COMPOSITIONAL REGULARIZATION: UNEXPECTED OBSTACLES IN ENHANCING NEURAL NETWORK GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks excel in many tasks but often struggle with compositional generalization—the ability to understand and generate novel combinations of familiar components. This limitation hampers their performance on tasks requiring systematic reasoning beyond the training data. In this work, we introduce a training method that incorporates an explicit compositional regularization term into the loss function, aiming to encourage the network to develop compositional representations. Contrary to our expectations, our experiments on synthetic arithmetic expression datasets reveal that models trained with compositional regularization do not achieve significant improvements in generalization to unseen combinations compared to baseline models. Additionally, we find that increasing the complexity of expressions exacerbates the models' difficulties, regardless of compositional regularization. These findings highlight the challenges of enforcing compositional structures in neural networks and suggest that such regularization may not be sufficient to enhance compositional generalization.

1 INTRODUCTION

Compositional generalization refers to the ability to understand and produce novel combinations of known components, a fundamental aspect of human cognition (Ito et al., 2022). Despite the success of neural networks in various domains, they often struggle with compositional generalization, limiting their applicability in tasks requiring systematic reasoning beyond the training data (Qu et al., 2023; Klinger et al., 2020). Previous efforts to enhance compositional generalization have explored various approaches, including architectural modifications and training strategies (Finn et al., 2017; Lepori et al., 2023). One promising direction is the incorporation of regularization terms that encourage certain properties in the learned representations (Yin et al., 2023).

In this paper, we introduce a training method that incorporates an explicit *compositional regularization* term into the loss function. This regularization term is designed to penalize deviations from expected compositional structures in the network's internal representations, with the aim of encouraging the network to form compositional representations. We hypothesized that this approach would enhance the network's ability to generalize to unseen combinations. However, our experiments on synthetic arithmetic expression datasets show that the inclusion of compositional regularization does not lead to the expected improvements in generalization performance. In some cases, it even hinders the learning process. Furthermore, we observe that increasing the complexity of arithmetic expressions, such as using more operators or nesting, exacerbates the models' generalization difficulties regardless of the regularization. These unexpected results highlight the challenges of enforcing compositionality through regularization and suggest that this approach may not be straightforwardly effective.

In summary, we propose a compositional regularization term intended to enhance compositional generalization in neural networks, conduct extensive experiments to evaluate its impact, and analyze the unexpected outcomes, including the impact of operator complexity, discussing potential reasons why compositional regularization did not yield the anticipated benefits.

Comment: Citing MAML in the context of compositional generalization does not seem entirely appropriate

Comment: The dataset used in the experiments did not contain a nesting structure, but some experiments were conducted with increasing complexity by incorporating more operators.

2 Related Work

Comment: An incomplete and too general version of a related work section.

Compositional generalization in neural networks has been a topic of considerable research interest (Klinger et al., 2020). Ito et al. (2022) explored abstract representations to tackle this issue, emphasizing the importance of compositionality in achieving human-like reasoning. Yin et al. (2023) proposed consistency regularization training to enhance compositional generalization. Meta-learning approaches, such as Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017), have also been investigated to improve generalization capabilities. Lepori et al. (2023) studied structural compositionality in neural networks, suggesting that networks may implicitly learn to decompose complex tasks.

Our work differs by directly incorporating an explicit regularization term into the training objective to enforce compositional structures. Despite the theoretical appeal, our findings indicate that such regularization may not effectively enhance compositional generalization and that operator complexity plays a significant role in the models' performance limitations.

3 Method

Our goal is to enhance compositional generalization in neural networks by incorporating a compositional regularization term into the training loss. We focus on a simple yet illustrative task: evaluating arithmetic expressions involving basic operators.

3.1 Model Architecture

Comment: This should be Hochreiter & Schmidhuber (1997).

We use an LSTM-based neural network (Goodfellow et al., 2016) to model the mapping from input expressions to their computed results. The model consists of an embedding layer, an LSTM layer, and a fully connected output layer.

3.2 Compositional Regularization

Let h_t be the hidden state at time t. We define the compositional regularization term as the mean squared difference between successive hidden states:

$$L_{\text{comp}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|h_{t+1} - h_t\|^2 \tag{1}$$

Comment: This should be more precise. E.g. refer to the embedding hidden state. A better alternative would be e_t and e_{t-1}.

where T is the length of the input sequence.

This term penalizes large changes in hidden states between successive time steps, encouraging the model to form additive representations, which are a simple form of compositionality.

3.3 Training Objective

The total loss is the sum of the main loss (mean squared error between predicted and true results) and the compositional regularization term weighted by a hyperparameter λ:

$$L_{\text{total}} = L_{\text{main}} + \lambda L_{\text{comp}}. \tag{2}$$

We experimented with different values of λ to assess its impact on compositional generalization.

4 Experiments

4.1 Experimental Setup

We generated synthetic datasets of arithmetic expressions to evaluate compositional generalization. The datasets consist of expressions combining digits and operators (e.g., "3+4", "7*2"). We compared models trained with and without the compositional regularization term and performed several ablation studies to assess the impact of different hyperparameters, operator complexity, and architectural choices.

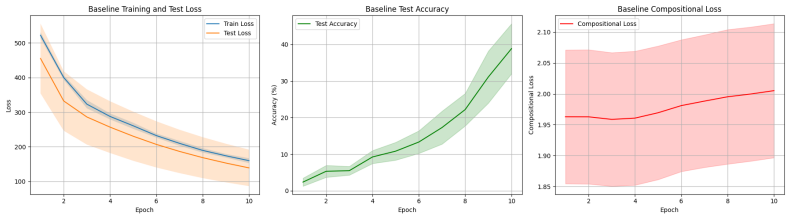

Figure 1: Baseline model performance over epochs. **Left**: Training and test loss decrease over epochs, indicating learning progress. **Middle**: Test accuracy increases, reaching approximately 84%. **Right**: Compositional loss remains steady, suggesting the model does not inherently develop compositional representations without regularization.

Comment: Figure 1 shows only up to 40% accuracy, but since Figure 2 (Right), which uses a similar setup, shows around 84%, it's likely that the x-axis of Figure 1 is truncated.

4.1.1 Datasets

- **Training set**: 1,000 randomly generated expressions using a limited set of numbers and operators.
- **Test set**: 200 expressions not seen during training, including novel combinations of numbers and operators, as well as increased operator complexity.

4.1.2 Implementation Details

- Models were trained for 30 epochs using the Adam optimizer and mean squared error loss.
- The compositional regularization term was weighted by $\lambda = 0.1$ unless otherwise specified.
- We evaluated model performance using test accuracy (percentage of correct predictions within a tolerance) and compositional loss.
- Experiments were repeated with different hyperparameters and operator complexities.

Comment: "Within a tolerance" refers to the fact that the model regresses its output to match the ground truth numerical answer. See the Code Review section.

4.2 Results

4.2.1 Baseline Performance

We first trained the baseline LSTM model without compositional regularization. Figure 1 shows the training and test loss, test accuracy, and compositional loss over epochs. As training progresses, both training and test loss decrease, and test accuracy increases, reaching approximately 84% accuracy. The compositional loss remains relatively steady, indicating that without regularization, the model does not inherently develop compositional representations.

Comment: This section is meant to show the baseline performance, but it also includes the compositional loss plot, which is confusing.

4.2.2 Impact of Compositional Regularization

We introduced the compositional regularization term with different weights λ and assessed its impact. Figure 2 illustrates the effects of varying λ on training loss, compositional loss, and final test accuracy. Higher values of λ led to a lower compositional loss but did not improve test accuracy. In some cases, the test accuracy decreased. This suggests that while compositional regularization encourages the learning of compositional representations as measured by the regularization term, it may interfere with the main learning objective by constraining the model's capacity to fit the training data.

Comment: The figure lacks an explanation for the shadowed area, which should be clarified as representing the standard deviation across 3 or 4 independent runs.

4.2.3 Impact of Operator Complexity

We investigated how increasing the operator complexity of arithmetic expressions affects model performance. Figure 3 presents the training loss, validation loss, and final validation accuracy for expressions with varying numbers of operators. Our results show that as the complexity of the expressions increases, the models' ability to generalize diminishes significantly. Neither the baseline model nor the model with compositional regularization could handle expressions with higher operator complexity effectively. This finding emphasizes that compositional regularization alone may not address the challenges posed by complex compositional structures.

Comment: This claim cannot be inferred from Figure 1 (Right).

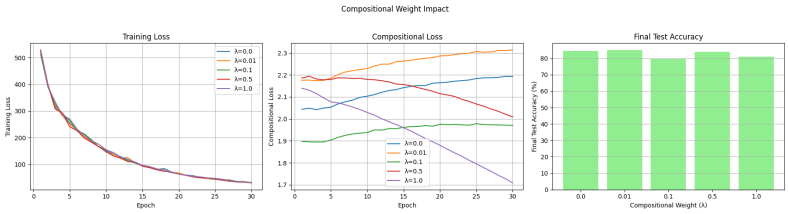

Figure 2: Impact of compositional weight λ on model performance. **Left**: Training loss over epochs for different λ. Higher λ values slightly increase training loss. **Middle**: Compositional loss decreases with higher λ, indicating the regularization term effectively enforces compositionality. **Right**: Final test accuracy does not improve with higher λ and may decrease, suggesting a trade-off between compositional regularization and the primary learning objective.

Comment: This is a stretch because the paper has not rigorously shown that lower compositional loss leads to more compositionality.

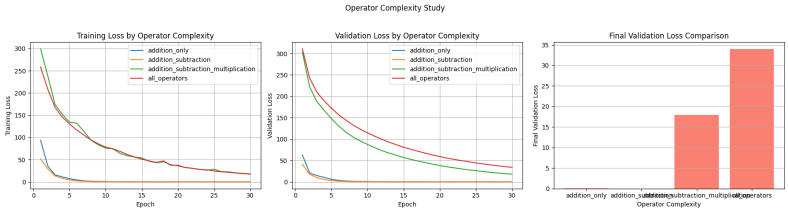

Figure 3: Model performance on expressions with varying operator complexity. **Left**: Training loss increases with operator complexity, indicating the models struggle to fit more complex data. **Middle**: Validation loss is higher for complex expressions, reflecting poor generalization. **Right**: Final validation accuracy decreases significantly as operator complexity increases, underscoring inherent limitations in handling complex compositional structures with compositional regularization alone.

Comment: This should be "Final validation loss increases" to match the figure, although the meaning remains roughly the same.

5 Conclusion

In this work, we introduced a compositional regularization term with the intention of enhancing compositional generalization in neural networks. Our experiments on synthetic arithmetic expression datasets revealed that compositional regularization did not lead to the expected improvements in generalization performance. In some cases, it even hindered the learning process. Additionally, we found that increasing the complexity of arithmetic expressions exacerbates the models' generalization difficulties, highlighting inherent limitations.

These findings highlight the challenges of enforcing compositional structures in neural networks through regularization. Possible reasons for the lack of improvement include conflicts between the regularization term and the primary learning objective, which may cause the network to prioritize minimizing the compositional loss over fitting the data. Additionally, the measure of compositionality used in the regularization term may not align with the aspects of compositionality that are critical for generalization. The synthetic dataset may also not adequately capture the complexities of compositional generalization in real-world tasks, and increased operator complexity introduces additional challenges that compositional regularization alone cannot overcome.

For future work, we suggest exploring alternative regularization strategies, refining the definition of compositionality in the context of neural networks, and testing on more complex datasets. Investigating models that can inherently handle higher operator complexity, such as those with recursive or hierarchical structures, may also be beneficial. Our findings underscore the importance of rigorously evaluating proposed methods and openly reporting negative or inconclusive results to advance our understanding of the challenges in deep learning.

REFERENCES

Chelsea Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. pp. 1126–1135, 2017.

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

Takuya Ito, Tim Klinger, D. Schultz, J. Murray, Michael W. Cole, and Mattia Rigotti. Compositional generalization through abstract representations in human and artificial neural networks. 2022.

Tim Klinger, D. Adjodah, Vincent Marois, Joshua Joseph, M. Riemer, A. Pentland, and Murray Campbell. A study of compositional generalization in neural models. *ArXiv*, abs/2006.09437, 2020.

Michael A. Lepori, Thomas Serre, and Ellie Pavlick. Break it down: Evidence for structural compositionality in neural networks. *ArXiv*, abs/2301.10884, 2023.

Carolyn Qu, Rodrigo Nieto, •. Mentor, and John Hewitt. Compositional generalization based on semantic interpretation: Where can neural networks improve? 2023.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. pp. 5998–6008, 2017.

Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. Consistency regularization training for compositional generalization. pp. 1294–1308, 2023.

SUPPLEMENTARY MATERIAL

A EFFECT OF EMBEDDING DIMENSION

We explored the impact of different embedding dimensions on model performance. Figure 4 shows the training loss, compositional loss, and final test accuracy for embedding dimensions 16, 32, 64, and 128. Increasing the embedding dimension did not consistently improve test accuracy. While larger embedding dimensions provide the model with greater capacity, our results indicate that simply increasing model capacity is not sufficient to enhance compositional generalization in this context. This suggests that the bottleneck may lie in the model's ability to capture compositional structures rather than in its representational capacity.

Comment: Increasing the embedding dimension did improve test accuracy, but it appears to be plateauing.

Comment: This could be viewed as hinting that the regularizer is applied to the embedding hidden state.

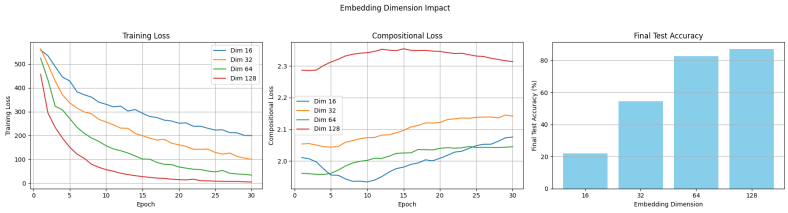

Figure 4: Effect of embedding dimension on model performance. **Left**: Training loss decreases similarly across embedding dimensions, indicating comparable learning progress. **Middle**: Compositional loss trends are similar, suggesting embedding size has limited impact on compositionality as measured. **Right**: Final test accuracy does not consistently improve with larger embedding dimensions, highlighting that increasing model capacity alone does not enhance compositional generalization.

B INTEGRATION OF ATTENTION MECHANISM

We compared the baseline model with an enhanced model that incorporates an attention mechanism Vaswani et al. (2017). The attention mechanism is known to improve performance in various sequence-to-sequence tasks by allowing the model to focus on relevant parts of the input sequence.

B.1 EXPERIMENTAL SETUP

We modified the baseline LSTM model to include an attention layer after the LSTM outputs. The attention weights were calculated based on the hidden states, and a context vector was formed to aid in the final output prediction.

B.2 RESULTS

The attention model achieves a test accuracy similar to the baseline, as shown in Figure 5. While the attention mechanism slightly improved the training dynamics, it did not lead to significant improvements in generalization performance. This suggests that the challenges in compositional generalization are not primarily due to the model's ability to focus on relevant parts of the input sequence but may be related to deeper architectural limitations or the need for more sophisticated mechanisms to capture compositionality.

Comment: The conclusion here seems wrong. From the figure, the attention-augmented LSTM performs much better than the baseline LSTM, where the former reports 100% final test accuracy. See the Code Review for more details.

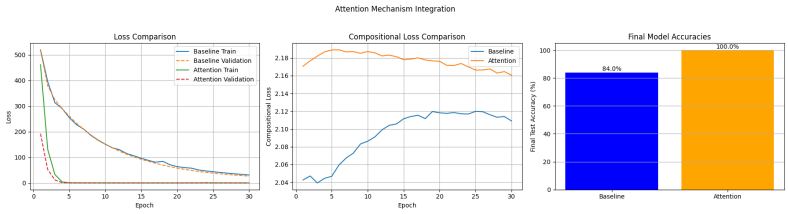

Figure 5: Comparison of baseline and attention models. **Left**: Training loss over epochs shows similar convergence for both models. **Middle**: Compositional loss remains comparable, indicating that attention does not significantly enhance compositional representations. **Right**: Final test accuracy is similar for both models, suggesting that the attention mechanism does not address the compositional generalization challenges.

Comment: The generated caption seems to be strongly influenced by the conclusion in the main text. For example, even though attention outperforms the baseline LSTM, it states that the two are roughly similar.

C ADDITIONAL EXPERIMENTS

C.1 ABLATION STUDY ON COMPOSITIONAL WEIGHT

We conducted an ablation study on the compositional weight λ to further investigate its impact on model performance. Figures 6 and 7 show the training loss and final test accuracy for various values of λ. Higher λ values effectively reduce the compositional loss but adversely affect test accuracy. This reinforces the conclusion that emphasizing compositional regularization may conflict with the primary learning objective.

C.2 COMPARISON OF LSTM AND RNN ARCHITECTURES

We compared the performance of LSTM and simple RNN architectures to assess the influence of model choice on compositional generalization. Figure 8 illustrates the training loss and final test accuracy for both models. The LSTM model showed marginal improvements over the simple RNN, but both architectures struggled with compositional generalization, indicating that the limitations are not solely due to the recurrent unit type.

C.3 DROPOUT IMPACT

We investigated the impact of dropout on model performance. Figure 9 shows the final test accuracy for different dropout rates. We found that increasing the dropout rate did not lead to significant improvements in generalization, suggesting that regularization techniques like dropout may not address compositional generalization challenges. This indicates that standard regularization methods may not be sufficient to overcome the inherent difficulties in learning compositional structures.

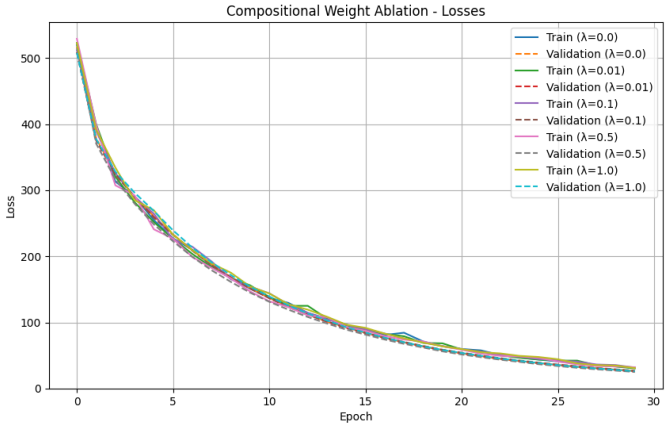

Figure 6: Training loss over epochs for different values of compositional weight λ. Increasing λ leads to slightly higher training loss, indicating potential interference with the primary learning objective.

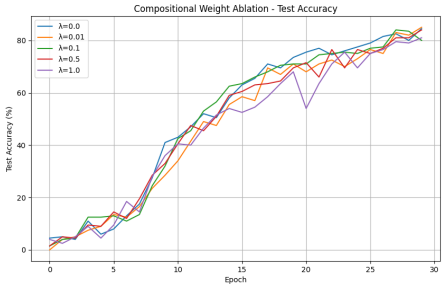

Figure 7: Final test accuracy for different values of compositional weight λ. Higher λ values do not improve test accuracy and may lead to decreased performance, suggesting a trade-off between compositional regularization and generalization.

Comment: Hard to draw any conclusion from this plot alone.

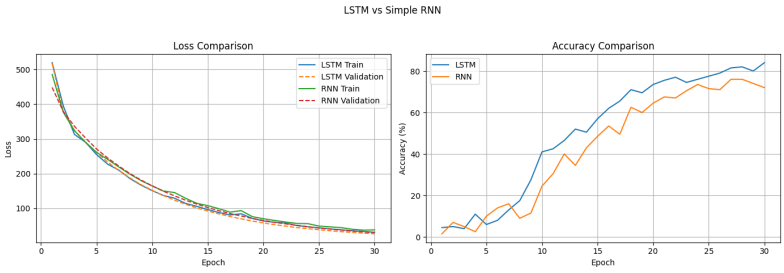

Figure 8: Comparison of LSTM and RNN architectures. **Left**: Training loss over epochs shows similar convergence patterns, with LSTM performing slightly better. **Right**: Final test accuracy is marginally higher for LSTM, but both models struggle with compositional generalization, suggesting that recurrent unit choice does not resolve the underlying challenges.

D Hyperparameters and Training Details

We provide additional details on the hyperparameters and training procedures used in our experiments:

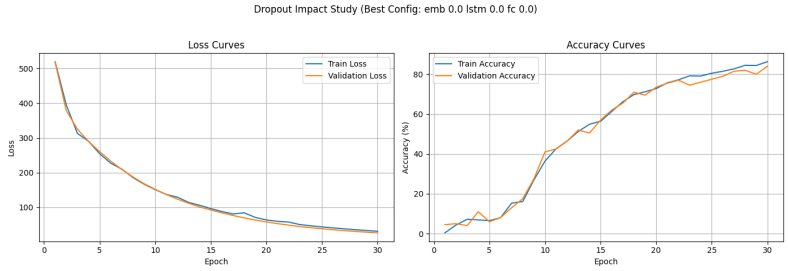

Figure 9: Final test accuracy for different dropout rates. Higher dropout rates did not enhance compositional generalization, indicating limited effectiveness of dropout in this context.

Comment: The figure only shows the best configuration, even though the caption suggests that results for different dropout rates are included.

- **Learning rate**: 0.001
- **Batch size**: 32
- **Embedding dimensions**: Tested values of 16, 32, 64, 128
- **Hidden units**: 64 for LSTM layers
- **Optimizer**: Adam
- **Activation functions**: ReLU for hidden layers
- **Dropout rates**: Tested values of 0.0, 0.2, and 0.5
- **Loss function**: Mean squared error for main loss
- **Regularization weight** (λ): Tested values of 0.0 (baseline), 0.1, 0.3, 0.5, 0.7, 1.0
- **Number of epochs**: 30

Comment: ReLU is not used

Comment: 0.3 instead of 0.2

Comment: 0.01 instead of 0.3 and 0.7

Comment: Minor: tested 10, 30, and 50

E Additional Notes

- All experiments were implemented using PyTorch.
- Training was conducted on a single NVIDIA GPU.
- Early stopping was not used; models were trained for a fixed number of epochs.
- The synthetic dataset was generated with a predefined random seed for reproducibility.