Distilling Structured Knowledge for Text-Based Relational Reasoning

Jin Dong [1,2], **Marc-Antoine Rondeau** [3], **William L. Hamilton** [1,2,4]
[1] School of Computer Science, McGill University, Canada
[2] Quebec AI Institute (Mila), Canada
[3] Microsoft Research, Montreal, Canada
[4] Canada CIFAR AI Chair
jin.dong@mail.mcgill.ca, marcantoine.rondeau@microsoft.com
wlh@cs.mcgill.ca

Abstract

There is an increasing interest in developing text-based relational reasoning systems, which are capable of systematically reasoning about the relationships between entities mentioned in a text. However, there remains a substantial performance gap between NLP models for relational reasoning and models based on graph neural networks (GNNs), which have access to an underlying symbolic representation of the text. In this work, we investigate how the structured knowledge of a GNN can be distilled into various NLP models in order to improve their performance. We first pre-train a GNN on a reasoning task using structured inputs and then incorporate its knowledge into an NLP model (e.g., an LSTM) via *knowledge distillation*. To overcome the difficulty of cross-modal knowledge transfer, we also employ a *contrastive learning* based module to align the latent representations of NLP models and the GNN. We test our approach with two state-of-the-art NLP models on 12 different inductive reasoning datasets from the CLUTRR benchmark and obtain significant improvements.

1 Introduction

The task of text-based relational reasoning—where an agent must infer and compose relations between entities based on a passage of text—has received increasing attention in natural language processing (NLP) (Andreas, 2019). This task has been especially prominent in the context of systematic generalization in NLP, with synthetic datasets, such as CLUTTR and SCAN, being used to probe the ability of NLP models to reason in a systematic and logical way (Lake and Baroni, 2018; Sinha et al., 2019). More generally, these investigations dovetail with the rising prominence of relational reasoning throughout machine learning and cognitive science (Alexander et al., 2016; Battaglia et al., 2018; Hamilton et al., 2017).

However, despite the increased attention and research on text-based relational reasoning, serious challenges remain. Perhaps one of the biggest challenges is the persistent gap between the performance that can be achieved using NLP models and the performance of structured models—such as graph neural networks (GNNs)—which perform relational reasoning based on structured or symbolic inputs. This gap was made particularly evident in the the CLUTRR benchmark. CLUTRR includes relational reasoning problems that can be posed both in textual or symbolic form, and preliminary investigations using CLUTRR show that GNN-based models—which leverage the structured symbolic input—are able to achieve higher accuracy, better generalization, and are more robust than purely text-based systems (Sinha et al., 2019).

In this work, we investigate one potential avenue to close this gap. We design an approach to *distill* the structured knowledge learned by a GNN—which has access to the underlying symbolic representation of a reasoning problem—into an NLP model. Our goal is to do this knowledge distillation (Hinton et al., 2015) only during training so that the NLP model can achieve higher performance at test time, when only unstructured textual inputs are available. Due to the challenges inherent in cross-model knowledge distillation (Tian et al., 2020), we design an approach that combines both a KL-based distillation objective (Hinton et al., 2015) and a contrastive estimation loss (Hjelm et al., 2019), which aims to maximize the mutual information between the latent states of text-based NLP and graph-based GNN models.

Empirical results on 12 different datasets from the CLUTRR benchmark suite highlight the potential utility of this approach. We find that extending two state-of-the-art NLP models using our structured distillation approach significantly improves performance and that the gains are espe-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6782–6791,
November 16–20, 2020. ©2020 Association for Computational Linguistics

cially prominent in the context of noisy input data, on which we obtain an 13.6% relative improvement on accuracy.[1]

2 Related Work

Our work is closely related to recent research on machine reading comprehension (MRC), question answering (QA), and relational reasoning in NLP.

Prominent examples of large-scale QA benchmarks include datasets such as SQuAD (Rajpurkar et al., 2016) and TriviaQA (Joshi et al., 2017). However, these traditional datasets do not consider the reasoning aspect of MRC and only target extractive QA tasks. Usually, these tasks only require extracting a single fact (or span of text) and do not necessitate complex relational reasoning.

To address this shortcoming, there has been a surge of work tackling the relational reasoning and systematic generalization. Johnson et al. (2017) first proposed the CLEVR dataset that focuses on the relational reasoning aspect of visual question answering (VQA). Similarly, Sinha et al. (2019) released CLUTRR involving both text and graphs. These relational reasoning datasets also share inspirations with *multi-hop* QA, such as HotPotQA (Yang et al., 2018). Generally, the key distinction in the multi-hop setting is that an agent must reason about the relationship between multiple entities in order to answer a query.

Finally, the development of these relational reasoning datasets has also dovetailed with an increasing interest in combining NLP models with graph neural networks (GNNs) (Hamilton et al., 2017). This includes the use of GNNs for processing syntax trees (Marcheggiani and Titov, 2017), as well as the use of GNNs for reasoning over entity graphs extracted from text (Fang et al., 2019).

3 Task and Dataset

We use the CLUTRR benchmark suite as a testbed for our investigations (Sinha et al., 2019). CLUTRR is a relational reasoning dataset that requires an agent to infer family relationships between different characters in a passage of text. Importantly, the dataset was constructed in a *semi-synthetic* fashion, which facilitates a principled investigation of text-based relational reasoning. Every question-answer pair in CLUTRR was generated based on underlying family graph structure, where crowd workers were instructed to paraphrase natural language stories from a given set of family relations. To answer a question in the CLUTRR dataset, the model must infer the family relationship between a pair of entities, whose relationship is not explicitly mentioned. Doing so requires extracting the family relationships mentioned in the text and deducing the relationship between the query entities through inductive reasoning (e.g., learning that *a parent of a parent is a grandparent*.

A key element of CLUTRR is that it provides both text representations and the underlying family graphs used to generate the questions. This allowed Sinha et al. (2019) to compare the performance of NLP models, which use only text, with GNN-based models, which reason upon the underlying graph structure, and their analysis revealed a substantial gap in performance between the NLP and GNN models—a gap which we seek to address here.

Moreover, following Sinha et al. (2019), the semi-synthetic nature of CLUTRR allows us to evaluate performance in different settings based on the structure of the underlying family graph and the difficulty of the query, including evaluating performance on queries that require a varying number of steps of reasoning and family graphs that include different types of noisy facts (i.e., distractors).

4 Methodology

We now describe our approach for *structured distillation*, which involves improving the performance of an NLP model by distilling structured knowledge from a GNN (Fig. 1).

Graph encoder and text encoder. Our base model architectures follow Sinha et al. (2019), with minor improvements. As shown in Fig. 1, we implement both a *graph encoder*, which generates a vector embedding $\mathbf{p}_{\text{graph}}$ based on the input family graph, as well as a *text encoder*, which generates a vector embedding $\mathbf{p}_{\text{text}}$ of the input text. We use a variant of the graph isomorphism network (GIN) architecture Xu et al. (2019) as our graph encoder, since we found this model to outperform the GNN from Sinha et al. (2019). For our text encoders, we experiment with the two top-performing NLP models from Sinha et al. (2019): (1) a variation of an LSTM model with attention (Bahdanau et al., 2015) and (2) an adapted version of the MAC architecture (Hudson and Manning, 2018). See Appendix A for details on the model architectures.

[1] Code and data can be found at `https://github.com/djdongjin/gnnlogic`.

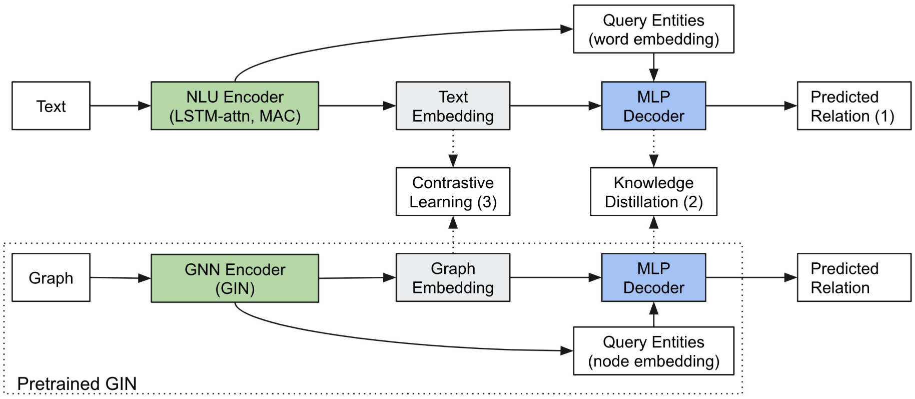

Figure 1: Model architecture with both knowledge distillation and contrastive learning. The supervised signal is produced in (1) with cross-entropy loss. We first pretrain a GIN model which is used later for knowledge distillation and contrastive learning. Knowledge distillation module (2) aligns the predictions made by a GIN model and an NLP model, via KL-divergence loss (Eq. 1). The contrastive learning module (3) aligns the latent space of these two models via a MI-based contrastive loss (Eq. 2).

Integration with knowledge distillation. We utilize knowledge distillation as a surrogate for the structured knowledge transfer from GNNs to NLP models. We take the text encoder as the student and a pretrained GNN as the teacher. After generating the representations of the paragraph $\mathbf{p}_{\text{text}}$ and the question entities $(\mathbf{h}^{(m)}, \mathbf{h}^{(n)})$, the text encoder sends the concatenation of these embeddings to an MLP decoder to obtain the logits $\mathbf{z}_{\text{text}}$. Similarly, a pretrained GNN can produce logits $\mathbf{z}_{\text{graph}}$ from a given underlying graph. We feed the two logits into a KL-based distillation term:

$$\mathcal{L}_{\text{KD}} = T^2 \cdot \text{KL}\left(\sigma\left(\frac{\mathbf{z}_{\text{text}}}{T}\right) \middle| \sigma\left(\frac{\mathbf{z}_{\text{graph}}}{T}\right)\right), \quad (1)$$

where σ is the softmax function and T is the temperature hyperparameter of softmax.

Integration with contrastive estimation. Although knowledge distillation enables NLP models to learn directly from the prediction of GNNs, there is no regularization between their latent representations. We mitigate this by using a mutual information (MI) based contrastive learning method to maximize the MI between graph representations from GNNs and paragraph representations from NLP models. Under our setting, we pair the text representation $\mathbf{p}_{\text{text}}$ and the graph representation $\mathbf{p}_{\text{graph}}$ of the same example as *positive pairs*, and take other graph representations in the same batch as *negative pairs*. Then, following Hjelm et al. (2019), we use a Jensen-Shannon estimator to compute the MI, resulting in the contrastive objective:

$$\begin{aligned}\mathcal{L}_{\text{MI}} = -\widehat{\mathcal{I}}(\mathbf{p}_{\text{text}}, \mathbf{p}_{\text{graph}}) = \\ -\mathbb{E}_{\mathbb{P}(p,g|c=1)}\left[-\text{sp}(-T(\mathbf{p}_{\text{text}}, \mathbf{p}_{\text{graph}}))\right] \\ + \mathbb{E}_{\mathbb{P}(p,g|c=0)}\left[\text{sp}(T(\mathbf{p}_{\text{text}}, \mathbf{p}_{\text{graph}}))\right], \quad (2)\end{aligned}$$

where $\mathbb{P}(p, g|c = 1)$ and $\mathbb{P}(p, g|c = 0)$ indicate the conditional probability of whether the given paragraph p and graph g correspond to the same question-answering example ($c = 1$) or not ($c = 0$). We use sp to denote the softplus function, and we use T to denote MLP that is trained to discriminate between positive and negative pairs.

Note that the contrastive loss in Eq. 2 is also composable with both the supervised cross-entropy loss (from the original CLUTRR task) and knowledge distillation loss (Eq. 1).

5 Experiments

Our key experimental question is whether an NLP model can be improved by distilling structured knowledge from a GNN. We investigate this question using the GNN and NLP models defined in the previous section, and we follow the experimental protocol from Sinha et al. (2019). We investigate if and how structured distillation can improve generalization and robustness. In all experiments, the NLP models only have access to information from the GNN during training. Appendix A contains detailed hyperparameter information.

Impact on generalization. We first test on the CLUTRR benchmark tasks where the model must generalize to reasoning problems that require more steps of reasoning than those seen during training.

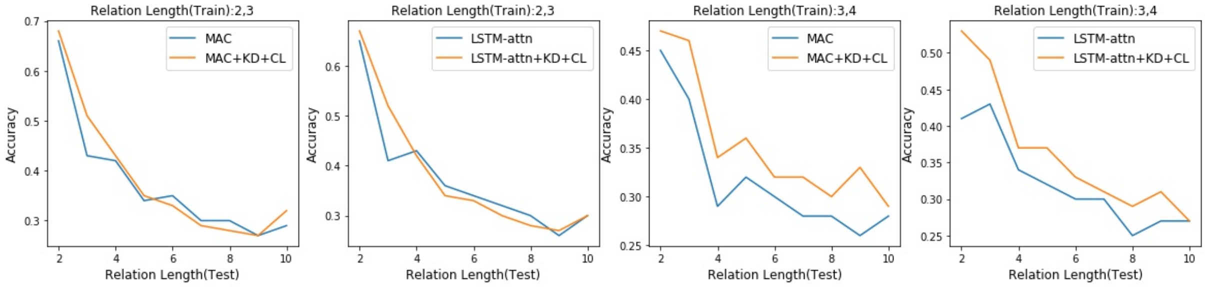

Figure 2: Accuracy on test sets with relation length of 2-10. KD denotes knowledge distillation; CL denotes the MI-based contrastive learning. All results are averaged over 5 runs with different random seeds. The maximum standard deviation is less than 0.05. Detailed accuracy values can be found in Appendix B.

Dataset		Model			
Relation length	Distractor	MAC	MAC+KD+CL	LSTM-attn	LSTM-attn+KD+CL
2,3	Clean	0.54	0.59	0.54	**0.60**
2,3	Supporting	0.54	0.57	0.45	**0.59**
2,3	Irrelevant	0.47	**0.52**	0.40	**0.52**
2,3	Disconnected	0.40	**0.45**	0.41	0.42

Table 1: Accuracy on test sets with different distractors. All results are averaged over 5 runs with different random seeds. The maximum standard deviation is less than 0.05.

Fig. 2 shows the results when we set the number of training reasoning steps to be $(2, 3)$ and $3, 4$, and where the test examples require between 2 and 10 reasoning steps. Both of the two NLP model obtain higher average performance on test sets with our proposed method. Interestingly, however, the positive impact of structured distillation is most apparent when training on examples with longer reasoning paths. Appendix B contains results from training on other reasoning path lengths, which are consistent with the trends in Fig. 2.

Impact on robustness. We next investigated the impact of structured generalization on how robust the NLP models are with respect to noise. Following Sinha et al. (2019), we examined settings where different types of noise facts are added into the CLUTRR reasoning problems. Tab. 1 shows the results where we train and test on reasoning problems with different types of noise. Here, we see that structured distillation consistently and substantially improves performance of both NLP models, providing an average 13.6% relative improvement on accuracy. The results also shows the distillation and contrastive estimation based on GNNs help NLP models ignore noise. However, their fundamental architecture difference limits the extent to which NLP models can learn from GNNs. Appendix C contains additional results, where the train and test sets do not have the same noise added and which further support this trend.

Ablation analysis. We found that both knowledge distillation and contrastive estimation (Eq. 1-2) losses are necessary in tandem to obtain the benefits of structured generalization. We found no significant gains when adding one loss alone. Appendix D contains detailed results on these ablations.

6 Discussion and Conclusion

Our structured distillation approach achieves promising results. Most prominently, the structured distillation approach significantly improved the performance of the NLP models in settings where noisy facts were added to the CLUTRR reasoning problems. The GNN-based models are particularly strong in this setting (see Appendix C), and this suggests that transferring knowledge about the relevancy of facts from structured to unstructured models may be a promising direction.

However, at the same time, the improvements for generalization were less substantial, indicating that some reasoning capacities are difficult to distill in this manner. Moreover, despite the improvements we observed, the performance of the NLP models is still substantially below the performance of the GNN teacher used for distillation (see Appendices B & C), highlighting that significant work that remains to close the gap between the reasoning performance of text-based and GNN-based models.

Acknowledgments

This research was funded in part by an academic grant from Microsoft Research, as well as a Canada CIFAR AI Chair, held by Prof. Hamilton at the Mila - Quebec AI Institute. We also thank Compute Canada for providing computational resources.

References

Patricia A Alexander, Sophie Jablansky, Lauren M Singer, and Denis Dumas. 2016. Relational reasoning: what we know and why it matters. *Policy insights from the behavioral and brain sciences*, 3(1):36–44.

Jacob Andreas. 2019. Measuring compositionality in representation learning. *7th International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *3th International Conference on Learning Representations*.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2019. Systematic generalization: what is required and can it be learned? *7th International Conference on Learning Representations*.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2019. Hierarchical graph network for multi-hop question answering. *arXiv preprint arXiv:1911.03631*.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin, September 2017*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. *7th International Conference on Learning Representations*.

Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations (ICLR)*.

Justin Johnson, Li Fei-Fei, Bharath Hariharan, C Lawrence Zitnick, Laurens Van Der Maaten, and Ross Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 1988–1997.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations, ICLR 2015*.

Brenden M Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *35th International Conference on Machine Learning, ICML 2018*.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuad: 100,000+ questions for machine comprehension of text. In *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 2383–2392.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. 2019. CLUTRR: A Diagnostic Benchmark for Inductive Reasoning from Text. 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 4505–4514, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive representation distillation. *8th International Conference on Learning Representations*.

Keyulu Xu, Stefanie Jegelka, Weihua Hu, and Jure Leskovec. 2019. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

A Hyperparameters

For all experiments in this section, we train the model for 50 epochs with a batch size of 100. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001.

In the encoder part, we use 100-dimensional word embeddings and train them from scratch for all NLP models. For LSTM-based models, we use a 2-layer bidirectional LSTM with 100 hidden units. For the MAC network, we use 6 MAC cell units (6 reasoning steps), and 0.2 dropout (Srivastava et al., 2014) on all updates in the three units to avoid overfitting. We use a two-layer MLP with 100 hidden units as the score function for all attention modules. For the GIN model, we use 2 GIN layers with 100-dimensional node embeddings and 20-dimensional edge embeddings. All node embeddings and edge embeddings are uniformly initialized.

In the decoder part, we use MLPs with the same architecture (2 layers, 200 hidden units) for all encoders. The inputs will be the concatenation of the graph representation and two question node representations if the encoder is GIN, or the concatenation of the paragraph representation and two word representations if the encoder is an NLP model.

All hyperparameters were tuned based on the validation accuracy. Full setups and hyperparameters can be found in the corresponding configuration files in our codebase after releasing.

For knowledge distillation, the temperature used to compute KL-divergence loss is 3.5. For contrastive learning, the negative sampling size is equal to the batch size (e.g. 100). The weighting hyperparameters for supervised cross-entropy loss, KL-divergence loss and MI maximization loss are chosen from $\{[0.1, 0.6, 0.3], [1, 1, 5]\}$.

B Full Results on Generalization

Tab. 2 shows all empirical results on datasets that have different relation lengths in training sets. we observe that our proposed method can improve the performance of vanilla NLP models in 7 out of 8 CLUTRR datasets. Another observation is that the NLP models still cannot learn the superb generalization ability of GNNs regardless of the difficulty of the tasks. The improvement of reasoning ability, measured by accuracy, is most significant when the training set and test set have the same reasoning length. This is not surprising as the generalization ability is a known issue in modern NLP models and is an ongoing research topic (Bahdanau et al., 2019; Andreas, 2019). However, the generalization is in parallel with our contribution that is to improve the reasoning ability of NLP models. We refer readers to (Bahdanau et al., 2019; Andreas, 2019) for a comprehensive understanding of current progress in generalization of NLP models.

C Full Result on Robustness

Tab. 3 shows results on the CLUTRR tasks with various. For each dataset, the training set contains a single type of noise, and we test on four test sets, each of which has one different type of distractor. Our augmented models via knowledge distillation (KD) and contrastive learning (CL) still outperform corresponding baselines by 3%-13%, depending on datasets and models. The MAC+KD+CL achieves the best accuracy on three out of four CLUTRR datasets, and LSTM-attn+KD+CL achieves the best on the left one. This shows that our method is able to improve the robustness of NLP models as well.

D Ablation Study on Contrastive Learning and Knowledge Distillation

We enable knowledge distillation and MI-based contrastive learning by weighing their corresponding losses as well as the supervised cross-entropy loss. The three of them can be treated as individual modules, each of which has different effectiveness. The cross-entropy loss enables a model to learn from supervised labels; the knowledge distillation loss enables a model to learn from soft targets produced by a teacher model (in our setting, a GIN); the contrastive learning loss enables a model to learn latent representations (embeddings) in an unsupervised manner.

Tab. 4 shows the ablation study among these three objectives. First we can observe that the best models trained with our method outperforms the vanilla MAC network by 3%-13%. Surprisingly, a MAC network trained with only soft signals produced by a GIN teacher can match the performance of a MAC network trained with supervised signals. If a MAC network is trained with both the supervised signal and soft signal, it outperforms the vanilla MAC network on 3 out of 4 CLUTRR datasets. When the MI-based contrastive learning loss is added, the MAC network performs the best on all the four datasets. These observations show that both knowledge distillation and contrastive learning are important for the model performance.

Model — Test relation length	2	3	4	5	6	7	8	9	10
Training relation length: 2, 3									
MAC	0.66	0.43	0.42	0.34	0.35	0.30	0.30	0.27	0.29
MAC+KD+CL	0.68	0.51	0.43	0.35	0.33	0.29	0.28	0.27	0.32
LSTM-attn	0.65	0.41	0.43	0.36	0.34	0.32	0.30	0.26	0.30
LSTM-attn+KD+CL	0.67	0.52	0.42	0.34	0.33	0.30	0.28	0.27	0.30
Training relation length: 3, 4									
MAC	0.45	0.40	0.29	0.32	0.30	0.28	0.28	0.26	0.28
MAC+KD+CL	0.47	0.46	0.34	0.36	0.32	0.32	0.30	0.33	0.29
LSTM-attn	0.41	0.43	0.34	0.32	0.30	0.30	0.25	0.27	0.27
LSTM-attn+KD+CL	0.53	0.49	0.37	0.37	0.33	0.31	0.29	0.31	0.27
Training relation length: 4, 5									
MAC	0.34	0.42	0.34	0.38	0.36	0.34	0.33	0.26	0.31
MAC+KD+CL	0.46	0.44	0.32	0.38	0.36	0.31	0.34	0.31	0.27
LSTM-attn	0.37	0.45	0.37	0.39	0.38	0.33	0.36	0.31	0.35
LSTM-attn+KD+CL	0.41	0.48	0.37	0.41	0.36	0.34	0.36	0.32	0.31
Training relation length: 5, 6									
MAC	0.42	0.38	0.39	0.38	0.38	0.38	0.39	0.36	0.38
MAC+KD+CL	0.43	0.37	0.35	0.34	0.35	0.34	0.35	0.34	0.32
LSTM-attn	0.36	0.36	0.36	0.37	0.37	0.38	0.36	0.35	0.37
LSTM-attn+KD+CL	0.37	0.36	0.40	0.37	0.38	0.41	0.40	0.37	0.39
Training relation length: 6, 7									
MAC	0.37	0.32	0.38	0.39	0.36	0.40	0.41	0.40	0.38
MAC+KD+CL	0.39	0.35	0.39	0.40	0.39	0.40	0.41	0.40	0.38
LSTM-attn	0.37	0.30	0.37	0.36	0.34	0.39	0.40	0.40	0.34
LSTM-attn+KD+CL	0.44	0.34	0.41	0.40	0.39	0.42	0.46	0.44	0.37
Training relation length: 7, 8									
MAC	0.34	0.31	0.35	0.38	0.51	0.40	0.44	0.42	0.44
MAC+KD+CL	0.37	0.35	0.37	0.38	0.50	0.36	0.39	0.39	0.40
LSTM-attn	0.41	0.27	0.34	0.37	0.37	0.40	0.41	0.41	0.41
LSTM-attn+KD+CL	0.42	0.35	0.37	0.43	0.55	0.42	0.45	0.43	0.47
Training relation length: 8, 9									
MAC	0.36	0.32	0.35	0.40	0.42	0.42	0.44	0.38	0.45
MAC+KD+CL	0.40	0.32	0.36	0.42	0.41	0.46	0.43	0.37	0.50
LSTM-attn	0.40	0.28	0.31	0.36	0.38	0.39	0.38	0.38	0.46
LSTM-attn+KD+CL	0.40	0.28	0.31	0.36	0.38	0.39	0.38	0.38	0.46
Training relation length: 9, 10									
MAC	0.30	0.33	0.35	0.39	0.42	0.42	0.44	0.46	0.43
MAC+KD+CL	0.35	0.36	0.38	0.40	0.43	0.43	0.46	0.45	0.45
LSTM-attn	0.29	0.31	0.34	0.34	0.40	0.39	0.40	0.42	0.39
LSTM-attn+KD+CL	0.32	0.34	0.37	0.38	0.41	0.43	0.44	0.45	0.43

Table 2: Accuracy on test sets with relation length of 2-10. KD denotes knowledge distillation; CL denotes the MI-based contrastive learning. All results are averaged over 5 runs with different random seeds. The maximum standard deviation is less than 0.05.

Model — Test distractor	Clean	Supporting	Irrelevant	Disconnected
Training set: no distractor				
MAC	0.56	0.49	0.49	0.49
MAC+KD+CL	0.59	0.48	0.55	0.54
LSTM-attn	0.54	0.46	0.50	0.48
LSTM-attn+KD+CL	0.60	0.49	0.57	0.57
Training set: supporting distractor				
MAC	0.50	0.54	0.53	0.53
MAC+KD+CL	0.63	0.57	0.56	0.59
LSTM-attn	0.50	0.45	0.46	0.50
LSTM-attn+KD+CL	0.57	0.59	0.59	0.60
Training set: irrelevant distractor				
MAC	0.42	0.45	0.47	0.42
MAC+KD+CL	0.48	0.50	0.52	0.46
LSTM-attn	0.37	0.38	0.40	0.39
LSTM-attn+KD+CL	0.49	0.51	0.52	0.45
Training set: disconnected distractor				
MAC	0.40	0.41	0.39	0.40
MAC+KD+CL	0.47	0.45	0.44	0.45
LSTM-attn	0.40	0.38	0.37	0.41
LSTM-attn+KD+CL	0.39	0.42	0.39	0.42

Table 3: Accuracy on test sets with different distractors. The distractor types in training sets are given in the table. We augment the MAC network and LSTM by incorporating graph knowledge from GNNs, via knowledge distillaton (KD) and contrastive learning (CL). All results are averaged over 5 runs with different random seeds. The maximum standard deviation is less than 0.05.

Model — Test distractor	Clean	Supporting	Irrelevant	Disconnected
Training set: no distractor				
MAC	0.56	0.49	0.49	0.49
MAC+KD(w/o label)	0.55	0.46	0.53	0.54
MAC+KD(w/ label)	0.59	0.47	0.54	0.52
MAC+KD+CL	0.59	0.48	0.55	0.54
Training set: supporting distractor				
MAC	0.50	0.54	0.53	0.53
MAC+KD(w/o label)	0.62	0.58	0.56	0.60
MAC+KD(w/ label)	0.62	0.57	0.56	0.59
MAC+KD+CL	0.63	0.57	0.56	0.59
Training set: irrelevant distractor				
MAC	0.42	0.45	0.47	0.42
MAC+KD(w/o label)	0.47	0.47	0.49	0.45
MAC+KD(w/ label)	0.48	0.46	0.49	0.44
MAC+KD+CL	0.48	0.50	0.52	0.46
Training set: disconnected distractor				
MAC	0.40	0.41	0.39	0.40
MAC+KD(w/o label)	0.36	0.45	0.41	0.42
MAC+KD(w/ label)	0.40	0.41	0.39	0.40
MAC+KD+CL	0.47	0.45	0.44	0.45

Table 4: Ablation study on different learning objectives. MAC means a MAC network trained with only supervised signals. MAC+KD is a MAC network with knowledge distillation, and we can choose to use labels together with KD (w/ label) or only use soft target produced by a teacher model (w/o label). MAC+KD+CL is a MAC network trained with all three objectives: supervised loss, knowledge distillation loss, and contrastive learning loss. We also tried a model trained with only contrastive learning objective. Its performance is too worse and thus we didn't include it in comparison. A possible reason is that a solo contrastive learning based model is usually trained in two separate periods in which we train an encoder first with contrastive learning, and then train a decoder with labels according to the evaluation task. In our setting, however, we train an encoder and a decoder all together in an end-to-end manner. All results are averaged over 5 runs with different random seeds. The maximum standard deviations is less than 0.05.