CONTEXT-AWARE CONTRASTIVE LEARNING FOR ENHANCED SYMBOLIC PATTERN RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose leveraging context-aware self-supervised contrastive learning to enhance feature representations for the Synthetic PolyRule Reasoning (SPR) task. The SPR task involves classifying symbolic sequences governed by hidden logical rules, with potential applications in automated reasoning systems. Motivated by successes of prior contrastive frameworks (Chen et al., 2020), we hypothesize that introducing targeted data augmentations and denoising strategies can produce more robust, contextually informed embeddings. We then fine-tune these embeddings on the SPR task, using shape-weighted and color-weighted accuracies as evaluation metrics. Although we aimed to surpass the existing performance of 65.0% SWA and 70.0% CWA, our initial results highlight difficulties in adapting contrastive learning to symbolic data, suggesting further investigation is needed.

1 INTRODUCTION

Symbolic reasoning tasks often demand the capture of subtle compositional rules within sequences. Conventional supervised approaches may struggle to generalize, especially when labeled data are limited or the logic is complex (Sun et al., 2025). Self-supervised learning, particularly contrastive learning, has demonstrated success in capturing robust representations from unlabeled data (Chen et al., 2020). In this work, we investigate a context-aware variant of contrastive learning. Our approach uses tailored augmentations and denoising mechanisms to learn embeddings that retain structural properties of symbolic sequences.

We focus on the Synthetic PolyRule Reasoning (SPR) task, for which we measure performance via shape-weighted and color-weighted accuracies (SWA and CWA). Despite promising theoretical motivations, direct gains over a simple baseline remain inconsistent, underscoring real-world pitfalls that can emerge when adapting self-supervision to symbolic tasks.

2 RELATED WORK

Contrastive learning has become a mainstream approach for representation learning (Chen et al., 2020; Goodfellow et al., 2016). By comparing similar and dissimilar samples, models learn embeddings that maintain semantic relationships. Extensions of contrastive learning have tackled more specialized tasks by integrating denoising or domain-specific augmentations (Lopez-Avila & Suárez-Paniagua, 2024). In symbolic reasoning, capturing compositional structure is critical (Sun et al., 2025). Our work adapts context-aware augmentations, aiming to infuse knowledge about shape and color patterns into the learned representations.

3 BACKGROUND

The Synthetic PolyRule Reasoning (SPR) benchmark, or `SPR_BENCH`, consists of symbolic sequences annotated with class labels. Each sequence includes shapes and colors following a hidden combinatorial logic. The two primary metrics are:

SWA: shape-weighted accuracy, CWA: color-weighted accuracy.

These metrics assign higher weight to predictions on sequences exhibiting greater variety in shapes or colors, highlighting a model's ability to handle more diverse samples.

4 METHOD

We propose a *context-aware contrastive* framework:

1. *Pre-training phase:* We generate augmented samples of unlabeled sequences via token shuffling and denoising-based transformations (Lopez-Avila & Suárez-Paniagua, 2024). Positive pairs share similar shape or color contexts, and negative pairs are more distinct.
2. *Fine-tuning phase:* We add a classification head to the pre-trained encoder and train on labeled sequences. We optimize cross-entropy with a small learning rate to preserve the pre-trained embeddings.

The primary novelty is the emphasis on shape and color context. Sequences that share structure in these features are pushed closer in embedding space, reinforcing relevant compositional cues.

5 EXPERIMENTAL SETUP

Our experiments lean on the official `SPR_BENCH` with 20k training sequences, 5k dev, and 10k test. We use the Adam optimizer with a base learning rate of 1×10^{-4} for pre-training and 5×10^{-5} for fine-tuning. Batch size is set to 256 for the contrastive objective. During pre-training, we monitor contrastive loss on the dev set to steer hyperparameter tuning, including the temperature parameter in the InfoNCE objective.

6 EXPERIMENTS

We compare a simple baseline (supervised only) against our context-aware contrastive approach followed by fine-tuning. Early results showed potential clustering of embeddings (assessed via a t-SNE), yet final classification gains were modest. We observed up to 1–2% improvement in SWA in some runs but inconsistent gains in CWA.

A practical pitfall emerged when certain data augmentations inadvertently distorted logical structures, leading to ambiguous sequence variants. Specifically, heavy token shuffling on short sequences produced almost random outcomes. Also, incorrectly paired augmentations (where color context was lost) degraded performance. These challenges reflect real-world complexities in customizing contrastive learning for symbolic data: while some hypotheses about augmentations may seem reasonable, subtle differences in sequence structure can produce negative or inconclusive results.

In repeated trials, strong sensitivity to hyperparameters and data augmentation level was observed. Minor changes in the degree of noise injection could shift results significantly, underscoring a crucial practical lesson that naive adaptation of general contrastive strategies may not directly translate to better symbolic reasoning performance.

7 CONCLUSION

We presented a context-aware contrastive learning setup to tackle symbolic reasoning under shape and color constraints. Despite theoretical motivations and partial gains, outcomes were neither consistently nor markedly better than a plain supervised baseline. Our findings highlight pitfalls around data augmentation design, negative pair selection, and hyperparameter tuning in symbolic contexts. We encourage further research into specialized self-supervised strategies that account for the nuanced logic of symbolic tasks, guiding the community to carefully validate assumptions on augmentation and representation.

REFERENCES

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

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

Alejo Lopez-Avila and Víctor Suárez-Paniagua. Combining denoising autoencoders with contrastive learning to fine-tune transformer models. *ArXiv*, abs/2405.14437, 2024.

Zhong-Hua Sun, Rui Zhang, Zonglei Zhen, Da-Hui Wang, Yong-Jie Li, Xiaohong Wan, and Hongzhi You. Systematic abductive reasoning via diverse relation representations in vector-symbolic architecture. *ArXiv*, abs/2501.11896, 2025.

Supplementary Material

This section provides additional implementation details, hyperparameters, and usage notes for reproducibility. We also include supplementary code examples.

A Implementation Snippets

Below is an excerpt from our utility for loading `SPR_BENCH` and computing metrics:

```
import pathlib
from typing import Dict
from datasets import load_dataset, DatasetDict

def load_spr_bench(root: pathlib.Path) -> DatasetDict:
    def _load(split_csv: str):
        return load_dataset(
            "csv",
            data_files=str(root / split_csv),
            split="train",
            cache_dir=".cache_dsets"
        )
    dset = DatasetDict()
    dset["train"] = _load("train.csv")
    dset["dev"]   = _load("dev.csv")
    dset["test"]  = _load("test.csv")
    return dset

def count_shape_variety(sequence: str) -> int:
    return len(set(token[0] for token in sequence.strip().split() if token))

def count_color_variety(sequence: str) -> int:
    return len(set(token[1] for token in sequence.strip().split() if len(token)>1))

def shape_weighted_accuracy(sequences, y_true, y_pred):
    weights = [count_shape_variety(seq) for seq in sequences]
    correct = [w if yt == yp else 0 for w, yt, yp in zip(weights,y_true,y_pred)]
    return sum(correct) / sum(weights) if sum(weights)>0 else 0.0

def color_weighted_accuracy(sequences, y_true, y_pred):
    weights = [count_color_variety(seq) for seq in sequences]
    correct = [w if yt == yp else 0 for w, yt, yp in zip(weights,y_true,y_pred)]
    return sum(correct) / sum(weights) if sum(weights)>0 else 0.0
```

Additional Hyperparameters. We tested temperature values in $\{0.07, 0.1, 0.2\}$ for the contrastive loss, finding 0.1 offered a reasonable balance. For token-shuffling augmentations, we tried shuffling up to 50% of tokens in each sequence. Larger shuffling rates at times disrupted the underlying logic. We also tested different masking probabilities (5%–20%), observing that extreme masking occasionally collapsed the representations.