# A Topological Insight To Improve SET

An optimization for the training speed of sparse evolutionary training using topological insight

submitted in partial fulfillment for the degree of master of science

Yuan Su
14534053

Master Information Studies
data science
Faculty of Science
University of Amsterdam

Submitted on 30.06.2022

| | UvA Supervisor |
|---|---|
| **Title, Name** | Hongyun Liu |
| **Affiliation** | Graduate School of Informatics |
| **Email** | h.liu@uva.nl |

Amsterdam Data Science

# ABSTRACT

This research aims to optimize the generation of sparse neural networks using the Sparse Evolutionary Training (SET) algorithm. The SET algorithm has demonstrated its effectiveness in achieving high accuracy and faster training speeds compared to ordinary fully connected neural networks. However, despite these accomplishments, there is still room for improvement in the algorithm. One specific area for optimization is enhancing the level of topological influence within the network. This study focuses on introducing a critical optimization method called 'motif' to the SET algorithm, referred to as SET-motif. The goal is to further enhance the training speed and overall performance of the algorithm. By incorporating this optimization technique, extensive A/B testing has shown that SET-motif is approximately 20% faster with only 1% absolute accuracy drop than SET. The introduction of the set-motif optimization method addresses the need for increased topological influence within the network, leading to enhanced performance. The findings of this research provide valuable insights into optimizing sparse neural network generation and contribute to the ongoing advancements in deep learning architectures.



# KEYWORDS

DEEP LEARNING, SPARSITY, TOPOLOGY, MACHINE LEARNING, SPARSE NEURAL NETWORK


# GITHUB REPOSITORY

https://github.com/Raysyu/Thesis-

# 1 INTRODUCTION

The rapid growth of Artificial Intelligence (AI) has led to increased attention on deep networks, which have had a profound impact not only in academia [10, 19] but also in industries [15]. Sparse Connected Neural Networks (SCNNs) [1] have been introduced as a potential solution to more effectively allocate computational resources and alleviate the computational burden through sparse connections. Instead of fully connected layers, SCNNs utilize sparse connected layers with fewer connections, which can significantly reduce training time and conserve computational power. Previous research [2, 11] has shown the advantages of SCNNs in terms of reducing training time while maintaining relatively high accuracy compared to state-of-the-art solutions.Sparse Evolutionary Training (SET) algorithm is widely recognized as one of the profound techniques in the field of Sparse Convolutional Neural Networks (SCNNs) [14]. SET has demonstrated notable advantages in terms of time efficiency and power consumption, making it an appealing approach. By selectively retaining significant weights in the network, SET can leverage various potentials, including the assessment of feature importance, which is not possible with genetic neural networks. Prior to the introduction of SET, the state-of-the-art SCNNs were merely a byproduct of the end-of-training phase [8], lacking practical benefits but indicating the potential of sparse connections. With the ability to generate and train SCNNs from scratch, SET re-establishes its superiority over conventional deep learning architectures. However, SET still involves a considerable degree of stochastic simulations, motivating the interest in reducing their level to optimize the algorithm further. Importantly, there are few optimizations applied on this algorithm. Therefore, the core of this algorithm has remained largely unchanged since its birth.

Complex tasks often require high-dimensional data to address specific requirements, leading to a significant increase in computational power and training time [21]. Additionally, the prevalent use of multilayer perceptrons (MLPs) with fully connected dense layers [9] has resulted in overparametrization, which contributes to power and computational inefficiency [16]. To overcome these obstacles, alternative approaches have been explored, such as the work referenced in [7]. Deep networks have demonstrated exceptional accuracy in fulfilling various needs. However, there are inherent drawbacks associated with their extensive learning capacity. The vast amount of information learned by these models to meet the desired objectives [16] has made deep networks computationally demanding, rendering them inaccessible to the majority of users.

Pruning connections and subsequently reintroducing them with 'motif' instead of individuals allows for a reduction in the total number of iterations, thereby mitigating the level of stochasticity. It is essential to investigate how this approach compares to genetic SET algorithms in SCNNs, with regards to time consumption and potential compromises in evaluation metrics. It is anticipated that alternative methods may yield lower metrics compared to SET due to the possibility of pruning important weights along with the less significant ones. Consequently, it is essential to consider that if the alternative approach proves effective, the resulting metrics should still be deemed acceptable. Maintaining an acceptable level of performance metrics becomes crucial when evaluating the viability of the alternative approach. In the end, the main research question and its associated sub-questions are as follows:

- How can we optimize SCNN training performance by integrate Motif, a topological concept of complex network, into its pruning?
- To what extent, and aspect, can we improve SCNN performance?

# 2 RELATED WORK

## 2.1 Algorithms to generate and train SCNNs

There have been certain attention paid to training and pruning of SCNNs: The use of Very Sparse Convolutional Neural Networks [3] with deep rewiring techniques has significantly reduced computational requirements and hardware resources. However, this technique may not be suitable for complex tasks and replacing dense neural networks in scenarios such as convolutional neural networks. Additionally, the deep rewiring technique can increase training time due to the additional steps involved. The technique utilizes a topological graph theory to ensure that the SCNN remains topologically equivalent to genetic Artificial Neural Networks (ANNs), which introduces some redundancy in the algorithm.

Boosting pruning plasticity [12] and the lottery ticket hypothesis [6] have explored the extreme end of pruning techniques, showcasing the potential for significant computational power savings on resource-limited platforms. However, the process of pruning and reinitialization can be time-consuming. It is crucial to note that the accuracy of these techniques heavily depends on the level of sparsity achieved, making it challenging to strike a balance between sparsity and accuracy based on the findings of this study. Furthermore, these algorithms focus more on pruning and boosting aspects, with limited emphasis on topology optimization. Moreover, the heavy use of boosting introduces a high level of stochasticity due to its bootstrap method.

## 2.2 Sparse evolution training

Sparse Evolutionary Training (SET) is also a prune-based algorithm introduced by Mocanu et al. [14]. What sets SET apart from other algorithms is its utilization of a random sparse network as the foundation for training. It presents a comprehensive approach for generating and training Sparse Connected Neural Networks (SCNNs). One of the key features of SET is its utilization of topological initialization and real-time weight changes to reduce low-weight connections in the network. This approach provides an intuitive method for achieving high-efficiency SCNNs without compromising accuracy. Figure 1 demonstrates the superiority of SET over Multilayer Perceptron (MLP) in terms of accuracy, while consuming fewer floating points ($n^w$). Comparing SET's accuracy to that of MLP, which utilizes the same number of floating-point operations in $MLP_{FixProb}$, reveals a significant drop in accuracy for MLP. This emphasizes the advantages of SET in achieving both high accuracy and efficiency.

In a study by Kichler [11], SET-generated SCNNs were employed as a dimensionality reduction technique for training traditional machine learning classifiers. The results were remarkable, demonstrating that even with 85% of the features neglected, the accuracy remained high, as depicted in Figure 2. Another study by Evci et al. [5] utilized SET to replace dense connections in state-of-the-art deep network architectures, such as ResNet-50, and datasets like ImageNet. The results showcased the superiority of SET in these scenarios.

Overall, SET exhibits the advantage of stable accuracy, which is not dependent on sparsity, and surpasses other pruning-based algorithms (Section 2.1). However, there is potential for topological optimization within SET. The core aspect of SET involves individually dropping and adding back weights, resulting in a high level of stochasticity as the algorithm evolves over the designated epochs. To potentially reduce the overall training time, a more fuzzy approach can be explored. Instead of singular pruning, the pruning process can be performed on clusters of neurons, introducing a level of fuzziness to the algorithm.

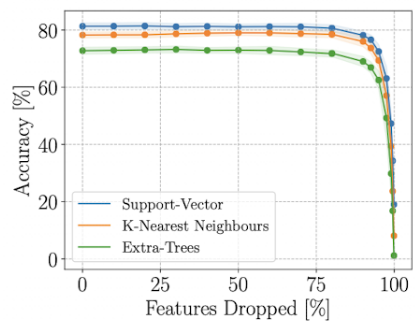


**Figure 2: Feature Dimensions Reduction[11]**

## 2.3 Topology inspiration

In the most cases, deep learning was heavily involved in solving topological problems and the results were promising[[17],[20],[22]]. However, despite topological optimization is widely used in optimizing complex networks[4],the use of it in deep learning architectures is still in exploring, commonly and specifically for SCNNs.

In topology, scholars have researched and classified 'network motifs'[13] from different domains of studies such as biochemistry, neurobiology, ecology and so on. The 'network motifs' are concluded through a figurative amount of networks to be concentrated on 13 exact types. Those 'motifs' are consistent with three nodes with all possible connections either unilateral or bilateral shown in figure. The scholars claim that those 'motifs' could construct most every network.

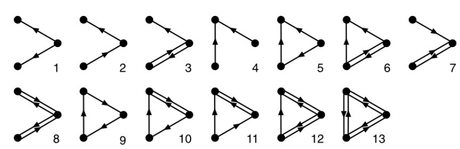


**Figure 3: Network Motifs[13]**

## 2.4 Motif in SET

The SET algorithm is an example of a technique that uses topological optimization. However, there is potential to expand on these topology ideas. The concept of mofits could be useful in further research.

The Erdős–Rényi random graph is a model that is used in SET as a foundation for constructing a SCNN (Sparse Connected Neural Network). However, its computational complexity can be suboptimal. This is because pairwise combination is required to determine whether a connection should be established between two nodes. The time complexity of this process can be expressed as $O(nm)$, where $n$ and $m$ are the number of nodes in each bipartite network, and the adjacent matrix is heavily relied on for the SET algorithm. Although there are ways to generate a random graph that satisfies the need, the adjacent matrix always results in $O(nm)$.

To address this issue, this research proposes an alternative implementation using the configuration model [18]. Despite it is a

| Dataset | Data augmentation | Architecture | Activation | MLP | | $MLP_{FixProb}$ | | SET-MLP | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Accuracy [%] | $n^W$ | Accuracy [%] | $n^W$ | Accuracy [%] | $n^W$ |
| MNIST | No | 784-1000-1000-1000-10 | SReLU | 98.55 | 2,794,000 | 97.68 | 89,797 | 98.74 | 89,797 |
| CIFAR10 | Yes | 3072-4000-1000-4000-10 | SReLU | 68.70 | 20,328,000 | 62.19 | 278,630 | 74.84 | 278,630 |
| HIGGS | No | 28-1000-1000-1000-2 | SReLU | 78.44 | 2,038,000 | 76.69 | 80,614 | 78.47 | 80,614 |

**Figure 1: SET Result Comparison[14]**

long-lived model to generate random graph, it has a faster time complexity $O(Max(n, m))$.

The utilization of motifs as the fundamental units can introduce a level of fuzziness that enhances the speed of the process. However, it is important to note that while there have been identified 13 distinct types of motifs, they are unsuitable in SCNNs due to their inability to adhere to the requisite left-to-right directionality, wherein no connections are allowed within a pair of nodes in the left bipartite graph.shown as figure 4.

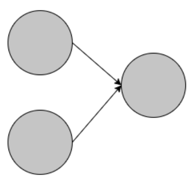

**Figure 4: Designed Motifs**

Thus, a suitable motif comprises a single node and its adjacent node in the left bipartite graph, along with a node in the other part. Both nodes in the left bipartite graph are connected to the node on the right. This motif constitutes the smallest unit in the system, replacing pairwise connections in the configuration random graph and resulting in a new time complexity of $O(Max(n, m))$ using configuration model. Furthermore, in order to reduce the stochasticity during the prune process, motif-as-the-smallest-unit concept will also be adapted thus reduce the iteration to reach convergence.

# 3 METHODOLOGY

## 3.1 Theory design

*3.1.1 SET.* The SET algorithm shown in figure 5, in its original form, employs an initial state that comprises an Erdős–Rényi random graph $g$ , in any but the input and output layer $l_i$ where $l_i \in g$, with $n$ neurons in $l_i$ and $m$ neurons in $L_{i+1}$, a value $\sigma$ which $\{\sigma | \sigma \in R^*\}$ to control the sparsity of the model by generating a random number $\{k | k \in [0, 1]\}$to compare such probability given by:

$$p(g) = \frac{\sigma \cdot (l_i(n) + l_{i+1}(m))}{nm}$$

where $l_i(n) + l_{i+1}(m)$ is the pairwise connection $w$ if $k(n_i, m_j) \leq p(n_i, m_j)$. Based on the researches of [11, 14], both mentioned that a SCNN does not depend on $\sigma$ so much using SET, as the highest accuracy can be always obtained from a range of $\sigma$. The reason is the everytime a random graph is generated, the initial state will have a random start, but when it starts to prune, with enough epochs, the model will converge and produce optimal result. Therefore, in the following experiments, an increasing sequence of $\sigma$ will be applied for each dataset. And each sequence will be repeatably executed to minimize the random results from the initialization state. There The algorithm proceeds by conducting propagations during the first epoch. In every subsequent bipartite layer, the algorithm selectively discards the largest negative or smallest positive weights during the second epoch, which is also controllable by a parameter $\theta$ where $\{\theta | \theta \in [0, 1]\}$. Subsequently, the algorithm reintroduces the same amount of weights into the SCNN during the third epoch, and repeats this process until the desired number of epochs is achieved showed in figure 6. The algorithm can be delineated into two phases, namely the initial phase and the prune phase. To optimize the SET algorithm, both states can be subject to modification.

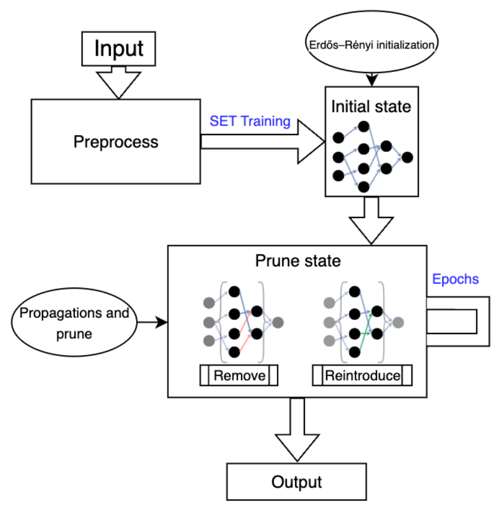


**Figure 5: SET workflow**

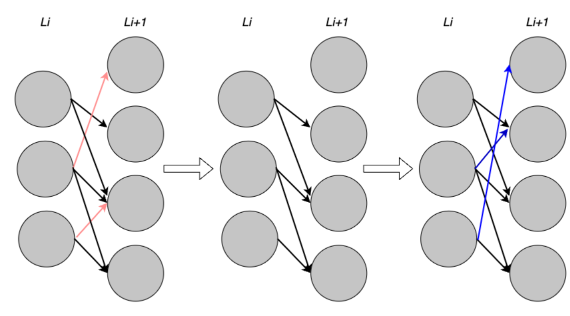


**Figure 6: Prune state**

*3.1.2 Motif.* The proposed SET-motif algorithm utilizes motifs as the smallest unit , where each motif consists of two neurons in the left part of a bipartite structure pointing to a single neuron in the right part. In an ANN architecture of a bipartite section, a motif $w$‘ is show as:

$$l_i(n) + l_{i+1}(m), l_i(n+1) + l_{i+1}(m)$$

where $l_i(n) + l_{i+1}(m)$ is one connection from $n_i$ to $m_j$ and $l_i(n+1) + l_{i+1}(m)$ is the connection from $n_{i+1}$ to $m_j$. In the two aforementioned states, SET-motif will employ motif in generating random graph and pruning.

*3.1.3 Implementation.* After applying motif alternative to the initialization phase, the pseudo-code to represent the Erdős–Rényi random graph is shown in algorithm 1 and alternative used configuration graph in algorithm 2. The time complexity of algorithm 2 has $O(Max(n, m))$ compare to $O(nm)$ in algorithm 2.

**Algorithm 1:** Erdős–Rényi

```
p ← probability;
x ← empty_matix;
For each bipartite layer;
for i in left_part do
    for j in right_part do
        a ← random_number();
        if a ≤ p then
            x_{i,j} ← 1;
        else
            x_{i,j} ← 0;
        end
    end
end
    x.append(x_{i,j});
end
```

After the initialization phase, the SET algorithm proceeds into the pruning state, where weights are eliminated based on propagation results. In each iteration following the initial state, the algorithm identifies a set of a fraction $\theta$ of the total neurons which are the smallest positive and largest negative weights then remove. In the subsequent iteration, an equal number of weights $\theta$ are randomly reintroduced, and this process continues until the desired number of epochs is reached. In contrast, the proposed SET-motif alternative adapts motifs $w$‘ in the pruning stage, where weights are iterated with $w$‘ rather than $w$. A visual representation of the comparative process is illustrated in Figure 7 and 8. To conduct a thorough A/B test, this research introduces an experiment that compares both SET and SET-motif under identical conditions.

**Algorithm 2:** Configuration model with motifs

```
Input: numVertices, probability
Output: x
x = empty_matrix(numVertices, numVertices);
for j = 1 to numVertices do
    if random_number() ≤ probability then
        x[1, j] = 1;
    end
    else
        x[1, j] = 0;
    end
end
for i = 2 to numVertices do
    for j = 1 to numVertices do
        if x[i-1, j] == 1 then
            if random_number() ≤ probability then
                x[i, j] = 1;
            end
            else
                x[i, j] = 0;
            end
        end
        else
            x[i, j] = 0;
        end
    end
end
return x;
```

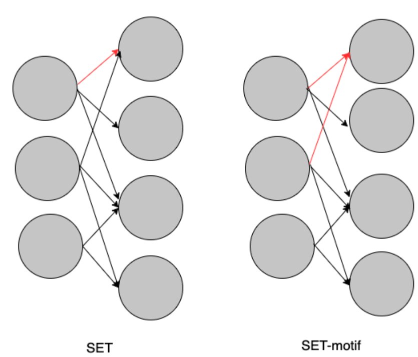


**Figure 7: Remove weights**

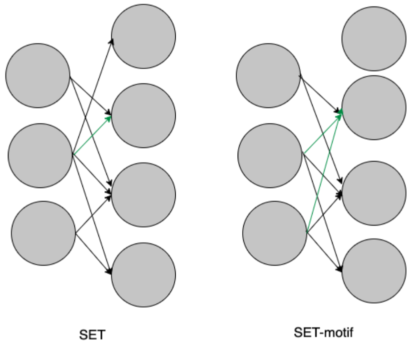


**Figure 8: Reintroduce weights**

## 3.2 Experiment

The main goal of this experiment is to assess the performance of the SET-motif algorithm in comparison to the original SET through the total time consumption and absolute maxium accuracy in complex classification tasks. The total time consumption is to observe if there is superiority in SET-motif than SET, and absolute maxium refers to a global maxium accuracy. In order to carry out this evaluation, we will refer to Kichler's study [11] as it offers comprehensive data that can be utilized for analysis purposes. Although Kichler's research primarily utilized SET as a feature selection tool, only employing a single layer generated and iterated from SET to separate features without fully leveraging its architectural capabilities, the methodology can still be effectively adapted by increasing the number of sparse layers. Referring to the original research on SET, Figure 1 illustrates that a five-layer architecture consistently yielded optimal results for image classification. Therefore, this experiment will adopt a five-layer architecture as well. The experimental data utilized in this research is presented in Table 1 and Table 2. These tables provide the necessary information for conducting a comparative analysis of the SET and SET-motif algorithms.

| Name | #samples | #features | #Classes |
|---|---|---|---|
| Fminist | 70000 | 784 | 10 |
| Lung | 203 | 3312 | 5 |

**Table 1: Data Used**

| Name | #Training | #Testing |
|---|---|---|
| FMNIST | 5000 | 1000 |
| Lung | 134 | 69 |

**Table 2: Train Test Split**

### 3.2.1 *Fashion Mnist.*

The Fashion-Mnist is a graphical data set each of which is a grayscale image with a resolution of 28 * 28 pixels. The dataset contains 10 labels. Some sample data is shown as figure 9.

**Figure 9: FMNIST samples**

The initial step involves shuffling all the samples to randomize their order. Then, each sample is converted into an array and being normalization. Additionally, the labels associated with the samples are encoded using the one-hot coding technique.

The architecture and hyper parameters are presented in table 3 and table 4.

| Layer | Size | Activation |
|---|---|---|
| Input | input | None |
| Layer 1 | 3000 | Relu |
| Layer 2 | 3000 | Relu |
| Layer 3 | 3000 | Relu |
| Output | 10 | Softmax |

**Table 3: Architecture FMNIST**

| Hyperparameter | value |
|---|---|
| epochs | 100 |
| $\theta$ | 0.3 |
| batch size | 40 |
| dropout | 0 |
| learning rate | 0.05 |
| momentum | 0.9 |
| weight decay | 0.0002 |

**Table 4: Hyperparameters FMNIST**

The FMNIST dataset is widely recognized as a benchmark for testing deep learning architectures. It contains a large volume of data with a moderate number of features and is commonly used for classic classification tasks.

In this experiment, both the SET and SET-motif alternative algorithms will be evaluated using a sequence of increasing sparsity levels $\sigma_{FMNIST}$, specifically a sequence of sixteen levels given:

$$\sigma_{FMNIST} \in [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31]$$

Each sparsity level sequence will be executed four times to mitigate the potential impact of random variations caused from random graph in the initialization state. This approach ensures more reliable and robust results during the evaluation process.

### 3.2.2 *Lung.*

The lung dataset contain also greyscale x-ray medical scan of lung which indicate 5 lung conditions which is 5 labels. A sample data is shown in figure 10.

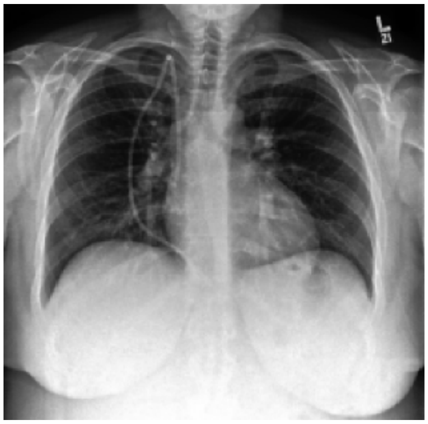

**Figure 10: Lung sample**

The preprocess is similar as FMNIST dataset. All the samples have been converted to arrays and normalized, the labels are one-hot encoded.

The architecture and hyperparameters are shown in table 5 and table 6.

| Layer | Size | Activation |
|---|---|---|
| Input | input | None |
| Layer 1 | 3000 | Relu |
| Layer 2 | 3000 | Relu |
| Layer 3 | 3000 | Relu |
| Output | 5 | Softmax |

**Table 5: Architecture Lung**

| Hyperparameter | value |
|---|---|
| epochs | 100 |
| $\theta$ | 0.3 |
| batch size | 2 |
| dropout | 0 |
| learning rate | 0.01 |
| momentum | 0.9 |
| weight decay | 0.0002 |

**Table 6: Hyperparameters Lung**

The experiment will utilize lung data for training, which will be subjected to an increasing sequence of eight sparsity levels for both the SET and SET-motif algorithms given:

$$\sigma_{lung} \in [1, 2, 3, 4, 5, 6, 13, 32]$$

Each method will be executed four times to ensure reliable results. Although the lung data has a smaller sample size compared to FMNIST, it contains a comparable larger number of features.

In order to examine the robustness of the SET-motif alternative algorithm, the training on FMNIST data will be intensified by increasing the number of repetitions from four to sixteen times. This stress test aims to evaluate the algorithm's performance under more challenging conditions and ascertain its resilience to variations in the training process.

## 3.3 Evalutation

The experiment will be assessed by calculating the total time spent for each dataset run in both SET and SET-motif algorithms across their designated runs. This evaluation will involve comparing the absolute time difference between the two. To ensure that SET-motif produces an equally good model, the accuracy of both SET and SET-motif will be measured and recorded every 10 epochs per $\sigma$. The highest accuracy achieved among all sigma values will be reported, along with the range of accuracy observed for each $\sigma$.

All the code will be executed on a Apple MacMini under MacOS Ventura 13.4 with 8g memory and 7g metal GPU. Which is to showcase capability of SET and its variant can be done in a low-resourse environment.

# 4 RESULT

## 4.1 Time consumption

After running both methods, SET-motif is not converging at all. Therefore, motif is removed from the initialization phase. The result of this experiment which only had motifs in prune phase is shown in table 7 and table 8 of time spent with the absolute highest accuracy.

| Name | Accuracy | Total Time(s) |
|---|---|---|
| FMNIST | 0.827 | 23309 |
| Lung | 0.985 | 754 |

**Table 7: SET**

| Name | Accuracy | Total time(s) |
|---|---|---|
| FMNIST | 0.823 | 19522 |
| Lung | 0.970 | 595 |

**Table 8: SET-motif**

The comparative analysis reveals that the SET-motif alternative algorithm exhibits a notable advantage in terms of time efficiency, requiring less time compared to SET while maintaining high accuracy levels. In the FMNIST dataset experiment, each sequence of SET-motif is approximately 947 seconds faster than SET which is about 17% faster with 0.4% absolute accuracy dropped. Similarly, in the lung data experiment, each sequence of SET-motif is around 40 seconds faster than SET which is about 21% faster with a 1.5% absolute accuracy dropped. These time savings are recorded over four repetitions of the experiments.

Furthermore, the robustness stress test results, as presented in Table 9, demonstrate a significant difference between SET and SET-motif. The table provides insights into the performance of both algorithms under more challenging conditions, highlighting the superior efficiency of SET-motif compared to SET.

| SET | Accuracy | Total time(s) |
|---|---|---|
| FMNIST | 0.841 | 100207 |
| SET-motif | Accuracy | Total time(s) |
| FMNIST | 0.835 | 77965 |

**Table 9: Robustness Result comparison**

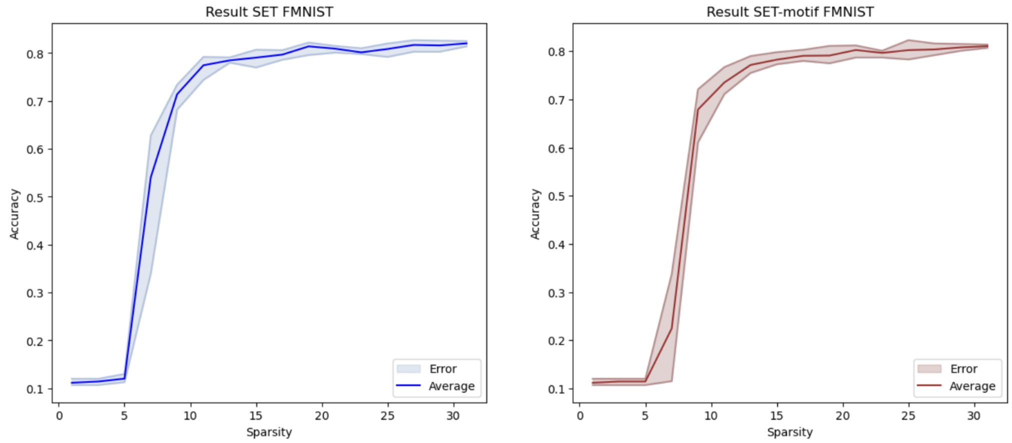


**Figure 11: Accuracy evolution comparison between SET and SET-motif in FMNIST (The error margin is surrounded by the maximum and minimal at that sparsity in all sequence runs. The average is the average value at the sparsity in all sequence runs.)**

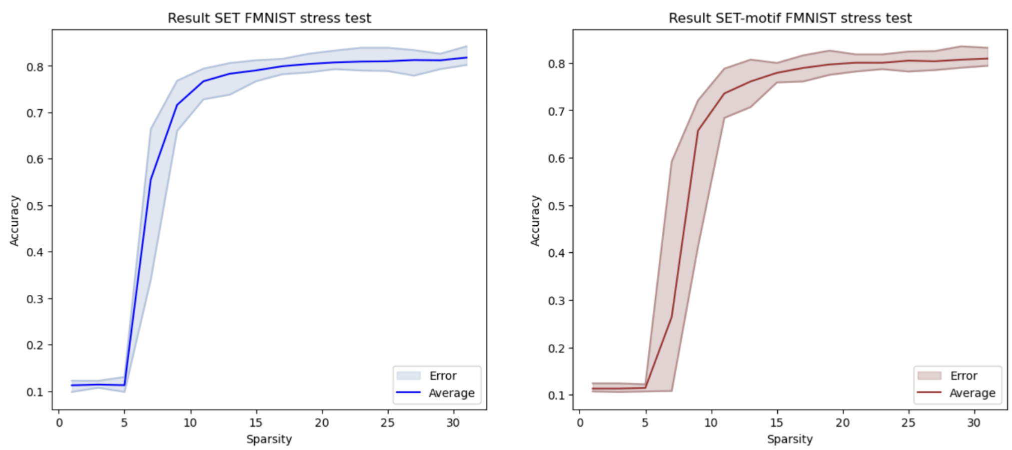


**Figure 12: Accuracy evolution comparison between SET and SET-motif FMNIST stress test (The error margin is surrounded by the maximum and minimal at that sparsity in all sequence runs. The average is the average value at the sparsity in all sequence runs.)**

The time difference is about 1390s per sequence, which is about 22% faster with a 0.6% absolute accuracy dropped in SET motif than SET.

## 4.2 Accuracy evolution

In FMNIST dataset, the accuracy per $\sigma$ of both SET and SET-motif is shown in figure 11.

It appears that the fuzziness introduced in the model requires a slight increase in sparsity for the convergence to occur. However, this increase in sparsity does not seem to be significant. Additionally, when comparing the accuracy plateau in both cases, the absolute maximum accuracy achieved is slightly lower than that of the SET of SET-motif, and this difference can be caused by the fuzziness.

The stress test experiment, which utilized FMNIST, exhibits similar behavior as shown in Figure 12. While the error margin slightly increases with sparsity in the SET case, the SET-motif demonstrates a decreased error margin when the accuracy reaches the plateau but slightly larger margin when the accuracy is climbing.

In the experiment with lung dataset, the result is shown in figure 13. The over all behavior is similar to the experiment carried out with FMNIST dataset, except the error margin is larger in SET-motif than SET. The highest accuracy occurs in a smaller sparsity in both cases.

Overall, combining three experiment, it can be conclude that SET-motif can be a topological optimization to SET as the excellence of accuracy is remained with a bit fuzziness, the total running time is more efficient.

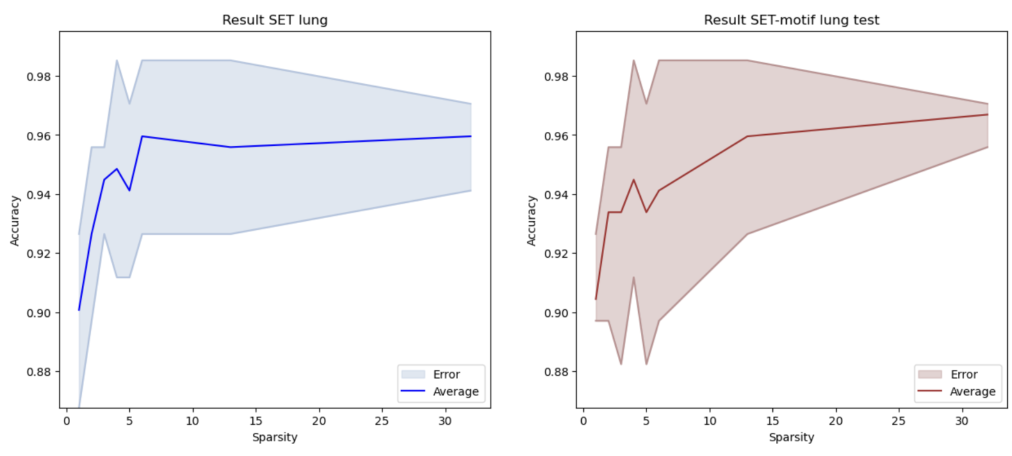


**Figure 13: Accuracy evolution comparison between SET and SET-motif in lung (The error margin is surrounded by the maximum and minimal at that sparsity in all sequence runs. The average is the average value at the sparsity in all sequence runs.)**

## 5 DISCUSSION

The implementation of the configuration graph was successful; however, the model did not show any convergence. This lack of convergence may be attributed to the possibility that the underlying network lacks certain crucial connections, thereby hindering the normal transmission of propagations. Unfortunately, the exact reason remains unknown within the constraints of the time limit imposed on this study. If further implementation of the configuration graph is pursued, it is anticipated that the total time required for the training process could be further reduced.

Additionally, an attempt was made to train the Madelon dataset, an artificial dataset containing noise, following Kichler's research methodology. Kichler noted that his model did not yield positive results, as the initial layers of his experiment failed to effectively separate features. While using the original SET method as a binary classification technique yielded satisfactory results for the Madelon dataset, switching to SET-motif resulted in the model's failure to converge at a low sparsity level. Moreover, even when convergence was achieved, the accuracy remained around 50%. Despite incorporating some simple fine-tuning techniques, no significant improvement was observed. Consequently, the Madelon dataset was excluded from this study.

Finally, due to the time limit, only limited datasets are tested, the ultimate robustness is not determined. Therefore, for further reseach, testing more and preferably unbalanced data to further investigate the robustness.

## 6 CONCLUSION

SCNNs as its advantages of energy and training efficiency attract lots of attention in the field. In this work, we integrate a topological optimization based on Motif with SET, resulting in improved performance. By incorporating motif into the pruning phase of SET to reduce stochasticity, it is expected that fewer iterations will be required to achieve convergence while maintaining relatively high accuracy. Although the introduction of motif brings some level of fuzziness, as pruning additional weights may remove important connections, the decrease in accuracy is small and falls within an acceptable range.

To evaluate the performance of SET and SET-motif, an A/B test is conducted in which both algorithms are tested under the same conditions. Using the FMNIST dataset, the regular run with SET-motif is found to be approximately 17% faster and 0.4% absolute accuracy droped, the lung dataset run is 21% faster and 1.5% absolute accuracy dropped, and the FMNIST stress run is 22% faster and 0.6% absolute accuracy dropped compared to SET alone. These results indicate that SET-motif achieves an overall improvement in speed, performing roughly 20% faster than SET and around 1% accuracy dropped.

Based on the results presented in this research, it can be concluded that SET-motif improves the training performance of SET. The SET-motif approach offers a faster training experience compared to SET, and the observed drop in accuracy is insignificant. Therefore, adopting SET-motif can be beneficial in terms of improved training efficiency without compromising the overall accuracy of the model.

## REFERENCES

[1] Arash Ardakani, Carlo Condo, and Warren J. Gross. 2017. Sparsely-connected neural networks: Towards efficient VLSI implementation of Deep Neural Networks. https://arxiv.org/abs/1611.01427

[2] Zahra Atashgahi, Ghada Sokar, Tim van der Lee, Elena Mocanu, Decebal Constantin Mocanu, Raymond Veldhuis, and Mykola Pechenizkiy. 2020. Quick and Robust Feature Selection: the Strength of Energy-efficient Sparse Training for Autoencoders. https://doi.org/10.48550/ARXIV.2012.00560

[3] Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. 2018. Deep rewiring: Training very sparse Deep Networks. https://arxiv.org/abs/1711.05136

[4] Zhang Bin, Gan Zhi-chun, Chen wen, Hu Qiang-qiang, and Hou Jian-feng. 2019. Topology Optimization of Complex Network based on NSGA-II. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Vol. 1. 1680–1685. https://doi.org/10.1109/IAEAC47372.2019.8997597

[5] Utku Evci, Trevor Gale, Jacob Menick, P. S. Castro, and Erich Elsen. 1970. [PDF] rigging the lottery: Making all tickets winners: Semantic scholar. https://www.semanticscholar.org/paper/Rigging-the-Lottery%3A-Making-All-Tickets-Winners-Evci-Gale/2e3002f131e1815bda7a10303eff97f79dea01ec

[6] Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. https://arxiv.org/abs/1803.03635

[7] Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. https://arxiv.org/abs/1510.00149

[8] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both Weights and Connections for Efficient Neural Network. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. https://arxiv.org/abs/1512.03385

[10] Alex Krizhevsky Google Inc, Alex Krizhevsky, Google Inc, Google IncView Profile, Ilya Sutskever Google Inc, Ilya Sutskever, Geoffrey E. Hinton OpenAI, Geoffrey E. Hinton, OpenAI, OpenAIView Profile, and et al. 2017. ImageNet classification with deep convolutional Neural Networks. https://dl.acm.org/doi/10.1145/3065386

[11] Neil Kichler. 1970. Robustness of sparse MLPS for supervised feature selection. http://essay.utwente.nl/86886/

[12] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, Decebal Constantin Mocanu, and et al. 2022. Sparse training via boosting pruning plasticity with neuroregeneration. https://arxiv.org/abs/2106.10404

[13] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, and Uri Alon. [n. d.]. Network motifs: Simple building blocks of complex networks | science. https://www.science.org/doi/10.1126/science.298.5594.824

[14] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. 2018. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by Network Science. https://www.nature.com/articles/s41467-018-04316-3#citeas

[15] Harry A. Pierson and Michael S. Gashler. 2017. Deep learning in robotics: A review of recent research. https://arxiv.org/abs/1707.07217

[16] Karthik A. Sankararaman, Soham De, Zheng Xu, W. Ronny Huang, and Tom Goldstein. 2020. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. https://arxiv.org/abs/1904.06963

[17] Hidenori Sasaki and Hajime Igarashi. 2019. Topology Optimization Accelerated by Deep Learning. *IEEE TRANSACTIONS ON MAGNETICS* 55, 6 (JUN 2019). https://doi.org/10.1109/TMAG.2019.2901906

[18] Albert-La szlo Baraba si and Reka Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (oct 1999), 509–512. https://doi.org/10.1126/science.286.5439.509

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. https://arxiv.org/abs/1706.03762

[20] Mowei Wang, Yong Cui, Shihan Xiao, Xin Wang, Dan Yang, Kai Chen, and Jun Zhu. 2018. Neural Network Meets DCN: Traffic-driven Topology Adaptation with Deep Learning. *PROCEEDINGS OF THE ACM ON MEASUREMENT AND ANALYSIS OF COMPUTING SYSTEMS* 2, 2 (JUN 2018). https://doi.org/10.1145/3224421

[21] Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. https://arxiv.org/abs/1812.04224

[22] Shuai Zhang, Bo Yin, Weiyi Zhang, and Yu Cheng. 2022. Topology Aware Deep Learning for Wireless Network Optimization. *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS* 21, 11 (NOV 2022), 9791–9805. https://doi.org/10.1109/TWC.2022.3179352