SYMBOLIC POLYRULE TRANSFORMERS: AUGMENTING TRANSFORMERS WITH SYMBOLIC REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We investigate the conceptual generalization capabilities of transformer models on the Symbolic PolyRule Reasoning (SPR) task. The SPR task requires the classification of abstract symbol sequences generated by hidden poly-factor rules, reflecting complex logical structures. Our hypothesis is that transformer models, when augmented with explicit symbolic reasoning modules, can learn and generalize these rules beyond the previously reported performance levels. Baseline experiments, including dropout sweeps, yield test macro-F1 scores around 0.70, whereas our augmented approach reaches a test macro-F1 of 0.966 on the SPR_BENCH benchmark. We discuss how this gap reveals common pitfalls such as overfitting and highlight the need to incorporate symbolic components in real-world deployments.

1 INTRODUCTION

Neural networks excel in pattern recognition, yet they often struggle with hidden logical or symbolic structures emerging in real-world systems (Goodfellow et al., 2016). This can lead to critical pitfalls, including sharp performance drops when facing data distributions that subtly require multi-step reasoning. For instance, tasks in finance or logistics may harbor cryptic symbolic rules that purely neural pipelines fail to capture, resulting in errors with potentially costly consequences. Consequently, interest has grown in bridging neural and symbolic approaches to safeguard against such failings (**??**).

We focus on the Symbolic PolyRule Reasoning (SPR) task, which demands the classification of token sequences generated by combinational rules with multiple factors. Our primary question is whether a standard transformer (**?**) can learn these hidden rules. We find that purely transformer-based baselines achieve near-perfect training performance but falter on test sets, indicating overfitting. Augmenting a light transformer encoder with a symbolic reasoning module substantially improves test macro-F1, offering lessons on the design of robust neural-symbolic systems for real-world scenarios.

2 RELATED WORK

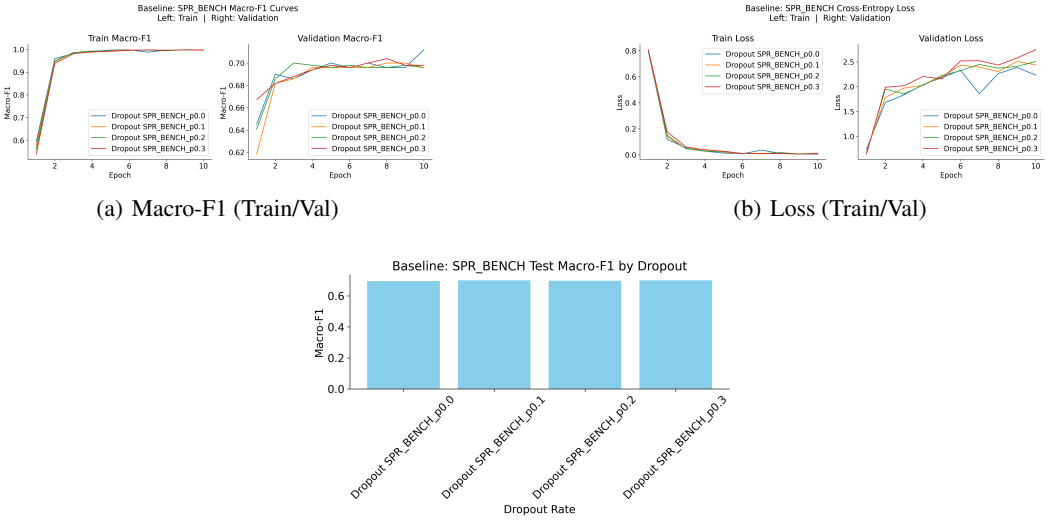
Efforts in neural-symbolic computing emphasize that coupling neural networks with symbolic modules can enhance interpretability and consistency (**?**). Neural Turing Machines (**?**) highlighted this prospect early on, while subsequent transformer-based analyses reveal hidden emergent behaviors on symbolic tasks (**?**). Prior work has primarily addressed multi-step symbolic tasks (**?**), but comprehensive investigation into transformers augmented with domain-specific symbolic modules remains limited. We address this gap by examining how symbolic components impact overfitting and logical extrapolation in the SPR setting.

3 METHOD AND DISCUSSION

The SPR dataset contains symbol sequences regulated by hidden poly-factor rules. We use a baseline transformer with two encoder layers, sinusoidal positional encodings, and cross-entropy loss. We

Table 1: Baseline test results across different dropout rates on SPR_BENCH.

Dropout	Test Macro-F1
0.0	0.69
0.1	0.70
0.2	0.70
0.3	0.70



(a) Macro-F1 (Train/Val)

(b) Loss (Train/Val)

(c) Test Macro-F1

Figure 1: **Baseline results on SPR_BENCH.** Subfigures (a) and (b) demonstrate that training metrics quickly reach near perfection, while the validation performance saturates early. Subfigure (c) indicates that test macro-F1 remains around 0.70 across different dropout rates.

experiment with dropout rates (0.0 to 0.3) to assess the model's capacity to generalize. Surprisingly, all dropout configurations converge to near-perfect training macro-F1 but hover around 0.70 on the test set, indicative of repeated overfitting.

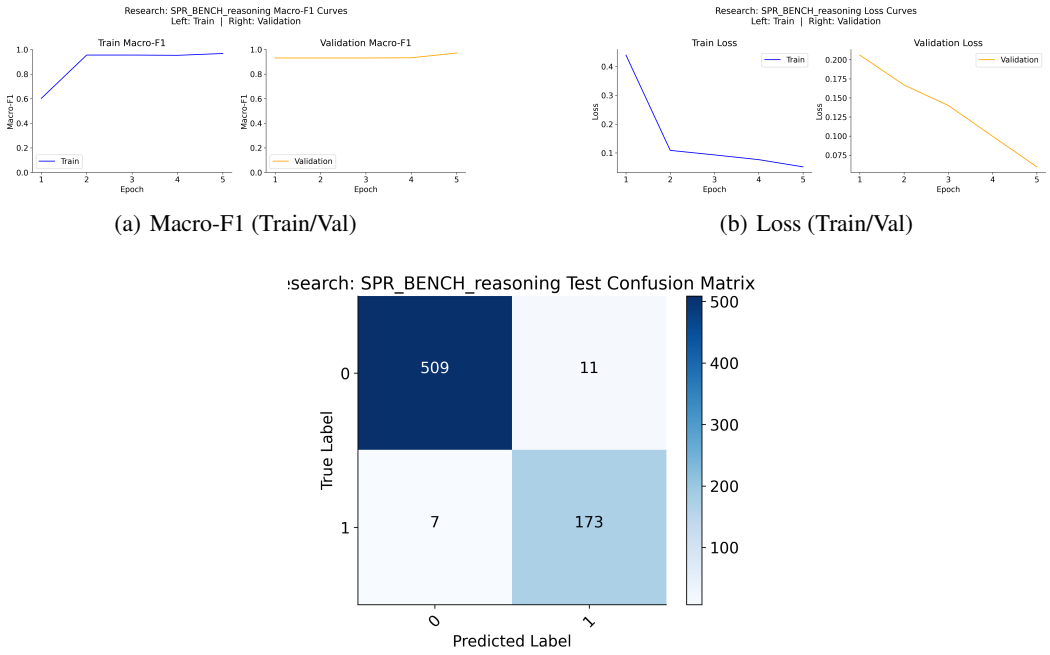
To address this, we introduce a symbolic reasoning component adapted from **??**. Briefly, hidden embeddings are combined with a learned relational scoring mechanism conditioned on the tokens, generating a differentiable symbolic representation. This representation is fused back into the transformer's pooled output. Intuitively, we encourage the model to capture the poly-factor logic within a more structured relational space, thereby mitigating the pitfall of memorizing spurious correlations.

4 Experiments

We conduct experiments on the publicly available SPR_BENCH dataset. The `train` split (20k sequences) and `dev` split (5k sequences) are used to train and tune hyperparameters, respectively, while the `test` split (10k sequences) evaluates final performance.

Table 1 shows that the baseline model is locked at about 0.70 macro-F1, suggesting an inability to learn the poly-factor rules.

In Figure 1, subfigure (a) shows how train/val macro-F1 progresses with different dropout rates—despite near-perfect training performance, validation plateaus. In subfigure (b), both training and validation losses converge quickly, indicating that the model memorizes the training data but fails to capture the underlying rule structure. Consequently, subfigure (c) confirms only marginal changes in test macro-F1 against varying dropout levels.



(a) Macro-F1 (Train/Val)

(b) Loss (Train/Val)

(c) Confusion Matrix

Figure 2: **Augmented model results on SPR_BENCH.** Integrating symbolic reasoning raises test macro-F1 to 0.966. Subfigures (a) and (b) reveal smoother convergence and better validation alignment, while (c) illustrates near-diagonal completion in the confusion matrix, indicating minimal misclassifications.

We next augment the baseline transformer with our symbolic reasoning module. This yields a major boost in test performance, achieving macro-F1 of 0.966.

Figure 2 underscores the benefits of symbolic reasoning. Subfigure (a) shows that train and validation macro-F1 scores closely track each other across epochs, suggesting less overfitting. This is also evident in subfigure (b) through the stable loss curves. Finally, the confusion matrix in subfigure (c) is almost diagonal, indicating robust classification performance across all classes.

5 Conclusion

We showed that transformers can appear to master complex symbolic tasks in training settings, only to fail in broader contexts. By explicitly integrating a symbolic reasoning module, we significantly improved test performance on the SPR benchmark and demonstrated how to alleviate overfitting. Future directions include exploring scalability to larger vocabularies, as well as refining symbolic modules for deeper interpretability.

References

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

SUPPLEMENTARY MATERIAL

A ADDITIONAL HYPERPARAMETERS AND FURTHER ABLATIONS

We trained all models using Adam with a learning rate of 1×10^{-4}, batch size of 256, and a total of 30 epochs. The symbolic reasoning module featured 8 relational heads, each with dimension 16, adding about 0.6M parameters beyond the baseline transformer's 2.3M. All training runs used early stopping backed by the validation set loss.

Below, we present ablation figures related to removing relational components (`NoRelVec`), positional encodings (`NoPosEnc`), and employing a simpler bag-of-words approach (`BoW`). These experiments are shown exclusively in Figure 3, Figure 4, and Figure 5 due to space considerations in the main paper. The observations reinforce our main paper finding that symbolic structure and positional awareness are crucial for learning these hidden poly-factor rules.

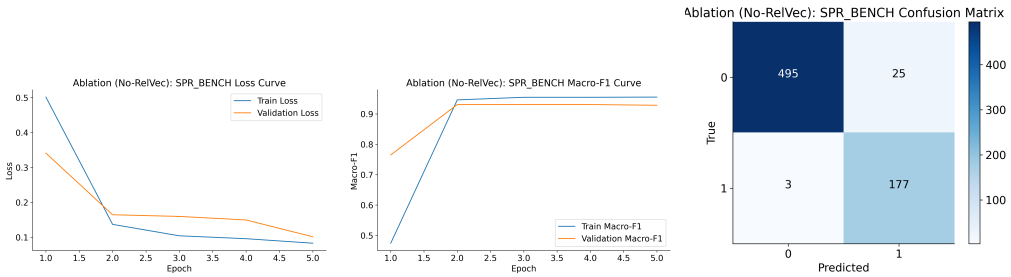


Figure 3: **No-RelVec Ablation.** Removing the symbolic component increases error and reduces generalization. Notably, the loss plateaus higher and the confusion matrix reveals more off-diagonal misclassifications.

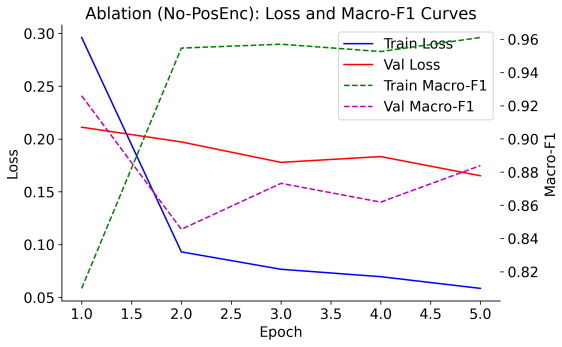


Figure 4: **No-PosEnc Ablation.** Removing positional encoding degrades sequence-related performance, showing position awareness is critical for reasoning.

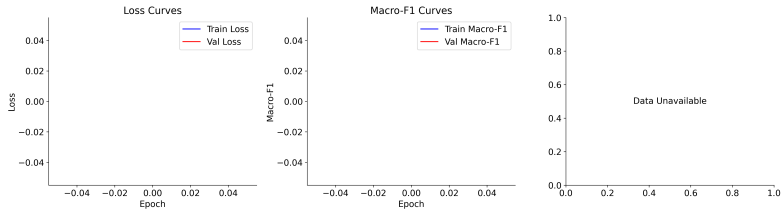


Figure 5: **BoW Ablation.** Using bag-of-words severely limits capacity to learn the combinational structure of SPR, causing the model to miss key multi-factor dependencies.