TAPERING NEURAL NETWORKS WITH TEMPORALLY AGGREGATED FIRST-ORDER IMPORTANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

In gradual pruning, network architecture is compressed simultaneously while learning the network parameters. Importance scoring of the parameters to prune allows to compare them such as for their ranking to cull a desired parameter set at a given pruning instance. In this paper we propose Temporal Aggregation of Parameter Effects for Reduction (TAPER), which uses an effective first-order estimation of weight importance on lost network capacity, combined by aggregation of importance scores for statistical robustness of gradual pruning decisions to batch variations. TAPER is evaluated favorably against the state of the art in gradual pruning in standard datasets using established metrics. It performs well both for pruning simpler CNNs such as VGG as well as more complex architectures such as ConvNeXt. Since network accuracy metrics compound major effects from training choices with pruning criteria, we also introduce Pruning Impact as a differential metric for assessing pruning criteria better isolated from cumulative training effects.

1 INTRODUCTION

In deep learning, overparameterized networks are often used to capture patterns in the data. However, this approach leads to large networks and suboptimal training and inference times. Further, strong regularization techniques need to be used to avoid overfitting. The need for smaller and more efficient AI models are increasing with the widespread use of large AI models in daily life. Several methods for compressing networks and increasing efficiency have been proposed, including neural architecture search, quantization, and pruning. Given a dense network, pruning aims to find and remove the least important parts to create a smaller architecture adapted to the given task. Pruning for adapting network architectures was first proposed quite early, around the times of early neural networks themselves. Seminal initial works include Optimal Brain Damage (OBD) (LeCun et al., 1989) and Optimal Brain Surgeon (OBS) (Hassibi & Stork, 1992).

State-of-the-art gradual methods combine weight and gradient information in two stages, ranking by one quantity before or after the other (Liu et al., 2021; Zhu et al., 2025). In this work, we further refine and develop the state-of-the-art metrics used for unstructured gradual pruning. By combining weight and gradient magnitude information into one metric, a simpler metric with better performance is created.

The main contributions of our work include: (1) First-order Taylor approximation as a single-step importance scoring for gradual pruning; (2) temporal aggregation of importance scores or parts thereof for improving pruning effectiveness; and (3) Pruning Impact as a new evaluation metric for instantaneous comparison of importance scoring approaches.

2 BACKGROUND

With the development of deep neural networks and CNNs, model size and training/inference times multiplied. Around the same time, results of pruning showed that it was possible to reduce the number of connections in AlexNet from 61 million to 6.7 million without loss of performance (Han et al., 2015). Since then, pruning has advanced to be a broad field of research, including *structured pruning*, which removes structural elements such as neurons or convolutions hence sets of param-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

eters altogether and *unstructured pruning*, which removes individual weights. For decisions on the elements and timing to prune, importance scores and schedules have been widely studied.

2.1 Pruning Schedules

Pruning can either be performed before (Lee et al., 2019), after (LeCun et al., 1989), or during training (Zhu & Gupta, 2017). Pruning after training often requires iterations of pruning and training to reach high accuracy (LeCun et al., 1989), whereas pruning before training avoids the overhead of training the full dense network (Lee et al., 2019) but yields a data-independent pruning mask that can degrade the performance of the final network (Kumar et al., 2024). To combine the advantages of both pruning before and pruning after training, gradual pruning was proposed (Zhu & Gupta, 2017). In gradual pruning, the training and pruning schemes are combined, so the network is gradually pruned during training. Pruning at a fixed interval of X batches and utilizing a cubic schedule for determining the number of weights to remove at each pruning step was proposed in Zhu & Gupta (2017), and has been adopted as the standard for gradual pruning since (Liu et al., 2021; Zhu et al., 2025).

A recent paper suggests a novel pipeline for efficient iterative pruning by reducing the cost of fine-tuning after each iteration of parameter removal (Hu et al., 2025). Their method utilizes the difference in accuracy directly before and after pruning to investigate the damage made by pruning and judge if fine-tuning is needed. This metric is similar to what we define as Pruning Impact in Section 3.3, however we propose it as an evaluation metric. Pruning after training has also been proposed as a one-shot method avoiding iterations of pruning and fine-tuning (Frantar & Alistarh, 2023).

2.2 Importance scores

Importance scores are used to rate the importance of weights or structured elements to remove. For structured pruning, it is common to calculate importance at the weight level and then aggregate to evaluate the influence of convolutions (Li et al., 2017) or other structural elements.

Early pruning works (LeCun et al., 1989; Hassibi & Stork, 1992) utilized a second-order Taylor approximation to estimate the impact of weight removal. In OBD (LeCun et al., 1989), pruning is applied to a converged network, and therefore they make the assumption that the gradient will be zero. Thus, the first-order term can be disregarded, and only the diagonal of the Hessian is used. This was later developed into the OBS framework, including information from the non-diagonal elements of the Hessian by adding an update of the remaining weights to compensate for parameters removed. The OBS framework has inspired numerous pruning algorithms, including one-shot pruning methods of LLMs (Frantar & Alistarh, 2023; Ling et al., 2024). Second-order scores can also be approximated from gradients alone by replacing the Hessian with the empirical Fisher information (Singh & Alistarh, 2020).

In (Molchanov et al., 2019), it was empirically demonstrated for structured pruning of filters that a function ($(g * w)^2$) of the first-order Taylor-expansion term can be useful for ranking even when pruning a converged network. This contradicts the assumption that the first-order term can be disregarded. Movement pruning (Sanh et al., 2020) similarly used the first-order term ($-w_i g_i$) as a signed score for iterative pruning of pretrained models aimed for transfer learning via finetuning. Both works used accumulation of the entire first-order score, *i.e.*, over several batches, but aggregate the product of weight and gradient: Molchanov et al. (2019) aggregate $(g*w)^2$ for structured pruning of filters, and Sanh et al. (2020) as a signed score in the transfer-learning setting. In contrast, TAPER targets unstructured gradual pruning when training from scratch, and aggregates weight and gradient magnitudes separately before combining them, which we empirically show to be more effective in our ablation study. Gradient magnitude combined with weight magnitude is also commonly used for pruning before training in methods such as SNIP (Lee et al., 2019) and SynFlow (Tanaka et al., 2020). In SynFlow, they define a class of importance scores based on gradient and weight information called Synaptic Saliency. Using this definition, they prove that importance scores belonging to this class will target wider layers.

For gradual pruning, a simple score for ranking is weight magnitude, first proposed by Han et al. (2015). This metric was also used in the Lottery Ticket Hypothesis paper (Frankle & Carbin, 2019),

where networks are iteratively pruned and, after pruning, the weights are reset to the original initialized values before the compressed network is trained until convergence. In later research, weight magnitude pruning is combined with regularization to improve performance (Wang et al., 2020; Zhou et al., 2024). Early gradual pruning methods relied on simple weight magnitude as a metric (Zhu & Gupta, 2017), while modern methods utilize a two-step comparison of both weight and gradient magnitudes for more well-informed parameter removal (Liu et al., 2021; Zhu et al., 2025).

3 METHODS

A neural network f, parameterized $\Theta = \{w_1, w_2, ..., w_n\} \cup \{b_1, b_2, ..., b_m\}$ with n weights and m biases can be trained on a dataset $D = \{(x_1, y_1), (x_2, y_2), ..., (x_K, y_K)\}$ with samples x_r and labels y_r. This is an optimization problem with the goal of minimizing

$$\min_{\Theta} L = \sum_{r=1}^{K} l(f(x_r; \Theta), y_r)\,. \tag{1}$$

Here, $l(\cdot, \cdot)$ denotes the loss function chosen to evaluate how close the prediction of f is to the ground truth y_r. A simple solution to the above optimization problem iteratively updates network parameters in the direction of the negative gradient, $-g$, known as gradient descent. Many sophisticated alternatives are extensions and variations involving gradients.

3.1 FIRST-ORDER TAYLOR APPROXIMATION FOR IMPORTANCE SCORING

ODB (LeCun et al., 1989) formalizes the change δE in loss when removing parameter w_i using the following Taylor approximation:

$$\delta E = \underbrace{\sum_i g_i \delta w_i}_{\text{first-order}} + \underbrace{\frac{1}{2}\sum_i h_{ii}\delta w_i^2 + \frac{1}{2}\sum_{i\neq j} h_{ij}\delta w_i \delta w_j}_{\text{second-order}} + \underbrace{O(\delta w^3)}_{\text{high-orders}}\,. \tag{2}$$

In contrast to early works utilizing near-zero gradients of converged network, gradual pruning cannot omit the gradients, making the first-order term in the above Taylor expansion the leading term in general, especially when the weight magnitudes are sufficiently small. Unlike the Hessian in the second-order terms, the gradients are already calculated for backpropagation, hence allowing for efficient implementations for importance ranking. Indeed, the utility of added information from gradients was demonstrated for gradual pruning in (Liu et al., 2021). However, since gradients may vary a lot from batch to batch and may also temporarily vanish several times during network training, earlier works proposed their use as part of a multi-step decision process, *e.g.*, ranking them separately before (Zhu et al., 2025) or after (Liu et al., 2021) the weights.

In this work, we propose an importance score inspired by the first-order approximation of the OBD Taylor expansion. By omitting the higher-order terms, this yields the following importance score approximation: $I(w_i) = |g_i| \cdot |w_i|\,.$

Note that multiplication can easily amplify noise from gradient values that vary between batches. Furthermore, when weights temporarily cross zero, *e.g.*, when switching their sign early in training, this would vanish the importance score even if the instantaneous gradient magnitude is relatively large, risking removing crucial parts of the network. To prevent such damage, we propose to use the expected values of the terms of longer temporal horizons for statistically removing the effects of noise via aggregation. To that end, we calculate the mean values of weight and gradient magnitudes throughout a predefined number of training iterations. The resulting TAPER importance score is:

$$I(w_i) = \langle |g_i| \rangle \cdot \langle |w_i| \rangle\,, \tag{3}$$

where $\langle \cdot \rangle$ denotes the mean. By aggregating weight and gradient magnitudes over several batches, our method also becomes more robust to batches with uneven class distributions.

3.2 PRUNING AND TRAINING SCHEDULES

In gradual pruning, the network is pruned and trained simultaneously and gradually. Following all pruning instances, a training-only phase without any pruning, commonly called fine-tuning, typically

follows to help the model recover from the damage caused by pruning. The number of iterations and schedules for pruning and later fine-tuning vary between approaches: For example, FGGP (Zhu et al., 2025) uses 128 epochs for pruning and 32 for fine-tuning, while GraNet (Liu et al., 2021) uses 110 for pruning and 50 for fine-tuning. Herein we follow the latter settings for comparability. During training, at every pruning iteration t between given intervals of every Δt iterations/batches, pruning removes a number of parameters to gradually increase sparsity from the initial sparsity to the intended target sparsity. We adopt a cubic function for sparsity scheduling (Zhu & Gupta, 2017) similarly to earlier gradual pruning works GraNet (Liu et al., 2021) and FGGP (Zhu et al., 2025). These and other gradual pruning works typically used a step-wise decaying learning rate (LR) for the optimizer during training. In contrast to these, we use a cosine LR decay in TAPER, inspired by its recent success such as in ConvNeXt (Liu et al., 2022).

3.3 Assessing the impact of pruning

When evaluating and comparing pruning methods, the results are affected not only by importance scoring but also by a combination of training hyperparameters and pruning/training scheduling. Evaluation in the literature is, however, often performed by comparing merely the final accuracies between different approaches. The importance score is commonly credited when a pruning algorithm is successful, although the above approach may fail to isolate the cause, since the results are easily confounded by cross interactions between the importance score choices, training hyperparameters, the pruning and training scheduling, among others. Recent research also highlights the importance of the learning rate for recovery of pruned models (Liu et al., 2023; Wang & Fu, 2023), further highlighting the need for evaluation methods that isolate the performance of the importance score. Indeed, a superior importance score paired with a too-low LR during later pruning iterations and a short fine-tuning preventing recovery might result in inferior pruning performance compared to a poorer importance score run with a better training configuration (Le & Hua, 2021).

To evaluate the instantaneous effect of pruning, we propose comparing the direct change when pruning is applied, which is termed as Pruning Impact (PrIm). PrIm at pruning iteration t is defined as $c_t = a^-_{t+1} - a^+_t$, where $-$ and $+$ indicate the beginning and end of a training sequence uninterrupted by pruning. Then, a^+_t and a^-_{t+1} denote the test accuracy of the model directly before and after pruning without parameter updates by training. To compare methods across entire training episodes, mean pruning impact can be calculated as $c = \frac{1}{n}\sum_{t=1}^{n} c_t$ with n denoting the number of pruning iterations. Herein, we utilize PrIm to better isolate the effect of impact scoring from training.

4 Experiments and Results

To evaluate the proposed method, we used architectures and datasets of different complexity. Our experiments include the following architechture/dataset pairs: VGG19/CIFAR10, VGG19/CIFAR100, ResNet50/CIFAR100 and ConvNeXt-Tiny/Tiny-ImageNet. We use the first three to enable comparison with previous papers, and we evaluate pruned VGG and ResNet models at target sparsities from 98 to 99.95% to overlap with previous gradual-pruning results and to use higher sparsities for clearer method separation. We also include experiments on ConvNeXt-Tiny as a modern, state-of-the-art architecture. All experiments are performed starting with a dense network.

Implementation For CIFAR10 and CIFAR100 experiments, the training setup from Liu et al. (2021) is replicated, and all layers are pruned. To allow fair comparison, all methods use the same number of epochs for pruning and fine-tuning for a given architecture. For GraNet on VGG-19, the original code was used, while the FGGP and other GraNet results were produced using our own codebase developed from the original FGGP code.

When training ConvNeXt-Tiny on Tiny-ImageNet, the stem is modified to fit the Tiny-ImageNet resolution. The first convolutional layer and last classification layer of ConvNeXt are left dense and model sparsity is calculated as a fraction of pruned versus prunable parameters. Further, to replicate the proposed training pipeline of ConvNeXt-Tiny, the AdamW optimizer is used together with the following augmentations for Tiny-ImageNet: randomized crop (scale $0.8 - 1.0$), random horizontal flip ($p = 0.5$), random erasing ($p = 0.25$) and RandAugment (number of options=2, magnitude=9). For GraNet, a stepwise LR schedule with decays at epochs 100 and 150 is used, and for TAPER, we use the proposed cosine LR decay. To assess the relative gains from importance

Table 1: Test Accuracy of ResNet50 networks trained on CIFAR100 and pruned to 98% sparsity. Baselines are reported from Liu et al. (2021), except for DST-EE, which is from the original work, and FGGP, which was reproduced by us. Bold indicates the results within std of the best.

Approach	Method	Accuracy (mean ± std)
Pruning at Initialization	SNIP (Lee et al., 2019)	58.43
	SynFlow (Tanaka et al., 2020)	62.17
Dynamic-Sparse	RigL (Evci et al., 2020)	75.06 ± 0.27
	DST-EE (Huang et al., 2023)	75.68 ± 0.11
Gradual Pruning	GMP (Zhu & Gupta, 2017)	75.58 ± 0.20
	GraNet (Liu et al., 2021)	**76.10 ± 0.20**
	FGGP (Zhu et al., 2025)	75.33 ± 0.76
	TAPER (ours)	**76.24 ± 0.45**

Table 2: Test accuracy (mean ± std) on CIFAR10 (C10) with VGG19 and CIFAR100 (C100) with VGG19 and ResNet50, trained to different sparsity levels. Bold indicates the results within std of the best score within that corresponding experimental setting.

	Pruning Method	Sparsity 99.95 %	99.90 %	99.80 %	99.50 %	99.00 %	98.00 %
C10 VGG19	TAPER	**84.64 ± 1.09**	**89.43 ± 0.19**	**91.35 ± 0.14**	**93.16 ± 0.10**	**93.59 ± 0.08**	**93.91 ± 0.13**
	GraNet	**84.58 ± 0.76**	88.97 ± 0.14	**91.26 ± 0.10**	92.59 ± 0.05	93.19 ± 0.16	93.50 ± 0.13
	FGGP	81.59 ± 0.93	87.94 ± 0.13	**91.22 ± 0.28**	92.66 ± 0.13	93.24 ± 0.12	93.67 ± 0.14
C100 VGG19	TAPER	**29.00 ± 5.05**	**55.01 ± 0.69**	**64.99 ± 0.47**	**69.67 ± 0.37**	71.38 ± 0.28	**72.30 ± 0.16**
	GraNet	**27.93 ± 3.75**	44.05 ± 0.44	**64.59 ± 0.27**	**69.88 ± 0.28**	**71.58 ± 0.10**	**72.23 ± 0.18**
	FGGP	**25.08 ± 1.32**	51.65 ± 0.49	62.00 ± 0.16	68.98 ± 0.28	70.82 ± 0.20	71.87 ± 0.31
C100 ResNet50	TAPER	**51.18 ± 0.75**	**62.44 ± 0.49**	**69.28 ± 0.39**	**73.34 ± 0.14**	**74.89 ± 0.04**	**76.24 ± 0.45**
	GraNet	34.95 ± 0.54	54.27 ± 0.64	65.67 ± 0.35	72.53 ± 0.63	74.36 ± 0.45	75.51 ± 0.74
	FGGP	38.07 ± 0.42	53.74 ± 0.57	64.97 ± 0.71	71.98 ± 0.69	74.50 ± 0.62	75.33 ± 0.76

scoring, LR schedule, and other components, we later present results from ablation experiments of these. Finally, batch-level mixup ($\alpha = 0.2$) and cutmix ($\alpha = 1.0$) augmentation were applied with a probability of 1.0 and a switching probability of 0.5 and label smoothing was used with a probability of 0.1. The strength of the augmentations was updated relative to the original ConvNeXt paper (Liu et al., 2022) to suit Tiny-ImageNet. Hyperparameters used for training are reported in the Appendix. For TAPER, Δt is set to one or several epochs to ensure equal contribution from all batches to the temporally aggregated importance score. For CIFAR10 and CIFAR100, which use VGG-19 and ResNet, 1 epoch was feasible, while for Tiny-ImageNet (with ConvNext-Tiny) 4 epochs were used. For GraNet and FGGP we used $\Delta t = 1000$ batches on CIFAR10 and CIFAR100 to conform with their original setup for these datasets and networks. For ConvNeXt, GraNet and TAPER use the same Δt. All experiments were performed with three different seeds, and results are reported as mean ± standard deviation (std). We implemented TAPER in PyTorch 2 based on the torch.nn.utils.prune package.

Experiment 1 We begin by comparing TAPER against state-of-the-art pruning methods with a 98% sparse ResNet50 trained on CIFAR100 as presented in Table 1. 98% sparse ResNet50 is one of the most commonly used baselines for pruning evaluation. As presented, TAPER performs similarly to and marginally better than GraNet, which together are superior to other compared approaches.

Experiment 2 Further experiments compare TAPER against its two closest contestants, GraNet and FGGP, for different datasets, network architectures, and at higher sparsities up to 99.95%. Since the latter methods do not have results published for all these setting combinations, we run them using their own proposed settings and otherwise with minimal changes required to fairly adapt to the dataset and architecture. On both CIFAR datasets, TAPER performs competitively or superior to the state of the art at all tested sparsity levels, as seen in Table 2 and Figure 1 (a-b). When pruning

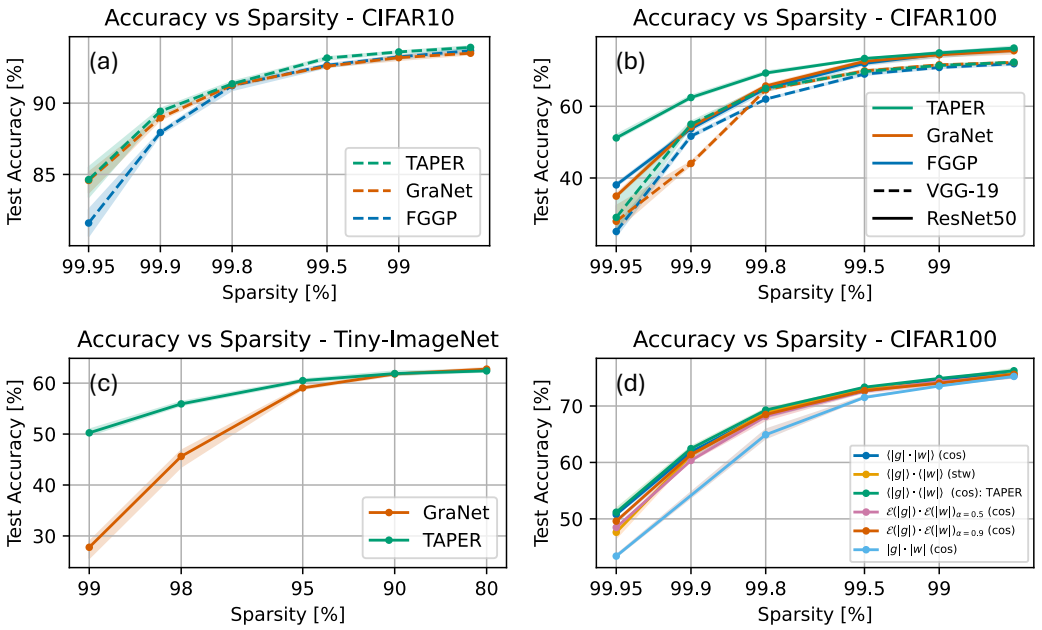

Figure 1: Accuracy of pruned models at different sparsity levels showing mean, min, and max of three seeds. TAPER is comparatively evaluated against GraNet and FGGP as state-of-the-art in gradual pruning (a) on CIFAR10 with VGG-19 and (b) on CIFAR100 with both VGG-19 and ResNet50. (c) TAPER is compared to the closest prior method, GraNet, on Tiny-ImageNet. (d) Ablation study showing importance scoring and training schedule variations, where TAPER uses $\langle|g|\rangle \cdot \langle|w|\rangle$ (cos).

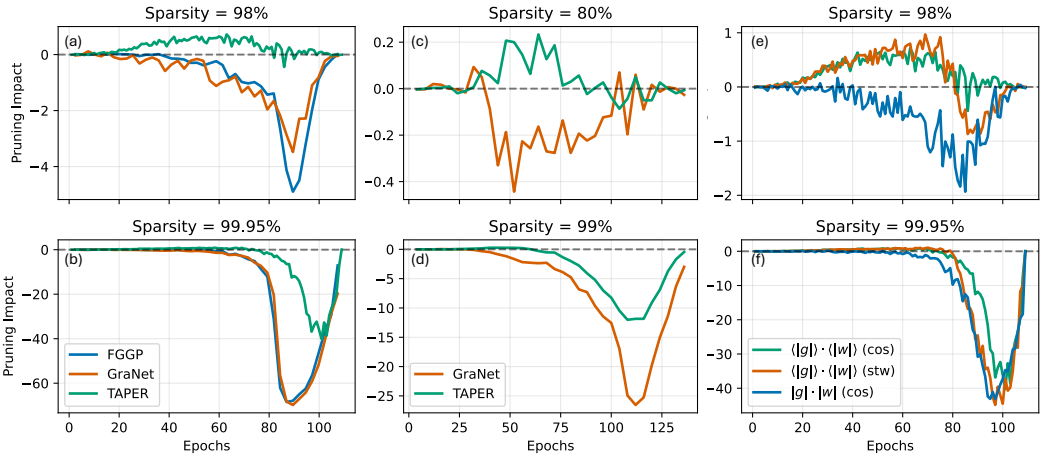

Figure 2: Pruning impact at every pruning step for runs with different aimed sparsities. The average PrIm over three seeds at every pruning step is presented for the highest and lowest tested sparsities in (a-b) for CIFAR100 and in (c-d) for Tiny-ImageNet. (e-f) show ablation results for CIFAR100.

VGG-19 to very high sparsities, GraNet and TAPER perform similarly, but when pruning ResNet50, TAPER outperforms all previous methods at sparsities of 99.8% and higher.

Further, we present the pruning impact at each pruning iteration (averaged over three seeds) in Figure 2(a-b). When investigating PrIm, TAPER is seen to degrade network performance less than FGGP and GraNet. The mean PrIm for every tested sparsity further supports this, as visualized in Figure 3 (a). Surprisingly, at relatively lower sparsity of 98%, PrIm even stays consistently positive valued over the training, indicating TAPER impact scoring finding weights the removal of which

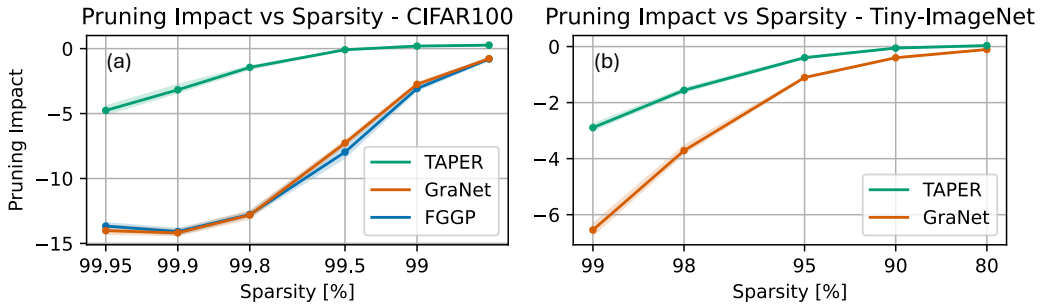

Figure 3: Averaged pruning impact of models at different sparsities. The aggregated PrIms are compared on (a) CIFAR100 with ResNet50 and (b) Tiny-ImageNet with ConvNeXt-Tiny.

Table 3: Test accuracy, reported as mean±std of ConvNeXt-Tiny at different sparsity levels, trained on the Tiny-ImageNet dataset. Sparsities are calculated as fractions of pruned and prunable parameters.

	Sparsity					Dense
Pruning Method	99 %	98 %	95 %	90 %	80 %	0 %
TAPER	**50.24 ± 0.60**	**55.92 ± 0.37**	**60.49 ± 0.28**	**61.87 ± 0.40**	62.40 ± 0.18	63.64 ± 0.52
GraNet	27.77 ± 1.84	45.62±1.78	59.07±0.30	**61.8±0.21**	**62.75±0.26**	

increases the network performance. This is an interesting direct observation that pruning may not always cause damage, and can even improve accuracy in certain cases when performed effectively.

Experiment 3: Tiny-ImageNet and ConvNeXt-Tiny provide a challenging dataset-network pair for pruning, with larger images, 200 classes, a more complex architecture, and lower dense network upperbound. For this task, TAPER is evaluated against the closest-performing baseline method, GraNet, with results presented in Table 3 and Figure 1 (c). Despite comparable results at 80% and 90% sparsity, TAPER outperforms GraNet at higher sparsities, as the performance of the latter deteriorates relatively much faster with increasing sparsity. At 99% sparsity, GraNet stays below 28% accuracy, while TAPER achieves above 50% compared to a dense upperbound of 63.64%. This accuracy difference is also well explained by the PrIm metric visualizations for Tiny-ImageNet in Figure 2(c-d) and Figure 3(b). At lower sparsity of 80%, TAPER even yields positive PrIm again, while GraNet mostly deteriorates the model performance at the time of pruning. At a high sparsity of 99%, TAPER is seen to consistently cause less than half the damage caused by GraNet pruning.

Ablation: To comparatively assess the aggregation of importance scores and different aggregation methods, an ablation study was performed. The results are presented in Table 4 and Figure 1 (d). Among the combinations evaluated, instantaneous importance scoring without aggregation ($|g|\cdot|w|$) yields substantially inferior results, highlighting the value of TAPER's aggregation approach. We have tested aggregation by exponentially moving average $\mathcal{E}(\cdot)_\alpha$ with different exponential weights α=$\{0.5,0.9\}$ that increase the importance of batches closer to the pruning decision, but this has

Table 4: Accuracy (mean±std) of ablations with ResNet50 trained on CIFAR100. $\mathcal{E}$ denotes exponentially moving average and $\langle\cdot\rangle$ denotes the mean. GraNet baseline is included for comparison.

Pruning method	LR schedule	Sparsity 99.95 %	99.90 %	99.80 %	99.50 %	99.00 %	98.00 %
$\langle\|g\|\rangle\cdot\langle\|w\|\rangle$: TAPER	cos	**51.18 ± 0.75**	**62.44 ± 0.49**	**69.28 ± 0.39**	**73.34 ± 0.14**	**74.89 ± 0.04**	**76.24 ± 0.45**
$\langle\|g\|\cdot\|w\|\rangle$	cos	50.77 ± 0.30	61.86 ± 1.00	68.67 ± 0.80	72.67 ± 0.31	74.73 ± 0.22	75.70 ± 0.79
$\mathcal{E}(\|g\|)\cdot\mathcal{E}(\|w\|)_{\alpha=0.9}$	cos	49.60 ± 0.49	61.40 ± 0.59	68.49 ± 0.42	72.73 ± 0.49	74.04 ± 0.51	75.59 ± 0.54
$\mathcal{E}(\|g\|)\cdot\mathcal{E}(\|w\|)_{\alpha=0.5}$	cos	48.48 ± 0.32	60.35 ± 0.32	68.16 ± 0.77	72.65 ± 0.30	74.31 ± 0.30	75.32 ± 0.45
$\langle\|g\|\rangle\cdot\langle\|w\|\rangle$	stw	47.62 ± 1.10	61.29 ± 0.57	68.85 ± 0.69	72.99 ± 0.58	74.16 ± 0.61	75.69 ± 0.53
$\|g\|\cdot\|w\|$	cos	43.41 ± 0.38	55.56 ± 0.82	64.91 ± 1.00	71.52 ± 0.33	73.53 ± 0.09	75.24 ± 0.53
$\|w\| \rightarrow \|g\|$: GraNet	stw	34.95 ± 0.54	54.27 ± 0.64	65.67 ± 0.35	72.53 ± 0.63	74.36 ± 0.45	75.51 ± 0.74

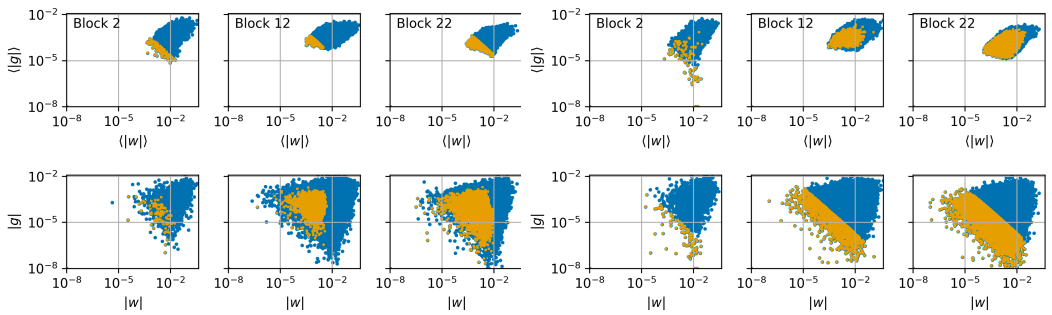

Figure 4: Comparison of parameters removed from three ResNet50 blocks (early, mid, and late in the architecture) when using different metrics. Removed parameters at epoch 20 of training on CIFAR100 are visualized in orange, while parameters remaining are visualized in blue. To the left, the TAPER importance score $\langle|g|\rangle \cdot \langle|w|\rangle$ is used and, to the right, $|g| \cdot |w|$ is used as the importance score. The top row shows aggregated gradient and weight values, while the bottom row shows the values after the last batch.

performed inferior to simple mean with equal batch-wise importance, which can be explained by smaller α approximating to the instantaneous importance scoring in the limit case. We also tested aggregating after multiplying the terms ($\langle|g|\cdot|w|\rangle$), similarly to an unsigned variant of the movement pruning score Sanh et al. (2020). This performed inferior to TAPER's aggregate-then-multiply approach at all tested sparsities. This can be explained by multiplication amplifying batch-level noise in the components if aggregation is not performed before multiplication. To assess the effect of LR schedule on the results, we include GraNet as a baseline comparator by also adding a TAPER ablation using GraNet's stepwise LR decay, *i.e.*, $\langle|g|\rangle\cdot\langle|w|\rangle$ (stw). The proposed TAPER importance score is seen to perform superior to GraNet even when using the same LR schedule, while the TAPER's cosine LR further improves the final performance.

Figure 2 (e-f) presents PrIm for selected ablations. Compared to no-aggregation, the proposed score aggregation is seen to provide less damage (@99.95%) and even improvement (@98%) in accuracy. Further, comparing the LR schedules, a large drop in PrIm is observed around at epoch 80 for stepwise (stw) LR decay, which reduces LR by a factor of 10 at epoch 80. This highlights the importance of a large enough learning rate to allow the network to recover in between pruning iterations, and the value of a smooth reduction as in the cosine scheduling.

Figure 4 provides further insight into the mechanisms of parameter aggregation, by plotting weight and gradient magnitudes for pruned and non-pruned weights at early, mid, and late network blocks. These plots demonstrate the noisy nature of weight and gradient magnitudes and how pruning based on only information from the last batch may remove weights despite their relatively large average weight and/or gradient magnitude, which in turn indicate the corresponding parameters being potentially important and/or not having converged. When using aggregated importance, the weights with currently small magnitudes or gradients are still kept if they have historically large values across the earlier batches.

5 Discussion and Conclusion

In this paper, we present TAPER as a method for gradual pruning based on first-order Taylor-expansion inspired importance scores. Our results demonstrate that first-order information can provide effective guidance for gradual pruning and, in particular, that combining gradient- and weight-based information can yield robust pruning decisions across architectures and datasets. These results suggest that both the current parameter values and their local sensitivity provide useful and complementary signals for identifying parameters to prune.

We have shown that the aggregation of importance scores is beneficial when combining gradient and weight information. This reduces the sensitivity of pruning decisions to noisy gradients, sign changes in weights, and batch-to-batch variations, thereby making the resulting importance estimates more robust. The findings highlight that the effectiveness of an importance score depends

not only on the nature of its information source, but also on how this information is functionalized. Our ablations show that aggregating the gradient and weight magnitudes separately before multiplication consistently outperforms aggregation performed after multiplication. This suggests that the benefit of aggregation is not solely attributable to smoothing the importance score. A plausible explanation for the superiority of aggregating before multiplication is that batch-level fluctuations in either component are amplified by multiplication, whereas robustly smoothing the components before multiplication produces more stable final estimates.

The ablation results demonstrate how TAPER's importance score as well as LR schedule both contribute to its final superior performance. When TAPER is evaluated with the stepwise schedule used by GraNet, it retains its performance advantage in the ablation, while the cosine schedule provides an additional improvement in final accuracy.

We further introduce Pruning Impact (PrIm) as a complementary pruning metric, for evaluating the immediate effect of a pruning operation on the network performance. This provides a more controlled basis for comparing importance scores and analyzing their contribution to pruning performance.

PrIm analysis further demonstrates that LR scheduling may substantially affect the model's ability to recover between pruning iterations, particularly following an abrupt stepwise LR decay. This highlights that while importance scoring determines which parameters are removed, the overall optimization dynamics influence how well the remaining network can recover from that removal, indicating that their inter-dependence need further studies and future pruning work should very clearly report their settings.

Although PrIm provides a more direct measure of the effect of parameter removal, it does not fully isolate the importance score: The impact depends on the network state at the time of pruning and must therefore be compared from equivalent network states. Moreover, it does not capture subsequent recovery, nor if a resulting pruned network is more or less serviceable by following training. In theory, a pruning criterion may have a small instantaneous impact while still producing a subnet that is harder to optimize.

An exciting observation is that some pruning operations produce positive PrIm, *i.e.*, yielding test accuracy increasing immediately after pruning. While this may be explained with a regularizing effect of parameter removal, the current experiments cannot distinguish this explanation from other sources of short-term variation. Still, the results indicate evidence that pruning is not always necessarily damaging the network performance and under certain conditions can provide a boost at least at that given network state.

Finally, our evaluation focuses on dense-to-sparse gradual pruning. Although the proposed importance score and pruning pipeline do not inherently require a dense starting network, we do not systematically evaluate the sparse-to-sparse setting. Extending the evaluation to sparse-to-sparse pruning and a broader range of architectures and tasks remains an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for generating drafts of python functions, finding related works and polishing the writing of the paper. All LLM-generated code that was integrated into our experimental setup has been thoroughly tested, and related works proposed by LLMs have been reviewed and verified. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI. We have not used generative AI tools for other tasks with required disclosure.

REPRODUCIBILITY STATEMENT

To ensure reproducibility of our results, a code repository will be made available upon publication.

ACKNOWLEDGMENTS

(anonymized for double-blind review)

REFERENCES

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*, volume 119 of *Proceedings of Machine Learning Research*, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, 2023.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems*, 1992.

Wenhao Hu, Paul Henderson, and José Cano. ICE-Pruning: an iterative cost-efficient pruning pipeline for deep neural networks. In *2025 International Joint Conference on Neural Networks (IJCNN)*, 2025.

Shaoyi Huang, Bowen Lei, Dongkuan Xu, Hongwu Peng, Yue Sun, Mimi Xie, and Caiwen Ding. Dynamic sparse training via balancing the exploration-exploitation trade-off. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, 2023.

Tanishq Kumar, Kevin Luo, and Mark Sellke. No free prune: Information-theoretic barriers to pruning at initialization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 2024.

Duong Hoang Le and Binh-Son Hua. Network pruning that matters: A case study on retraining variants. In *International Conference on Learning Representations*, 2021.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations*, 2017.

Gui Ling, Ziyang Wang, Yuliang Yan, and Qingwen Liu. SlimGPT: Layer-wise structured pruning for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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 *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

Shiyu Liu, Rohan Ghosh, John Chong Min Tan, and Mehul Motani. Optimizing learning rate schedules for iterative pruning of deep neural networks. *Transactions on Machine Learning Research*, 2023.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Huan Wang and Yun Fu. Trainability preserving neural pruning. In *The Eleventh International Conference on Learning Representations*, 2023.

Huan Wang, Xinyi Hu, Qiming Zhang, Yuehai Wang, Lu Yu, and Haoji Hu. Structured pruning for efficient convolutional neural networks via incremental regularization. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):775–788, 2020.

Yexu Zhou, Haibin Zhao, Michael Hefenbrock, Siyan Li, Yiran Huang, and Michael Beigl. Deep neural network pruning with progressive regularizer. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2024.

Lingkai Zhu, Can Deniz Bezek, and Orcun Goksel. Fggp: Fixed-rate gradient-first gradual pruning. In *Scandinavian Conference on Image Analysis (SCIA)*, 2025.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprints*, arXiv/1710.01878, 2017.

A APPENDIX

Training configurations for TAPER on all dataset/network pairs used are presented in Table 5. For all datasets, a batch size of 128 was used, and TAPER is used with a cosine LR decay when no decay approach is specified, *e.g.*, step-wise as "(stw)". Moreover, an SGD optimizer was used for CIFAR experiments, while AdamW was used for training on Tiny-ImageNet with $\beta_1 = 0.9, \beta_2 = 0.999$). Optimizer weight decay parameters are tabulated.

Table 5: TAPER hyperparameters for datasets and network architectures.

Dataset	Network	Learning rate range	#Epochs all	#Epochs finetuning	Pruning rate Δt [batches]	Weight decay
CIFAR10	VGG–19	$10^{-1} \rightarrow 10^{-3}$	160	50	1 epoch: 391	$5 \cdot 10^{-4}$
CIFAR100	VGG–19	$10^{-1} \rightarrow 10^{-3}$	160	50	1 epoch: 391	$5 \cdot 10^{-4}$
CIFAR100	ResNet50	$10^{-1} \rightarrow 10^{-3}$	160	50	1 epoch: 391	$5 \cdot 10^{-4}$
Tiny-ImageNet	ConvNext-Tiny	$3{\cdot}10^{-4} \rightarrow 10^{-6}$	200	60	4 epochs: 3128	$5 \cdot 10^{-2}$