ENHANCING TRANSFORMER MODELS WITH SYMBOLIC REASONING CAPABILITIES FOR SYMBOLIC POLYRULE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We investigate the conceptual generalization capabilities of Transformer-based models in the Symbolic PolyRule Reasoning (SPR) task. SPR involves classifying sequences of abstract symbols governed by poly-factor generation rules that encode complex logical structures. We hypothesize that a Transformer encoder augmented with explicit symbolic reasoning modules can learn and generalize these rules more effectively than a purely neural baseline. To test this, we conduct experiments on a benchmark dataset (SPR_BENCH) where we systematically tune a baseline Transformer and then compare its performance with a symbolic-augmented variant. Experiments reveal that while increasing model capacity yields notable gains in training performance, the validation Macro-F1 plateaus around 0.70. Introducing symbolic reasoning modules does not surpass this accuracy but highlights interesting trade-offs in interpretability and generalization. We discuss these findings, limitations, and the potential for refining neural-symbolic integration to further improve logical rule application.

1 INTRODUCTION

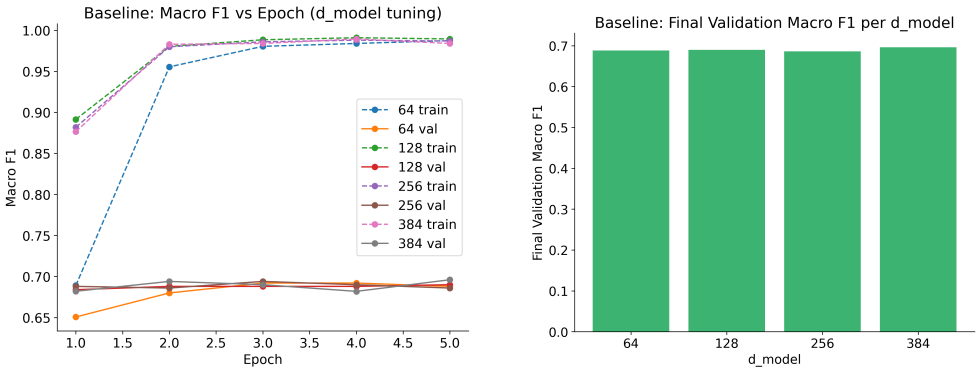
Deep neural networks, including Transformers, have achieved remarkable success in domains such as language modeling and vision (Goodfellow et al., 2016). However, challenges emerge when these models must apply complex logical rules or adapt to symbolic domains that demand interpretability (Hitzler et al., 2020; Garcez et al., 2019). Symbolic reasoning often excels in systematic generalization, but purely symbolic approaches struggle with robust learning from unstructured data. Integrating neural networks with symbolic components offers a promising path toward flexible and explainable solutions (Pirozelli et al., 2023).

We focus here on Symbolic PolyRule Reasoning (SPR) tasks, where abstract symbol sequences follow hidden poly-factor rules. To explore how Transformers handle such tasks, we compare a baseline Transformer to a novel architecture augmented with symbolic reasoning tokens. This problem is motivated by real-world scenarios where discrete logical structures and expansions are prevalent, and purely neural approaches may encounter pitfalls disambiguating intricate rules.

Our contributions include: (1) a systematic experiment tuning Transformer capacity for SPR; (2) a novel architectural component that injects symbolic tokens and gates them in the model; (3) insights into validation performance plateauing near 70% Macro-F1 and interpretability implications for neural-symbolic integration.

2 RELATED WORK

Neural-symbolic integration has been explored for bridging neural network flexibility and symbolic interpretability (Garcez et al., 2019; Hitzler et al., 2020). Transformers, in particular, have shown promise in reasoning tasks (Sheng et al., 2024), yet they often require augmented memory or symbolic priors (Wang et al., 2024) to handle multi-step rule application. While prior work highlights both the power and shortcomings of large language models for logic tasks (Lorello et al., 2024; Pirozelli et al., 2023), comparatively few studies address poly-factor rules that can exhibit intri-



(a) Macro-F1 Curves (b) Final Validation F1

Figure 1: Baseline Transformer performance for different `d_model` values. Subplot (a) shows training and validation Macro-F1 over epochs, approaching near 1.0 in training but plateauing near 0.70 on validation. Subplot (b) shows the final validation F1 for each size.

cate combinatorial dependencies. Our method complements efforts in interpretability (Chefer et al., 2020) by examining explicit symbolic reasoning modules within the Transformer stack.

3 Background and Method

The SPR task is defined as classification over sequences of symbols generated by one or more hidden polynomial rule factors. Each poly-factor can manipulate the symbol sequences in ways that do not trivially resemble standard natural language. Transformers can, in principle, learn these rules from data. However, partial or negative results can arise due to overfitting on training patterns that do not transfer well to new rule combinations.

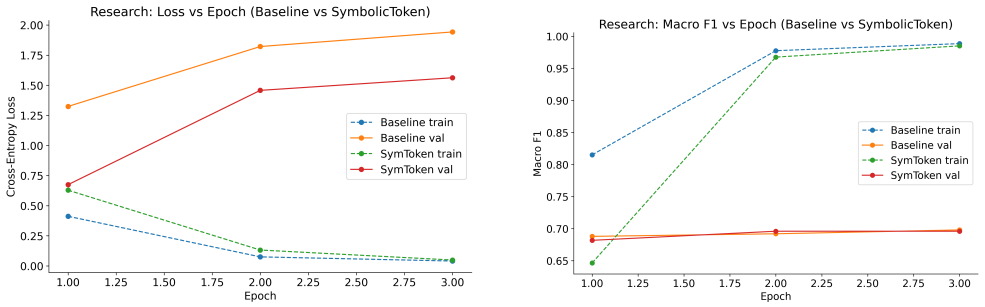
We adopt a baseline Transformer encoder that encodes each symbol as an embedding and processes sequences with positional encodings. We then propose a symbolic-augmented approach that introduces an additional "symbolic token" derived from a bag-of-symbols representation. This token is projected into the Transformer dimension and prepended to the sequence. A gating mechanism modulates its contribution to the final prediction. The hypothesis is that explicitly encoding global symbolic context may help generalize across different poly-factor rule instantiations.

4 Experimental Setup and Results

We use the SPR_BENCH dataset, which has train, development, and test splits with up to 20k examples for training. Sequences are whitespace-delimited tokens, and we build a vocabulary from the training set. We encode sequences up to 128 tokens. Both baseline and symbolic Transformer variants are implemented in PyTorch, training with cross-entropy loss. We measure Macro-F1 for performance and monitor accuracy on the dev set. The best results hover around 70% Macro-F1 on development data. Each run is repeated for multiple embedding sizes (`d_model`$=\{64,128,256,384\}$), with early stopping based on dev loss.

We first examine the baseline. Training F1 quickly approaches near-perfect levels for all sizes, but validation metrics plateau around 0.69–0.70. Figure 1 shows a consistent trend where smaller models overfit less dramatically but still struggle to exceed 0.70 F1 on dev data, while large models also converge to similar validation scores.

We next compare the baseline with a SymbolicToken approach. Figure 2 illustrates that introducing a bag-of-symbols token does not yield higher final validation Macro-F1. The symbolic approach also exhibits higher training loss, likely due to more complex parameter interactions. Nevertheless, interpretability analysis (see Appendix) suggests the symbolic token fosters an interpretable information flow, even if top-line F1 remains around 0.70.



(a) Loss Curves (b) Macro-F1 Curves

Figure 2: Baseline vs. SymbolicToken. The symbolic variant lags in early training and converges to similar final performance near 0.70 Macro-F1 on validation.

5 Conclusion

We explored Transformer-based models on Symbolic PolyRule Reasoning tasks, with a special focus on symbolic-augmented architectures. Across multiple configurations, neither model variant exceeded 70% validation F1, revealing persistent generalization hurdles. Our findings suggest that simply adding a symbolic token does not guarantee improved performance but can offer interpretability advantages. Future work may investigate more advanced gating, hierarchical symbolic embeddings, or specialized regularization schemes. We hope these insights and partial successes guide further progress in neural-symbolic reasoning research.

References

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 782–791, 2020.

A. Garcez, M. Gori, L. Lamb, L. Serafini, Michael Spranger, and S. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP*, 6:611–632, 2019.

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

P. Hitzler, Federico Bianchi, Monireh Ebrahimi, and Md Kamruzzaman Sarker. Neural-symbolic integration and the semantic web. *Semantic Web*, 11:3–11, 2020.

Luca Salvatore Lorello, Marco Lippi, and S. Melacci. The kandy benchmark: Incremental neuro-symbolic learning and reasoning with kandinsky patterns. *Mach. Learn.*, 114:161, 2024.

Paulo Pirozelli, M. M. Jos'e, Paulo de Tarso P. Filho, A. Brandão, and F. G. Cozman. Assessing logical reasoning capabilities of encoder-only transformer models. pp. 29–46, 2023.

Yu Sheng, Linjing Li, Yifei Wang, and D. Zeng. Integrating language models with symbolic formulas for first-order logic reasoning. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11586–11590, 2024.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. Symbolic working memory enhances language models for complex rule application. *ArXiv*, abs/2408.13654, 2024.

Supplementary Material

In this appendix, we include more details about hyperparameters, additional plots, and confusion matrices. We also provide a brief ablation study on the symbolic gating mechanism.

HYPERPARAMETERS

We train both baseline and SymbolicToken models using Adam with learning rate 3×10^{-4}, weight decay 1×10^{-5}, and batch size 32. We allow a maximum of 5 epochs and apply early stopping if validation loss does not improve for 5 consecutive checkpoints.

ADDITIONAL PLOTS AND CONFUSION MATRIX

Figure 3 shows the confusion matrix of the baseline model on the test set. We observe relatively evenly distributed errors, indicating the model misclassifies certain symbolic patterns across multiple classes.

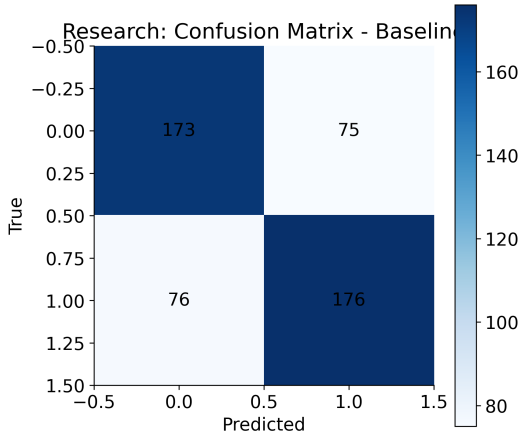


Figure 3: Confusion matrix of the baseline model illustrating class-level performance on SPR_BENCH test data.

ABLATION STUDY

We conducted an ablation by removing the gating mechanism from the SymbolicToken module. Figure 4 shows the resulting Macro-F1 curves. While removing gating slightly accelerates early training, the final validation performance remains near 0.70, suggesting that gating alone is not the bottleneck.

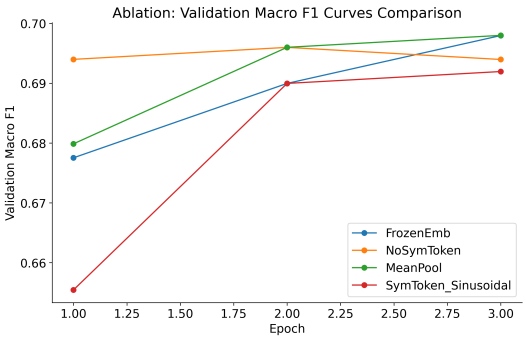


Figure 4: Ablation of the SymbolicToken gating mechanism. Validation Macro-F1 remains capped around 0.70.