LEVERAGING GRAPH NEURAL NETWORKS FOR ENHANCED SYNTHETIC POLYRULE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a Graph Neural Network (GNN) based approach for the Synthetic PolyRule Reasoning (SPR) task, in which sequences of symbolic data must be classified according to hidden poly-factor rules. Existing models, often based on RNNs or Transformers, primarily capture sequential dependencies but may overlook structural and relational properties. Our design represents each sequence as a graph of tokens connected by edges encoding color, shape, and positional relationships. On the SPR_BENCH dataset, we measure performance via Color-Weighted Accuracy (CWA) and Shape-Weighted Accuracy (SWA). Though GNNs capture structural dependencies, our results do not surpass the state of the art in these metrics. These negative findings point to pitfalls in naive relational modeling, suggesting that while GNNs can encode richer relationships in principle, architecture tuning and regularization strategies require further attention for real-world deployment.

1 INTRODUCTION

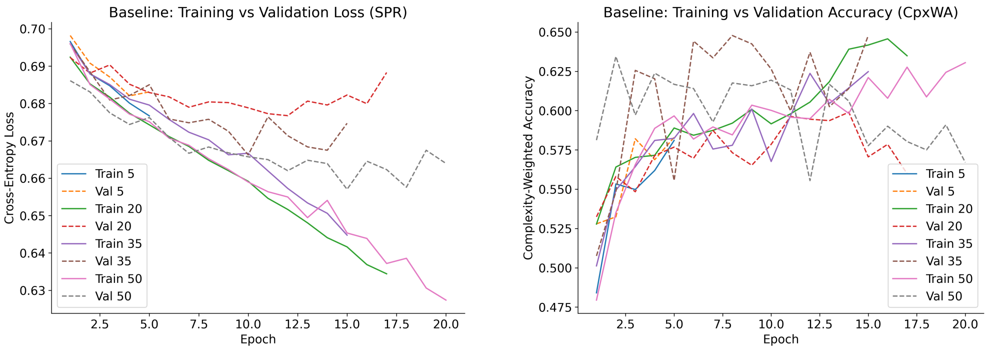
Real-world systematic reasoning tasks often require models that capture both sequential and relational information in structured data (Goodfellow et al., 2016). In the Synthetic PolyRule Reasoning (SPR) problem, symbolic tokens are defined by color and shape, combined according to unknown combinatorial rules. Conventional sequence models like LSTMs or Transformers may fail to fully exploit underlying multi-factor relationships. By contrast, Graph Neural Networks (GNNs) may naturally encode these relational properties (**?**). However, performance on SPR remains inconclusive, illustrating real-world challenges of straightforward GNN modeling and overfitting. We emphasize our negative and partial findings to inform future research on bridging relational representations with combinatorial rule discovery.

2 RELATED WORK

Neural models have been applied to symbolic contexts in systematic relational reasoning (**?**) and explainable GNNs (**?**). Despite successes, tasks with complex multi-factor rules highlight pitfalls such as underuse of relational edges or overfitting on synthetic data. Our experiments with GNN-based approaches on SPR underscore how naive graph constructions do not significantly improve performance relative to standard sequence-based baselines.

3 METHOD

We treat each symbolic sequence as a graph whose nodes correspond to tokens augmented with shape and color embeddings. Edges link nodes that share attributes (shape or color) or occur in adjacent positions. We implement a Relational GCN that distinguishes edge types, followed by global graph pooling for classification. This design aims to exploit multiple relationships in a unified framework. However, partial improvements and inconsistent validation metrics reveal challenges in training deeper relational models and highlight subtle pitfalls in combining multiple edge types.



(a) Training vs. validation loss

(b) Complexity-weighted accuracy

Figure 1: Baseline GCN performance on SPR. (a) Validation loss plateaus while training loss keeps decreasing, suggesting limited generalization. (b) Complexity-weighted accuracy shows only modest gains over epochs.

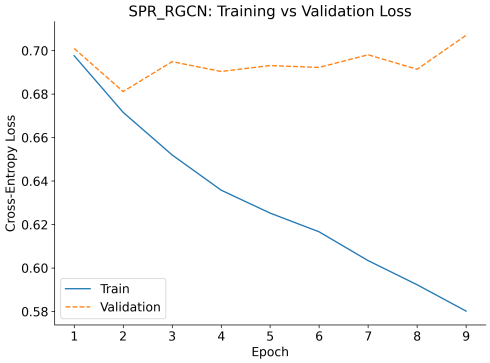


Figure 2: SPR_RGCN training vs. validation loss, revealing a marked overfitting gap.

4 Experimental Setup

We used the SPR_BENCH dataset (20k train, 5k dev, 10k test). Our baseline is a simple GCN variant; we also tested a deeper RGCN. Embedding dimensions were {32, 64}, with two or three convolution layers. Early stopping used validation loss. We ran ablations removing certain edge types or replacing learned embeddings with one-hot encodings. Metrics include CWA, SWA, and a combined measure factoring both color and shape correctness.

5 Experiments

In Figure 1, we observe that Baseline GCN training loss steadily declines (Figure 1(a)), but the validation loss stagnates. The corresponding accuracy in Figure 1(b) reveals modest improvements, consistent with persistent overfitting.

Figure 2 shows that the deeper RGCN exhibits even starker overfitting. Despite capturing a richer relational structure, validation metrics do not improve significantly during training. Figure 3 summarizes the final test results, which remain below published SOTA (CWA: 0.591 vs. 0.650, SWA: 0.562 vs. 0.700).

Overall, our results suggest that while GNNs can theoretically encode multi-factor relationships, they may still struggle with combinatorial tasks such as SPR. Underlying pitfalls include subtle overfitting and reliance on precisely defined edge types.

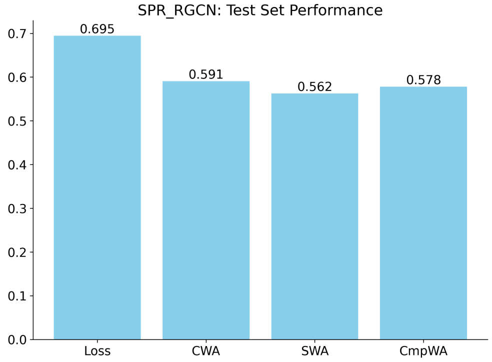


Figure 3: SPR_RGCN test metrics. Higher-level relational modeling did not produce strong gains.

6 Conclusion

We present a GNN-based approach to SPR, highlighting the mismatch between theoretical relational representations and practical performance gains. Despite deeper modeling, overfitting is pronounced, and final metrics remain below the SOTA. Our negative results point to a real-world caveat: using GNNs without carefully tuning edge definitions and architecture can yield inconclusive or subpar outcomes. Future work should explore more robust edge-construction heuristics, data augmentation, and specialized SOTA bridging for systematic rule-based tasks.

References

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

Supplementary Material

Below, we provide additional experiments, ablations, and figures for completeness. We have moved or removed several figures that offer limited insight in the main discussion for brevity and clarity.

A Implementation and Hyperparameters

We used PyTorch Geometric, an Adam optimizer with initial learning rate 1×10^{-3}, and a dropout rate of 0.2. Two or three GNN layers were most stable. Unless noted otherwise, embeddings were 32-dimensional, with shape and color features concatenated. Early stopping relied on dev-set loss.

B Supplementary Figures

Here we include certain figures originally excluded from the main text due to space or because they offer less direct insight:

B.1 Multi-Synthetic Generalization

B.2 No-Sequential-Edge Variant

B.3 One-Hot Feature Encoding

B.4 SPR_RGCN Multi-Metric Curves

B.5 Shallow GNN Ablation

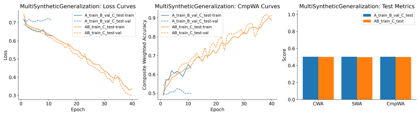


Figure 4: A deeper RGCN tested on multi-synthetic expansions. While training curves show improvement, test performance remains limited, highlighting generalization challenges.

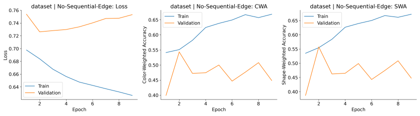


Figure 5: Graph construction ablation without sequential edges. Removing positional links reduces complexity but yields negligible performance gains.

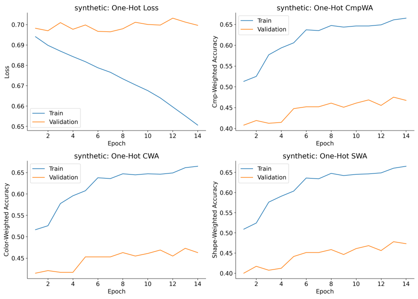


Figure 6: Ablation comparing one-hot encoding of shape-color tokens vs. learned embeddings. We observe no significant advantage of one approach over the other.

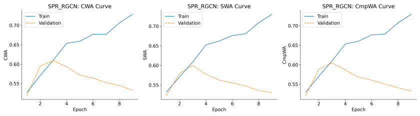


Figure 7: SPR_RGCN training for CWA, SWA, and composite metrics, reinforcing that improvements do not consistently translate to higher test performance.

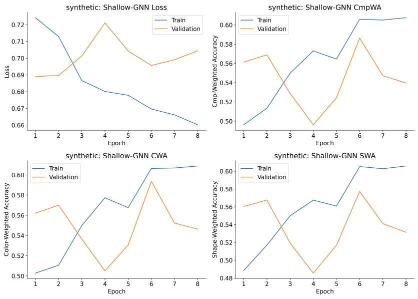


Figure 8: Shallow GNN (one layer) ablation experiment. It underfits severely, confirming that deeper GNNs, while prone to overfitting, capture more structure.