Efficient Model Compression Techniques with FishLeg

Jamie McGowan[*]
MediaTek Research

Wei Sheng Lai
University College London

Weibin Chen
University College London

Henry Aldridge
University College London

Jools Clarke
University College London

Jezabel R Garcia
MediaTek Research

Rui Xia
University of Cambridge

Yilei Liang[†]
University of Cambridge

Guillaume Hennequin
MediaTek Research & University of Cambridge

Alberto Bernacchia
MediaTek Research

Abstract

In many domains, the most successful AI models tend to be the largest, indeed often too large to be handled by AI players with limited computational resources. To mitigate this, a number of compression methods have been developed, including methods that prune the network down to high sparsity whilst retaining performance. The best-performing pruning techniques are often those that use second-order curvature information (such as an estimate of the Fisher information matrix) to score the importance of each weight and to predict the optimal compensation for weight deletion. However, these methods are difficult to scale to high-dimensional parameter spaces without making heavy approximations. Here, we propose the FishLeg surgeon (FLS), a new second-order pruning method based on the Fisher-Legendre (FishLeg) optimizer. At the heart of FishLeg is a meta-learning approach to amortising the action of the *inverse* FIM, which brings a number of advantages. Firstly, the parameterisation enables the use of flexible tensor factorisation techniques to improve computational and memory efficiency without sacrificing much accuracy, alleviating challenges associated with scalability of most second-order pruning methods. Secondly, directly estimating the inverse FIM leads to less sensitivity to the amplification of stochasticity during inversion, thereby resulting in more precise estimates. Thirdly, our approach also allows for progressive assimilation of the curvature into the parameterization. In the gradual pruning regime, this results in a more efficient estimate refinement as opposed to re-estimation. We find that FishLeg achieves higher or comparable performance against two common baselines in the area, most notably in the high sparsity regime when considering a ResNet18 model on CIFAR-10 (84% accuracy at 95% sparsity vs 60% for OBS) and TinyIM (53% accuracy at 80% sparsity vs 48% for OBS).

1 Introduction

The current staggering growth of AI models is threatening to sideline small and medium-sized AI contributors with limited access to compute resources, who cannot afford to run the largest models

[*]Correspondence to `jamie.mcgowan@mtkresearch.com`
[†]Work done while at MediaTek Research.

and must therefore compromise on performance. Large models are also expensive to serve and hard to deploy on low-power devices. Consequently, there is a growing need for methods that can compress these models down to a fraction of their original size whilst retaining their performance (Liu and Wang, 2023), or indeed train models from scratch to be sparse (Liu et al., 2023b).

Some of the most promising directions for neural compression revolve around leveraging second order information in order to selectively prune least important parameters, while simultaneously updating those that remain. Several recent studies have shown that second-order parameter importance scores are more accurate than more rudimentary measures derived from weight magnitudes and/or gradients (Gale et al., 2019; Sanh et al., 2020), yielding more effective pruning in convolutional (Singh and Alistarh, 2020; Theis et al., 2018) or transformer (Kurtic et al., 2022; Kuznedelev et al., 2022a) architectures. Moreover, second-order methods have shown some promise in pruning benchmarks specifically chosen to "fail current sparse neural networks" (Liu et al., 2023a). However to obtain state-of-the-art performance, compressed models often require a period of retraining after or during the process of model compression which necessitates hand-crafted compression recipes to be designed – usually switching between compression and training phases (Kuznedelev et al., 2022b).

Despite the promise of OBS-derived approaches, they are faced with a severe tradeoff between scalability and accuracy that has proven hard to navigate. Specifically, both the importance scores and the weight updates rely on estimating the action of the inverse Hessian H^{-1} (or, in our case, the inverse Fisher matrix F^{-1}) on a high-dimensional parameter space ($\boldsymbol{v} \mapsto H^{-1}\boldsymbol{v}$), which inevitably calls for approximations. Indeed, all recent applications of the OBS framework to pruning have had to make significant simplifications, such as (i) ignoring correlations between most weights or groups of weights (Kurtic et al., 2022; Kuznedelev et al., 2022a), even those that belong to the same layer, or (ii) making low-rank approximations to the Hessian (Frantar et al., 2021; Singh and Alistarh, 2020) which are as good as the memory they consume. Moreover, in the gradual pruning regime where the model changes little from stage to stage, what has been learned about the curvature at the current stage is often unduly discarded in the next one.

In parallel to the advancements of second-order pruning techniques, *optimization* has also been the subject of many improvements that tackle similar computational challenges. In particular, the FishLeg optimizer introduced by (Garcia et al., 2023) attacks the scalability-accuracy dilemma by learning to directly amortize $F^{-1}\boldsymbol{v}$ products in an easy-to-evaluate $Q(\boldsymbol{\lambda})\boldsymbol{v}$ form. This is done by minimizing an auxiliary loss $\mathcal{A}(\boldsymbol{\lambda})$ derived from Legendre duality principles, w.r.t. a set of auxiliary parameters $\boldsymbol{\lambda}$ (details in Appendix A). In contrast to low-rank approximations of the Fisher matrix that require hundreds of gradients to be stored, FishLeg allows the progressive distillation of a large number of gradients into the auxiliary parameter set $\boldsymbol{\lambda}$. This direct and gradual learning of F^{-1} in $Q(\boldsymbol{\lambda})$ is particularly relevant to the gradual pruning setting, where other methods typically have to recompute F from scratch following pruning, and re-invert it. By means of low-parameter tensor factorization techniques, the size of $\boldsymbol{\lambda}$ can be kept within a small multiple of the size of the model itself, enabling pruning of large models with limited memory. Whilst such memory efficiency can also be attained through KFAC-based methods (Martens and Grosse, 2015; Wang et al., 2019), FishLeg's direct estimation of the *inverse* Fisher is less sensitive to gradient noise. Moreover, the form of KFAC's F^{-1} follows rigidly from approximate mathematical derivations, whereas FishLeg's $Q(\boldsymbol{\lambda})$ can be any user-specified positive-definite quadratic form, yielding greater flexibility and accuracy. We use this flexibility to develop a novel variation on the well-known Kronecker-factored curvature approximation for dense layers, as well as new approximations for the convolutional layer.

In this work, we introduce the FishLeg Surgeon (FLS) — a novel pruning algorithm that exploits the inverse curvature estimation machinery of the Fisher-Legendre (FishLeg) optimizer (Garcia et al., 2023). We build on the Optimal Brain Surgeon (OBS; Hassibi and Stork, 1992; LeCun et al., 1989), a classical approach to pruning that approximates the network's loss function in quadratic form to determine (i) the importance (or saliency) of each weight and (ii) the optimal way of compensating for their deletion.

Our contributions are:

- We provide the first example of using a second-order optimizer for unstructured and semi-structured pruning – allowing for the Fisher matrix to be updated online during pruning/training.
- We modify the auxiliary loss $\mathcal{A}(\boldsymbol{\lambda})$ to facilitate assessment of its convergence and to promote learning of the full F^{-1} as required for pruning (as opposed to learning the action of F^{-1}

on the subspace of momentary noisy gradients, as relevant to the optimization setting of Garcia et al., 2023).

- We propose a new preconditioner for this (often ill-conditioned) auxiliary loss, and show analytically that it accelerates convergence asymptotically.
- We propose a new initialization scheme for $Q(\boldsymbol{\lambda})$ that leads to better estimation of F^{-1} especially when it is ill-conditioned.
- We evaluate our proposed method on CIFAR-10 Krizhevsky et al., 2009 and TinyIM Le and Yang, 2015 datasets with a ResNet18 model in the unstructured and N:M semi-structured pruning regimes and compare these against readily available baselines.

2 Neural Compression with FishLeg

Algorithm 1 FishLeg Surgeon (FLS)

```
1: **Goal**: gradual pruning to $f_{\text{end}}\%$ sparsity.
2: Choose hyperparameters: damping factor $\gamma$, learning rate $\eta$, Adam parameters, sparsity schedule $\{f_t\}$.
3: Pretrain $Q(\boldsymbol{\lambda}_0)$.    ▷ starts with a good estimate of the inverse FIM
4: $t \leftarrow 0$, $\boldsymbol{w}_0 \leftarrow \boldsymbol{w}^*$
5: while not finished do
6:     ▷ Pruning step ◁
7:     Select and prune the $(f_{t+1}-f_t)\%$ least important weights using $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t^2/\text{diag}(F_\gamma^{-1}(\boldsymbol{w}_t))$ and the latest approximation $F_\gamma^{-1}(\boldsymbol{w}_t) \approx Q(\boldsymbol{\lambda}_t)$, where $\boldsymbol{w}_t$ is the masked parameters from the previous sparsity level.
8:     for $s = 1 : S$ do
9:         $\mathcal{L}, \boldsymbol{g} \leftarrow$ value and gradient of loss evaluated at the masked $\tilde{\boldsymbol{w}}_s$ on a data minibatch
10:        ▷ Fine-tuning ◁
11:        $\tilde{\boldsymbol{w}}_{s+1} \leftarrow \tilde{\boldsymbol{w}}_s - \eta[Q(\tilde{\boldsymbol{\lambda}}_s)\boldsymbol{g}]$    ▷ masked update that preserves current sparsity
12:        ▷ Update the inverse FIM approximation, taking into account the new parameters ◁
13:        Perform one step of auxiliary loss minimization, yielding a new $\tilde{\boldsymbol{\lambda}}_{s+1}$.
14:    ▷ resume pruning with the fine-tuned parameters and updated inverse FIM estimate ◁
15:    $\boldsymbol{w}_{t+1} \leftarrow \tilde{\boldsymbol{w}}_S, \boldsymbol{\lambda}_{t+1} \leftarrow \tilde{\boldsymbol{\lambda}}_S, t \leftarrow t+1$
```

Under the standard assumption that the gradient at the current point $\mathbf{w}$ is negligible for a pretrained model, the OBS formulas for the optimal weight to be pruned w_p and the corresponding update δ_p can be derived by writing the locally quadratic problem under the constraint that element p of δ_p is equal to $-w_p$, which means that w_p is zero after applying the update to $\mathbf{w}$. This problem has the following Lagrangian:

$$L(\delta_p, \lambda) = \delta_p^\top \mathbf{H}\delta_p + \lambda(\mathbf{e}_p^\top \delta_p - (-w_p)), \quad (6) \tag{1}$$

where $\mathbf{H}$ denotes the Hessian at $\mathbf{w}$ and $\mathbf{e}_p$ is the p-th canonical basis vector.

The inverse FIM used to score and update the weights is typically recomputed at regular intervals in current second order pruning methods. The reason for this being that OBS derived methods are based around inverting an estimation of the empirical Fisher. However during pruning, as parameters are removed and others updated, the inverse FIM estimation becomes increasingly inaccurate, necessitating a complete re-estimation of the empirical Fisher before its explicit inversion.[†]

Here, we reason that FishLeg's parametric estimation of the inverse FIM, $Q(\boldsymbol{\lambda})$, can be actively updated in a rolling fashion between consecutive pruning steps by simply performing a certain number of auxiliary loss minimization steps. Crucially, by amortizing the re-computation of the inverse FIM in this way, we can afford to update our model directly (for which we use the FishLeg optimizer, also based on the running estimate $Q(\boldsymbol{\lambda})$), as outlined in Algorithm 1. Hence, unlike previous approaches to gradual second-order pruning, we need not re-estimate and re-invert the Fisher matrix from scratch after each pruning step – we simply refine our current estimate.

[†]Indeed, much of the advancements of OBS methods revolve around efficient inversion techniques.

2.1 Memory efficient parameterization of the inverse Fisher approximation

For scalability, we approximate F^{-1} in block-diagonal form, with each layer contributing one block. Note that these blocks are orders of magnitude larger than the ones used in previous second-order approaches that implemented direct inversion (e.g. Kurtic et al., 2022 used blocks of size 50).

Our choice of structure for $Q(\boldsymbol{\lambda})$ is slightly more constrained by our pruning objective than it is for the FishLeg optimizer: we require efficient evaluation of not only $Q\boldsymbol{v}$ products (which essentially sketch the curvature in the direction of steepest descent), but also $\text{diag}(Q)$ (required in Equation 28). For dense layers with n_i inputs and n_o outputs, we parameterize the corresponding inverse Fisher block as

$$Q(\boldsymbol{\lambda}) \triangleq D(LL^\top \otimes RR^\top)D \tag{2}$$

where $L \in \mathbb{R}^{n_\text{o}\times n_\text{o}}$ and $R \in \mathbb{R}^{n_\text{i}\times n_\text{i}}$ are two parameter matrices, D is a diagonal matrix with $(n_\text{i}+1)n_\text{o}$ parameters, and $\otimes$ denotes the Kronecker product. This construction is such that, for $V \in \mathbb{R}^{n_\text{o}\times n_\text{i}}$,

$$Q(\boldsymbol{\lambda})\text{vec}(V) = D \odot \text{vec}(LL^\top(V \odot \bar{D})RR^\top) \tag{3}$$

with the (unusual) convention that $\text{vec}(\cdot)$ vectorizes row-wise (corresponding to a no-copy reshape in numerical code), and $\odot$ denotes elementwise (Hadamard) product. Here, $\bar{D} \in \mathbb{R}^{n_\text{o}\times(n_\text{i}+1)}$ is the un-vectorized version of the diagonal of D. Similarly,

$$\text{diag}(Q) = \text{diag}(D)^2 \odot (\text{diag}(LL^\top) \otimes \text{diag}(RR^\top)) \tag{4}$$

can be evaluated efficiently, with $\text{diag}(LL^\top) = (L \odot L)(1,\dots,1)^\top$. Note that the inclusion of D makes it more expressive than the standard KFAC approximation which is limited to the Kronecker product. For completeness in Appendix F, we compare the above parameterisation with a pure diagonal parameterisation and also a more restrictive block diagonal structure similar to other second-order pruning methods (i.e. oBERT & M-FAC).

For convolutional layers, we follow a similar tensor factorization strategy. Filter parameters are tensors of dimensions n_o(output channels) $\times$ n_i(input channels) $\times$ K(kernel size). Whilst we could parameterize the inverse Fisher block as a 3-way Kronecker product, Grosse and Martens (2016)'s KFAC derivation for convolutional layers suggests combining together the input and kernel-size dimensions. We therefore use the same structure as in Equation 2, but with R of size $n_\text{i}K$ and D of size $n_\text{o}n_\text{i}K$.

2.2 Initialization of Q

Our experiments with FishLeg have revealed that the minimization of the auxiliary loss is very sensitive to initialization – to the point that getting it wrong can yield useless estimates of F_γ^{-1}. In the context of neural network optimization, Garcia et al. (2023) advocated an identity initialization $Q_0 = \alpha I$. To choose the value of α, they observed that this identity initialization implied that the FishLeg update $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta Q(\boldsymbol{\lambda})\nabla_{\boldsymbol{w}}\mathcal{L}$ would initially correspond to SGD. Thus, given a learning rate η_SGD known to work well for SGD, they set $\alpha \triangleq \eta_\text{SGD}/\eta$. However, in the context of pruning this rationale no longer applies; we therefore revisited the choice of α.

We found that good pruning results could only be obtained for sufficiently large α. To understand this, we studied the idealized dynamics of auxiliary loss gradient descent (Figure 1; see also Appendix C). Let $F = U\Xi U^\top$ be the eigendecomposition of the Fisher matrix, with $\Xi = \text{diag}(\xi_1,\dots,\xi_n)$. Assuming $\boldsymbol{u} \sim \mathcal{N}(0, I_n)$, the auxiliary loss (Equation 9) reduces to $A(\boldsymbol{\lambda}) = \frac{1}{2}\text{Tr}(QF_\gamma Q) - \text{Tr}(Q))$. Expressing Q in the eigenbasis of F as $Q = U\beta U^\top$, the gradient flow for this deterministic loss function takes the form $\dot{\beta} = -(\Xi + \gamma I)\beta + I$ with $\beta(0) = \alpha I$. It is then easy to see that β will remain diagonal throughout, and that the i^th eigenvalue of Q has the following dynamics:

$$\underbrace{(\xi_i+\gamma)^{-1}}_{\text{time constant}}\frac{d\beta_i}{dt} = -\beta_i + \underbrace{(\xi_i+\gamma)^{-1}}_{\text{optimal steady state}} \quad \text{with} \quad \beta_i(0) = \alpha. \tag{5}$$

Thus, the eigenvalues of Q – all initially equal to α – converge at very different speeds depending on their optimal steady states: eigenvalues that must reach large (resp. small) values evolve slowly (resp. fast). We therefore conclude that a good initialization is to set α to be as large as the largest eigenvalues of F_γ^{-1}, namely $(\min\{\xi_i\}+\gamma)^{-1} \approx \gamma^{-1}$. This way, the eigenvalues of Q that would

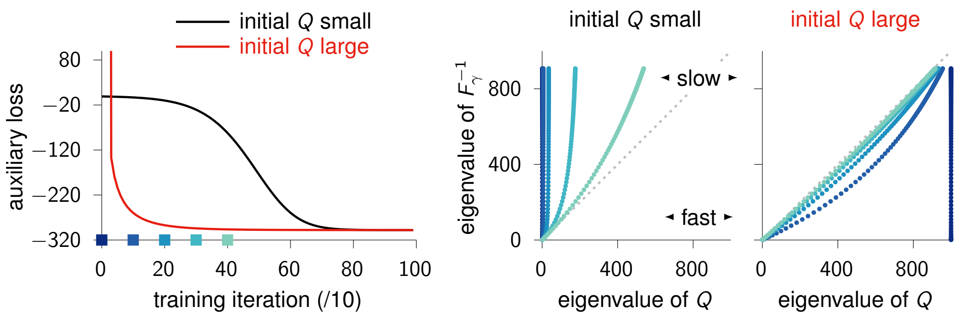

Figure 1: **The initialization of $Q(\boldsymbol{\lambda})$ matters much.** In this toy experiment, the true Fisher matrix ($n = 100$) was chosen so that its i^{th} eigenvalue is $\xi_i \triangleq 1/i^2$, and the damping parameter γ was set to 10^{-3}. Thus, the eigenvalues of F_γ^{-1} lie roughly in the $[1 - 1000]$ range. The auxiliary loss $\mathcal{A}(Q) = \frac{1}{2}\text{Tr}(QFQ) - \text{Tr}(Q)$ (left) was minimized by gradient descent w.r.t. the Cholesky factor of $Q(\boldsymbol{\lambda})$, initialized such that $Q(\boldsymbol{\lambda}) = I$ (black) or $Q(\boldsymbol{\lambda}) = \gamma^{-1}I = 1000 \times I$ (red). The learning rate was optimized separately for each case. This simulation shows that it is clearly better to initialize Q to be large rather than small. Indeed, a simple derivation shows that each eigenvalue β_i of Q approaches its target $1/(\xi_i + \gamma)$ at a speed proportional to $(\xi_i + \gamma)$ (Equation 5). In other words, the eigenvalues of Q that must end up large are also those that evolve the slowest. It, therefore makes sense to initialize them to be large so they have less to travel; the eigenvalues that must end up small will become small rapidly anyway. The right panels illustrate this behaviour by plotting the eigenvalues of Q against their respective targets, at regular intervals during optimization (color-coded), for both initialization schemes. The auxiliary loss is minimized when $\beta_i = 1/(\xi_i + \gamma)$, i.e. when the dots lie along the identity line (dashed grey).

normally slowly evolve towards γ^{-1} are positioned there from the outset, and the eigenvalues that are set to decrease do so rapidly. Figure 1 illustrates this behaviour and shows that large initialization of Q (with $\alpha \approx \gamma^{-1}$) results in faster minimization of the auxiliary loss.

2.3 Preconditioning of the auxiliary loss

Learning the full F^{-1} is a hard problem when F is ill-conditioned, as the auxiliary loss inherits this ill-conditioning. Our theoretical analysis of this problem (Appendix C) has led to the discovery of a good preconditioner which only costs a single additional $Q(\boldsymbol{\lambda})v$ product per iteration. This preconditioner greatly accelerates asymptotic convergence of the auxiliary loss (Figure 5A), leading to better estimates of the inverse FIM.

3 Empirical Investigations

In order to validate our approach for pruning and training in the same breath using a second order optimizer such as FishLeg, we provide initial studies with a ResNet18 model with a single layer classification layer on CIFAR-10 and TinyIM. We consider two pruning approaches that are common in the field: *unstructured* and *semi-structured* pruning. Across all experiments we consider the same dense models trained with Adam to 83.4% test accuracy on CIFAR-10 and 55.4% test accuracy on TinyIM. Results are reported as mean over 3 random seeds. All experiments were run on a single NVIDIA GeForce RTX 2080 GPU with 8GB of VRAM.

In addition to the results presented in this section, in Appendix F we provide extensive ablation studies for the methods introduced in Section 2 – including a direct comparison between the block diagonal approximation used in OBS methods across multiple block sizes.

3.1 Unstructured Pruning

Unstructured pruning consists of choosing parameters across the entire network to prune and achieve some non-uniformly sparse model which maintains performance on a task. In these experiments, the non-zero parameters were fine-tuned for 1 epoch (with SGDm for magnitude and OBS) after each pruning step following a exponential sparsity schedule. Figure 2 displays a marked improvement

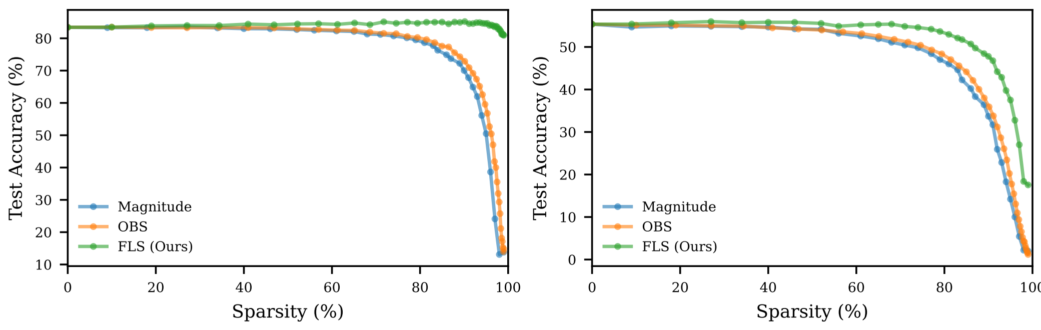

Figure 2: Test accuracy as a function of model sparsity for ResNet18 on CIFAR-10 (left) and TinyIM (right). Different pruning frameworks are used, which are magnitude pruning (blue), OBS (orange) and FLS (green).

over magnitude and OBS pruning methods across both CIFAR-10 and TinyIM experiments – with only a reduction of $\sim -2.5\%$ from dense performance at 99% sparsity on CIFAR-10. In addition to this, FLS benefits from being able to continually adapt its second-order information between pruning steps. By amortizing the overhead in recomputing the empirical FIM we afford efficient second-order updates during the finetuning phases – which, combined with some potential effect from Occam's razor (Blumer et al., 1987), explains the slight increase in performance across the experiment.

3.2 Semi-Structured Pruning

Semi-structured (N:M) pruning involves pruning N parameters in each block of M parameters. This repeated pattern can be readily applied to Sparse Tensor Cores which perform calculations on a compressed version of a sparse matrix. Here we consider the setting of 2:4 sparsity which results in a 50% sparse network. Similar to the unstructured setting, all models were pruned with an exponential schedule up to a 2:4 pattern and retrained for 1 epoch after each pruning step. Our method achieves greater or similar performance when compared to the baselines, and even improves upon the dense model performance in the case of CIFAR-10.

Method	Test Accuracy (%)	
	CIFAR-10	TinyIM
Magnitude	80.28	47.53
OBS	81.50	**52.04**
FLS (Ours)	**84.03**	51.09

Figure 3: 2:4 semi-structured pruning performance of ResNet18 model finetuned on CIFAR-10 and TinyIM data.

4 Discussion, limitations and future work

We have introduced a new perspective on second-order pruning that blurs the lines between second-order optimization and compression. We have identified challenges with the naive approach of using an “optimization" sketch of the FIM for compression and addressed these with expanded parameterisations, which we have justified theoretically and thorough ablation studies. In addition, we have provided empirical evidence which demonstrates our method on ResNet18 with CIFAR-10 and TinyIM.

Despite this, pruning with FishLeg has several limitations. One of the key assumptions in our approach is that the inverse Fisher $F_\gamma^{-1}(\boldsymbol{w}^\star)$ can be well approximated by a specific form of positive definite matrix $Q(\boldsymbol{\lambda})$; however, the structure chosen for Q is largely dictated by scalability requirements, and may not be appropriate under certain conditions. We have proposed memory-efficient factorizations of Q which we have found effective for dense and convolutional layers, and we leave the development of other types of neural network layers to future research. Indeed, it remains to be seen whether these results scale to larger models, other compression techniques (i.e., fully structured, quantisation, a mixed setting etc.) and to a wider variety of layers. We hope that further research in these areas will likely extend and refine the capabilities of the proposed method.

References

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1987). Occam's razor. *Information processing letters*, 24(6):377–380.

Frantar, E., Kurtic, E., and Alistarh, D. (2021). M-FAC: Efficient matrix-free approximations of second-order information.

Gale, T., Elsen, E., and Hooker, S. (2019). The state of sparsity in deep neural networks. *CoRR*, abs/1902.09574.

Garcia, J. R., Freddi, F., Fotiadis, S., Li, M., Vakili, S., Bernacchia, A., and Hennequin, G. (2023). Fisher-Legendre (FishLeg) optimization of deep neural networks. In *The Eleventh International Conference on Learning Representations*.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. (2018). Fast approximate natural gradient descent in a Kronecker factored eigenbasis. *Advances in Neural Information Processing Systems*, 31.

Grosse, R. and Martens, J. (2016). A Kronecker-factored approximate Fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582. PMLR.

Hassibi, B. and Stork, D. (1992). Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., and Alistarh, D. (2022). The optimal BERT surgeon: Scalable and accurate second-order pruning for large language models.

Kuznedelev, D., Kurtic, E., Frantar, E., and Alistarh, D. (2022a). oViT: An accurate second-order pruning framework for vision transformers.

Kuznedelev, D., Kurtic, E., Frantar, E., and Alistarh, D. (2022b). oViT: An Accurate Second-Order Pruning Framework for Vision Transformers. arXiv:2210.09223 [cs].

Le, Y. and Yang, X. (2015). Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3.

LeCun, Y., Denker, J., and Solla, S. (1989). Optimal brain damage. *Advances in neural information processing systems*, 2.

Liu, S., Chen, T., Zhang, Z., Chen, X., Huang, T., JAISWAL, A. K., and Wang, Z. (2023a). Sparsity may cry: Let us fail (current) sparse neural networks together! In *The Eleventh International Conference on Learning Representations*.

Liu, S. and Wang, Z. (2023). Ten Lessons We Have Learned in the New "Sparseland": A Short Handbook for Sparse Neural Network Researchers. arXiv:2302.02596 [cs].

Liu, Z., Gan, E., and Tegmark, M. (2023b). Seeing is believing: Brain-inspired modular training for mechanistic interpretability.

Martens, J. and Grosse, R. (2015). Optimizing neural networks with Kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR.

Sanh, V., Wolf, T., and Rush, A. (2020). Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389.

Singh, S. P. and Alistarh, D. (2020). WoodFisher: Efficient second-order approximation for neural network compression.

Theis, L., Korshunova, I., Tejani, A., and Huszár, F. (2018). Faster gaze prediction with dense networks and fisher pruning.

Van Loan, C. F. and Pitsianis, N. (1993). *Approximation with Kronecker products*. Springer.

Wang, C., Grosse, R., Fidler, S., and Zhang, G. (2019). EigenDamage: Structured pruning in the Kronecker-factored eigenbasis.

A Fisher-Legendre (FishLeg) Optimizer

FishLeg (Garcia et al., 2023) is a scalable second-order optimizer that approximates the natural gradient $F^{-1}\nabla_{\boldsymbol{w}}\ell(\boldsymbol{w},\mathcal{D})$ based on the following insights. Let $\boldsymbol{w}$ be a fixed set of model parameters. Consider the regularized cross entropy between $p(\mathcal{D}|\boldsymbol{w})$ and $p(\mathcal{D}|\boldsymbol{w}+\boldsymbol{\delta})$,

$$\mathcal{H}_\gamma(\boldsymbol{\delta}) = \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D}|\boldsymbol{w})}\ell(\boldsymbol{w}+\boldsymbol{\delta},\mathcal{D}) + \frac{\gamma}{2}\|\boldsymbol{\delta}\|^2, \tag{6}$$

where $\gamma > 0$ is a small damping parameter. The Legendre-Fenchel conjugate of $\mathcal{H}_\gamma(\boldsymbol{\delta})$ is defined as

$$\mathcal{H}^\star_\gamma(\boldsymbol{u}) \triangleq \min_{\boldsymbol{\delta}} \mathcal{H}_\gamma(\boldsymbol{\delta}) - \boldsymbol{u}^\top\boldsymbol{\delta} \quad \text{with minimizer denoted by } \tilde{\boldsymbol{\delta}}_\gamma(\boldsymbol{u}). \tag{7}$$

Garcia et al. were able to prove that, if the negative log-likelihood $\ell(\boldsymbol{w},\mathcal{D}) = -\log p(\mathcal{D}|\boldsymbol{w})$ is twice differentiable, then the inverse damped Fisher information matrix exists and is equal to

$$F_\gamma^{-1} \triangleq [F+\gamma I]^{-1} = \nabla_{\boldsymbol{u}}\tilde{\boldsymbol{\delta}}_\gamma(\boldsymbol{0}). \tag{8}$$

FishLeg meta-learns a parametric approximation $\overline{\boldsymbol{\delta}}(\boldsymbol{u},\boldsymbol{\lambda})$ of $\tilde{\boldsymbol{\delta}}_\gamma(\boldsymbol{u})$, by minimizing the auxiliary loss $\mathcal{A}(\boldsymbol{\lambda},\boldsymbol{u}) \triangleq \mathcal{H}_\gamma(\overline{\boldsymbol{\delta}}(\boldsymbol{u},\boldsymbol{\lambda})) - \boldsymbol{u}^\top\overline{\boldsymbol{\delta}}(\boldsymbol{u},\boldsymbol{\lambda})$ w.r.t. meta-parameters $\boldsymbol{\lambda}$, as prescribed by Equation 7. Importantly, Equation 8 shows that one only needs to learn the *local* behaviour of the vector field $\tilde{\boldsymbol{\delta}}_\gamma(\boldsymbol{u})$ around small $\boldsymbol{u}$; thus, Garcia et al. directly parameterized its (symmetric, positive definite) Jacobian $Q(\boldsymbol{\lambda})$ at $\boldsymbol{u}=\boldsymbol{0}$, corresponding to the choice $\overline{\boldsymbol{\delta}}(\boldsymbol{u},\boldsymbol{\lambda}) \triangleq Q(\boldsymbol{\lambda})\boldsymbol{u}$. Furthermore, considering the limit of small $\boldsymbol{u}$ and averaging over a relevant distribution (more on this below and in Appendix B), the auxiliary loss becomes

$$\mathcal{A}(\boldsymbol{\lambda}) \triangleq \mathbb{E}_{\boldsymbol{u}}\left\{\frac{1}{\|\boldsymbol{u}\|^2}\left[\frac{1}{2}\boldsymbol{u}^\top Q(\boldsymbol{\lambda})F_\gamma Q(\boldsymbol{\lambda})\boldsymbol{u} - \boldsymbol{u}^\top Q(\boldsymbol{\lambda})\boldsymbol{u}\right]\right\} \tag{9}$$

which can be estimated and differentiated efficiently in a number of ways (details in Section 2).

Practical note: as $Q(\boldsymbol{\lambda})$ converges towards F_γ^{-1}, the auxiliary loss as defined by Equation 9 converges towards $\langle -\boldsymbol{u}^\top F_\gamma^{-1}\boldsymbol{u}/\|\boldsymbol{u}\|^2\rangle$, which is problem-dependent; this makes it hard to assess the quality of our inverse Fisher estimation. We therefore assess convergence by computing a slightly modified auxiliary loss where we drop the $\frac{1}{2}$ factor; this should converge to zero.

Taking the gradient of Equation 9 w.r.t. $\boldsymbol{\lambda}$ makes is clear that $Q(\boldsymbol{\lambda})$ will learn to approximate the action of F_γ^{-1} on the subspace spanned by the $\boldsymbol{u}$'s. Given their application to natural gradient optimization, Garcia et al. took those $\boldsymbol{u}$'s to be stochastic gradients of the model's primary loss function. For our pruning purposes, however, Equation 27 suggests that we must accurately estimate the action of F on the entire parameter space; we will therefore work with a more isotropic distribution of $\boldsymbol{u}$ (Section 2).

Directly estimating the inverse Fisher matrix, and doing so in this way, brings a number of advantages. First, the FishLeg approach is flexible: one can specify any form of $Q(\boldsymbol{\lambda})$, and in particular combine structured approximations obtained through mathematical derivations (as in e.g. KFAC; George et al., 2018; Grosse and Martens, 2016; Martens and Grosse, 2015) with a variety of parametric adjustments for greater expressiveness. We give examples of such choices in Section 2.1. Second, the FishLeg approach is less biased than KFAC and related methods. These methods start by assuming that F has a certain structure (e.g. block diagonal), obtain a good approximation of F conforming to this structure, and then invert it. One expects both systematic errors as well as stochasticity in the estimate of F to propagate to F^{-1}. In contrast, FishLeg 'fits' a parametric approximation to F^{-1} directly, conveniently avoiding inversion. Relatedly, a key property of Equation 9 is that it is *not* biased by stochasticity in the estimate of F_γ (Appendix D; Figure 4) – unlike other seemingly sensible auxiliary loss functions such as $\mathbb{E}_{\boldsymbol{u}}\|Q(\boldsymbol{\lambda})\hat{F}_\gamma\boldsymbol{u}-\boldsymbol{u}\|^2$ or $\mathbb{E}_{\boldsymbol{u}}\|\hat{F}_\gamma Q(\boldsymbol{\lambda})\boldsymbol{u}-\boldsymbol{u}\|^2$ whose quadratic terms in $\hat{F}_\gamma$ do survive averaging.

B Auxiliary loss derivation

Starting from the auxiliary loss definition given in Equation 9 and in Equation 15 of Garcia et al. (2023), we can expand the first term with a Taylor Expansion as:

$$\mathcal{H}_\gamma(\overline{\boldsymbol{\delta}}(\boldsymbol{u},\boldsymbol{\lambda})) = \mathcal{H}_\gamma(\boldsymbol{0}) + \nabla_{\overline{\boldsymbol{\delta}}}\mathcal{H}_\gamma(\boldsymbol{\theta},\overline{\boldsymbol{\delta}})|_{\overline{\boldsymbol{\delta}}=\boldsymbol{0}}\overline{\boldsymbol{\delta}} + \frac{1}{2}\overline{\boldsymbol{\delta}}^\top\nabla^2_{\overline{\boldsymbol{\delta}}}\mathcal{H}_\gamma(\boldsymbol{\theta},\overline{\boldsymbol{\delta}})|_{\overline{\boldsymbol{\delta}}=\boldsymbol{0}}\overline{\boldsymbol{\delta}}. \tag{10}$$

As stated in Appendix A.2 of Garcia et al. (2023), each term in this Taylor expansion can be expressed as:

$$\nabla_{\boldsymbol{\delta}} \mathcal{H}_\gamma(\boldsymbol{\theta}, \boldsymbol{\delta})|_{\boldsymbol{\delta}=\mathbf{0}} = \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D}|\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathcal{D}) + \mathbf{0} = \mathbf{0} \tag{11}$$

$$\nabla^2_{\boldsymbol{\delta}} \mathcal{H}_\gamma(\boldsymbol{\theta}, \boldsymbol{\delta})|_{\boldsymbol{\delta}=\mathbf{0}} = \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D}|\boldsymbol{\theta})} \nabla^2_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathcal{D}) + \gamma I = \mathcal{I}(\boldsymbol{\theta}) + \gamma I = F_\gamma. \tag{12}$$

where the 0th order term follows from the fact that we define the minimum at $\boldsymbol{\delta} = \mathbf{0}$, the 1st order term is zero since we are at a minimum and the 2nd order term characterizes the Fisher information matrix.

Using the above definitions, one can arrive at,

$$\mathcal{A}(\boldsymbol{\lambda}, \boldsymbol{u}) = \frac{1}{2} \overline{\boldsymbol{\delta}}(\boldsymbol{u}, \boldsymbol{\lambda})^\top F_\gamma \overline{\boldsymbol{\delta}}(\boldsymbol{u}, \boldsymbol{\lambda}) - \boldsymbol{u}^\top \overline{\boldsymbol{\delta}}(\boldsymbol{u}, \boldsymbol{\lambda}) \tag{13}$$

where the second term in Equation 9 is unchanged.

C Analysis of FishLeg's auxiliary loss & preconditioning

In this section, we analyze the minimization dynamics of a generalized version of FishLeg's auxiliary loss:

$$\mathcal{A}(Q) = \langle \frac{1}{2} \boldsymbol{u}^\top Q^\top P F_\gamma Q \boldsymbol{u} - \boldsymbol{u}^\top Q^\top P \boldsymbol{u} \rangle_{\boldsymbol{u}\sim\mathcal{N}(0,I)} \tag{14}$$

where F is the model's (damped) Fisher information matrix, P is a symmetric positive definite matrix, and Q is our approximation of F_γ^{-1}. For simplicity, we will assume that the parameterization of Q is non-limiting, i.e. we will consider the minimization of $\mathcal{A}$ directly as a function of Q.

This loss can be evaluated analytically:

$$\mathcal{A}(Q) = \left\langle \text{Tr}\left(\frac{1}{2} Q^\top P F_\gamma Q \boldsymbol{u}\boldsymbol{u}^\top - Q^\top P \boldsymbol{u}\boldsymbol{u}^\top \right) \right\rangle_{\boldsymbol{u}\sim\mathcal{N}(0,I)} \tag{15}$$

$$= \text{Tr}\left[\left(\frac{1}{2} Q^\top P F_\gamma Q - Q^\top P \right) \left\langle \boldsymbol{u}\boldsymbol{u}^\top \right\rangle_{\boldsymbol{u}\sim\mathcal{N}(0,I)} \right] \tag{16}$$

$$= \text{Tr}\left(\frac{1}{2} Q^\top P F Q - Q^\top P \right) \tag{17}$$

The optimal $Q^\star$ must satisfy

$$0 = \left.\frac{\partial \mathcal{A}}{\partial Q}\right|_{Q=Q^\star} = P(FQ^\star - I) \tag{18}$$

Therefore, if P and F_γ are both invertible, then $Q^\star = F_\gamma^{-1}$ as desired. To understand how quickly Q will converge to this solution, it is useful to analyze the gradient flow

$$\frac{dQ}{dt} = -P(F_\gamma Q(t) - I) \tag{19}$$

with initial condition $Q(0) = \alpha I$. Let $F = U \Lambda U^\top$ be the eigendecomposition of the Fisher matrix, with $\Lambda = \text{diagm}(\lambda_1, \ldots, \lambda_n)$ and $U^\top U = U U^\top = I$. We will assume that P has the same eigenvectors as F, i.e. $P = U \text{diagm}(p_1, \ldots, p_n) U^\top$. Rewriting the above gradient flow in the eigenbasis of F, we obtain

$$\frac{d}{dt}(U^\top Q(t) U) = -U^\top P(F_\gamma Q - I)U \tag{20}$$

$$= -U^\top U \text{diagm}(p_1, \ldots, p_n) U^\top (U(\Lambda + \gamma I)U^\top Q - I)U \tag{21}$$

$$= -\text{diagm}(p_1, \ldots, p_n)((\Lambda + \gamma I)U^\top Q U - I) \tag{22}$$

We see that if $U^\top Q U$ is diagonal at time t, it will remain diagonal. Given that $U^\top Q U = U^\top(\alpha I)U = \alpha I$ is diagonal, we conclude that at any time t, $U^\top Q(t) U = \text{diagm}(\beta_1(t), \ldots, \beta_n(t))$. Thus, Equation 22 boils down to a set of n decoupled, scalar flows,

$$\frac{d\beta_i}{dt} = -p_i \left[(\lambda_i + \gamma)\beta_i - 1 \right] \quad \text{with } \beta_i(0) = \alpha \tag{23}$$

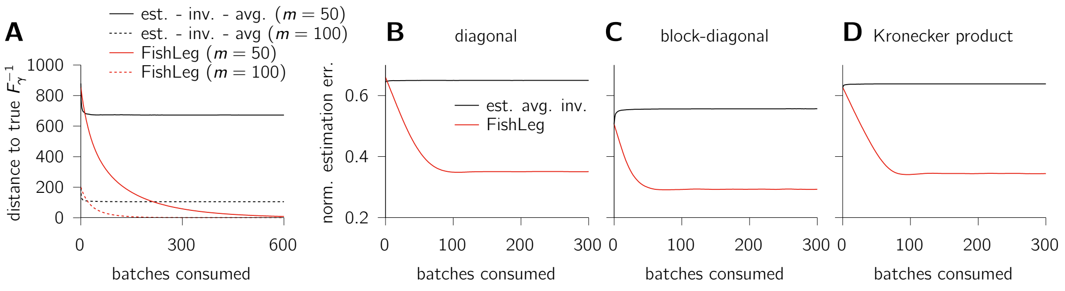

Figure 4: **Assessing FishLeg's inverse curvature estimation in a controlled setting**. In this figure, the true Fisher matrix $F \in \mathbb{R}^{100\times100}$ is constructed to have a random orthonormal eigenbasis and eigenvalues $\lambda_i \propto e^{-i/30}$. All results are averaged over 20 independent realizations of the corresponding experiment with different random seeds. **(A)**: standard affine-invariant Riemannian distance between FishLeg's Q and F_γ^{-1} ($\gamma = 0.01$), as a function of the number of data mini-batches of size m consumed so far. Each Adam step of auxiliary loss minimization consumes one minibatch. In this case, we use a full parameterization $Q = LL^\top$ that contains the solution F_γ^{-1}; in that case, FishLeg's inverse curvature estimation is consistent and the error goes to zero. As a baseline, we show the behaviour of a simple but biased estimator that estimates F_γ on each new minibatch, inverts that noisy estimate, and averages the result over minibatches; inverting noisy estimates yields a bias that persists asymptotically. **(B-D)**: In these panels, the inverse Fisher is estimated in structured form (B: diagonal; C: block-diagonal, 5 blocks; D: Kronecker product, $(5 \times 5) \otimes (20 \times 20)$. This is done either by FishLeg assuming a correspondingly structured form for Q (red), or by (i) approximating F_γ in structured form for each minibatch (for the Kronecker approximation, we use a permuted SVD to find the nearest Kronecker product in the least-squares sense; Van Loan and Pitsianis (1993)), (ii) averaging over minibatches (for the Kronecker approximation the two factors are averaged separately, as in KFAC), and (iii) inverting the result (black; note that in this case, the inverse inherits the structure). We report the squared error between $Q\boldsymbol{u}$ and $F_\gamma^{-1}\boldsymbol{u}$, averaged over $\boldsymbol{u} \sim \mathcal{N}(0, \Sigma_u)$, and normalized by the average norm of $F_\gamma^{-1}\boldsymbol{u}$. Here, to reflect the need of accurately estimating the action of F_γ^{-1} on the least salient parameter dimensions, we have chosen $\Sigma_u = F^{-1}$.

These equations are more easily interpreted when rewritten as

$$\frac{\beta_i^\star}{p_i}\frac{d\beta_i}{dt} = -\beta_i + \beta_i^\star \tag{24}$$

where $\beta_i^\star = (\lambda_i + \gamma)^{-1}$ is the corresponding eigenvalue of the solution $Q^\star$ (the "target eigenvalues"). The solution to these dynamics is

$$\beta_i(t) = \beta_i^\star + (\alpha - \beta_i^\star)\exp\left(\frac{-t}{\tau_i}\right) \quad \text{with } \tau_i \triangleq \frac{\beta_i^\star}{p_i}. \tag{25}$$

For $P = I$, i.e. $p_i = 1$, we recover the result of the main text (c.f. Figure 1): β_i converges exponentially to its target $\beta_i^\star$, but on a timescale τ_i proportional to $\beta_i^\star$ itself. This is a problem when F_γ is poorly conditioned, such that there is a broad range of $\beta_i^\star$: in this case, some β_i's will converge rapidly, and some others will converge very slowly.

Equation 25 suggests a solution based on a judicious choice of the preconditioner P. If somehow we could precondition the loss with $P = F_\gamma^{-1}$, then $p_i = \beta_i^\star$ and therefore $\tau_i = 1$ for all i – this case we have rapid uniform convergence of the inverse Fisher in all directions. While we do not know F_γ^{-1} (indeed this is what we are trying to learn ...), we do know that $Q(t)$ is supposed to converge (albeit slowly) towards F_γ^{-1}. Thus, we propose a simple time-dependent preconditioner $P(t) = Q(t)$. Empirically, we do find that this choice leads to better asymptotic convergence of the auxiliary loss, as illustrated in Figure 5A. Note that this only costs a single additional $Q\boldsymbol{v}$ product in every iteration.

D FishLeg inverse curvature estimation: flexible and accurate

In this section, we report on a series of simple experiments that show that FishLeg's inverse curvature estimation is typically more accurate and flexible than more conventional approaches.

First, Figure 4A shows that – when the parameterization of FishLeg's Q is sufficiently expressive to include F_γ^{-1}, Q converges to F_γ^{-1} as desired, despite only having access to stochastic estimates of F. This is because, using standard unbiased estimates of the Fisher matrix (or, practically, Fisher-vector products) on mini-batches in Equation 9, FishLeg's auxiliary loss and its gradient are also unbiased. With sufficiently small learning rate, we therefore expect Q to converge to the inverse damped Fisher solution. In contrast, a more naive scheme that computes an average of inverses of noisy Fisher estimates ('est. – inv. – avg.' in Figure 4A) yields a bias that persists asymptotically.

Second, when F_γ^{-1} lies outside the domain of the structured approximation (e.g. when it is not exactly a single Kronecker product, or a block-diagonal matrix), there is an advantage to directly approximating F_γ^{-1} in the desired structured form Q (FishLeg's strategy), rather than approximating F in such a form and then inverting the result. For one, (Garcia et al., 2023) had already argued that the former is more flexible than the latter, because one can use structured forms that need not be easily inverted (indeed FishLeg does not invert anything). Here, we show that even when the structured form is easily inverted, FishLeg still has a marked advantage (Figure 4B-D). In particular, the auxiliary loss allows the specification of a distribution of vectors $\boldsymbol{u}$ (specifically, their covariance) to promote learning the action of F_γ^{-1} on select directions in parameter space. This is not possible in a more conventional approach whereby the Fisher matrix is first approximated in structured form, then averaged, and finally inverted.

E Second-order Pruning: OBS-based methods

Most second-order pruning methods are based on the Optimal Brain Surgeon (OBS; Hassibi and Stork, 1992). OBS begins with a quadratic approximation of the loss function around the pre-trained parameter set $\boldsymbol{w}^\star$, typically assumed to be a minimum of the loss,

$$\delta\mathcal{L}(\delta\boldsymbol{w}) \quad \triangleq \quad \mathcal{L}(\boldsymbol{w}^\star + \delta\boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}^\star) \quad \approx \quad \frac{1}{2}\delta\boldsymbol{w}^\top H(\boldsymbol{w}^\star)\delta\boldsymbol{w}, \tag{26}$$

where $H(\boldsymbol{w}^\star)$ is the Hessian of the loss at $\boldsymbol{w}^\star$. Here, we will approximate the Hessian by the Fisher $F(\boldsymbol{w}^\star)$; most other works use the empirical Fisher matrix instead. This quadratic approximation leads to an analytical solution to the problem of optimally compensating for the deletion of a given weight w_i:

$$\delta\boldsymbol{w}^\star = -\frac{w_i^\star}{[F^{-1}(\boldsymbol{w}^\star)]_{ii}} F^{-1}(\boldsymbol{w}^\star)\boldsymbol{e}_i \tag{27}$$

where $\boldsymbol{e}_i$ is the i^{th} canonical basis vector (Hassibi and Stork, 1992). The corresponding (minimal) increase in loss resulting from the deletion of weight w_i is taken as its importance score:

$$\rho_i = \frac{w_i^2}{2[F^{-1}(\boldsymbol{w}^\star)]_{ii}}. \tag{28}$$

These equations have also been extended to handle the semi-structured pruning setting whereby small blocks of weights are treated as single units (Kurtic et al., 2022).

Existing second-order pruning methods mostly differ in the way they estimate $F^{-1}\boldsymbol{v}$ products to compute Equations 27 and 28. All scalable methods make a block-diagonal approximation for F. WoodFisher (Singh and Alistarh, 2020) and oBERT (Kurtic et al., 2022) partition the parameter space into small blocks assumed to be independent, and use the Woodbury identity to recursively update an estimate of the inverse empirical Fisher $\hat{F}_\mathcal{B}^{-1}$ for each block $\mathcal{B}$. These approaches have substantial memory requirements ($\mathcal{O}(|\mathcal{B}|n)$, where $|\mathcal{B}|$ is the block size and n is the total number of parameters in the model). M-FAC (Frantar et al., 2021) modifies this recursion to operate directly on $\hat{F}_\mathcal{B}^{-1}\boldsymbol{v}$ products, in a way that obviates the need for storing $\hat{F}_\mathcal{B}^{-1}$ (some parts of the computation can be cached and reused for any $\boldsymbol{v}$). This is typically much slower but requires less memory. In our work, FLS too approximates F^{-1} in block-diagonal form, but with much larger blocks corresponding to entire layers, and with blocks structured to guarantee computational and memory efficiency.

F Ablation Experiments

For the experiments discussed in this section, a simple linear layer with n inputs and a single output is used to perform controlled ablations and compare various approximations of the inverse Fisher

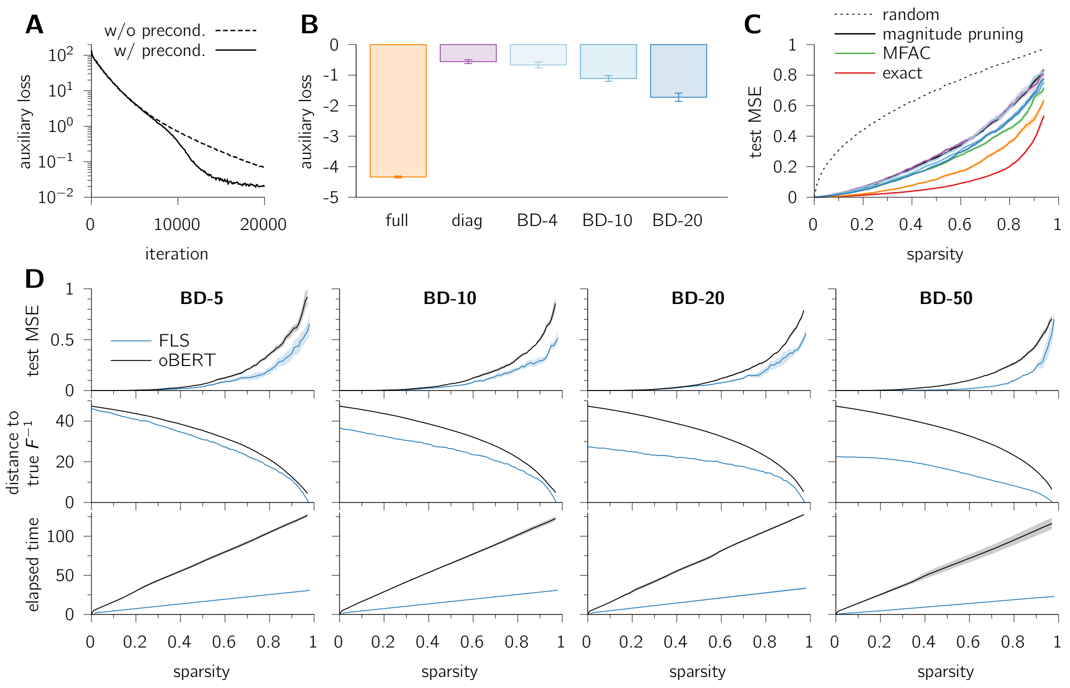

Figure 5: **Ablation experiments on synthetic data** in a toy setup to show: (A) the utility of preconditioning the auxiliary loss, (B) the predicted quality of the approximated Fisher in different scenario's, (C) the one-shot pruning performance of various Fisher approximations (including other baselines) and (D) the effect of implementing a block diagonal FishLeg approximation and it's comparison to oBERT (an OBS-derived approach) at various block sizes.

and their impact on one-shot pruning. In Figure 5A-C we choose $n = 100$ and in Figure 5D we set $n = 500$. The layer weights are drawn from $\mathcal{N}(0, 1/n)$, and inputs are drawn from $\mathcal{N}(0, \Sigma_x)$, where Σ_x is a random covariance matrix with eigenvalues $\{\lambda_i \propto e^{-i/10}\}$. Results are reported as mean $\pm$ s.e.m. over random seeds. Across all experiments, a batch size of 100 is chosen along with a damping parameter $\gamma = 0.01$. Note that in this toy example, the Fisher matrix is $F = \Sigma_x$, and does not depend on the weights. Figure 5A shows the effect of preconditioning the FishLeg auxiliary loss using the momentary approximation $Q(\boldsymbol{\lambda})$ of the inverse Fisher matrix. We observe that this preconditioning does indeed lead to faster asymptotic convergence. This is shown here for the 'full' approximation $Q = LL^\top$, which – in this case – is as expressive as the Kronecker parameterization of dense layers we have used in the experiments from the main text.

Figure 5B displays the quality of approximation of the inverse damped Fisher matrix, as measured by FishLeg's auxiliary loss after convergence[†], for various parameterizations of $Q(\boldsymbol{\lambda})$. We compare the 'full' parameterization $Q = LL^\top$ (orange), a positive diagonal parameterization (purple), and a set of positive-definite block-diagonal approximations with various block sizes (blues). These results show very clearly that a full approximation can achieve a much lower auxiliary loss when compared to less powerful approximations in this case.

Following from this, Figure 5C is reporting the one-shot pruning performance (test MSE) for the various FishLeg parameterizations shown in Figure 5B, as well as for magnitude pruning (black), MFAC ($m = 10$; green) and 'exact FLS' with $F = \Sigma_x$ appropriately masked and inverted before each pruning step (red). One can observe that the full approximation achieves a far closer performance to the 'exact' result across all other baselines in this study. Note that in this case, the 'exact FLS' characterises the limit of performance for second-order pruning methods. In this setting, we therefore find a strong correlation between the quality of the iFIM approximation (as measured by Garcia et al. (2023)'s auxiliary loss after convergence) and one-shot pruning performance (comparing Figure 5B and Figure 5C). In particular, block-diagonal approximations (as used by OBS/oBERT) perform

[†]Where the Adam learning rate is separately tuned for each approximation.

worse than the Kronecker-factored approximation (in this case also exact) and, indeed not much better than magnitude pruning or a simple diagonal approximation of the iFIM. Likewise, FLS with a Kronecker-factored Q performs better than MFAC (with rank parameter m generously set to 10, i.e. 10% of the parameter count, which would normally be intractable memory-wise).

Finally, Figure 5D provides a comparison between FLS with block-diagonal parameterization and oBERT for various block sizes (5, 10, 20, 50). In particular, this ablation study shows benefits of directly estimating the inverse FIM than estimating the FIM and inverting it. oBERT utilizies the WSM formula for effective estimation without explicit inversion, resulting in iterative update of the inverse of moving average for the empricial Fisher matrix. In the top panels, we present one-shot pruning performance (test MSE) as a function of sparsity for the two methods. In the middle panels, the standard affine-invariant Riemannian distance between the masked approximate block-diagonal inverse and the true masked Fisher inverse are shown, for each method. In the bottom panels, the wall-clock time as a function of sparsity is shown. For these experiments, oBERT uses 512 gradients at each pruning step, whereas FLS performs 20 steps of auxiliary loss minimization between pruning updates. These results show a systematic improvement in the inverse FIM estimates when using FLS, which implies that directly approximating the inverse Fisher in block-diagonal form (FLS) is better than approximating the Fisher in block-diagonal form before inverting each block (oBERT).

G Additional Experimental Details

Across all experiments we used a batch size of 128 and additionally applied standard flipping and cropping augmentations. Table 1 show the hyperparameter values used for each of the experimental setups, for the FishLeg optimizer. For other methods, we used an implementation of SGDm (with learning rate set at $1e-3$ and a momentum value of 0.9 and all others set at the PyTorch SGDm default) which preserved the sparsity map. More details of the experiments can be found in Section 3.

Hyperparameter	CIFAR-10	TinyIM
Batch Size	128	2048
η	10^{-3}	10^{-2}
α	10^{-5}	10^{-5}
η_{aux}	10^{-5}	10^{-6}
β	0.9	0.9
Damping γ	10^{-3}	1.0
Scale	10	1
Warmup	100	100

Table 1: Optimal hyperparameter values for FishLeg, identified as the result of a grid search. These hyperparameters were chosen to minimise the training loss across pruning. Any parameters not shown are left as default values in the FishLeg optimizer library.

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: All claims made in the abstract and introduction are reflected in the paper. To the best of our knowledge we show the first example of a tractable second-order optimizer being used for unstructured and semi-structured pruning. We provide explanations and extensive ablations to explain the modifications made to the FishLeg estimator and we then study these modifications on CIFAR-10 and TinyIM in Section 3.

   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: Limitations are discussed in Section 4.

   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: [NA]

Justification: We do not consider any formal theoretical proofs. Nevertheless, theoretical modifications to FishLeg are discussed and motivated in the main body and Appendices.

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 provide all details about architecture, datasets, and hyperparameters in the main text and appendix. Code will be made available in the final version of the paper.

   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 will provide code via GitHub with instructions to replicate our results. At the time of submission, uploading code is not an option for the OpenReview submission.

   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: Yes we provide all information regarding the training details and test details.

   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: [Yes]

   Justification: Yes our results are averaged over multiple seeds, and we include error bars where these are visible over the thickness of the line.

   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).
- 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: In the experimental section we provide a description of our computational setting (a single NVIDIA GeForce RTX 2080 GPU).

   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: The paper adheres to the NeurIPS Code of Ethics.

   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: This paper is a first investigation into pruning techniques with a second-order optimizer. The experiments, although not small, are still at a proof-of-concept stage. Therefore this paper does not contain any direct societal impact.

    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: the paper poses no such risks.

    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: [Yes]

    Justification: we provide a reference for the datasets used in this paper.

    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.
- 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: the paper does not release new assets.

    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: the paper does not involve crowdsourcing nor research with human subjects.

    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: the paper does not involve crowdsourcing nor research with human subjects.

    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.