Compressing Language Models for Specialized Domains

Miles Williams[♢♠] **George Chrysostomou**[♠] **Vitor Jeronymo**[♠] **Nikolaos Aletras**[♢]
[♢]University of Sheffield
[♠]Enterprise AI Services, AstraZeneca
{mwilliams15, n.aletras}@sheffield.ac.uk

Abstract

Language models (LMs) excel at tasks across diverse domains, yet require substantial computational resources during inference. Compression techniques such as pruning and quantization offer a practical path towards efficient LM deployment, exemplified by their ability to preserve performance on general-purpose benchmarks. However, general-purpose LM compression methods can negatively affect performance in specialized domains (e.g. biomedical or legal). Recent work has sought to address this issue, but requires a computationally expensive full-parameter fine-tuning pipeline. To this end, we propose MixCal, a novel calibration method designed to improve the in-domain performance of compressed LMs in a post-training setting. Through extensive experimentation, we demonstrate that MixCal substantially outperforms existing approaches on domain-specific tasks and preserves general performance. Notably, these performance gains are achieved while also reducing the computational cost of LM compression.[1]

1 Introduction

Language models (LMs) have demonstrated remarkable performance across tasks from a range of domains (Walsh et al., 2025; Guo et al., 2025; Kamath et al., 2025). Behind this success lies a recipe with two key ingredients: highly parameterized models and extensive training. However, the vast scale of these models presents substantial challenges in their deployment and application (Treviso et al., 2023; Zhu et al., 2024). Luccioni et al. (2024) suggest that the trend towards *general-purpose* models has introduced substantial yet potentially unnecessary inference costs.

Model compression techniques, such as quantization and pruning, are foundational approaches aimed at reducing the computational footprint of LMs during inference (Zhu et al., 2024). Quantization represents weights (and/or activations) with lower precision, while pruning removes less important weights. Notably, recent work has shown the successful application of quantization (Frantar et al., 2023; Lin et al., 2024) and pruning (Frantar and Alistarh, 2023; Sun et al., 2024) to general-purpose LMs without any additional training.

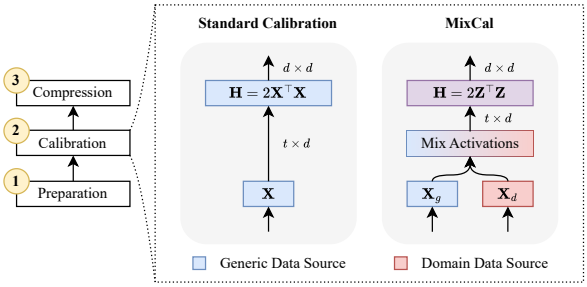

Figure 1: MixCal (§3) is applied within the calibration phase of model compression, leveraging a combination of activations from generic and domain-specific data.

LM compression studies typically focus on preserving general-purpose performance, i.e. language modeling and commonsense reasoning capabilities (Frantar and Alistarh, 2023; Ma et al., 2023; Sun et al., 2024). However, in practice, LMs may be deployed within only one particular domain, e.g. biomedical or legal (Labrak et al., 2024; Colombo et al., 2024; Ling et al., 2024; Chen et al., 2024). This scenario unlocks new paths towards improving inference efficiency by extracting domain-specific LMs from general-purpose models.

Zhang et al. (2024) proposed D-Pruner, a pruning method aiming to preserve weights that are influential to both domain-specific and general capabilities. To identify such weights, D-Pruner leverages the gradient information from a composite loss function that incorporates general weight importance scores. However, this requires a full-parameter fine-tuning pipeline for the LM, thus incurring substantial computational costs.

[1] https://github.com/mlsw/domain-compression

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7393–7415
March 24-29, 2026

The majority of post-training LM compression methods rely upon *calibration data*, a small amount of data used to aid the analysis of layer activations. Recent work has shown that calibration data can impact task performance (Williams and Aletras, 2024; Bandari et al., 2024; Williams et al., 2025). Inspired by this line of work, we investigate how to effectively leverage domain-specific data for calibration, aiming to maximize in-domain performance without sacrificing general capabilities. Our main contributions are as follows:

1. We find that in-domain calibration data can play an important role in LM pruning, maximizing performance retention on domain-specific tasks.

2. We propose MixCal, a novel calibration method for Hessian-based compression, enabling the effective use of in-domain calibration data (Figure 1). Our approach outperforms existing general and domain-specific pruning methods while offering greater computational efficiency.

2 Related Work

Quantization. The objective of quantization is to represent weights (and optionally activations) using fewer bits. This reduction in precision reduces memory requirements, and typically enables inference speedups (Gholami et al., 2021). Beyond their scale, contemporary LMs pose unique challenges for effective quantization, including the existence of high-magnitude outlier features (Bondarenko et al., 2021; Dettmers et al., 2022). Recent directions include: holding outlier weights in higher precision (Dettmers et al., 2022), Hessian-based weight sensitivity (Frantar et al., 2023), searching for optimal clipping thresholds (Wei et al., 2023), or combinations of these approaches (Lin et al., 2024; Dettmers et al., 2024).

Pruning. The aim of neural network pruning is to remove less important weights, therefore reducing the overall model size (LeCun et al., 1989). Pruning can be performed at the level of individual weights (unstructured), within groups of weights (semi-structured), or entire dimensions (structured) (Han et al., 2015; Mishra et al., 2021; Ma et al., 2023). In particular, 2:4 semi-structured sparsity (i.e. pruning two weights in every block of four) enables enhanced inference performance on NVIDIA GPUs (Mishra et al., 2021). However, the extensive size of Transformer-based (Vaswani et al., 2017) LMs presents challenges in pruning them optimally (Hassibi et al., 1993). Recent work has instead decomposed LM pruning into a sequential layer-wise approach, demonstrating remarkable performance retention, even at high sparsity levels (Frantar and Alistarh, 2023; Sun et al., 2024; Yin et al., 2024).

Domain-specific pruning. Early work focused on pruning deep neural networks for specific tasks (Han et al., 2015; Molchanov et al., 2017). This trend continued (Sanh et al., 2020; Lagunas et al., 2021; Kwon et al., 2022) following the advent of BERT (Devlin et al., 2019). However, the shift towards general-purpose LMs (Brown et al., 2020; Dredze et al., 2024) has led to a focus on preserving general performance (Frantar and Alistarh, 2023; Ma et al., 2023; Sun et al., 2024), i.e. language modeling and reasoning. Most recently, Zhang et al. (2024) proposed D-Pruner for domain-specific pruning, leveraging general weight importance to form a domain-specific training loss. However, this requires an expensive full-parameter fine-tuning pipeline, yet does not consistently outperform general-purpose pruning methods.

Calibration data. In a post-training setting, model compression usually relies upon calibration data (Wan et al., 2024). Calibration data consists of a small number of unlabeled examples for the generation of layer activations (Nagel et al., 2020; Hubara et al., 2021). Typically, these examples are randomly sampled from web text or pre-training datasets (e.g. C4; Raffel et al., 2020). Recent work has illustrated the influential role that calibration data can play, impacting the downstream performance of compressed models (Williams and Aletras, 2024; Williams et al., 2025). However, Bandari et al. (2024) suggest that pruning using downstream task data does not necessarily outperform generic data on the corresponding task.

3 MixCal

3.1 Preliminaries

The Optimal Brain Surgeon (OBS; Hassibi et al., 1993) algorithm leverages second-order derivatives to accurately prune weights from a neural network. These second-order derivatives, which indicate the curvature of the loss function with respect to the weights, are organized in a square matrix known as the Hessian $\mathbf{H}$. Based on the Hessian, the OBS algorithm iteratively removes the weight w_m with the lowest saliency ε_m, followed by applying the

optimal update for the remaining weights $\boldsymbol{\delta}_m$:

$$\varepsilon_m = \frac{1}{2}\frac{w_m^2}{[\mathbf{H}^{-1}]_{mm}}, \quad \boldsymbol{\delta}_m = -\frac{w_m}{[\mathbf{H}^{-1}]_{mm}} \cdot \mathbf{H}^{-1}_{:,m}$$

Frantar and Alistarh (2022) reformulate pruning as a case of the layer-wise compression problem, while retaining the OBS weight update procedure. Given a layer with input activations $\mathbf{X}$, the aim is to minimize the error between the original layer weights $\mathbf{W}$ and newly compressed weights $\widehat{\mathbf{W}}$:

$$\arg\min_{\widehat{\mathbf{W}}} \|\mathbf{X}\mathbf{W} - \mathbf{X}\widehat{\mathbf{W}}\|_F^2$$

As the layer input activations are derived from a fixed set of calibration data, the layer outputs (i.e. $\mathbf{Y} = \mathbf{X}\mathbf{W}$) are also fixed. Consequently, the layer-wise Hessian is computed as:

$$\mathbf{H} = 2\mathbf{X}^\top\mathbf{X}$$

3.2 Motivation

Language models are trained over diverse and expansive corpora. Accordingly, a random sample of generic calibration data often proves sufficient to approximate the Hessian (Frantar and Alistarh, 2023). These generic samples can also help preserve general capabilities (Bandari et al., 2024). However, LM weights additionally encode domain-specific knowledge (Singhal et al., 2023). For example, Figure 2 illustrates that a feature may be sensitive in a target domain yet appear unimportant under generic data. Therefore, we hypothesize that compressing LMs for specialized domains while maintaining general performance requires preserving weights that are important under both settings. To this end, we propose *MixCal* (Algorithm 1).

3.3 Mixing Activations

First, we extend layer-wise compression to a multi-objective setting. The standard layer-wise reconstruction loss for a layer with weights $\mathbf{W}$, compressed weights $\widehat{\mathbf{W}}$, and input activations $\mathbf{X}$ is:

$$\mathcal{L}(\widehat{\mathbf{W}}, \mathbf{X}) = \|\mathbf{X}\mathbf{W} - \mathbf{X}\widehat{\mathbf{W}}\|_F^2$$

We introduce two distinct sources of calibration data, a domain-specific dataset $\mathcal{D}_d$, and a general-purpose dataset $\mathcal{D}_g$. Samples from these datasets are used to form the domain-specific and general-purpose activations, $\mathbf{X}_d$ and $\mathbf{X}_g$, respectively. We then define a combined loss, which balances reconstruction between both calibration datasets:

$$\mathcal{L}_{\text{weighted}} = \alpha\mathcal{L}(\widehat{\mathbf{W}}, \mathbf{X}_d) + \beta\mathcal{L}(\widehat{\mathbf{W}}, \mathbf{X}_g)$$

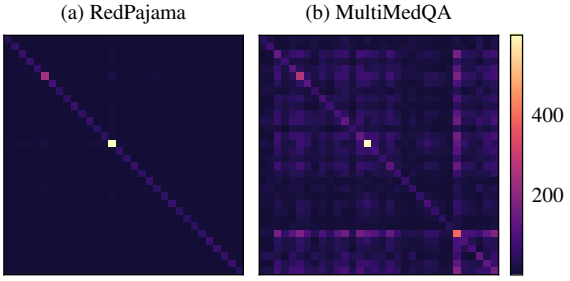

Figure 2: The Hessian at layer 16 of Mistral NeMo 12B, computed with (a) generic calibration data, and (b) domain-specific calibration data. For clarity, we present the magnitude of the elements for the first 32 features.

To control the relative contribution from each loss term, we introduce the weight coefficients α and β. We impose $\alpha + \beta = 1$ to avoid an arbitrary scaling, setting $\beta = 1 - \alpha$. The loss is therefore parameterized by a single coefficient, $\alpha \in [0, 1]$. This loss function yields the following Hessian:

$$\mathbf{H}_{\text{weighted}} = 2(\alpha\mathbf{X}_d^\top\mathbf{X}_d + \beta\mathbf{X}_g^\top\mathbf{X}_g)$$

However, this formulation treats the domain-specific and general-purpose activations independently. The Hessian $\mathbf{H}_{\text{weighted}}$ is a weighted sum of the domain-specific and general-purpose Hessians, namely $2\mathbf{X}_d^\top\mathbf{X}_d$ and $2\mathbf{X}_g^\top\mathbf{X}_g$. These terms only capture the second-order structure for their corresponding dataset. Consequently, $\mathbf{H}_{\text{weighted}}$ does not encode interactions between the two datasets.

We therefore propose a strategy which also incorporates such cross-dataset interactions. Concretely, we construct a mixed activation matrix $\mathbf{Z}$ that combines the activations from each of the calibration datasets. This is used to form the loss $\mathcal{L}_{\text{mix}}$:

$$\mathbf{Z} = \sqrt{\alpha}\mathbf{X}_d + \sqrt{\beta}\mathbf{X}_g, \quad \mathcal{L}_{\text{mix}} = \mathcal{L}(\widehat{\mathbf{W}}, \mathbf{Z})$$

Expanding the Hessian for $\mathcal{L}_{\text{mix}}$ illustrates the additional cross term that is introduced:

$$\mathbf{H}_{\text{mix}} = \mathbf{H}_{\text{weighted}} + 2\sqrt{\alpha\beta}(\mathbf{X}_d^\top\mathbf{X}_g + \mathbf{X}_g^\top\mathbf{X}_d)$$

The contribution from this term can be interpreted as how strongly the domain-specific and general-purpose activations align.

Finally, we note a connection to the training-time data augmentation concept of *mixup* (Zhang et al., 2018), which forms linear combinations of pairs of examples and their labels. Mixup has previously been applied directly to activations (Verma et al., 2019) and Transformer LMs (Sun et al., 2020). While our approach also forms linear combinations of activations, we use unlabeled examples solely to approximate the Hessian for compression.

3.4 Implementation

In practice, we compute the Hessian using the form $\mathbf{H}_{\text{mix}} = 2\mathbf{Z}^\top\mathbf{Z}$. When the total number of calibration examples is held constant, this can reduce the cost of calibration, since each update to the Hessian incorporates two examples simultaneously. We analyze the resulting efficiency gains in §5.3.

To empirically validate our approach, we integrate MixCal with SparseGPT (Frantar and Alistarh, 2023), a popular Hessian-based pruning algorithm. However, we emphasize that MixCal is not tied to any specific compression algorithm (see Figure 7, Appendix A.1 for results with GPTQ-M).

Algorithm 1 MixCal (simplified).

Require: Domain dataset $\mathcal{D}_d$, generic dataset $\mathcal{D}_g$, weight $\alpha \in [0, 1]$, embedding matrix $\mathbf{U}$, and weight matrices $\mathcal{W}$.

for $i \leftarrow 1$ to $|\mathcal{D}_d|$ **do** ▷ Since $|\mathcal{D}_d| = |\mathcal{D}_g|$.
  $(\mathbf{X}_{d,i}, \mathbf{X}_{g,i}) \leftarrow (\mathcal{D}_{d,i} \cdot \mathbf{U}, \mathcal{D}_{g,i} \cdot \mathbf{U})$
end for
for $\mathbf{W} \in \mathcal{W}$ **do**
  $\mathbf{H} \leftarrow \mathbf{0}$
  for $i \leftarrow 1$ to $|\mathcal{D}_d|$ **do**
    $\mathbf{Z}_i \leftarrow \sqrt{\alpha}\mathbf{X}_{d,i} + \sqrt{1-\alpha}\mathbf{X}_{g,i}$
    $\mathbf{H} \leftarrow \frac{i-1}{i}\mathbf{H} + \frac{2}{i}(\mathbf{Z}_i^\top \mathbf{Z}_i)$
  end for
  $\mathbf{W} \leftarrow \text{Compress}(\mathbf{W}, \mathbf{H})$ ▷ E.g. via SparseGPT.
  for $i \leftarrow 1$ to $|\mathcal{D}_d|$ **do**
    $(\mathbf{X}_{d,i}, \mathbf{X}_{g,i}) \leftarrow (\mathbf{X}_{d,i}\mathbf{W}, \mathbf{X}_{g,i}\mathbf{W})$
  end for
end for

4 Experimental Setup

4.1 Compression Methods

Pruning methods can be formulated as a function that computes a saliency score $\mathbf{S}_{ij}$ for each weight $\mathbf{W}_{ij}$ in a given layer. They optionally use the layer input activations $\mathbf{X}$, derived from calibration data. We adopt the following methods as baselines.

Magnitude (Janowsky, 1989; Han et al., 2015). Based on the assumption that removing the smallest weights will have the least effect, magnitude pruning simply uses the weight magnitude for saliency:

$$\mathbf{S}_{ij} = |\mathbf{W}_{ij}|$$

SparseGPT (Frantar and Alistarh, 2023). Building upon the OBS procedure, SparseGPT offers an efficient iterative approximation. The saliency metric is computed as follows, where λ is a dampening factor to enable inversion of the Hessian:

$$\mathbf{S}_{ij} = \left[|\mathbf{W}|^2 / \text{diag}\left((\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\right)\right]_{ij}$$

Wanda (Sun et al., 2024). Improving upon the computational efficiency of SparseGPT, the Wanda pruning metric approximates the diagonal of the inverse Hessian via the ℓ_2 norm of the activations:

$$\mathbf{S}_{ij} = |\mathbf{W}_{ij}| \cdot \|\mathbf{X}\|_2$$

D-Pruner (Zhang et al., 2024). D-Pruner is a domain-specific pruning method. The first step of D-Pruner is to compute the general importance $\mathbf{G}_{ij}$ using a general dataset $\mathcal{D}_g$, similar to SparseGPT. Second, it uses a composite loss function $\mathcal{L}_{\text{DP}}$ to identify weights that are important for both general and domain-specific knowledge. This consists of the cross-entropy loss $\mathcal{L}_{\text{CE}}$ with a regularization term controlled by hyperparameter λ_g, where $\mathbf{W}'$ is the updated weight matrix:

$$\mathcal{L}_{\text{DP}} \approx \mathcal{L}_{\text{CE}} + \lambda_g \sum_{i,j} \mathbf{G}_{ij}(\mathbf{W}'_{ij} - \mathbf{W}_{ij})^2$$

The gradients are computed using a full-parameter fine-tuning pipeline (Lv et al., 2024). Given a domain-specific dataset $\mathcal{D}_d$, the saliency is:

$$\mathbf{S}_{ij} \approx \left| \frac{\partial \mathcal{L}_{\text{DP}}(\mathcal{D}_d)}{\partial \mathbf{W}_{ij}} \mathbf{W}_{ij} + \frac{1}{2}\left[\frac{\partial \mathcal{L}_{\text{DP}}(\mathcal{D}_d)}{\partial \mathbf{W}_{ij}} \mathbf{W}_{ij}\right]^2 \right|$$

GPTQ-M (Frantar et al., 2025). GPTQ adopts a Hessian-based weight sensitivity metric for quantization (Frantar et al., 2023). We select this method to enable a fair comparison with SparseGPT, which uses GPTQ for joint sparsification and quantization. We include improvements to the original algorithm suggested by Frantar et al. (2025). Specifically, this identifies optimal group-wise clipping thresholds, similar to AWQ (Lin et al., 2024). For clarity, we refer to this improved method as GPTQ-M.

Compression configurations. Guided by prior work (Sun et al., 2024; Frantar et al., 2025), we focus our experiments on the following settings:

- **50% (unstructured) sparsity**. First, we experiment with individually pruning half of all layer weights, offering the highest possible granularity.
- **2:4 (semi-structured) sparsity**. We then examine pruning at the granularity of two weights in every group of four, enabling enhanced GPU inference performance (Mishra et al., 2021).
- **4-bit quantization with 2:4 sparsity**. Finally, we combine 2:4 sparsity with 4-bit quantization of the remaining weights, enabling up to 5.3× GPU inference speedups (Frantar et al., 2025).

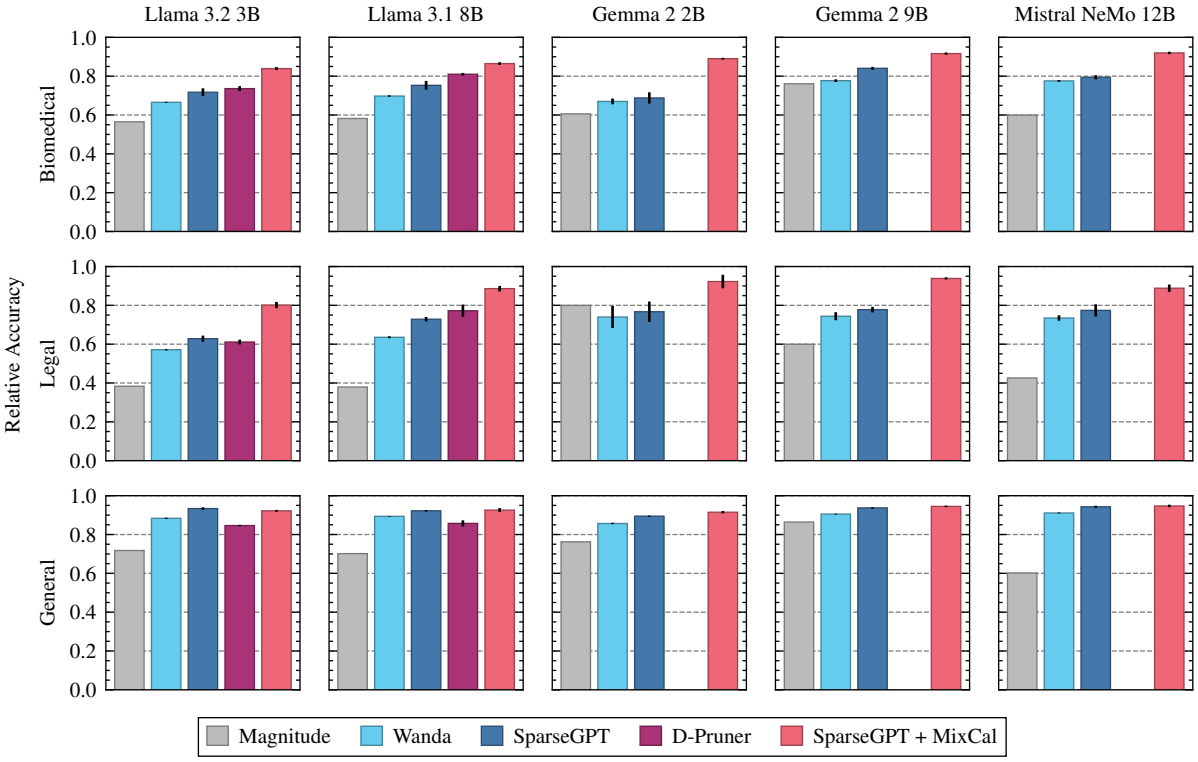

Figure 3: The average benchmark accuracy when pruning to 50% sparsity, relative to the original model.

4.2 Domains and Tasks

To assess the efficacy of our approach on downstream tasks, we experiment with two of the most widely explored domains in NLP, the *biomedical* (Lee et al., 2019; Gu et al., 2021; Luo et al., 2022; Singhal et al., 2023, 2025) and *legal* (Chalkidis et al., 2019; Zheng et al., 2021; Henderson et al., 2022; T.y.s.s et al., 2024; Niklaus et al., 2024) domains. See Appendix C for concrete task examples.

Biomedical. We use the MultiMedQA benchmark (Singhal et al., 2023), specifically the PubMedQA (Jin et al., 2019), MedQA (Jin et al., 2021), MedMCQA (Pal et al., 2022) tasks, and relevant subsets from MMLU (Hendrycks et al., 2021) (anatomy, clinical knowledge, college medicine, medical genetics, professional medicine, college biology). To assess language modeling performance, we use the BioLaySumm PLOS dataset (Goldsack et al., 2022, 2023) comprising biomedical articles.

Legal. We follow Colombo et al. (2024) in using Legal-MMLU, covering jurisprudence, professional law, and international law specialties (Hendrycks et al., 2021). We also use the CaseHOLD (Zheng et al., 2021) and ECtHR (Task A) (Chalkidis et al., 2019) datasets from the LexGLUE benchmark (Chalkidis et al., 2022), comprising US Supreme Court opinions and European Court of Human Rights cases, respectively. To evaluate language modeling performance, we use the BillSum dataset (Kornilova and Eidelman, 2019) of US Congressional and California state bills.

General. To assess general performance, we use all commonsense reasoning tasks adopted by Frantar and Alistarh (2023) and Sun et al. (2024): ARC (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), LAMBADA (Paperno et al., 2016), OpenBookQA (Banerjee et al., 2019), PIQA (Bisk et al., 2020), RTE (Dagan et al., 2006), StoryCloze (Mostafazadeh et al., 2016), and WinoGrande (Sakaguchi et al., 2021). We use WikiText-2 (Merity et al., 2017) to assess language modeling.

4.3 Calibration Data

Data sources. To create our general-purpose calibration sets, we follow Dettmers et al. (2024) in using RedPajama (Weber et al., 2024), an open reproduction of the LLaMA training data. For the domain-specific calibration sets, we use MultiMedQA (biomedical) and LexGLUE (legal). In all cases, we sample data from the training splits only.

Data quantity. For a fair comparison between compression methods, we use 1024 calibration examples. As D-Pruner consists of two distinct stages

to identify general and domain-specific weight importance, we allow 1024 examples from each dataset to better match the original work (Zhang et al., 2024). For our own method, we simply select half of the examples (i.e. 512) from each dataset.

Sampling. We randomly sample segments of 2048 tokens following Frantar et al. (2023), avoiding any selection bias. In the case of the domain-specific datasets, which may contain shorter examples, we follow Touvron et al. (2023) in concatenating examples for a consistent length. We repeat the sampling process to create five distinct calibration sets, used to assess the variance in performance.

4.4 Models

We experiment with popular open-weights LMs, covering different model families and sizes: (1) **Llama 3.2 3B** and **3.1 8B** (Grattafiori et al., 2024), (2) **Gemma 2 2B** and **9B** (Riviere et al., 2024), and (3) **Mistral NeMo 12B** (2407) (Jiang et al., 2024).

5 Results & Discussion

5.1 Pruning

Figure 3 presents the benchmark accuracy when pruning to 50% sparsity, relative to the original model.[2] We report the mean value and standard deviation across five calibration sets. For brevity, we present the average performance across domain-specific models. We additionally present complete results across all models in Appendix E.

A note on hyperparameters. To maximize the performance of the D-Pruner baseline, we perform an extensive hyperparameter search across $\lambda_g \in \{0.1, 0.01, 0.001\}$ and group size $\in \{\text{None}, 128\}$ for each model and domain. We then present results for only the best performing combinations. We present complete results across all hyperparameters in Appendix E. In contrast, we do not optimize the hyperparameter for our approach (α) and simply use 0.5 across all models and domains. We ablate the impact of this hyperparameter in Appendix A.3.

MixCal benefits domain performance. We observe that across both biomedical and legal domains, our approach consistently outperforms all other compression methods. For example, we observe that MixCal achieves an average relative accuracy of 88.6% on the legal benchmark for Llama 3.1 8B. In comparison, SparseGPT and D-Pruner see 72.9% and 77.2%, respectively. For the biomedical benchmark, a similar trend can be observed. MixCal achieves 86.5%, while SparseGPT and D-Pruner reach 75.3% and 81.0%, respectively. This indicates that MixCal can be effective at identifying weights influential to in-domain performance.

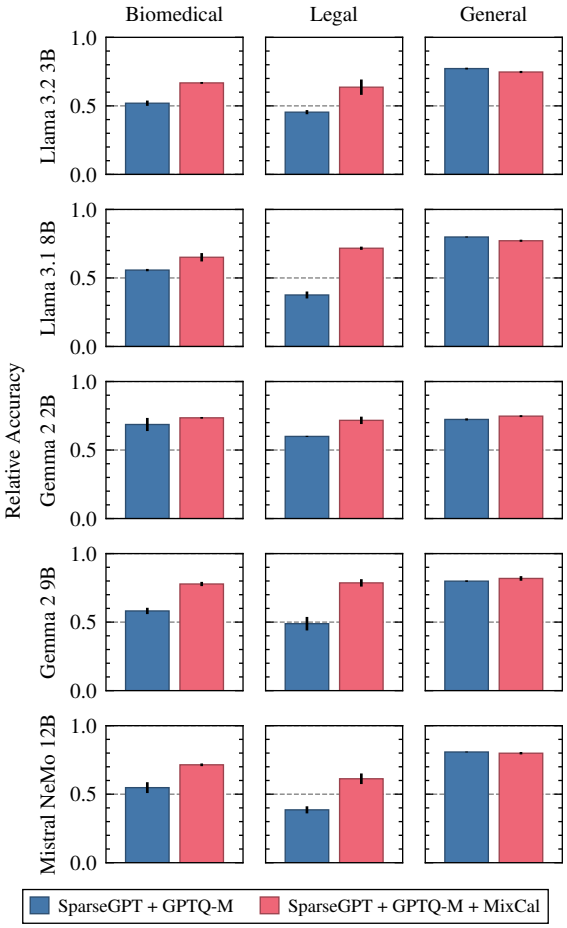

Figure 4: Average accuracy when applying 4-bit quantization and 2:4 sparsity, relative to the original model.

General performance is comparable in all cases. In addition to substantial improvements on domain-specific benchmarks, general performance remains similar to the general-purpose pruning baselines. With Llama 3.1 8B, MixCal achieves 92.6% relative accuracy for general tasks on average. In comparison, SparseGPT achieves a slightly lower value of 92.2%, while D-Pruner reaches only 85.7%. Intriguingly, we note that MixCal achieves higher performance than even the general-purpose methods for most model families. For example, MixCal sees an average of 94.5% with Gemma 2 9B, while SparseGPT reaches 93.7%. This suggests that MixCal can effectively establish domain-specific features without sacrificing general performance.

[2]The D-Pruner (Zhang et al., 2024) implementation only supports models with the Llama architecture. See Limitations.

Performance gains are generally model-agnostic. Finally, we observe that the performance benefits of MixCal are similar irrespective of the model size and family. For example, we consider two similarly sized models, Llama 3.1 8B and Gemma 2 9B. In the biomedical benchmark, we observe an 11.2 and 7.5 point increase in relative accuracy over SparseGPT, respectively. For the legal benchmark, we see a 15.7 and 16.1 point increase. This suggests that the performance gains from MixCal are independent of the model family.

Language modeling follows a similar trend. Table 7 (Appendix E) presents perplexity results. Similar to the downstream task experiments, we note that MixCal achieves the best performance on in-domain language modeling. Considering Llama 3.1 8B, MixCal achieves a perplexity of 6.0 compared to 6.7 with D-Pruner for the legal domain. For the biomedical domain, MixCal has a perplexity of 10.7, compared to 14.7 from D-Pruner. The datasets used to evaluate perplexity are not used for calibration, suggesting that MixCal assists with identifying *domain*-specific features.

MixCal appears to generalize beyond specific tasks. Across all models and domains, we observe that the performance benefits of MixCal continue to tasks not included in the calibration data. In Table 10 (Appendix E), we present complete per-task results. We consider the MMLU tasks, which do not have training data, and are therefore not represented in the calibration data. For Llama 3.1 8B, MixCal achieves an absolute increase in accuracy of 7.7 points over SparseGPT in the biomedical domain. In the legal domain, MixCal achieves an increase of 2.7 points in accuracy. This suggests that MixCal can identify features that are relevant to the domain, rather than only a specific task.

5.2 Joint Pruning and Quantization

We further examine the performance of our approach when jointly applying pruning and quantization by reusing the same inverse Hessian (Frantar and Alistarh, 2023). This has the advantage of allowing pruning and quantization decisions to influence each other, and enables quantization at almost no extra cost. Figure 4 presents benchmark accuracy when jointly applying 2:4 sparsity with 4-bit quantization, relative to the original model.

MixCal improves in-domain performance while sustaining general performance. MixCal achieves substantially greater domain performance than SparseGPT for 2:4 sparsity. For example, it sees a relative accuracy of 77.8% on the biomedical benchmark for Gemma 2 9B, versus 58.2% with SparseGPT. For general performance, we observe a similar trend to the pruning results, with MixCal also performing comparably. This illustrates that MixCal can be reliably used with quantization.

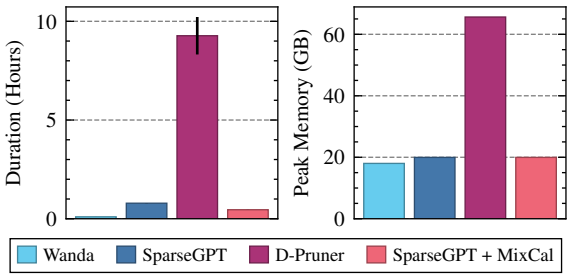

Figure 5: The average duration and peak memory allocated when pruning Llama 3.1 8B with each method, as measured using an NVIDIA A100 80GB GPU.

5.3 Compression Efficiency

MixCal does not sacrifice efficiency. Figure 5 presents the time and memory requirements of each method, which are often limiting factors in practice. First, we observe that MixCal (0.5 hours) reduces the duration of compression compared to SparseGPT (0.8 hours). This is considerably faster than D-Pruner, which on average takes 9.3 hours, 18 times longer than our approach. We also observe that MixCal does not increase the memory required for compression over SparseGPT, with both using up to 20 GB of memory. In contrast, D-Pruner uses up to 65.6 GB of memory, over three times more than our approach. This suggests that MixCal is more practical than D-Pruner, offering lower computational resource requirements.

6 Analysis

6.1 Ablating MixCal

To better understand the role that in-domain data plays in pruning, we conduct an ablation study of MixCal. Table 1 presents the in-domain and general language modeling performance (perplexity) versus a SparseGPT baseline with generic calibration data. We specifically show the effect of (1) adding in-domain data to the calibration data mixture, and (2) additionally using MixCal. In both cases, we use the same equal mixture of in-domain and generic data. Results are averaged across models, with granular results in Table 11, Appendix E.

Method	In-domain	General	Average
Biomedical			
SparseGPT	16.4	14.0	15.2
+ In-domain data	15.1	13.8	14.4
+ MixCal (Ours)	**14.4**	**13.4**	**13.9**
Legal			
SparseGPT	7.2	14.0	10.6
+ In-domain data	**6.7**	13.7	10.2
+ MixCal (Ours)	**6.7**	**13.5**	**10.1**

Table 1: Perplexity when pruning to 50% unstructured sparsity with (1) in-domain data, and (2) MixCal, averaged across all models.

Domain-specific data benefits language modeling performance. We first observe that using a mix of domain-specific and generic calibration data can substantially improve in-domain performance. For example, when using a mix of in-domain data, we find lower perplexity scores for both domains. Specifically, we observe an average perplexity of 15.1 versus 16.4 for the biomedical domain, and 6.7 versus 7.2 for the legal domain. This highlights the influential role played by the calibration data, corroborating Williams and Aletras (2024).

MixCal maximizes overall performance. Finally, we observe that by applying MixCal, language modeling performance is maximized compared to introducing domain-specific calibration data alone. For example, MixCal achieves a reduction of 0.7 (average) perplexity for the biomedical domain (14.4 versus 15.1). For the legal domain, we note that the average perplexity is equivalent. Considering overall performance, i.e. the average of domain-specific and general performance, MixCal yields the best results. This suggests that the addition of MixCal can balance domain-specific and general performance effectively.

6.2 Performance in Other Languages

To explore whether MixCal generalizes beyond English, we further experiment with a Chinese-language model and benchmarks. We select the Chinese language as it is morphologically distinct from English and well-resourced in terms of models and domain-specific evaluation tasks.

Experimental setup. We select the Yi 1.5 6B model (Young et al., 2025) as it (1) uses the Llama architecture, enabling experiments with D-Pruner, and (2) achieves strong performance on standard benchmarks. To assess biomedical performance, we use the Comprehensive Medical Benchmark (CMB; Wang et al., 2024). For legal performance, we use the Chinese AI and Law (CAIL2018) challenge dataset (Xiao et al., 2018). Similar to our experimental setup in English, we sample in-domain calibration data from the training split of each benchmark. For general-purpose data, we follow Kurz et al. (2026) in using mC4 (Xue et al., 2021). We use a mixture of Chinese and English to reflect the model pre-training data (Young et al., 2025).

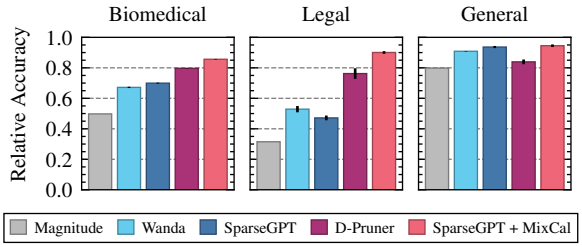

Figure 6: The mean accuracy for Yi 1.5 6B on Chinese-language benchmarks, relative to the original model.

MixCal appears language-agnostic. Figure 6 presents the relative benchmark accuracy when pruning to 50% sparsity, similar to earlier experiments (§5.1). The results indicate that MixCal outperforms other pruning approaches for domain performance, while maintaining comparable general performance to SparseGPT. For example, MixCal achieves a relative accuracy of 85.6% in the biomedical domain, compared to 70.0% and 79.6% from SparseGPT and D-Pruner, respectively. In the legal domain, the performance retention from MixCal (90.1%) is substantially greater than both SparseGPT (47.1%) and D-Pruner (76.2%). These findings are in line with the English language tasks, suggesting that MixCal is language-agnostic.

7 Conclusion

In this paper, we proposed MixCal as a solution for creating compressed LMs for specialized domains. We empirically validated MixCal using a plethora of pre-trained models and evaluation tasks. Our approach represents a substantial advancement over earlier work such as D-Pruner, offering consistent performance improvements with a smaller computational footprint. We hope that our study will inspire further work towards the efficient deployment of LMs in specialized domains. As future work, we are interested in exploring how synthetic calibration data could be incorporated to further enhance the performance of domain-specific LM compression (Williams et al., 2025).

Limitations

Model selection in the D-Pruner experiments. The D-Pruner (Zhang et al., 2024) implementation supports only the Llama model architecture.[3] Consequently, we are limited to offering comparisons for only the models using the Llama architecture (i.e. Llama 3.2 3B, Yi 1.5 6B, and Llama 3.1 8B). We emphasize that our approach substantially outperforms D-Pruner across all tested models and domains. Therefore, we expect that this trend would continue for the models not supported by D-Pruner.

Ethical Considerations

Our work enables the efficient and effective compression of LMs for specialized domains. We note that this poses dual-use concerns, as it may enable misuse at a lower cost (Weidinger et al., 2022). However, we emphasize that our approach is unlikely to enhance or introduce new harmful abilities, as the performance of compressed models is constrained by the capabilities of the original.

Acknowledgments

We would like to thank Huiyin Xue and the anonymous reviewers for their invaluable feedback. Additionally, we are sincerely grateful to Huiyin Xue for assistance with the Chinese-language tasks. MW is supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation grant EP/S023062/1. NA is supported by EPSRC grant EP/Y009800/1, part of the RAI UK Keystone projects.

References

Abhinav Bandari, Lu Yin, Cheng-Yu Hsieh, Ajay Kumar Jaiswal, Tianlong Chen, Li Shen, Ranjay Krishna, and Shiwei Liu. 2024. Is c4 dataset optimal for pruning? an investigation of calibration data for LLM pruning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18089–18099, Miami, Florida, USA. Association for Computational Linguistics.

Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129, Florence, Italy. Association for Computational Linguistics.

[3] https://github.com/psunlpgroup/D-Pruner

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Ilias Chalkidis. 2023. ChatGPT may pass the bar exam soon, but has a long way to go for the LexGLUE benchmark. *Preprint*, arXiv:2304.12202.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Zhiyu Chen, Jing Ma, Xinlu Zhang, Nan Hao, An Yan, Armineh Nourbakhsh, Xianjun Yang, Julian McAuley, Linda Ruth Petzold, and William Yang Wang. 2024. A survey on large language models for critical societal domains: Finance, healthcare, and law. *Transactions on Machine Learning Research*. Survey Certification.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind

Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. SaulLM-7B: A pioneering large language model for law. *Preprint*, arXiv:2403.03883.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Tim Dettmers, Ruslan A. Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2024. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. In *The Twelfth International Conference on Learning Representations*.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7750–7774. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Dredze, Genta Indra Winata, Prabhanjan Kambadur, Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, David S Rosenberg, and Sebastian Gehrmann. 2024. Academics can contribute to domain-specialized language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5100–5110, Miami, Florida, USA. Association for Computational Linguistics.

Elias Frantar and Dan Alistarh. 2022. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems*, volume 35, pages 4475–4488. Curran Associates, Inc.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Elias Frantar, Roberto L. Castro, Jiale Chen, Torsten Hoefler, and Dan Alistarh. 2025. MARLIN: Mixed-precision auto-regressive parallel inference on large language models. In *Proceedings of the 30th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, PPoPP '25, page 239–251, New York, NY, USA. Association for Computing Machinery.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference. *Preprint*, arXiv:2103.13630.

Tomas Goldsack, Zheheng Luo, Qianqian Xie, Carolina Scarton, Matthew Shardlow, Sophia Ananiadou, and Chenghua Lin. 2023. Overview of the biolaysumm 2023 shared task on lay summarization of biomedical research articles. In *Proceedings of the 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 468–477, Toronto, Canada. Association for Computational Linguistics.

Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. 2022. Making science simple: Corpora for the lay summarisation of scientific literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10589–10604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng

Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Aditya K, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 44123–44279. Curran Associates, Inc.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 180 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

B. Hassibi, D.G. Stork, and G.J. Wolff. 1993. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1.

Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256GB open-source legal dataset. In *Advances in Neural Information Processing Systems*, volume 35, pages 29217–29234. Curran Associates, Inc.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. 2021. Accurate post training quantization with small calibration sets. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4466–4475. PMLR.

Steven A. Janowsky. 1989. Pruning versus clipping in neural networks. *Phys. Rev. A*, 39:6600–6603.

Albert Jiang, Alexandre Sablayrolles, Alexis Tacnet, Alok Kothari, Antoine Roux, Arthur Mensch, Audrey Herblin-Stoop, Augustin Garreau, Austin Birky, Bam4d, Baptiste Bout, Baudouin de Monicault, Blanche Savary, Carole Rambaud, Caroline Feldman, Devendra Singh Chaplot, Diego de las Casas, Eleonore Arcelin, Emma Bou Hanna, and 49 others. 2024. Mistral NeMo.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, and 196 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China. Association for Computational Linguistics.

Simon Kurz, Jian-Jia Chen, Lucie Flek, and Zhixue Zhao. 2026. On the limitations of language-targeted pruning: Investigating the calibration language impact in multilingual LLM pruning. *Transactions of the Association for Computational Linguistics*, 14:167–192.

Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. 2022. A fast post-training pruning framework for transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 24101–24116. Curran Associates, Inc.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. BioMistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5848–5864, Bangkok, Thailand. Association for Computational Linguistics.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander Rush. 2021. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10619–10629, Online and Punta Cana,

Dominican Republic. Association for Computational Linguistics.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device llm compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Dhagash Mehta, Stefano Pasquali, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, and 5 others. 2024. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *Preprint*, arXiv:2305.18703.

Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. Power hungry processing: Watts driving the cost of AI deployment? In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 85–99, New York, NY, USA. Association for Computing Machinery.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409.

Kai Lv, Yuqing Yang, Tengxiao Liu, Qipeng Guo, and Xipeng Qiu. 2024. Full parameter fine-tuning for large language models with limited resources. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8187–8198, Bangkok, Thailand. Association for Computational Linguistics.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-Pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. *Preprint*, arXiv:2104.08378.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel Ho. 2024. MultiLegalPile: A 689GB multilingual legal corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15077–15094, Bangkok, Thailand. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor

Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 178 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, pages 20378–20389. Curran Associates, Inc.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, and 13 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R. Pfohl, Heather Cole-Lewis, Darlene Neal, Qazi Mamunur Rashid, Mike Schaekermann, Amy Wang, Dev Dash, Jonathan H. Chen, Nigam H. Shah, Sami Lachgar, Philip Andrew Mansfield, and 16 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R. Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H. Martins, André F. T. Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjan Balasubramanian, and 3 others. 2023. Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11:826–860.

Santosh T.y.s.s, Vatsal Venkatkrishna, Saptarshi Ghosh, and Matthias Grabmair. 2024. Beyond borders: Investigating cross-jurisdiction transfer in legal case summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4136–4150, Mexico City, Mexico. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Preprint*, arXiv:1706.03762.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, and 23 others. 2025. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. Efficient large language models: A survey. *Transactions on Machine Learning Research*. Survey Certification.

Xidong Wang, Guiming Chen, Song Dingjie, Zhang Zhiyi, Zhihong Chen, Qingying Xiao, Junying Chen,

Feng Jiang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2024. CMB: A comprehensive medical benchmark in Chinese. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6184–6205, Mexico City, Mexico. Association for Computational Linguistics.

Maurice Weber, Daniel Y Fu, Quentin Gregory Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Re, Irina Rish, and Ce Zhang. 2024. RedPajama: an open dataset for training large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1648–1665, Singapore. Association for Computational Linguistics.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, and 4 others. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 214–229, New York, NY, USA. Association for Computing Machinery.

Miles Williams and Nikolaos Aletras. 2024. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118, Bangkok, Thailand. Association for Computational Linguistics.

Miles Williams, George Chrysostomou, and Nikolaos Aletras. 2025. Self-calibration for language model quantization and pruning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10149–10167, Albuquerque, New Mexico. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. CAIL2018: A large-scale legal dataset for judgment prediction. *Preprint*, arXiv:1807.02478.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Mykola Pechenizkiy, Yi Liang, Zhangyang Wang, and Shiwei Liu. 2024. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, and 12 others. 2025. Yi: Open foundation models by 01.AI. *Preprint*, arXiv:2403.04652.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Nan Zhang, Yanchi Liu, Xujiang Zhao, Wei Cheng, Runxue Bao, Rui Zhang, Prasenjit Mitra, and Haifeng Chen. 2024. Pruning as a domain-specific LLM extractor. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1417–1428, Mexico City, Mexico. Association for Computational Linguistics.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help?: assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 159–168. ACM.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

A Additional Ablation Studies

A.1 4-bit Quantization

MixCal generalizes to quantization. We further examine the performance and transferability of our approach under the setting of quantization. Figure 7 presents the benchmark accuracy when applying 4-bit quantization with GPTQ-M and MixCal, relative to the original model. This uncovers the extent to which MixCal benefits each method independently in the joint sparsification and quantization experiments (§5.2). We observe that general performance is consistently preserved, with GPTQ-M + MixCal performing comparably to GPTQ-M across all models. This suggests that MixCal is transferable across settings in Hessian-based compression.

MixCal improves in-domain performance. We observe that MixCal can achieve greater domain performance than GPTQ-M alone. For example, MixCal sees a relative accuracy of 96.6% on the biomedical benchmark for Llama 3.1 8B, versus 94.7% with GPTQ-M (Figure 7). This indicates that MixCal can be reliably used for LM quantization. We note that the performance gains for quantization are smaller than in the pruning experiments. This is expected, as quantization is less sensitive to calibration data at the tested sparsity level and bit width (Williams and Aletras, 2024).

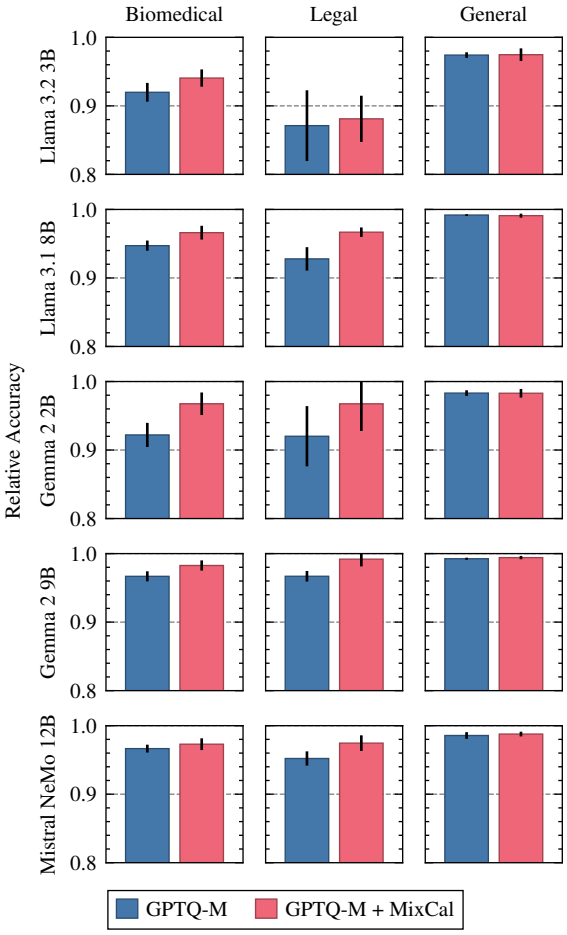

Figure 7: Average accuracy when applying 4-bit quantization, relative to the original model.

A.2 Performance Across Sparsity Levels

In Figure 8, we examine how MixCal performs compared to the other pruning methods across different sparsity levels. We report the average in-domain and general benchmark performance, with standard deviation denoted by the shaded regions. We first observe that in-domain performance with MixCal consistently remains higher than other approaches, even beyond 50% sparsity. For example, accuracy on the legal benchmark at 60% sparsity is 43.3% with MixCal, compared to 27.3% with SparseGPT. Similarly, general benchmark performance remains comparable to SparseGPT across all sparsity levels. This suggests that MixCal can be used to effectively isolate domain-specific features without sacrificing generic performance.

A.3 MixCal Hyperparameter Analysis

In Figure 9, we examine how different contributions from domain-specific and general-purpose data impact benchmark performance, i.e. by varying α. Following the experiments in §5.1, we examine this at 50% unstructured sparsity.

Increasing α generally increases in-domain performance. First, we observe that in-domain performance (i.e. biomedical or legal) increases following a larger contribution from the domain-specific Hessian. Considering Gemma 2 9B for the biomedical domain, we observe that $\alpha = 0.1$ achieves an average accuracy of 89.1%, whilst $\alpha = 0.9$ sees 93.1%. Notably, we observe that even a small contribution from the in-domain Hessian ($\alpha = 0.1$) is enough to offer improvements for in-domain performance. For example, in the legal domain Llama 3.2 3B achieves an average accuracy of 63.1% with $\alpha = 0$, yet 71.0% with $\alpha = 0.1$.

Maximizing α can harm reasoning performance. We observe that using $\alpha = 1$ can lead to a drop in reasoning performance. For example, when targeting the legal domain, Llama 3.2 3B achieves an average accuracy of 91.5% with $\alpha = 0.9$, yet sees 89.3% when $\alpha = 1$. This suggests that the use of generic data can benefit reasoning performance.

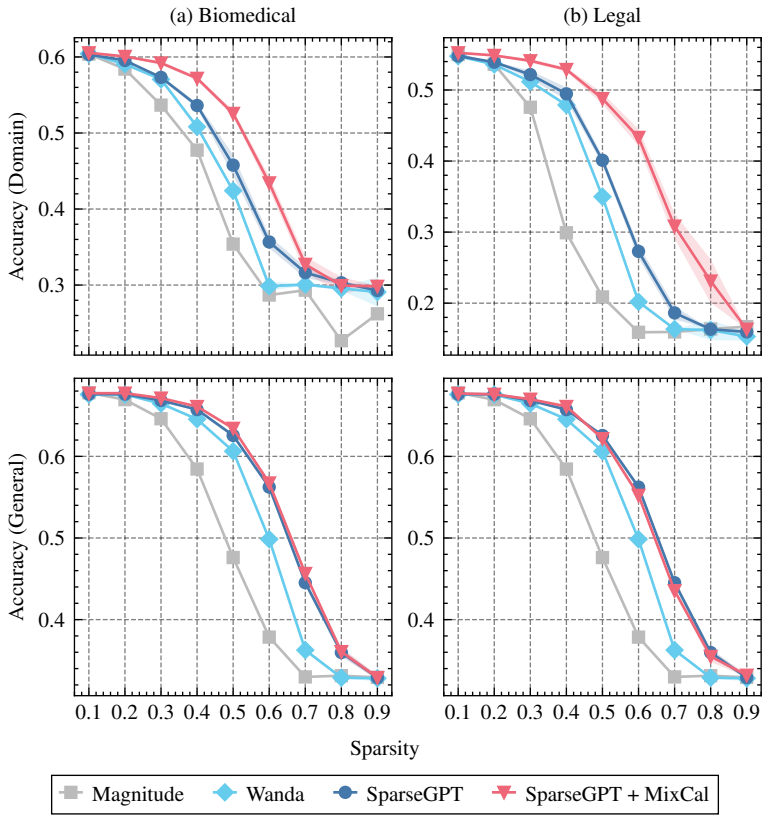

Figure 8: Benchmark accuracy when pruning Llama 3.1 8B to different sparsity levels with each method. Standard deviation is denoted by the shaded regions.

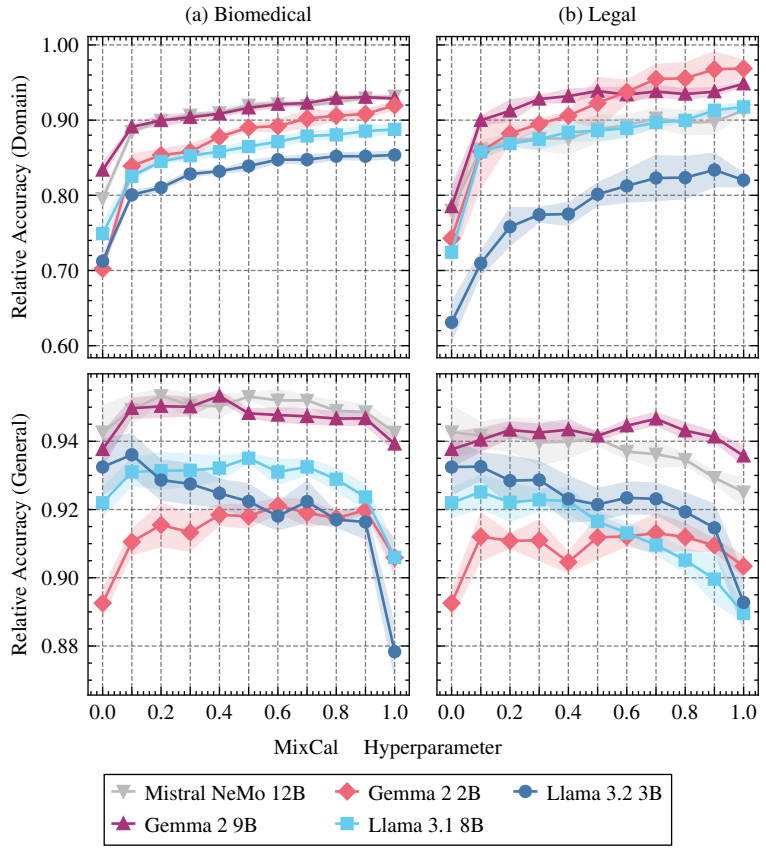

Figure 9: Benchmark accuracy relative to the original model when varying the MixCal α hyperparameter for pruning. Standard deviation is denoted by the shaded regions.

Method	Hyperparameter	Value
D-Pruner	Loss Regularization	{0.001, 0.01, 0.1}
	Group Size	{None, 128}
	Learning Rate	0.03
GPTQ-M	Bits per Weight	4
	Dampening	0.01
	Group Size	128
	Symmetric Quantization	Yes
SparseGPT	Dampening	0.01
	Group Size	128
	Sparsity	{0.5, 2:4}
Wanda	Group Size	1
	Sparsity	{0.5, 2:4}

Table 2: Hyperparameters used for all compression methods evaluated in our experiments.

B Hyperparameters

Table 2 presents the hyperparameters of all the compression methods in our experiments. In general, we adopt the optimal hyperparameters used for each method in the original work. For completeness, we also present results for every tested D-Pruner hyperparameter combination in Table 9.

C Data & Processing

Table 3 shows the splits for the datasets used in our experiments, split by category. We also show examples of how the domain-specific tasks are formatted for zero-shot LM evaluation. These are presented in Tables 4 and 5 for the legal and biomedical domains, respectively.

In the majority of cases, we use the task datasets exactly as implemented by the EleutherAI LM Evaluation Harness (Gao et al., 2024). We highlight the exceptions where additional preprocessing was required, below:

- **CaseHOLD** and **ECtHR (Task A)**. We adopt the versions of these datasets as provided by the LexGLUE benchmark (Chalkidis et al., 2022). To enable evaluation in a zero-shot setting, we adapt the prompts from Chalkidis (2023).
- **ECtHR (Task A)**, **CAIL2018**, and **CMB**. We additionally filter examples with multiple labels from these datasets, following prior work towards adapting existing tasks for few-shot LM evaluation (Guha et al., 2023).
- **CAIL2018** and **CMB**. For Chinese-language evaluation tasks, we use the prompts presented in Table 6, Appendix C.
- **BillSum** and **BioLaySumm PLOS**. Due to the extensive size of the test split in these datasets, we follow Frantar et al. (2023) in using the first 256 examples to assess perplexity. We highlight that perplexity is a stable metric which can be assessed using only a small number of examples (Dettmers and Zettlemoyer, 2023).

Name	Train	Val.	Test
Biomedical			
BioLaySumm PLOS	24,773	1,376	256
CMB (Chinese)	231,902	228	9,325
PubMedQA	450	50	500
MedMCQA	182,822	4,183	4,183
MedQA	10,178	1,272	1,273
MMLU Anatomy			135
MMLU Clinical Knowledge			265
MMLU College Medicine			173
MMLU Medical Genetics			100
MMLU Professional Medicine			272
MMLU College Biology			144
Legal			
BillSum	18,949		256
CAIL2018 (Chinese)	110,905	14,147	27,484
MMLU International Law			121
MMLU Jurisprudence			108
MMLU Professional Law			1,534
LexGLUE CaseHOLD	45,000	3,900	3,600
LexGLUE ECtHR (Task A)	6,838	802	808
General			
ARC (Easy)	2,251	570	2,376
ARC (Challenge)	1,119	299	1,172
BoolQ	9,427	3,270	
HellaSwag	39,905	10,042	
LAMBADA (Standard)		4,869	5,153
OpenBookQA	4,957	500	500
PIQA	16,113	1,838	
RTE	2,490	277	
WinoGrande	40,398	1,267	
StoryCloze	360	1,511	

Table 3: Number of examples in each evaluation task.

D Infrastructure

We implement all experiments using PyTorch (Paszke et al., 2019) with the model implementations from Hugging Face Transformers (Wolf et al., 2020). We additionally use Hugging Face Datasets (Lhoest et al., 2021) for all dataset manipulation, including for tasks implemented via the EleutherAI LM Evaluation Harness (Gao et al., 2024). Finally, we conduct all experiments using a single NVIDIA A100 (SXM4 80GB) GPU.

E Complete Results

Table 7 presents the full benchmark results (accuracy and perplexity) corresponding to Figure 3. For D-Pruner, we present results using iterative blocking and a loss regularization of 0.001. Finally, in Table 10 we further decompose benchmark results into their constituent tasks for completeness.

Task	Input	Answer
CaseHOLD	*Given the following excerpt from a United States court opinion:* **Citing Text:** ... Warner-Lambert Co., 427 Mass. at 49 (“ [Confidential and proprietary business information may be entitled to protection, even if such information cannot claim trade secret protection”); see, e.g., Augat, Inc., 409 Mass. at 173 (<HOLDING>). “Matters of public knowledge or of general... *Given the following excerpt from a United States court opinion:* *Which one of the following options should replace the <HOLDING> placeholder?* **Holdings:** A. Recognizing that even if a plaintiff claims certain information constitutes trade secrets its claim may not depend on that determination. B. Holding that included among trade secrets employee may not appropriate from employer is certain information such as lists of customers. C. ...	Holding B
ECtHR Task A	*Given the following facts from a European Court of Human Rights (ECtHR) case:* **Articles:** 2. On 8 May 1996 the applicant was arrested in New York (USA) and placed in detention on the basis of a extradition request from the authorities of the Netherlands Antilles where ... 3. In a document dated 18 June 1996 bearing the applicant’s ... *Which article of the European Convention on Human Rights (ECHR) has been violated?* *A. Article 2* *B. Article 3* *C. Article 5* ... *K. None of the above*	Choice K
Jurisprudence	*The following are multiple choice questions (with answers) about jurisprudence:* **Question:** Which statement best explains the purpose of Hart’s distinction between ’being obliged’ and ’having an obligation’? **Choices:** A. It demonstrates the difference between the internal and the external aspect of a rule. B. It refutes the natural lawyer’ ... C. ...	Choice B
International / Professional Law	*The following are multiple choice questions (with answers) about professional / international law:* **Statement:** One afternoon, a pilot was flying a small airplane when it suddenly ran out of gas. As he was coming in for an emergency landing ... The attorney’s testimony is: **Choices:** A. admissible, because the... B. inadmissible because ... C. ...	Choice B

Table 4: Representative task examples from the legal benchmark. *Italicized* text denotes the prompt.

Task	Input	Answer
MMLU Tasks: Clinical Knowledge; College Medicine; College Biology; Professional Medicine; Anatomy; Medical Genetics.	*The following are multiple choice questions (with answers) about <TASK>.* **Question:** What size of cannula would you use in a patient who needed a rapid blood transfusion (as of 2020 medical knowledge)? **Choices:** A. 18 gauge B. 20 gauge C. ...	Choice A
PubMedQA	*Abstract* To evaluate the degree to which histologic chorioamnionitis, a frequent finding in placentas submitted for histopathologic evaluation, correlates with clinical indicators of infection in the mother. A retrospective review was performed on 52 cases with a histologic diagnosis of acute chorioamnionitis from 2,051 ... *Question* Does histologic chorioamnionitis correspond to clinical chorioamnionitis? *Choices:* A. Yes B. No C. Maybe	Choice A - Yes
MedMCQA	*Question:* All of the following are surgical options for morbid obesity except -: *Choices:* A. Adjustable gastric banding B. Biliopancreatic diversion C. Duodenal Switch D. ...	Choice D
MedQA	*Question:* A 5-year-old girl is brought to the clinic by her mother for excessive hair growth. Her mother reports that for the past 2 months she has noticed ... studies demonstrates an elevated level of estrogen. What is the most likely diagnosis? *Choices:* A. Granulosa cell tumor B. Idiopathic precocious puberty C. ...	Choice A

Table 5: Representative task examples from the biomedical benchmark. *Italicized* text denotes the prompt.

Task	Input	Answer
CAIL2018	根据以下法律案件的事实: <FACT> 请问中华人民共和国刑法中的哪一条适用于本案? A. 第 <ARTICLE 1> 条 B. 第 <ARTICLE 2> 条 C. 第 <ARTICLE 3> 条 D. 第 <ARTICLE 4> 条 答案:	Choice A
CMB	问题: <QUESTION> A. <OPTION 1> B. <OPTION 2> C. <OPTION 3> D. <OPTION 4> E. <OPTION 5> 答案:	Choice A

Table 6: Example format of tasks in Chinese-language evaluation (CAIL2018 for legal and CMB for biomedical).

Model	Method	Target Domain	General		Legal		Biomedical	
			Accuracy	Perplexity	Accuracy	Perplexity	Accuracy	Perplexity
Llama 3.2 3B	-	-	$61.6_{0.0}$	$9.3_{0.0}$	$45.5_{0.0}$	$5.2_{0.0}$	$53.3_{0.0}$	$9.4_{0.0}$
	Magnitude	-	$44.2_{0.0}$	$50.7_{0.0}$	$17.4_{0.0}$	$24.9_{0.0}$	$30.1_{0.0}$	$62.2_{0.0}$
	Wanda	-	$54.4_{0.2}$	$15.4_{0.1}$	$26.0_{0.2}$	$8.9_{0.0}$	$35.5_{0.1}$	$15.5_{0.0}$
	SparseGPT	-	$\mathbf{57.5}_{0.4}$	$13.8_{0.1}$	$28.6_{0.7}$	$8.2_{0.1}$	$38.2_{1.1}$	$14.7_{0.1}$
	D-Pruner	Legal	$52.2_{0.2}$	$16.0_{0.1}$	$27.8_{0.6}$	$8.2_{0.0}$	-	-
		Biomedical	$52.1_{0.1}$	$19.0_{0.2}$	-	-	$39.0_{0.6}$	$16.3_{0.1}$
	SparseGPT + MixCal	Legal	$56.8_{0.3}$	$13.6_{0.0}$	$\mathbf{36.5}_{0.7}$	$\mathbf{7.7}_{0.2}$	-	-
		Biomedical	$56.8_{0.3}$	$\mathbf{13.3}_{0.0}$	-	-	$\mathbf{44.7}_{0.4}$	$\mathbf{13.5}_{0.1}$
Llama 3.1 8B	-	-	$67.8_{0.0}$	$7.3_{0.0}$	$55.1_{0.0}$	$4.2_{0.0}$	$60.8_{0.0}$	$8.0_{0.0}$
	Magnitude	-	$47.6_{0.0}$	$57.7_{0.0}$	$20.9_{0.0}$	$65.7_{0.0}$	$35.4_{0.0}$	$55.7_{0.0}$
	Wanda	-	$60.6_{0.1}$	$11.7_{0.0}$	$35.0_{0.3}$	$6.8_{0.0}$	$42.4_{0.2}$	$11.9_{0.0}$
	SparseGPT	-	$62.6_{0.2}$	$10.7_{0.1}$	$40.1_{0.6}$	$6.3_{0.1}$	$45.8_{1.4}$	$11.5_{0.1}$
	D-Pruner	Legal	$57.2_{0.3}$	$12.6_{0.2}$	$42.5_{1.8}$	$6.7_{0.0}$	-	-
		Biomedical	$59.1_{0.2}$	$13.5_{0.1}$	-	-	$49.2_{0.4}$	$14.7_{0.3}$
	SparseGPT + MixCal	Legal	$62.2_{0.2}$	$10.7_{0.0}$	$\mathbf{48.8}_{0.8}$	$\mathbf{6.0}_{0.1}$	-	-
		Biomedical	$\mathbf{63.4}_{0.1}$	$\mathbf{10.5}_{0.0}$	-	-	$\mathbf{52.6}_{0.4}$	$\mathbf{10.7}_{0.0}$
Gemma 2 2B	-	-	$63.3_{0.0}$	$13.1_{0.0}$	$27.1_{0.0}$	$6.2_{0.0}$	$44.2_{0.0}$	$15.0_{0.0}$
	Magnitude	-	$48.2_{0.0}$	$172.5_{0.0}$	$21.7_{0.0}$	$34.7_{0.0}$	$26.8_{0.0}$	$825.8_{0.0}$
	Wanda	-	$54.2_{0.2}$	$25.0_{0.2}$	$20.1_{1.5}$	$10.7_{0.0}$	$29.6_{0.7}$	$33.0_{0.5}$
	SparseGPT	-	$56.6_{0.2}$	$20.9_{0.4}$	$20.8_{1.4}$	$9.2_{0.1}$	$30.4_{1.3}$	$26.6_{0.3}$
	SparseGPT + MixCal	Legal	$57.7_{0.4}$	$\mathbf{18.9}_{0.1}$	$\mathbf{25.0}_{0.9}$	$\mathbf{8.3}_{0.0}$	-	-
		Biomedical	$\mathbf{58.1}_{0.2}$	$19.1_{0.1}$	-	-	$\mathbf{39.4}_{0.2}$	$\mathbf{21.5}_{0.2}$
Gemma 2 9B	-	-	$70.2_{0.0}$	$10.6_{0.0}$	$56.2_{0.0}$	$4.8_{0.0}$	$62.8_{0.0}$	$12.0_{0.0}$
	Magnitude	-	$60.6_{0.0}$	$33.5_{0.0}$	$33.7_{0.0}$	$11.0_{0.0}$	$47.8_{0.0}$	$57.4_{0.0}$
	Wanda	-	$63.5_{0.2}$	$16.6_{0.1}$	$41.8_{1.2}$	$6.7_{0.0}$	$48.8_{0.4}$	$20.8_{0.2}$
	SparseGPT	-	$65.8_{0.2}$	$15.2_{0.2}$	$43.7_{0.8}$	$6.4_{0.0}$	$52.8_{0.5}$	$19.3_{0.2}$
	SparseGPT + MixCal	Legal	$66.1_{0.1}$	$14.6_{0.0}$	$\mathbf{52.8}_{0.3}$	$\mathbf{5.9}_{0.0}$	-	-
		Biomedical	$\mathbf{66.5}_{0.1}$	$\mathbf{14.5}_{0.1}$	-	-	$\mathbf{57.6}_{0.4}$	$\mathbf{16.8}_{0.1}$
Mistral NeMo 12B	-	-	$69.4_{0.0}$	$7.1_{0.0}$	$57.5_{0.0}$	$4.3_{0.0}$	$58.4_{0.0}$	$7.6_{0.0}$
	Magnitude	-	$41.8_{0.0}$	$465.5_{0.0}$	$24.5_{0.0}$	$3.6\times10^{3}{}_{0.0}$	$35.0_{0.0}$	$10.3\times10^{3}{}_{0.0}$
	Wanda	-	$63.2_{0.2}$	$10.3_{0.0}$	$42.3_{0.8}$	$6.0_{0.0}$	$45.3_{0.3}$	$10.6_{0.0}$
	SparseGPT	-	$65.4_{0.4}$	$\mathbf{9.4}_{0.0}$	$44.5_{1.8}$	$5.6_{0.0}$	$46.4_{0.6}$	$9.8_{0.0}$
	SparseGPT + MixCal	Legal	$65.3_{0.2}$	$9.5_{0.0}$	$\mathbf{51.1}_{1.1}$	$\mathbf{5.5}_{0.0}$	-	-
		Biomedical	$\mathbf{66.2}_{0.1}$	$\mathbf{9.4}_{0.0}$	-	-	$\mathbf{53.7}_{0.4}$	$\mathbf{9.6}_{0.0}$

Table 7: Average performance for the general and domain-specific benchmarks when pruning with 50% sparsity. Standard deviations are shown as subscripts. **Bold** values denote the best performing method for each model. For reference, the top row of each model shows the original model performance (i.e. prior to compression).

Model	Method	Target Domain	General		Legal		Biomedical	
			Accuracy	Perplexity	Accuracy	Perplexity	Accuracy	Perplexity
Llama 3.2 3B	-	-	61.6 0.0	9.3 0.0	45.5 0.0	5.2 0.0	53.3 0.0	9.4 0.0
	SparseGPT	-	49.6 0.2	**22.6** 0.2	21.7 1.0	14.6 0.2	27.3 0.4	25.7 0.2
	SparseGPT + MixCal	Legal	49.1 0.1	23.3 0.1	**29.9** 1.1	**12.7** 0.2	-	-
		Biomedical	**50.1** 0.1	22.8 0.1	-	-	**37.6** 0.5	**23.0** 0.3
	SparseGPT + GPTQ-M	-	**47.6** 0.4	**24.5** 0.4	20.7 0.7	16.8 0.5	27.7 1.0	28.6 0.2
	SparseGPT + GPTQ-M + MixCal	Legal	46.0 0.6	25.2 0.3	**28.9** 2.5	**14.0** 0.3	-	-
		Biomedical	46.1 0.4	24.9 0.3	-	-	**35.6** 0.4	**25.3** 0.4
Llama 3.1 8B	-	-	67.8 0.0	7.3 0.0	55.1 0.0	4.2 0.0	60.8 0.0	8.0 0.0
	SparseGPT	-	54.5 0.2	17.9 0.1	22.1 1.2	10.7 0.2	34.8 0.3	19.3 0.2
	SparseGPT + MixCal	Legal	53.6 0.1	18.2 0.2	**41.5** 0.8	**9.6** 0.1	-	-
		Biomedical	**54.9** 0.5	**17.6** 0.1	-	-	**42.4** 0.7	**17.0** 0.2
	SparseGPT + GPTQ-M	-	**54.2** 0.3	19.3 0.6	20.7 1.4	12.0 0.2	33.9 0.5	20.9 0.2
	SparseGPT + GPTQ-M + MixCal	Legal	51.9 0.1	19.4 0.2	**39.5** 0.6	**10.8** 0.2	-	-
		Biomedical	52.8 0.3	**19.0** 0.2	-	-	**39.6** 1.9	**18.2** 0.3
Gemma 2 2B	-	-	63.3 0.0	13.1 0.0	27.1 0.0	6.2 0.0	44.2 0.0	15.0 0.0
	SparseGPT	-	47.1 0.6	40.0 1.7	16.3 0.0	17.1 0.7	30.4 0.8	57.9 2.4
	SparseGPT + MixCal	Legal	48.2 0.3	**34.4** 0.4	**21.4** 0.5	**13.0** 0.1	-	-
		Biomedical	**49.2** 0.4	36.2 0.4	-	-	**32.0** 0.8	**40.7** 0.8
	SparseGPT + GPTQ-M	-	45.8 0.5	45.2 2.0	16.3 0.1	19.2 0.8	30.4 2.1	68.3 5.8
	SparseGPT + GPTQ-M + MixCal	Legal	46.9 0.2	**37.7** 0.4	**19.4** 0.7	**14.5** 0.2	-	-
		Biomedical	**47.6** 0.2	40.1 0.5	-	-	**32.5** 0.3	**45.0** 1.7
Gemma 2 9B	-	-	70.2 0.0	10.6 0.0	56.2 0.0	4.8 0.0	62.8 0.0	12.0 0.0
	SparseGPT	-	56.9 1.0	22.2 0.5	32.1 2.9	9.6 0.1	38.9 0.9	30.0 0.7
	SparseGPT + MixCal	Legal	57.6 0.2	**21.1** 0.1	**44.6** 1.1	**7.8** 0.0	-	-
		Biomedical	**59.3** 0.4	21.7 0.1	-	-	**49.2** 0.6	**23.6** 0.1
	SparseGPT + GPTQ-M	-	56.1 0.4	24.3 0.4	27.5 2.8	10.4 0.1	36.5 1.4	33.8 0.6
	SparseGPT + GPTQ-M + MixCal	Legal	56.4 0.5	**22.9** 0.1	**44.2** 1.5	**8.2** 0.1	-	-
		Biomedical	**58.5** 0.6	23.5 0.1	-	-	**48.9** 1.0	**26.1** 0.2
Mistral NeMo 12B	-	-	69.4 0.0	7.1 0.0	57.5 0.0	4.3 0.0	58.4 0.0	7.6 0.0
	SparseGPT	-	**57.4** 0.4	**15.6** 0.1	24.9 1.4	9.1 0.1	33.1 1.2	16.2 0.1
	SparseGPT + MixCal	Legal	56.6 0.2	16.3 0.1	**38.0** 0.5	**8.2** 0.1	-	-
		Biomedical	**57.4** 0.1	16.0 0.1	-	-	**43.7** 0.3	**15.1** 0.1
	SparseGPT + GPTQ-M	-	**56.1** 0.2	**17.3** 0.3	22.2 1.5	10.2 0.2	32.0 2.3	17.9 0.3
	SparseGPT + GPTQ-M + MixCal	Legal	54.9 0.3	18.4 0.2	**35.2** 2.2	**9.2** 0.2	-	-
		Biomedical	56.0 0.1	18.1 0.5	-	-	**41.8** 0.6	**16.6** 0.3

Table 8: Average performance for the general and domain-specific benchmarks when pruning with 2:4 sparsity. Standard deviations are shown as subscripts. Rows using the GPTQ-M method additionally employ 4-bit quantization. **Bold** values denote the best performing method per model for (1) pruning, and (2) joint pruning and quantization. For reference, the top row of each model shows the original model performance (i.e. prior to compression).

Group Size	Model	Target Domain	λ_g	General Accuracy	General Perplexity	Legal Accuracy	Legal Perplexity	Biomedical Accuracy	Biomedical Perplexity
None	Llama 3.2 3B	Legal	0.1	**52.4**$_{0.2}$	16.0$_{0.0}$	27.3$_{0.4}$	8.2$_{0.0}$	-	-
		Biomedical	0.1	50.9$_{0.6}$	18.0$_{0.0}$	-	-	38.7$_{1.3}$	16.1$_{0.1}$
		Legal	0.01	52.2$_{0.4}$	16.0$_{0.0}$	27.3$_{0.5}$	8.2$_{0.0}$	-	-
		Biomedical	0.01	51.0$_{0.6}$	17.9$_{0.1}$	-	-	**39.2**$_{0.9}$	16.2$_{0.1}$
		Legal	0.001	52.2$_{0.2}$	16.0$_{0.1}$	**27.8**$_{0.6}$	8.2$_{0.0}$	-	-
		Biomedical	0.001	50.9$_{0.5}$	17.8$_{0.1}$	-	-	38.9$_{1.1}$	16.1$_{0.1}$
	Llama 3.1 8B	Legal	0.1	57.0$_{0.3}$	12.5$_{0.1}$	42.2$_{2.5}$	6.6$_{0.0}$	-	-
		Biomedical	0.1	57.6$_{0.7}$	14.0$_{0.4}$	-	-	47.4$_{0.3}$	15.0$_{0.5}$
		Legal	0.01	57.2$_{0.3}$	12.6$_{0.2}$	**42.5**$_{1.8}$	6.7$_{0.0}$	-	-
		Biomedical	0.01	58.8$_{0.2}$	13.3$_{0.1}$	-	-	48.8$_{0.8}$	15.1$_{0.4}$
		Legal	0.001	57.0$_{0.4}$	12.6$_{0.2}$	41.7$_{2.2}$	6.7$_{0.1}$	-	-
		Biomedical	0.001	57.7$_{1.2}$	14.2$_{1.0}$	-	-	47.5$_{1.8}$	15.5$_{0.5}$
128	Llama 3.2 3B	Legal	0.1	52.0$_{0.2}$	16.7$_{0.1}$	27.3$_{0.7}$	8.2$_{0.0}$	-	-
		Biomedical	0.1	52.0$_{0.2}$	18.9$_{0.2}$	-	-	**39.2**$_{0.7}$	16.3$_{0.1}$
		Legal	0.01	52.0$_{0.3}$	16.6$_{0.1}$	27.4$_{0.6}$	8.2$_{0.0}$	-	-
		Biomedical	0.01	**52.1**$_{0.1}$	19.0$_{0.2}$	-	-	39.0$_{0.6}$	16.3$_{0.1}$
		Legal	0.001	52.1$_{0.1}$	16.6$_{0.1}$	27.2$_{0.6}$	8.2$_{0.0}$	-	-
		Biomedical	0.001	52.0$_{0.3}$	18.9$_{0.2}$	-	-	**39.2**$_{0.5}$	16.2$_{0.1}$
	Llama 3.1 8B	Legal	0.1	57.3$_{0.3}$	12.9$_{0.1}$	42.0$_{2.0}$	6.6$_{0.0}$	-	-
		Biomedical	0.1	58.1$_{0.8}$	14.2$_{0.4}$	-	-	47.8$_{0.8}$	14.7$_{0.5}$
		Legal	0.01	**57.4**$_{0.3}$	12.9$_{0.2}$	42.1$_{1.3}$	6.6$_{0.0}$	-	-
		Biomedical	0.01	**59.1**$_{0.2}$	13.5$_{0.1}$	-	-	**49.2**$_{0.4}$	14.7$_{0.3}$
		Legal	0.001	57.1$_{0.3}$	13.0$_{0.2}$	40.6$_{2.6}$	6.7$_{0.1}$	-	-
		Biomedical	0.001	58.1$_{1.2}$	14.3$_{0.9}$	-	-	47.8$_{1.7}$	15.1$_{0.6}$

Table 9: Average D-Pruner performance for general and domain-specific benchmarks. We vary the group size (i.e. iterative blocking) and regularization (λ_g) hyperparameters. Standard deviations are shown as subscripts. **Bold** values denote the hyperparameter combination with the highest accuracy for each model and target domain.

Model	Method	Biomedical MedMCQA	Biomedical MedQA (4)	Biomedical PubMedQA	Biomedical Bio. MMLU	Legal CaseHOLD	Legal ECtHR	Legal Legal MMLU
Llama 3.2 3B	-	49.5$_{0.0}$	51.5$_{0.0}$	72.8$_{0.0}$	61.1$_{0.0}$	42.7$_{0.0}$	49.6$_{0.0}$	44.2$_{0.0}$
	Magnitude	28.8$_{0.0}$	28.8$_{0.0}$	50.6$_{0.0}$	27.5$_{0.0}$	20.6$_{0.0}$	4.5$_{0.0}$	27.2$_{0.0}$
	Wanda	30.4$_{0.2}$	35.6$_{0.6}$	63.6$_{0.5}$	41.8$_{0.3}$	28.9$_{0.7}$	13.7$_{0.7}$	35.4$_{0.4}$
	SparseGPT	33.6$_{1.9}$	35.4$_{1.4}$	69.5$_{1.2}$	45.1$_{1.2}$	32.9$_{0.6}$	15.8$_{1.6}$	37.1$_{0.5}$
	D-Pruner	35.3$_{0.8}$	39.2$_{0.3}$	66.8$_{1.2}$	41.5$_{0.6}$	22.9$_{1.3}$	25.6$_{0.1}$	33.1$_{0.7}$
	SparseGPT + MixCal	**41.1**$_{0.4}$	**41.9**$_{0.8}$	**70.8**$_{0.8}$	**51.3**$_{1.1}$	**40.2**$_{1.3}$	**28.3**$_{3.5}$	**40.8**$_{0.8}$
Llama 3.1 8B	-	56.4$_{0.0}$	60.1$_{0.0}$	75.8$_{0.0}$	71.7$_{0.0}$	51.9$_{0.0}$	60.9$_{0.0}$	52.4$_{0.0}$
	Magnitude	32.4$_{0.0}$	34.2$_{0.0}$	59.6$_{0.0}$	37.0$_{0.0}$	20.4$_{0.0}$	10.0$_{0.0}$	32.2$_{0.0}$
	Wanda	38.5$_{0.3}$	38.0$_{0.9}$	66.8$_{0.5}$	51.3$_{0.6}$	29.4$_{0.7}$	35.9$_{0.3}$	39.7$_{0.3}$
	SparseGPT	41.0$_{1.8}$	43.2$_{1.0}$	70.7$_{1.0}$	55.7$_{1.1}$	38.5$_{1.5}$	38.0$_{1.5}$	43.9$_{0.5}$
	D-Pruner	43.4$_{1.5}$	45.1$_{2.1}$	69.9$_{2.8}$	57.5$_{2.0}$	37.1$_{2.4}$	41.4$_{6.6}$	43.4$_{0.8}$
	SparseGPT + MixCal	**48.3**$_{0.5}$	**49.2**$_{0.8}$	**72.3**$_{0.9}$	**63.4**$_{0.6}$	**45.6**$_{0.5}$	**54.2**$_{1.4}$	**46.6**$_{1.0}$
Gemma 2 2B	-	40.9$_{0.0}$	35.3$_{0.0}$	74.0$_{0.0}$	53.8$_{0.0}$	32.5$_{0.0}$	8.5$_{0.0}$	40.4$_{0.0}$
	Magnitude	22.5$_{0.0}$	24.1$_{0.0}$	56.4$_{0.0}$	32.9$_{0.0}$	21.2$_{0.0}$	**17.9**$_{0.0}$	25.9$_{0.0}$
	Wanda	25.8$_{0.9}$	24.4$_{0.9}$	57.4$_{0.6}$	37.6$_{1.0}$	19.8$_{0.1}$	12.2$_{4.8}$	28.3$_{0.6}$
	SparseGPT	26.0$_{2.1}$	27.7$_{2.0}$	60.4$_{1.9}$	36.9$_{1.2}$	24.0$_{2.1}$	6.2$_{2.4}$	32.3$_{1.7}$
	SparseGPT + MixCal	**36.6**$_{0.2}$	**33.4**$_{1.1}$	**67.0**$_{1.1}$	**45.8**$_{0.6}$	**32.2**$_{2.2}$	5.1$_{1.1}$	**37.8**$_{0.3}$
Gemma 2 9B	-	57.9$_{0.0}$	60.5$_{0.0}$	78.6$_{0.0}$	77.2$_{0.0}$	51.7$_{0.0}$	60.1$_{0.0}$	56.8$_{0.0}$
	Magnitude	43.7$_{0.0}$	46.3$_{0.0}$	71.8$_{0.0}$	54.5$_{0.0}$	37.1$_{0.0}$	19.9$_{0.0}$	44.1$_{0.0}$
	Wanda	45.0$_{0.5}$	44.7$_{1.5}$	71.4$_{0.7}$	57.7$_{0.7}$	43.6$_{1.6}$	36.1$_{2.1}$	45.8$_{1.0}$
	SparseGPT	48.0$_{0.5}$	48.6$_{0.9}$	74.4$_{1.1}$	66.1$_{0.4}$	45.8$_{1.5}$	36.2$_{0.9}$	49.2$_{0.9}$
	SparseGPT + MixCal	**53.0**$_{0.4}$	**54.1**$_{1.1}$	**76.4**$_{0.9}$	**70.5**$_{0.2}$	**49.8**$_{0.1}$	**55.0**$_{0.4}$	**53.6**$_{0.5}$
Mistral NeMo 12B	-	52.4$_{0.0}$	60.6$_{0.0}$	74.4$_{0.0}$	71.9$_{0.0}$	55.8$_{0.0}$	62.0$_{0.0}$	54.8$_{0.0}$
	Magnitude	30.0$_{0.0}$	33.0$_{0.0}$	67.0$_{0.0}$	42.3$_{0.0}$	25.8$_{0.0}$	9.8$_{0.0}$	37.9$_{0.0}$
	Wanda	39.7$_{0.4}$	45.5$_{0.3}$	60.6$_{0.1}$	59.7$_{0.4}$	44.0$_{0.5}$	37.4$_{2.8}$	45.4$_{0.5}$
	SparseGPT	40.0$_{0.5}$	46.5$_{0.9}$	72.0$_{0.7}$	59.2$_{1.5}$	40.5$_{3.1}$	49.3$_{3.4}$	43.8$_{0.9}$
	SparseGPT + MixCal	**47.7**$_{0.7}$	**55.3**$_{0.4}$	**73.2**$_{0.5}$	**65.2**$_{0.7}$	**48.3**$_{2.8}$	**55.1**$_{1.1}$	**49.9**$_{0.5}$

Table 10: Average performance across calibration sets for tasks from the biomedical and legal domains. Standard deviations are shown as subscripts. **Bold** values denote the best performing method for each task.

Model	Method	Target Domain	General	Legal	Biomedical
Llama 3.2 3B	SparseGPT	-	13.8$_{0.1}$	8.2$_{0.1}$	14.7$_{0.1}$
	+ In-domain data	Legal	13.7$_{0.0}$	**7.6**$_{0.1}$	-
		Biomedical	13.5$_{0.0}$	-	13.7$_{0.1}$
	+ MixCal (Ours)	Legal	**13.6**$_{0.0}$	7.7$_{0.2}$	-
		Biomedical	**13.3**$_{0.0}$	-	**13.5**$_{0.1}$
Llama 3.1 8B	SparseGPT	-	**10.7**$_{0.1}$	6.3$_{0.1}$	11.5$_{0.1}$
	+ In-domain data	Legal	10.8$_{0.0}$	**6.0**$_{0.0}$	-
		Biomedical	10.7$_{0.0}$	-	10.9$_{0.0}$
	+ MixCal (Ours)	Legal	**10.7**$_{0.0}$	**6.0**$_{0.1}$	-
		Biomedical	**10.5**$_{0.0}$	-	**10.7**$_{0.0}$
Gemma 2 2B	SparseGPT	-	20.8$_{2.0}$	9.3$_{0.8}$	24.3$_{1.8}$
	+ In-domain data	Legal	19.9$_{0.1}$	**8.3**$_{0.0}$	-
		Biomedical	20.5$_{0.1}$	-	23.5$_{0.2}$
	+ MixCal (Ours)	Legal	**18.9**$_{0.1}$	**8.3**$_{0.0}$	-
		Biomedical	**19.1**$_{0.1}$	-	**21.5**$_{0.2}$
Gemma 2 9B	SparseGPT	-	15.2$_{0.2}$	6.4$_{0.0}$	19.3$_{0.2}$
	+ In-domain data	Legal	14.7$_{0.0}$	**5.9**$_{0.0}$	-
		Biomedical	14.9$_{0.0}$	-	17.5$_{0.1}$
	+ MixCal (Ours)	Legal	**14.6**$_{0.0}$	**5.9**$_{0.0}$	-
		Biomedical	**14.5**$_{0.1}$	-	**16.8**$_{0.1}$
Mistral NeMo 12B	SparseGPT	-	**9.4**$_{0.0}$	5.6$_{0.0}$	9.8$_{0.0}$
	+ In-domain data	Legal	9.5$_{0.0}$	**5.5**$_{0.0}$	-
		Biomedical	9.5$_{0.0}$	-	9.7$_{0.0}$
	+ MixCal (Ours)	Legal	9.5$_{0.0}$	**5.5**$_{0.0}$	-
		Biomedical	**9.4**$_{0.0}$	-	**9.6**$_{0.0}$

Table 11: Average performance when pruning to 50% unstructured sparsity with (1) in-domain data, and (2) MixCal. Standard deviations are shown as subscripts. **Bold** values denote the best performing method for each model.