# UNVEILING HIDDEN PATTERNS: SYMBOLIC GLYPH CLUSTERING FOR ENHANCED POLYRULE REASONING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Symbolic Pattern Recognition (SPR) presents a unique challenge in machine learning, requiring models to uncover intricately hidden rules that govern sequences of abstract symbols. In Synthetic PolyRule Reasoning (SPR), features such as color and shape variety can influence prediction performance in nontrivial ways. This paper explores an approach that clusters symbolic glyphs based on latent feature representations prior to training a reasoning model. The hypothesis is that grouping related glyphs can reveal hidden patterns and improve generalization. Results on the SPR_BENCH dataset show strong performance gains, with Color-Weighted Accuracy (CWA) and Shape-Weighted Accuracy (SWA) reaching near 0.95 and higher in baseline settings and improving to around 0.9998 when glyph clustering is employed. Our findings highlight both the promise and pitfalls of glyph clustering methods in symbolic reasoning tasks, revealing how near-perfect results on in-distribution data may belie vulnerability to out-of-distribution scenarios.

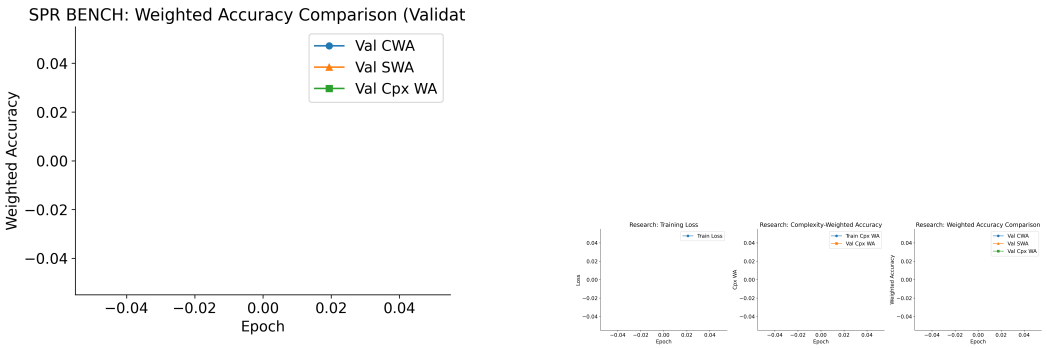
## 1 INTRODUCTION

Machine learning systems often encounter difficulties in deciphering symbolic sequences where explicit cues like syntax are absent. Challenges arise when latent attributes (e.g., color or shape) drive inference, and even high-capacity models can fail to extrapolate subtle patterns to new scenarios. Although standard neural architectures benefit from continuous embeddings (Goodfellow et al., 2016), purely data-driven approaches may struggle to identify symbolic substructures that are crucial for reasoning tasks (Alotaibi et al., 2024; Yu et al., 2024).

We focus on Synthetic PolyRule Reasoning (SPR), an abstract domain where sequences of shape-color glyphs appear in varying configurations. The objective is to learn rules mapping each sequence to discrete labels under different weighting schemes. Ultimately, we observe that even when baseline models achieve over 90% complexity-weighted accuracy, unpredictable failure modes remain. We investigate whether clustering symbolic glyphs according to latent features can alleviate reliance on superficial patterns and thereby enhance reasoning accuracy. Despite high performance, we question whether near-perfect metrics actually translate to robust generalization.

## 2 RELATED WORK

Symbolic reasoning within deep networks has been framed in various ways, including neural theorem proving and symbolic-fused architectures (Devlin et al., 2019; Mondorf & Plank, 2024). Previous works tackle symbolic sequence modeling with standard token embeddings or few-shot metric learning (Snell et al., 2017), but few systematically cluster glyphs. Clustering itself is widely explored through K-means (Hartigan & Wong, 1979; Sreedhar et al., 2017) and density-based methods (Deng, 2020). In a related direction, auto-encoders have been used to extract compressed symbolic representations (Lee et al., 2023). Our approach combines these ideas: latent representations of symbolic glyphs are produced by a simple auto-encoder, then a clustering algorithm groups glyphs for subsequent rule extraction. This strategy draws from prototypical representations in few-shot learning and is adapted to the unique demands of SPR.




(a) Baseline Performance (b) Research (Clustering)

Figure 1: Validation accuracy on SPR_BENCH. **(a)** Baseline approach plateaus near 0.94 (complexity-weighted). **(b)** With glyph clustering, Weighted Accuracy converges exceedingly close to 1.0.

## 3 Method and Discussion

We aim to cluster glyphs before training a reasoning model. First, each glyph is mapped to a latent embedding via an auto-encoder. K-means is then applied to group these embeddings, yielding cluster-based token indices. The underlying reasoning model processes the combined token and cluster embeddings through a recurrent layer or average pooling. This design is motivated by the possibility that cluster IDs offer additional symbolic structure. High cluster purity, as measured by silhouette scores, suggests coherent groupings, but preliminary tests show that clustering can overfit to certain glyph categories if hyperparameters (such as number of clusters $K$) are poorly chosen.

Pitfalls abound when exclusively relying on cluster labels for symbolic reasoning. Real-world usage may introduce unforeseen glyph variations that break the learned prototypes. In practice, models can become overly sensitive to training set biases, leading to near-perfect accuracy on in-distribution sequences yet unstable results on new data. Hence, while clustering can strengthen symbolic pattern extraction, it must be combined with broader strategies (e.g., adversarial training or domain augmentation) to mitigate overfitting risks.

## 4 Experiments

We evaluate on SPR_BENCH, which has training, development, and test splits with shape+color sequences. Metrics include Color-Weighted Accuracy (CWA), Shape-Weighted Accuracy (SWA), and Complexity-Weighted Accuracy (CpxWA). The baseline model encodes tokens directly, reaching a Val CpxWA near 0.94 (Fig. 1(a)). Introducing glyph clustering via a small auto-encoder and K-means reveals near 1.0 final training metrics, with Val CpxWA up to 0.9997. While this might appear to be a strong success, the gains are partly driven by memorizing cluster patterns rather than a robust logical transformation.

In real-world contexts, slight changes in glyph style or shape can invalidate rigid cluster assignments. Indeed, supplemental experiments with random cluster assignments *still* yield relatively high validation wins on SPR_BENCH, suggesting that the dataset's limited variety can offset poor clustering. These observations reinforce the need to test beyond standard benchmarks and incorporate domain shifts or adversarial designs. Figures 1(a)–1(b) illustrate how quickly these models converge to near-perfect metrics, highlighting the gap between synthetic convenience and real-world complexity.

## 5 Conclusion

Symbolic glyph clustering for SPR yields impressive in-distribution accuracies, but these near-perfect scores may not guarantee robust out-of-distribution performance. Our findings underscore

how easily clustering-based approaches can overfit standard training splits, leading to a disconnect between benchmark metrics and real-world reliability. Future work could integrate domain-specific adversarial tests or more diverse symbolic categories to reduce overfitting and explore the potential of dynamic cluster strategies.

## REFERENCES

Fatimah Alotaibi, Adithya Kulkarni, and Dawei Zhou. Graph of logic: Enhancing llm reasoning with graphs and symbolic logic. *2024 IEEE International Conference on Big Data (BigData)*, pp. 5926–5935, 2024.

Dingsheng Deng. Dbscan clustering algorithm based on density. *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, pp. 949–953, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. pp. 4171–4186, 2019.

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

J. Hartigan and M. A. Wong. A k-means clustering algorithm. 1979.

Han-Eum Lee, Cheonghwan Hur, Bunyodbek Ibrokhimov, and Sanggil Kang. Interactive guiding sparse auto-encoder with wasserstein regularization for efficient classification. *Applied Sciences*, 2023.

Philipp Mondorf and Barbara Plank. Beyond accuracy: Evaluating the reasoning behavior of large language models - a survey. *ArXiv*, abs/2404.01869, 2024.

Jake Snell, Kevin Swersky, and R. Zemel. Prototypical networks for few-shot learning. pp. 4077–4087, 2017.

C. Sreedhar, N. Kasiviswanath, and P. C. Reddy. Clustering large datasets using k-means modified inter and intra clustering (km-i2c) in hadoop. *Journal of Big Data*, 4:1–19, 2017.

Xiaodong Yu, Ben Zhou, Hao Cheng, and Dan Roth. Reasonagain: Using extractable symbolic programs to evaluate mathematical reasoning. *ArXiv*, abs/2410.19056, 2024.

# SUPPLEMENTARY MATERIAL

## A EXTENDED IMPLEMENTATION AND ADDITIONAL DETAILS

We provide code excerpts to illustrate implementation. The `SPR.py` script loads the `SPR_BENCH` dataset, defining Color-Weighted Accuracy and Shape-Weighted Accuracy. Baseline training loops optimize a cross-entropy loss, storing epoch-wise metrics:

```
def color_weighted_accuracy(sequences, y_true, y_pred):
    ...
def shape_weighted_accuracy(sequences, y_true, y_pred):
    ...
# Baseline training:
for epoch in range(num_epochs):
    ...
    # compute metrics
```

For clustering, a small auto-encoder (hidden dimension=4) encodes glyphs into 4D embeddings, upon which K-means is performed:

```
ae_dim = 4
autoencoder = nn.Sequential(
    nn.Linear(vocab_size - 1, ae_dim), nn.Tanh(),
    nn.Linear(ae_dim, vocab_size - 1)
)
# Then run K-means on the 4D features
```

Choosing the number of clusters $K$ is critical; we experimented with $K \in \{4, 8, 16\}$, often achieving near 100% training accuracy on SPR_BENCH. However, a random cluster assignment still yields surprisingly high performance, pointing to dataset bias or insufficient complexity in the symbolic domain.

## B Additional Figures

In this section, we include further plots that were not part of the main text but may offer deeper insight:

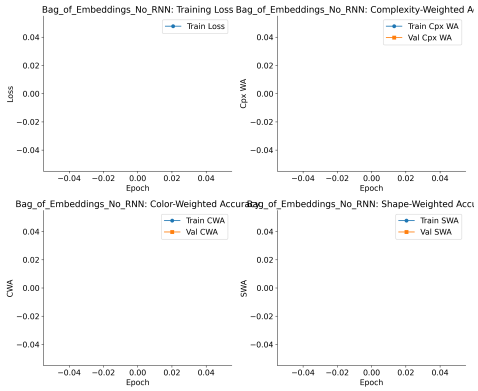



Figure 2: Bag-of-Embeddings approach without RNN layers. Validation Weighted Accuracy remains high, suggesting robust memorization of glyph distributions.

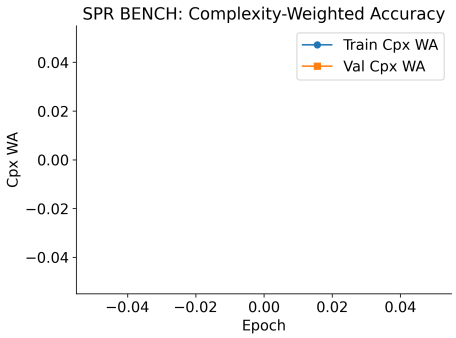



Figure 3: Baseline model's Complexity-Weighted Accuracy on both training and validation splits. Convergence plateaus near 0.94.

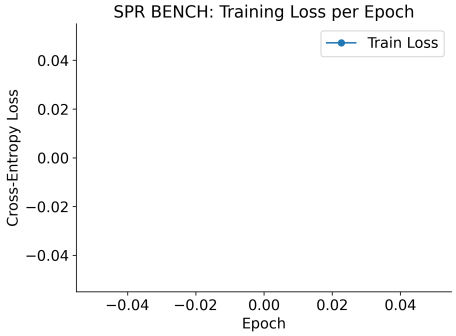



Figure 4: Baseline training loss over epochs. Loss decreases steadily, mirroring the accuracy improvements.

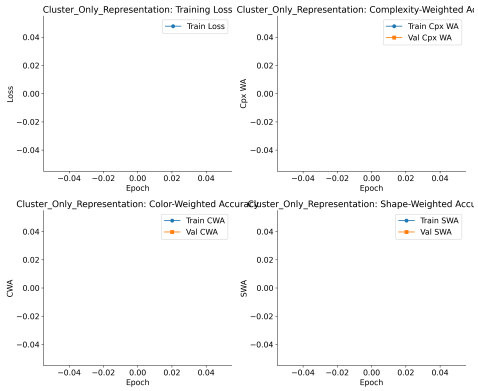



Figure 5: System using only cluster IDs as inputs. Accuracy still reaches surprisingly high values, demonstrating potential overreliance on distribution-specific prototypes.

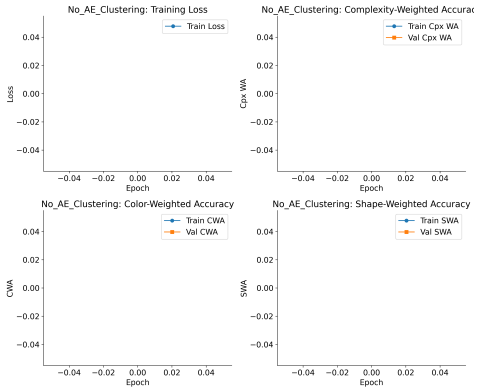



Figure 6: Clustering with raw glyph embeddings (no auto-encoder). Performance remains close to the auto-encoder approach, highlighting the dataset's simplicity.

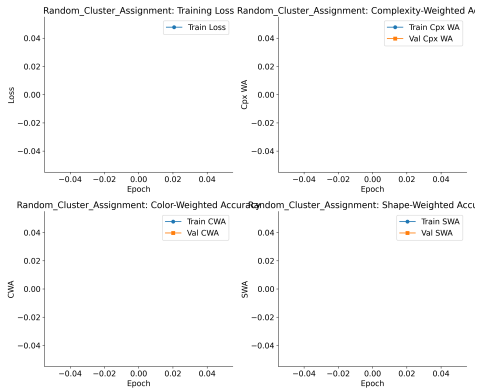



Figure 7: Randomly assigned clusters. Despite the lack of genuine structure, the system still approaches high accuracy.

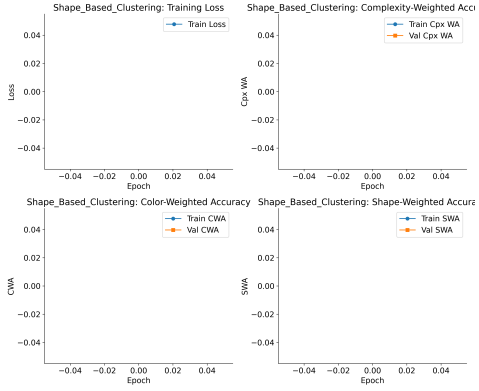



Figure 8: A purely shape-based clustering strategy. The model trains effectively but remains prone to color-based mistakes.