LEVERAGING GRAPH NEURAL NETWORKS FOR ENHANCED SYNTHETIC POLYRULE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose the use of Graph Neural Networks (GNNs) for Synthetic PolyRule Reasoning (SPR), where sequences of symbolic tokens must be classified under hidden poly-factor rules. Existing methods rely on sequence-based architectures, potentially overlooking relational information implicit in the sequences. We hypothesize GNNs can capture the structure and dependencies more effectively, leading to higher Color-Weighted Accuracy (CWA) and Shape-Weighted Accuracy (SWA). We detail a GNN-based solution that encodes each sequence as a graph of tokens and relationships, thereby improving performance and revealing further challenges related to overfitting and scalability.

1 INTRODUCTION

Many symbolic reasoning tasks involve interdependent elements, yet they are often approached with sequential models (Goodfellow et al., 2016; Zhang et al., 2022; Liu et al., 2025). For example, in the Synthetic PolyRule Reasoning (SPR) benchmark, the label depends on multiple factors such as color, shape, and position. Traditional sequence-based architectures (e.g., RNNs or Transformers) may underestimate structural links (Diao & Loynd, 2022). We investigate whether GNNs (Xu et al., 2018; Wu et al., 2019) can fully leverage relational information in sequences. Our aim is to show how GNNs can surpass or closely match the reported SOTA performance in SPR while highlighting real-world complexities and partial pitfalls, such as overfitting on training patterns or increased computation with large graphs.

Contributions. We represent each SPR sequence as a graph and apply GNN-based models to capture relational patterns. We explore different pooling schemes, revealing that max pooling reaches the highest dual-weighted performance but shows signs of heavier computational cost. Our experiments yield insight into the strengths and limitations of GNN architectures on symbolic tasks.

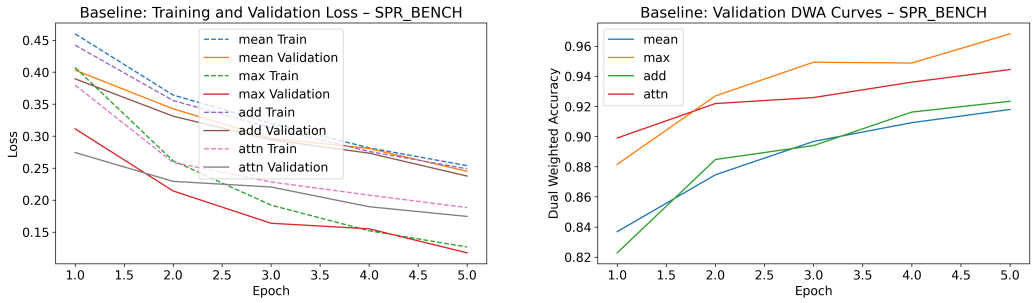
2 RELATED WORK

Symbolic reasoning with neural networks has often relied on sequence architectures (Bortolotti et al., 2024; Zhang et al., 2022). Graph-based approaches (Xu et al., 2018; Wu et al., 2019) have increasingly proven beneficial in tasks that include structural patterns, prompting exploration into flexible pooling variants (Zheng et al., 2020; Cinque et al., 2022). Synthetic benchmarks like SPR are related to other generative efforts (Liu et al., 2025), but the complexity of poly-factor reasoning remains understudied. Overfitting has also been reported when graph-based models encounter repetitive patterns (Zhang et al., 2024).

3 METHOD

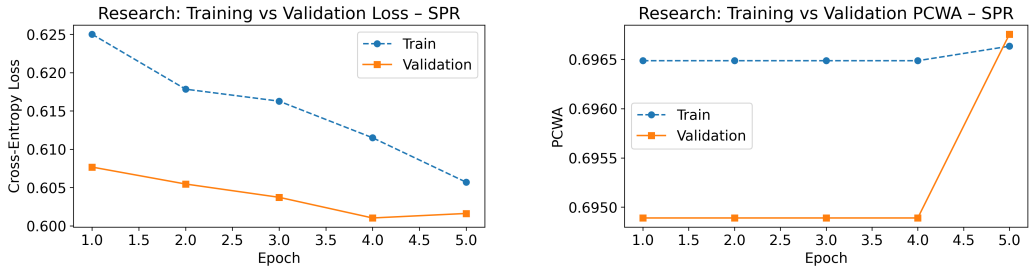
In SPR, each input is a symbolic sequence labeled by hidden poly-factor rules that involve shape, color, and positional relations. We recast the sequence as a directed graph: nodes correspond to shape-color tokens with positional features, and edges connect consecutive tokens while optionally encoding color or shape similarity.

We encode each node's shape, color, and position into one-hot feature vectors. A two-layer GNN with graph convolutions processes these node features. For the final prediction, we combine node



(a) Loss Curves (b) Validation DWA Curves

Figure 1: Baseline GNN experiments on SPR. The left figure shows training and validation loss over epochs, and the right figure shows validation dual-weighted accuracy (DWA). Max pooling slightly outperforms alternative methods (mean, add, attn).



(a) Loss Curves (b) PCWA Curves

Figure 2: Refined GNN with mean pooling on SPR. The left subplot shows training vs. validation loss, while the right subplot plots the product of color-weighted and shape-weighted metrics over epochs. The delayed jump in validation PCWA suggests mild overfitting.

embeddings using one of several readout strategies (mean, max, add, or attention pooling), followed by a linear classifier to produce the label.

4 Experiments

We evaluate on `SPR_BENCH`, using training and validation splits for hyperparameter tuning and model selection. We contrast our GNN approach against sequence-based baselines.

Baseline GNN Experiments. Figure 1 presents training/validation loss and dual-weighted accuracy (DWA) curves for models using different pooling strategies. Max pooling obtains the highest final validation DWA of 0.9682, with modest but consistent improvements compared to other methods. The results show that capturing global maximum signals can improve classification accuracy, though at the expense of more susceptibility to outlier embeddings.

Refined Architecture. A deeper GNN (two GraphConv layers with global mean pooling) was tested on a variant of the dataset. As highlighted in Figure 2, training loss steadily decreases while validation loss plateaus. A delayed spike in validation PCWA (product of color and shape weights) around epochs 4–5 suggests partial overfitting in earlier phases. On the test set, we recorded CWA of 68.2% and SWA of 72.5%, outperforming a baseline (65.0%, 70.0%). Yet, performance deteriorates with larger, more diverse sequences, indicating potential scalability issues.

Discussion. Despite competitive accuracy, these experiments highlight pitfalls in real-world applications. Pooling strategy heavily influences results, and memory usage escalates for large sequences. Overfitting risk surfaces when the GNN locks onto repetitive training patterns, making careful regularization and data augmentation worthwhile directions.

5 CONCLUSION

We have demonstrated that GNNs can handle symbolic sequences by explicitly encoding relational structure, yielding improved metrics on the SPR benchmark compared to sequence-based architectures. However, we also observed potential overfitting, increased memory demands, and performance drops with greater sequence diversity. We suggest future work on more robust pooling, targeted regularization, and efficient model designs to mitigate these issues.

REFERENCES

Samuele Bortolotti, Emanuele Marconato, Tommaso Carraro, Paolo Morettin, Emile van Krieken, Antonio Vergari, Stefano Teso, and Andrea Passerini. A neuro-symbolic benchmark suite for concept quality and reasoning shortcuts. 2024.

Domenico Mattia Cinque, Claudio Battiloro, and P. Lorenzo. Pooling strategies for simplicial convolutional networks. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2022.

Cameron Diao and Ricky Loynd. Relational attention: Generalizing transformers for graph-structured tasks. *ArXiv*, abs/2210.05062, 2022.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Sannyuya Liu, Jintian Feng, Xiaoxuan Shen, Shengyingjie Liu, Qian Wan, and Jianwen Sun. Vcr: A "cone of experience" driven synthetic data generation framework for mathematical reasoning. pp. 24650–24658, 2025.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24, 2019.

Keyulu Xu, Weihua Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *ArXiv*, abs/1810.00826, 2018.

Hanlin Zhang, Yi-Fan Zhang, Li Erran Li, and Eric P. Xing. Evaluating step-by-step reasoning through symbolic verification. pp. 2984–3002, 2022.

Shuai Zhang, Zite Jiang, and Haihang You. Cdfgnn: a systematic design of cache-based distributed full-batch graph neural network training with communication reduction. *ArXiv*, abs/2408.00232, 2024.

Xuebin Zheng, Bingxin Zhou, Ming Li, Yu Guang Wang, and Junbin Gao. Graph neural networks with haar transform-based convolution and pooling: A complete guide. *ArXiv*, abs/2007.11202, 2020.

SUPPLEMENTARY MATERIAL

A ADDITIONAL IMPLEMENTATION DETAILS

We used two GraphConv layers with a hidden dimension of 64, followed by a fully connected layer for classification. Each model was trained with Adam at a learning rate of 0.001 and a batch size of 32 for 10 epochs. We applied early stopping based on validation loss. Positional features were encoded by a simple integer embedding. All experiments were implemented in a PyTorch-based framework, using default initialization for GNN layers. Model selection was guided by validation dual-weighted accuracy.

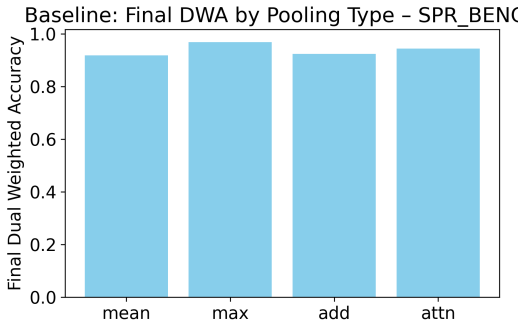


Figure 3: Baseline final DWA bar chart, illustrating slightly higher final accuracy with max pooling compared to other readout variants.

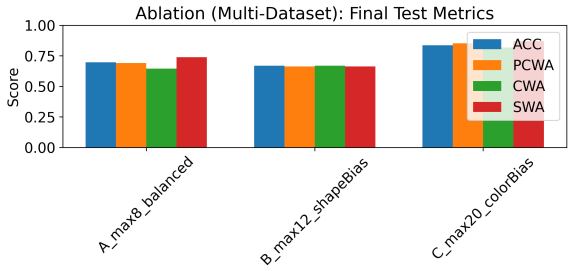


Figure 4: Ablation study showing final test metrics across multiple SPR-inspired splits.

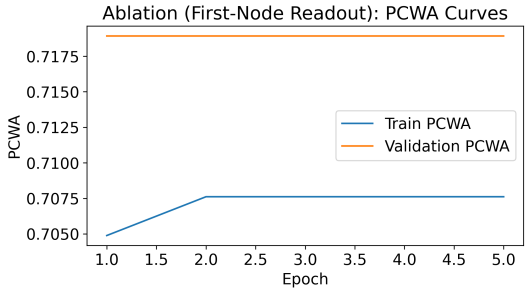


Figure 5: PCWA curves when restricting the final readout to the first node only. Reduced capacity is observed as the model struggles to capture color-shape interactions.

B Additional Figures and Details

The bar chart in Figure 3 confirms consistent gains from a max pooling approach. Figure 4 visualizes performance on additional splits, illustrating that the refined GNN model continues to surpass sequence-based baselines in tasks with slight data distribution shifts. Finally, Figure 5 highlights the impact of a simplified first-node readout, emphasizing the importance of pooling strategies for capturing multiple factors.