Research Report: An Investigation into Neural-Symbolic Systems for SPR Tasks

Agent Laboratory

Abstract

In this work, we introduce a novel neuro-symbolic approach to Sequence Pattern Recognition (SPR) that unifies neural representation learning with symbolic rule induction. The proposed framework integrates a Long Short-Term Memory (LSTM) network with a constrained Answer Set Programming (ASP) module in order to simultaneously learn latent representations from raw data and induce interpretable symbolic rules. The training objective is formulated as a joint optimization problem:

$$(\theta^*, H^*) = \arg\max_{\theta, H} \prod_{i=1}^{N} P(y_i \mid x_i, B, H, \theta),$$

where x_i denotes the raw sequence input, y_i is the corresponding label, B denotes background knowledge, H is the induced hypothesis (symbolic rule set), and θ represents the parameters of the neural network. Our experiments on the SPR_BENCH dataset demonstrate that the training loss decreases from 0.6486 to 0.1806 over three epochs. The the final test SWA stands at 67.87%, and the final test CWA is 66.50%. Although the in-domain performance is promising, the notable generalization gap motivates further exploration in regularization and adaptive rule induction strategies. This paper presents the technical details, experimental evidence, and a discussion of potential future directions for robust neuro-symbolic integration.

1 Introduction

Neural-symbolic integration represents an important research direction aimed at combining the robust data-driven learning capabilities of neural networks with the inherent interpretability of symbolic reasoning. This paper addresses the problem of Sequence Pattern Recognition (SPR), a task that requires mapping sequences of abstract symbols (e.g., "$\bullet y \bullet g \bullet r \,\square r\, \Delta y\, \Delta g$") to corresponding labels. The inherent challenge lies in transforming raw symbolic sequences into a latent representation that benefits from both numerical optimization and logical inference.

Our approach leverages a hybrid framework in which an LSTM-based neural network learns continuous representations from raw data, while an ASP module

is used to induce a symbolic hypothesis that faithfully describes the underlying structure. The combined optimization objective is given by:

$$(\theta^*, H^*) = \arg\max_{\theta, H} \prod_{i=1}^{N} P(y_i \mid x_i, B, H, \theta),$$

which emphasizes the simultaneous optimization over both the continuous parameter space of the neural network and the discrete space of symbolic rules. Although the neural network is able to rapidly reduce the training loss, as evidenced by the decrease from 0.6486 to 0.1806 over three epochs, the large discrepancy between development and test SWA signifies potential overfitting or data distribution issues.

In this paper, we describe the structure of our framework, outline the experimental design, and discuss the implications of our findings. Given the importance of reliable performance in safety-critical applications and the need for explainability in decision-making systems, the impact of our integrated approach is twofold: it offers promising accuracy for SPR tasks while maintaining a pathway towards interpretability in complex environments.

2 Background

The integration of neural networks and symbolic reasoning methods has a long history in the literature. Classical work in symbolic AI has established methods such as logic programming and inductive logic programming (ILP) that provide clear and interpretable decision processes. More recent advances in deep learning, particularly in recurrent neural networks such as LSTMs, have shown great promise in modeling sequential data with high accuracy. However, neural methods lack direct interpretability, a key disadvantage in scenarios where transparency is necessary.

Answer Set Programming (ASP) has emerged as a powerful tool for encoding logical rules and constraints. In our work, ASP is used to impose logical consistency on the latent representations learned by the neural component. The ASP component is guided by background knowledge B and a constrained hypothesis space $\mathcal{H}_{cand}$. By defining a semantic loss term in addition to a conventional cross-entropy loss, we enforce a mapping between the neural outputs and a logically consistent symbolic domain:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \, \mathcal{L}_{sem}.$$

The theoretical foundations for combining these two modalities have been extensively explored in the literature. In our framework, the neural network $f_\theta : \mathcal{X} \to \mathcal{Z}$ projects raw symbolic sequences into a latent space $\mathcal{Z}$, and the symbolic reasoning engine then induces rule sets H that satisfy:

$$I \models B \cup H \quad \text{if and only if} \quad H \cup \{z\} \models y.$$

This formulation ensures that the neural representations are not merely effective for prediction, but also interpretable through a set of logically deduced rules, thus creating a bridge between continuous and discrete representations.

3 Related Work

Recent advances in neuro-symbolic systems have aimed to integrate learning with logical reasoning in various application domains. For example, approaches such as NeuralFastLAS and MetaABD have attempted to merge neural output with ILP techniques for rule induction from raw data. However, many of these methods rely on hand-engineered rules or require separate stages for neural training and symbolic extraction, leading to scalability issues and a lack of end-to-end training.

Other works in the literature have addressed pattern matching using techniques like conservative degenerate pattern matching or fuzzy sequence matching, focusing on specific aspects of symbolic manipulation. Our proposed framework distinguishes itself by providing a unified architecture that performs both representation learning and rule extraction simultaneously. Unlike methods that assume pre-encoded symbolic inputs, our system starts from raw symbolic sequences, which necessitates the development of robust tokenization, padding, and latent representation mechanisms.

Additionally, recent work employing transformer architectures for sequence-to-sequence mapping have demonstrated high performance driven by attention mechanisms; however, these models often lack interpretability. In contrast, our method produces an explicit symbolic rule set H that can be inspected and verified by domain experts. Table 1 summarizes key differences between our work and related methods.

Table 1: Comparison between our approach and key related works.

Method	**Input Representation**	**Learning Component**
MatchPy (Symbolic Matching)	Pre-encoded symbols	None
SSL for Symbolic Sequences	Raw visual data	Self-supervised learning
Conservative Pattern Matching	Fixed degenerate symbols	No neural adaptation
Ours	Raw sequence data	End-to-end neural-symbolic joint training

The literature has also highlighted the benefits and challenges associated with hybrid approaches, including the need for enhanced regularization, adaptive loss balancing, and scalable rule induction strategies. Our work contributes to this discussion by empirically evaluating the performance implications of jointly optimizing neural and symbolic components.

4 Methods

Our methodological contribution is based on constructing a coupled neuro-symbolic framework that links continuous representation learning with discrete rule induction. The overall objective can be expressed as:

$$(\theta^*, H^*) = \arg\max_{\theta, H} \prod_{i=1}^{N} P(y_i \mid x_i, B, H, \theta),$$

where the parameters θ of the neural model, and the symbolic hypothesis H are optimized simultaneously.

The approach consists of several key stages:

1. **Latent Representation Learning:** A neural network with an embedding layer, an LSTM module, and a fully connected classification layer is used to convert raw token sequences into dense feature representations. These features are then mapped into a latent symbolic space $\mathcal{Z}$ through the learned function f_θ.

2. **Hypothesis Space Construction:** Leveraging background knowledge B and mode declarations, a candidate hypothesis space $\mathcal{H}_{cand}$ is constructed. This space encompasses potential symbolic rules that are consistent with the observed data.

3. **Semantic Loss Integration:** In order to ensure that the latent representations are aligned with logical constraints, a semantic loss term $\mathcal{L}_{sem}$ is added to the conventional cross-entropy loss. The total loss function is thus defined as:

   $$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{sem},$$

   where λ controls the trade-off between predictive accuracy and logical consistency.

4. **Rule Induction and Pruning:** Given the candidate hypothesis space, symbolic rules are induced by searching for rules that maximize the consistency between predicted latent labels and ground-truth labels. A pruning strategy further reduces the rule set to an optimal subset S_{opt} defined as:

   $$S_{opt} = \min_{H \subset \mathcal{H}_{cand}} \{\text{Length}(H) \mid H \cup \{z\} \models y \quad \forall (x, y) \in \mathcal{D}\}.$$

   This step ensures interpretability by favoring simpler rule sets and prevents overfitting to spurious patterns.

Through these integrated steps, the neural and symbolic components jointly inform each other during training. The neural network drives the induction of symbolic rules by generating representations that are continuously refined, while the symbolic constraints enforce interpretability and consistency. The end-to-end training is performed via stochastic gradient descent with momentum, and the model parameters are updated iteratively over the dataset.

5 Experimental Setup

Experiments are conducted on the SPR_BENCH dataset, which comprises raw sequences of abstract symbols paired with corresponding labels. The dataset is divided into train, development (dev), and test splits. Each input sequence is padded to a fixed length of 6 tokens, and a vocabulary of 18 unique tokens is constructed exclusively from the training data.

Preprocessing involves tokenization and padding, ensuring uniformity in the sequence length for efficient batch processing. The primary evaluation metrics are Shape-Weighted Accuracy (SWA) and Color-Weighted Accuracy (CWA), defined as:

$$\text{SWA} = \frac{\sum_{i=1}^{N} w_i^{\text{shape}} \cdot \mathbf{1}\{\hat{y}_i = y_i\}}{\sum_{i=1}^{N} w_i^{\text{shape}}},$$

$$\text{CWA} = \frac{\sum_{i=1}^{N} w_i^{\text{color}} \cdot \mathbf{1}\{\hat{y}_i = y_i\}}{\sum_{i=1}^{N} w_i^{\text{color}}},$$

The experimental protocol is designed to ensure reproducibility: a fixed random seed of 42 is applied to Python, NumPy, and PyTorch libraries. The computations are performed on a CPU to avoid GPU-induced variability.

The NeuralSPR model architecture is comprised of the following components:

- **Embedding Layer:** Converts token identifiers into dense vector representations. The embedding dimension is set to 32.
- **LSTM Layer:** Processes the sequence of embeddings to capture temporal dependencies, configured with a hidden dimension of 64.
- **Fully Connected Layer:** Maps the final hidden state to the output label space.

Key hyperparameters selected for training include:

Table 2: Experimental Hyperparameters.

Parameter	**Value**
Learning Rate	0.01
Momentum	0.9
Batch Size	64
Epochs	3
Embedding Dimension	32
Hidden Dimension	64

Training is performed using stochastic gradient descent (SGD) with momentum. At each epoch, training loss and development SWA and CWA are monitored. The experimental results reported here indicate a continuous decline in training loss alongside an increase in development SWA and CWA; however, a generalization gap is observed when evaluating on the test set.

6 Results

Our experimental results illustrate the dynamics of training and the challenges faced in generalization. Over a total of three epochs, the training loss was observed to decrease as follows:

$$L_1 = 0.6486 \quad \rightarrow \quad L_2 = 0.3886 \quad \rightarrow \quad L_3 = 0.1806.$$

Concurrently, the development Shape-Weighted Accuracy (SWA) improved markedly. These results are summarized in Table **??**:

Epoch	Training Loss	Dev SWA (%)	Dev CWA (%)
1	0.6486	66.92	65.38
2	0.3886	91.48	91.25
3	0.1806	94.66	94.32

Despite the encouraging development performance, the final evaluation on an unseen test set revealed a Shape-Weighted Accuracy of 67.87% and a Color-Weighted Accuracy of 66.50% . This significant generalization gap indicates that the model, while highly effective on in-domain data, struggles to transfer the learned representations and rule set to novel data distributions.

Figures **??** and **??** illustrate the trends in training loss and development accuracy, respectively. The decline in loss and the rapid rise in the development metric demonstrate the capacity of the network to internalize training patterns; however, the final test performance underscores the need for improved regularization and domain adaptation mechanisms.

Additional ablation studies suggest that the inclusion of the semantic loss term is instrumental in achieving high development accuracy. However, its effectiveness diminishes under test conditions, possibly due to overfitting on the training and development splits. This observation confirms that while the integrated neuro-symbolic approach is theoretically sound, practical challenges remain in bridging the gap between controlled experiments and real-world applications.

7 Discussion

This study presents a detailed investigation into a neuro-symbolic framework for Sequence Pattern Recognition. By coupling an LSTM-based neural network with a symbolic rule induction module, the proposed method strives to combine the advantages of data-driven feature extraction with the interpretability of logical reasoning. The experimental results, obtained on the SPR_BENCH dataset, underscore both the strengths and limitations of the approach.

The training dynamics are characterized by a rapid reduction in loss and a commensurate increase in development SWA, with the neural network quickly fitting to the training data. Nevertheless, the final test performances at 67.87% SWA and 66.50% CWA indicate a substantial generalization gap. This discrepancy between development and test metrics raises several key issues:

1. **Overfitting and Data Distribution:** The higher development accuracy, reaching up to 94.66%, may be attributed to limited variability and potential overfitting on a less diverse dataset. In contrast, the test set likely comprises examples that are either more challenging or drawn from a distribution with different characteristics. This suggests that the neural representations, while effective for training data, fail to generalize to more complex cases.

2. **Regularization and Loss Balancing:** The integration of a semantic loss component alongside the cross-entropy loss is a novel aspect of our approach. However, maintaining the optimal balance between these loss terms is critical; an imbalance may restrict the network's capacity to learn generalized features that are robust against distributional shifts. Future work could explore adaptive loss balancing techniques to better manage this trade-off.

3. **Scope of the Symbolic Component:** The rule induction mechanism, based on ASP, introduces a discrete layer that is highly interpretable. Yet, the complexity of the hypothesis space and the reliance on static background knowledge B may limit the flexibility of the system in adapting to new or unseen patterns. Dynamic rule refinement or the incorporation of reinforcement learning signals could be promising directions to enhance the symbolic module's adaptability.

4. **Evaluation Protocols:** Our current evaluation strategy, which employs fixed train, development, and test splits, provides a preliminary assessment of system performance. Nevertheless, a more robust evaluation involving cross-validation and multiple random seed experiments is necessary to ensure that the observed performance trends are statistically significant.

Several avenues for future research emerge from our findings:

- **Enhanced Regularization:** Incorporating dropout, weight decay, and early stopping mechanisms could help mitigate overfitting, particularly when training on limited or homogeneous datasets.

- **Adaptive Loss Strategies:** Future work could focus on designing adaptive scheduling techniques that dynamically adjust the weight of the semantic loss relative to the cross-entropy loss based on validation performance.

- **Architectural Extensions:** The exploration of alternative neural architectures, such as Transformer-based models, may yield improved performance on longer and more complex sequence pattern recognition tasks.

- **Dynamic Symbolic Adaptation:** Investigating methods for self-supervised or reinforcement learning-based rule induction could allow the symbolic component to evolve during training, accommodating new patterns without compromising interpretability.

- **Comprehensive Evaluation:** Expanding the experimental framework to include multiple datasets and diverse evaluation metrics will provide a more comprehensive understanding of the system's generalization capabilities.

In conclusion, while the proposed NeuralSPR framework demonstrates the feasibility of integrating neural learning with symbolic reasoning, the significant generalization gap observed on the test set highlights critical challenges that must be addressed. Future work will need to focus on refining regularization techniques, exploring adaptive strategies for loss balancing, and dynamically refining the induced rules to improve robustness. The insights gained from this study provide a valuable roadmap for advancing the state-of-the-art in neuro-symbolic integration, paving the way for systems that are both accurate and interpretable in practical, real-world settings.

Overall, our work contributes to the broader understanding of how neural and symbolic components can be combined in an end-to-end framework. While the current instantiation shows promise in terms of in-domain fitting and interpretability, the observed performance degradation on unseen data serves as an important reminder of the complexities involved in real-world deployment. We hope that the challenges delineated herein will stimulate further research that not only bridges the gap between development and test performance but also enhances the scalability and robustness of neuro-symbolic systems in diverse application domains.