We thank the anonymous reviewers and action editors for their helpful comments.

References

- Jacob Andreas, Dan Klein, and Sergey Levine. 2017. Modular multitask reinforcement learning with policy sketches. In *International Conference on Machine Learning*, pages 166–175. PMLR.
- Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. 2017. Hind-sight experience replay. In *Advances in Neural Information Processing Systems*, volume 30.
- Gabor Angeli and Christopher D. Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar. <https://doi.org/10.3115/v1/D14-1059>
- Murat Aydede. 1997. Language of thought: The connectionist contribution. *Minds and Machines*, 7(1):57–101. <https://doi.org/10.1023/A:1008203301671>
- Johan van Benthem. 1988. The semantics of variety in categorial grammar. *Categorial Grammar*, 25:37–55. <https://doi.org/10.1075/11see.25.06ben>.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal. <https://doi.org/10.18653/v1/D15-1075>.
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural language inference with natural language explanations, *Advances in Neural Information Processing Systems*.
- Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2017a. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. Melbourne, Australia. <https://doi.org/10.18653/v1/P18-1224>
- Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vancouver, Canada. <https://doi.org/10.18653/v1/P17-1152>
- Zeming Chen, Qiyue Gao, and Lawrence S. Moss. 2021. Neurallog: Natural language inference with joint neural and logical reasoning. *arXiv preprint arXiv:2105.14167*. <https://doi.org/10.18653/v1/2021.starsem-1.7>
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*. https://doi.org/10.1007/11736790_9
- Luc De Raedt, Robin Manhaeve, Sebastijan Dumancic, Thomas Demeester, and Angelika Kimmig. 2019. Neuro-symbolic= neural+ logical+ probabilistic. In *NeSy’19@ IJCAI, the 14th*

- International Workshop on Neural-Symbolic Learning and Reasoning*. Macao, China.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Minneapolis, USA.
- Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. Eraser: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458. <https://doi.org/10.18653/v1/2020.acl-main.408>
- Richard Evans and Edward Grefenstette. 2018. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research (JAIR)*, 61:1–64. <https://doi.org/10.1613/jair.5714>
- Yufei Feng, Zi’ou Zheng, Quan Liu, Michael Greenspan, and Xiaodan Zhu. 2020. Exploring end-to-end differentiable natural logic modeling. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1172–1185. <https://doi.org/10.18653/v1/2020.coling-main.101>
- Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71. [https://doi.org/10.1016/0010-0277\(88\)90031-5](https://doi.org/10.1016/0010-0277(88)90031-5)
- Artur d’Avila Garcez, Tarek R. Besold, Luc De Raedt, Peter Földiak, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luis C. Lamb, Risto Miikkulainen, and Daniel L. Silver. 2015. Neural-symbolic learning and reasoning: contributions and challenges. In *2015 AAAI Spring Symposium Series*. Austin, Texas, USA.
- Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2019. Posing fair generalization tasks for natural language inference. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4485–4495. <https://doi.org/10.18653/v1/D19-1456>
- Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173. <https://doi.org/10.18653/v1/2020.blackboxnlp-1.16>
- Emily Goodwin, Koustuv Sinha, and Timothy J. O’Donnell. 2020. Probing linguistic systematicity. *ACL 2020*. <https://doi.org/10.18653/v1/2020.acl-main.177>
- Prasoon Goyal, Scott Niekum, and Raymond J. Mooney. 2019. Using natural language for reward shaping in reinforcement learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2385–2391. International Joint Conferences on Artificial Intelligence Organization. <https://doi.org/10.24963/ijcai.2019/331>
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL-HLT(2)*. <https://doi.org/10.18653/v1/N18-2017>
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. <https://doi.org/10.5281/zenodo.1212303>.
- Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: A lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*.
- Thomas F. Icard. 2012. Inclusion and exclusion in natural language. *Studia Logica*. <https://doi.org/10.1007/s11225-012-9425-8>
- Thomas F. Icard and Lawrence S. Moss. 2014. Recent progress on monotonicity. In *Linguistic Issues in Language Technology*. Citeseer.
- Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How

- should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.386>.
- Aikaterini-Lida Kalouli, Richard Crouch, and Valeria de Paiva. 2020. Hy-NLI: A hybrid system for natural language inference. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5235–5249. <https://doi.org/10.18653/v1/2020.coling-main.459>
- Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.
- George Lakoff. 1970. Linguistics and natural logic. *Synthese*. <https://doi.org/10.1007/BF00413602>
- Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117. <https://doi.org/10.18653/v1/D16-1011>
- Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. 2020. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *International Conference on Machine Learning*, pages 5884–5894. PMLR.
- Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on Freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada. <https://doi.org/10.18653/v1/P17-1003>
- Bill MacCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University.
- Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802–811. <https://doi.org/10.3115/1613715.1613817>
- Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 140–156. Tilburg, The Netherlands. Association for Computational Linguistics. <https://doi.org/10.3115/1693756.1693772>.
- Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2018. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.
- David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. 2018. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4942–4950. <https://doi.org/10.1109/CVPR.2018.00519>
- George A. Miller. 1998. *WordNet: An electronic lexical database*. MIT Press.
- Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the Fifth International Workshop on Inference in Computational Semantics (icos-5)*.
- Jessica Ouyang and Kathy McKeown. 2019. Neural network alignment for sentential paraphrases. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics. <https://doi.org/10.18653/v1/P19-1467>
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*. <https://doi.org/10.18653/v1/S18-2023>
- Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, and Martin Riedmiller. 2017.

- Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*. New York, USA. <https://doi.org/10.1609/aaai.v34i05.6397>
- Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degraeve, Tom Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. 2018. Learning by playing solving sparse reward tasks from scratch. In *International Conference on Machine Learning*, pages 4344–4353. PMLR.
- Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end differentiable proving. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*. Long Beach, USA.
- Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. 2019. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. In *NeurIPS*.
- Víctor Manuel Sánchez Valencia. 1991. *Studies on natural logic and categorial grammar*. Universiteit van Amsterdam.
- Johan Van Benthem. 1986. *Essays in logical semantics*, Springer. <https://doi.org/10.1007/978-94-009-4540-1>
- Johan Van Benthem. 1995. *Language in Action: categories, lambdas and dynamic logic*. MIT Press.
- Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. Nlprolog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. Austin, Texas, United States. <https://doi.org/10.18653/v1/P19-1618>
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. <https://doi.org/10.18653/v1/N18-1101>
- Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256. <https://doi.org/10.1023/A:1022672621406>, <https://doi.org/10.1007/BF00992696>
- Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117. <https://doi.org/10.18653/v1/2020.acl-main.543>
- Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Austin, Texas, United States. <https://doi.org/10.18653/v1/W19-4804>
- Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. Help: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM)*. Minneapolis, Minnesota, USA. <https://doi.org/10.18653/v1/S19-1027>
- Fan Yang, Zhilin Yang, and William W. Cohen. 2017. Differentiable learning of logical rules

for knowledge base reasoning. In *Advances in Neural Information Processing Systems*. Long Beach, USA.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. A lightweight and high performance monolingual word aligner. In *Proceedings of the 51st Annual Meeting of the Association for*

Computational Linguistics (Volume 2: Short Papers), pages 702–707.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. 2018. Neural-symbolic VQA: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050.