



## RESEARCH PAPER

# Deep GreenAI: Effective Layer Pruning Method for Modern Neural Networks

Ian Pons  [Universidade de São Paulo | [ian.pons@usp.br](mailto:ian.pons@usp.br)]

Bruno Yamamoto  [Universidade de São Paulo | [brunolyamamoto@usp.br](mailto:brunolyamamoto@usp.br)]

Artur Jordão  [Universidade de São Paulo | [arturjordao@usp.br](mailto:arturjordao@usp.br)]

 Escola Politécnica, Universidade de São Paulo, Av. Professor Luciano Gualberto, nº 380, Cidade Universitária, São Paulo, SP, 05508-010, Brazil.

**Abstract.** Deep neural networks have been the predominant paradigm in machine learning for solving cognitive tasks. Such models, however, are restricted by a high computational overhead, limiting their applicability and hindering advancements in the field. Extensive research demonstrated that pruning structures from these models is a straightforward approach to reducing network complexity. In this direction, most efforts focus on removing weights or filters. Studies have also been devoted to layer pruning as it promotes superior computational gains. However, layer removal often hurts network predictive ability (i.e., accuracy) at high compression rates. This work introduces an effective layer-pruning strategy that meets all underlying properties pursued by pruning methods. Our method estimates the relative importance of a layer using the Centered Kernel Alignment (CKA) metric, employed to measure the similarity between representations of the unpruned model and a candidate subnetwork for pruning. We confirm the effectiveness of our method on standard architectures and benchmarks, in which it outperforms existing layer-pruning strategies and other state-of-the-art pruning techniques. Specifically, we remove more than 75% of computation while improving predictive ability and reducing  $CO_2$  emissions required for training by 80%, taking an important step towards GreenAI. At higher compression regimes, our method exhibits negligible accuracy drop, while others notably deteriorate it. Apart from these benefits, our pruned models exhibit robustness to adversarial and out-of-distribution samples.

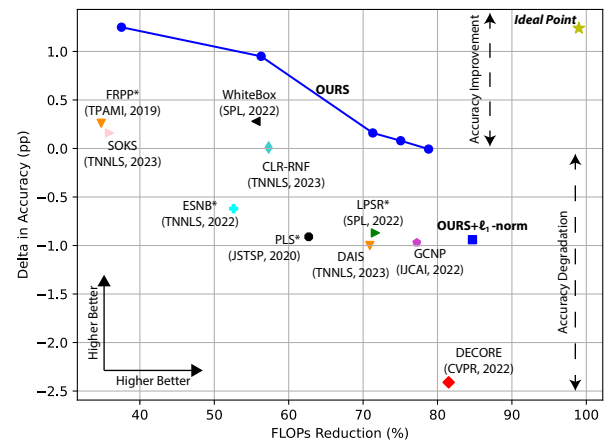
**Keywords:** Green AI, Layer Pruning, Efficient Deep Learning, Similarity Metric

Received: 16 May 2025 • Accepted: 19 June 2025 • Published: 16 July 2025

## 1 Introduction

It is well known that deep neural networks are capable of obtaining remarkable results in various cognitive fields, often outperforming humans from image recognition to complex games such as chess and go [Silver *et al.*, 2016; Han *et al.*, 2023]. However, this performance comes with high computational cost and storage demand. Advances in the foundation model paradigm – large deep learning models trained on a broad range of data with the capacity to transfer its knowledge to unseen (downstream) tasks – have further intensified the resource-intensive nature of the field, as large models play a crucial role in transferring knowledge to downstream tasks [Bommasani *et al.*, 2021; Amatriain, 2023]. To mitigate the above issues, compression techniques are becoming more popular due to their positive results in improving the resource demand of deep models [He and Xiao, 2024; Xu and McAuley, 2023; Wang *et al.*, 2022]. Among the most promising techniques, pruning emerges as a straightforward approach capable of enhancing different performance metrics such as Floating Point Operations (FLOPs), memory consumption and number of parameters. At the heart of a pruning technique lies the task of accurately estimating the importance of structures that compose a model and subsequently removing the least important ones.

Studies classify pruning into unstructured and structured categories [He and Xiao, 2024; Xu and McAuley, 2023; Wang *et al.*, 2022]. The first removes individual connections (weights), while the latter focuses on eliminating entire structures such as filters and layers. Despite achieving high



**Figure 1.** Comparison with state-of-the-art on the popular ResNet56 + CIFAR-10 setting. Overall, our method obtains the best compromises between accuracy and computational reduction (estimated by Floating Point Operations – FLOPs). Specifically, our method dominates existing layer pruning methods (indicated by symbol \*) by a remarkable margin. Compared to state-of-the-art pruning techniques, our method removes more than 75% of FLOPs without hurting accuracy (sometimes improving it). Other methods, however, degrade accuracy when operating at these high FLOP reduction regimes. Since our method is orthogonal to modern structured filter pruning, we can combine them to achieve even higher computation gains (e.g., Ours +  $\ell_1$ -norm). The behavior shown in this figure is consistent across other benchmarks and architectures.

compression rates, unstructured approaches promote theoretical inference speed-up only, requiring specific hardware to handle sparse matrix computations to obtain practical speed-up [Wang *et al.*, 2022; Xu and McAuley, 2023]. On the other hand, structured pruning facilitates practical accelera-

tion without any hardware/software constraints. Furthermore, its advantages extend beyond computational gains: it acts as a regularization mechanism that improves generalization and robustness [Zhao and Wressnegger, 2023; Bair *et al.*, 2024].

Most efforts on structured pruning strategies focus on eliminating small structures and are often optimized for standard metrics such as FLOP and parameter reduction [He and Xiao, 2024; Xu and McAuley, 2023]. Unfortunately, recent studies suggest that these metrics may correlate weakly with inference time [Dehghani *et al.*, 2022; Vasu *et al.*, 2023]. On the other hand, layer pruning reduces network depth, which directly addresses model latency while also providing all the benefits of filter pruning, such as FLOP and parameter reduction, without specialized software or hardware [Zhang *et al.*, 2022; Zhou *et al.*, 2022]. The idea behind pruning layers is not novel and dates back to 2016 [Veit *et al.*, 2016]. Efforts in this direction, however, either apply simple filter criteria and combine (e.g., average) the scores to compose the importance of a layer [Jordão *et al.*, 2020; Zhang *et al.*, 2022] or solve a (computationally expensive) multi objective optimization [Zhou *et al.*, 2022]. Therefore, one of the challenges in layer removal is developing a criterion capable of accurately ranking the importance of all layers. It turns out that existing criteria operate well on small structures but may be inadequate when applied to large ones, primarily because of varying magnitudes (i.e.,  $\ell_1$ -norm and its variations) exhibited by layers [Zhang *et al.*, 2022; Jordão *et al.*, 2023]. Additionally, recent studies highlight that different layers play a distinct role in the expressive power and training dynamics of deep models [Zhang *et al.*, 2022; Masarczyk *et al.*, 2023; Chen *et al.*, 2023]. Such factors suggest that simple criteria are unable to characterize all these underlying properties exhibited by layers. Lastly, a layer-pruning method must inherit a fundamental requirement of pruning techniques: remove structures without significantly compromising predictive ability.

To meet the aforementioned requirements and achieve computational-friendly models, we propose a novel layer pruning method. Our method relies on the hypothesis that similar representations between a dense (unpruned) network and its optimal sparse (pruned) candidate indicate lower relative importance. Existing evidence support this idea, revealing that layers share similar representations [Zhang *et al.*, 2022; Masarczyk *et al.*, 2023]. By eliminating unimportant layers, we can preserve predictive capability and reduce computational demand.

For this purpose, we employ Centered Kernel Alignment (CKA) due to its effectiveness and flexibility in measuring similarity between two networks [Kornblith *et al.*, 2019; Masarczyk *et al.*, 2023]. Leveraging CKA, the overall of our method is the following. Given a dense network, we first extract its representation from some input examples. Here, representation refers to the feature maps of the layer just before the classification layer. Then, we create a temporary pruned model by removing a candidate layer. Building upon previous works [Veit *et al.*, 2016; Zhang *et al.*, 2022], at this step, we avoid any fine-tuning or parameter adjustment, since modern architectures are robust to single layer removal and perturbations. Afterward, for each temporary model ( $n$  layers,  $n$  candidates), we compare their representations with the

original network using CKA. Finally, we select the temporary pruned network that exhibits the closest similarity with the original unpruned model. Performing this process for all candidates in a trained network allows us to estimate its relative importance without fine-tuning.

**Contributions.** We highlight the following key contributions. First, we propose a novel pruning criterion that leverages an effective similarity representation metric: CKA. To the best of our knowledge, we are the first to explore CKA as a pruning criterion, as previous works widely employ it for comparing network representations. Powered by this criterion, we develop a layer-pruning method that removes entire layers from neural networks without compromising predictive ability. Such a result is possible since our criterion identifies unimportant layers – layers that, when removed, preserve similarity regarding the original model. Second, unlike most existing layer-pruning criteria that fail to capture underlying properties of layers, our method effectively assigns layer importance and thus prevents model collapse (see Figure 1). Besides, it is efficient and scales linearly as a function of the network depth. Third, we outperform state-of-the-art pruning methods by a notable margin. We believe our results open new opportunities to prune through the lens of emerging similarities metrics [Duong *et al.*, 2023] and encourage further efforts on layer pruning.

Through extensive experiments on standard architectures and benchmarks, we demonstrate that our method outperforms state-of-the-art pruning approaches. Specifically, it surpasses existing layer-pruning strategies by a large margin. In particular, as we increase the levels of FLOP reduction, most layer-pruning methods fail to preserve accuracy. Our method, on the other hand, successfully maintains accuracy while eliminating more than 75% of FLOPs. At reductions exceeding 80%, our method exhibits negligible accuracy drop, whereas other state-of-the-art techniques are unable to achieve similar performance without compromising accuracy roughly  $2\times$  more. We also demonstrate that our method preserves generalization in out-of-distribution and adversarial robustness scenarios, which is crucial for deploying pruned models in security-critical applications such as autonomous driving. In terms of Green AI [Lacoste *et al.*, 2019; Strubell *et al.*, 2019], our method reduces the carbon emissions required in the training/fine-tuning phase by up to 80.85%, representing an important step towards sustainable AI. Code available at: [github.com/IanPons/CKA-Layer-Pruning](https://github.com/IanPons/CKA-Layer-Pruning)

**Publications.** The results from this research have been accepted at the International Conference on Pattern Recognition (ICPR 2024, Oral presentation)<sup>1</sup> and at a workshop from the International Conference on Machine Learning (WANT-ICML 2024)<sup>2</sup>, both where the student is the first author.

**Student Role within the Broader Project.** The present work is part of a broader scientific initiation project (FAPESP grant #2023/11163-0 and CNPq grant#402734/2023-8) focusing on reducing the environmental impact of artificial Intelligence (GreenAI). In these projects, Ian Pons contributed to designing a novel method for reducing the computational cost of neural networks, including its code implementation and paper

<sup>1</sup>[https://link.springer.com/chapter/10.1007/978-3-031-78169-8\\_28](https://link.springer.com/chapter/10.1007/978-3-031-78169-8_28)

<sup>2</sup><https://openreview.net/pdf?id=7DPNITf7ui>

writing. Additionally, there are two ongoing Master’s theses expanding the ideas of this research with works under review of international conferences in the field.

## 2 Related Work

The main (and most challenging) task of pruning is to estimate the relative importance of a given structure to differentiate between those essential for predictive ability and the less important ones. A popular criterion focuses on the magnitude of weights, namely  $\ell_p$ -norm. Researchers extensively explore these criteria in the context of the lottery ticket hypothesis and pruning at initialization [Wang et al., 2022]. Despite their simplicity, previous works pointed out pitfalls in these criteria [Zhang et al., 2022; Huang et al., 2021; He et al., 2019]. For example, Huang et al. [2021] argued that constraining the analysis to surrounding structures, as  $\ell_1$ -norm does, incurs a low variance of importance scores, hindering unimportant structure search. Furthermore, comparing norms across layers becomes impractical, as different layers exhibit distinct magnitudes, posing a challenge for global pruning (i.e., ranking all structures at once) [Zhang et al., 2022; Jordão et al., 2023]. These issues have motivated efforts towards more elaborate criteria [Shen et al., 2022]. Taking the work by Lin et al. [2020] as an example, the authors proposed estimating filter importance based on the rank of its feature maps. Pruning strategies that leverage information from feature maps (thus involving data forwarding through the network) are named *data-driven* techniques. Since we measure similarity from feature maps, our method belongs to this category of pruning.

Shen et al. [2022] measure filter importance based on the Taylor expansion of the loss change. Importantly, they highlighted the relevance of focusing on latency instead of standard metrics such as FLOPs. To tackle this challenge, the authors transformed the objective of maximizing accuracy within a given latency budget into a resource allocation optimization problem, then solved it using the Knapsack paradigm. In an alternative line of research, studies have demonstrated that standard performance metrics may correlate weakly with inference time [Dehghani et al., 2022; Vasu et al., 2023]. Aligned with these efforts, we demonstrate that our method achieves notable latency improvements and other computational benefits. Differently from Shen et al. [2022], we address the accuracy/latency trade-off without solving any optimization problem. This is possible because layer pruning reduces network depth, directly translating into latency improvement, and our criterion accurately identifies unimportant layers that preserve accuracy, enabling us to achieve higher FLOP reductions while maintaining accuracy simultaneously.

According to existing works [Xu and McAuley, 2023; He and Xiao, 2024], most efforts have been devoted to filter pruning techniques. In contrast to this family of methods that may exhibit bias toward specific metrics like FLOPs or parameters [Dehghani et al., 2022; Vasu et al., 2023], layer pruning achieves performance gains across all computational metrics [Chen and Zhao, 2019; Jordão et al., 2020; Zhou et al., 2022]. In this direction, Chen and Zhao [2019] proposed learning classifiers using features from prunable layers to assign their importance. Following this modeling, layer importance relies on the performance of classifiers. Similar to

ours, the criterion by Chen and Zhao [2019] is layer-specific; however, our criterion focuses on similarity representations through CKA, which we reveal to be more effective. More recently, the work by Zhang and Liu [2022] disconnects residual mapping and estimates its effect using Taylor expansion. Zhou et al. [2022] proposed an evolutionary-based approach, using the weights distribution as one of the inputs for creating the initial population of candidate pruned networks. It is worth mentioning that the methods by Zhou et al. [2022] and Chen and Zhao [2019] require knowledge distillation to recover accuracy from the pruned models, while our method relies on straightforward fine-tuning rounds. Such observations suggest that our CKA criterion is more precise than the previously mentioned strategies for selecting layers.

Apart from pruning, efforts have also been devoted to understanding the role of layers in the expressive power and training dynamics of the models [Zhang et al., 2022; Masarczyk et al., 2023; Chen et al., 2023]. For example, Masarczyk et al. [2023] suggested that the layers of deep networks split into two distinct groups. The initial layers have linearly separable representations, and the subsequent layers, or the tunnel, have less impact on the performance, compressing the already learned representations. This behavior, named *Tunnel Effect*, emerges at the early stages of the training process and corroborates with the notion of redundancy in overparameterized models. Additionally, their work argued that the tunnel is responsible for the performance degradation in out-of-distribution (OOD) samples. We show that our layer pruning method preserves OOD generalization, indicating that its degradation is not restricted to tunnel layers. In summary, we believe our work contributes to these efforts by demonstrating that unimportant layers can be effectively identified and removed without compromising the expressive power of the model and its training dynamics.

## 3 Preliminaries and Proposed Method

**Problem Statement.** According to previous works [Veit et al., 2016; Chen et al., 2016; Dong et al., 2021], residual-based architectures enable the information flow (i.e., the representation) to take different paths through the network. Thereby, layers may not always strongly depend on each other, reinforcing the idea of redundancy in this type of structure, which suggests the possibility of removing layers without compromising the network representation. Upon this evidence, our problem becomes identifying and removing unimportant layers, preserving the representation capacity of the model, and avoiding network collapse. Formally, given a network  $\mathcal{N}$  composed of a layer set  $L$ , our goal is to remove certain layers to produce a shallower network  $\mathcal{N}'$  composed by  $L'$ , where  $|L'| \ll |L|$  and the accuracy of  $\mathcal{N}'$  is as close as possible (ideally better) than its unpruned version  $\mathcal{N}$ .

Naively, one could estimate optimal layers to prune by iterating over all possible candidates, removing one at a time, fine-tuning the model, and selecting the candidate that exhibits the lowest performance degradation. However, this approach becomes computationally expensive as the network depth increases, hence it is unfeasible for most modern architectures and large scale datasets.

**Algorithm 1** Layer Pruning using our CKA criterion

**Input:** Trained Neural Network  $\mathcal{N}$  and Candidate Layers  $l_i \in L$   
**Output:** Pruned Version of  $\mathcal{N}$

- 1:  $R \leftarrow M(\mathcal{N})$
- 2: **for**  $i \leftarrow 1$  **to**  $|L|$  **do**
- 3:    $\mathcal{N}_{l_i} \leftarrow \mathcal{N} \setminus l_i \triangleright$  Removes layer  $l_i$  from  $\mathcal{N}$
- 4:    $R_i \leftarrow M(\mathcal{N}_{l_i}) \triangleright$  Representation extraction of  $\mathcal{N}_{l_i}$
- 5:    $S \leftarrow S \cup s(R, R_{l_i})$
- 6: **end for**
- 7:  $j \leftarrow \text{argmin}(S) \triangleright$  Index of lowest score in  $S$
- 8:  $\mathcal{N} \leftarrow \mathcal{N}_{l_j} \triangleright \mathcal{N}$  becomes its pruned version
- 9: Update  $\mathcal{N}$  via standard fine-tuning

**Definitions.** Consider  $X$  and  $Y$  a set of training samples (e.g., images) and their respective class labels. Let  $\mathcal{N}$  be a dense (unpruned) network trained using  $X$  and  $Y$  (i.e., the traditional supervised paradigm). Consider  $M(\cdot)$  as a function that extracts the representation of a network from the samples  $X$ . Following Xu et al. Evci et al. [2022],  $M$  extracts the feature maps from the layer immediately preceding the classification layer of the network. It is worth mentioning that  $M(\cdot)$  does not take into account the labels  $Y$ . Let  $l_i \in L$  be the candidate layers (i.e., layers the pruning can eliminate) and, finally, define  $\mathcal{N}_{l_i}$  as the network yielded by removing the layer  $l_i$  from  $\mathcal{N}$ .

**Proposed Criterion.** For each  $l_i \in L$ , we obtain  $\mathcal{N}_{l_i}$  w.r.t the previous definition, and apply  $M(\mathcal{N}_{l_i})$  to extract its representation, denoted by  $R_{l_i}$ . Define  $s(\cdot, \cdot)$  as our CKA criterion which takes  $R$  and  $R_{l_i}$ , where  $R \leftarrow M(\mathcal{N})$  (i.e., the original representation), and outputs the score (importance) of  $l_i$ . Following Kornblith et al. [Kornblith et al., 2019], we compute CKA in terms of

$$CKA(R, R_{l_i}) = \frac{HSIC(R, R_{l_i})}{\sqrt{HSIC(R, R)HSIC(R_{l_i}, R_{l_i})}}, \quad (1)$$

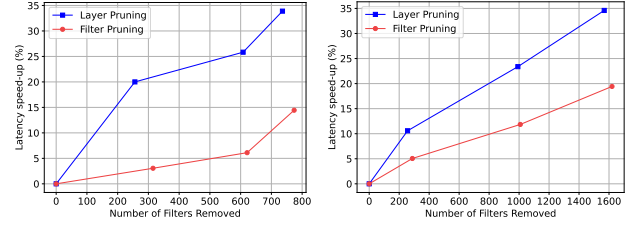
where HSIC is the Hilbert-Schmidt Independence Criterion [Gretton et al., 2005]. We refer interested readers to the works by Kornblith et al. [2019] and Nguyen et al. [2021, 2022] for additional information.

It follows from Equation 1 that  $CKA(R, R_{l_i}) \in [0, 1]$ , where a value of 1 indicates identical feature maps (i.e., the highest similarity preservation). However, an intuitive practice is to remove the lowest-scoring candidate layer. Therefore, we adjust the score in terms of  $s(R, R_{l_i}) = 1 - CKA(R, R_{l_i})$ , ensuring that lower scores are assigned to layers yielding more similar representations.

Algorithm 1 summarizes the process above. From it, we highlight the following points. First, after estimating the importance of all candidate layers, we indeed remove the lowest scoring one. Second, representation extractions employ the same set  $X$ . Finally, the construction of  $\mathcal{N}_{l_i}$  does not involve any fine-tuning. Note that, for a single iteration, our method scales linearly w.r.t the number of layers, implying an  $O(|L|)$  complexity.

## 4 Experiments

**Experimental Setup.** We conduct experiments on CIFAR-10, CIFAR-100 and ImageNet using different versions of the ResNet architecture [He et al., 2016]. Such settings are a



**Figure 2.** Relationship between the number of filters removed (x-axis) and latency speed-up (y-axis) for models obtained from filter and layer pruning. Importantly, such a comparison is possible because when pruning removes layers, it eliminates all filters from that layer. Left and right plots stand for ResNet56 and ResNet110, respectively. Overall, layer pruning notably promotes higher speed-up than filter pruning.

common choice for general compression/acceleration studies [Chen and Zhao, 2019; Jordão et al., 2020; Zhang et al., 2022; Zhou et al., 2022; He and Xiao, 2024].

In order to compare the predictive ability of the unpruned models with their pruned counterparts, we follow common practices [He and Xiao, 2024] and report the difference between accuracies. In this metric, negative and positive values indicate a decrease and an improvement in accuracy (in percentage points – pp), respectively.

**The Effect of Layer Pruning on Efficiency.** Our point of start is illustrating the advantages of layer over filter pruning, as the latter is the most popular family of methods that yield gains without requiring specific hardware [Shen et al., 2022; He and Xiao, 2024].

According to recent studies [Dehghani et al., 2022; Vasu et al., 2023], standard metrics such as FLOPs and parameters, when singly employed, may overlook model efficiency. Therefore, we begin our discussion by considering latency – the time for forwarding a sample (or a set) through the network. To do so, we follow the same process as Jordão et al. [2023], which creates two pruned networks: one obtained through layer removal and the other from filters, aiming for both models to have a similar number of neurons (filters).

**Table 1.** Comparison with state-of-the-art layer-pruning methods. We highlight the best results in bold.

	Method	$\Delta$ Acc.	FLOPs (%)
ResNet56 on CIFAR10	PLS	(-) 0.98	30.00
	FRPP	(+) 0.26	34.80
	ESNB	(-) 0.62	52.60
	LPSR	(+) 0.19	52.75
	CKA (ours)	<b>(+) 0.95</b>	<b>56.29</b>
ResNet110 on CIFAR10	PLS	(-) 0.91	62.69
	LPSR	(-) 0.87	71.65
	CKA (ours)	<b>(+) 0.16</b>	71.30
	CKA (ours)	(+) 0.08	<b>75.05</b>
ResNet50 on ImageNet	ESNB	(+) 1.15	29.89
	PLS	(+) 0.06	37.73
	CKA (Ours)	<b>(+) 1.16</b>	50.33
	CKA (Ours)	(+) 0.80	<b>67.10</b>
ResNet50 on ImageNet	LPSR	(-) 0.57	37.38
	CKA (Ours)	<b>(+) 0.23</b>	<b>39.62</b>
	PLS	(-) 0.67	45.28
	CKA (Ours)	<b>(-) 0.65</b>	45.28

This procedure makes possible a fair comparison in terms of latency performance.

Iteratively repeating this process yields models with varying numbers of filters removed, from which we measure their average latency across 30 runs by forwarding 10K samples and report the speed-up obtained from the pruning process with respect to the original (unpruned) model.

Figure 2 shows the results. It follows that layer pruning yields a higher speed-up than filter removal. For example, in ResNet110, with both methods eliminating around a thousand filters, layer pruning achieves an 11 pp speedup over filter pruning. This advantage persists even when removing approximately 1,600 filters, underscoring the effectiveness of removing layers for network acceleration. Such gains have motivated previous efforts on layer removal [Jordão *et al.*, 2020; Zhang and Liu, 2022; Zhou *et al.*, 2022].

**Effectiveness of the Proposed CKA Criterion.** Our point of start is evaluating the effectiveness of the proposed CKA criterion in assigning layer importance. For this purpose, we take into account representative layer pruning techniques [Jordão *et al.*, 2020; Zhang *et al.*, 2022; Zhou *et al.*, 2022; Chen and Zhao, 2019]. It is worth mentioning that we exclude works on dynamic inference since they belong to a different category of compression and acceleration techniques [Han *et al.*, 2022].

Table 1 summarizes the results. According to this table, our method outperforms existing techniques by a large margin. On ResNet56, compared to the best strategy in terms of delta in accuracy, LPSR [Zhang and Liu, 2022], our method outperforms it by up to 0.76 pp while exhibiting better gains. Regarding FLOP reduction, the best method underperforms ours by 3.4 pp. This behavior is prevalent in ResNet110 and ResNet50 (on ImageNet). Notably, we reduce around  $2\times$  more FLOPs than other criteria while obtaining an improvement in accuracy.

The reason for these remarkable results is that our method carefully selects which layers to eliminate. For example, Jordão *et al.* [2020] and Zhang and Liu [2022] compute scores for layers by aggregating the sum of scores from the individual filters that compose a layer. Table 1 suggests that this aggregating scheme may be inappropriate. This finding concurs with the observations made by Masarczyk *et al.* [2023], where the authors argued that aggregating all features of a layer to compose its final representation is suboptimal, particularly for transfer learning.

In terms of computational cost, compared to Zhou *et al.* [2022], our method is more cost-friendly. It turns out that this approach solves the score assignment problem through an evolutionary algorithm. Therefore, their method scales expensively as the depth (i.e.,  $|L|$ ) increases. On the other hand, to prune a model with  $|L|$  layers our approach requires  $|L|$  forwards and CKA comparisons, scaling linearly (see Algorithm 1). The method by Jordão *et al.* [2020] is also linear w.r.t the number of layers, however, it is unable to prune a layer from any region of the network. Specifically, to eliminate a layer  $i$ , their method requires the removal of all subsequent layers  $j$  where  $i < j < |L|$ .

The previous evidence corroborates the suitability of our criterion for selecting unimportant layers compared to existing state-of-the-art layer pruning methods. Importantly, the discussion above confirms our hypothesis that similar

**Table 2.** Comparison with state-of-the-art pruning methods on CIFAR-10. For each level of FLOP reduction (%), we highlight the best results in bold.

	Method	$\Delta$ Acc.	FLOPs (%)
ResNet56	WhiteBox (TNNLS, 2023)	+0.28	55.60
	CLR-RNF (TNNLS, 2023)	+ 0.01	57.30
	<b>CKA (ours)</b>	<b>+0.78</b>	<b>60.04</b>
	DAIS (TNNLS, 2023)	-1.00	70.90
	GCNP (IJCAI, 2022)	-0.97	77.22
	<b>CKA (ours)</b>	<b>+0.08</b>	75.05
	<b>CKA (ours)</b>	-0.66	<b>78.80</b>
	DECORE (CVPR, 2022)	-2.41	81.50
ResNet110	<b>CKA (ours) + <math>\ell_1</math></b>	<b>-0.94</b>	<b>84.70</b>
	EPruner (TNNLS, 2022)	+0.12	65.91
	CRL-RNF (TNNLS, 2023)	+0.14	66.00
	WhiteBox (TNNLS, 2023)	+0.62	66.00
	<b>CKA (ours)</b>	<b>+0.80</b>	67.10
	<b>CKA (ours)</b>	+0.59	<b>70.83</b>
	DECORE (CVPR, 2022)	-0.79	76.92
	<b>CKA (ours)</b>	<b>+0.23</b>	76.42
	<b>CKA (ours)</b>	-0.41	<b>87.61</b>

representations between a dense (unpruned) network and its optimal pruning candidate indicate lower relative importance.

**Comparison with the State of the Art.** The previous experiments shed light on the benefits of layer pruning and the effectiveness of our criterion for selecting layers to remove. We now compare our method with general state-of-the-art pruning techniques. For this purpose, we evaluate our method against the most recent and top-performing techniques mainly based on the survey by He and Xiao [2024]. Specifically, we consider methods capable of achieving notable FLOP reduction with negligible accuracy drop.

Table 2 shows the results on CIFAR-10 for ResNet56/110. On these architectures, our method outperforms state-of-the-art techniques by removing more FLOPs and achieving the best delta in accuracy. For example, in Table 2 (left), within comparable FLOP reduction regimes, we outperform state-of-the-art methods by a margin starting at approximately 0.4 pp and reaching up to more than 2.5 pp.

Table 2 poses an interesting behavior: at high FLOP reduction levels (i.e., above 70%), all methods fail to preserve accuracy. In contrast, our method removes more than 75% of FLOPs with no accuracy drop. Most cases, our method promotes predictive ability improvements. This benefit is expected, as layer pruning (and its variations) acts as a form of regularization Chen *et al.* [2016]; Han *et al.* [2022]. Table 2 highlights this behavior in other pruning techniques, but unlike ours, exhibited only in low compression regimes.

As we mentioned before, our method is orthogonal to other pruning categories (i.e., the ones in Table 2); therefore, we can combine it with these techniques. Built upon previous ideas [Jordão *et al.*, 2020], we take one of our pruned models and further prune it using the popular  $\ell_1$ -norm filter pruning. In this scenario, we achieve even better results, surpassing our best performance gains (using layer pruning only) in terms of FLOP reduction by 5.9 pp. Specifically, our method achieves



**Table 3.** Comparison with state-of-the-art pruning methods on ImageNet using ResNet50 and CIFAR-100 using ResNet56.

	Method	$\Delta$ Acc.	FLOPs (%)
ResNet50 on ImageNet	GKP-TMI (ICLR, 2022)	(-) 0.62	33.74
	LSPR (SPL, 2022)	(-) 0.57	37.38
	<b>CKA (Ours)</b>	<b>(-) 0.18</b>	<b>39.62</b>
	CLR-RNF (TNNLS, 2023)	(-) 1.16	40.39
	DECORE (CVPR, 2022)	(-) 1.57	42.30
	SOSP (ICLR, 2022)	(-) 0.94	45.00
	WhiteBox (TNNLS, 2023)	(-) 0.83	<b>45.60</b>
	<b>CKA (Ours)</b>	<b>(-) 0.90</b>	45.28
	DECORE (CVPR, 2022)	(-) <b>4.09</b>	60.88
	<b>CKA (Ours) + <math>\ell_1</math>-norm</b>	(-) 5.15	<b>62.00</b>
ResNet110 on CIFAR-100	DLRFC (ECCV, 2022)	(+) 0.27	25.50
	FRPP (TPAMI, 2019)	(-) 0.23	38.30
	GCNP (IJCAI, 2022)	0.00	48.77
	GCNP (IJCAI, 2022)	(-) 0.64	52.22
	LSPR (SPL, 2022)	(-) 1.22	52.68
	DAIS (TNNLS, 2023)	(+) <b>0.81</b>	53.60
	<b>CKA (ours)</b>	(+) 0.71	<b>63.79</b>
	<b>CKA (ours)</b>	(-) <b>0.59</b>	71.29
	<b>CKA (ours)</b>	(-) 1.96	<b>75.05</b>

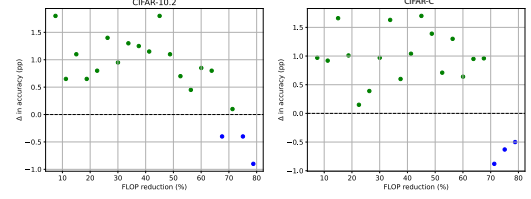
a FLOP reduction above 80% while maintaining the accuracy drop below one pp. The single method paired with this level of reduction, DECORE [Alwani *et al.*, 2022], exhibits an accuracy degradation of 2.41 pp compared to the original model.

We also evaluate our method on ImageNet and CIFAR-100 in Table 3. On these datasets, we observe a similar trend with the CIFAR-10 discussion when comparing our method against state-of-the-art pruning techniques. Particularly, on ImageNet, the layer-pruning approach LSPR [Zhang and Liu, 2022] notably hurts the accuracy, whereas our method is capable of improving it while removing more FLOPs. We also combine our method with the  $\ell_1$ -norm criterion to further prune our model in terms of filters, achieving a higher computational reduction of 62.00%.

**Robustness to Adversarial Samples.** Evaluating pruned models in adversarial scenarios plays a critical role, as we need to guarantee the trustworthiness of these models before deploying them in real-world applications such as autonomous driving. To assess the adversarial robustness of the pruned models, we employ CIFAR-C [Hendrycks and Dietterich, 2019] and CIFAR-10.2 [Lu *et al.*, 2020]. On these datasets, our pruned models obtained superior robustness compared to the unpruned model as seen in Figure 3. We notice a similar trend when evaluating the pruned models against the FGSM attack (see Section A.1 in our publicly available paper).

The previous discussion indicates that our method operated as a defense mechanism against adversarial attacks. Such a finding is unsurprising, as previous works have demonstrated the potential of pruning as a defense mechanism against adversarial attacks.

**GreenAI and Transfer Learning.** The current consensus is that deeper models yield better predictive ability. A consequence of this paradigm is the computational overhead seen in modern architectures, contributing to an increase in energy demands, both in the training and deployment phases. According to previous works [Lacoste *et al.*, 2019; Strubell *et al.*, 2019; Faiz *et al.*, 2024], these demands result in high carbon emissions ( $\text{CO}_2$ ). Fortunately, the benefits in FLOP reduction and latency promoted by our method directly translate into a

**Figure 3.** Results of pruned models for CIFAR-10.2 and CIFAR-C. Green/blue points correspond to an accuracy improvement/degradation, respectively.

reduction of  $\text{CO}_2$ . For example, our best-pruned version of ResNet56 implies a reduction of  $\text{CO}_2$  by 67.88% during the fine-tuning. On ResNet110, our pruned model at the highest FLOP reduction regime leads to 80.85% of  $\text{CO}_2$  reduction. We can further evidence this practical reduction in transfer learning scenarios, where fine-tuning the models is necessary for downstream tasks. To do so, we employ the pruned versions of ResNet56 on CIFAR-100 and transfer their knowledge to CIFAR-10. Interestingly, we observe that our pruned model with the highest FLOP reduction achieved a  $\text{CO}_2$  reduction of 68.23% while maintaining accuracy within 1 pp compared to the unpruned model. In addition, pruned models with lower FLOP reductions achieve better transfer learning results, corroborating the findings by Xu and McAuley [2023]. Such behavior suggests a challenge for the current evaluation of pruning techniques: the quality of pruning should consider its performance in transfer learning tasks.

We believe the results above pose an important step towards Green AI. Particularly on the learning paradigm involving foundation models, as the success of this emerging field relies on transfer-learning (and self-supervised), hence, requiring fine-tuning [Bommasani *et al.*, 2021; Evci *et al.*, 2022; Amatriain, 2023].

## 5 Conclusions

Layer pruning emerges as an exciting compression and acceleration technique due to more pronounced benefits in FLOP reduction and latency speed-up than other forms of pruning. Despite achieving promising results, existing criteria for selecting layers fail to fully characterize the underlying properties of these structures. To mitigate this, we proposed a novel criterion for identifying unimportant layers. Our method leverages the Centered Kernel Alignment (CKA) similarity metric to select such layers from a set of candidates. Extensive experiments on standard benchmarks and architectures confirm the effectiveness of our method. Specifically, we outperforms existing layer-pruning techniques in terms of both accuracy and FLOP reduction by a large margin. Compared to other state-of-the-art pruning methods, we obtain the best compromise between accuracy and FLOP reduction. Particularly, at high reduction levels all methods fail to preserve accuracy, whereas our method exhibits either an improvement or negligible drop. In addition, our pruned models exhibit robustness to adversarial samples and positive out-of-distribution generation. Finally, our work poses an important step towards Green AI, reducing up to 80.85% of carbon emissions for training and fine-tuning modern architectures.

## Declarations

## Acknowledgements

This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number #2023/11163-0 and #2024/17684-4. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. The authors would like to thank grant #402734/2023-8, National Council for Scientific and Technological Development (CNPq). Artur Jordao Lima Correia would like to thank Edital Programa de Apoio a Novos Docentes 2023. Processo USP nº: 22.1.09345.01.2.

## Authors' Contributions

Ian Pons: Formal analysis, Methodology, Investigation, Writing – review & editing;

Bruno Yamamoto: Formal analysis, Investigation Writing – review & editing;

Artur Jordao: Writing – review & editing, Supervision.

All authors read and approved the final manuscript.

## Competing interests

The authors declare that they have no competing interests

## Availability of data and materials

Source code available at: [github.com/IanPons/CKA-Layer-Pruning](https://github.com/IanPons/CKA-Layer-Pruning)

## References

- Alwani, M., Wang, Y., and Madhavan, V. (2022). DECORE: deep compression with reinforcement learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 12339–12349. IEEE. DOI: 10.1109/CVPR52688.2022.01203.
- Amatriain, X. (2023). Transformer models: an introduction and catalog. *CoRR*, abs/2302.07730. DOI: 10.48550/ARXIV.2302.07730.
- Bair, A., Yin, H., Shen, M., Molchanov, P., and Álvarez, J. M. (2024). Adaptive sharpness-aware pruning for robust sparse networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Bommasani, R., Hudson, D. A., Adeli, E., and et al. (2021). On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.
- Chen, D., Zhang, W., Xu, X., and Xing, X. (2016). Deep networks with stochastic depth for acoustic modelling. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2016, Jeju, South Korea, December 13-16, 2016*, pages 1–4. IEEE. DOI: 10.1109/APSIPA.2016.7820692.
- Chen, S. and Zhao, Q. (2019). Shallowing deep networks: Layer-wise pruning based on feature representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):3048–3056. DOI: 10.1109/TPAMI.2018.2874634.
- Chen, Y., Yuille, A. L., and Zhou, Z. (2023). Which layer is learning faster? A systematic exploration of layer-wise convergence rate for deep neural networks. In *ICLR*.
- Dehghani, M., Tay, Y., Arnab, A., Beyer, L., and Vaswani, A. (2022). The efficiency misnomer. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Dong, Y., Cordonnier, J., and Loukas, A. (2021). Attention is not all you need: pure attention loses rank doubly exponentially with depth. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 2793–2803. PMLR.
- Duong, L. R., Zhou, J., Nassar, J., Berman, J., Olieslagers, J., and Williams, A. H. (2023). Representational dissimilarity metric spaces for stochastic neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Evci, U., Dumoulin, V., Larochelle, H., and Mozer, M. C. (2022). Head2toe: Utilizing intermediate representations for better transfer learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S., editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 6009–6033. PMLR.
- Faiz, A., Kaneda, S., Wang, R., Osi, R. C., Sharma, P., Chen, F., and Jiang, L. (2024). LLMCarbon: Modeling the end-to-end carbon footprint of large language models. In *The Twelfth International Conference on Learning Representations*.
- Gretton, A., Bousquet, O., Smola, A. J., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In Jain, S., Simon, H. U., and Tomita, E., editors, *Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 8-11, 2005, Proceedings*, volume 3734 of *Lecture Notes in Computer Science*, pages 63–77. Springer. DOI: 10.1007/11564089\_7.
- Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., and Tao, D. (2023). A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):87–110. DOI: 10.1109/TPAMI.2022.3152247.
- Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. (2022). Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7436–7456. DOI: 10.1109/TPAMI.2021.3117837.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society. DOI: 10.1109/CVPR.2016.90.
- He, Y., Liu, P., Wang, Z., Hu, Z., and Yang, Y. (2019). Filter pruning via geometric median for deep convolutional neural networks acceleration. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4340–4349. Computer Vision Foundation / IEEE. DOI: 10.1109/CVPR.2019.00447.
- He, Y. and Xiao, L. (2024). Structured pruning for deep convolutional neural networks: A survey. *IEEE*

- Trans. Pattern Anal. Mach. Intell.*, 46(5):2900–2919. DOI: 10.1109/TPAMI.2023.3334614.
- Hendrycks, D. and Dietterich, T. G. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Huang, Z., Shao, W., Wang, X., Lin, L., and Luo, P. (2021). Rethinking the pruning criteria for convolutional neural network. In *NeurIPS*.
- Jordão, A., de Araújo, G. C., de Almeida Maia, H., and Pedrini, H. (2023). When layers play the lottery, all tickets win at initialization. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023 - Workshops, Paris, France, October 2-6, 2023*, pages 1196–1205. IEEE. DOI: 10.1109/ICCVW60793.2023.00130.
- Jordão, A., Lie, M., and Schwartz, W. R. (2020). Discriminative layer pruning for convolutional neural networks. *IEEE J. Sel. Top. Signal Process.*, 14(4):828–837. DOI: 10.1109/JSTSP.2020.2975987.
- Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. E. (2019). Similarity of neural network representations revisited. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.
- Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. (2019). Quantifying the carbon emissions of machine learning. *CoRR*, abs/1910.09700.
- Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., and Shao, L. (2020). Hrank: Filter pruning using high-rank feature map. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 1526–1535. Computer Vision Foundation / IEEE. DOI: 10.1109/CVPR42600.2020.00160.
- Lu, S., Nott, B., Olson, A., Todeschini, A., Vahabi, H., Carmon, Y., and Schmidt, L. (2020). Harder or different? a closer look at distribution shift in dataset reproduction. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, volume 5, page 15.
- Masarczyk, W., Ostaszewski, M., Imani, E., Pascanu, R., Miłoś, P., and Trzcinski, T. (2023). The tunnel effect: Building data representations in deep neural networks. In *NeurIPS*.
- Nguyen, T., Raghu, M., and Kornblith, S. (2021). Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Nguyen, T., Raghu, M., and Kornblith, S. (2022). On the origins of the block structure phenomenon in neural network representations. *Trans. Mach. Learn. Res.*, 2022.
- Shen, M., Yin, H., Molchanov, P., Mao, L., Liu, J., and Álvarez, J. M. (2022). Structural pruning via latency-saliency knapsack. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Silver, D., Huang, A., Maddison, C. J., and et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489. DOI: 10.1038/NATURE16961.
- Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Korhonen, A., Traum, D. R., and Màrquez, L., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3645–3650. Association for Computational Linguistics. DOI: 10.18653/V1/P19-1355.
- Vasu, P. K. A., Gabriel, J., Zhu, J., Tuzel, O., and Ranjan, A. (2023). Mobileone: An improved one millisecond mobile backbone. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 7907–7917. IEEE. DOI: 10.1109/CVPR52729.2023.00764.
- Veit, A., Wilber, M. J., and Belongie, S. J. (2016). Residual networks behave like ensembles of relatively shallow networks. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 550–558.
- Wang, H., Qin, C., Bai, Y., Zhang, Y., and Fu, Y. (2022). Recent advances on neural network pruning at initialization. In Raedt, L. D., editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5638–5645. ijcai.org. DOI: 10.24963/IJCAI.2022/786.
- Xu, C. and McAuley, J. J. (2023). A survey on model compression and acceleration for pretrained language models. In Williams, B., Chen, Y., and Neville, J., editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 10566–10575. AAAI Press. DOI: 10.1609/AAAI.V37I9.26255.
- Zhang, C., Bengio, S., and Singer, Y. (2022). Are all layers created equal? *Journal of Machine Learning Research*.
- Zhang, K. and Liu, G. (2022). Layer pruning for obtaining shallower resnets. *IEEE Signal Process. Lett.*, 29:1172–1176. DOI: 10.1109/LSP.2022.3171128.
- Zhao, Q. and Wressnegger, C. (2023). Holistic adversarially robust pruning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Zhou, Y., Yen, G. G., and Yi, Z. (2022). Evolutionary shallowing deep neural networks at block levels. *IEEE Trans. Neural Networks Learn. Syst.*, 33(9):4635–4647. DOI: 10.1109/TNNLS.2021.3059529.