

# A Differentiable Integer Linear Programming Solver for Explanation-Based Natural Language Inference

Mokanarangan Thayaparan<sup>1,3</sup>, Marco Valentino<sup>3</sup>, André Freitas<sup>1,2,3</sup>

<sup>1</sup>Department of Computer Science, University of Manchester, UK

<sup>2</sup> National Biomarker Centre, CRUK-MI, University of Manchester, UK

<sup>3</sup>Idiap Research Institute, Switzerland

mokanarangan.thayaparan@manchester.uk, marco.valentino@idiap.ch, andre.freitas@idiap.ch

## Abstract

Integer Linear Programming (ILP) has been proposed as a formalism for encoding precise structural and semantic constraints for Natural Language Inference (NLI). However, traditional ILP frameworks are non-differentiable, posing critical challenges for the integration of continuous language representations based on deep learning. In this paper, we introduce a novel approach, named Diff-Comb Explainer, a neuro-symbolic architecture for explanation-based NLI based on Differentiable BlackBox Combinatorial Solvers (DBCS). Differently from existing neuro-symbolic solvers, Diff-Comb Explainer does not necessitate a continuous relaxation of the semantic constraints, enabling a direct, more precise, and efficient incorporation of neural representations into the ILP formulation. Our experiments demonstrate that Diff-Comb Explainer achieves superior performance when compared to conventional ILP solvers, neuro-symbolic black-box solvers, and Transformer-based encoders. Moreover, a deeper analysis reveals that Diff-Comb Explainer can significantly improve the precision, consistency, and faithfulness of the constructed explanations, opening new opportunities for research on neuro-symbolic architectures for explainable and transparent NLI in complex domains.

**Keywords:** Natural Language Inference, Neuro-Symbolic AI, Constrained Optimization

## 1. Introduction

Given a question and a possible candidate answer expressed in natural language (which we refer to as *hypothesis*), ILP-based Natural Language Inference (NLI) aims to construct an explanation graph consisting of interconnected sentences to support and deduce the hypothesis (Figure 1).

To contrast the lack of interpretability of large deep learning models, ILP has been proposed as a formalism to build transparent and explainable NLI models, able to leverage explicit semantic constraints to address downstream inference tasks. ILP-based solvers, in fact, provide a viable mechanism to encode explicit and controllable assumptions about the structure of the explanations (Khashabi et al., 2018; Khot et al., 2017; Khashabi et al., 2016; Thayaparan et al., 2020).

However, existing ILP frameworks are *non-differentiable* and cannot be integrated as part of a broader deep learning architecture (Paulus et al., 2021; Pogančić et al., 2019). Therefore, these approaches have been so far limited by the exclusive adoption of hard-coded heuristics and cannot be easily optimised end-to-end to achieve performance comparable to deep learning counterparts (Thayaparan et al., 2021; Khashabi et al., 2018). At the same time, existing solutions for the integration of combinatorial solvers and deep learning require a continuous approximation of the semantic constraints, which inevitably lowers the

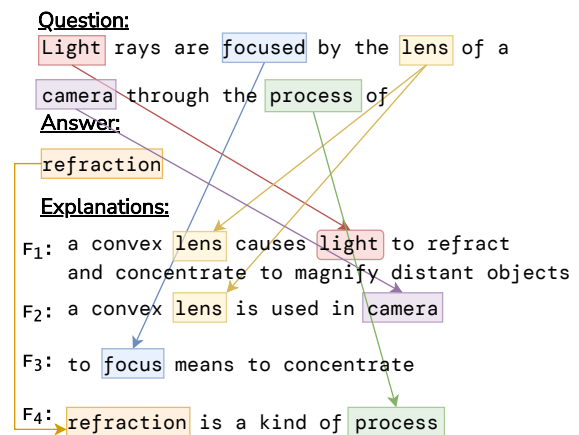


Figure 1: Example of a hypothesis (i.e., question + answer) and an explanation graph constructed via an ILP-based NLI model (Xie et al., 2020; Jansen et al., 2018).

expressive power of the ILP formulation and, as a consequence, the precision in the downstream inference process (Thayaparan et al., 2022).

To address these challenges, this paper proposes an end-to-end methodology for the integration of *Differentiable BlackBox Combinatorial Solvers* (DBCS) (Pogančić et al., 2019) into explanation-based NLI architectures. DBCS, in fact, enables an efficient approximation of the gradient for end-to-end optimisation, without the need

for a continuous relaxation of the original ILP formulation. Specifically, we demonstrate how a combinatorial optimization solver can be adopted as a composable building block in a broader NLI architecture with Transformers (Devlin et al., 2019).

In summary, the contributions of this paper are as follows:

1. We propose *Diff-Comb Explainer*, a novel ILP-based NLI solver that combines a differentiable black box combinatorial solver with Transformer-based encoders, enabling end-to-end differentiability without the need for a continuous relaxation of the ILP formulation.
2. We empirically demonstrate that *Diff-Comb Explainer* achieves superior performance on both explanation generation and answer selection when compared to differentiable and non-differentiable counterparts.
3. We demonstrate that thanks to the preservation of the original ILP formulation, *Diff-Comb Explainer* better reflects the underlying explanatory inference process leading to the final answer prediction, outperforming existing combinatorial solvers in terms of faithfulness and consistency.<sup>1</sup>

## 2. Related Work

**Constraint-based multi-hop inference** ILP has been applied for structured representation (Khashabi et al., 2016) and over semi-structured representation extracted from text (Khot et al., 2017; Khashabi et al., 2018). Early approaches were unsupervised. However, recently Thayaparan et al. (2021) proposed the ExplanationLP model optimised towards answer selection via Bayesian optimisation. ExplanationLP was limited to fine-tuning only nine parameters and used pre-trained neural embedding. *Diff-Explainer* (Thayaparan et al., 2022) was the first approach to integrate constraints into a deep-learning network via Differentiable Convex Optimisation Layer (Agrawal et al., 2019) by approximating ILP constraints using Semi-definite programming. (Lovász and Schrijver, 1991). Differently from our approach, however, *Diff-Explainer* requires a continuous approximation of the semantic constraints, a feature that inevitably affects the precision in the downstream inference process.

**Hybrid reasoning with Transformers** Clark et al. (2021) proposed “soft theorem provers” operating over explicit theories in language. This hybrid reasoning solver integrates natural language

<sup>1</sup>Code and data available at [https://github.com/neuro-symbolic-ai/diff\\_comb\\_explainer](https://github.com/neuro-symbolic-ai/diff_comb_explainer)

rules with transformers to perform deductive reasoning. Saha et al. (2020) improved on top of it, enabling the answering of binary questions along with the proofs supporting the prediction. The multiProver (Saha et al., 2021) evolves on top of these conceptions to produce an approach that is capable of producing multiple proofs supporting the answer. While these hybrid reasoning approaches produce explainable and controllable inference, they assume the existence of natural language rules and have only been applied to synthetic datasets. On the other hand, our approach does not require extensive rules set and can tackle complex scientific and commonsense NLI problems.

## 3. *Diff-Comb Explainer*: Differentiable Integer Linear Programming Solver for Explanation-Based Natural Language Inference

ILP-based NLI is typically applied to multiple-choice question answering problems (Khashabi et al., 2018; Khot et al., 2017; Khashabi et al., 2016; Thayaparan et al., 2021). Given a Question ( $Q$ ) and a set of candidate answers  $C = \{c_1, c_2, c_3, \dots, c_n\}$  the aim of the task is to select the correct answer  $c_{ans}$ .

In order to achieve this, ILP-based approaches convert question-answer pairs into a list of hypotheses  $H = \{h_1, h_2, h_3, \dots, h_n\}$  (where  $h_i$  is the concatenation of  $Q$  with  $c_i$ ) and typically adopt a retrieval model (e.g: BM25, FAISS (Johnson et al., 2017)), to select a list of candidate explanatory facts  $F = \{f_1, f_2, f_3, \dots, f_k\}$ . Then construct a weighted graph  $G = (V, E, W)$  with edge weights  $W: E \rightarrow \mathbb{R}$  where  $V = \{\{h_i\} \cup F\}$ , edge weight  $W_{ik}$  of each edge  $E_{ik}$  denote how relevant a fact  $f_k$  is with respect to the hypothesis  $h_i$ .

Given this premise, ILP-based NLI can be defined as follows (Thayaparan et al., 2022):

**Definition 3.1 (ILP-Based NLI).** Find a subset  $V^* \subseteq V$ ,  $h \in V^*$  and  $E^* \subseteq E$  such that the induced subgraph  $G^* = (V^*, E^*)$  is connected, weight  $W[G^* = (V^*, E^*)] := \sum_{e \in E^*} W(e)$  is maximal and adheres to set of constraints  $M_c$  designed to emulate multi-hop inference. The hypothesis  $h_i$  with the highest subgraph weight  $W[G^* = (V^*, E^*)]$  is selected to be the correct answer  $c_{ans}$ .

As illustrated in Figure 2, *Diff-Comb Explainer* has 3 major parts: *Graph Construction*, *Subgraph Selection* and *Answer/Explanation Selection*. In *Graph Construction*, for each candidate answer  $c_i$  we construct a graph  $G^i = (V^i, E^i, W^i)$  where  $V^i = \{h_i\} \cup \{F\}$  and the weights  $W_{ik}^i$  of each edge  $E_{ik}^i$  denote how relevant a fact  $f_k$  is with respect

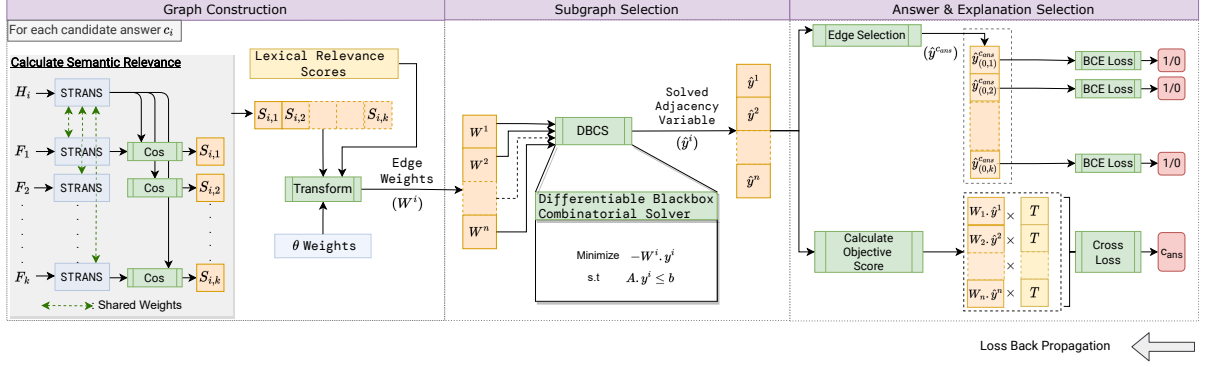


Figure 2: End-to-end architectural diagram of *Diff-Comb Explainer*. The integration of Differentiable Blackbox Combinatorial solvers will result in better explanation generation and answer prediction.

to the hypothesis  $h_i$ . These edge weights ( $W_{ik}^i$ ) are calculated using a weighted ( $\theta$ ) sum of scores calculated using transformer-based (*STrans*) embeddings and lexical models.

In the *Subgraph Selection* step, for each  $G^i$ , Differentiable Blackbox Combinatorial Solver (DBCS) with constraints are applied to extract subgraph  $G^*$ . In this paper, we adopt the constraints proposed for ExplanationLP (Thayaparan et al., 2021) for ensuring a fair comparison between different models. ExplanationLP explicit abstraction by grouping facts into abstract ( $F_A$ ) and grounding ( $F_G$ ). Abstract facts are the core scientific statements that a question is attempting to test, and grounding facts link concepts in the abstract facts to specific terms in the hypothesis (e.g. taxonomic relations). For example, in Figure 1 the core scientific fact is about the nature of convex lens and how they refract light ( $F_1$ ). Grounding Facts  $F_2, F_3, F_4$  are usually is-a relations that help to connect abstract concepts to concrete instances in the hypothesis.

Finally, in *Answer/Explanation Selection* the model predicts the final answer  $c_{ans}$  and relevant explanations  $F_{exp}$ . During training time, the loss is calculated based on gold answer/explanations to fine-tune the transformers (*STrans*) and weights ( $\theta$ ). The rest of the section explains each of the components in detail.

### 3.1. Graph Construction

In order to facilitate the construction of inference chains, the retrieved facts  $F$  are classified into *grounding* facts  $F_G = \{f_1^g, f_2^g, f_3^g, \dots, f_l^g\}$  and *abstract* facts  $F_A = \{f_1^a, f_2^a, f_3^a, \dots, f_m^a\}$  such that  $F = F_A \cup F_G$  and  $l + m = k$  (Thayaparan et al., 2021). Here, We use two relevance scoring functions to calculate the edge weights: semantic and lexical scores. We use a Sentence-Transformer (STrans) (Reimers et al., 2019) bi-encoder architecture to calculate the semantic relevance. The semantic relevance score from STrans is comple-

mented with the lexical relevance score. The semantic and lexical relevance scores are calculated as follows:

**Semantic Relevance ( $s$ )** : Given a hypothesis  $h_i$  and fact  $f_j$  we compute sentence vectors of  $\vec{h}_i = STrans(h_i)$  and  $\vec{f}_j = STrans(f_j)$  and calculate the semantic relevance score using cosine-similarity as follows:

$$s_{ij} = S(\vec{h}_i, \vec{f}_j) = \frac{\vec{h}_i \cdot \vec{f}_j}{\|\vec{h}_i\| \|\vec{f}_j\|} \quad (1)$$

**Lexical Relevance ( $l$ )** : The lexical relevance score of hypothesis  $h_i$  and  $f_j$  is given by the percentage of overlaps between unique terms (here, the function *trm* extracts the lemmatized set of unique terms from the given text):

$$l_{ij} = L(h_i, f_j) = \frac{|trm(h_i) \cap trm(f_j)|}{max(|trm(h_i)|, |trm(f_j)|)} \quad (2)$$

Given the above scoring function, we construct the edge weights matrix ( $W^i$ ) for each hypothesis  $h_i$  as follows:

$$W^i_{jk} = \begin{cases} -\theta_{gg}l_{jk} & (j, k) \in F_G \\ -\theta_{aa}l_{jk} & (j, k) \in F_A \\ \theta_{ga}l_{jk} & j \in F_G, k \in F_A \\ \theta_{qgl}l_{jk} + \theta_{qgs}s_{jk} & j \in F_G, k = h_i \\ \theta_{qal}l_{jk} + \theta_{qal}s_{jk} & j \in F_A, k = h_i \end{cases} \quad (3)$$

Here relevance scores are weighted by  $\theta$  parameters which are clamped to  $[0, 1]$ .

### 3.2. Subgraph Selection

Given the above premises, the objective function is defined as:

$$\min -1(W^i \cdot y^i) \quad (4)$$

We adopt the edge variable  $y^i \in \{0, 1\}$  where  $y_{j,k}^i$  ( $j \neq k$ ) takes the value of 1 if edge  $E_{jk}^i$  belongs to the subgraph and  $y_{jj}^i$  takes the value of 1 iff  $V_j^i$  belongs to the subgraph.

Given the above variable, the constraints are defined as follows:

**Answer selection constraint** The candidate hypothesis should be part of the induced subgraph:

$$\sum_{j \in \{h_i\}} y_{jj}^i = 1 \quad (5)$$

**Edge and Node selection constraint** If node  $V_j^i$  and  $V_k^i$  are selected then edges  $E_{jk}^i$  and  $E_{kj}^i$  will be selected. If node  $V_j^i$  is selected, then edge  $E_{jj}^i$  will also be selected:

$$y_{jk}^i \leq y_{jj}^i \quad \forall (j, k) \in E \quad (6)$$

$$y_{jk}^i \leq y_{kk}^i \quad \forall (j, k) \in E \quad (7)$$

$$y_{jk}^i \geq y_{jj}^i + y_{kk}^i - 1 \quad \forall (j, k) \in E \quad (8)$$

**Abstract fact selection constraint** Limit the number of abstract facts selected to  $M$ :

$$\sum_i y_{jj}^i \leq M \quad \forall j \in F_A \quad (9)$$

### 3.3. Adopting Differentiable Blackbox Combinatorial Optimisation Solvers

The subgraph selection problem can be written in the following ILP canonical form:

$$\min_{y^i \in \{0,1\}} -(W^i \cdot y^i) \quad \text{subject to} \quad Ay^i \leq b, \quad (10)$$

where  $A = [a_1, \dots, a_m] \in \mathbb{R}^{m \times n}$  is the matrix of constraint coefficients and  $b \in \mathbb{R}^m$  is the bias term. The output of the solver  $g(W^i)$  returns the  $\text{argmin}_{y^i \in \{0,1\}}$  of the integer problem.

Differentiable Combinatorial Optimisation Solver (Pogančić et al., 2019) (DBCS) assumes that  $A$ ,  $b$  are constant and the task is to find the  $dL/dy^i$  given global loss function  $L$  with respect to solver output  $y^i$  at a given point  $\hat{y}^i = g(\hat{W}^i)$ . However, a small change in  $W$  is typically not going to change the optimal ILP solution resulting in the true gradient being zero. DBCS provides a theoretical framework that can potentially make any black box combinatorial solver differentiable. In this paper, we adopt the proposed algorithm as following to design the forward/backward pass:

Here the hyper-parameter  $\lambda_{dbcs} > 0$  controls the trade-off between *informativeness of the gradient* and *faithfulness to the original function*.

---

#### Algorithm 1: Forward and Backward for DBCS

---

**Function** ForwardPass ( $\hat{W}^i$ ):

```

 $\hat{y}^i = g(\hat{W})$ 
save  $\hat{W}^i$  and  $\hat{y}^i$  for backward pass
return  $\hat{y}^i$ ;

```

**Function** BackwardPass ( $\frac{dL}{dy^i}(\hat{y}^i)$ ,  $\lambda_{dbcs}$ ):

```

load  $\hat{W}^i$  and  $\hat{y}^i$  for backward pass
 $\hat{W}^i = \hat{W}^i + \lambda_{dbcs} \frac{dL}{dy^i}(\hat{y}^i)$ 
 $y_{\lambda_{dbcs}}^i = g(\hat{W}^i)$ 
return  $-\frac{1}{\lambda} [\hat{y}^i - y_{\lambda_{dbcs}}^i]$ 

```

---

### 3.4. Answer and Explanation Selection

The solved adjacency variable  $\hat{y}^i$  represents the selected edges for each candidate answer choice  $c_i$ . Not all datasets provide gold explanations. Moreover, even when the gold explanations are available, they are only available for the correct answer with no explanations for the *wrong* answer.

In order to tackle these shortcomings and ensure end-to-end differentiability, we use the softmax ( $\sigma$ ) of the objective score ( $\hat{W}^i \cdot \hat{y}^i$ ) as the probability score for each candidate answer.

We multiply each objective score  $\hat{W}^i \cdot \hat{y}^i$  value by the temperature hyperparameter ( $T$ ) to obtain soft probability distributions  $\gamma^i$  (where  $\gamma^i = (\hat{W}^i \cdot \hat{y}^i) \cdot T$ ). The aim is for the correct answer  $c_{ans}$  to have the highest probability.

In order to achieve this, we use the cross entropy loss  $l_c$  as follows to calculate the answer selection loss  $\mathcal{L}_{ans}$  as follows:

$$\mathcal{L}_{ans} = l_c(\sigma(\gamma^1, \gamma^2, \dots, \gamma^n), c_{ans}) \quad (11)$$

If gold explanations are available, we complement  $\mathcal{L}_{ans}$  with explanation loss  $\mathcal{L}_{exp}$ . We employ binary cross entropy loss  $l_b$  between the selected explanatory facts and gold explanatory facts  $F_{exp}$  for the explanatory loss as follows:

$$\mathcal{L}_{exp} = l_b(\hat{y}^{ans}[f_1, f_2, \dots, f_k], F_{exp}) \quad (12)$$

We calculate the total loss ( $\mathcal{L}$ ) as weighted by hyperparameters  $\lambda_{ans}$ ,  $\lambda_{exp}$  as follows:

$$\mathcal{L} = \lambda_{ans} \mathcal{L}_{ans} + \lambda_{exp} \mathcal{L}_{exp} \quad (13)$$

## 4. Empirical Evaluation

### 4.1. Answer and Explanation Selection

We use the WorldTree corpus (Xie et al., 2020) for the evaluation of explanation and answer selection. The 4,400 questions and explanations in the



Model	Explanation Selection ( <i>dev</i> )					Faithfulness	Answer Selection ( <i>test</i> )
	Precision		Explanatory Consistency				
	@2	@1	@3	@2	@1		
<b>Baselines</b>							
BERT <sub>Base</sub>	-	-	-	-	-	-	45.43
BERT <sub>Large</sub>	-	-	-	-	-	-	49.63
Fact Retrieval (FR) Only	30.19	38.49	21.42	15.69	11.64	-	-
BERT <sub>Base</sub> + FR	-	-	-	-	-	52.65	58.06
BERT <sub>Large</sub> + FR	-	-	-	-	-	51.23	59.32
ExplanationLP	40.41	51.99	29.04	14.14	11.79	71.11	62.57
Diff-Explainer	41.91	56.77	39.04	20.64	17.01	72.22	71.48
<b>Diff-Comb Explainer</b>							
- Answer selection only	45.75	61.01	<b>49.04</b>	29.99	18.88	73.37	72.04
- Answer and explanation selection	<b>47.57</b>	<b>63.23</b>	43.33	<b>33.36</b>	<b>20.71</b>	<b>74.47</b>	<b>73.46</b>

Table 1: Comparison of explanation and answer selection of *Diff-Comb Explainer* against other baselines. Explanation Selection was carried out on the *dev* set as the *test* explanation was not public available.

WorldTree corpus are split into three different subsets: *train-set*, *dev-set* and *test-set*. We use the *dev-set* to assess the explainability performance since the explanations for *test-set* are not publicly available. The background knowledge consists of 5000 abstract facts from the WorldTree table store (WTree) (Xie et al., 2020) and over 100,000 *is-a* grounding facts from ConceptNet (Speer et al., 2017).

**Baselines:** We use the following baselines to compare against our approach for the WorldTree corpus:

1. **BERT<sub>Base</sub> and BERT<sub>Large</sub>** (Devlin et al., 2019): To use BERT for this task, we concatenate every hypothesis with  $k$  retrieved facts, using the separator token [SEP]. We use the Hugging-Face (Wolf et al., 2019) implementation of *Bert-ForSequenceClassification*, taking the prediction with the highest probability for the positive class as the correct answer. Notice that *Diff-Comb Explainer* adopts the same neural architecture for sentence representation and inference.
2. **ExplanationLP:** Non-differentiable version of ExplanationLP. Using the constraints stated in Section 3, we fine-tune the  $\theta$  parameters using Bayesian optimization and frozen STans representations following (Thayaparan et al., 2021). This baseline aims to evaluate the impact of end-to-end fine-tuning over the non-differentiable solver.
3. **Diff-Explainer:** *Diff-Explainer* has already exhibited better performance over

existing ILP-based approaches including TableLP (Khashabi et al., 2016), TupleLP (Khot et al., 2017) and graph-based neural approach PathNet (Kundu et al., 2019) and represents the current state-of-the-art combinatorial solver in the field. To allows for a direct comparison with our approach, we use the ExplanationLP constraints with *Diff-Explainer*. We adopt the same set of hyperparameters and knowledge base reported in Thayaparan et al. (2022).

**Evaluation Metrics.** The answer selection is evaluated using accuracy. For evaluation of explanation selection, we use Precision@ $K$ . In addition to Precision@ $K$ , we introduce two new metrics to evaluate the truthfulness of the answer selection to the underlying inference. The metrics are as follows:

**Faithfulness** The aim of faithfulness is to measure the extent to which correct answer predictions are indeed derived from correct explanations and, vice-versa, wrong answer predictions are derived from wrong explanations. Given the set of correct answers  $A_{Q_c}$ , wrong answers  $A_{Q_w}$ , answers with at least one correctly retrieved explanation sentence  $A_{Q_1}$  and set of questions where no correct explanation fact is retrieved  $A_{Q_0}$ , the faithfulness score is defined as follows:

$$\frac{|A_{Q_w} \cap A_{Q_0}| + |A_{Q_c} \cap A_{Q_1}|}{|A_{Q_c} \cup A_{Q_w}|} \quad (14)$$

A higher faithfulness score implies, therefore, that the underlying inference performed by the

model is more in line with the final answer prediction.

**Explanation Consistency@K** As similar questions require similar underlying inference (Valentino et al., 2021, 2022), we expect the explanations to reflect the similarity between test instances. Specifically, given a test question  $Q_t$  and the explanation constructed by the model  $E_t$ , we select the set of Questions  $Q_t^s = \{Q_t^1, Q_t^2, \dots\}$  that have at least  $K$  gold explanatory facts overlapping with the gold explanation for  $Q_t$ ,  $E_t^s = \{e_t^1, e_t^2, \dots\}$ . Given this premise, we measure the explanation consistency@K as follows:

$$\frac{\sum_{e_t^i \in E_t^s} |e_t^i \cap E_t|}{\sum_{e_t^i \in E_t^s} |e_t^i|} \quad (15)$$

Specifically, explanation consistency measures the percentage of overlapping explanatory facts retrieved by the models between hypotheses that share at least  $K$  similar gold explanations. Higher explanatory consistency, therefore, implies that the reasoning across similar questions tends to produce similar explanations.

## 4.2. Results

Table 1 illustrates the explanation and answer selection performance of *Diff-Comb* Explainer and the baselines. We report scores for *Diff-Comb* Explainer trained for only the answer and optimised jointly for answer and explanation selection. Since BERT does not provide explanations, we use facts retrieved from the fact retrieval for the best  $k$  configuration ( $k = 3$ ) as explanations. We also report the scores for BERT without explanations.

We draw the following conclusions from the results obtained in Table 1 (The performance increase here are expressed in absolute terms):

(1) *Diff-Comb* Explainer improves answer selection performance over the non-differentiable solver by 9.47% when optimised only on answer selection and 10.89% when optimised on answer and explanation selection. This observation highlights the impact of the end-to-end differentiable framework. We can also observe that strong supervision on explanation selection yields better performance than weak supervision on answer selection only.

(2) *Diff-Comb* Explainer outperforms the best BERT model by 14.14% on answer selection. This increase in performance demonstrates that integrating constraints with Transformer-based encoders can lead to better NLI performance.

(3) *Diff-Comb* Explainer outperform the best *Diff-Explainer* configuration (answer and explanation selection) by 0.56% even in the weak supervision setting (answer only optimization). We also outperform *Diff-Explainer* by 1.98% in the best setting.

(4) *Diff-Comb* Explainer shows superior performance in selecting relevant explanations over existing combinatorial solvers. *Diff-Comb* Explainer outperforms the non-differentiable solver at Precision@K by 8.41% ( $k = 1$ ) and 6.05% ( $k = 2$ ). Moreover, the proposed approach outperforms *Diff-Explainer* by 3.63% ( $k = 1$ ) and 4.55% ( $k = 2$ ). This demonstrates the impact of preserving the original ILP formulation on downstream inference. Finally, the improvement of Precision@K over the Fact Retrieval only (demonstrated with BERT + FR) by 16.98% ( $k = 1$ ) and 24.74% ( $k = 2$ ) underlines the robustness of our approach to noise propagated by the upstream fact retrieval.

(5) *Diff-Comb* Explainer exhibits higher Explanation Consistency over the other solvers. This performance shows that the optimization model is learning and applying consistent inference across similar instances. We also outperform the fact retrieval model based on Transformers and fine-tuned on gold explanations.

(6) Answer prediction by *black-box* models like BERT do not reflect the explanation provided. This fact is indicated by the low Faithfulness score obtained by both BERT<sub>Base</sub>/BERT<sub>Large</sub>. In contrast, the high constraint-based solver’s Faithfulness scores emphasise how the underlying inference reflects on the final prediction. In particular, our approach performs better than the non-differentiable models and *Diff-Explainer*.

In summary, despite the fact that *Diff-Explainer* and *Diff-Comb* Explainer adopts the same underlying set of constraints, the results demonstrate that our model yields better performance, indicating that accurate predictions generated by ILP solvers reflect in better inference performance than sub-optimal counterparts.

## 4.3. Comparing Answer Selection with ARC Baselines

Table 2 presents a comparison of publicly reported baselines on the ARC Challenge-Corpus (Clark et al., 2018) and our approach. These questions have proven to be challenging to answer for other LP-based NLI and neural approaches.

In Table 2, to provide a fair comparison, we list the models that have been trained *only* on the ARC corpus. Here, the explainability column indicates if the model delivers an explanation for the predicted answer. A subset of the models produces evidence for the answer but remains intrinsically black-boxes. These models have been marked as *Partial*. As illustrated in the Table 2, *Diff-Comb* Explainer outperforms the best non-differentiable ILP solver model (ExplanationLP) by 2.8%. We also outperform a transformer-based model (AutoRoc) by 1.97%. While our improvement over

Model	Explainable	Accuracy
BERT <i>Large</i>	No	35.11
IR Solver (Clark et al., 2016)	Yes	20.26
TupleLP (Khot et al., 2017)	Yes	23.83
TableLP (Khashabi et al., 2016)	Yes	26.97
ExplanationLP (Thayaparan et al., 2021)	Yes	40.21
DGEM (Clark et al., 2016)	Partial	27.11
KG <sup>2</sup> (Zhang et al., 2018)	Partial	31.70
ET-RR (Ni et al., 2019)	Partial	36.61
Unsupervised AHE (Yadav et al., 2019a)	Partial	33.87
Supervised AHE (Yadav et al., 2019a)	Partial	34.47
AutoRocc (Yadav et al., 2019b)	Partial	41.24
<i>Diff-Explainer</i> (ExplanationLP) (Thayaparan et al., 2022)	Yes	<b>42.95</b>
<i>Diff-Comb Explainer</i> (ExplanationLP)	Yes	<b>43.21</b>

Table 2: ARC challenge scores compared with other Fully or Partially explainable approaches trained *only* on the ARC dataset.

*Diff-Explainer* is small, we still demonstrate performance improvements for answer selection.

#### 4.4. Qualitative Analysis

Table 3 illustrates some of the explanations constructed by ExplanationLP, *Diff-Explainer* and *Diff-Comb Explainer*. Both explanations and answer predictions in Question (1) are entirely correct for our model. In this example, both ExplanationLP and *Diff-Explainer* have failed to retrieve any correct explanations or predict the correct answer. Both the approaches are distracted by the strong lexical overlaps with the wrong answer.

In Question (2) *Diff-Comb Explainer* produces the correct answer with at least one explanation sentence marked as correct. Here, *Diff-Explainer* provides the correct answer prediction with both the retrieved facts marked as incorrect showing a lower faithfulness compared to our proposed approach.

In Question (3) both our model and *Diff-Explainer* provide the correct answer supported by an incorrect explanation. While, the aforementioned qualitative (Question 1 and 2) and quantitative measures (Explanatory Consistency@ $K$ , Faithfulness) indicate the superiority of *Diff-Comb Explainer* in generating supporting explanations, we still find cases where incorrect inference can lead to the correct answer. In most of the cases, we found that the inference is distracted by strong lexical overlaps with the question.

In general, however, we can conclude that *Diff-*

*Comb Explainer* is more robust when compared to *Diff-Explainer* and existing ILP models.

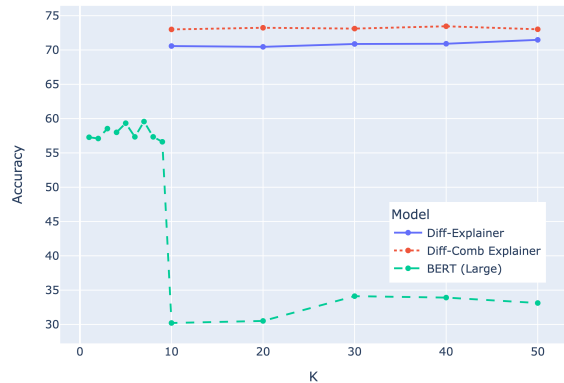


Figure 3: Comparison of accuracy for different number of retrieved facts.

#### 4.5. Knowledge aggregation with increasing distractors

One of the key characteristics identified by Thayaparan et al. (2022) is the robustness of *Diff-Explainer* to distracting noise. In order to evaluate if our model also exhibits the same characteristics, we tested our model with an increasing number of retrieved facts  $k$  from the KB and plot the resulting answer accuracy in Figure 3.

As illustrated in the Figure, similar to *Diff-Explainer*, our approach performance remains stable with increasing distractors. We also continue to outperform *Diff-Explainer* across all sets of  $k$ .

BERT performance, on the other side, drops drastically with increasing distractors. This phenomenon is in line with existing findings (Thayaparan et al., 2022; Yadav et al., 2019b). We hypothesise that with increasing distractors, BERT overfits quickly with spurious inference correlation. On the other hand, our approach circumvents this problem with the inductive bias provided by the constraint optimization layer.

## 5. Conclusion

This paper proposed a novel framework for encoding explicit and controllable assumptions as part of an end-to-end learning framework for explanation-based NLI using Differentiable Blackbox Combinatorial Solvers (Pogančić et al., 2019). We empirically demonstrated improved answer and explanation selection performance compared with the existing differentiable constraint-based solvers (Thayaparan et al., 2022). Moreover, we demonstrated performance gain and increased robustness to noise when combining precise constraints with transformer-based architectures. In this paper, we

---

**Question (1):** Which measurement is best expressed in light-years?: **Correct Answer:** the distance between stars in the Milky Way.

ExplanationLP

**Answer:** the time it takes for planets to complete their orbits. **Explanations:** (i) a complete revolution; orbit of a planet around its star takes 1; one planetary year, (ii) a light-year is used for describing long distances

Diff-Explainer

**Answer:** the time it takes for planets to complete their orbits. **Explanations:** (i) a light-year is used for describing long distances, (ii) light year is a measure of the distance light travels in one year

Diff-Comb Explainer

**Answer:** the distance between stars in the Milky Way. **Explanations:** (i) light years are a astronomy unit used for measuring length, (ii) stars are located light years apart from each other

---

**Question (2):** Which type of precipitation consists of frozen rain drops?: **Correct Answer:** sleet.

ExplanationLP

**Answer:** snow. **Explanations:** (i) precipitation is when snow fall from clouds to the Earth, (ii) snow falls

Diff-Explainer

**Answer:** sleet. **Explanations:** (i) snow falls, (ii) precipitation is when water falls from the sky

Diff-Comb Explainer

**Answer:** sleet. **Explanations:** (i) sleet is when raindrops freeze as they fall, (ii) sleet is made of ice

---

**Question (3):** Most of the mass of the atom consists of?: **Correct Answer:** protons and neutrons.

ExplanationLP

**Answer:** neutrons and electrons. **Explanations:** (i) neutrons have more mass than an electron, (ii) neutrons have more mass than an electron

Diff-Explainer

**Answer:** protons and neutrons. **Explanations:** (i) the atomic mass is made of the number of protons and neutrons, (ii) precipitation is when water falls from the sky

Diff-Comb Explainer

**Answer:** protons and neutrons. **Explanations:** (i) the atomic mass is made of the number of protons and neutrons, (ii) precipitation is when water falls from the sky

---

Table 3: Example of predicted answers and explanations obtained from our model with different combinatorial solvers.

adopted a specific set of constraints to ensure fairness in the evaluation, but, in principle, it is possible to extend the proposed architecture with different inference constraints.

*Diff-Comb Explainer* improves upon recent work by [Thayaparan et al. \(2022\)](#) and investigates the combination of symbolic knowledge (expressed via constraints) with neural representations. We hope this work will encourage researchers to encode different domain-specific priors, leading to more robust, transparent and controllable neuro-symbolic inference models for NLP.

## 6. Limitations

The ILP formulation is NP-complete. Therefore, as the number of variables in the problem increases the computational complexity also increases exponentially. However, this challenge can be alleviated with strong solvers and the relaxation of some constraints. The performance of the

proposed approach is not yet comparable with the scores achieved by Large Language Models (LLMs) ([Khashabi et al., 2020](#)). However, it should be noted that, unlike LLMs, the proposed approach was trained only on the dataset it was tested on and exhibited higher performance compared to the same Transformer encoders integrated in the architecture. These types of neuro-symbolic models can potentially lead to smaller, domain-specific models that require little training data for more efficient and transparent natural language inference.

## Acknowledgements

This work was partially funded by the Swiss National Science Foundation (SNSF) project NeuMath (200021\_204617), by the EPSRC grant EP/T026995/1 entitled “EnnCore: End-to-End Conceptual Guarding of Neural Architectures” under Security for all in an AI enabled society, by the CRUK National Biomarker Centre, and supported



by the Manchester Experimental Cancer Medicine Centre. The authors would like to thank the anonymous reviewers and editors of LREC-COLING for their constructive feedback. Additionally, we would like to thank the Shared Facilities of the University of Manchester and Idiap Research Institute for providing the infrastructure to run our experiments.

## 7. Bibliographical References

- Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. 2019. [Differentiable convex optimization layers](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter D Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*, pages 2580–2586. Citeseer.
- Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3882–3890.
- 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, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. [Billion-scale similarity search with gpus](#). *CoRR*, abs/1702.08734.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. Question answering via integer programming over semi-structured knowledge. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1145–1152.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. *arXiv preprint arXiv:1704.05572*.
- Souvik Kundu, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2019. [Exploiting explicit paths for multi-hop reading comprehension](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2737–2747, Florence, Italy. Association for Computational Linguistics.
- L. Lovász and A. Schrijver. 1991. Cones of matrices and set-functions and 0-1 optimization. *SIAM JOURNAL ON OPTIMIZATION*, 1:166–190.
- Jianmo Ni, Chenguang Zhu, Weizhu Chen, and Julian McAuley. 2019. Learning to attend on essential terms: An enhanced retriever-reader model for open-domain question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 335–344.
- Anselm Paulus, Michal Rolínek, Vít Musil, Brandon Amos, and Georg Martius. 2021. Combopnet: Fit the right np-hard problem by learning integer programming constraints. In *International Conference on Machine Learning*, pages 8443–8453. PMLR.
- Marin Vlastelica Pogančić, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolinek. 2019. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*.

- Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. Prover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 122–136.
- Swarnadeep Saha, Prateek Yadav, and Mohit Bansal. 2021. multiprover: Generating multiple proofs for improved interpretability in rule reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3662–3677.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.
- Mokanarangan Thayaparan, Marco Valentino, Deborah Ferreira, Julia Rozanova, and André Freitas. 2022. [Diff-Explainer: Differentiable Convex Optimization for Explainable Multi-hop Inference](#). *Transactions of the Association for Computational Linguistics*, 10:1103–1a119.
- Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A survey on explainability in machine reading comprehension. *arXiv preprint arXiv:2010.00389*.
- Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Explainable inference over grounding-abstract chains for science questions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1–12.
- Marco Valentino, Mokanarangan Thayaparan, Deborah Ferreira, and André Freitas. 2022. Hybrid autoregressive inference for scalable multi-hop explanation regeneration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11403–11411.
- Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021. Unification-based reconstruction of multi-hop explanations for science questions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 200–211.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Hugging-face’s transformers: State-of-the-art natural language processing](#). *CoRR*, abs/1910.03771.
- Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. Worldtree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5456–5473.
- Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019a. Alignment over heterogeneous embeddings for question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2681–2691.
- Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019b. [Quick and \(not so\) dirty: Unsupervised selection of justification sentences for multi-hop question answering](#). 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 2578–2589, Hong Kong, China. Association for Computational Linguistics.
- Yuyu Zhang, Hanjun Dai, Kamil Toraman, and Le Song. 2018. Kg<sup>2</sup>: Learning to reason science exam questions with contextual knowledge graph embeddings. *arXiv preprint arXiv:1805.12393*.