SequentialAttention++ for Block Sparsification: Differentiable Pruning Meets Combinatorial Optimization

Taisuke Yasuda[*†]
Voleon Group
yasuda.taisuke1@gmail.com

Kyriakos Axiotis[†]
Google Research
axiotis@google.com

Gang Fu[†]
Google Research
thomasfu@google.com

MohammadHossein Bateni
Google Research
bateni@google.com

Vahab Mirrokni
Google Research
mirrokni@google.com

Abstract

Neural network pruning is a key technique towards engineering large yet scalable, interpretable, and generalizable models. Prior work on the subject has developed largely along two orthogonal directions: (1) differentiable pruning for efficiently and accurately scoring the importance of parameters, and (2) combinatorial optimization for efficiently searching over the space of sparse models. We unite the two approaches, both theoretically and empirically, to produce a coherent framework for structured neural network pruning in which differentiable pruning guides combinatorial optimization algorithms to select the most important sparse set of parameters. Theoretically, we show how many existing differentiable pruning techniques can be understood as nonconvex regularization for group sparse optimization, and prove that for a wide class of nonconvex regularizers, the global optimum is unique, group-sparse, and provably yields an approximate solution to a sparse convex optimization problem. The resulting algorithm that we propose, *SequentialAttention++*, advances the state of the art in large-scale neural network block-wise pruning tasks on the ImageNet and Criteo datasets.

1 Introduction

Pruning methods for neural networks [LeCun et al., 1989] replace dense weight matrices by sparse approximations, which offer improved generalization and inference efficiency in terms of storage, energy consumption, and other computational resources. In various common formulations, the problem of computing the best sparse approximation to a dense weight matrix is intractable as it generalizes the sparse linear regression problem, which is known to be NP-hard even to approximate [Natarajan, 1995, Foster et al., 2015, Gupte and Vaikuntanathan, 2021, Price et al., 2022]. Despite this fact, a wide variety of techniques have proven to be quite successful in practice. This includes magnitude pruning, ℓ_1 regularization, greedy coordinate descent, sampling, among others.

While earlier works have focused on unstructured (i.e., entrywise) sparsity, which has been an active and fruitful area, researchers have rapidly recognized the importance of *structured* sparsity, which enforces that the sparse approximation respects certain patterns, such as block structure. These structural constraints often lead to further efficiency gains due to improved hardware utilization

[*] Work done while at Google Research.
[†] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2354

[Anwar et al., 2017, Pool and Yu, 2021, Liu et al., 2022]. Our work thus focuses on developing new and improved techniques for structured sparsification of weight matrices, and in particular on block sparsification [Ma et al., 2023], which allow for a balance between performance gains from hardware utilization and reduced computation due to sparsity [Gale et al., 2023].

1.1 Importance scoring and combinatorial optimization

We argue that existing approaches to neural network pruning have developed along two orthogonal directions: algorithms for *importance scoring* and algorithms for *combinatorial optimization*. We roughly think of importance scoring algorithms as those that aim to select a small number of important entries (or blocks) of weight matrices, while we think of combinatorial optimization algorithms as wrapper methods that use the importance scoring algorithms as oracles to iteratively construct the desired (block) sparse weight matrix.

Among importance scoring algorithms, early popular choices have included magnitude pruning [Thimm and Fiesler, 1995, Han et al., 2015], where the magnitude of each trainable parameter serves as a proxy for its importance, as well as methods based on gradients [Karnin, 1990, Sanh et al., 2020], Hessians [LeCun et al., 1989, Hassibi et al., 1993, Singh and Alistarh, 2020, Frantar and Alistarh, 2023], and other statistics of the weights. Other works have incorporated ℓ_1 regularization [Wen et al., 2016, Yang et al., 2019] to encourage sparsity. More recently, a class of techniques broadly termed *differentiable pruning* inspired by techniques for differentiable neural architecture search [Liu et al., 2019] have increased in popularity, where importance scores and/or soft masks are trained together with the network weights in a differentiable manner [Xiao et al., 2019, Voita et al., 2019, Kang and Han, 2020, Ramakrishnan et al., 2020, Savarese et al., 2020, Zhang et al., 2022]. Variations of this idea use the network weights themselves to represent the "importance scores", and simply use a transformation of the original network weights [Schwarz et al., 2021, Vanderschueren and Vleeschouwer, 2023, Cho et al., 2023].

As for the combinatorial optimization aspects of pruning, the use of iterative or greedy procedures has long been explored and is known to improve sparsification quality over "one-shot" uses of importance scoring algorithms [LeCun et al., 1989, Hassibi et al., 1993, Ström, 1997, Frankle and Carbin, 2019]. The work of Halabi et al. [2022] gives a theoretical justification of this observation via connections to weakly submodular optimization. Combinatorial optimization algorithms beyond greedy approaches, especially local search methods that improve sparsity patterns via local swaps such as iterative hard thresholding (IHT), have long been known in the submodular optimization literature, and have recently been shown to be extremely effective when combined with magnitude pruning [Evci et al., 2020, Peste et al., 2021, Kuznedelev et al., 2023b, Benbaki et al., 2023]. The work of Peste et al. [2021] also provides strong theoretical guarantees for their approach, *ACDC*. Similar ideas have also been termed as "neuroregeneration" in work of Liu et al. [2021].

Given these two highly fruitful approaches to the problem of pruning neural networks, it is natural to ask how recent advances in importance scoring algorithms and combinatorial optimization algorithms can work in concert. We investigate this question from both theoretical and empirical perspectives.

1.2 Theoretical results

We first present a theoretical investigation of differentiable pruning techniques for block sparsification when the objective function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ is strictly convex and differentiable. This already captures several interesting problems where block sparsification of weight matrices is desired, such as multinomial logistic regression and multiple response linear regression. We take the n variables of our objective function to be partitioned into disjoint groups $\{T_i\}_{i=1}^t$ where $T_i \subseteq [n]$ and possibly have varying size. For instance, in the context of block sparsification, $\mathcal{L}$ could correspond to the multinomial logistic regression objective function with K classes and d features, and the $n = Kd$ variables could be partitioned into t blocks $T_1, T_2, \ldots, T_t$. Furthermore, we will also consider an ℓ_2 regularization term on the parameters $\boldsymbol{\beta}$, that is, we study variants of the problem $\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2$. Note that explicit ℓ_2 regularization is a standard component of machine learning architectures, and also appears *implicitly* whenever a loss function is optimized with gradient descent [Shalev-Shwartz, 2012], with the regularization parameter λ being controlled by learning rate parameters and early stopping [Suggala et al., 2018].

Our contributions are twofold: (1) we show that a wide variety of differentiable pruning techniques can all be understood as an implementation of nonconvex regularization that generalizes the group LASSO, and (2) we show that a wide class of nonconvex regularizers give a unique 1-sparse global minimum that coincides with the unique 1-sparse global minimum of a corresponding group LASSO problem. These two results together establish that many differentiable pruning techniques work simply by identifying the *same* 1*-sparse solution as the group LASSO*. In turn, it is known that the 1-sparse solution found by the group LASSO is the variable block with the largest squared gradient [Axiotis and Yasuda, 2023], which is equivalent to the Orthogonal Matching Pursuit [Pati et al., 1993, Shalev-Shwartz et al., 2010, Liberty and Sviridenko, 2017, Elenberg et al., 2018] when applied sequentially (see Appendix B). Thus together, these results make progress towards understanding the inner workings of modern differentiable pruning methods.

1.2.1 Differentiable pruning as nonconvex regularization

For our first contribution, we observe that if we minimize the loss $\mathcal{L}$ with each of the variable groups $\boldsymbol{\beta}|_{T_i}$ for $i \in [t]$ replaced by a "masked" version $q(\mathbf{w}_i)\boldsymbol{\beta}|_{T_i}$, and with regularization on $\mathbf{w}$ and $\boldsymbol{\beta}$, then this problem is equivalent to another optimization problem that simply optimizes $\mathcal{L}$ with a different, and often sparsity-inducing, regularizer. A basic version of this observation already appears in works of Hoff [2017], Axiotis and Yasuda [2023], where it is shown that if the masks q are just the identity, then we recover the usual group LASSO problem, that is,

$$\min_{\mathbf{w}\in\mathbb{R}^t,\boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\{\mathbf{w}_i\boldsymbol{\beta}|_{T_i}\}_{i=1}^t) + \frac{\lambda}{2}\left(\|\mathbf{w}\|_2^2 + \|\boldsymbol{\beta}\|_2^2\right) = \min_{\boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\boldsymbol{\beta}) + \lambda\sum_{i=1}^t \|\boldsymbol{\beta}|_{T_i}\|_2$$

where $\{\mathbf{w}_i\boldsymbol{\beta}|_{T_i}\}_{i=1}^t$ denotes the concatenation of the "masked" groups $\mathbf{w}_i\boldsymbol{\beta}|_{T_i}$ for $i \in [t]$. We generalize this observation and show that this framework also applies to other ideas popular in the differentiable pruning literature, such as applying ℓ_1 regularization on the masks $\mathbf{w}$ to induce sparsity [Yang et al., 2019] or applying softmax-type masks such as $\exp(\mathbf{w}_i)$ [Yasuda et al., 2023]. We note that prior to our work, there was little theoretical understanding on the value of applying such techniques in the context of differentiable pruning.

We also apply similar ideas to differentiable pruning techniques that use the network weights themselves as importance scores [Schwarz et al., 2021, Cho et al., 2023]. Here, the basic observation is that if one optimizes a loss function $\mathcal{L}$ with variables $\boldsymbol{\beta}$ replaced by the (signed) entrywise square $\boldsymbol{\beta} \odot \boldsymbol{\beta}$, then this results in a "rich get richer" dynamic where large weights evolve to be larger while smaller weights are driven to zero, resulting in sparse solutions. This idea also has connections to exponentiated gradient descent which also results in sparse solutions [Vaskevicius et al., 2019, Amid and Warmuth, 2020a,b]. However, prior work only handles entrywise sparsity and does not address the question of structured pruning. We show that these ideas can also be understood in the framework of sparsity-inducing regularizers, even in the group setting. Here, we show that "masking" each of the variable groups $\boldsymbol{\beta}|_{T_i}$ by its ℓ_2 norm $\|\boldsymbol{\beta}|_{T_i}\|_2$ gives a natural group generalization of this technique, and that this gives an optimization problem that is again equivalent to the group LASSO.

1.2.2 Unique sparse global minima

Our second set of contributions is to analyze the solutions of a wide class of nonconvex regularizers. We now consider the following regularized problem, where $q : \mathbb{R}_+ \to \mathbb{R}_+$ is a strictly increasing and subadditive function with $q(0) = 0$, and $\lambda > 0$ is a regularization parameter:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\boldsymbol{\beta}) + \lambda \cdot q^{-1}\left(\sum_{i=1}^t q(\|\boldsymbol{\beta}|_{T_i}\|_2)\right). \tag{1}$$

For instance, some popular choices of q include the absolute value $q(x) = |x|$, p-th powers $q(x) = |x|^p$ for $p < 1$, or logarithmic regularizers such as $q(x) = \log(1 + x)$. In general, the class of such q (strictly) contains the set of all concave functions q that vanish at the origin. Note that the form of (1) slightly differs from the usual form of nonconvex regularizers, as it applies q^{-1} on the sum $\sum_{i=1}^t q(\|\boldsymbol{\beta}|_{T_i}\|_2)$ rather than taking the regularizer to just be $\sum_{i=1}^t q(\|\boldsymbol{\beta}|_{T_i}\|_2)$. This does not substantially change the nature of the optimization problem as it is the Lagrangian dual for the same constraint. The main result of this section is Theorem 1.1, which relates the group q-regularized

https://doi.org/10.52202/079017-2354

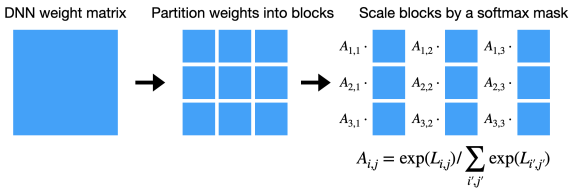


Figure 1: Differentiable pruning of weight blocks

objective (1) to the following corresponding group LASSO objective:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\boldsymbol{\beta}) + \lambda \sum_{i=1}^{t} \|\boldsymbol{\beta}|_{T_i}\|_2. \tag{2}$$

Theorem 1.1 (Unique sparse global minima)**.** *Let $q : \mathbb{R}_+ \to \mathbb{R}_+$ be strictly increasing, subadditive (i.e., $q(a+b) \leq q(a) + q(b)$ for $a, b \in \mathbb{R}^+$), and satisfy $q(0) = 0$. If (2) has a unique minimizer $\boldsymbol{\beta}^*$ with group sparsity at most 1, then $\boldsymbol{\beta}^*$ is also the unique minimizer for (1).*

We make several remarks about Theorem 1.1. First, we justify why the assumption of the theorem is not vacuous: that is, we explain why the group LASSO objective (2) has sparse solutions. In recent work of Axiotis and Yasuda [2023] the following are shown if $\mathcal{L}$ is strongly convex and differentiable:

- If $\lambda \geq \tau$ for $\tau = \max_{i=1}^{t} \|\nabla\mathcal{L}(0)|_{T_i}\|_2$, then (2) has a unique global minimizer at $\boldsymbol{\beta} = 0$.
- If $\lambda < \tau$ is sufficiently close to τ, then (1) has a unique 1-sparse global minimizer.

Thus, when λ is large enough, Theorem 1.1 establishes that (1) has a unique sparse global minimum.

Furthermore, Axiotis and Yasuda [2023] also show that the above global minimizer of the group LASSO problem (2) with group sparsity 1 is supported on a group T_i that maximizes $\|\nabla\mathcal{L}(0)|_{T_i}\|_2$, that is, it selects the group of variables that locally provides the largest improvement in the objective function cost. Repeatedly alternating between selecting such a feature and re-optimizing over the support is an algorithm known as the *group orthogonal matching pursuit (group OMP)*, and has provable guarantees for group sparse convex optimization when $\mathcal{L}$ satisfies the restricted strong convexity and restricted smoothness properties [Axiotis and Yasuda, 2023]. It is also shown that a related local search algorithm known as *group orthogonal matching pursuit with replacement (group OMPR)* also applies in this context, which has improved guarantees.

Finally, we emphasize that it is generally difficult to establish structural results for nonconvex optimization problems, even for simple convex problems with nonconvex regularizers. Thus, we believe that our results may be of independent interest in the literature of nonconvex optimization.

1.3 Empirical results: SequentialAttention++

We now apply our theoretical insights of combining differentiable pruning and combinatorial optimization to develop a novel algorithm for block neural network pruning, which we call *SequentialAttention++*. SequentialAttention++ is primarily a fusion of two prior techniques: *Sequential Attention*, a feature selection technique based on differentiable pruning developed in work of Yasuda et al. [2023], and *ACDC*, which is a highly effective stochastic adaptation of the classic iterative hard thresholding (IHT) algorithm [Peste et al., 2021] from the combinatorial optimization literature.

Sequential Attention [Yasuda et al., 2023] is an algorithm for feature selection on neural networks, that introduces a softmax mask that is trained together with the neural network weights. Each of the n input features is scaled by a differentiable mask $A_i = \exp(L_i)/\sum_{j=1}^{n} \exp(L_j)$ for a vector $L \in \mathbb{R}^n$ of logits. Note that our theoretical results on differentiable pruning, and in particular Lemma 2.1, suggests that this roughly corresponds to performing a log-sum regularization on the corresponding weights for these features. We first extend this to the block sparsification setting by instead scaling each block of weights to prune by a similar softmax mask (see Figure 1). Note that in this new setting, Lemma 2.1 shows that this corresponds to a *group* log-sum regularization on each of the blocks.

We then use this differentiable pruning technique as part of a local search procedure inspired by ACDC [Peste et al., 2021]. In the originally proposed ACDC algorithm, the neural network is trained

in multiple phases, where the phases alternate between a "dense" training phase and a "sparse" training phase. During the dense phases, the weights are trained in the standard way, whereas in the sparse phases, only a sparse set of weights corresponding to the top k weights at the beginning of the phase (i.e., chosen by magnitude pruning) are used. The idea here is that if a suboptimal sparse support is selected during the sparse phase, then this support can be modified during the dense phase. We note that one of the weaknesses of this algorithm is the use of the weight magnitudes as a proxy for the importance of the weights, whereas improved parameter importance estimation is possible by introducing differentiable pruning techniques. Thus in our SequentialAttention++ algorithm, we modify the ACDC algorithm by training a softmax mask together with the neural network weights during the dense phase as in Figure 1, and then using the softmax mask to select a sparse support during the sparse phases. Our theoretical results establish provable guarantees for a slightly modified version of this algorithm, by showing that log-sum regularization can be integrated with a similar local search algorithm that alternates between dropping small weights from the support, selecting weights via regularization, and optimizing on the new support (see Theorem B.3 and Appendix B).

2 Theory

In Section 2, we present our theoretical results on differentiable pruning and local search algorithms for DNN sparsification. Missing proofs can be found in Appendix A.

2.1 Differentiable pruning as nonconvex regularization

In this section, we show how a wide variety of differentiable pruning techniques studied in the literature can be viewed as nonconvex regularizers. As described earlier in Section 1.2.2, we later show that nonconvex regularization can in fact be connected to provable guarantees for sparse convex optimization by implementing the orthogonal matching pursuit algorithm and its variants. Thus, together, we give the first steps towards a full theoretical analysis of many popular differentiable pruning techniques in the literature.

2.1.1 Unnormalized softmax

The softmax is a popular differentiable sparsity-inducing technique, where a vector is transformed by exponentiating each entry and normalizing the result. The softmax forms the backbone of many modern ML techniques ranging from multinomial logistic regression to differentiable architecture search [Liu et al., 2019] to attention mechanisms and transformers [Vaswani et al., 2017], and thus a theoretical understanding of the softmax is critical mission for modern machine learning theory.

We take a step towards this by considering *unnormalized* softmax, which corresponds to a simple entrywise exponentiation. The unnormalized softmax is a popular alternative to the usual softmax as it still captures its sparsity-inducing properties [Amid and Warmuth, 2020a,b], while its simplicity allows for more efficient implementations. We show that, in fact, unnormalized softmax can be viewed as a type of log-sum regularization, which is a popular relaxation of the $\|\cdot\|_0$ norm that has been often considered in the machine learning and signal processing literatures [Rao and Kreutz-Delgado, 1999, Wipf and Nagarajan, 2009, Qiao et al., 2020, Tugnait, 2022, Zhou et al., 2023].

Lemma 2.1 (Unnormalized softmax as log-sum regularization)**.**

$$\min_{\mathbf{w}\in\mathbb{R}^t,\boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\{\exp(\mathbf{w}_i)\boldsymbol{\beta}|_{T_i}\}_{i=1}^t) + \lambda\big(\|\mathbf{w}\|_2^2 + \|\boldsymbol{\beta}\|_2^2\big) = \min_{\mathbf{u}\in\mathbb{R}^n} \mathcal{L}(\mathbf{u}) + \lambda\sum_{i=1}^t q(\|\mathbf{u}|_{T_i}\|_2)$$

where $q(a) = W(2a^2)^2/4 + W(2a^2)/2$ *and* W *is the Lambert* W *function, i.e., the inverse of* $f(W) = We^W$.

2.1.2 ℓ_1-regularized masks

Next, we consider the idea of applying a sparsity-inducing regularization on a mask (see, e.g., the work of Yang et al. [2019]). We show that by regularizing the mask instead of the parameters themselves, the resulting optimization leads to a "more nonconvex" regularizer.

https://doi.org/10.52202/079017-2354

Lemma 2.2 (ℓ_1-regularized masks as ℓ_q regularization)**.**

$$\min_{\mathbf{w}\in\mathbb{R}^t,\boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\{\mathbf{w}_i\boldsymbol{\beta}|_{T_i}\}_{i=1}^t) + \lambda\big(\|\mathbf{w}\|_1 + \|\boldsymbol{\beta}\|_2^2\big) = \min_{\mathbf{u}\in\mathbb{R}^n} \mathcal{L}(\mathbf{u}) + \frac{3}{2}2^{1/3}\lambda\sum_{i=1}^t \|\mathbf{u}|_{T_i}\|_2^{2/3}$$

2.1.3 Powerpropagation

Finally, we study differentiable pruning techniques that use the network weights themselves as importance scores. The most straightforward implementation of this idea is to square each of the weights, as explored in works such as powerpropagation for neural networks [Schwarz et al., 2021], but more complex versions have also been considered [Cho et al., 2023]. We show how these techniques can be generalized to handle the group setting, and show how they can also be interpreted as an implementation of group sparsity-inducing regularization.

Lemma 2.3 (Group powerpropagation as Group LASSO)**.**

$$\min_{\mathbf{w}\in\mathbb{R}^t,\boldsymbol{\beta}\in\mathbb{R}^n} \mathcal{L}(\{\|\boldsymbol{\beta}|_{T_i}\|_2\boldsymbol{\beta}|_{T_i}\}_{i=1}^t) + \lambda\|\boldsymbol{\beta}\|_2^2 = \min_{\mathbf{u}\in\mathbb{R}^n} \mathcal{L}(\mathbf{u}) + \lambda\sum_{i=1}^t \|\mathbf{u}|_{T_i}\|_2$$

2.2 Unique sparse global minima

We will prove the following theorem in this section, which establishes natural conditions for which nonconvex regularization of a convex function produces a unique group-sparse global minimum. As discussed in Section 1.2.2, this theorem is the main crucial result for proving that local search algorithms give provable guarantees for sparse convex optimization.

Theorem 1.1 (Unique sparse global minima)**.** *Let $q : \mathbb{R}_+ \to \mathbb{R}_+$ be strictly increasing, subadditive (i.e., $q(a+b) \le q(a) + q(b)$ for $a, b \in \mathbb{R}^+$), and satisfy $q(0) = 0$. If (2) has a unique minimizer $\boldsymbol{\beta}^*$ with group sparsity at most 1, then $\boldsymbol{\beta}^*$ is also the unique minimizer for (1).*

We have the following lemma that shows that if q is strictly increasing and subadditive, then the group q-regularization is always larger than group LASSO regularization. Thus, the group LASSO objective is always a lower bound on the q-regularized objective.

Lemma 2.4. *Let $q : \mathbb{R}_+ \to \mathbb{R}_+$ be strictly increasing and subadditive. Then,*

$$\sum_{i=1}^t \|\boldsymbol{\beta}|_{T_i}\|_2 \le q^{-1}\left(\sum_{i=1}^t q(\|\boldsymbol{\beta}|_{T_i}\|_2)\right)$$

Proof. Since q is invertible, applying the subadditivity condition on $q(\sum_{i=1}^t \|\boldsymbol{\beta}|_{T_i}\|_2)$ and then applying q^{-1} on both sides of the inequality yields the result. □

Furthermore, note that for solutions $\boldsymbol{\beta}$ that have group sparsity at most 1, the group q-regularization has the same value as the group LASSO regularization. That is, the lower bound value can be achieved when the group sparsity is at most 1.

Lemma 2.5. *Let $q : \mathbb{R}_+ \to \mathbb{R}_+$ be strictly increasing and satisfy $q(0) = 0$. Then, for any $\boldsymbol{\beta} \in \mathbb{R}^n$ with group sparsity 1,*

$$\sum_{i=1}^t \|\boldsymbol{\beta}|_{T_i}\|_2 = q^{-1}\left(\sum_{i=1}^t q(\|\boldsymbol{\beta}|_{T_i}\|_2)\right).$$

Proof. If $\boldsymbol{\beta}$ has group sparsity at most 1, say supported on T_j for some $j \in [t]$, then we have

$$q^{-1}\left(\sum_{i=1}^t q(\|\boldsymbol{\beta}|_{T_i}\|_2)\right) = q^{-1}\big(q(\|\boldsymbol{\beta}|_{T_j}\|_2)\big) = \|\boldsymbol{\beta}|_{T_j}\|_2.$$ □

Together, Lemmas 2.4 and 2.5 imply that if the group LASSO objective has a unique sparse minimum, then this is a lower bound on the optimal value that can be achieved by the q-regularized objective. This proves Theorem 1.1. The formal argument can be found in Appendix A.

3 The SequentialAttention++ algorithm

Weight magnitude is a simple and reliable importance score used to prune candidates (in our case, blocks) in a sparse optimization problem. In many cases, however, the magnitudes do not correlate very well with the true importances of the candidates. This has been observed e.g. in Axiotis and Sviridenko [2021, 2022], who showed that the magnitude pruning criterion used in the IHT algorithm is provably suboptimal even for simple sparse regression tasks, and proposed an adaptive weight decay to deal with this issue. One reason for the suboptimality of magnitude pruning is that the weights are not encouraged to be sparse during model training, leading to redundancy. Methods such as Powerpropagation [Schwarz et al., 2021] and Sequential Attention [Yasuda et al., 2023] have been proposed to address this issue by explicitly encoding a non-convexity that encourages weights to be concentrated on a sparse subset (this can be viewed as weight re-parameterization or concave regularization, as shown in Section 2).

To test the hypothesis that softmax attention weights are higher-quality importance scores, we consider one-shot block pruning based on the softmax attention scores used in Sequential Attention (see Figure 1 on how to apply it to blocks), and we compare it with block magnitude (Frobenius norm) pruning. The results in Figure 2a suggest that softmax attention scores are generally more reliable as block importance scores, especially for larger block sizes. This leads us to adopt the softmax parameterization in our algorithm.

As observed e.g. in Peste et al. [2021], one-shot pruning approaches are significantly suboptimal compared to iterative pruning approaches such as ACDC. We use a similar alternating compressed and decompressed phases approach as ACDC, but we apply it *on the softmax attention weights* instead of the block magnitudes. This establishes SequentialAttention++ as a combination between Sequential Attention and ACDC. The basic algorithm can be seen in Algorithm 2.

Algorithm 1 Feed-forward layer with the basic version of SequentialAttention++ to select top k parameters from a kernel $\mathbf{W}$.

function FF($\mathbf{X} \in \mathbb{R}^{b \times n}$: input batch, t : training step)
  Trainable params:
  Kernel $\mathbf{W} \in \mathbb{R}^{n \times m}$, Logits $\mathbf{L} \in \mathbb{R}^{n \times m}$

  $\mathbf{A} = nm \cdot e^{\mathbf{L}} / \sum e^{\mathbf{L}}$
  $\hat{\mathbf{W}} = \mathbf{W} \odot \mathbf{A} \odot \text{Mask}(\mathbf{A}, t)$
  return $\mathbf{X}\hat{\mathbf{W}}$
end function

Algorithm 2 Attention mask. We omit SPARSIFICATION phases for simplicity.

function Mask($\mathbf{A}$: attention weights, t : training step)
  Non-trainable state: mask $\in \{0, 1\}^{n \times m}$

  if t is in a DENSE phase **then**
    mask $\leftarrow \text{top}_k(\mathbf{A})$
    return $\mathbf{1}_{n \times m}$
  else if t is in a SPARSE phase **then**
    return mask
  end if
end function

3.1 The SPARSIFICATION phase

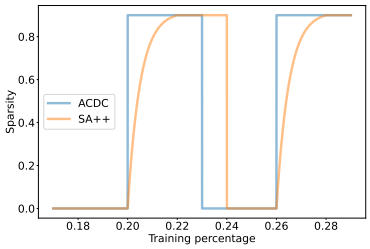
One drawback of sparse/dense (compression/decompression) phases is that the dense-to-sparse transition is abrupt. Since the lowest-magnitude weights are instantly pruned, this neglects correlations between these pruned parameters. If we were to re-train the model after pruning one parameter at a time, the picture could be drastically different, since low-magnitude weights could grow (this could happen e.g. due to parameter redundancy). In fact, this effect was highlighted by Kuznedelev et al. [2023a], who devised a backward selection method based on correlations as captured by the Hessian.

Inspired by this approach, we incorporate a backward selection phase between the DENSE and SPARSE phases, which we call the SPARSIFICATION phase. In this phase, we gradually prune the least important features based on the attention weights. This gradual process allows the model to re-adjust the attention weights after some parameters are pruned. The importance of this phase is validated by ablation experiments in Appendix D.1. We use an exponential pruning schedule, to prune more aggressively in the beginning of the phase, and more carefully at the end (as we approach the desired number of candidates k). A comparison of the sparsity schedules of ACDC and SequentialAttention++ can be found in Figure 2b. We use the sparsity schedule $\text{sparsity}(t) = s \cdot \frac{1-e^{-ct}}{1-e^{-c}}$ for $t \in [0, 1]$,

https://doi.org/10.52202/079017-2354

Validation Accuracy							
Block size: 8×8				Block size: 16×16			
70%	80%	90%	95%	70%	80%	90%	95%
−0.12	−0.11	+0.10	—	+0.19	+0.13	−0.21	−1.33
Block size: 32×32				Block size: 64×64			
68%	78%	88%	92%	58%	66%	74%	79%
+0.32	+0.58	+0.71	+3.17	+2.54	+2.81	+2.85	+5.54

(a) Quantifying the effectiveness of magnitude vs softmax attention as block importance scores (ResNet50 on ImageNet). For different block sizes and sparsities, we show the difference between the validation accuracy (in percentage points) of a model pruned one-shot based on the softmax attention scores minus one pruned based on the block magnitudes (Frobenius norms). We train dense models for the first half of training, prune once, and then continue to train the remaining blocks for the second half of training. The experimental setup is as in Section 4.



(b) Sparsity schedules of ACDC and SequentialAttention++. ACDC uses an instant dense-to-sparse transition, while SequentialAttention++ uses an exponential sparsity schedule.

Figure 2: (a) Softmax attention vs magnitude pruning, and (b) the SPARSIFICATION phase.

where s is the target sparsity. This interpolates between sparsity 0 and s, and constitutes a single SPARSIFICATION phase. We choose the constant $c = 4$ (for an ablation analysis, see Appendix D.2).

4 Experiments

We evaluate our algorithms on sparsification tasks where a dense DNN is approximated by block-sparse counterparts, at various block sizes B and sparsities p, where a sparsity p indicates that the DNN layer will only have a $1 - p$ fraction of nonzero entries, and a block size of B indicates that the nonzero entries are arranged in $B \times B$ blocks. Note that for a fixed sparsity, larger block sizes generally translate to improved efficiency due to improved hardware utilization, but also degrades quality. Block size of 1 corresponds to unstructured pruning. Our experiments are performed on the ImageNet and Criteo datasets. More details on the setup can be found in Section C.1.

4.1 Baseline algorithms

We compare our SequentialAttention++ algorithm to three other representative prior algorithms for DNN pruning. The first is basic magnitude pruning, which is a popular and effective algorithm where the weights are sparsified by keeping the weights with the largest magnitude after training [Frankle and Carbin, 2019]. We use it in the block setting by keeping the largest blocks in Frobenius norm. The second algorithm is a block generalization of Powerpropagation [Schwarz et al., 2021], which combines magnitude pruning with a differentiable pruning technique where sparsity is encouraged by squaring the weights. While the original Powerpropagation algorithm did not handle the block sparsification setting, we show that multiplying each block by the Frobenius norm leads to a provable generalization (see Lemma 2.3). Finally, we consider ACDC [Peste et al., 2021], which is an adaptation of iterative hard thresholding (IHT) [Blumensath and Davies, 2009] to the setting of neural network sparsification, and has produced the state-of-the-art pruning results for ImageNet [Kuznedelev et al., 2023b]. For all algorithms and datasets, we include a fine-tuning phase at the end of training, using the pruned model, and evaluate the final pruned model on the test set.

4.2 Results

Our results on ImageNet are summarized in Table 1. The sparsities range over 58-95% and the block sizes over $8, 16, 32, 64$. We compare ACDC and SequentialAttention++. Our ACDC implementation closely follows the implementation in Peste et al. [2021][3]. We use the phase schedule suggested by Kuznedelev et al. [2023b] (10% dense, 7 equal SPARSE-DENSE phases where the last dense phase is extended by 5%, 15% sparse). For SequentialAttention++, we additionally replace each sparse-dense

[3]We sanity-checked our ACDC implementation by verifying that the accuracy of 90% unstructured global pruning matches that of the ACDC paper (75.01 vs 75.03).

Table 1: Block sparsification of ResNet50 on ImageNet. Our dense baseline validation accuracy is 76.90. The dashes are results where the algorithms diverged because of extreme sparsity. The sparsities where chosen as 70%, 80%, 90%, 95%. As seen in the table, for larger block sizes the real sparsity is lower because we are only sparsifying layers with at least 100 blocks.

	Validation Accuracy							
	Block size: 8×8				Block size: 16×16			
Sparsity:	70%	80%	90%	95%	70%	80%	90%	95%
ACDC	74.11	72.47	67.74	—	74.08	72.56	68.61	61.42
SequentialAttention++ (ours)	**74.14**	**72.90**	**69.56**	—	**74.40**	**73.50**	**69.92**	**64.27**
	Block size: 32×32				Block size: 64×64			
Sparsity:	68%	78%	88%	92%	58%	66%	74%	79%
ACDC	74.40	72.39	68.96	63.03	75.18	74.49	71.95	67.36
SequentialAttention++ (ours)	**74.82**	**73.78**	**70.82**	**65.41**	**75.53**	**74.52**	**72.76**	**70.30**

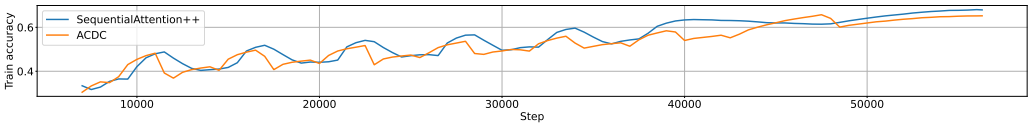


Figure 3: Training accuracy vs step on ImageNet: Comparison between ACDC and SequentialAttention++. The setting is 90% sparsity and 32×32-size blocks.

phase by a SPARSIFICATION-SPARSE-DENSE phase, as described in Section 3.1, and we replace the last of the 7 phases (including its extension) by a SPARSIFICATION phase. We use a batch size of 2048 and a maximum learning rate of 0.8.

We observe that SequentialAttention++ generally outperforms ACDC on the block sparsification task, across all different block sizes and sparsities that we tested. It should be mentioned that this comes at the cost of introducing additional trainable parameters to the model (one parameter per block). This overhead could be concerning in some applications if block size is too small (e.g., 1), in which case the model's parameters are being doubled. However, the overhead is negligible for larger (e.g., ≥ 8) block sizes.

Our results on the Criteo dataset are presented in Table 2. The sparsities range over $p \in \{90\%, 95\%, 97\%, 98\%, 99\%\}$ and block sizes over $B \in \{5, 10, 20\}$. In this experiment, we used a schedule of 10 sparse-dense phases, in addition to a 20% initial dense phase and a final 20% sparse phase. Note that for this experiment, we used masking instead of pruning for ACDC, meaning that unselected blocks are not pruned but multiplied with an all-zero mask. We observe that SequentialAttention++ is the best performing algorithm. In fact, we notice that the gap widens with large block sizes and high sparsity, suggesting that SequentialAttention++ is a highly accurate block sparsification algorithm for large block sizes and extreme sparsities.

5 Conclusion

In this work, we unified, generalized, and improved prior approaches to neural network pruning via a framework which combines differentiable pruning with combinatorial optimization algorithms, in particular local search techniques. Theoretically, we gave a unified analysis of a wide class of existing techniques via a connection to nonconvex regularization, and proved novel properties about sparse convex optimization with nonconvex regularization. In particular, we established natural conditions under which nonconvex regularization yields a unique group-sparse global minimum that is supported on the group that maximizes the ℓ_2 norm of the gradient, thus yielding provable guarantees for group sparse convex optimization. Empirically, we proposed a novel algorithm, SequentialAttention++, which outperforms prior methods on standard benchmark datasets for neural network sparsification.

We conclude with a few open directions which we believe to be interesting for future work. The first is on characterizing the nature of critical points and local minima of nonconvex-regularized convex problems. This would be a more practically useful variation on our result, which only establishes provable guarantees for the global minimizer. For our second question, we ask whether

https://doi.org/10.52202/079017-2354

one can theoretically establish that nonconvex regularization yields *better* optimization guarantees than the LASSO. In our work, we have only shown that the quality of solutions found by nonconvex regularization can match the LASSO for a wide variety of nonconvex regularizers, but we do not theoretically establish that this formulation is better. It would be interesting to show, e.g., that nonconvex regularizers allow for faster convergence to the sparse global minimizer.

References

Ehsan Amid and Manfred K. Warmuth. Reparameterizing mirror descent as gradient descent. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/604b37ea63ea51fa5fb3d8a89ec056e6-Abstract.html.

Ehsan Amid and Manfred K. Warmuth. Winnowing with gradient descent. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 163–182. PMLR, 2020b. URL http://proceedings.mlr.press/v125/amid20a.html.

Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM J. Emerg. Technol. Comput. Syst.*, 13(3):32:1–32:18, 2017. doi: 10.1145/3005348. URL https://doi.org/10.1145/3005348.

Kyriakos Axiotis and Maxim Sviridenko. Sparse convex optimization via adaptively regularized hard thresholding. *Journal of Machine Learning Research*, 22(122):1–47, 2021.

Kyriakos Axiotis and Maxim Sviridenko. Iterative hard thresholding with adaptive regularization: Sparser solutions without sacrificing runtime. In *International Conference on Machine Learning*, pages 1175–1197. PMLR, 2022.

Kyriakos Axiotis and Taisuke Yasuda. Performance of ℓ_1 regularization for sparse convex optimization. *CoRR*, abs/2307.07405, 2023. doi: 10.48550/ARXIV.2307.07405. URL https://doi.org/10.48550/arXiv.2307.07405.

Riade Benbaki, Wenyu Chen, Xiang Meng, Hussein Hazimeh, Natalia Ponomareva, Zhe Zhao, and Rahul Mazumder. Fast as CHITA: neural network pruning with combinatorial optimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2031–2049. PMLR, 2023. URL https://proceedings.mlr.press/v202/benbaki23a.html.

Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Minsik Cho, Saurabh Adya, and Devang Naik. PDP: parameter-free differentiable pruning is all you need. *CoRR*, abs/2305.11203, 2023. doi: 10.48550/ARXIV.2305.11203. URL https://doi.org/10.48550/arXiv.2305.11203.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Eustache Diemert, Julien Meynet, Pierre Galland, and Damien Lefortier. Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the ADKDD'17*, pages 1–6. 2017.

Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568, 2018.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR, 2020. URL http://proceedings.mlr.press/v119/evci20a.html.

Dean P. Foster, Howard J. Karloff, and Justin Thaler. Variable selection is hard. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 696–709. JMLR.org, 2015. URL http://proceedings.mlr.press/v40/Foster15.html.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR, 2023.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. Megablocks: Efficient sparse training with mixture-of-experts. *Proceedings of Machine Learning and Systems*, 5, 2023.

Aparna Gupte and Vinod Vaikuntanathan. The fine-grained hardness of sparse linear regression. *CoRR*, abs/2106.03131, 2021. URL https://arxiv.org/abs/2106.03131.

Marwa El Halabi, Suraj Srinivas, and Simon Lacoste-Julien. Data-efficient structured pruning via submodular optimization. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/ed5854c456e136afa3faa5e41b1f3509-Abstract-Conference.html.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1135–1143, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/ae0eb3eed39d2bcef4622b2499a05fe6-Abstract.html.

Babak Hassibi, David G. Stork, and Gregory J. Wolff. Optimal brain surgeon and general network pruning. In *Proceedings of International Conference on Neural Networks (ICNN'88), San Francisco, CA, USA, March 28 - April 1, 1993*, pages 293–299. IEEE, 1993. doi: 10.1109/ICNN.1993.298572. URL https://doi.org/10.1109/ICNN.1993.298572.

Peter D Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. *Computational Statistics & Data Analysis*, 115:186–198, 2017.

Minsoo Kang and Bohyung Han. Operation-aware soft channel pruning using differentiable masks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5122–5131. PMLR, 2020. URL http://proceedings.mlr.press/v119/kang20a.html.

Ehud D. Karnin. A simple procedure for pruning back-propagation trained neural networks. *IEEE Trans. Neural Networks*, 1(2):239–242, 1990. doi: 10.1109/72.80236. URL https://doi.org/10.1109/72.80236.

Denis Kuznedelev, Eldar Kurtic, Elias Frantar, and Dan Alistarh. Cap: Correlation-aware pruning for highly-accurate sparse vision models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Denis Kuznedelev, Eldar Kurtic, Eugenia Iofinova, Elias Frantar, Alexandra Peste, and Dan Alistarh. Accurate neural network pruning requires rethinking sparse optimization. *arXiv preprint arXiv:2308.02060*, 2023b.

 https://doi.org/10.52202/079017-2354

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 598–605. Morgan Kaufmann, 1989. URL http://papers.nips.cc/paper/250-optimal-brain-damage.

Edo Liberty and Maxim Sviridenko. Greedy minimization of weakly supermodular set functions. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 19:1–19:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi: 10.4230/LIPIcs.APPROX-RANDOM.2017.19. URL https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2017.19.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=S1eYHoC5FX.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9908–9922, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/5227b6aaf294f5f027273aebf16015f2-Abstract.html.

Zhi Gang Liu, Paul N. Whatmough, Yuhao Zhu, and Matthew Mattina. S2TA: exploiting structured sparsity for energy-efficient mobile CNN acceleration. In *IEEE International Symposium on High-Performance Computer Architecture, HPCA 2022, Seoul, South Korea, April 2-6, 2022*, pages 573–586. IEEE, 2022. doi: 10.1109/HPCA53966.2022.00049. URL https://doi.org/10.1109/HPCA53966.2022.00049.

Haoyu Ma, Chengming Zhang, Lizhi Xiang, Xiaolong Ma, Geng Yuan, Wenkai Zhang, Shiwei Liu, Tianlong Chen, Dingwen Tao, Yanzhi Wang, Zhangyang Wang, and Xiaohui Xie. HRBP: Hardware-friendly regrouping towards block-based pruning for sparse CNN training. In *Conference on Parsimony and Learning (Proceedings Track)*, 2023. URL https://openreview.net/forum?id=VP1XrdzOBp.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995. ISSN 0097-5397. doi: 10.1137/S0097539792240406. URL https://doi.org/10.1137/S0097539792240406.

Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. Deep learning recommendation model for personalization and recommendation systems. *CoRR*, abs/1906.00091, 2019. URL http://arxiv.org/abs/1906.00091.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

Alexandra Peste, Eugenia Iofinova, Adrian Vladu, and Dan Alistarh. AC/DC: alternating compressed/decompressed training of deep neural networks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8557–8570, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/48000647b315f6f00f913caa757a70b3-Abstract.html.

Jeff Pool and Chong Yu. Channel permutations for N: M sparsity. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13316–13327, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6e8404c3b93a9527c8db241a1846599a-Abstract.html.

Eric Price, Sandeep Silwal, and Samson Zhou. Hardness and algorithms for robust and sparse optimization. In *International Conference on Machine Learning*, pages 17926–17944. PMLR, 2022.

Chen Qiao, Yan Shi, Yu-Xian Diao, Vince D. Calhoun, and Yu-Ping Wang. Log-sum enhanced sparse deep neural network. *Neurocomputing*, 407:206–220, 2020. doi: 10.1016/J.NEUCOM.2020.04.118. URL https://doi.org/10.1016/j.neucom.2020.04.118.

Ramchalam Kinattinkara Ramakrishnan, Eyyüb Sari, and Vahid Partovi Nia. Differentiable mask for pruning convolutional and recurrent networks. In *17th Conference on Computer and Robot Vision, CRV 2020, Ottawa, ON, Canada, May 13-15, 2020*, pages 222–229. IEEE, 2020. doi: 10.1109/CRV50864.2020.00037. URL https://doi.org/10.1109/CRV50864.2020.00037.

Bhaskar D. Rao and Kenneth Kreutz-Delgado. An affine scaling methodology for best basis selection. *IEEE Trans. Signal Process.*, 47(1):187–200, 1999. doi: 10.1109/78.738251. URL https://doi.org/10.1109/78.738251.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. Movement pruning: Adaptive sparsity by fine-tuning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/eae15aabaa768ae4a5993a8a4f4fa6e4-Abstract.html.

Pedro Savarese, Hugo Silva, and Michael Maire. Winning the lottery with continuous sparsification. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/83004190b1793d7aa15f8d0d49a13eba-Abstract.html.

Jonathan Schwarz, Siddhant M. Jayakumar, Razvan Pascanu, Peter E. Latham, and Yee Whye Teh. Powerpropagation: A sparsity inducing weight reparameterisation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28889–28903, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f1e709e6aef16ba2f0cd6c7e4f52b9b6-Abstract.html.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4 (2):107–194, 2012. doi: 10.1561/2200000018. URL https://doi.org/10.1561/2200000018.

Shai Shalev-Shwartz, Nathan Srebro, and Tong Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM J. Optim.*, 20(6):2807–2832, 2010. ISSN 1052-6234. doi: 10.1137/090759574. URL https://doi.org/10.1137/090759574.

Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d1ff1ec86b62cd5f3903ff19c3a326b2-Abstract.html.

Nikko Ström. Sparse connection and pruning in large dynamic artificial neural networks. In George Kokkinakis, Nikos Fakotakis, and Evangelos Dermatas, editors, *Fifth European Conference*

on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997, pages 2807–2810. ISCA, 1997. doi: 10.21437/EUROSPEECH.1997-708. URL https://doi.org/10.21437/Eurospeech.1997-708.

Arun Sai Suggala, Adarsh Prasad, and Pradeep Ravikumar. Connecting optimization and regularization paths. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10631–10641, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/6459257ddab7b85bf4b57845e875e4d4-Abstract.html.

Georg Thimm and Emile Fiesler. Evaluating pruning methods. In *Proceedings of the International Symposium on Artificial neural networks*, pages 20–25, 1995.

Jitendra K. Tugnait. Sparse-group log-sum penalized graphical model learning for time series. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 5822–5826. IEEE, 2022. doi: 10.1109/ICASSP43922.2022.9747446. URL https://doi.org/10.1109/ICASSP43922.2022.9747446.

Antoine Vanderschueren and Christophe De Vleeschouwer. Are straight-through gradients and soft-thresholding all you need for sparse training? In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pages 3797–3806. IEEE, 2023. doi: 10.1109/WACV56688.2023.00380. URL https://doi.org/10.1109/WACV56688.2023.00380.

Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2968–2979, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5cf21ce30208cfffaa832c6e44bb567d-Abstract.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Anna Korhonen, David R. Traum, and Lluís Màrquez, 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 5797–5808. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1580. URL https://doi.org/10.18653/v1/p19-1580.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2074–2082, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/41bfd20a38bb1b0bec75acf0845530a7-Abstract.html.

David Wipf and Srikantan Nagarajan. Solving sparse linear inverse problems: Analysis of reweighted l1 and l2 methods. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

Xia Xiao, Zigeng Wang, and Sanguthevar Rajasekaran. Autoprune: Automatic network pruning by regularizing auxiliary parameters. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information*

Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 13681–13691, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4efc9e02abdab6b6166251918570a307-Abstract.html.

Chen Yang, Zhenghong Yang, Abdul Mateen Khattak, Liu Yang, Wenxin Zhang, Wanlin Gao, and Minjuan Wang. Structured pruning of convolutional neural networks via L1 regularization. *IEEE Access*, 7:106385–106394, 2019. doi: 10.1109/ACCESS.2019.2933032. URL https://doi.org/10.1109/ACCESS.2019.2933032.

Taisuke Yasuda, Mohammad Hossein Bateni, Lin Chen, Matthew Fahrbach, Gang Fu, and Vahab Mirrokni. Sequential attention for feature selection. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=TTLLGx3eet.

Yuxin Zhang, Mingbao Lin, Mengzhao Chen, Fei Chao, and Rongrong Ji. Optg: Optimizing gradient-driven criteria in network sparsity. *arXiv preprint arXiv:2201.12826*, 2022.

Xin Zhou, Xiao-Wen Liu, Gong Zhang, Luliang Jia, Xu Wang, and Zhiyuan Zhao. An iterative threshold algorithm of log-sum regularization for sparse problem. *IEEE Trans. Circuits Syst. Video Technol.*, 33(9):4728–4740, 2023. doi: 10.1109/TCSVT.2023.3247944. URL https://doi.org/10.1109/TCSVT.2023.3247944.

A Missing proofs from Section 2

Proof of Lemma 2.1. Note first that for a fixed $a > 0$, the function $w \mapsto w^2 + a^2/\exp(2w)$ is minimized at w satisfying $2w - 2a^2\exp(-2w) = 0$, that is, $w = W(2a^2)/2$. Then, for each group $i \in [t]$, we can set $\mathbf{u}|_{T_i} = \exp(\mathbf{w}_i)\boldsymbol{\beta}|_{T_i}$ so

$$\mathbf{w}_i^2 + \|\boldsymbol{\beta}|_{T_i}\|_2^2 = \mathbf{w}_i^2 + \frac{\|\mathbf{u}|_{T_i}\|_2^2}{\exp(2\mathbf{w}_i)} \geq w^2 + w$$

where $w = W(2\|\mathbf{u}|_{T_i}\|_2^2)/2$. Summing over the groups $i \in [t]$ gives the desired result. □

Proof of Lemma 2.2. Note first that for a fixed $a > 0$, the function $w \mapsto w + a^2/w^2$ is minimized at w satisfying $1 - 2a^2w^{-3} = 0$, that is, $w = 2^{1/3}a^{2/3}$. Then, for each group $i \in [t]$, we can set $\mathbf{u}|_{T_i} = \mathbf{w}_i\boldsymbol{\beta}|_{T_i}$ so

$$|\mathbf{w}_i| + \|\boldsymbol{\beta}|_{T_i}\|_2^2 = |\mathbf{w}_i| + \frac{\|\mathbf{u}|_{T_i}\|_2^2}{\mathbf{w}_i^2} \geq \frac{3}{2}w$$

where $w = 2^{1/3}\|\mathbf{u}|_{T_i}\|_2^{2/3}$. Summing over the groups $i \in [t]$ gives the desired result. □

Proof of Lemma 2.3. Set $\mathbf{u}|_{T_i} = \|\boldsymbol{\beta}|_{T_i}\|_2\boldsymbol{\beta}|_{T_i}$. Then,

$$\|\mathbf{u}|_{T_i}\|_2 = \|\|\boldsymbol{\beta}|_{T_i}\|_2\boldsymbol{\beta}|_{T_i}\|_2 = \|\boldsymbol{\beta}|_{T_i}\|_2^2$$

so summing over the groups gives the claimed result. □

A.1 Unique sparse global minima

Proof of Theorem 1.1. Suppose that the optimal group LASSO solution $\boldsymbol{\beta}^*$ of objective (2) has group sparsity at most 1. Then for any other solution $\boldsymbol{\beta}'$, we have that

$$\begin{aligned}
& \mathcal{L}(\boldsymbol{\beta}') + \lambda q^{-1}\left(\sum_{i=1}^{t} q(\|\boldsymbol{\beta}'|_{T_i}\|_2)\right) \\
\geq\ & \mathcal{L}(\boldsymbol{\beta}') + \lambda\sum_{i=1}^{t}\|\boldsymbol{\beta}'|_{T_i}\|_2 && \text{by Lemma 2.4} \\
>\ & \mathcal{L}(\boldsymbol{\beta}^*) + \lambda\sum_{i=1}^{t}\|\boldsymbol{\beta}^*|_{T_i}\|_2 && \text{by optimality}
\end{aligned}$$

 https://doi.org/10.52202/079017-2354

$$= \mathcal{L}(\boldsymbol{\beta}^*) + \lambda q^{-1}\left(\sum_{i=1}^{t} q(\|\boldsymbol{\beta}^*|_{T_i}\|_2)\right) \qquad \text{by Lemma 2.5.}$$

Thus, $\boldsymbol{\beta}^*$ must be the unique minimizer of (1). □

B OMPR via nonconvex regularization

We show that our results from Section 2.2 together with recent work of Axiotis and Yasuda [2023] give provable guarantees for a local search algorithm based on orthogonal matching pursuit with replacement using nonconvex regularization.

We first introduce some definitions needed to state our result.

Definition B.1. Let $T_i \subseteq [n]$ for $i \in [t]$ form a partition of $[n]$. Then, we define

$$\|\boldsymbol{\beta}\|_{\text{group}} \coloneqq |\{i \in [t] : \boldsymbol{\beta}|_{T_i} \neq 0\}|.$$

Definition B.2 (Restricted strong convexity and smoothness)**.** Let $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ be differentiable. Let $T_i \subseteq [n]$ for $i \in [t]$ form a partition of $[n]$. Then, l is *μ_s-restricted strongly convex at group sparsity s* if for any $\boldsymbol{\beta} \in \mathbb{R}^n$ and $\boldsymbol{\Delta} \in \mathbb{R}^n$ with $\|\boldsymbol{\Delta}\|_{\text{group}} \leq s$,

$$\mathcal{L}(\boldsymbol{\beta} + \boldsymbol{\Delta}) - \mathcal{L}(\boldsymbol{\beta}) - \langle \nabla \mathcal{L}(\boldsymbol{\beta}), \boldsymbol{\Delta} \rangle \geq \frac{\mu_s}{2}\|\boldsymbol{\Delta}\|_2^2,$$

and *L_s-restricted smooth at group sparsity s* if for any $\boldsymbol{\beta} \in \mathbb{R}^n$ and $\boldsymbol{\Delta} \in \mathbb{R}^n$ with $\|\boldsymbol{\Delta}\|_{\text{group}} \leq s$,

$$\mathcal{L}(\boldsymbol{\beta} + \boldsymbol{\Delta}) - \mathcal{L}(\boldsymbol{\beta}) - \langle \nabla \mathcal{L}(\boldsymbol{\beta}), \boldsymbol{\Delta} \rangle \leq \frac{L_s}{2}\|\boldsymbol{\Delta}\|_2^2.$$

We will now obtain provable guarantees for Algorithm 3 in Theorem B.3.

Algorithm 3 OMPR via nonconvex regularization

Initialize S arbitrarily such that $|S| = k'$
for $i = 1, \ldots, R$ **do**
  Let

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{\beta}) + \lambda \cdot q^{-1}\left(\sum_{i \notin S} q(\|\boldsymbol{\beta}|_{T_i}\|_2)\right)$$

  for λ sufficiently large
  Let $i \notin S$ be the group maximizing $\hat{\boldsymbol{\beta}}|_{T_i}$ and $j \in S$ be the group minimizing $\|\boldsymbol{\beta}\|_2|_{T_j}$
  $S \leftarrow S \cup \{i\} \setminus \{j\}$
end for

Theorem B.3 (OMPR via nonconvex regularization)**.** *Let $q : \mathbb{R}_+ \to \mathbb{R}_+$ be strictly increasing, subadditive, and 0 at the origin. After R iterations of Algorithm 3 with $k' \geq k\left(\frac{L_2^2}{\mu_{k+k'}^2} + 1\right)$, for*

$$R \geq k \cdot \frac{L_2}{\mu_{k+k'}} \log \frac{\mathcal{L}(\boldsymbol{\beta}^{(0)}) - \mathcal{L}(\boldsymbol{\beta}^*)}{\varepsilon},$$

then $\hat{\boldsymbol{\beta}}$ has group sparsity $\|\boldsymbol{\beta}^\infty\|_{\text{group}} \leq k'$ and satisfies

$$\mathcal{L}(\boldsymbol{\beta}^\infty) \leq \mathcal{L}(\boldsymbol{\beta}^*) + \varepsilon\,,$$

where $\mu_{k+k'}$ is a lower bound on the restricted strong convexity constant of l at group sparsity $k + k'$ and L_2 is an upper bound on the restricted smoothness constant of l at group sparsity 2 (see Definition B.2).

Proof. By Theorem 1.1, if the optimization problem in Algorithm 3 with q replaced by the absolute value function has a unique minimizer with group sparsity at most 1, then $\hat{\boldsymbol{\beta}}$ is a unique global minimizer with group sparsity at most 1, and coincides with this Group LASSO solution. Lemma 3.2 of Axiotis and Yasuda [2023] then establishes that this solution is supported on the group that maximizes the ℓ_2 norm of the gradient, which in turn implies Theorem B.3 via guarantees for the group orthogonal matching pursuit with replacement algorithm (Corollary A.10 of Axiotis and Yasuda [2023]). □

Table 2: Block sparsification on Criteo. The validation losses are an average of three runs. Our dense baseline validation loss is 0.4489.

	Validation Loss		
Sparsity: 90%	Block size: 5	Block size: 10	Block size: 20
Magnitude	0.4523	0.4693	0.4923
PowerPropagation	0.4521	0.4572	0.4920
ACDC	0.4517	0.4580	0.4829
SequentialAttention++ (ours)	**0.4515**	**0.4535**	**0.4596**
Sparsity: 95%	Block size: 5	Block size: 10	Block size: 20
Magnitude	0.4586	0.4892	0.4998
PowerPropagation	0.4547	0.4768	0.4946
ACDC	0.4547	0.4754	0.4961
SequentialAttention++ (ours)	**0.4540**	**0.4595**	**0.4715**
Sparsity: 97%	Block size: 5	Block size: 10	Block size: 20
Magnitude	0.4656	0.5004	0.5079
PowerPropagation	0.4587	0.5061	0.5093
ACDC	0.4606	0.4936	0.5056
SequentialAttention++ (ours)	**0.4570**	**0.4708**	**0.4865**
Sparsity: 98%	Block size: 5	Block size: 10	Block size: 20
Magnitude	0.4717	0.5145	0.5447
PowerPropagation	0.4622	0.5158	0.5379
ACDC	0.4692	0.4929	0.5184
SequentialAttention++ (ours)	**0.4601**	**0.4904**	**0.5162**
Sparsity: 99%	Block size: 5	Block size: 10	Block size: 20
Magnitude	0.4881	0.5376	0.5482
PowerPropagation	0.5017	0.5295	0.5425
ACDC	0.5050	0.5153	0.5427
SequentialAttention++ (ours)	**0.4803**	**0.5068**	**0.5253**

C Additional details on experiments

C.1 Experimental setup

ImageNet [Deng et al., 2009]. ImageNet is the most widely used vision dataset and is considered as the de facto benchmark in the neural network pruning literature, culminating in the state of the art results in Kuznedelev et al. [2023b]. We use ResNet50 and a standard training setup (90 epochs, SGD with cosine learning rate and momentum, weight decay). We reshape the 4-dimensional ($H \times W \times C_{\text{in}} \times C_{\text{out}}$) kernel tensors used in convolutional layers to 2D matrices of shape $HWC_{\text{in}} \times C_{\text{out}}$, which define the 2D block candidates for pruning. We prune all layers uniformly, except for layers with < 100 blocks, which we do not prune at all, to avoid degeneracy at high sparsities.

Criteo [Diemert et al., 2017]. Criteo is a standard public dataset for the clickthrough rate (CTR) prediction task, which consists of 33M training examples with 13 numerical and 26 categorical features. The model we sparsify is a standard fully connected DNN with three 400-width layers and an additional embedding layer to transform each input feature into an embedding vector of size 10 (for a total embedding width of 390). We note that a simple MLP is often a fairly competitive model for this task [Naumov et al., 2019]. We prune the first dense layer after the embedding layer. We use Adam optimizer with a learning rate that decays exponentially from $2 \cdot 10^{-2}$ to $3 \cdot 10^{-4}$. We train to minimize the cross-entropy loss for 25 epochs with a batch size of 32768.

C.2 Block sparsification results on Criteo

We give our block sparsification results on the Criteo dataset in Table 2.

https://doi.org/10.52202/079017-2354

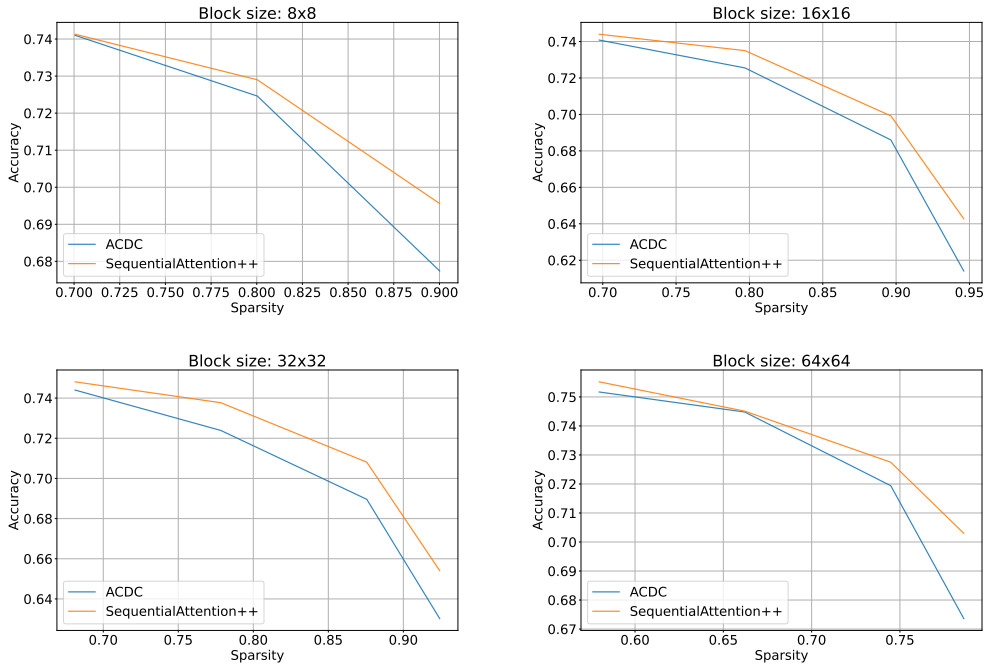


Figure 4: Block sparsification on Imagenet.

C.3 Additional tricks

In addition to the basic algorithm described in Section 3, our implementation of SequentialAttention++ incorporates several other ingredients for improved empirical performance. First, we confirm the observation of Peste et al. [2021] that resetting the optimizer between each phase of SequentialAttention++ is crucial for good performance. We note that this is also suggested by our theoretical results (Theorem B.3), which suggests that each of the dense and sparse phases should be thought of as a separate optimization problem that is solved independently. Similarly to Kuznedelev et al. [2023b], we also observe that weight decay significantly boosts performance, even when applied to the attention logits.

Second, we observe that pruning each layer of the network separately performs better than a global pruning algorithm which attempts to prune all layers at once. We suggest that this may be the case due to "bottlenecking" behavior, where a global pruning algorithm may choose to almost completely eliminate a layer which may destroy the connectivity of the neural network. While this is not the case when pruning individual parameters, pruning large blocks can easily eliminate a layer. We use uniform sparsity across layers, but choose not to sparsify layers containing less than 100 blocks. This is because layers have greatly varying sizes, and want to avoid a sharp quality drop from overpruning smaller layers, which was observed in experiments. Finally, we clip attention weights to the range $[n \cdot \text{density}, n/\text{density}]$ to avoid them becoming too small or too large.

C.4 Additional results

We provide additional plots for our experiments in Figures 4 and 5. In Figure 4, we plot tradeoffs between the validation accuracy and weight matrix sparsity for SequentialAttention++ and ACDC Peste et al. [2021]. In Figure 5, we plot tradeoffs between the validation loss and AUC against weight matrix sparsity for SequentialAttention++ and our three baseline algorithms of Magnitude Pruning, Powerpropagation Schwarz et al. [2021], and ACDC Peste et al. [2021].

https://doi.org/10.52202/079017-2354

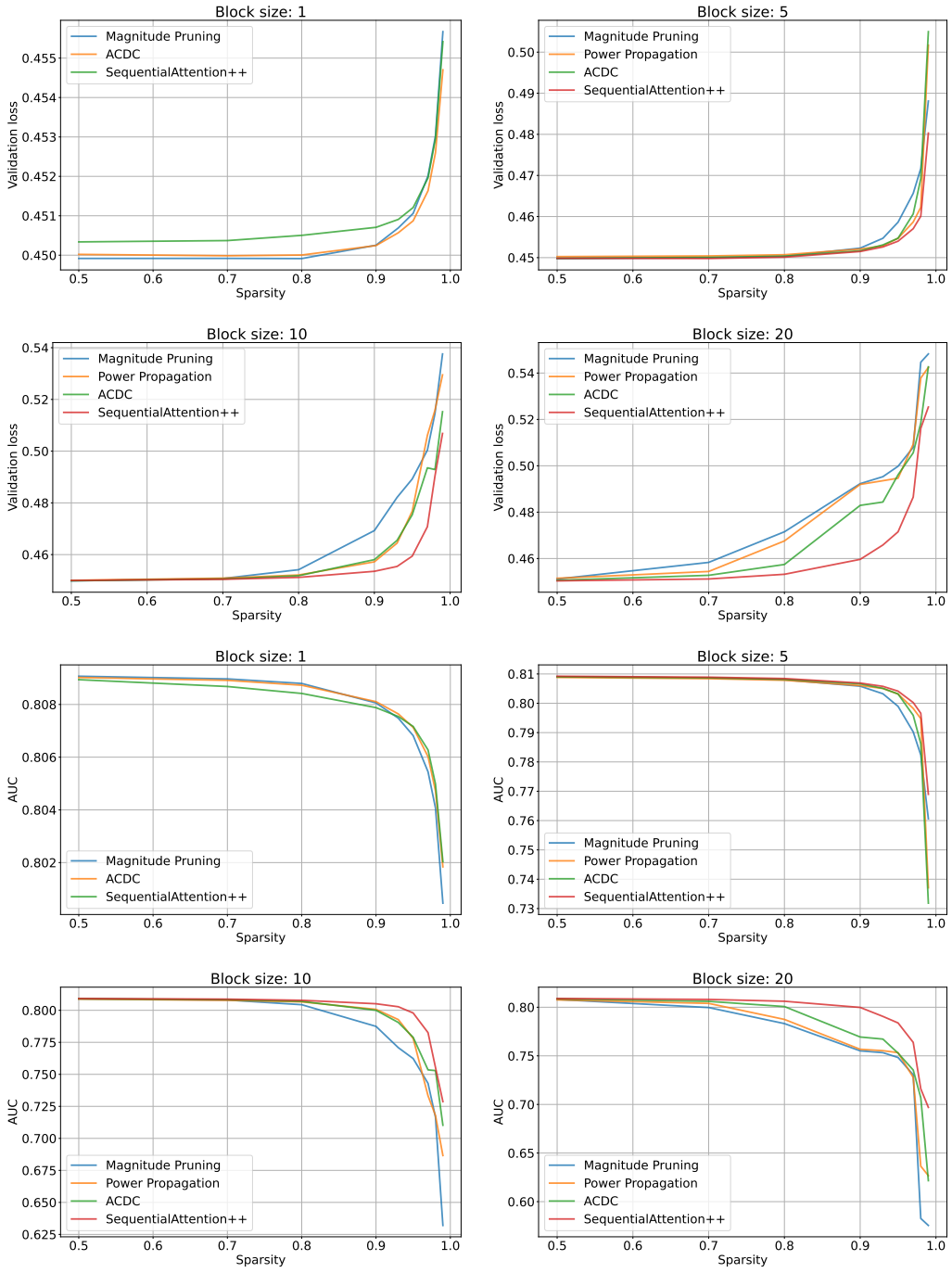


Figure 5: Block sparsification on Criteo. There are no Powerpropagation results for block size 1 because the algorithm diverged.

D Ablations

D.1 Importance of the SPARSIFICATION phase.

We perform experiments to study the effect of the SPARSIFICATION phase, as described in Section 3.1, to the final accuracy. To that end, we remove the SPARSIFICATION phase and only apply alternating

https://doi.org/10.52202/079017-2354

Table 3: Removing the SPARSIFICATION phase from SequentialAttention++. The results show validation accuracy for training block-sparse ResNet50 on ImageNet. We use the same sparsities as in Table 1.

	Validation Accuracy							
	Block size: 8×8				Block size: 16×16			
Sparsity:	70%	80%	90%	95%	70%	80%	90%	95%
Validation Accuracy	—	0.61922	0.61405	0.69678	0.68152	0.68658	0.70079	0.72845
Diff from baseline	—	−0.0235	−0.04006	−0.00624	−0.01408	−0.01262	−0.00738	+0.00089
	Block size: 32×32				Block size: 64×64			
Sparsity:	68%	78%	88%	92%	58%	66%	74%	79%
Validation Accuracy	0.72333	0.72666	0.73346	0.7432	0.74194	0.74099	0.74268	0.75104
Diff from baseline	−0.00569	−0.00837	−0.00429	−0.00196	+0.00059	−0.00301	−0.00547	−0.00421

Table 4: Modifying the exponent constant in the schedule of the SPARSIFICATION phase. Block-sparse training of ResNet50 on ImageNet for 90% sparsity.

	Validation Accuracy			
Block size	8×8	16×16	32×32	64×64
$c = 2$	0.69403	0.69613	0.70614	0.72264
$c = 4$	0.6956	0.6992	0.70817	0.72756
$c = 8$	0.69202	0.70036	0.70976	0.72614

DENSE and SPARSE phases, each of equal duration. The final phase before the last fine-tuning is now a DENSE phase.

The results in Table 3 show that, on average over different block sizes and densities, removing the SPARSIFICATION phase decreases validation accuracy by 0.009, or 0.9 percentage points. We conclude that the SPARSIFICATION phase is an important feature of SequentialAttention++.

D.2 Choice of the SPARSIFICATION exponent.

In this section, we try different values of the constant used in the exponent of the schedule of the SPARSIFICATION operation. We remind that during a SPARSIFICATION phase, the sparsity varies as $\text{sparsity}(t) = s \cdot \frac{1-e^{-ct}}{1-e^{-c}}$ for $t \in [0, 1]$, where s is the target sparsity. The constant c determines how non-linearly the sparsity interpolates from 0 to s.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We support our main claims with theorems and empirical results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss some limitations in Section 5 and suggest these as research directions for future work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-2354

Justification: Our theoretical results are supported by full proofs, which can be found in the main text and the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We plan to release the code used in experiments if accepted.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We plan to release the code used in experiments if accepted. The datasets used in experiments are popular and publicly available datasets.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We plan to release the code used in experiments if accepted, which will contain this information.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-2354

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We plan to release the code used in experiments if accepted, which will contain this information.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-2354

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.