Universiteit Leiden
The Netherlands

Opleiding DSAI

Pruning-aware Shapley values estimation
for Large Language Models
nonuniform semistructured pruning

Anna Marini

Supervisors:
Qinyu Chen & Nusa Zidaric

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

01/07/2026

Abstract

As Large Language Models (LLMs) grow in size, so do their memory requirements and computational costs. The term pruning defines a series of techniques applied to neural networks that remove elements from a model (weights, layers) to reduce such costs. A specific method, called 2:4 pruning, mandatorily removes 50% of the weights per model matrix, according to a pattern. While this brings advantages in terms of compression, this method's strictness can noticeably harm the model's abilities.

To tame the negative effects of 2:4 pruning, we propose a novel approach to the application of nonuniform 2:4 sparsity to LLMs. In most cases, pruning targeted at weights is applied uniformly to all layers in a model; nonuniform pruning instead treats layers differently, preserving the ones estimated to matter most. The approach proposed here fully prunes some layers with 2:4 pruning, leaving others untouched. The layers to be preserved are selected using the nonuniform pruning framework SV-NUP [SYCL25], a framework initially targeted at a different pruning strategy (unstructured pruning) which we adapt to 2:4 sparsity. SV-NUP (Shapley Values-based Nonuniform Pruning) uses Shapley values, a cooperative game theory concept, to attribute importance scores to different layers. We introduce two extensions to the framework, which directly target how the Shapley values are computed. First, pruning awareness incorporates information about the target pruning strategy into the Shapley value computation. Second, finer layer granularity assigns Shapley value scores to smaller subsystems than SV-NUP, which treats decoder blocks (the sequential macro components that constitute LLMs) as the smallest atomic element in score attribution. We then investigate how the resulting Shapley values influence the pruning order and performance degradation. Overall, we report that the nonuniform semistructured framework detects critical layers and that, as demonstrated through successive pruning experiments, their preservation from pruning greatly reduces performance degradation, compared to a fully pruned model. In particular, we report that the preservation of two decoder block layers (out of 32) reduces the perplexity of Llama 2 7B pruned with magnitude by around two-thirds. The pruning-aware variation we introduced consistently outperforms the original SV-NUP framework at the task, proposing pruning orders that better pinpoint the crucial layers based on the pruning methods applied underneath. The application of a finer granularity, instead, does not produce particular improvement: when applied to the original method, it reduces effectiveness, while its pruning-aware variation shows no particular improvement, although it still enhances flexibility in the pruning ratio selection.

Contents

1 Introduction

Large Language Models (LLMs) occupy a prominent status within deep learning research, having achieved state-of-the-art results across a wide range of natural language processing (NLP) tasks. Most modern LLMs are built on a decoder-only transformer architecture [HSSL24] and contain billions of parameters. Their scale, however, results in substantial memory requirements and computational costs during inference [ZLL+24]. Model compression techniques address this issue by reducing storage requirements, potentially improving inference speed and energy consumption. Among these, pruning is one of the most widely adopted approaches [CZS24]. Pruning removes elements from a model, whether individual weights (fine-grained) or entire components, such as nodes and layers (coarse-grained or structured). Fine-grained pruning can be divided into two strategies, unstructured and semistructured pruning (sparsity). Unstructured pruning removes an arbitrary selection of weights from the matrices that constitute the models. Semistructured pruning divides the matrices into small stripes composed of m values; for every stripe, only n nonzero elements are preserved. It is thus also called pattern-based, or $n : m$ pruning. Unstructured pruning has a higher degree of freedom and, as observable in the experimental findings of Section 6, this consistently leads to less damage to the LLM's capabilities. On the other side, semistructured pruning, 2:4 pruning specifically, is directly supported by Sparse Tensor Cores [MLP+21], which exploit its fixed pattern to perform sparse matrix multiplication without first converting the data into a dense format, potentially reducing memory overhead and accelerating inference. This structural constraint, however, generally leads to greater degradation in language generation quality than unstructured pruning. One promising strategy to mitigate this effect is nonuniform pruning.
One promising strategy to mitigate this effect is nonuniform pruning. Conventionally, pruning strategies apply the same pruning ratio to every layer of the model (e.g., 50%), assuming that all layers contribute equally to the model's performance. Since multiple studies challenge this assumption [ZDK24][YZG+25], nonuniform strategies were created to assign different pruning ratios to different layers (see Section 5.1.1). Existing approaches to nonuniform semistructured pruning, however, achieve this flexibility by modifying the sparsity pattern [SZS+21] [HMD26]; these methods are therefore incompatible with the support provided by the NVIDIA ecosystem.
This thesis investigates an alternative form of nonuniform semistructured pruning, designed to remain fully compatible with existing hardware. Rather than modifying the 2:4 format pattern, we introduce nonuniformity at the layer level: a selection of layers are pruned using 2:4 sparsity, while others remain entirely dense. The resulting problem is no longer how much to prune within each layer, but rather which layers should be pruned. Zhang et al. [ZDK24] report the existence of a small subset of highly critical layers (cornerstone layers): we thus further investigate whether the preservation of such a subset can significantly preserve model performance.
To quantify layer importance, we adopt a Shapley value-based strategy [Sha53]. Shapley values are a concept brought from cooperative game theory, initially formulated to solve the fair contribution attribution problem. Besides the already cited Zhang et al. [ZDK24], they have shown promising results for quantifying layer importance in multiple nonuniform model compression frameworks [ZDH+25][SYCL25][DTD+26]. Our study builds directly on 'Efficient Shapley Value-based Non-Uniform Pruning of Large Language Models', abbreviated as SV-NUP [SYCL25]. This framework proposes a robust and efficient way to compute the Shapley values specifically for nonuniform pruning, called the SWSV (Sliding Window Shapley Value) approximator. Going beyond the traditional game theory setting, the SWSV approximator accounts for the sequential organization

of the model layers; we can thus describe it as a model-aware estimation approach.
In this paper, we extend the SWSV approximator with a pruning-aware approach. Previous nonuniform pruning studies based on Shapley values compute a layer's importance under the assumption that the layer is entirely removed. We instead take inspiration from CoopQ [ZDH+25], another Shapley value-based compression framework, which instead assumes the layer is preserved, but compressed, to propose a pruning-aware version of the SWSV approximator.
Another extension we propose concerns layer granularity. Previous Shapley values-based pruning studies use the term "layer" to refer to decoder blocks, sequential macro components that constitute the network (see Section 3.2). Zhang et al. [ZDK24] assign the values to the single decoder-block subsystems (attention and feed-forward network), and report significantly different values within the same decoder block. We thus investigate Shapley values computed separately for the attention and feed-forward sub-components within each decoder block.
In short, 2:4 pruning is a more aggressive compression method than unstructured pruning, due to its strictness. Since the nonuniform pruning criterion we propose still applies this fixed 2:4 ratio to each selected layer, with nonuniformity coming only from which layers are chosen for pruning, we explore several variations of Shapley-value estimation to verify that this criterion reliably identifies the layers most critical to preserve for a given pruning method.

2 Research questions

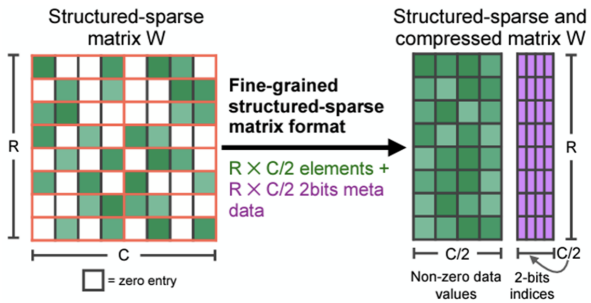
As anticipated, this study adapts SV-NUP [SYCL25] to semistructured sparsity, introducing some additional elements of novelty to the framework [ZDK24][ZDH+25]. To explore the consequent outcomes, we formulate the following research questions:

- Q1: Does the SWSV approximator identify a limited subset of significantly important layers? Does their exclusion from 2:4 pruning significantly improve the model performance?
- Q2: Do pruning-agnostic or pruning-aware Shapley values produce a better nonuniform, semistructured sparsity criterion?
- Q3: Does splitting Shapley values between the decoder block's attention and feed-forward network subsystems produce a better nonuniform, semistructured sparsity criterion?

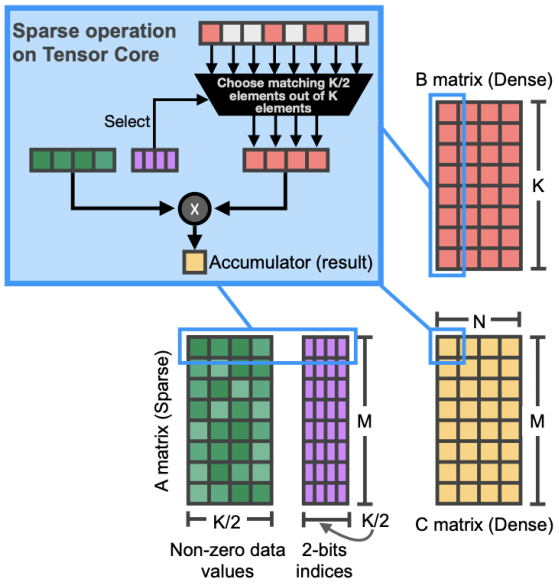
3 Preliminaries

3.1 2:4 sparsity

2:4 sparsity is a semistructured sparsity pattern that gained popularity after the launch of the NVIDIA Ampere architecture, which provides support for its computational speedup [MLP+21]. This section introduces the compression of matrices pruned with a 2:4 pattern and briefly describes how Sparse Tensor Cores exploit this sparsity scheme to perform efficient matrix multiplication [RPS21]. 2:4 sparsity helps bridge a significant gap in the field: software designers tend to favor unstructured sparsity for its smaller impact on model performance, while hardware designers favor structured sparsity patterns, since only these are efficiently accelerated on existing parallel hardware [JTB+25]. This creates a mismatch between the sparsity pattern that best preserves model quality and the one that hardware can actually exploit. 2:4 sparsity and Sparse Tensor Cores offer a practical middle ground, providing real compression and acceleration on NVIDIA GPUs.



(a) Compression of a 2:4 sparse matrix



(b) GEMM operation with 2:4 sparse matrix

Figure 1: 2:4 sparse matrices compression and acceleration with NVIDIA's Sparse Tensor Cores. In Figure 1a, the matrix is pruned and then compressed in two separate components: the preserved values (in green) and indexes representing their original placement (in purple). Figure 1b shows how this compressed representation is inherently supported by Sparse Tensor Cores to perform GEMM (General Matrix Multiplication). Image taken by [MLP+21].

3.1.1 Compression

2:4 sparsity can be used to compress a matrix, like the one displayed on the left in Figure 1a. In this case, the original matrix has dimensions $R \times C$. 2:4 pruning partitions the values into a series of horizontal stripes, each composed of 4 elements. Applying one of the many criteria presented in Section 4, a pruning technique removes two values, zeroing them. At that point, the matrix is

50% sparse, and it is possible to convert the initial format, where the zeroed values still occupy the original memory space, to a compressed one.
Fundamentally, the information is divided into values and indices. For each stripe, 2 values remain, stored contiguously in a $R \times (\frac{C}{2})$ matrix; this clips the horizontal dimension in half. No information about these remaining values is lost, as they keep their original data type (e.g., 16FP). To reconstruct the original matrix, only one other piece of information is needed: the position of the values in the original stripe. These indices always occupy only 2 bits, as these are sufficient to encode the four possible placements. This creates another $R \times (\frac{C}{2})$ matrix, which, however, occupies a fraction of the space. The final compression rate varies: the larger the data type, the more significant the effect. For example, a matrix cast to 16 floating-point corresponds to a compression rate of x1.78.

3.1.2 Acceleration

Sparse Tensor Cores were first introduced in NVIDIA GPUs with the Ampere microarchitecture. These components, like their dense counterparts (standard Tensor Cores), are specialized in GEMM, General Matrix Multiplication [NVI23]. This consists of the operation:

$$C = \alpha AB + \beta C \tag{1}$$

where A and B are matrices, α and β are scalar coefficients, and C is a pre-existing matrix (accumulator) which stores the intermediate results. In Sparse Tensor Cores, only A (corresponding to model weights in neural networks) is stored in the sparse compressed format. As can be seen in Figure 1b, B (input activations) and C (output activations) remain dense matrices.
From a theoretical perspective, 2:4 sparsity can at most double the speed of normal Tensor Cores. However, in practical settings, the actual acceleration of neural network inference is limited by several factors. First, many operations in the network are not GEMMs. For example, element-wise functions, such as the nonlinear activation ReLU, are not affected. Most importantly, sometimes the bottleneck is not constituted by the arithmetic intensity. The GPU can only take information at a certain rate, represented by the GPU memory bandwidth: any computational speedup will be nullified when this roofline is reached [WWP09].

3.2 Large Language Models

This section introduces the architecture of LLMs, introducing as examples Qwen 3 and Llama 3.2 [LT24][YLY+25]; their structure is summarized in Figure 2. The same models, and their larger counterparts, listed in Section 5.2.1, will be used in the experiments in Section 6. The vast majority of LLMs, such as the ones presented here, belong to the category of transformers. The transformer architecture was first introduced in 2017, with 'Attention is All You Need' [VSP+25].

3.2.1 Network structure

LLMs are trained to perform conditional generation: given a piece of text, they predict the next word or piece of word (token) that should follow. The models ingest information in the form of tokens, which can represent words or subwords. Each token is assigned a unique numeric ID, used to link it with the corresponding embedding vector, an initial representation that already contains semantic information, stored in the token embedding layer. The embedding then enters the main

body of the model, which consists of a series of decoder blocks (the colored section in the two figures). The final results are then passed to a linear output layer (also called a linear head), which, depending on the task, produces a probability distribution over which token should follow the ones already fed to the model. The embedding layer and the final linear layer constitute the link to the actual natural language vocabulary. They are generally not pruned using 2:4 sparsity, as reducing the embedding matrix or the linear head would directly remove or distort vocabulary representations, severely damaging model performance.

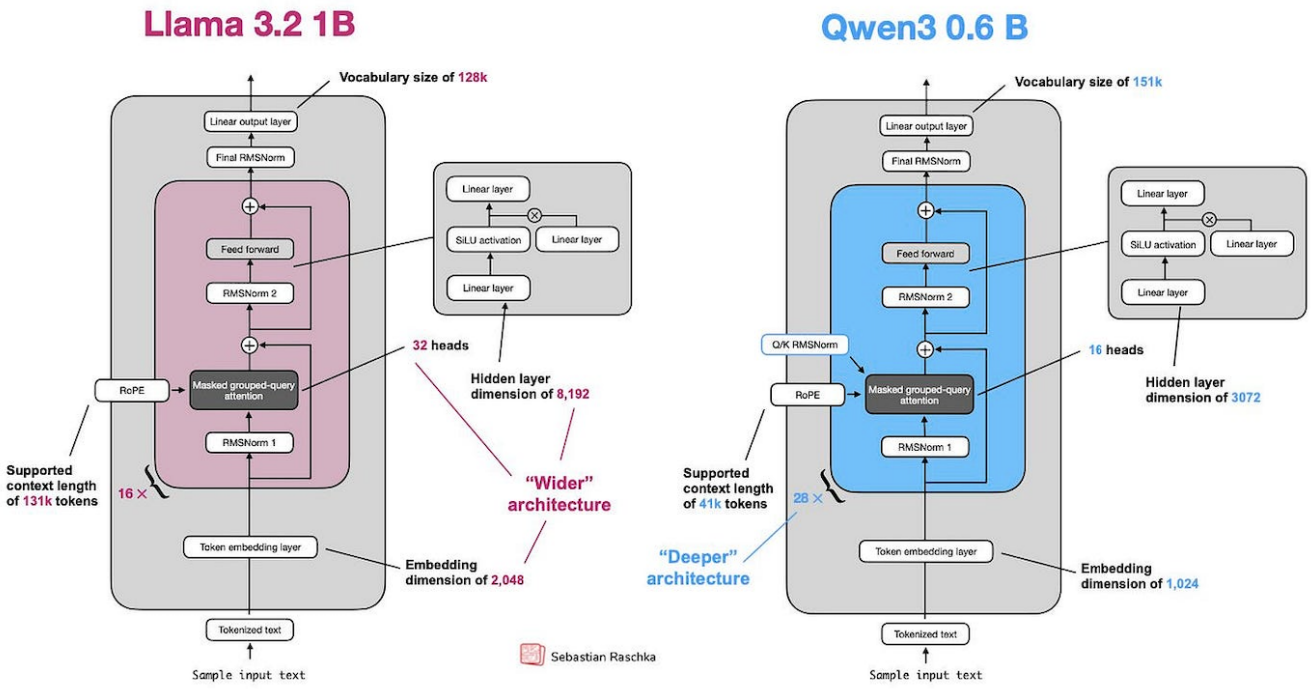


Figure 2: Qwen 0.6B and Llama 1B architectures. These models are both composed of the same main elements: an initial token embedding layer, a series of decoder blocks, and a final linear layer. They differ, however, in the number of decoder blocks and in the size of their hidden dimensions. Images taken by [Ras].

3.2.2 Decoder blocks

Each block consists of two subsystems: the masked grouped-query attention, here also called the attention mechanism, and the feed-forward or multilayer perceptron (MLP) network. The attention mechanism primarily enriches the original embedding with information on how the different tokens relate to each other. It is also aided by Rotary Positional Embeddings (RoPE), used to encode the relative position of the tokens inside the sentence (the relative distance from each other). The second portion of the decoder block is composed of a series of linear layers, interleaved with an activation function that introduces non-linearity. Attention mechanisms and feed-forward networks are interleaved with normalization operations (RMS Normalization), which continuously rescale the intermediate outputs of the blocks (activations) to stabilize activation distributions during training and inference, preventing numerical instability such as activation explosion or collapse. In the case

of the Qwen architecture, RMS normalization is also applied during the intermediate steps of the attention mechanism.
Another fundamental element is residual connections, or residual streams: these transmit the unchanged information alongside the block, making sure they can reach the deeper layers of the model. To formalize the concept, the two main operations can be represented like this:

$$\begin{aligned} \mathbf{O} &= \mathbf{X} + \text{MultiHeadAttention}(\text{RMSNorm}(\mathbf{X})) \\ \mathbf{H} &= \mathbf{O} + \text{FFN}(\text{RMSNorm}(\mathbf{O})) \end{aligned} \tag{2}$$

where $\mathbf{X}$ is the input of the block, $\mathbf{O}$ is the intermediate output of the attention mechanism, and $\mathbf{H}$ is the final output of the decoder block.
Inside the block, only attention mechanisms and feed-forward networks are composed of large matrices; the exact matrix dimensions vary across models and are listed in Table 1. RMS normalization only requires a scalar tunable parameter, and its element-wise operation is not processed by Tensor Cores. Thus, although we use the terms "decoder block pruning" or "model pruning" in this work and related literature, the actual pruning target is restricted to the attention and feed-forward subcomponents.

4 Related Work

Pruning LLMs There exist multiple strategies to reduce the costs of deep learning models, which fall under the umbrella term of model compression. Pruning was one of the first to be formulated, with the seminal papers of LeCun et al. [LDS89] and Hassibi et al. [HS93]. Beyond the granularity distinction introduced earlier (unstructured, semistructured, structured), pruning methods can also be classified by the timing of their application: before training (PBT), during training (PDT), or after training (PAT) [ZLL+24]. Instances of these variations are common in all pruning categories, including specific 2:4 sparsity examples [HZH+24]. However, LLMs are notoriously large, making training from scratch extremely expensive [YDC+25]; research on LLM pruning therefore focuses primarily on PAT techniques, applied to already pretrained models. After pruning, models are also often finetuned, using standard backpropagation [MLP+21], or more lightweight methods, such as LoRA [HSW+21], either to prime the model for a downstream task or to recover performance lost during pruning.

Activations outliers LLMs also present a specific challenge, absent in other models, called activation outliers or massive activations. Timkey and van Schijndel [TvS21] were the first to report how LLMs tend to present a small subset of activations, whose magnitude surpasses that of the other values produced by the same layer by 10 to 20 times [DLBZ22]. Disrupting these outliers severely damages performance, so compression methods generally aim to preserve them.

Pruning criteria Applying pruning in practice requires a criterion for selecting which weights to remove; many semistructured pruning criteria assign importance scores to weights, most of these originally created for unstructured pruning techniques. A ubiquitous option is magnitude pruning, which selects weights based solely on their absolute value [HPTD15]; another popular option, Wanda [SLBK24], instead incorporates the activation norm to assign importance scores to the weights to prune. SparseGPT [FA22] formulates pruning as a reconstruction problem, using second-order approximations to minimize reconstruction error. Wanda++ [YZG+25] is an extension of Wanda, natively built for 2:4 sparsity, which computes regional gradients, both to guide pruning decisions and to locally update the remaining weights after pruning.

Nonuniform pruning The same trend, techniques originally built for unstructured pruning being adapted to other settings, appears in nonuniform pruning. Nonuniform pruning has been widely studied for unstructured sparsity, where it consists of the nonuniform allocation of pruning ratios across layers. By default, sparsity is allocated on a layer basis, where each layer is pruned to the same rate (e.g., 40%, 50%). Restricting the ratio allocation decision to individual layers keeps the search space manageable and avoids the overhead of a global solution; Yang et al. [YDC+25] is a noteworthy exception, selecting weights globally, where nonuniformity across layers emerges as a natural consequence. In every other case discussed here, however, ratio allocation is a separate system added on top of a weight selection criterion, such as Wanda.
Nonuniform semistructured pruning specifically is comparatively less explored, but there still exist some precedents. DominoSearch [SZS+21] applies different patterns (e.g., 1:8, 5:8) to different layers, while PATCH [HMD26] divides matrices into patches, pruning some with 2:4 sparsity while others remain dense. However, neither of these frameworks benefits from standard NVIDIA support.

Taking into account the outlier phenomenon, some researchers used it as a criterion for nonuniform sparsity. This attempt led to OWL (Outlier-Weighted Layerwise Sparsity) [YWZ+25], which distributes lower pruning ratios to layers that present a higher occurrence of outliers. A similar study, although more exploratory, was conducted by [MMBR25] on outliers (in such paper referred to as 'spikes') in Llama models. This study did not define a systematic criterion, but through direct observation, reported how a few layers clearly exhibited more outliers, and how their preservation brought significant performance preservation.

Shapley values for nonuniform pruning Other than outlier presence, importance scores for nonuniform allocation have also been computed using Shapley values [Sha53], a concept from cooperative game theory that summarizes layer importance in a single score while accounting for inter-layer relationships. Unlike outlier-based criteria, Shapley values do not rely on a local metric, but average the contributions of individual layers based on overall model performance. 'Investigating Layer Importance in Large Language Models' [ZDK24] is one of the first pieces of evidence we found of Shapley values assigned on a layer basis, used in this case for purely exploratory purposes; the study reports that a small subset of three layers generally accounts for 29–37% of the model's performance. SV-NUP [SYCL25] instead employs a model-aware Shapley-value estimator directly to assign pruning ratios, consistently outperforming OWL at nonuniform ratio allocation. It was later followed by [DTD+26], a similarly Shapley-based pruning ratio allocation method which augments the estimation of the Shapley values with an additional neural network. Overall, the application of Shapley values to the problem has produced positive results in nonuniform pruning; the approach proposed here for 2:4 sparsity builds directly on this line of work.

Shapley values for nonuniform quantization Shapley values have also been applied beyond pruning, to a parallel line of nonuniform compression research: quantization [GKD+21]. Where pruning removes selected values entirely and leaves the rest untouched, quantization uniformly reduces the storage size of all values (e.g., FP16, FP8, INT4). CoopQ [ZDH+25] applies this idea nonuniformly and, notably, also uses Shapley values: it applies aggressive quantization to most layers, but reserves a slightly larger datatype for the layers identified as critical. [DTD+26] similarly applied Shapley values to assign different quantization types (INT8, INT4) to layers of an LLM. As anticipated, CoopQ introduced a significant variation to the Shapley values estimation method, compression awareness, which will be reproposed in this research.

5 Methodology

5.1 Framework

5.1.1 SV-NUP

As outlined in Section 1, the proposed experiments are primarily grounded in SV-NUP (Shapley values-based Nonuniform Pruning) [SYCL25]. Computing exact Shapley values is NP-hard, so practical implementations typically rely on approximation strategies. Among the different approaches, the SWSV approximator SV-NUP proposed appears particularly promising, because it incorporates information about the model itself to significantly speed up the estimation process, still achieving robust results. This section first introduces the theoretical foundation of Shapley values, translates their original formulation to this specific problem setting, and shows how the SWSV approximator exploits this additional context. It then introduces how pruning ratios are allocated across layers, and finally shows how Shapley values are translated into these per-layer pruning ratios.

Shapley values Shapley values are a cooperative (coalitional) game theory concept first introduced by Lloyd Shapley in "A Value for n-person Games" [Sha53]. In cooperative game theory, all players or a subset of them must form a team (a coalition) to play a game; once the players join a coalition, they fully cooperate to achieve a collective reward (payoff). When different players come together in a coalition, however, the effect of their joint participation is generally not just the sum of individual contributions, but the result of different interdependence relations among the players. The problem of fair contribution attribution arises: how to estimate how much each player contributes to the final payoff, independently of who they are playing with. The Shapley values represent the expected marginal contribution of a player across all coalitions, taking into account every possible interaction with other players. The formula is as follows:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left[v(S \cup \{i\}) - v(S) \right], \quad \forall i \in N \tag{3}$$

A Shapley value $\phi_i(v)$ represents the expected marginal contribution of player i. N represents the set of all players, while S represents any subset (coalition) of these players that does not include player i. The function $v : 2^N \to \mathbb{R}$ is the payoff generated by the considered coalition. The term $[v(S \cup \{i\}) - v(S)]$ then represents the marginal contribution of the player i to coalition S, as it computes the difference in payoff between the player participating in the coalition and not participating in it.
This marginal contribution is weighted by $\frac{|S|!\,(|N|-|S|-1)!}{|N|!}$. This weighting conceptually represents players joining the coalition in every possible order, to determine how much the arrival of player i contributes, given that the players in S have already joined. $|S|!$ represents all possible orders for players to join the coalition before i, and $(|N| - |S| - 1)!$ represents the possible orders that remain after i has joined. $|N|!$, the total number of possible orderings of all N players, normalizes the product of these two terms into a probability. This weight ensures that every possible ordering in which the players could join the coalition is given equal importance, regardless of the size of the coalition itself; without it, coalitions of intermediate sizes would be overrepresented in the sum.

Layers as players In a neural network, the final performance is not the sum of the individual layers' contributions, but results from the complex interrelations across them. Shapley values represent a sound way of disentangling these interactions and assigning each layer a score; to do so, we treat each layer as a player in a cooperative game.
As anticipated, in related literature, including SV-NUP, the term "layer" refers to a full decoder block. We adopt the same term throughout this section; later deviations will be explicitly pointed out. Similarly, we will use ℓ to denote the layer/player from this point onward, in place of the generic player index i used in the original formulation, while L will represent the complete set of layers/players. The selected payoff function v, inverse perplexity, is introduced below. Based on it, the Shapley value of a layer then quantifies its contribution to the model performance across all possible coalitions.
In this problem setting, a coalition represents a specific subset of layers that perform their normal computations. If a layer is excluded, then its internal computations are skipped entirely: the attention mechanism, FFN layers, and normalization steps are bypassed. This does not disconnect the surrounding layers from one another, however, since residual streams (see Formula 2) transmit information from one block to the next untouched, regardless of whether the block in between is active. Since the model is already partially trained to propagate information around skipped blocks, removing a decoder block is not necessarily catastrophic to its output.
To formulate the payoff function, SV-NUP and other nonuniform pruning studies [ZDH+25][DTD+26] use inverse perplexity $v = \frac{1}{PPL}$. Perplexity summarizes the model's ability to predict the next token in a given sentence, the fundamental task LLMs are pretrained on, and thus constitutes a task-agnostic metric of the model's language modeling capabilities. Given a tokenized sequence $x = (x_1, x_2, \ldots, x_T)$, perplexity measures the exponentiated average negative log-likelihood (cross-entropy) assigned by the model to each token based on its preceding context:

$$\mathrm{PPL}(x) = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log p_\theta(x_t \mid x_{<t})\right)$$

A lower perplexity indicates that the model assigns higher probability to the observed token sequence, reflecting better predictive accuracy in the autoregressive task. Computing perplexity, however, requires the model to perform inference. By definition, to compute the exact Shapley values for all the layers, the estimator should attempt all possible configurations of dense and removed layers, thus 2^N combinations. Once the overhead of model inference is taken into account, this approach becomes infeasible, even for small models. To circumvent this obstacle, SV-NUP introduces the Sliding Window Shapley Values (SWSV) approximator.

SWSV approximation Many methods have been developed to approximate Shapley values at a lower cost than exact estimation, relying in particular on random permutation sampling, to ensure no bias is introduced in the computation of the value. Still, such methods require a large pool of samples to avoid excessive instability [CGT09].
These techniques, however, do not take into account the specific characteristics of the application setting, which is model pruning. The intuition behind SWSV is that, while in a general case no particular information about player relations is known, neural network layers are connected sequentially. In a network, a removed or pruned layer is far more likely to impact its neighbors' performance than distant layers. Thus, SWSV defines a window around the block that is being

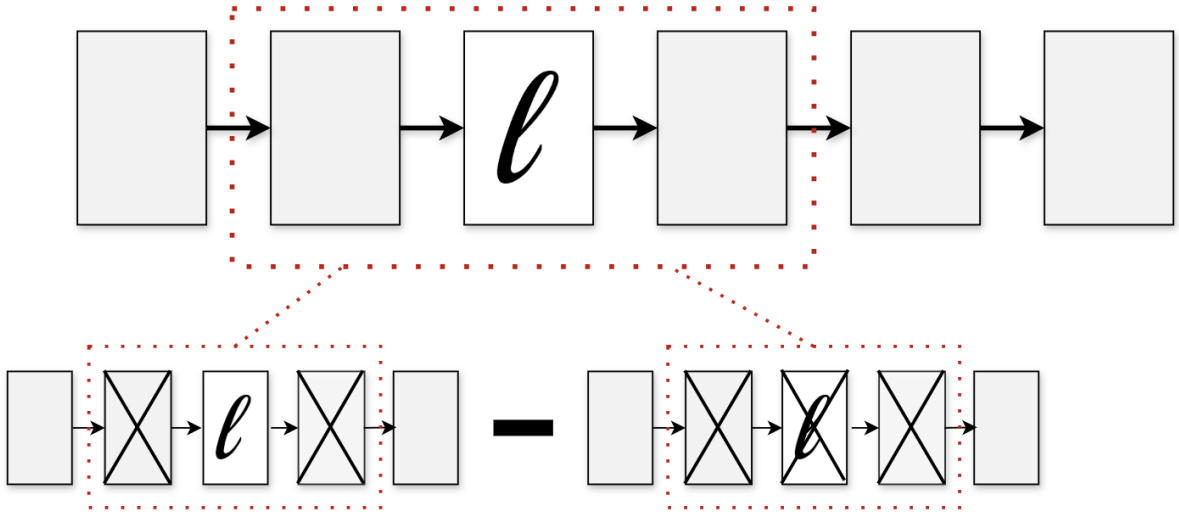


Figure 3: SWSV approximation. In this example, the method is applied to the third layer of the network, with a window of size $w = 3$. Inside this window, all the configurations of 'in' and 'out' are attempted. In the lower part of the image, two possible configurations are presented, where the only difference consists in the target layer ℓ. As described in the formula 4, these two configurations will be paired to compute one of the marginal contributions of layer ℓ.

evaluated, and ensures that all coalitions inside the window are attempted when calculating its Shapley value. The modified Shapley formula is composed as follows:

$$\hat{\phi}_\ell(v) = \sum_{S \subseteq S_\ell \setminus \{\ell\}} w_S \left[v(S \cup L_\ell \cup \{\ell\}) - v(S \cup L_\ell) \right], \quad \forall \ell \in L \tag{4}$$

S_ℓ represents the window, the subset of all layers surrounding the examined layer ℓ. Figure 3 shows the symmetric window around a target layer in the middle section of the model. For edge layers, the window is shifted to the side to still contain the same number of blocks as the other cases. $L_\ell = L \setminus S_\ell$ instead represents the layers which remain outside of the window. The SWSV approximator assumes these layers are always active; otherwise, the resulting perplexity measurements would be highly unstable. The estimation thus requires attempting only $L * 2^w$ model configurations, where w is the window size, rather than the initial 2^L. The reason why we take $L * 2^w$ measurements, even if some configurations are repeated, is empirical: SV-NUP does not use a fixed dataset to compute perplexity, but smaller samples (see Section 5.2.3); the perplexity is computed with different samples per window, but the sample remains constant inside the window itself.

A small caveat The formula we presented before is taken directly from the SV-NUP study. We did not work out the composition of the weight w_S, because the paper itself does not include it. Without direct reference, we initially applied the version which appeared more faithful to the original Formula 5.1.1:

$$w_S = \frac{|S \cup L_\ell|! \, (|L| - |S \cup L_\ell| - 1)!}{|L|!} \tag{5}$$

based on the assumption that, despite considering only a small subset of marginal contributions, all the players N were theoretically involved in the game, and thus should be taken into account into the weighting. The reason for the ambiguity is the definition of S, which here represents an element of $S_\ell \setminus \{\ell\}$, compared to the original $N \setminus \{\ell\}$. Our extension (see the following Section 5.1.3) worked robustly with this assumption, but the original SV-NUP method lagged. After some experimentation, we detected that:

$$w_S = \frac{|S|! \, (|L| - |S| - 1)!}{|L|!} \tag{6}$$

produced better results for the original method, but led to (limited) instability in our extended work. This variation of the weight w_S represents every window as a separate, small game, while still adopting a global payoff function. Since the choice of weight is not declared in SV-NUP, and it concerns the translation of the Shapley values to the problem settings, we made the design choice of applying the weight definition that worked best to each method.

From Shapley values to pruning ratios At this point, the task becomes to translate Shapley values into a pruning criterion. We first formalize how nonuniform sparsity is defined in the case of unstructured pruning, then we tackle the specific case of a Shapley-value-based criterion.
Nonuniform applications of unstructured sparsity generally define a target sparsity ratio (e.g., 50%) such that, by the end of pruning, an equivalent portion of weights is removed from the model overall. The ratio, however, is not constant across layers; it is distributed so that some layers are pruned more, and others less, as long as the total reaches the target.

$$\frac{1}{|L|} \sum_{\ell=1}^{L} r_\ell \cdot \frac{|\mathbf{W}_\ell|}{|\mathbf{W}|} = r_{\text{target}} \tag{7}$$

As can be seen in Equation 7, for each layer ℓ, where $\ell \in L$, the set of all layers, the sparsity ratio r_ℓ is weighted by the fraction of weights that belong to the layer: $|W_\ell|$ is the number of weights in layer ℓ, $|W|$ is the total number of weights. The resulting pruning allocation can be visualized in Figure 4.

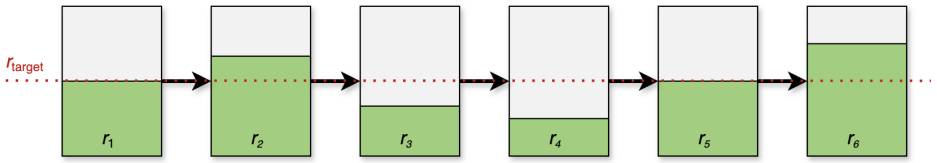


Figure 4: Nonuniform unstructured pruning. In the case of nonuniform unstructured sparsity, different sparsity ratios can be attributed to the layers, but the total sparsity ratio will still be the target ratio r_{target}. The pruned portion of each layer is presented in green.

In this setting, the Shapley values are converted into per-layer pruning ratios using the following formula:

$$r_\ell = r_{\text{target}} - \alpha_\ell + \text{mean}\{a_\ell\}_{\ell \in L} \qquad \text{where } a_\ell = \frac{2\lambda \left(\hat{\phi}_\ell - \min\{\hat{\phi}_\ell\}_{\ell \in L} \right)}{\max\{\hat{\phi}_\ell\}_{\ell \in L} - \min\{\hat{\phi}_\ell\}_{\ell \in L}} \tag{8}$$

where r_{target} is the global target sparsity ratio, and $\hat{\phi}_\ell$ is the approximated Shapley value for layer ℓ. Note that the Shapley values are not normalized: they represent the expected marginal contribution to the payoff function, perplexity, which is itself unnormalized. The values $\hat{\phi}_\ell$ are therefore first min-max rescaled and then multiplied by 2λ, so that the final term a_ℓ falls within the range $[-\lambda, \lambda]$. This rescaling step ensures that the resulting pruning ratios do not deviate excessively from the target sparsity, remaining within the range $[r_{\text{target}} - \lambda,\ r_{\text{target}} + \lambda]$. λ is a small scalar (e.g. 0.5) that bounds the extent to which the ratios are allowed to deviate from the target; according to the authors of SV-NUP, wider oscillations in the sparsity ratio can be detrimental to the model performance.

5.1.2 Adapting SV-NUP to 2:4 Pruning

Section 5.1.1 described how the layer-wise Shapley values are converted to an unstructured pruning criterion. In this section, we propose our adaptation of the allocation problem to 2:4 sparsity, whose fixed 50% ratio per matrix leaves no room for variable ratios. First, we define the all-or-nothing pruning scheme we aim to apply as an alternative to the variable pruning ratios, then we describe how Shapley values are used to select which layers to prune.

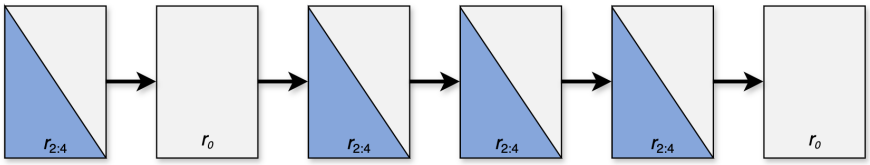


Figure 5: Nonuniform semistructured pruning. Our constrained approach to nonuniform sparsity only offers a binary choice: prune exactly to 50% according to the 2:4 pattern, or preserve the dense layer untouched. The resulting sparsity ratio will always be lower than 50%.

Semistructured pruning (all-or-nothing approach) The alternative we propose is to interleave fully dense layers with fully compressed layers. In this case, each layer is assigned either the fixed semistructured ratio $r_{2:4}$, corresponding to 50% weights removed according to the 2:4 pattern, or the ratio 0, which corresponds to no pruning at all.

$$r_{\text{eff}} = \sum_{\ell=1}^{L} r^\ell \cdot \frac{|\mathbf{W}\ell|}{|\mathbf{W}|} \leq r_{2:4}, \qquad r^\ell \in \{0, r_{2:4}\} \tag{9}$$

the pruning ratio to assign to layer ℓ can only be 0 or $r_{2:4}$. The final effective sparsity ratio r_{eff} will always be inferior to what we would have achieved using uniform 2:4 pruning, 0.5.

From Shapley values to semistructured pruning The first step to convert Shapley values ϕ to a pruning strategy consists of estimating the layer-wise Shapley values, either using the original SV-NUP procedure or the variant introduced below in Section 5.1.3. Once these values have been computed, they are used to determine which layers should be pruned. Specifically, layers are sorted in ascending order, based on their Shapley values and selected greedily, starting from those with the smallest importance score. The resulting selection procedure is presented in Algorithm 1. Starting from the least important layer (lowest associated Shapley score), elements of L are progressively added to the set of layers selected for pruning, S, which contains layers sparsified to $r_{2:4}$, until the pruning budget k is reached. The rest maintain a pruning ratio of 0.

Algorithm 1 Greedy Selection by Shapley value $\hat{\phi}$

Require: Set $L = \{\ell_1, \ldots, \ell_n\}$, score $\hat{\phi} : L \to \mathbb{R}$, budget $k \leq n$

$S \leftarrow [\,]$
$\tilde{L} \leftarrow \text{Sort}(L,\ \text{key} = \hat{\phi},\ \text{order} = \uparrow)$ ▷ select element with smallest $\hat{\phi}$ to prune
for $i = 1$ **to** k **do**
    $s_i \leftarrow \tilde{L}[i]$
    $S.\text{append}(s_i)$
end for
return $\mathcal{S}$

5.1.3 Beyond SV-NUP

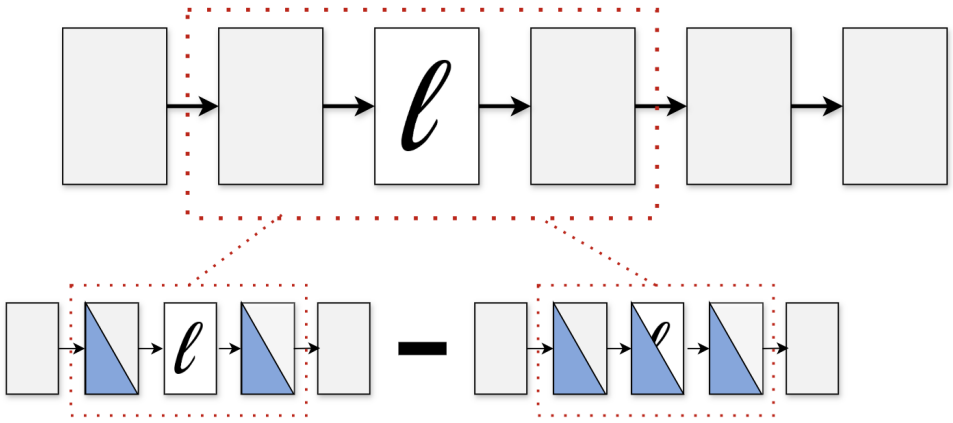


Figure 6: Pruning-aware SWSV approximator. As in Figure 3, we illustrate two possible configurations used to compute a marginal contribution. The only difference is that block removal (previously marked with an X) is replaced by 2:4 pruning (denoted by the blue triangles).

Pruning-aware SWSV In the original SV-NUP framework, the layers inside the window, but outside of the coalition are removed. The removal of the layer influences the payoff function,

perplexity, which is in turn reflected on the (approximated) Shapley values attributed to the single layers.
Under this formulation, however, the Shapley values cannot absorb any information about how the model actually responds to a specific pruning method. Our second research question hypothesizes that incorporating this information would yield different value trends. To test this hypothesis, we propose a pruning-aware version of the SWSV approximator. In Figure 6, we repeat the example proposed in Figure 3 with the new pruning-aware approach.
The theoretical foundation remains unchanged: the window approximation and relative Shapley computation described by Formula 5.1.1 still hold. The difference is constituted by the interpretation of exclusion from a coalition: rather than skipping all intermediate operations except the residual stream, an excluded layer is instead compressed using the target method, 2:4 sparsity, while continuing to execute its regular computations.

Q2: Layer granularity Throughout this section, we have used the term "layer" to denote a decoder block. As noted in Section 3.2, however, a decoder block is composed of two main subsystems, attention mechanisms and MLP layers, each containing multiple prunable matrices. Zhang et al. [ZDK24] apply a Shapley-based estimation, similar to SV-NUP itself, to these two components separately. Our third research question thus investigates whether considering them individually improves the pruning criterion, compared to the block-wise approach. This question is not specific to 2:4 pruning: finer-grained estimation remains underexplored in unstructured pruning frameworks as well. Unstructured pruning, however, can rely on the flexibility of variable ratios, whereas our nonuniform semistructured criterion is constrained to a stricter binary choice (pruned or not). The main aim of this section, then, is to increase flexibility within that constraint.
Adapting the SWSV approximator to this finer granularity, however, is not straightforward. A naive approach would treat each subsystem as a separate player and apply the same window size used for standard decoder-block estimation; this would, however, double the number of players. For SV-NUP, the authors report window sizes of 5 and 7 as robust (we adopt a window of 5 throughout the experiments in Section 6). Doubling the number of players within a window, by splitting each decoder block in two, would require 2^{10} or 2^{14} configurations per window, respectively. Combined with the cost of inference for each configuration, this quickly becomes prohibitively expensive.
Instead, we adopt a constrained approximation. The target decoder block is divided into two independent players, corresponding to its attention (a) and feed-forward (f) subsystems. All remaining decoder blocks, however, are treated exactly as in the original SWSV formulation: for every non-target block, the attention and FFN subsystems are constrained to share the same state, meaning they are either both present in a coalition or both absent. Only the two subsystems of the target block are allowed to vary independently. We thus represent a single player as a pair, the decoder block ℓ, and the targeted subsystem $c \in \{a, f\}$. The resulting approximation is then:

$$\hat{\phi}_{(\ell,c)}(v) = \sum_{S \in \mathcal{S}_{(\ell,c)}} w_S \left[v(S \cup \{(\ell, c)\} \cup L_\ell) - v(S \cup L_\ell) \right], \qquad c \in \{a, f\} \tag{10}$$

where L_ℓ still represents the section of the model excluded by the coalition 5.1.1, while the admissible coalitions satisfy:

$$(\ell', a) \in S \iff (\ell', f) \in S, \qquad \forall \ell' \neq \ell. \tag{11}$$

In practice, unless layer ℓ is the decoder block whose subsystem is being estimated, both components a and f should either belong to the coalition S or be excluded. As a result, every decoder block except the target contributes with both subsystems or neither, while only the target block allows the attention and FFN components to be evaluated independently. Under this constraint, the number of evaluated configurations becomes $L * 2^{w+1}$.

5.2 Components

5.2.1 Models

Architecture Targets of this study are models of the Llama family: Llama 3.2 1B, Llama 3.2 3B, and Llama 2 7B, together with two Qwen 3 architectures, in the 0.6B and 4B versions.
As anticipated in Section 3, model size within a family scales through the number of decoder blocks and the hidden dimension, and during the experiments, the only components to be pruned will be the attention mechanisms and feed-forward (or MLP) networks.

Model	Parameters	Blocks	Hidden Dim (ATTN)	FFN Dim
Llama 3.2 1B	1B	16	2048	8192
Llama 3.2 3B	3B	28	3072	8192
Llama 2 7B	7B	32	4096	11008
Qwen3 0.6B	0.6B	28	1024	3072
Qwen3 4B	4B	36	2560	9728

Table 1: Architectural comparison of selected LLMs.

5.2.2 Datasets

Wikitext-2 is a subset of the larger Wikitext-103 [MXBS16], a verified collection of full-length Wikipedia pages. Wikitext-2 was used for testing and calibration in SV-NUP, but also by Wanda and Wanda++ [SYCL25, YZG+25, SLBK24]. We chose to repeat the exact initial setting to validate, before further comparisons, our implementation against the results of the original SV-NUP (the source code is not publicly available); the results of this validation are reported in Appendix A.
For all samples and evaluations, we use sequences (sentences) of length 2048.

5.2.3 SWSV approximator hyperparameters

The SWSV approximator is a direct implementation of the formula presented in Equation 4. The original formulation requires a set of hyperparameters, which are reported here.

- **Window size**: the authors of SV-NUP investigate window sizes $w \in \{3, 5, 7\}$. After an ablation study, the authors report a certain robustness to the model in the choice of window, with 5 and 7 as equally valid options. To maintain the results uniform, we use a window size of 5 across all experiments.
- **Sample size**: when it comes to the calculation of the payoff function v, the SWSV approximator uses only a subset of Wikitext-2. The original setting uses a set of only 32 samples. SV-NUP reported robust results with such a limited sample; the pruning-aware method, however, was not as robust. As will be visible in the next section 6, the values computed with a pruning-aware Shapley values estimation method have a significantly lower standard deviation, which we hypothesize makes them more susceptible to the variations given by the sampling. For this reason, we increased the sample for pruning-aware methods to 128.

5.2.4 Pruning methods and hyperparameters

Pruning methods Both the pruning-aware Shapley value estimation and it final application, the pruning itself, requires an underlying pruning method to determine which weights to remove inside the model matrices. For this experiment, we select three baselines: magnitude [HPTD15], Wanda [SLBK24], and Wanda++ [YZG+25] pruning. The first two baselines were employed by SV-NUP, too: as anticipated, in Appendix A, we draw a comparison between the original SV-NUP results and ours. Wanda++, by contrast, is a more recent strategy designed specifically for 2:4 sparsity. It outperforms SparseGPT [FA23], the third baseline used by SV-NUP, on the task, while being considerably cheaper computationally. Both Wanda and Wanda++ require running model inference to estimate the Shapley values: Wanda uses inference uniquely to collect the average size of the activations passing through each weight, while Wanda++ computes regional gradients, both to guide pruning (RGS, regional gradient score) and to update the remaining weights (RO, regional optimization). All the pruning criteria just listed assign importance scores to each weight, then remove the least important ones following the 2:4 pattern described in Section 3.1.1.

Magnitude hyperparameters Magnitude pruning is an inference-free method, and thus does not require any hyperparameter.

Wanda hyperparameters Wanda uses the following hyperparameter:

- **Sample size**: 128 samples, with sequence length 2048. The sequences are used to compute a metric that summarizes activations scale (ℓ_2 norm) for a specific channel.

Wanda++ hyperparameters Wanda++ uses the same sample size as Wanda, but also requires as hyperparameters:

- **RGS scaling weight (α):** 1/100. The weight attributed to the additional regional gradient term is set to give it significantly less influence than the other term (vanilla Wanda score).
- **Learning rate:** 3e-7. The learning rate for the optimizer used during regional backpropagation.
- **RO samples:** 32. The number of samples used for backpropagation.
- **RO iterations:** 5. The number of regional optimization iterations, each interleaving a weight adjustment with a backpropagation cycle.

5.3 Experiments

In this section, we give an overview of the experiments and findings presented in Section 6.

Shapley values analysis As described in Section 5.1, the initial step in the framework is the estimation of the Shapley values. In the experimental case, we perform this estimation using the original SV-NUP method from Section 5.1.1 (also referred to as the "full removal" method), and the pruning-aware variation that we present in Section 5.1.3. Our second research question asks whether pruning-aware Shapley values differ from pruning-agnostic ones; however, we cannot exclude (and

in fact, we consider likely) that different pruning methods will themselves produce different Shapley values. We thus perform pruning-aware estimations using both magnitude pruning and Wanda. We exclude Wanda++, since it includes weight updates, and that would introduce a further layer of complexity. During this estimation, we do not recompute the Wanda scores for every different input, but store a single pruned configuration, which assumes the previous layers, used to compute the activation scores, are fully pruned. This experiment was repeated both for decoder block Shapley values and the subsystems Shapley values.

Sequential pruning In the application of nonuniform semistructured sparsity, we do not have a target ratio r_{target}, like for unstructured pruning, but a variable number of layers to prune. We also cannot preselect a threshold, as we need to investigate how the nonuniform criterion behaves with different amounts of sparsity (Q1). To produce a complete overview of the model pruning for each criterion, we perform a series of experiments we refer to as sequential pruning.
The nonuniform pruning criterion is based on one of the three series of Shapley values described before (full removal, magnitude or Wanda-based Shapley values), while the experiment itself is performed with magnitude pruning, Wanda, and Wanda++. In practice, going from no sparsity to 0.5 sparsity, we select a new layer to prune for every step, and we measure model perplexity (already introduced in Section 5.1.1) on the full Wikitext-2 test dataset considering the new set of pruned layers. For Wanda and Wanda++, we perform the pruning from scratch each time, to ensure the results remain consistent.
To draw a comparison across pruning methods, we add as a threshold the perplexity of unstructured sparsity at 50%, to show how semistructured sparsity relates to it

Summary tables The results of sequential pruning (see Figure 10 and 12), the perplexity measurements, describe a curve as they increase. To summarize the large amount of collected data (see Appendix B), we divide this curve into quarters: the number of decoder blocks of all models is divisible by 4, thus these thresholds correspond to a real measurement points for all models. The only exception is the fourth quarter, whose end corresponds to the fully pruned model; in that case, we also measure the perplexity at 0.45 sparsity. These individual measurements, however, disregard the values in the intermediate stages and are not robust to single-value oscillations. We therefore complement it with the area under the curve (AUC) of each quarter: considering the area delimited by a quarter, we compute the integral of the perplexity. To make the visualization clearer, we marked the quarter boundaries with vertical bars in the sequential pruning plots.
After that, we take a similar approach to summarize the split Shapley values findings. Since there is no guarantee the combination of pruned attention and MLP layers will reach the exact sparsity ratio that represents the intermediate thresholds (0.125,0.25,0.375), to maintain consistency, we measure the perplexity described by the curve at that exact spot, even if it does not match a real measurement. We also compute the AUC based on the same principle. To make the AUC comparable across decoder blocks and values split between attention and feed-forwardpley values, we assign to both of them a horizontal space of N (number of decoder blocks).

6 Results

The first section evaluates whether SV-NUP identifies a small subset of critical decoder blocks (Q1), and compares the pruning-agnostic and pruning-aware Shapley value estimators (Q2). The second section applies the Shapley values estimation to attention mechanisms and feed-forward networks (Q3).

6.1 Decoder block Shapley values

In Figure 7, we present the decoder block Shapley values for the model Llama 2 7B. The values are computed based on the original SV-NUP method (Section 5.1.1), plus two pruning-aware variations, based on magnitude and Wanda pruning (Section 5.1.3). The original method is based on the hypothesis that the layers would be fully removed from the model, and thus we also refer to it as the full removal method; it produces, overall, the largest Shapley values. This is expected, since the payoff function (inverse perplexity) is not standardized: the difference between a layer being present or absent is far more pronounced than the difference between a layer being dense or pruned, so marginal contributions (and with them, the Shapley values) increase accordingly. We thus adopt a separate scale for the full removal method to propose a comparison with the pruning-aware values.

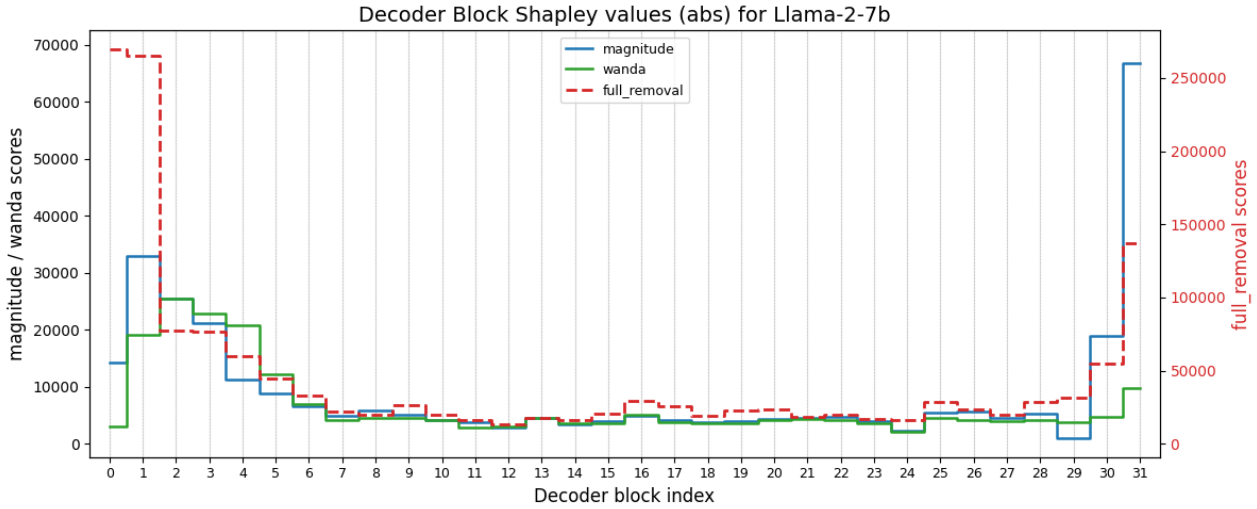
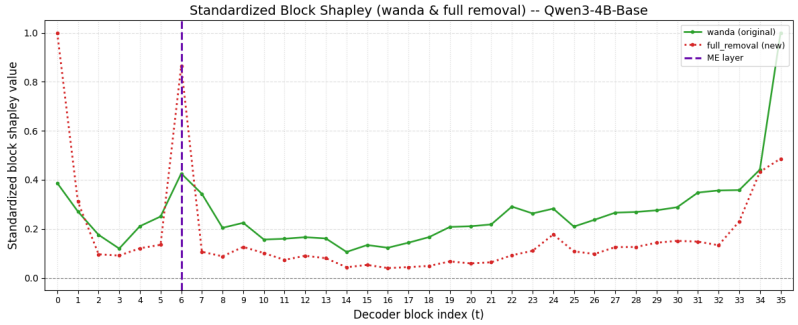


Figure 7: Shapley values for the decoder blocks of model Llama 2 7B. The raw Shapley values occupy different ranges across different estimation methods. Magnitude and Wanda pruning methods produce values in a similar range (scale on the left). The original SV-NUP method, on average, produces values 5 times larger (scale on the right). Figure 9 presents the same values, not overlapped and standardized.

The full removal method confirms the attribution of high values to a small number of layers (Q1), in this case placed exactly at the beginning and end of the network: layers 0, 1, and 31. These outliers, however, partially disappear in the pruning-aware estimations, although magnitude-based Shapley values still report an abrupt increase in layer 31. The findings in Section 6.2 on the split attention and MLP Shapley scores will later highlight how these importance scores are not necessarily

attributed to the whole decoder block, but to a specific subcomponent; thus, we postpone a more in-depth analysis to Section 6.2.
While in this analysis we mainly focus on Llama 2 7B, similar trends to the ones observed in Figure 7 appear across the other measurements made on the Llama architecture family. This does not mean that models (with variable numbers of decoder blocks) exhibit the same distributions, but the general trends reported in this analysis constitute most of their major variations, although smaller models show more instability. This is likely due to the removal or pruning of the layer damaging performance more than in larger models, independent of the non-uniformity of the pruning.
Looking at Figure 7, we observe that the vast majority of mid-to-deep intermediate layers present uniformly low Shapley values in the intermediate area between layers 7 and 28, and that this behavior is consistent across estimation methods. This homogeneity across the mid-deep layers has already been reported in multiple studies [SPNJ25][HSSL24][IYT+25]. "Transformer Layers as Painters", by Sun et al. [SPNJ25] in particular suggests this intermediate stage shares a common representation space, and works like an assembly line: the different layers specialize in a set of tasks and only contribute to the computation when it concerns them; the rest of the information passes, untouched, through the residual streams that accompany the attention and MLP layers, as previously presented in the Equation 2.
Apart from the sharp outliers and this uniformly low stage, all the estimation methods applied to Llama 2 7B report other values higher than average in the layers; these, however, do not constitute sharp peaks, but a curve spanning multiple layers in the mid-initial stage, up to layer 6. This phenomenon is already present in the full removal-based Shapley values, but it is of limited magnitude compared to the three outliers.



(a) Qwen 3 4B

Figure 8: Wanda-based standardized Shapley values for the model Qwen 3 4B. The massive emergence (ME) layer, represented by the dashed purple line, is placed at the position identified by Shi et al. [STY+26].

The model Qwen 3 4B reports a similar curve, observed both in the full removal-based and Wanda-based Shapley values, which peaks at layer 6. This behavior matches what was reported in "A Single Layer to Explain Them All: Understanding Massive Activations in Large Language Models," by Shi et al. [STY+26]. Using Qwen 3 4B as a main example, the study identifies the existence, across different architectures, of a single Massive Emergence Layer (ME Layer), responsible for introducing the vast majority of outliers (or massive activations) in the model. Using multiple metrics, the study locates this layer at position 6 in Qwen 3 4B, matching exactly the peaks observed in the full removal and Wanda-based Shapley values (see Figure 8). In this specific case, the phenomenon

is also visible in the full removal Shapley values; however, we did not observe a comparable peak at this position in any other layer, nor does Zhang et al. [ZDK24] report a cornerstone layer in a similar initial-to-intermediate position. We therefore hypothesize that Qwen 3 4B is a model in which this phenomenon is particularly strong, possibly the same reason why it was selected as an example by Shi et al. [STY+26].

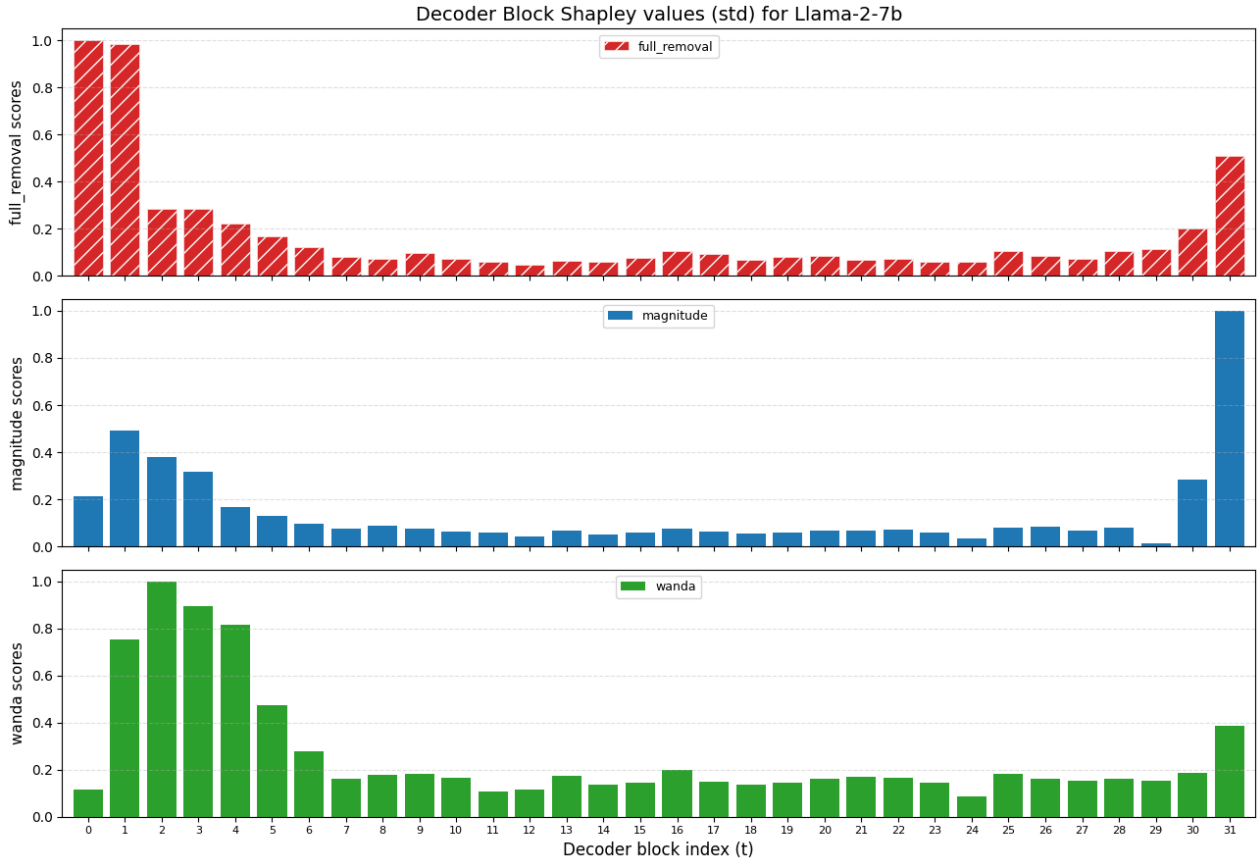
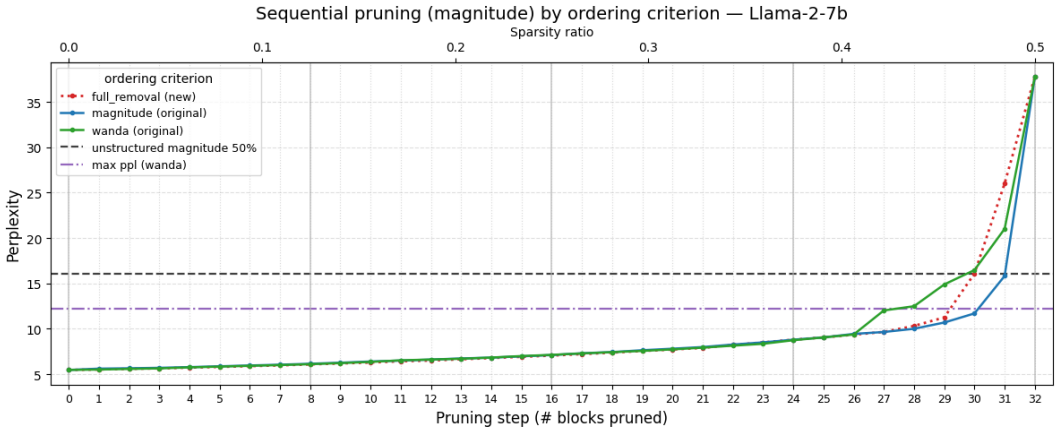
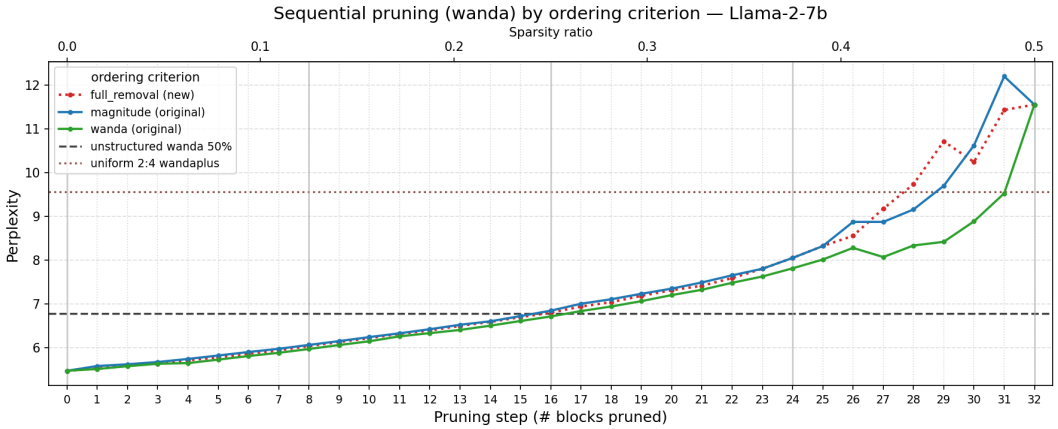


Figure 9: Standardized Shapley values for the decoder blocks of the model Llama 2 7B. The three series represent the values computed based on the original model-agnostic method (in red and dashed, presented in Section 5.1.1), and the pruning-aware versions, based on magnitude and Wanda pruning (in blue and in green, described in Section 5.1.3). The values were standardized by dividing them by the maximum value of the corresponding series.

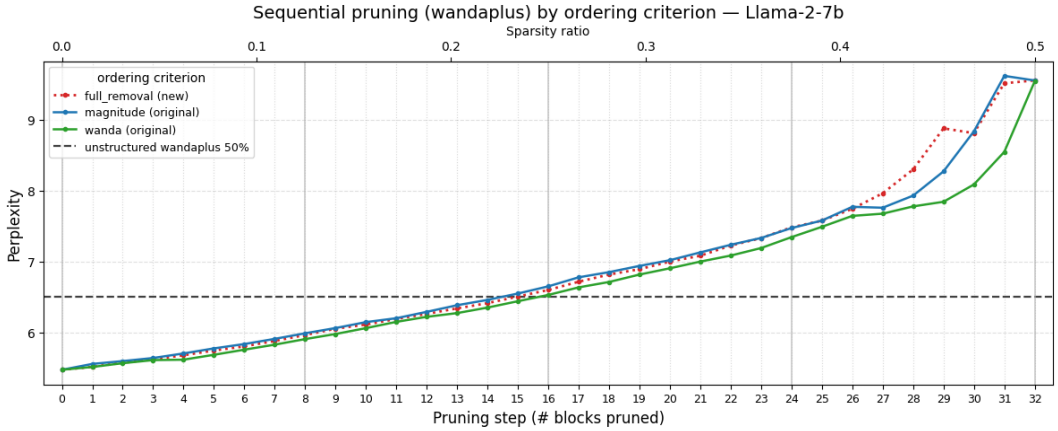
Decoder block sequential pruning At this point, we applied Algorithm 1 to convert the Shapley values into pruning criteria. To make the priority order clearer, we present the standardized Shapley values in three separate plots in Figure 9. To evaluate how each method performs at every possible sparsity level, in Figure 10 we perform a sequential pruning experiment, as described in Section 5.3; results for the remaining models are summarized in Table 2.
Given the three pruning criteria, we can see how up to 26 layers, all of them produce a linear increase in the perplexity. Not only the number of layers, but also the perplexity increase is similar, going from 5.47 to around 9: we note that the pruning methods operate in overall different ranges,



(a) Magnitude pruning



(b) Wanda pruning



(c) Wanda++ pruning

Figure 10: Sequential pruning of Llama 2-7B: decoder block Shapley in value attributions. In the model, an increasing number of layers was pruned, according to the pruning criteria, which differ on the values they are based on. From top to bottom, the pruning methods are magnitude, Wanda, and Wanda++. The pruning curve is divided into quarters, which correspond to the AUCs shown in Table 2.

thus different plot scales. These layers mostly represent the intermediate-late stage of the model, highlighted before, with overall uniform, low Shapley values. The exception is constituted by layer 0, which the Wanda criterion prunes at step 4. Both the full removal and magnitude criteria assigned it moderate to high importance, while in Wanda, its importance score dropped. **This is a clear example of how the pruning-aware estimation method identified a different priority in the pruning order, consequently reducing the perplexity of the model.**

Overall, the best order for each pruning baseline seems to match the most similar estimation method: magnitude-based criterion for magnitude pruning, Wanda-based criterion for Wanda and Wanda++. This is further confirmation that the Shapley values, incorporating in their estimation information about the pruning itself, detected trends specific to the pruning method that can be used to enhance the nonuniformity criterion.

In the final section (around a sparsity rate of 0.45), the baseline pruning methods start to show different trends, which will lead them to the final perplexity score, lower for Wanda++ and Wanda, significantly higher for magnitude. The most extreme effect appears with magnitude pruning; in the last two steps, the method leaves the 5-12 perplexity range that, up to that point, was shared with Wanda, and reaches the perplexity of 37.76 (see Table 2).

Both Wanda and Wanda++ report their best results with the Wanda criterion. When they surpass their unstructured counterparts, they do so having achieved a pruning ratio of 0.25. It must be noted that only half of the parameters are pruned; in comparison, only 2:4 sparsity can translate this advantage into in-GPU compression. While less pronounced than magnitude, they also experience a jump in the final stage. While it is sufficient for magnitude pruning to preserve two layers to achieve a perplexity lower than uniform 2:4 Wanda, preserving just one layer for Wanda (the ME layer hypothesized in the previous section) makes it reach the same level of perplexity as 2:4 uniform Wanda++. **Given the fully pruned model (that, as anticipated, varies across pruning baselines), we see that the pruning of the last three layers (for this model, a resulting sparsity ratio of 0.453) contributed 71.7%, 27.1%, and 17.7% of the final perplexity for magnitude, Wanda, and Wanda++ pruning, respectively.**

In Table 2, we show the results across various models. As anticipated in Section 5.3, while it is unfeasible to present here for all models all the pruning steps as in Figure 10, single perplexity measurements alone can be misleading. Thus, for every quarter of the sequential pruning run (we marked the boundaries in Figure 10 for easier visualization), we measured both the final perplexity and the AUC curve.

As we can see in the table, the results collected in Llama 7b are mostly replicated across all models; magnitude pruning runs tend to consistently favor the pruning criterion, while Wanda and Wanda++ experiments appear more unstable. The model that presents the highest instability is Llama 3B. In this model, the initial and last layers, whose elimination in Llama 2 7b even decreased perplexity, present equally high values, unlike in the smaller model, which instead consistently performs better with the Wanda criterion for Wanda and Wanda++ pruning. Similarly, for the Qwen 4B, the Wanda-based criterion works better for the first three quarters, but in the last quarter, it is surpassed by the full removal method. As seen in Figure 8, the full removal method gives the second-highest priority to the ME layer, while in the Wanda criterion, its removal happens one step later, suggesting that the pruning criterion taking it into account positively contributed to the result.

		Perplexity					AUC				
Model	Sparsity	12.5%	25%	37.5%	*50% - 1 layer*	50%	0%-12.5%	12.5%-25%	25%-37.5%	37.5%-50%	Total
Llama-3.2-1B (9.75)	**Magnitude**										
	Full removal	**16.51**	55.13	211.99	**900.64**		**51.15**	134.43	414.50	3479.45	4079.53
	Magnitude	17.09	**42.54**	**161.33**	**900.64**	**3288.67**	51.72	**105.69**	**349.81**	**3451.38**	**3958.60**
	Wanda	17.09	49.15	316.40	3673.05		52.15	127.49	507.26	11282.60	11969.50
	Wanda										
	Full removal	13.39	25.12	45.67	83.75		45.76	75.60	127.42	263.57	512.34
	Magnitude	**13.33**	19.69	32.45	83.75	**93.05**	**45.27**	63.73	99.44	237.69	446.13
	Wanda	**13.33**	**18.39**	**30.15**	**57.47**		45.33	**61.81**	**93.61**	**202.17**	**402.91**
	Wandaplus										
	Full removal	**12.12**	17.27	26.24	38.14	43.41	**43.27**	58.31	83.09	134.88	319.56
	Magnitude	12.61	17.01	23.98	38.54	**42.80**	44.16	57.70	81.66	129.86	313.39
	Wanda	12.61	**16.26**	**23.61**	**35.50**	43.44	44.23	**56.66**	**77.85**	**126.11**	**304.86**
Llama-3.2-3B (7.81)	**Magnitude**										
	Full removal	**10.09**	**14.95**	70.21	228.68		**61.20**	85.40	215.06	1124.97	1486.63
	Magnitude	10.10	15.25	**30.65**	**201.40**	**387.54**	61.68	**85.31**	**154.62**	**778.82**	**1080.43**
	Wanda	10.12	14.98	33.73	294.16		61.62	85.35	156.87	1132.71	1436.54
	Wanda										
	Full removal	**9.25**	11.80	21.08	**27.60**		**59.14**	**73.00**	107.94	174.78	414.87
	Magnitude	9.46	11.68	**15.47**	29.09	**31.97**	60.03	73.54	**93.87**	155.65	383.09
	Wanda	9.44	**11.56**	15.91	29.82		60.07	73.18	94.42	**152.30**	**379.98**
	Wandaplus										
	Full removal	**8.98**	**10.98**	14.77	**18.46**	**20.13**	**58.27**	**69.41**	88.28	117.30	333.26
	Magnitude	9.28	11.28	**13.49**	18.49	20.25	59.50	71.53	86.72	**112.45**	**330.20**
	Wanda	9.26	11.13	13.52	19.62	20.21	59.39	71.23	**86.56**	114.93	332.10
Llama-2-7b (5.47)	**Magnitude**										
	Full removal	**6.08**	**7.05**	8.80	26.02		**45.94**	**52.33**	62.45	115.04	275.76
	Magnitude	6.15	7.11	8.80	**15.82**	**37.76**	46.45	52.96	62.93	**99.65**	**262.00**
	Wanda	6.11	7.13	**8.75**	21.02		46.06	52.90	**62.39**	118.58	279.92
	Wanda										
	Full removal	6.03	6.80	8.05	11.43		45.79	51.26	58.70	77.95	233.70
	Magnitude	6.06	6.84	8.05	12.19	**11.55**	46.08	51.44	59.07	77.52	234.12
	Wanda	**5.97**	**6.71**	**7.81**	**9.52**		**45.51**	**50.65**	**57.73**	**69.19**	**223.08**
	Wandaplus										
	Full removal	5.96	6.60	7.48	9.51	9.55	45.52	50.15	56.10	67.30	219.07
	Magnitude	5.99	6.65	7.47	9.62	9.55	45.74	50.40	56.34	66.27	218.75
	Wanda	**5.91**	**6.53**	**7.34**	**8.55**	**9.54**	**45.26**	**49.68**	**55.27**	**63.50**	**213.71**
Qwen3-0.6B-Base (12.66)	**Magnitude**										
	Full removal	21.90	79.94	**540.58**	15146.66		110.47	321.24	1758.23	40588.13	42778.07
	Magnitude	**20.45**	**51.58**	601.50	**5017.07**	**25307.58**	**109.87**	**214.84**	**1374.91**	**25375.97**	**27075.58**
	Wanda	21.43	60.69	758.84	**5017.07**		112.92	264.80	1821.27	27970.20	30169.19
	Wanda										
	Full removal	15.41	22.90	47.50	97.28		96.65	130.57	234.31	470.70	932.25
	Magnitude	15.81	21.43	35.17	**69.99**	**132.65**	97.50	127.67	194.94	**413.87**	**833.97**
	Wanda	**15.31**	**21.05**	**34.59**	**69.99**		**96.39**	**124.32**	**186.86**	439.85	847.41
	Wandaplus										
	Full removal	**14.79**	19.70	32.44	59.87	**70.55**	**95.18**	118.57	177.63	303.61	694.99
	Magnitude	15.49	19.62	28.87	**45.81**	73.87	96.53	121.17	165.58	**280.92**	**664.20**
	Wanda	14.96	**19.13**	**28.62**	46.48	72.47	95.27	**117.62**	**163.86**	288.51	665.25
Qwen3-4B-Base (7.90)	**Magnitude**										
	Full removal	**9.59**	28.24	77.30	210.92		**77.20**	138.61	443.34	1417.74	2076.89
	Magnitude	9.88	**15.54**	**35.77**	**176.52**	**559.87**	78.20	**107.14**	**211.04**	**954.82**	**1351.19**
	Wanda	9.82	33.31	51.91	550.89		77.71	174.99	390.33	1935.58	2578.60
	Wanda										
	Full removal	8.77	10.34	13.57	**17.74**		74.35	85.10	106.03	**142.32**	407.81
	Magnitude	8.89	10.70	14.33	17.90	**18.42**	74.84	86.93	110.95	152.87	425.59
	Wanda	**8.62**	**10.21**	**13.00**	19.51		**73.62**	**83.59**	**103.39**	144.42	**405.02**
	Wandaplus										
	Full removal	8.61	9.59	11.84	**14.62**	**15.56**	73.92	81.55	95.48	**120.43**	371.39
	Magnitude	8.70	9.80	12.15	15.31	15.65	74.29	82.96	97.50	124.84	379.59
	Wanda	**8.48**	**9.51**	**11.66**	15.39	15.58	**73.32**	**80.40**	**94.36**	122.00	**370.08**

Table 2: Per-quarter perplexity (left) and area under the curve (right) of the sequential-pruning runs under Shapley value attribution, for all models. Q1–Q3 are the perplexity at the end of each quarter; *Q4* (italic) is the perplexity at the step just before the model is fully pruned, and Q4 (pruned) is the fully-pruned perplexity. The original (pruning-agnostic) dense perplexity is given in parentheses under each model name. AUC is reported per quarter (not cumulative) and in total. Bold marks the best (lowest) value per column within each pruning method.

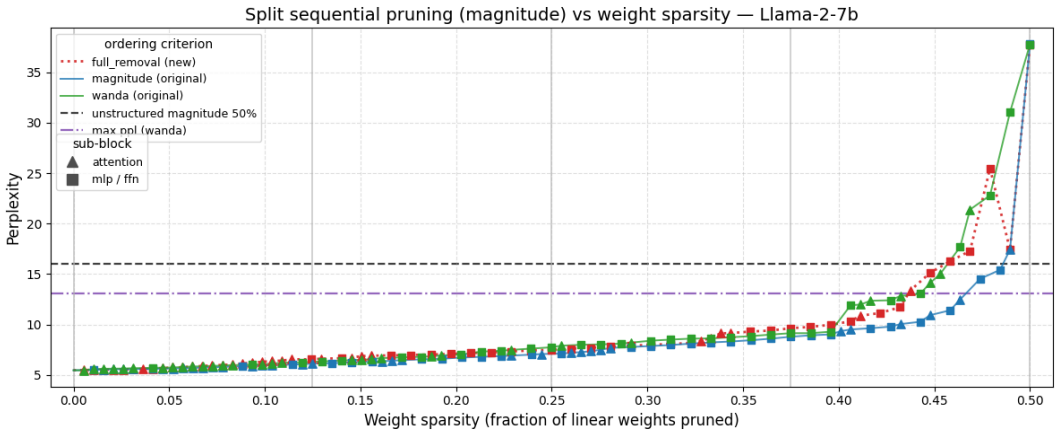
6.2 Attention & MLP Shapley values

Figure 11: Standardized Shapley values for the attention mechanisms and feed-forward networks of the model Llama 2 7B. The values are computed using the same pruning-agnostic and pruning-aware methods, but split based on 5.1.3. The values for the two components are paired and indexed based on the original decoder block, with the attention mechanism scores on the left and MLP scores on the right.

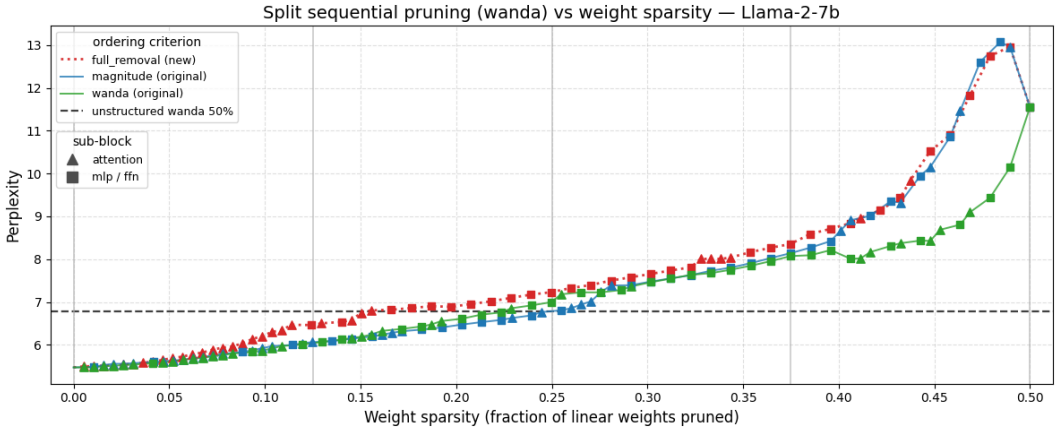
Attention & MLP Shapley values In Figure 11, we present the Shapley values attributed to the single attention mechanisms and feed-forward networks. Overall, the Shapley value trends trace back to the ones observed in Figure 9, thus we only present a standardized version.
Across the middle-deep layers, the Shapley values are, as before, uniformly low, but the MLP layers consistently present larger values than the attention mechanisms. We must note that the feed-forward networks themselves are consistently larger than the attention mechanism, as presented in Table 1: the higher values thus can be a direct effect of the greater number of weights removed or pruned during the estimation phase.
In most decoder blocks, we observe a noticeable difference between the values of attention and feed-forward networks. This is particularly noteworthy in layers 1, 2, and 31 for full removal, and layers 30 and 31 for magnitude-based Shapley values. In most of these layers, the MLP layer

constitutes the outlier, while the attention mechanism is assigned a low Shapley value. The main exception, in the full removal and magnitude Shapley values, is given by the attention mechanism of layer 0. Looking back at Figure 9, the first layer in full removal had the highest priority, while now its subsystems are second and fourth for importance, respectively.
A possible interpretation is given by Busigin et al. [BP26], who perform a mechanistic explainability study on the phenomenon of "detokenization" [KORS25]: since at times a word is encoded into multiple subword tokens, the early layers link these tokens together to create a full word representation. Given an initial subword token, they report that for the following subword token, the attention mechanisms produce a token-specific signal, based on the preceding tokens, and that the feed-forward network takes this signal, plus the original embedding, to link the token to its full-word representation. Thus, the components work jointly, but attention constitutes the fundamental link to the previous context.
A study from Lad et al. [LLGT25] instead proposes the existence of another effect called residual sharpening, which primarily takes place in the final layer. As seen in Equation 2, the output of each subsystem is the addition of the computation of the subsystem and the residual stream. Compared to the residual stream, the contribution to the block output from the MLP layer rises significantly in the last layer. Through further mechanistic interpretability experimentation, they report that this last layer tends to consist of suppression neurons, which have the primary goal of reducing the likelihood of specific tokens. As such, the layer as a whole would appear to "sharpen" the previous results by down-weighting unlikely features, limiting the use of residual streams that would bypass this filter. This constitutes a possible explanation for the abnormal behavior observed in the previous pruning experiment: in the previous sequential pruning experiments, its pruning caused a significant perplexity increase in the magnitude experiment, while it decreased the perplexity in the Wanda experiments. In the values presented below in Figure 12, the phenomenon also recurs. Overall, we cannot draw conclusions on the way the two pruning methods distort the weights to produce this contrasting behavior, but we still note it as worthy of future investigation.

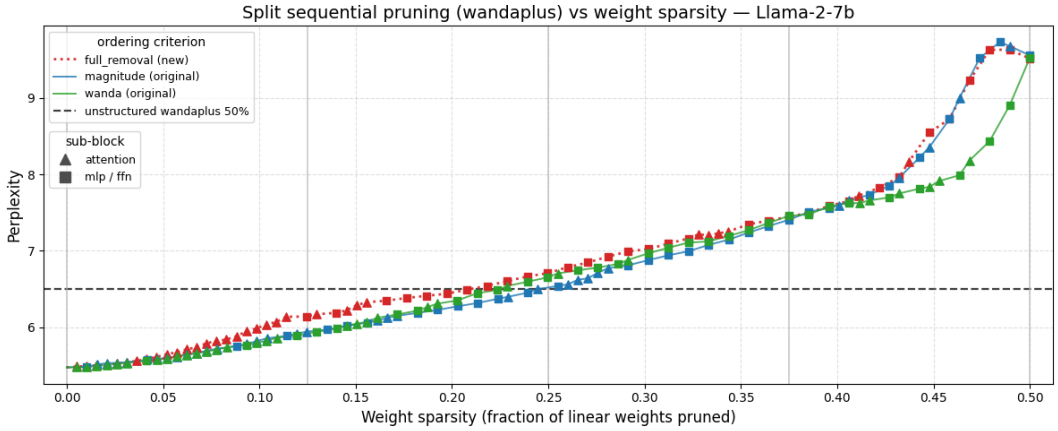
Attention & feed-forward blocks sequential pruning In Figure 10, we observed the pruning of the model Llama 2 7b decoder block-wise. In Figure 12, we repeat the same experiment, but in this case, we add to the pruned layer subset one attention layer or feed forward network at a time. The use of split values for the two subsystems produces similar trends to before. We can however identify a clear redistribution in how the criteria prioritize attention and feed forward networks. The shape of the placeholder in the plots highlights which element was pruned: triangle for an attention mechanism, square for a feed-forward layer. Unlike the previous example, the sparsity does not increase uniformly, since attention and feed-forward networks have different amounts of weights.
As displayed by the matrix dimensions in Table 1, the attention mechanism is consistently smaller in all models. This greatly reshapes the way we look at the step-by-step pruning. At the beginning of the sequential pruning, the criteria almost exclusively target attention heads. These, however, contain far fewer weights than the feed-forward networks: after pruning more than half of the attention blocks (18 for the Wanda criterion) in the first quarter, the model only reaches 12.5% sparsity. At that point, the pruning starts to target the MLP layers and continues to remove them almost exclusively: as before, pruning during this stage primarily targets the mid-to-deep portion of the model, although here it removes the attention heads first and the MLP feed-forward layers later. This continues (as before) until the model reaches around 0.45 sparsity, across all pruning baselines,



(a) Magnitude pruning



(b) Wanda pruning



(c) Wandaplus pruning

Figure 12: Sequential pruning of Llama 2-7B based on attention and feed-forward networks Shapley values. In the model, an increasing number of layers were pruned according to the pruning criteria, which differ in the values they are based on. From top to bottom, the pruning methods are magnitude, Wanda, and Wanda++. The pruning curve is divided into quarters, which correspond to the AUCs proposed in Table 3.

		Perplexity					AUC				
Model	Sparsity	12.5%	25%	37.5%	*50% - 1 layer*	50%	0%-12.5%	12.5%-25%	25%-37.5%	37.5%-50%	Total
Llama-3.2-1B (9.75)	**Magnitude**										
	Full removal	59.89	275.65	485.77	**1056.05**		88.89	673.42	1484.21	3837.81	6084.32
	Magnitude	**13.72**	**27.86**	**71.42**	2179.41	3288.67	**45.21**	**75.86**	**181.59**	**1690.70**	**1993.35**
	Wanda	26.49	60.21	218.77	2179.41		74.15	159.95	418.35	3264.70	3917.14
	Wanda										
	Full removal	18.62	38.34	57.24	83.74		52.54	115.49	195.80	278.60	642.42
	Magnitude	**12.41**	**18.91**	**25.76**	**68.34**	93.05	**43.46**	**61.56**	**90.67**	**161.06**	**356.74**
	Wanda	14.70	20.98	33.37	**68.34**		49.61	69.11	108.26	187.85	414.83
	Wandaplus										
	Full removal	14.08	20.80	29.24	39.77	44.00	47.14	70.22	100.74	141.30	359.41
	Magnitude	**11.87**	**16.09**	**21.99**	**39.45**	**42.92**	**42.63**	**55.41**	**75.46**	**114.63**	**288.13**
	Wanda	13.15	17.02	24.07	39.98	43.48	46.10	59.54	82.37	122.76	310.76
Llama-3.2-3B (7.81)	**Magnitude**										
	Full removal	29.97	83.44	134.34	**244.68**		115.48	402.42	750.99	1308.18	2577.08
	Magnitude	**9.59**	**12.88**	**21.28**	**244.68**	387.54	**60.31**	**78.46**	**115.61**	**593.47**	**847.85**
	Wanda	11.02	19.73	72.20	323.55		64.10	105.20	259.02	1273.48	1701.80
	Wanda										
	Full removal	9.95	14.86	20.37	**29.24**		61.55	87.02	121.00	176.01	445.58
	Magnitude	9.22	**11.34**	**14.43**	**29.24**	31.97	**58.99**	72.60	**89.19**	**148.32**	**369.10**
	Wanda	**9.17**	11.71	16.36	34.23		59.37	**72.07**	94.06	167.95	393.45
	Wandaplus										
	Full removal	9.23	11.23	14.54	18.95	20.15	59.02	71.27	89.42	116.21	335.93
	Magnitude	8.95	**10.77**	**13.51**	**18.91**	**20.09**	**58.21**	69.21	**84.27**	**114.69**	**326.37**
	Wanda	**8.92**	10.95	13.63	20.00	20.16	58.41	**68.76**	85.13	115.12	327.42
Llama-2-7b (5.47)	**Magnitude**										
	Full removal	6.57	7.51	9.64	**17.42**		47.36	56.43	67.03	120.69	291.51
	Magnitude	**6.13**	**7.10**	**8.80**	17.43	37.76	**45.96**	**52.77**	**63.70**	**98.92**	**261.35**
	Wanda	6.31	7.76	9.15	31.04		46.90	55.68	67.67	130.93	301.18
	Wanda										
	Full removal	6.47	7.23	8.36	12.96		46.92	55.01	62.23	82.12	246.28
	Magnitude	6.06	**6.79**	8.14	12.96	11.55	45.80	**51.12**	**60.18**	81.61	238.72
	Wanda	**6.04**	7.01	**8.07**	**10.14**		**45.61**	52.01	60.33	**69.89**	**227.84**
	Wandaplus										
	Full removal	6.14	6.71	7.45	9.63	**9.52**	46.01	51.37	56.76	67.21	221.35
	Magnitude	5.94	**6.52**	**7.41**	9.68	9.56	45.35	**49.68**	**55.58**	67.08	217.68
	Wanda	**5.92**	6.66	7.46	**8.91**	9.53	**45.25**	50.19	56.20	**63.74**	**215.38**
Qwen3-0.6B-Base (12.66)	**Magnitude**										
	Full removal	30.41	169.93	1321.45	19563.51		128.83	577.43	4876.25	110177.85	115760.37
	Magnitude	**20.78**	**41.47**	**251.17**	**6744.17**	25307.58	**109.31**	**207.25**	**687.45**	**17805.88**	**18809.88**
	Wanda	23.58	115.40	1045.96	**6744.17**		121.03	455.26	2258.23	21486.48	24321.00
	Wanda										
	Full removal	15.64	22.69	40.59	106.39		98.24	128.67	218.10	471.67	916.68
	Magnitude	15.45	**21.04**	35.14	**75.74**	132.65	**96.37**	**126.55**	**181.21**	404.63	808.75
	Wanda	**15.35**	21.39	**33.06**	**75.74**		97.87	127.55	186.30	**396.16**	**807.88**
	Wandaplus										
	Full removal	14.99	**19.51**	29.41	63.11	70.74	96.49	**118.35**	168.60	301.48	684.92
	Magnitude	15.01	19.69	29.12	51.39	71.82	**95.38**	120.39	**162.32**	278.57	656.66
	Wanda	**14.85**	19.65	**27.40**	**50.91**	**70.37**	96.43	118.82	164.50	**270.04**	**649.78**
Qwen3-4B-Base (7.90)	**Magnitude**										
	Full removal	17.60	36.67	59.65	292.18		100.36	228.41	521.91	1538.58	2389.26
	Magnitude	**9.30**	**14.04**	**35.79**	**180.23**	559.87	**76.14**	**100.43**	**199.79**	**926.70**	**1303.06**
	Wanda	10.66	46.32	72.66	691.43		82.04	133.69	717.42	2800.16	3733.31
	Wanda										
	Full removal	8.96	10.41	13.55	18.22		75.81	86.63	106.52	**149.20**	418.15
	Magnitude	**8.68**	**10.37**	14.29	**18.05**	18.42	**73.73**	**84.73**	108.65	152.79	419.89
	Wanda	8.72	10.46	**13.41**	21.86		74.92	85.40	**105.77**	149.96	**416.06**
	Wandaplus										
	Full removal	8.74	9.69	11.67	14.87	15.57	74.95	82.96	95.28	120.51	373.69
	Magnitude	8.58	9.73	11.98	15.30	15.62	73.50	81.94	96.95	123.55	375.93

Table 3: Per-quarter perplexity (left) and area under the curve (right) of the sequential-pruning runs for attention and feed-forward Shapley values, for all models. Q1-Q3 are the perplexity at the end of each quarter; *Q4* (italic) is the perplexity at the step just before the model is fully pruned, and Q4 (pruned) is the fully-pruned endpoint. The original (pruning-agnostic) dense perplexity is given in parentheses under each model name. AUC is reported per quarter (not cumulative) and in total. Bold marks the best (lowest) value per column within each pruning method.

and mostly results in a linear increase in perplexity. In this task, however, the full removal method begins to lag earlier, while in the previous sequential pruning experiment, the main discrepancies took place almost exclusively after the drifting point of sparsity 0.45.
In Table 3, we present a summary of all the models' pruning. Compared to the decoder block values, Llama 7B appears to be an exception: in its case, the use of separate values slightly, but consistently worsens the overall perplexity. We suspect this might be produced by the missing pruning of the MLP subsystem of layer 0. In the previous run, the entire block was pruned almost immediately, with minimal perplexity increase and subsequently all lower perplexity values. In this case, however, only the attention mechanism received a very low score from the Wanda criterion. In every other case, **splitting the Shapley values neither meaningfully lowers perplexity nor makes it worse; overall, the split values follow the same patterns already described in Section 6.1.** The precise ratio still depends on which pruning criterion is chosen, but this approach still adds flexibility to the final sparsity ratio, since it doubles the number of pruning steps available and, with them, the range of choices.

7 Limitations

In this section, we point out some methodology limits worth mentioning.

Payoff function Across all our experiments, we adopted perplexity as our main metric. Perplexity measures the performance of the model at next-token generation, the only task these models are pretrained on. [ZDK24], by contrast, used multiple payoff metrics, including accuracy on multiple benchmarks of the Harness Suite [GTA+23]. While the arbitrary use of one of these alternative metrics is equally likely to introduce its own biases in the setting, perplexity as a payoff function might also be concealing relevant phenomena.

SWSV estimator In this work, we introduced two extensions to the SWSV estimator proposed by SV-NUP [SYCL25]. Pruning awareness only reshaped how Shapley values are adapted to the problem setting, while the introduction of finer granularity modified the SWSV approximator itself. For the finer granularity SWSV estimator, we did not carry out an extensive, evidence-based analysis confirming that the resulting Shapley values are as robust as those estimated through random sampling methods, such as the one used by SV-NUP [SYCL25]. Section 5.1.3 laid out the reasoning behind our chosen redesign.

No finetuning In practice, pruned models deployed in real settings are often finetuned. Even when no further task specialization is needed, finetuning can help recover some of the capability lost during compression [HSW+21]. The weight update performed by Wanda++ [YZG+25] could be considered a partial substitute for this step, but overall, in our experiments, we limited ourselves to testing a pretrained model, without any further finetuning beyond the pruning method itself.

8 Future research

Pruning-aware Shapley values for unstructured pruning As a future research direction, we propose to go back to the task of unstructured pruning. The work was formulated from the start around semistructured pruning, and then, through selection, we identified in the Shapley values-based framework of SV-NUP a tool; the pruning awareness came later. However, during the experimentation, we observed how this extension can potentially bring positive improvements in the original field SV-NUP was built for, unstructured nonuniform pruning. We provide preliminary findings in Appendix A, where we offer a direct comparison of our method on the original unstructured task and the original SV-NUP.

Finer-granularity Shapley values for unstructured pruning While CoopQ [ZDH+25] inspired our pruning-aware approach, the method itself did not experiment with attention and MLP subsystems as fundamental units. In the case of nonuniform semistructured pruning, the introduction of a finer granularity did not provide a particular advantage. Quantization, however, follows different logic, since it does not remove weight, but reduces data type size uniformly; thus, we cannot exclude the possibility that finer granularity would not produce more positive results. Or it would, eventually, offer more flexibility.

Shapley values estimation for layerwise importance Another observation we make, related to future research on the use of Shapley values for pruning, concerns the SWSV approximator itself. As explained during the framework description, this method incorporates in the computation of the Shapley values the concept of sequentiality of the blocks, proving a certain robustness. We later created an extension of the SWSV approximator for decoder block subsystems, but overall, we did not aim to improve estimation efficiency. The introduction of the concept of sequentiality in the model, however, opens even more space for the estimation of Shapley values for layerwise importance. While SWSV adopts a local exhaustive search mechanism, it may be possible to translate a more complex search algorithm to the problem, such as Monte Carlo Tree Search [SM22]. Such an extension would also potentially address the robustness concerns reported in Section 7.

9 Conclusions

In conclusion, this research expands existing literature on both Shapley values-based pruning and 2:4 sparsity for Large Language models. As main contributions, we introduced a layerwise nonuniform sparsity criterion based on Shapley values and the SV-NUP framework, and we proceeded to enhance it with pruning-awareness and finer granularity. We here conclude by reconnecting to the main findings of this research.
Does the SWSV approximator identify a limited subset of significantly important layers? Does their exclusion from 2:4 pruning significantly improve the model performance? To answer the first research question, we confirm that the method identified some critical layers and that their preservation positively reflected on the pruning. In the most extreme case (magnitude pruning), 3 layers out of 32 account for 72% of the fully pruned model's perplexity. The Shapley values however, also report smoother curves that do not seem to correspond to the sharp outliers "cornerstone layers" identified by [ZDK24], but still positively impact the model pruning. While we

attribute these trends to massive activation emergence and the ME layer [STY+26], we still note that the Shapley values were able to identify the trend and take advantage of it to exclude from the pruning layers that, if included, would have strongly impacted the perplexity of Llama 2 7B when pruned with Wanda or Wanda++.

Do pruning-agnostic or pruning-aware Shapley values produce a better nonuniform, semistructured sparsity criterion?

Continuing with the second research question, we note that the pruning criteria based on pruning-aware Shapley values estimation performed overall better, as they were capable of identifying the most critical layer for the specific pruning method. The most noteworthy example we presented is the pruning of the last layer of Llama 2 7B, which causes a strong perplexity increase when performed with magnitude pruning but, at the same time, produces a significant performance improvement with Wanda and Wanda++.

Does splitting Shapley values between the decoder block's attention and feed-forward network subsystems produce a better nonuniform, semistructured sparsity criterion?

To finish with the third research question, we report that splitting Shapley values in the attention and feed-forward network did not reduce the effectiveness of the method, but it did not bring any substantial advantage, either, in terms of perplexity reduction. This, however, mainly applies to pruning-aware methods, because in some instances, the application of lower granularity to the original method noticeably reduces its ability to grasp layer importance even with a limited number of pruned layers, where up to that point all estimators and related criteria had performed robustly. Overall, it can still be considered a useful tool to increase the flexibility, in terms of the range of possible pruning ratios the model can be set to, but the pruning-aware estimation would in that case be a necessary prerequisite.

As a final reflection, we note that our findings do not provide a univocal indication of how many layers should be pruned. As introduced in Section 3, the choice of a stricter pruning method like 2:4 pruning is motivated by nonfunctional advantages that the more performance-preserving unstructured sparsity does not offer. While pruning in certain ranges appears more convenient, for example, at around 0.45 sparsity where the perplexity starts to grow faster, the final choice should be guided by the constraints of the target application. Our main goal was not to identify a single optimal threshold, but to introduce a form of pruning flexibility that preserves these hardware advantages while avoiding the purely linear, or worse, unpredictable, perplexity degradation resulting from an arbitrary pruning priority order. In this sense, our results support this flexibility, offering an adjustable tradeoff that a single fixed pruning ratio would not allow.

References

[BP26] Benzi Busigin and Yuval Pinter. Inside the llm word factory, 2026.

[CGT09] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers amp; Operations Research*, 36(5):1726–1730, May 2009.

[CZS24] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578, Dec 2024.

[DLBZ22] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022.

[DTD+26] Xuan Ding, Pengyu Tong, Ranjie Duan, Yunjian Zhang, Rui Sun, and Yao Zhu. Pruning as a cooperative game: Surrogate-assisted layer contribution estimation for large language models, Feb 2026.

[FA22] Elias Frantar and Dan Alistarh. Spdy: Accurate pruning with speedup guarantees, Aug 2022.

[FA23] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot, Mar 2023.

[GKD+21] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference, Jun 2021.

[GTA+23] 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, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023.

[HMD26] Younes Hourri, Mohammad Mozaffari, and Maryam Mehri Dehnavi. Patch: Learnable tile-level hybrid sparsity for llms, 2026.

[HPTD15] Song Han, Jeff Pool, John Tran, and William J. Dally, 2015.

[HS93] Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems*, volume 5, pages 164–171, 1993.

[HSSL24] Shwai He, Guoheng Sun, Zhenyu Shen, and Ang Li, Oct 2024.

[HSW+21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, Oct 2021.

[HZH+24] Yuezhou Hu, Kang Zhao, Weiyu Huang, Jianfei Chen, and Jun Zhu. Accelerating transformer pre-training with 2:4 sparsity, Oct 2024.

[IYT+25] Wataru Ikeda, Kazuki Yano, Ryosuke Takahashi, Jaesung Lee, Keigo Shibata, and Jun Suzuk, 2025.

[JTB+25] Geonhwa Jeong, Po-An Tsai, Abhimanyu R. Bambhaniya, Stephen W. Keckler, and Tushar Krishna. Enabling unstructured sparse acceleration on structured sparse accelerators, 2025.

[KORS25] Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From tokens to words: On the inner lexicon of llms, 2025.

[LDS89] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2, pages 598–605, 1989.

[LLGT25] Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference?, 2025.

[LT24] AI@Meta Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[MLP+21] Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks, Apr 2021.

[MMBR25] Lucas Maisonnave, Cyril Moineau, Olivier Bichler, and Fabrice Rastello. Precision where it matters: A novel spike aware mixed-precision quantization strategy for llama-based language models, Apr 2025.

[MXBS16] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[NVI23] Nvidia developer docs, 2023.

[Ras] Sebastian Raschka. LLMs-from-scratch: ch05/11_qwen3 – From-scratch implementation of Qwen3. GitHub repository.

[RPS21] Jay Rodge, Jeff Pool, and Abhishek Sawarkar. Accelerating inference with sparsity using the nvidia ampere architecture and nvidia tensorrt — nvidia technical blog, 2021.

[Sha53] Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker, editors, *Contributions to the Theory of Games II*, number 28 in Annals of Mathematics Studies, pages 307–317. Princeton University Press, Princeton, NJ, 1953.

[SLBK24] Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, May 2024.

[SPNJ25] Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones, Feb 2025.

[STY+26] Zeru Shi, Ruixiang Tang, Qifan Yang, Fan Yang, and Zhenting Wang, May 2026.

[SYCL25] Chuan Sun, Han Yu, Lizhen Cui, and Xiaoxiao Li. Efficient shapley value-based non-uniform pruning of large language models, May 2025.

[SZS+21] Wei Sun, Aojun Zhou, Sander Stuijk, Rob Wijnhoven, Andrew O. Nelson, hongsheng Li, and Henk Corporaal. Dominosearch: Find layer-wise fine-grained n:m sparse schemes from dense neural networks, Dec 2021.

[TvS21] William Timkey and Marten van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. *CoRR*, abs/2109.04404, 2021.

[VSP+25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N.Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention is all you need*, Aug 2025.

[WWP09] Samuel Williams, Andrew Waterman, and David Patterson. *Roofline: an insightful visual performance model for Floating-Point programs and multicore architectures*. September 2009.

[YDC+25] Zhiguo Yang, Changjian Deng, Qinke Chen, Zijing Zhou, and Jian Cheng. Lsa: Layer-wise sparsity allocation for large language model..., Oct 2025.

[YLY+25] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[YWZ+25] Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Jaiswal, Mykola Pechenizkiy, Yi Liang, and et al. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity, Jun 2025.

[YZG+25] Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M. Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, and et al., May 2025.

[ZDH+25] Junchen Zhao, Ali Derakhshan, Jayden Kana Hyman, Junhao Dong, Sangeetha Abdu Jyothi, and Ian Harris. Coopq: Cooperative game inspired layerwise mixed precision quantization for llms, Dec 2025.

[ZDK24] Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. Investigating layer importance in large language models. *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, page 469–479, 2024.

[ZLL+24] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024.

[SM22] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. Monte carlo tree search: a review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562, 2022.

Appendices

A Pruning-aware Shapley values for unstructured pruning

In the experiments of Section 6, we worked exclusively on nonuniform 2:4 pruning, as this was, from the start, the aim of the project. During implementation, however, we initially rebuilt the original unstructured sparsity task, which consists of the use of Equation 7 to compute the sparsity ratio and its application in the pruning stage to validate the baseline, since the source code for SV-NUP is not published. Since the pruning-awareness of the method revealed clearly different trends from vanilla SV-NUP, we applied them to the original unstructured task too and, for completeness, also computed the unstructured, pruning-aware values, with the same method used for 2:4 sparsity (see Section 5.1.3).

These findings, summarized in Table 4, show that **pruning-aware methods surpass full-removal across both methods (magnitude and Wanda pruning) and models on the original task; this applies both to the original and our reconstruction**. As in the semistructured pruning experiments, the most visible effect appears with magnitude pruning, which drops by around 5.5 in Llama 7b and 24 perplexity points in Llama 3B. Surprisingly, the Shapley values computed using 2:4 pruning report better results than values computed with 50% unstructured sparsity. This suggests that the application of pruning-aware values to unstructured pruning requires experimentation on which pruning ratio and method will achieve the best result; this concern was completely absent in 2:4 pruning, with its fixed 50% sparsity ratio.

Method	LLaMA2-7B	LLaMA3.2-3B
Dense	*5.472*	*7.813*
Magnitude	*16.030*	139.400 (*.412*)
w. SV-NUP (original)	*15.314*	*89.471*
w. SV-NUP (ours)	12.341	89.139
w. SV-NUP (unstr. 0.5)	9.999	72.238
w. SV-NUP (2:4)	**9.853**	**65.455**
Wanda	6.777 (*.774*)	12.694 (*.697*)
w. SV-NUP (original)	*6.796*	*12.158*
w. SV-NUP (ours)	6.882	12.553
w. SV-NUP (unstr. 0.5)	**6.778**	12.831
w. SV-NUP (2:4)	6.779	**12.149**

Table 4: Perplexity comparison across pruning methods and model sizes.

B Results tables

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	9.75	9.75	9.75	–	9.75	9.75	9.75	–	9.75	9.75	9.75
1	6	10.63	10.38	10.20	13	10.71	10.40	10.31	14	11.14	10.45	10.37
2	7	12.61	11.33	10.76	14	12.77	11.20	10.97	13	12.77	11.20	10.97
3	2	14.78	12.47	11.38	12	14.82	12.14	11.70	12	14.82	12.14	11.71
4	12	16.51	13.39	12.12	11	17.09	13.33	12.61	11	17.09	13.33	12.61
5	8	20.64	15.16	13.12	7	19.63	14.32	13.30	1	23.82	14.04	13.28
6	4	29.33	17.61	14.22	6	24.51	15.57	14.10	2	31.92	15.27	14.06
7	5	48.64	23.58	16.28	10	31.74	17.33	15.49	7	38.63	16.65	14.88
8	13	55.13	25.12	17.27	8	42.54	19.69	17.01	10	49.15	18.39	16.26
9	11	67.43	28.02	18.88	15	55.33	21.94	18.77	15	62.91	20.18	17.73
10	14	89.39	29.92	20.04	0	74.26	23.87	20.00	8	93.24	23.02	19.32
11	10	124.13	34.09	22.41	9	118.28	27.56	22.40	6	168.33	26.14	20.87
12	3	211.99	45.67	26.24	2	161.33	32.45	23.98	9	316.40	30.15	23.61
13	9	340.42	52.40	29.82	3	284.77	40.12	26.91	0	832.80	36.83	26.53
14	15	488.06	58.06	32.09	4	540.97	51.08	31.03	3	4974.22	46.27	30.57
15	0	900.64	83.75	38.14	5	900.64	83.75	38.54	4	3673.05	57.47	35.50
16	1	3288.67	93.05	43.41	1	3288.67	93.05	42.80	5	3288.67	93.05	43.44

Table 5: Sequential pruning results of the model Llama-3.2-1B using decoder block Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step.

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	7.81	7.81	7.81	–	7.81	7.81	7.81	–	7.81	7.81	7.81
1	8	8.01	7.96	7.92	23	8.01	7.98	7.97	25	8.12	8.05	7.99
2	10	8.28	8.14	8.05	21	8.22	8.16	8.14	23	8.32	8.23	8.16
3	23	8.49	8.31	8.21	26	8.69	8.47	8.40	21	8.58	8.44	8.35
4	11	8.86	8.52	8.37	22	8.94	8.67	8.58	17	8.90	8.67	8.56
5	22	9.12	8.72	8.55	18	9.26	8.93	8.81	18	9.22	8.91	8.79
6	17	9.49	8.96	8.77	20	9.60	9.19	9.06	22	9.52	9.16	9.00
7	12	10.09	9.25	8.98	25	10.10	9.46	9.28	26	10.12	9.44	9.26
8	18	10.44	9.53	9.22	17	10.52	9.71	9.51	24	10.63	9.68	9.48
9	13	11.26	9.88	9.49	19	11.00	10.04	9.78	16	11.09	9.94	9.71
10	21	11.68	10.14	9.72	24	11.58	10.29	10.02	20	11.57	10.25	9.99
11	19	12.39	10.54	10.07	16	12.21	10.61	10.30	19	12.21	10.61	10.30
12	9	13.24	11.02	10.31	15	13.07	10.93	10.62	15	13.07	10.93	10.63
13	20	13.86	11.38	10.63	14	14.25	11.38	11.02	13	14.24	11.28	10.92
14	16	14.95	11.80	10.98	11	15.25	11.68	11.28	10	14.98	11.56	11.13
15	14	17.44	12.43	11.46	12	16.65	12.14	11.62	12	16.28	11.95	11.44
16	7	21.22	13.78	11.87	10	18.02	12.58	11.89	11	17.85	12.43	11.76
17	3	23.07	14.68	12.20	13	21.06	13.13	12.29	14	21.06	13.13	12.29
18	15	27.30	15.44	12.75	9	23.34	13.75	12.59	9	23.34	13.75	12.62
19	5	34.52	16.20	13.23	8	24.64	14.09	12.82	5	25.47	14.34	12.91
20	6	48.92	18.97	13.91	0	27.95	14.59	13.12	4	28.51	15.09	13.22
21	4	70.21	21.08	14.77	3	30.65	15.47	13.49	8	33.73	15.91	13.52
22	2	112.61	22.97	15.66	27	36.13	16.57	14.09	6	44.72	16.60	14.06
23	26	122.40	23.49	15.90	5	44.08	17.89	14.58	3	57.92	18.24	14.56
24	25	127.60	23.80	16.17	6	58.38	19.11	15.21	2	107.31	19.37	15.34
25	24	134.14	24.26	16.51	7	92.37	23.17	16.06	1	156.30	20.95	16.41
26	27	170.67	26.15	17.15	4	137.37	26.10	17.16	0	261.66	23.38	18.07
27	1	228.68	27.60	18.46	2	201.40	29.09	18.49	7	294.16	29.82	19.62
28	0	387.54	31.97	20.13	1	387.54	31.97	20.25	27	387.54	31.97	20.21

Table 6: Sequential pruning results of the model Llama-3.2-3B using decoder block Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step.

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	5.47	5.47	5.47	–	5.47	5.47	5.47	–	5.47	5.47	5.47
1	12	5.52	5.52	5.51	29	5.61	5.58	5.56	24	5.51	5.51	5.51
2	14	5.59	5.59	5.57	24	5.66	5.62	5.59	11	5.58	5.58	5.57
3	24	5.65	5.64	5.62	12	5.70	5.67	5.64	12	5.64	5.63	5.61
4	11	5.73	5.71	5.68	14	5.78	5.74	5.70	0	5.76	5.65	5.61
5	23	5.81	5.78	5.74	18	5.87	5.82	5.77	14	5.84	5.73	5.68
6	13	5.89	5.86	5.80	11	5.96	5.90	5.84	18	5.93	5.81	5.76
7	21	5.98	5.94	5.88	23	6.05	5.97	5.91	23	6.01	5.88	5.83
8	18	6.08	6.03	5.96	19	6.15	6.06	5.99	19	6.11	5.97	5.91
9	22	6.19	6.13	6.05	15	6.27	6.15	6.06	15	6.23	6.06	5.98
10	10	6.29	6.22	6.11	17	6.40	6.24	6.14	17	6.36	6.15	6.06
11	8	6.42	6.32	6.19	10	6.50	6.33	6.20	29	6.53	6.26	6.15
12	27	6.51	6.39	6.26	20	6.61	6.42	6.29	27	6.62	6.33	6.22
13	15	6.65	6.50	6.34	21	6.74	6.52	6.38	7	6.71	6.40	6.27
14	7	6.78	6.59	6.41	27	6.83	6.60	6.46	28	6.84	6.50	6.35
15	19	6.92	6.70	6.51	13	6.97	6.72	6.55	26	7.00	6.61	6.44
16	26	7.05	6.80	6.60	22	7.11	6.84	6.65	20	7.13	6.71	6.53
17	20	7.22	6.94	6.71	16	7.31	7.00	6.78	22	7.29	6.84	6.63
18	17	7.38	7.04	6.82	7	7.44	7.11	6.85	10	7.43	6.94	6.71
19	9	7.56	7.18	6.89	9	7.64	7.23	6.94	21	7.57	7.06	6.82
20	25	7.72	7.31	7.00	28	7.81	7.35	7.02	13	7.74	7.20	6.90
21	28	7.90	7.41	7.08	25	7.99	7.49	7.13	8	7.93	7.32	7.00
22	16	8.23	7.59	7.22	26	8.27	7.65	7.24	9	8.15	7.48	7.08
23	29	8.51	7.80	7.33	8	8.51	7.80	7.33	25	8.35	7.62	7.19
24	6	8.80	8.05	7.48	6	8.80	8.05	7.47	30	8.75	7.81	7.34
25	5	9.06	8.32	7.58	5	9.06	8.32	7.58	16	9.05	8.01	7.49
26	30	9.35	8.55	7.74	4	9.44	8.87	7.77	6	9.37	8.28	7.64
27	4	9.66	9.17	7.96	0	9.65	8.87	7.76	31	12.02	8.07	7.67
28	3	10.32	9.73	8.30	30	10.01	9.16	7.93	5	12.49	8.33	7.78
29	2	11.26	10.71	8.88	3	10.69	9.69	8.27	1	14.89	8.42	7.84
30	31	16.08	10.24	8.81	2	11.70	10.62	8.84	4	16.47	8.88	8.09
31	1	26.02	11.43	9.51	1	15.82	12.19	9.62	3	21.02	9.52	8.55
32	0	37.78	11.55	9.55	31	37.76	11.55	9.55	2	37.76	11.55	9.54

Table 7: Sequential pruning results of the model Llama-2-7b using decoder block Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step.

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	12.66	12.66	12.66	–	12.66	12.66	12.66	–	12.66	12.66	12.66
1	14	13.16	12.91	12.88	14	13.16	12.91	12.88	13	13.43	12.92	12.90
2	13	13.90	13.17	13.11	12	13.94	13.20	13.14	14	13.90	13.17	13.11
3	12	14.82	13.48	13.38	25	14.85	13.73	13.63	12	14.82	13.48	13.37
4	15	15.94	13.84	13.66	13	15.79	14.03	13.88	15	15.94	13.84	13.66
5	8	16.87	14.29	14.00	16	17.13	14.51	14.30	10	18.19	14.23	13.96
6	7	18.51	14.92	14.42	15	18.44	14.89	14.63	23	19.61	14.75	14.45
7	9	21.90	15.41	14.79	26	20.45	15.81	15.49	24	21.43	15.31	14.96
8	10	27.21	16.15	15.22	8	21.78	16.34	15.83	9	24.46	15.84	15.37
9	11	31.62	17.28	15.86	24	24.01	16.98	16.42	18	29.10	16.42	15.84
10	18	38.78	17.86	16.38	23	26.82	17.66	17.00	22	33.89	17.07	16.45
11	17	48.42	18.93	17.16	7	30.13	18.43	17.50	8	37.84	17.82	16.92
12	16	58.74	20.16	17.98	11	35.01	19.42	18.11	11	44.83	18.94	17.59
13	23	65.55	21.04	18.73	22	41.08	20.22	18.76	16	53.61	20.05	18.40
14	3	79.94	22.90	19.70	17	51.58	21.43	19.62	25	60.69	21.05	19.13
15	19	102.84	24.43	20.84	10	69.62	22.53	20.42	20	85.43	22.16	20.06
16	5	171.75	28.12	22.82	6	88.76	25.40	21.95	21	123.02	23.56	21.27
17	25	190.90	29.50	23.80	18	116.20	26.73	22.87	17	171.79	25.38	22.60
18	4	270.23	34.88	25.97	9	159.04	28.73	23.77	19	251.06	27.49	24.19
19	24	306.08	36.36	26.92	21	228.00	30.53	25.40	7	341.74	29.57	25.38
20	6	406.16	45.82	31.21	20	386.75	32.72	26.93	26	438.47	30.88	26.49
21	22	540.58	47.50	32.44	19	601.50	35.17	28.87	5	758.84	34.59	28.62
22	21	864.54	50.79	34.71	4	856.21	42.84	31.87	4	1108.14	40.41	30.66
23	20	1923.42	55.08	37.05	2	784.11	45.80	33.48	6	1389.99	51.23	35.44
24	26	2466.81	57.57	38.35	3	1088.83	54.60	37.03	3	2466.81	57.57	38.48
25	27	4159.19	57.23	39.46	27	1571.78	54.06	38.66	2	1987.39	64.50	41.90
26	2	3103.44	62.68	42.69	5	3103.44	62.68	42.71	1	2967.59	72.54	45.01
27	0	15146.66	97.28	59.87	1	5017.07	69.99	45.81	27	5017.07	69.99	46.48
28	1	25307.58	132.65	70.55	0	25307.58	132.65	73.87	0	25307.58	132.65	72.47

Table 8: Sequential pruning results of the model Qwen3-0.6B-Base using decoder block Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step.

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	7.90	7.90	7.90	–	7.90	7.90	7.90	–	7.90	7.90	7.90
1	16	7.96	7.94	7.95	1	8.00	7.94	7.93	14	7.98	7.95	7.94
2	14	8.08	8.01	8.00	16	8.06	7.98	7.98	3	8.21	7.96	7.97
3	17	8.23	8.08	8.07	10	8.12	8.02	8.02	16	8.32	8.03	8.03
4	18	8.41	8.17	8.15	20	8.27	8.12	8.10	15	8.45	8.11	8.10
5	15	8.58	8.27	8.23	34	8.88	8.45	8.37	17	8.63	8.19	8.17
6	20	8.86	8.39	8.32	14	9.07	8.54	8.45	10	8.77	8.27	8.22
7	21	9.02	8.51	8.42	17	9.29	8.63	8.52	11	9.02	8.37	8.31
8	19	9.32	8.66	8.53	19	9.62	8.78	8.62	13	9.46	8.49	8.38
9	11	9.59	8.77	8.61	18	9.88	8.89	8.70	12	9.82	8.62	8.48
10	13	10.15	8.92	8.71	11	10.20	9.01	8.80	18	10.19	8.75	8.58
11	8	10.52	9.08	8.81	21	10.33	9.15	8.91	2	14.81	8.79	8.60
12	12	10.99	9.28	8.92	15	10.77	9.29	9.02	8	16.96	8.99	8.73
13	3	11.70	9.33	8.96	8	11.19	9.47	9.12	19	18.40	9.19	8.84
14	22	12.21	9.55	9.13	22	11.67	9.69	9.28	25	19.33	9.34	8.97
15	2	20.15	9.62	9.16	12	12.29	9.89	9.41	20	21.02	9.53	9.10
16	26	21.04	9.80	9.32	9	13.14	10.14	9.54	4	25.47	9.69	9.22
17	10	22.94	9.98	9.43	13	14.85	10.49	9.63	21	27.24	9.89	9.36
18	7	28.24	10.34	9.59	26	15.54	10.70	9.80	9	33.31	10.21	9.51
19	25	30.37	10.56	9.76	25	16.61	10.93	9.97	26	35.07	10.41	9.67
20	23	32.68	10.90	10.03	29	17.69	11.22	10.18	5	40.63	10.73	9.87
21	4	41.14	11.15	10.18	7	19.94	11.72	10.39	23	44.27	11.05	10.12
22	27	44.45	11.43	10.41	27	21.49	12.05	10.62	27	47.12	11.31	10.33
23	28	48.97	11.79	10.69	23	23.68	12.48	10.92	28	51.47	11.65	10.59
24	9	58.51	12.36	10.94	5	25.81	12.89	11.16	1	39.52	11.87	10.80
25	32	61.93	12.71	11.22	30	28.17	13.32	11.46	29	42.10	12.19	11.06
26	5	72.52	13.17	11.53	28	31.99	13.82	11.81	24	47.54	12.58	11.34
27	29	77.30	13.57	11.84	24	35.77	14.33	12.15	30	51.91	13.00	11.66
28	31	85.63	13.99	12.16	6	40.81	15.04	12.53	22	57.38	13.52	12.02
29	30	96.69	14.48	12.52	32	43.57	15.58	12.89	7	71.35	14.46	12.45
30	24	128.95	15.03	12.90	33	48.49	16.28	13.26	31	81.89	14.91	12.79
31	33	155.25	15.61	13.24	31	61.65	16.95	13.61	32	91.79	15.36	13.14
32	1	106.07	15.96	13.44	4	72.40	17.66	13.96	33	106.07	15.96	13.48
33	34	151.31	17.26	13.82	3	83.99	18.04	14.25	0	304.47	16.97	14.17
34	35	164.32	16.26	14.04	0	129.56	19.04	15.12	6	365.85	18.03	14.95
35	6	210.92	17.74	14.62	35	176.52	17.90	15.31	34	550.89	19.51	15.39
36	0	559.87	18.42	15.56	2	559.87	18.42	15.65	35	559.87	18.42	15.58

Table 9: Sequential pruning results of the model Qwen3-4B-Base using decoder block Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step.

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	9.75	9.75	9.75	–	9.75	9.75	9.75	–	9.75	9.75	9.75
1	2a	10.25	10.06	9.86	6m	10.14	10.07	10.01	13a	9.98	9.88	9.85
2	10a	10.66	10.29	10.02	2m	10.66	10.51	10.39	14a	10.28	10.04	9.96
3	11a	11.07	10.50	10.16	14a	10.98	10.67	10.52	1a	11.09	10.29	10.17
4	12a	11.44	10.69	10.29	13a	11.25	10.80	10.62	12a	11.45	10.50	10.28
5	13a	11.79	10.83	10.39	5m	11.92	11.31	11.02	11a	11.90	10.75	10.43
6	6a	12.75	11.16	10.62	3m	13.12	12.10	11.63	2a	14.21	11.09	10.57
7	8a	14.67	11.70	10.95	12a	13.50	12.29	11.76	15a	15.26	11.61	10.87
8	7a	17.92	12.32	11.29	7m	14.54	12.87	12.25	10a	16.41	11.90	11.10
9	14a	18.70	12.50	11.44	4m	16.10	14.23	13.17	0a	18.38	12.26	11.36
10	9a	23.02	13.25	11.97	0a	17.42	14.98	13.62	14m	20.90	13.00	11.92
11	6m	24.60	13.75	12.37	13m	18.81	15.74	14.24	13m	23.42	13.84	12.54
12	7m	26.63	14.67	12.81	15a	21.02	16.90	14.72	12m	26.63	14.74	13.17
13	15a	29.49	15.97	13.29	14m	24.59	18.02	15.45	7m	28.89	15.35	13.67
14	4a	34.55	16.63	13.58	11a	26.36	18.40	15.73	9a	34.21	15.96	14.14
15	3a	59.19	18.34	13.99	8m	29.95	19.62	16.58	8m	40.54	16.88	14.80
16	5m	64.83	20.60	14.72	12m	33.74	20.87	17.47	2m	42.10	18.16	15.45
17	1a	122.23	21.41	15.25	10a	38.97	21.40	17.93	6m	50.12	19.36	16.10
18	0a	142.34	22.38	15.73	11m	45.94	23.32	19.25	7a	56.66	20.31	16.58
19	4m	157.05	24.96	17.06	9m	54.26	25.04	20.55	11m	66.13	22.09	17.76
20	5a	190.91	33.35	18.40	9a	67.52	26.46	21.41	9m	76.36	23.72	18.99
21	12m	204.18	35.49	19.48	15m	72.20	25.62	22.11	8a	88.48	26.13	19.89
22	8m	270.88	37.72	20.49	10m	85.83	27.86	23.84	10m	102.58	28.39	21.51
23	13m	285.20	39.59	21.43	1m	141.15	30.91	25.82	5m	121.40	31.12	22.74
24	3m	316.83	47.49	24.08	8a	150.79	33.99	26.92	1m	218.77	33.37	24.07
25	9m	385.33	50.90	25.81	0m	219.42	41.74	31.08	3m	281.49	37.71	26.42
26	11m	441.74	54.70	27.76	7a	269.56	43.49	32.43	6a	468.53	41.30	27.44
27	14m	479.12	56.51	28.87	3a	486.52	49.11	33.94	15m	596.84	39.25	28.09
28	10m	519.03	60.90	31.09	2a	763.67	57.83	34.79	4m	580.52	45.89	31.73
29	2m	529.59	63.57	33.79	4a	832.70	68.18	36.18	3a	1081.40	50.34	33.46
30	15m	729.18	63.13	34.49	6a	1504.20	75.65	38.39	4a	910.63	56.48	34.49
31	0m	1056.05	83.74	39.77	1a	2179.41	68.34	39.45	0m	2179.41	68.34	39.98
32	1m	3288.67	93.05	44.00	5a	3288.67	93.05	42.92	5a	3288.67	93.05	43.48

Table 10: Sequential pruning results of the model Llama-3.2-1B using attention and feed-forward networks Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step (a = attention, m = feed-forward network).

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	7.81	7.81	7.81	–	7.81	7.81	7.81	–	7.81	7.81	7.81
1	2a	8.18	7.87	7.85	2m	7.94	7.93	7.92	26a	7.89	7.88	7.85
2	3a	8.44	7.97	7.88	26a	8.02	7.99	7.96	21a	7.92	7.91	7.87
3	18a	8.51	8.01	7.92	18a	8.08	8.04	8.00	23a	7.97	7.95	7.90
4	4a	8.96	8.09	7.94	21a	8.11	8.07	8.01	20a	8.02	7.98	7.92
5	26a	9.04	8.15	7.98	23a	8.15	8.10	8.04	18a	8.09	8.02	7.96
6	20a	9.10	8.19	8.01	20a	8.21	8.13	8.07	22a	8.16	8.05	7.99
7	17a	9.18	8.23	8.04	22a	8.29	8.17	8.09	24a	8.25	8.09	8.02
8	5a	10.41	8.32	8.08	24a	8.39	8.20	8.13	17a	8.34	8.13	8.06
9	21a	10.46	8.35	8.10	7m	8.51	8.34	8.24	19a	8.42	8.17	8.09
10	8a	11.54	8.43	8.14	17a	8.60	8.38	8.28	16a	8.57	8.23	8.14
11	16a	11.79	8.49	8.19	19a	8.69	8.42	8.31	25m	8.86	8.40	8.29
12	23a	11.84	8.53	8.22	5m	8.86	8.59	8.47	25a	8.92	8.48	8.32
13	6a	14.79	8.62	8.29	9m	9.01	8.73	8.58	15a	9.10	8.55	8.39
14	10a	16.10	8.71	8.35	8m	9.17	8.88	8.70	13m	9.34	8.67	8.50
15	12a	16.72	8.82	8.42	0a	9.36	9.00	8.76	14a	9.54	8.78	8.60
16	19a	16.90	8.87	8.46	6m	9.59	9.22	8.95	4a	9.66	8.83	8.63
17	22a	17.07	8.91	8.49	25m	9.92	9.40	9.11	2a	10.25	8.88	8.67
18	15a	17.92	9.00	8.58	3m	10.33	9.89	9.47	12m	10.58	9.01	8.77
19	24a	18.12	9.04	8.62	25a	10.36	9.94	9.50	11m	10.90	9.13	8.88
20	25a	18.22	9.10	8.66	16a	10.58	10.01	9.56	10m	11.25	9.26	9.00
21	14a	19.74	9.26	8.79	10m	10.80	10.20	9.72	1a	12.45	9.34	9.04
22	11a	23.72	9.43	8.92	11m	11.10	10.42	9.89	10a	12.88	9.46	9.11
23	13a	28.32	9.68	9.08	26m	11.62	10.61	10.06	16m	13.35	9.66	9.29
24	8m	29.87	9.84	9.15	21m	11.93	10.84	10.27	23m	13.59	9.83	9.46
25	11m	30.02	10.00	9.27	23m	12.20	11.03	10.46	12a	14.02	10.00	9.57
26	7a	40.72	10.39	9.42	12m	12.68	11.19	10.63	17m	14.52	10.23	9.78
27	9m	44.46	10.66	9.53	22m	12.98	11.42	10.84	15m	15.35	10.48	10.00
28	10m	46.89	10.90	9.66	13m	13.54	11.68	11.05	18m	15.94	10.75	10.24
29	9a	50.46	11.47	9.80	15a	14.24	11.78	11.16	11a	17.11	10.96	10.38
30	27a	55.42	12.20	9.95	14a	14.74	11.87	11.25	9m	17.83	11.14	10.52
31	13m	57.72	12.49	10.11	18m	15.24	12.17	11.54	13a	19.22	11.44	10.70
32	23m	58.97	12.70	10.30	16m	15.87	12.45	11.78	22m	19.73	11.71	10.95
33	22m	60.40	13.03	10.51	17m	16.52	12.76	12.06	26m	21.22	11.88	11.11
34	7m	62.72	14.29	10.75	19m	17.33	13.12	12.39	21m	22.01	12.18	11.37
35	1a	75.83	14.05	10.84	20m	17.98	13.47	12.71	24m	23.31	12.43	11.58
36	12m	81.19	14.61	11.03	15m	19.32	13.86	13.06	14m	25.54	12.83	11.90
37	17m	84.57	14.99	11.32	14m	20.91	14.35	13.43	8a	28.04	12.94	11.97
38	14m	92.58	15.50	11.62	24m	22.01	14.59	13.67	0a	29.52	13.03	12.03
39	19m	95.90	16.06	11.98	12a	23.05	14.93	13.85	19m	31.11	13.45	12.41
40	21m	97.25	16.52	12.27	11a	25.11	15.34	14.07	5a	44.89	13.67	12.51
41	18m	101.69	17.11	12.60	13a	27.34	15.88	14.29	20m	47.11	14.04	12.84
42	20m	104.33	17.63	12.95	10a	29.69	16.39	14.39	27a	58.56	15.22	13.08
43	0a	108.60	17.78	13.07	27a	39.71	18.05	14.68	3m	65.70	15.75	13.32
44	15m	116.59	18.26	13.45	0m	46.45	19.44	15.65	6m	72.20	16.36	13.63
45	16m	128.27	18.84	13.84	9a	51.22	20.34	15.72	5m	73.74	17.04	14.01
46	5m	134.44	19.42	14.22	27m	44.41	19.53	15.94	9a	83.60	17.75	14.15
47	6m	134.28	20.84	14.70	4m	48.79	21.60	17.27	3a	98.71	18.37	14.33
48	4m	142.06	22.95	15.34	8a	54.94	22.01	17.19	6a	123.40	18.68	14.49
49	25m	146.35	23.63	15.52	3a	58.97	22.97	17.44	8m	139.02	19.21	14.78
50	26m	154.40	23.89	15.63	1a	67.79	23.53	17.93	4m	150.03	21.13	15.45
51	24m	161.83	24.51	15.89	6a	77.09	23.96	18.12	7m	152.61	22.88	16.11
52	3m	176.99	26.07	16.61	4a	88.55	24.16	18.17	7a	176.99	26.07	16.58
53	27m	198.46	24.73	16.92	5a	117.07	25.37	18.22	1m	228.46	27.00	17.48
54	2m	213.78	26.39	17.79	7a	173.75	28.54	18.66	2m	215.10	29.72	18.61
55	0m	244.68	29.24	18.95	2a	244.68	29.24	18.91	0m	323.55	34.23	20.00
56	1m	387.54	31.97	20.15	1m	387.54	31.97	20.09	27m	387.54	31.97	20.16

Table 11: Sequential pruning results of the model Llama-3.2-3B using attention and feed-forward networks Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step (a = attention, m = feed-forward network).

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	5.47	5.47	5.47	–	5.47	5.47	5.47	–	5.47	5.47	5.47
1	4a	5.50	5.50	5.49	0m	5.50	5.48	5.48	24a	5.47	5.47	5.47
2	24a	5.50	5.50	5.49	29a	5.54	5.53	5.51	0a	5.59	5.48	5.47
3	27a	5.51	5.51	5.50	30a	5.57	5.56	5.53	11a	5.62	5.50	5.49
4	7a	5.54	5.53	5.51	24a	5.57	5.56	5.53	23a	5.63	5.51	5.51
5	21a	5.56	5.54	5.52	23a	5.58	5.57	5.54	21a	5.65	5.52	5.52
6	23a	5.58	5.56	5.54	12m	5.61	5.60	5.57	7a	5.68	5.54	5.53
7	18a	5.60	5.58	5.56	21a	5.63	5.61	5.58	12m	5.71	5.57	5.56
8	12a	5.64	5.61	5.58	20a	5.65	5.63	5.60	27a	5.73	5.59	5.57
9	11a	5.69	5.65	5.61	27a	5.66	5.65	5.61	22a	5.75	5.61	5.59
10	14a	5.75	5.70	5.65	16a	5.70	5.69	5.64	29a	5.80	5.65	5.62
11	22a	5.77	5.72	5.67	22a	5.73	5.71	5.66	26a	5.84	5.67	5.64
12	8a	5.85	5.78	5.71	18a	5.77	5.75	5.69	20a	5.87	5.70	5.66
13	19a	5.89	5.82	5.74	17a	5.80	5.79	5.72	17a	5.90	5.73	5.68
14	16a	5.96	5.89	5.79	10m	5.85	5.83	5.75	19a	5.94	5.76	5.71
15	20a	6.00	5.93	5.82	19a	5.90	5.87	5.78	18a	5.98	5.80	5.73
16	30a	6.04	5.96	5.84	15a	5.94	5.92	5.82	13m	6.03	5.83	5.77
17	9a	6.13	6.03	5.88	14a	5.99	5.97	5.85	28a	6.07	5.87	5.79
18	6a	6.24	6.13	5.94	14m	6.03	6.00	5.89	25a	6.13	5.92	5.82
19	5a	6.31	6.20	5.98	28a	6.07	6.04	5.92	6a	6.22	5.97	5.86
20	15a	6.38	6.30	6.04	11a	6.13	6.06	5.94	10m	6.26	6.01	5.89
21	28a	6.43	6.35	6.07	13m	6.17	6.08	5.97	15m	6.35	6.07	5.94
22	13a	6.55	6.47	6.14	29m	6.26	6.14	6.02	14m	6.43	6.12	5.98
23	12m	6.56	6.46	6.14	11m	6.33	6.20	6.06	14a	6.48	6.16	6.02
24	26a	6.62	6.50	6.17	12a	6.36	6.23	6.09	15a	6.52	6.19	6.05
25	10m	6.66	6.53	6.19	25a	6.42	6.28	6.12	30a	6.58	6.24	6.07
26	29a	6.71	6.58	6.22	26a	6.48	6.32	6.15	8a	6.68	6.32	6.12
27	10a	6.88	6.75	6.29	15m	6.55	6.36	6.19	19m	6.74	6.37	6.17
28	17a	6.91	6.80	6.33	7m	6.61	6.41	6.23	18m	6.81	6.43	6.21
29	13m	6.95	6.82	6.35	9m	6.73	6.47	6.27	16a	6.86	6.48	6.26
30	14m	6.99	6.86	6.38	18m	6.80	6.53	6.32	12a	6.94	6.56	6.31
31	11m	7.04	6.90	6.41	19m	6.88	6.58	6.37	9m	7.04	6.61	6.35
32	8m	7.09	6.89	6.44	9a	6.93	6.63	6.40	30m	7.32	6.70	6.44
33	25m	7.18	6.95	6.49	17m	7.02	6.69	6.45	11m	7.40	6.76	6.49
34	24m	7.25	7.02	6.54	10a	7.08	6.76	6.50	10a	7.49	6.85	6.54
35	23m	7.34	7.10	6.60	24m	7.12	6.81	6.54	17m	7.62	6.92	6.59
36	26m	7.43	7.17	6.66	7a	7.20	6.86	6.56	29m	7.75	7.00	6.65
37	15m	7.51	7.23	6.71	13a	7.27	6.94	6.62	9a	7.91	7.18	6.70
38	22m	7.61	7.33	6.78	5a	7.35	7.02	6.64	28m	8.00	7.23	6.74
39	18m	7.70	7.40	6.85	6a	7.52	7.23	6.71	8m	8.04	7.22	6.78
40	21m	7.81	7.50	6.92	8a	7.69	7.39	6.77	24m	8.11	7.28	6.83
41	17m	7.92	7.58	6.99	8m	7.78	7.39	6.81	13a	8.21	7.36	6.88
42	7m	7.99	7.66	7.02	23m	7.87	7.48	6.88	16m	8.40	7.46	6.97
43	19m	8.09	7.74	7.09	25m	7.98	7.55	6.94	23m	8.51	7.55	7.04
44	27m	8.16	7.81	7.16	28m	8.11	7.63	6.99	26m	8.61	7.63	7.11
45	31a	8.38	8.01	7.21	22m	8.23	7.73	7.08	0m	8.63	7.67	7.12
46	0a	8.68	8.01	7.21	27m	8.32	7.80	7.15	27m	8.74	7.75	7.19
47	1a	9.13	8.01	7.23	20m	8.45	7.91	7.24	25m	8.85	7.85	7.27
48	25a	9.16	8.04	7.25	26m	8.60	8.03	7.32	20m	9.01	7.95	7.37
49	20m	9.31	8.16	7.35	16m	8.80	8.14	7.41	22m	9.15	8.07	7.46
50	9m	9.41	8.26	7.40	21m	8.94	8.27	7.51	7m	9.15	8.09	7.48
51	28m	9.63	8.35	7.44	6m	9.03	8.42	7.56	21m	9.29	8.21	7.58
52	6m	9.77	8.59	7.50	31a	9.35	8.67	7.59	31m	11.91	8.01	7.62
53	16m	9.98	8.71	7.59	4a	9.51	8.91	7.66	31a	11.98	8.02	7.62
54	29m	10.32	8.84	7.65	4m	9.64	9.01	7.73	5a	12.36	8.17	7.66
55	3a	10.87	8.96	7.72	5m	9.80	9.35	7.85	6m	12.41	8.31	7.70
56	5m	11.15	9.14	7.82	2a	10.05	9.32	7.96	3a	12.83	8.37	7.75
57	3m	11.72	9.44	7.96	3m	10.28	9.95	8.22	5m	13.07	8.43	7.81
58	2a	13.36	9.83	8.17	3a	10.94	10.15	8.36	1a	14.13	8.43	7.84
59	2m	15.13	10.52	8.55	2m	11.41	10.86	8.72	4a	15.01	8.69	7.91
60	30m	16.27	10.90	8.72	1a	12.48	11.46	9.00	1m	17.68	8.81	7.99
61	4m	17.30	11.83	9.23	1m	14.54	12.60	9.52	2a	21.40	9.10	8.18
62	1m	25.47	12.74	9.63	30m	15.42	13.07	9.73	4m	22.79	9.43	8.44
63	0m	17.42	12.96	9.63	0a	17.43	12.96	9.68	2m	31.04	10.14	8.91
64	31m	37.76	11.55	9.52	31m	37.78	11.55	9.56	3m	37.76	11.55	9.53

Table 12: Sequential pruning results of the model Llama-2-7b using attention and feed-forward networks Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step (a = attention, m = feed-forward network).

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	12.66	12.66	12.66	–	12.66	12.66	12.66	–	12.66	12.66	12.66
1	27a	12.95	12.86	12.83	12a	12.97	12.77	12.76	9a	13.18	12.81	12.79
2	10a	13.26	12.95	12.91	14a	13.40	12.92	12.88	25a	13.55	12.97	12.93
3	16a	13.69	13.09	13.04	14m	13.56	13.03	12.98	23a	13.90	13.08	13.04
4	12a	14.16	13.22	13.15	15m	13.94	13.20	13.13	24a	14.43	13.21	13.15
5	14a	14.75	13.41	13.29	12m	14.39	13.38	13.30	10a	14.88	13.32	13.24
6	7a	15.34	13.54	13.40	8a	14.76	13.51	13.40	26a	15.64	13.65	13.49
7	1a	15.66	13.69	13.52	13m	15.17	13.66	13.53	22a	16.55	13.84	13.64
8	13a	16.67	13.87	13.67	11m	15.77	13.87	13.71	13m	17.08	14.00	13.79
9	2a	17.48	14.09	13.87	17m	16.47	14.13	13.94	14m	17.49	14.15	13.93
10	8a	18.21	14.27	14.01	25a	17.00	14.31	14.11	15m	18.18	14.34	14.11
11	3a	18.79	14.36	14.06	16m	18.09	14.62	14.39	12a	18.88	14.49	14.23
12	9a	21.12	14.57	14.23	27a	18.65	14.85	14.58	14a	19.82	14.66	14.35
13	14m	21.43	14.73	14.35	10a	19.63	15.03	14.72	13a	20.94	14.85	14.49
14	4a	24.60	15.04	14.54	26a	20.78	15.45	15.01	8a	21.73	15.01	14.59
15	13m	25.43	15.19	14.67	6m	22.05	15.85	15.34	12m	22.70	15.25	14.79
16	18a	28.61	15.46	14.87	25m	22.75	16.31	15.74	7a	24.46	15.44	14.92
17	15a	32.21	15.81	15.11	7m	24.28	16.78	16.12	27a	25.58	15.79	15.12
18	15m	32.87	15.94	15.25	18m	25.35	17.12	16.42	18a	29.71	16.00	15.28
19	11m	34.44	16.26	15.49	13a	26.61	17.51	16.60	20a	34.96	16.29	15.50
20	12m	36.99	16.61	15.73	24a	28.12	17.77	16.78	16a	38.37	16.65	15.76
21	17a	45.83	17.09	16.10	20m	30.06	18.18	17.22	15a	43.87	17.02	15.98
22	23a	49.08	17.32	16.29	26m	31.62	18.77	17.83	21a	51.94	17.48	16.31
23	5a	74.71	17.89	16.63	23a	33.44	18.99	17.98	11m	55.77	17.80	16.57
24	19a	85.13	18.36	16.97	24m	35.13	19.53	18.47	19a	69.96	18.38	16.97
25	8m	98.67	18.96	17.37	23m	37.29	20.08	18.97	17m	72.11	18.72	17.32
26	24a	108.51	19.16	17.52	16a	38.93	20.48	19.24	11a	82.05	19.24	17.73
27	25a	118.05	19.64	17.82	22m	41.47	21.04	19.69	18m	86.44	19.77	18.10
28	9m	128.80	20.22	18.17	5m	44.85	21.77	20.28	16m	99.68	20.26	18.55
29	7m	130.86	21.02	18.63	22a	49.45	22.12	20.57	20m	107.26	20.75	19.09
30	11a	159.37	22.30	19.21	21m	53.69	22.72	21.19	23m	115.40	21.39	19.65
31	18m	175.22	22.88	19.66	9a	59.25	23.06	21.51	21m	126.54	22.00	20.31
32	17m	183.97	23.51	19.99	1a	62.56	23.40	21.63	24m	133.14	22.70	20.90
33	26a	204.80	24.14	20.37	19m	70.22	24.12	22.32	17a	176.84	24.07	21.69
34	21a	277.03	24.97	21.00	15a	81.11	24.70	22.69	22m	200.72	24.78	22.34
35	6m	312.88	26.63	21.95	7a	90.73	25.11	22.90	10m	226.91	25.37	22.79
36	22a	408.20	27.11	22.20	8m	103.16	26.18	23.49	8m	279.66	26.38	23.34
37	20a	779.69	28.12	22.81	9m	110.29	27.52	24.07	19m	313.49	27.30	24.15
38	10m	819.83	29.97	23.81	11a	127.03	29.01	24.91	4a	349.39	27.88	24.49
39	6a	927.32	33.02	24.96	4m	152.11	31.17	26.56	9m	388.65	29.04	25.17
40	5m	907.29	36.09	26.08	21a	202.29	32.12	27.36	25m	410.90	29.75	25.73
41	23m	937.15	37.18	26.88	3m	251.17	35.14	29.12	1a	470.46	30.24	25.92
42	19m	1118.86	38.69	27.95	10m	289.33	37.96	30.25	3a	510.27	31.04	26.29
43	25m	1198.66	39.51	28.54	6a	333.48	40.34	32.19	5a	1004.52	32.23	27.13
44	22m	1321.45	40.59	29.41	17a	454.11	42.78	33.78	7m	1128.84	34.71	27.93
45	24m	1404.42	41.88	30.14	18a	590.40	44.00	34.38	2m	954.00	36.45	29.08
46	16m	1809.35	43.54	30.87	0m	714.15	47.42	35.94	5m	1086.56	39.79	30.67
47	21m	1874.74	44.89	31.94	4a	767.03	49.99	36.85	6m	1173.06	43.08	32.58
48	20m	1983.89	46.34	33.13	2a	880.39	51.84	36.67	6a	1317.04	47.24	34.46
49	4m	3014.89	52.91	35.80	2m	865.86	56.12	39.03	26m	1414.41	47.69	34.91
50	3m	5432.01	62.37	39.20	19a	1255.01	59.98	40.29	2a	1755.09	50.91	35.49
51	26m	5853.63	63.04	39.66	20a	1664.67	62.03	41.54	3m	2205.97	57.16	38.52
52	0a	31715.65	85.87	52.73	3a	1843.90	67.56	44.11	4m	2832.98	67.32	42.10
53	27m	44966.21	82.15	53.18	5a	3848.27	76.52	45.23	1m	3690.66	73.84	46.61
54	0m	58958.50	94.93	58.03	27m	5292.42	72.81	45.87	0m	4978.96	81.00	50.26
55	2m	19563.51	106.39	63.11	1m	6744.17	75.74	51.39	27m	6744.17	75.74	50.91
56	1m	25307.58	132.65	70.74	0a	25307.58	132.65	71.82	0a	25307.58	132.65	70.37

Table 13: Sequential pruning results of the model Qwen3-0.6B-Base using attention and feed-forward networks Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step (a = attention, m = feed-forward network).

	Full removal				Magnitude				Wanda			
Step	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++	Pruned	Mag.	Wanda	W++
0	–	7.90	7.90	7.90	–	7.90	7.90	7.90	–	7.90	7.90	7.90
1	11a	7.92	7.91	7.90	1m	7.96	7.91	7.91	11a	7.92	7.91	7.90
2	10a	7.94	7.92	7.91	34a	8.12	7.95	7.95	34a	8.08	7.95	7.94
3	9a	7.99	7.93	7.92	15m	8.17	8.00	7.99	8a	8.12	7.96	7.96
4	14a	8.04	7.96	7.95	10a	8.20	8.01	8.00	4a	8.14	7.98	7.97
5	17a	8.13	7.98	7.97	18m	8.26	8.06	8.04	14a	8.19	8.01	7.99
6	18a	8.23	8.03	8.01	8a	8.29	8.07	8.06	10a	8.21	8.02	8.00
7	32a	8.31	8.06	8.03	16a	8.33	8.10	8.10	16a	8.28	8.05	8.03
8	16a	8.39	8.09	8.07	17m	8.39	8.14	8.14	32a	8.36	8.09	8.06
9	26a	8.46	8.13	8.10	20m	8.44	8.19	8.18	12a	8.47	8.14	8.11
10	15a	8.62	8.18	8.13	14m	8.52	8.24	8.22	9a	8.54	8.15	8.12
11	28a	8.72	8.22	8.17	16m	8.59	8.29	8.25	2a	8.57	8.16	8.14
12	21a	8.86	8.27	8.20	19m	8.68	8.34	8.30	29a	8.63	8.19	8.16
13	25a	8.96	8.31	8.23	29a	8.74	8.37	8.33	25a	8.72	8.22	8.18
14	5a	9.02	8.32	8.25	21a	8.80	8.42	8.36	3a	8.78	8.24	8.19
15	8a	9.10	8.34	8.27	11a	8.87	8.44	8.38	15a	8.94	8.28	8.22
16	20a	9.24	8.41	8.33	12m	9.00	8.53	8.45	31a	9.06	8.33	8.25
17	16m	9.30	8.47	8.37	33a	9.16	8.60	8.51	14m	9.15	8.36	8.28
18	12a	9.49	8.53	8.43	22a	9.30	8.68	8.58	26a	9.28	8.40	8.31
19	4a	9.55	8.57	8.45	9a	9.37	8.70	8.60	5a	9.37	8.42	8.32
20	2m	13.26	8.58	8.47	32a	9.48	8.74	8.65	18m	9.45	8.48	8.36
21	22a	13.55	8.68	8.54	4a	9.52	8.78	8.67	17a	9.63	8.52	8.40
22	29a	13.70	8.72	8.57	13m	9.64	8.86	8.73	30a	9.81	8.56	8.43
23	2a	13.94	8.74	8.58	20a	9.76	8.94	8.79	17m	9.89	8.62	8.47
24	3a	14.63	8.76	8.60	3a	9.81	8.94	8.79	3m	10.40	8.65	8.51
25	13a	15.96	8.81	8.64	21m	9.95	9.01	8.84	15m	10.41	8.70	8.57
26	7a	17.22	8.87	8.67	25a	10.08	9.07	8.88	13a	11.02	8.75	8.60
27	14m	17.47	8.91	8.70	26a	10.25	9.11	8.91	33a	11.27	8.84	8.66
28	17m	17.64	8.98	8.75	30a	10.45	9.17	8.97	16m	11.35	8.90	8.71
29	31a	18.06	9.04	8.79	19a	10.67	9.28	9.03	28a	11.54	8.95	8.74
30	19m	17.86	9.10	8.84	14a	10.82	9.32	9.05	13m	11.76	9.01	8.77
31	30a	18.53	9.14	8.88	10m	11.00	9.40	9.10	19m	11.93	9.08	8.82
32	15m	18.43	9.19	8.94	27a	11.22	9.47	9.15	27a	12.18	9.13	8.87
33	20m	18.77	9.28	8.99	17a	11.42	9.55	9.20	20m	12.67	9.20	8.94
34	23a	19.49	9.42	9.10	11m	11.62	9.65	9.29	18a	12.96	9.30	8.99
35	18m	19.72	9.47	9.13	18a	11.87	9.72	9.32	21a	13.00	9.39	9.06
36	19a	22.70	9.66	9.21	28a	12.11	9.80	9.36	7a	13.57	9.45	9.07
37	27a	23.52	9.74	9.28	31a	12.39	9.86	9.40	23a	13.96	9.58	9.16
38	21m	24.06	9.80	9.35	5a	12.54	9.89	9.42	1a	14.34	9.60	9.19
39	3m	31.43	9.82	9.36	2a	12.64	9.93	9.45	12m	14.63	9.73	9.26
40	8m	33.78	9.96	9.47	7a	12.95	10.00	9.46	24a	15.32	9.85	9.35
41	13m	34.21	10.05	9.52	23a	13.40	10.16	9.58	6a	16.06	9.91	9.39
42	11m	35.92	10.20	9.61	22m	13.80	10.29	9.68	10m	16.44	10.02	–
43	12m	36.18	10.36	9.67	8m	14.28	10.45	9.78	21m	16.88	10.12	–
44	7m	39.68	10.67	9.82	26m	14.67	10.62	9.92	19a	18.65	10.31	–
45	22m	41.41	10.81	9.92	24m	15.18	10.79	10.06	20a	19.66	10.44	–
46	4m	54.25	10.99	10.04	34m	16.32	11.12	10.30	2m	55.84	10.46	–
47	10m	55.77	11.20	10.14	15a	17.82	11.28	10.38	11m	58.51	10.64	–
48	23m	57.68	11.39	10.29	6a	18.34	11.36	10.42	22a	58.87	10.85	–
49	33a	60.13	11.51	10.34	7m	19.34	11.70	10.61	22m	61.04	11.00	–
50	26m	62.82	11.71	10.51	24a	20.32	11.86	10.72	25m	63.65	11.15	–
51	24a	68.69	11.87	10.62	12a	21.45	12.00	10.76	23m	63.61	11.37	–
52	5m	80.39	12.21	10.85	23m	22.57	12.26	10.95	26m	67.02	11.58	–
53	27m	82.98	12.45	11.04	25m	23.55	12.46	11.10	24m	72.10	11.78	–
54	1m	51.46	12.62	11.15	9m	25.47	12.90	11.26	4m	105.28	11.94	–
55	9m	54.29	13.11	11.37	13a	28.79	13.20	11.30	27m	109.67	12.21	–
56	25m	57.19	13.38	11.55	6m	31.16	13.62	11.53	8m	114.59	12.53	–
57	24m	61.32	13.67	11.75	27m	33.16	13.90	11.74	9m	117.72	12.98	–
58	28m	65.70	14.05	12.03	28m	36.10	14.33	12.01	1m	70.04	13.21	–
59	32m	68.08	14.32	12.27	5m	39.20	14.82	12.30	28m	74.86	13.58	–
60	35a	81.09	14.98	12.41	29m	42.98	15.23	12.56	5m	85.83	14.08	–
61	29m	87.42	15.38	12.71	30m	46.69	15.66	12.84	0m	157.99	14.32	–
62	31m	96.27	15.85	12.99	32m	49.94	16.05	13.13	0a	242.08	14.97	–
63	30m	108.05	16.30	13.31	33m	54.35	16.43	13.37	30m	256.96	15.36	–
64	33m	115.27	16.73	13.54	31m	64.15	16.96	13.67	29m	274.95	15.75	–
65	6a	135.09	16.97	13.67	4m	75.76	17.61	13.91	7m	281.39	16.53	–
66	34a	142.64	17.81	13.77	0a	86.62	17.75	14.19	33m	301.40	16.89	–
67	1a	165.39	18.26	13.83	35a	116.92	19.86	14.42	31m	351.41	17.40	–
68	0a	227.91	18.91	14.07	3m	131.27	20.29	14.69	32m	377.88	17.86	–
69	34m	330.45	20.51	14.38	35m	129.74	17.18	14.65	6m	392.50	18.83	–
70	35m	275.34	17.22	14.33	1a	137.87	17.60	14.71	34m	550.89	19.51	–
71	6m	292.18	18.22	14.87	0m	180.23	18.05	15.30	35a	691.43	21.86	–
72	0m	559.87	18.42	15.57	2m	559.87	18.42	15.62	35m	559.87	18.42	–

Table 14: Sequential pruning results of the model Qwen3-4B-Base using attention and feed-forward networks Shapley values. The *Pruned* column represents the layer which was added to the pruned subset at that step (a = attention, m = feed-forward network).