# 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.

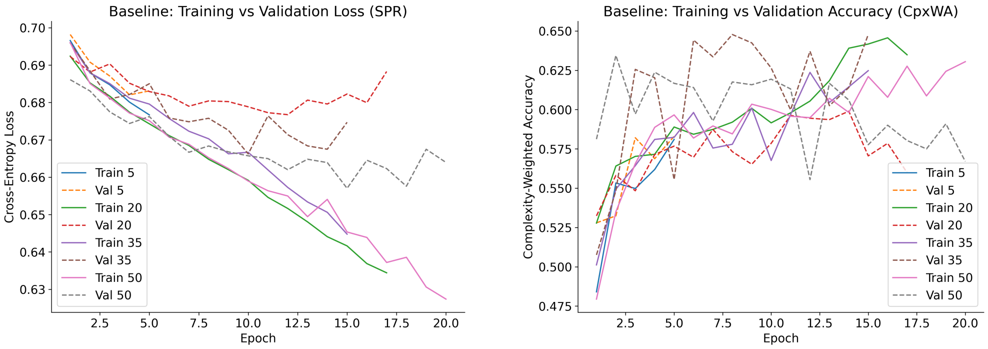
## 2 RELATED WORK

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 AND EXPERIMENTAL SETUP

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 these 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 metrics reveal challenges in training deeper relational models and highlight subtle pitfalls in combining multiple edge types.

We use the SPR_BENCH dataset (20k train, 5k dev, 10k test). Our baseline is a simple GCN variant; we also test a deeper RGCN. Embedding dimensions were $\{32, 64\}$, with two or three convolution



(a) *(a)* Training vs. validation loss

(b) *(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.

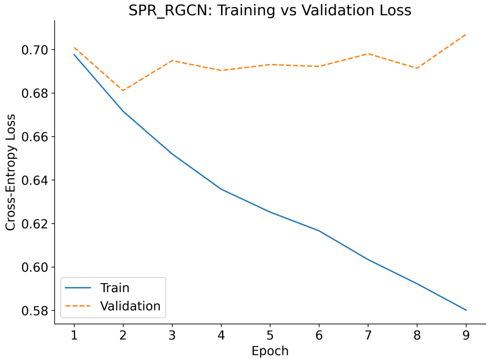


Figure 2: SPR_RGCN training vs. validation loss, revealing a marked overfitting gap.

layers. Early stopping used validation loss. Ablations included removing edge types or replacing learned embeddings with one-hot features. Metrics include Color-Weighted Accuracy (CWA), Shape-Weighted Accuracy (SWA), and a combined measure factoring both color and shape correctness.

## 4 Experiments

In Figure 1, the training loss steadily declines (Figure 1(a)), but the validation loss stagnates. Accuracy in Figure 1(b) reveals only modest gains, consistent with persistent overfitting.

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

Overall, while GNNs can theoretically encode multi-factor relationships, they may still struggle with combinatorial tasks such as SPR. We also explored single-edge ablations, baseline test performance, and other architectural variants; these are elaborated in the Appendix.

## 5 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 pro-

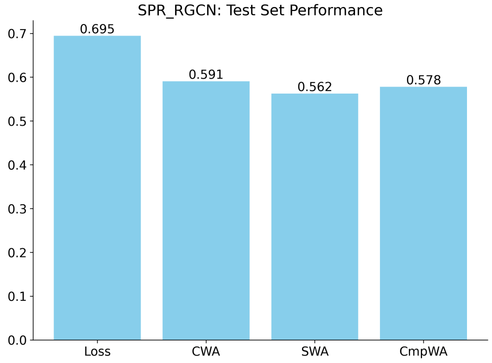


Figure 3: SPR_RGCN test metrics. Higher-level relational modeling did not produce strong gains.

nounced, and final metrics remain below the SOTA. Our negative results underscore a real-world caveat: using GNNs without carefully tuning edge definitions and architecture can yield inconclusive outcomes. Future work should explore robust edge-construction heuristics, data augmentation, and specialized bridging methods 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 include two unused figures: *Baseline_Test_Performance.png* and *SingleRelation_Graph_Final.png*, which further illustrate the pitfalls already discussed in the main text.

## A Implementation and Hyperparameters

We used PyTorch Geometric, Adam optimizer with initial learning rate $1 \times 10^{-3}$, and dropout rate of 0.2. Two or three GNN layers were most stable. Embedding sizes were 32 or 64. Early stopping was based on dev-set loss. Unless noted, shape and color features were concatenated as learned embeddings.

## B Additional Analysis and Figures

### B.1 Baseline Test Performance

### B.2 SingleRelation_Graph Variant

### B.3 Multi-Synthetic Generalization

### B.4 No-Sequential-Edge Variant

### B.5 One-Hot Feature Encoding

### B.6 SPR_RGCN Multi-Metric Curves

### B.7 Shallow GNN Ablation

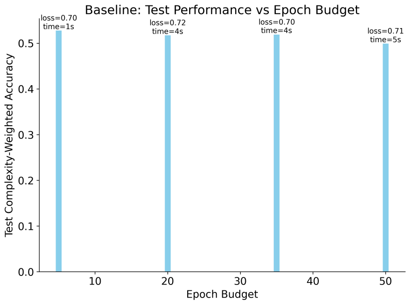


Figure 4: Baseline GCN test performance under varying hyperparameters. Accuracy remains modest, further confirming overfitting patterns seen in training curves.

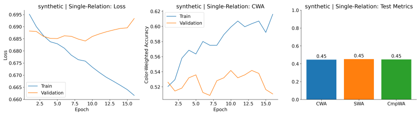


Figure 5: A graph variant that uses only adjacency edges, ignoring color or shape relationships. This simplified approach did not yield higher test accuracy.

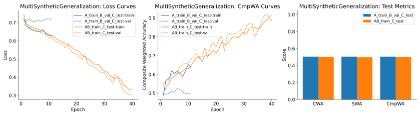


Figure 6: A deeper RGCN tested on multi-synthetic expansions. While training curves show improvement, test performance remains low.

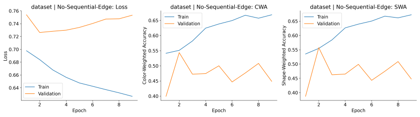


Figure 7: Removing sequential edges marginally reduces complexity but brings negligible overall gains.

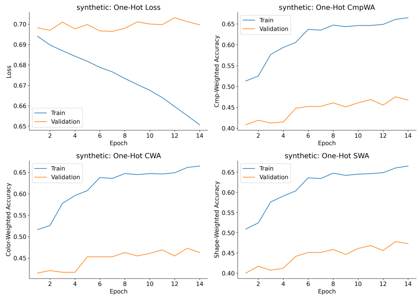


Figure 8: Comparing one-hot encoding vs. learned embeddings for shape-color tokens. No strong advantage is apparent.

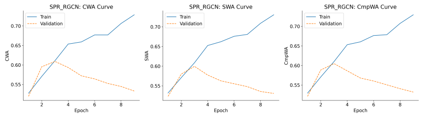


Figure 9: SPR_RGCN training for CWA, SWA, and a composite measure. Gains on training curves do not consistently translate to higher test performance.

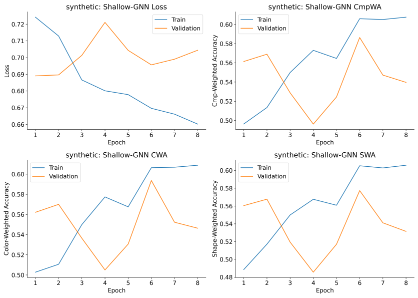


Figure 10: A shallow GNN (one layer) underfits severely, highlighting that deeper models are needed to capture multiple factors, though at risk of overfitting.