Tracking Concept-Level Representation Changes Under Network Pruning

Rishika Srinivas
University of California, Santa Cruz

Arnav Kartikeya
University of California, Santa Cruz

Leilani H. Gilpin
University of California, Santa Cruz

Biagio La Rosa
University of California, Santa Cruz
bilarosa@ucsc.edu

Abstract

Pruning aims to reduce the size of deep neural networks while preserving their performance. Although recent techniques achieve strong compression rates, it remains unclear how pruning affects the internal representations of these models. In this paper, we study concept-level representation changes under representative structured and unstructured pruning approaches, with and without retraining, across BiLSTM, BERT, and LLaMA in a natural language inference setting. We use compositional explanations to characterize concepts across neuron activation ranges, extend their application to transformer-based models, and introduce metrics for measuring concept preservation, redundancy, and stability across neurons and activation ranges. Using these metrics, we identify three main patterns. First, concept associations remain stable across activation ranges even when their neuron locations change. Second, retraining-based methods preserve a larger fraction of concepts while exposing greater changes in their neuron associations, whereas the examined one-shot method retains more original neuron–concept associations but preserves fewer concepts overall. Third, in modern networks, reductions in concept redundancy under the one-shot method coincide with accuracy degradation. By exposing these representational differences, the proposed metrics can support more informed comparisons of pruning methods and sparsity levels for a given task.

1 Introduction

In the last decade, deep learning models have grown in scale [1], capturing increasingly complex patterns and improving performance in natural language understanding and reasoning [17]. However, this increase in size makes inference more expensive and limits deployment on resource-constrained devices [22]. Model compression techniques aim to address these limitations, with pruning being one of the most widely studied approaches. Pruning reduces model size by removing parameters at different granularities, such as weights, neurons, or modules, often achieving high compression rates [6, 9]. Pruning is typically evaluated using performance metrics such as accuracy or loss [10]. These metrics quantify the effect of pruning on predictive performance but cannot characterize how the model's internal representations change. Therefore, prior work has used interpretability methods to study how pruning affects important features and to identify suitable compression levels [23, 26]. However, these analyses do not track how the same semantic information is preserved, relocated, or lost through pruning. This distinction is important in settings where pruning should preserve specific semantic information in addition to overall task performance.

Tracking information changes requires comparing the semantic associations of dense and pruned models. In the case of pruning techniques, this goes beyond structural correspondence between their

Preprint.

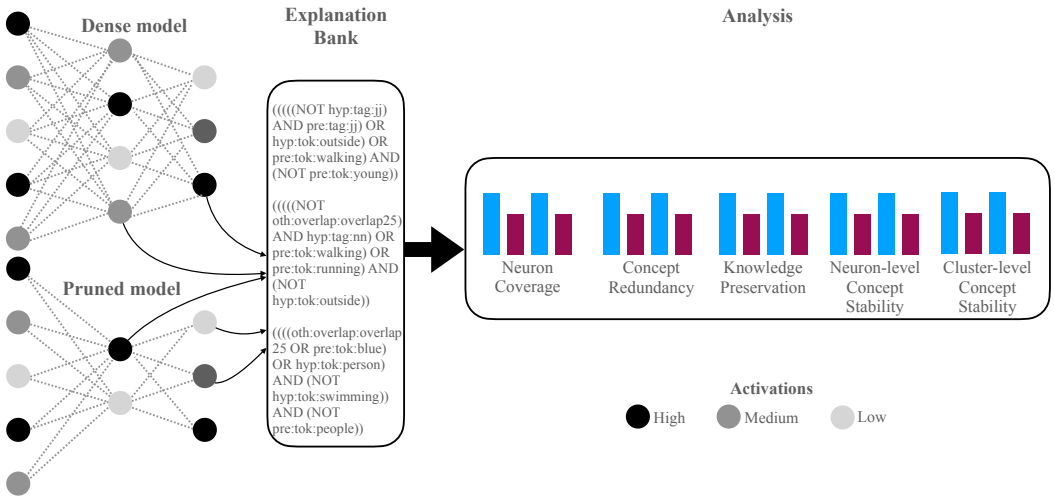

Figure 1: Our framework maps dense and pruned models to a common concept space through neuron explanations and quantifies changes in concept coverage, redundancy, preservation, and stability across neurons and activation ranges.

components. Indeed, preserving a neuron does not necessarily preserve its representation: pruning changes its incoming weights and activation patterns, potentially altering the concepts associated with it, while retraining can further reorganize these associations across neurons. To address this challenge, we use compositional explanations [3, 27, 19, 18] to characterize dense and pruned models relative to the same concept vocabulary. Compositional explanations associate combinations of human-interpretable concepts with neurons at different activation ranges, allowing us to identify concept-level representations broadly before and after pruning. We then introduce metrics that quantify concept preservation, redundancy, and stability across neurons and activation ranges (fig. 1), providing a way to measure whether concepts are preserved, reassigned, or lost as sparsity increases.

We study these changes across three pruning paradigms: structured pruning, unstructured pruning with retraining, and unstructured pruning without retraining. We evaluate these paradigms across three architectures (BiLSTM, BERT, and LLaMA) on a natural language inference task. Specifically, our contributions are as follows:

- We introduce metrics for comparing dense and pruned models at the concept level, quantifying concept preservation, redundancy, and stability across neurons and activation ranges.
- Using compositional explanations as a common concept-level representation, we apply these metrics to representative structured and unstructured pruning approaches, with and without retraining, across BiLSTM, BERT, and LLaMA in the natural language inference (NLI) setting, identifying distinct patterns of concept preservation, reassignment, and redundancy.
- We extend compositional explanations to transformer-based models by analyzing multiple activation ranges in BERT and LLaMA, enabling concept-level comparison in dense architectures where neurons are active for most inputs.

2 Related Work

We categorize pruning algorithms into structured, semi-structured, and unstructured approaches.

Structured pruning removes entire structural components of the model, such as convolutional filters [12], neurons, layers, or attention heads [35], from computation. To identify pruning candidates, methods rely on different importance criteria, including sensitivity of the loss [24], transformations [2], perplexity metrics [28], and probabilistic masks [31]. Structured pruning reduces both model size and computational cost, typically resulting in inference speedups.

Unstructured pruning removes individual weights [13], setting them to zero according to an importance criterion. Common approaches include magnitude-based pruning [9] and methods that

account for the relationship between weights and model behavior [29]. For example, SparseGPT [11] estimates pruning-induced errors over groups of weights using second-order approximations to guide iterative weight updates. Unlike structured pruning, unstructured pruning does not modify the architecture of the model: pruned weights remain in the computational graph but do not contribute to the output. As a result, while it can significantly reduce the effective parameter count, it typically does not yield inference speedups without specialized sparse computation support.

In between these two categories, semi-structured pruning represents an intermediate approach in which sparsity is imposed with regular patterns over groups of weights [34, 16]. This regularity enables partial hardware acceleration. However, similar to unstructured pruning, weights are not removed from the computational graph, and practical speedups depend on hardware support.

Across these categories, pruning methods can also be distinguished by when pruning is applied during training. **Pruning before training (PBT)** methods [20, 30] prune uninitialized models before training. **Pruning during training (PDT)** methods [21] jointly optimize weights and sparsity during training. **Pruning after training (PAT)** methods [9] prune pretrained dense models, often followed by fine-tuning to recover performance. In this work, we focus on structured and unstructured pruning methods in the PDT and PAT settings.

In the context of pruning, interpretability has been used to guide the pruning process [14], to study the re-emergence of ablated concepts after retraining [23], and to analyze how pruning affects model interpretability [26]. These works address complementary aspects of pruning but do not directly track whether the same semantic information is preserved, relocated, or lost between dense and pruned models. We address this question using compositional explanations to characterize both models relative to the same concept vocabulary and metrics that quantify concept preservation, redundancy, and stability across neurons and activation ranges.

3 Background and Metrics

3.1 Pruning

This section provides a high-level overview of the pruning techniques analyzed in this paper. For a more detailed discussion of these methods, we refer the reader to the corresponding references. In the context of pruning, we will use the term *sparsity* to denote the percentage of parameters that are removed or set to zero across all pruned layers. We consider three pruning methods spanning structured and unstructured pruning, with and without retraining.

Lottery Ticket Hypothesis (LTH) [9] is an unstructured PAT method based on iterative magnitude pruning. At each iteration, it removes a fraction of the remaining weights with the smallest magnitudes, resets the surviving weights to their original initialization, and retrains the resulting sparse network. This process is repeated until the target sparsity is reached.

Wanda [29] is an unstructured PAT method that prunes a pretrained model in a single step without retraining. It ranks weights using both their magnitude and the activation statistics of the corresponding layer and removes the lowest-scoring weights to reach the target sparsity.

CoFi [32] is a structured PDT method that learns masks over model components, including layers, attention heads, and neurons, while optimizing the model using task and distillation losses. The learned masks determine which components are retained as training progresses toward the target sparsity. In our setting, we extend CoFi by introducing an additional mask to control pruning in the final hidden layer of the MLP (see Section F for details).

3.2 Compositional Explanations

Compositional Explanations associate neuron activation ranges with logical formulas over human-interpretable concepts [27]. Given an annotated dataset, each concept is represented by a binary mask $\mathbb{M}_c$ indicating the samples in which that concept is present. Similarly, for a neuron n and an activation range $[\tau_1, \tau_2]$, a binary activation mask $\mathbb{A}$ indicates which samples produce activations within that range. Compositional Explanations search for a logical formula L over the available concepts whose mask $\mathbb{M}_L$ best overlaps with the activation mask, as measured by Intersection over

Union $IoU(\mathbb{A}, \mathbb{M}_L) = \frac{|\mathbb{A} \cap \mathbb{M}_L|}{|\mathbb{A} \cup \mathbb{M}_L|}$. The formulas combine concepts using the AND, OR and AND NOT operators. For example, a formula such as (woman AND wrestling) indicates that samples whose activations fall within the considered range tend to overlap with samples containing both concepts. Among the variants of Compositional Explanations, we use the clustered one [19], which partitions each neuron's activations into multiple ranges and computes a separate explanation for each range. We use these explanations to characterize **alignment** between concept masks and activation ranges of a given neuron. Such alignment does not imply that the model causally relies on these concepts for individual predictions. Detailed definitions and the optimization procedure are provided in Section C.1. According to established literature [27], an activation range is considered *interpretable* only if it contains at least 500 samples. For the analyses below, we consider the individual concepts occurring in each explanation formula and track their associations with neurons and activation ranges.

3.3 Metrics

To compare dense and pruned models in a common concept space, we introduce metrics that distinguish changes in concept content from changes in representational location. Specifically, the metrics quantify which concepts are identified, how redundantly they are represented, and whether preserved concepts remain associated with the same neurons and activation ranges. The first two metrics characterize a model independently, whereas the remaining three compare a dense model with its pruned counterpart.

Neuron coverage measures how much of the dataset falls within a given activation range. Let $a_n(s)$ be the activation of neuron n for sample s, and let $[l_{n,k}, u_{n,k}]$ denote the lower and upper activation bounds of cluster k for that neuron. We first define

$$f(n,k) = \frac{|\{s \in \mathfrak{D} : l_{n,k} \leq a_n(s) \leq u_{n,k}\}|}{|\mathfrak{D}|} \tag{1}$$

as the fraction of samples whose activation for neuron n falls within cluster k. Let N_k be the set of neurons for which cluster k is interpretable. Then, we define *neuron coverage*:

$$NeuronCov(k) \frac{1}{|N_k|} \sum_{n \in N_k} f(n,k) \tag{2}$$

as the average fraction of samples associated with activation cluster k across interpretable neurons. Higher values indicate activation ranges associated with a larger portion of the dataset, whereas lower values indicate more selective ranges.

Concept redundancy measures how often the same concepts recur across neuron explanations within an activation cluster. For model M and cluster k, let $F_M(k)$ denote the number of distinct concepts associated with that cluster, and let $T_M(k)$ be the total number of concept occurrences, including repetitions across neurons. Then, we define concept redundancy as:

$$RD_M(k) = \frac{T_M(k)}{F_M(k)},$$

A value of 1 indicates that every identified concept occurs only once within the cluster. Values greater than 1 indicate increasing redundancy, with larger values meaning that the same concepts are associated with multiple neurons.

Concept preservation measures how much of the concept set identified in the dense model is also identified after pruning. Let C_d and C_s denote the sets of concepts associated with the dense and sparse models, respectively. We define concept preservation as:

$$CP = \frac{|C_d \cap C_s|}{|C_d|},$$

A value of 1 indicates that every concept identified in the dense model is also identified in the sparse model, whereas a value of 0 indicates that none is preserved.

Neuron-level concept stability measures the fraction of concepts that remain associated with at least one of its original neurons. Let $C = C_d \cap C_s$ be the set of preserved concepts, and let $N_d(c)$ and $N_s(c)$ denote the sets of neurons associated with concept c in the dense and sparse models, respectively. We define neuron stability as:

$$NeuronStab = \frac{1}{|C|} \sum_{c \in C} \mathbf{1}\big(N_d(c) \cap N_s(c) \neq \emptyset\big),$$

A value of 1 indicates that every preserved concept remains associated with at least one neuron with which it was associated before pruning. A value of 0 indicates that every preserved concept is associated exclusively with different neurons after pruning. Because the metric is computed only over $C_d \cap C_s$, concept loss is captured separately by concept preservation.

Cluster-level concept stability measures the fraction of concepts that remain associated with at least one of their original activation ranges, regardless of the specific neuron carrying those associations. Let $K_d(c)$ and $K_s(c)$ denote the sets of activation clusters associated with concept c in the dense and sparse models, respectively. We define cluster-level concept stability as

$$ClusterStab = \frac{1}{|C|} \sum_{c \in C} \mathbf{1}\big(K_d(c) \cap K_s(c) \neq \emptyset\big),$$

A value of 1 indicates that every preserved concept remains associated with at least one of its original activation ranges, whereas a value of 0 indicates that all preserved concepts occur exclusively in different activation ranges after pruning.

All comparison metrics use exact concept identity within the fixed vocabulary. Consequently, semantically related but distinct concepts are treated as different representations.

4 Concept-Level Analysis

Setup In this paper, we consider the NLI task using the SNLI corpus [4]. We follow the standard configuration and hyperparameters used in prior work [27, 19]. Specifically, we define a concept set that includes lexical tokens, part-of-speech tags, and sentence-level features; we fix the maximum explanation length to 5 and the beam size to 10; and we use K-Means to identify activation ranges. We order the resulting clusters by their activation magnitude, providing corresponding low, intermediate, and high activation ranges across models. These analysis settings are held fixed across models, pruning methods, and sparsity levels so that all comparisons are performed under the same explanation configuration. We analyze the three pruning methods introduced in Section 3 as representative structured and unstructured approaches, with and without retraining. For CoFi and LTH, we report averages across three runs. Additional training and pruning details are provided in Section A. As models, we consider the BiLSTM architecture introduced by [5], as used in prior work [27], and extend the analysis to transformer-based encoder models, namely BERT [8] and Bi-Directional TinyLLaMA-1.0B [33]. To compute explanations, we focus on the later layers of each network, as they encode more complex representations.

4.1 Pre-Pruning Concept Representations

We first characterize the concept-level representations of the dense models and examine whether prior observations on compositional explanations [27, 19] extend to the considered settings.

Improving compositional explanations in NLI Compared to [27], our setup differs in two aspects: the inclusion of modern architectures and the use of clustering to identify multiple activation ranges. Mu and Andreas [27] define the activation range as all positive activations, treating the neuron behavior as a single cluster. This choice introduces two limitations. First, it conflates the concept associations occurring at low and high activation values. Second, it does not generalize to modern dense architectures such as BERT and LLaMA, where neurons are active for most inputs.

To address these limitations, we cluster neuron activations to partition them into multiple activation ranges. Table 1 highlights the benefit of this choice, showing that clustering reveals approximately $4\times$ more distinct concepts than the unclustered setting. Furthermore, only a subset of clustered concepts

Table 1: Percentage (and number) of concepts in a given cluster that also appear in the unclustered setting in BERT.

No	Cluster 1	Cluster 2	Cluster 3
1.00 (111)	0.63 (204)	0.51 (426)	0.22 (426)

Table 2: Average Neuron Coverage

	Cluster #		
Model	1	2	3
LLAMA	0.245	0.172	0.069
BiLSTM	0.003	0.002	0.000
BERT	0.238	0.177	0.077

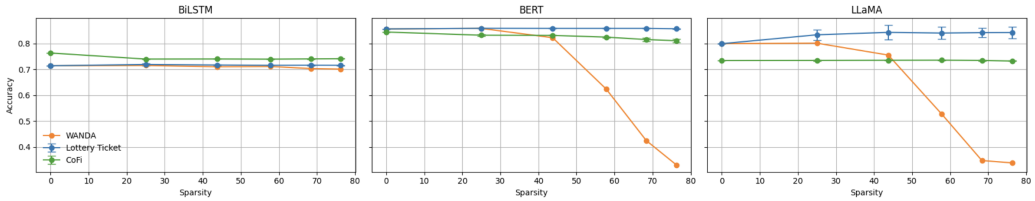

Figure 2: Accuracy reached by different models at different levels of sparsity.

overlap with those identified without clustering, indicating that a significant portion of neuron-concept alignment is missed when activations are not partitioned. We use 3 clusters throughout our analysis across architectures. As shown in Section D, finer partitions result in insufficient sample support for some high-activation ranges in the sparsest architecture, making three clusters a reasonable trade-off for shared cross-model analysis.

Initial concept representations First, we characterize the initial representations identified in the dense models. We observe a major difference in explanation coverage across architectures. BERT and LLaMA are densely explainable, with at least 70% of neurons associated with interpretable activation ranges (see Table 3 in Section B for a detailed overview), whereas compositional explanations identify such ranges for a much smaller fraction of BiLSTM neurons. The architectures also differ in the redundancy of the identified representations. In BERT and LLaMA, the same concepts are associated with multiple neurons and activation ranges, while the interpretable BiLSTM representations contain a larger fraction of distinct concepts. Specifically, BERT and LLaMA exhibit less than 20% distinct concepts across clusters, compared with at least 60% for BiLSTM.

Across all three models, approximately 25% of the available concept vocabulary is associated with the analyzed neuron representations, indicating a strong concentration of concept associations within a relatively small subset of the vocabulary (see Section B). Finally, with respect to activation clusters, Table 2 shows that neuron coverage decreases at higher activation ranges. This suggests that while higher activations correspond to more specialized behavior, the lower ones are associated with more general patterns. Specifically, we observe that, in BERT and LLaMA, more than 70% of the concepts identified in the lowest activation cluster correspond to grammatical structure, compared to less than 15% in the highest activation cluster. In the BiLSTM model, we observe a similar but less pronounced trend. Overall, these findings are aligned with the ones in the vision domain [19].

4.2 Concept Dynamics under Pruning

We now examine how the concept-level representations identified before pruning evolve as sparsity increases across the different pruning methods and architectures.

Concepts are more stable across activation ranges than across neurons. Across pruning methods and architectures, fewer than 15% of preserved concepts migrate across activation ranges (fig. 4). Thus, even when pruning changes the neuron associated with a concept, the concept tends to remain within one of its original activation ranges. Conversely, neuron-level associations are more dynamic (fig. 3), especially for methods involving training after or during pruning. This suggests a hierarchy in representational stability, where pruning can reorganize which neurons carry a concept while preserving the activation regime with which that concept is associated.

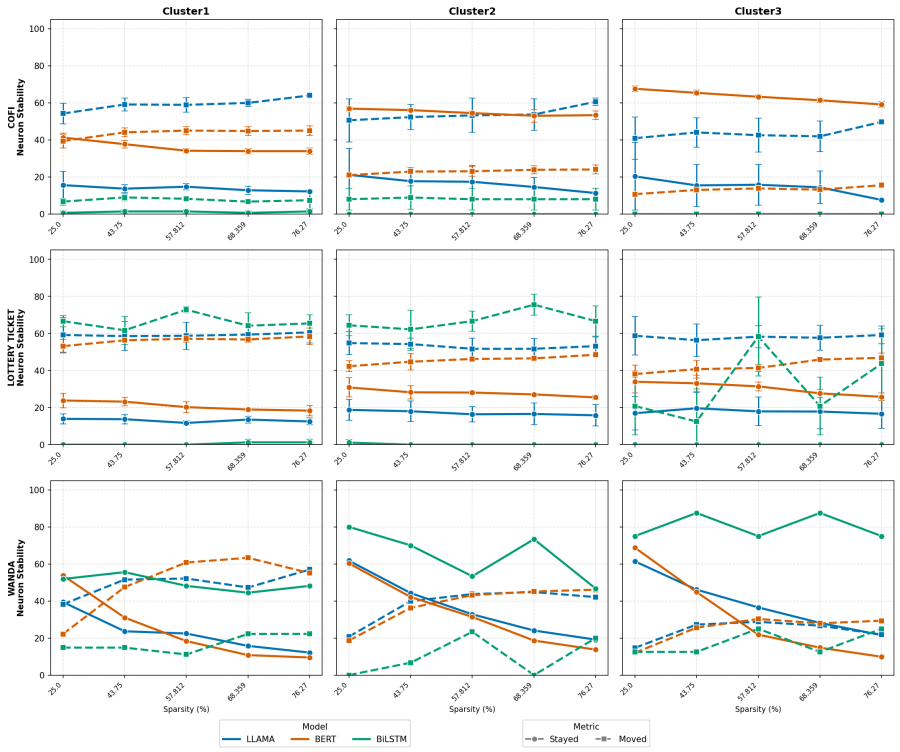

Figure 3: Evolution of Neuron Stability across sparsity levels for all the models.

The organization of these activation ranges differs across architectures. In BiLSTM, higher activation ranges exhibit greater changes under retraining, while lower ranges remain comparatively stable. In BERT and LLaMA, stability is more uniform across activation ranges. Moreover, in the transformer models, some concept associations that disappear at intermediate sparsity levels become observable again after retraining. These results suggest that activation magnitude provides a stable organization of concept associations even when their neuron-level location changes.

Pruning methods induce different forms of representational change. Wanda exhibits high neuron-level stability, with preserved concepts frequently remaining associated with the same neurons. This stability is non-trivial: although Wanda preserves neuron identities, removing weights changes their activation patterns and therefore can alter their concept associations. As sparsity increases, however, neuron- and cluster-level stability decrease together with predictive performance. In BERT and LLaMA, the largest decreases in stability occur in the same high-sparsity value in which accuracy approaches chance level (fig. 2). LTH and CoFi show a different pattern. Both involve additional training during or after pruning and exhibit greater changes in neuron-level concept associations, while preserving a larger portion of the original concept set. CoFi, in particular, achieves the highest cluster-level stability and overall concept preservation. These results suggest a trade-off between preserving concept content and preserving its neuron-level organization: some methods retain more concepts but reorganize them more extensively, whereas others preserve more of the original neuron–concept structure while losing a larger fraction of concepts.

Redundancy dynamics. Before pruning, BERT and LLaMA exhibit high concept redundancy, with some concepts associated with hundreds of neurons (up to 550 occurrences in the final layer), consistently with prior observations on transformer redundancy [7, 15]. The pruning methods affect this redundancy differently (fig. 5). LTH and CoFi preserve redundancy of the dense models across sparsity levels and activation ranges. Under Wanda, instead, redundancy progressively decreases as sparsity increases. It is worth noting that, in both BERT and LLaMA, the strongest reduction in redundancy occurs around 57% sparsity, approximately the same value at which predictive accuracy begins to degrade sharply. The recurrence of this pattern, together with the fact that retraining appears

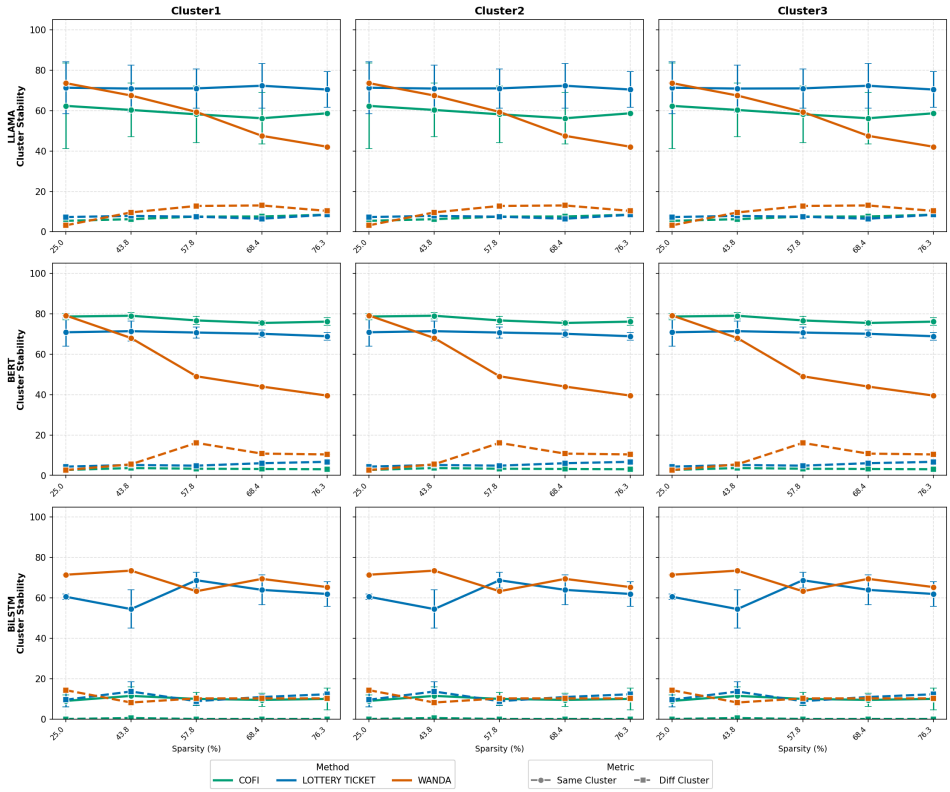

Figure 4: Evolution of Cluster Stability across sparsity levels for all the models.

to preserve redundancy instead of reducing it, identifies redundancy as a representation-level property associated with the transition from performance-preserving to performance-degrading pruning.

5 Discussion

Our results suggest that **representation preservation under pruning cannot be reduced to the preservation of individual neurons**. Across the settings considered, concept associations can remain stable at the level of activation ranges despite large changes in the neurons carrying them. This makes neuron identity a fragile reference point for comparing dense and pruned models and instead points to activation ranges as a more persistent level of organization. In this sense, the neuron-level organization of concept associations appears more plastic under pruning than their organization with respect to activation magnitude. In particular, concepts associated with higher activation ranges generally remain separated from those associated with lower ranges, even after several iterations of pruning. This observation motivates investigating whether objectives that explicitly preserve activation ranges could better preserve specific forms of concept-level representation.

This distinction is particularly informative for unstructured pruning with retraining. In LTH, unstructured pruning removes individual weights while preserving neuron identities in the architecture. Nevertheless, many preserved concepts no longer remain associated with their original neurons. Thus, the observed decrease in neuron-level stability cannot be explained simply by the deletion of the neurons that originally represented those concepts. Instead, pruning and retraining are associated with a redistribution of concept associations across neurons. At the same time, the high cluster-level stability indicates that many of these concepts remain associated with their original activation ranges. The resulting picture is therefore one in which the neuron-level organization can be subject to large changes while preserving much of the same concept content. The interpretation is slightly different for CoFi, where structured pruning directly removes neurons and other components. In that case, lower neuron-level stability partly reflects the disappearance of the original representational locations,

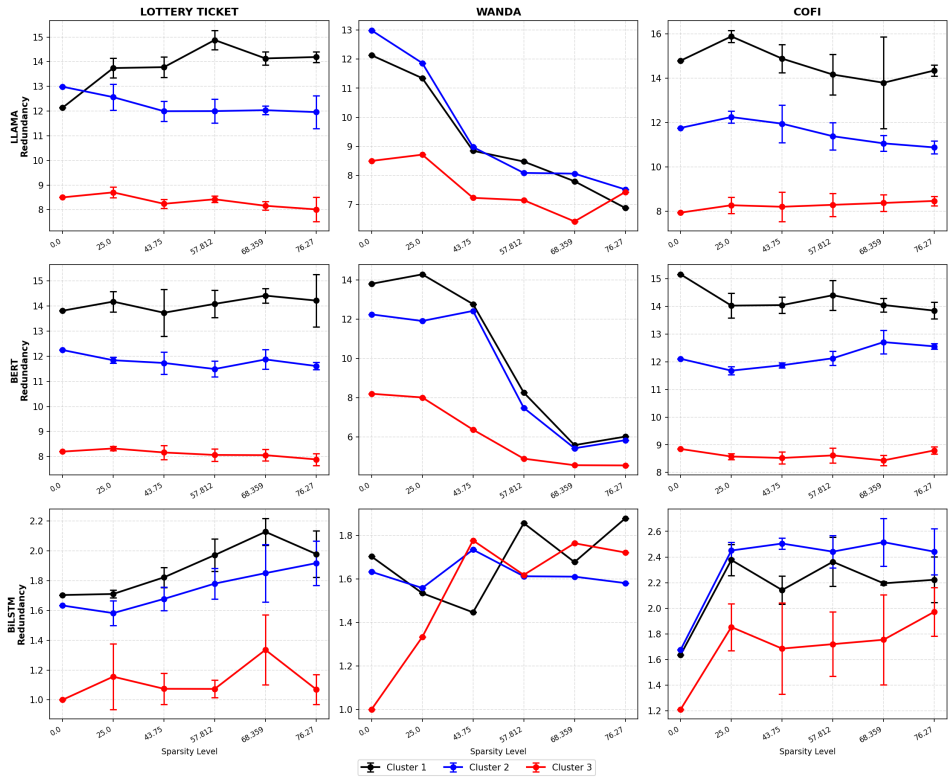

Figure 5: Redundancy evolution through pruning.

making concept preservation and cluster-level stability more informative measures of what remains after pruning.

The behavior of concept redundancy provides a **different perspective on what it means for information to be redundant in a neural representation**. In BERT and LLaMA, the same concepts are associated with many different neurons, which at first suggests that pruning could remove some of these repeated associations without affecting the model. Under Wanda, however, the substantial reduction in concept redundancy occurs at the same sparsity as the sharp degradation in predictive performance. This suggests that repeated associations with the same concept should not necessarily be interpreted as functionally interchangeable copies. Different occurrences may participate in different computations or capture the same semantic information for different subsets or contexts of the input, such that removing one association cannot simply be compensated for by another. Our analysis shows that semantic redundancy, as measured by repeated concept associations, does not imply functional substitutability. Understanding how apparently redundant concept representations play a role across the network represents a direction for future work.

More generally, these results suggest that pruning cannot be characterized by a single notion of representational preservation. Models may preserve semantic content while changing its neuron-level organization, preserve local neuron–concept associations while losing a larger fraction of the original concepts, or maintain performance while differing substantially in the redundancy of their representations. The proposed metrics make these distinctions observable and can complement task performance when comparing pruning configurations.

6 Conclusion

In this work, we introduced metrics for comparing dense and pruned models through concept-level representations identified by compositional explanations. Across three pruning approaches and three architectures in NLI, we find that concept associations remain comparatively stable across activation ranges even when their neuron-level location changes. Retraining-based methods preserve a larger

fraction of identified concepts while exhibiting greater changes in their neuron-level associations, whereas the examined one-shot method retains more original neuron–concept associations but preserves fewer concepts. In BERT and LLaMA, reductions in concept redundancy also coincide with sharp accuracy degradation. Overall, these results show that pruning can produce distinct forms of representational change that are not captured by task performance alone.

References

[1] R. Agarwal, N. Vieillard, Y. Zhou, P. Stanczyk, S. R. Garea, M. Geist, and O. Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=3zKtaqxLhW`.

[2] S. Ashkboos, M. L. Croci, M. G. d. Nascimento, T. Hoefler, and J. Hensman. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*, 2024.

[3] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

[4] S. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 632–642, 2015.

[5] S. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, 2016.

[6] H. Cheng, M. Zhang, and J. Q. 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. ISSN 1939-3539. doi: 10.1109/tpami.2024.3447085.

[7] F. Dalvi, H. Sajjad, N. Durrani, and Y. Belinkov. Analyzing redundancy in pretrained transformer models. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.398. URL `https://aclanthology.org/2020.emnlp-main.398/`.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

[9] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019. URL `https://arxiv.org/abs/1803.03635`.

[10] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin. Pruning neural networks at initialization: Why are we missing the mark?, 2021. URL `https://arxiv.org/abs/2009.08576`.

[11] E. Frantar and D. Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot, 2023. URL `https://arxiv.org/abs/2301.00774`.

[12] D. Ghimire, D. Kil, and S.-h. Kim. A study of structured pruning for hybrid neural networks. In *2024 24th International Conference on Control, Automation and Systems (ICCAS)*, pages 1110–1113, 2024. doi: 10.23919/ICCAS63016.2024.10773379.

[13] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks, 2015. URL `https://arxiv.org/abs/1506.02626`.

[14] S. M. V. Hatefi, M. Dreyer, R. Achtibat, P. Kahardipraja, T. Wiegand, W. Samek, and S. Lapuschkin. Attribution-guided pruning for compression, circuit discovery, and targeted correction in llms. *arXiv preprint arXiv:2506.13727*, 2025.

[15] S. He, G. Sun, Z. Shen, and A. Li. Uncovering the redundancy in transformers via a unified study of layer dropping. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=1I7PCbOPfe.

[16] I. Hubara, B. Chmiel, M. Island, R. Banner, S. Naor, and D. Soudry. Accelerated sparse neural training: A provable and efficient method to find n:m transposable masks, 2021. URL https://arxiv.org/abs/2102.08124.

[17] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. Jan. 2020. doi: 10.48550/ARXIV.2001.08361.

[18] B. La Rosa and L. H. Gilpin. Guaranteed optimal compositional explanations for neurons. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=MHiiwC3oFR.

[19] B. La Rosa, L. H. Gilpin, and R. Capobianco. Towards a fuller understanding of neurons with clustered compositional explanations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=51PLYhMFWz.

[20] N. Lee, T. Ajanthan, and P. H. Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

[21] S. Liu, T. Chen, X. Chen, Z. Atashgahi, L. Yin, H. Kou, L. Shen, M. Pechenizkiy, Z. Wang, and D. C. Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. *Advances in Neural Information Processing Systems*, 34:9908–9922, 2021.

[22] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJlnB3C5Ym.

[23] M. Lo, S. B. Cohen, and F. Barez. Large language models relearn removed concepts, 2024. URL https://arxiv.org/abs/2401.01814.

[24] X. Ma, G. Fang, and X. Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

[25] S. M. Makinwa, B. La Rosa, and R. Capobianco. Detection accuracy for evaluating compositional explanations of units. In *AIxIA 2021 - Advances in Artificial Intelligence*, pages 550–563. Springer International Publishing, 2022. doi: 10.1007/978-3-031-08421-8_38.

[26] F. Merkle, D. Weber, P. Schöttle, S. Schlögl, and M. Nocker. Less is more: The influence of pruning on the explainability of cnns. *IEEE Access*, 13:87909–87927, 2025. ISSN 2169-3536. doi: 10.1109/access.2025.3569575.

[27] J. Mu and J. Andreas. Compositional explanations of neurons. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[28] F. Sandri, E. Cunegatti, and G. Iacca. 2ssp: A two-stage framework for structured pruning of llms. *arXiv preprint arXiv:2501.17771*, 2025.

[29] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter. A simple and effective pruning approach for large language models, 2024. URL https://arxiv.org/abs/2306.11695.

[30] C. Wang, G. Zhang, and R. Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020.

[31] M. Xia, Z. Zhong, and D. Chen. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1528, 2022.

[32] M. Xia, Z. Zhong, and D. Chen. Structured pruning learns compact and accurate models, 2022. URL https://arxiv.org/abs/2204.00408.

[33] P. Zhang, G. Zeng, T. Wang, and W. Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

[34] A. Zhou, Y. Ma, J. Zhu, J. Liu, Z. Zhang, K. Yuan, W. Sun, and H. Li. Learning n:m fine-grained structured sparse neural networks from scratch, 2021. URL `https://arxiv.org/abs/2102.04010`.

[35] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024.

A Extended Setup

This section reports additional details about the experimental setup, complementing section 4.

The model configurations follow the structure and setup proposed by [27] to ensure consistency with prior work. Specifically, the BiLSTM architecture employs a bidirectional LSTM to encode the premise and hypothesis separately, producing two independent representations. These representations, together with their element-wise product and difference, are concatenated and passed to an MLP classifier. We adopt a similar setup for BERT and LLaMA, replacing the LSTM encoder with BERT (`bert-base-uncased`) and a bidirectional TinyLLaMA-1.0B encoder, respectively.

Following [27], all models are trained for 5 epochs over the training split of SNLI, after which no further improvements are observed. BERT and LLaMA are optimized using AdamW with learning rate 2×10^{-5} and achieve validation accuracies above 73%. The BiLSTM is trained using Adam with learning rate 10^{-3} and achieves validation accuracy above 70%, consistent with [27]. These trained models serve as the starting point for all pruning experiments. Wanda and LTH share the same starting model, whereas CoFi uses the dense model trained through the wrapper provided by the original CoFi implementation

For BERT and LLaMA, all pruning techniques are applied to all layers except the embedding layer. In the BiLSTM setting, pruning is restricted to the MLP following the LSTM encoder. To maintain consistency across pruning methods, unstructured pruning is applied only to the weights between the input layer of the MLP and the penultimate layer, while structured pruning is applied only to neurons in the penultimate layer. Additionally, following the original CoFi implementation, embedding, pooling, and normalization layers are excluded from the sparsity computation.

For Wanda, we use a fixed calibration dataset, following [29], to estimate weight importance. This choice is motivated by the robustness of Wanda with respect to the calibration data reported in the original work. The calibration set size is fixed to 100 samples.

For LTH, after each pruning iteration, the sparse model is fine-tuned until its accuracy matches or exceeds that of the dense model. For CoFi, we extend the method to support MLP layers, as described in Section F. We train the pruned model following the original optimization procedure, using a learning rate of 2×10^{-6} to optimize the BERT and LLaMA student models during knowledge distillation.

B Complementary Tables

This section includes the full results for some of the analyses reported in the main text.

Initial Concept Representations. The following tables characterize the pre-pruning representations of all the models analyzed: Table 3 shows the percentage of interpretable neurons; Table 4 shows the distribution of token and grammar concepts over the clusters; and Tables 5 to 7 shows the average and the standard deviation for the accuracies achieved by the models across sparsity levels.

Concept Dynamics under Pruning. The following figures report the evolution of concept associations during pruning across all models and pruning techniques. fig. 6 shows cluster migration and stability, while figs. 7 to 9 report neuron-level stability for Cluster 1, Cluster 2, and Cluster 3, respectively.

Table 3: Percentage of interpretable neurons in the dense models

Model	Method	Cluster	% of Intepretable Neurons
LLAMA	Lottery Ticket/WANDA	1	99.71%
		2	99.02%
		3	69.53%
	CoFi	1	100.00%
		2	100.00%
		3	98.14%
BERT	Lottery Ticket/WANDA	1	100.00%
		2	100.00%
		3	91.21%
	CoFi	1	100.00%
		2	100.00%
		3	95.70%
BiLSTM	Lottery Ticket/WANDA	1	1.66%
		2	1.17%
		3	0.20%
	CoFi	1	2.64%
		2	1.66%
		3	0.59%

Table 4: Token-based vs. grammar concepts per cluster across dense models

		Cluster 1		Cluster 2		Cluster 3	
Model	Method	Token	Grammar	Token	Grammar	Token	Grammar
LLAMA	LTH/Wanda	0.2802	0.7198	0.5786	0.4214	0.9334	0.0666
	CoFi	0.2793	0.7207	0.5538	0.4462	0.8441	0.1559
BiLSTM	LTH/Wanda	0.5652	0.4348	0.7959	0.2041	0.7500	0.2500
	CoFi	0.6034	0.3966	0.7500	0.2500	1.0000	–
BERT	LTH/Wanda	0.2445	0.7555	0.4900	0.5100	0.8518	0.1482
	CoFi	0.2093	0.7907	0.5108	0.4892	0.8783	0.1217

C Extended Preliminaries

C.1 Compositional Explanations

Let $\mathfrak{D} = \{x_1, x_2, \ldots, x_n\}$ be an annotated dataset, where each input x_i is associated with a set of annotations (possibly empty) indicating the presence of concepts within that sample. Let $\mathbb{C}$ denote the concept set. From these annotations, we construct the *concept matrix*, defined as a $N \times M$ binary matrix, where N is the number of samples and M is the total number of concepts (i.e., features). In this formulation, the concept matrix encodes only the *presence* of a concept, independently of its position within the sample and the number of times it appears. Therefore, each sample is represented by a binary vector, where 0 and 1 indicate the absence or presence of the corresponding feature, respectively.

Let $\mathfrak{L}^n$ denote the set of all logical formulas of maximum arity n constructed from concepts in $\mathbb{C}$. Let $[\tau_1, \tau_2]$ be an activation range for a given neuron k. The compositional explanation algorithm identifies the formula in $\mathfrak{L}^n$ that maximizes the following objective:

$$\underset{L \in \mathfrak{L}^n}{\arg,\max} IoU(\mathbb{A}, \mathbb{M}_L), \tag{3}$$

where $\mathbb{A}$ is defined as $\mathbb{A} = \{b_a(a_{k,j}, [\tau_i, \tau_l]), \forall j \in \mathbb{D}\}$, $b_a(a_{k,j}, [\tau_i, \tau_l])$ and $b_a(a_{k,j}, [\tau_1, \tau_2])$ is a function that maps activations within the specified range to 1 and all others to 0. The Intersection over

Table 5: Mean and standard deviation of accuracy across sparsity levels for BERT.

	BERT		
Sparsity (%)	LTH	Wanda	CoFi
0	0.856	0.856	0.844
25	0.858 ± 0.001	0.858	0.832 ± 0.004
43.75	0.858 ± 0.001	0.822	0.831 ± 0.001
57.81	0.858 ± 0.001	0.624	0.824 ± 0.002
68.36	0.858 ± 0.001	0.425	0.815 ± 0.006
76.27	0.857 ± 0.001	0.331	0.810 ± 0.007

Table 6: Mean and standard deviation of accuracy across sparsity levels for the BiLSTM model.

	BiLSTM		
Sparsity (%)	LTH	Wanda	CoFi
0	0.714	0.714	0.763
25	0.719 ± 0.004	0.715	0.740 ± 0.002
43.75	0.717 ± 0.002	0.710	0.740 ± 0.000
57.81	0.716 ± 0.001	0.711	0.739 ± 0.001
68.36	0.716 ± 0.002	0.703	0.740 ± 0.002
76.27	0.716 ± 0.001	0.701	0.741 ± 0.001

Union (IoU) measures the overlap between neuron activations and concept masks, and is defined as:

$$IoU(\mathbb{A}, \mathbb{M}_L) = \frac{|\mathbb{A} \cap \mathbb{M}_L|}{|\mathbb{A} \cup \mathbb{M}_L|}, \tag{4}$$

where $\mathbb{M}_L$ is the logical combination of the concept masks in $\mathbb{M}$ corresponding to the formula L. Following [27], we consider AND, OR, and AND NOT as logical connectives, implemented via bitwise operations over binary masks. Since $\mathfrak{L}^n$ grows combinatorially and is intractable to explore exhaustively, prior work adopts either vanilla [27, 25] or informed [19] beam search to approximate the optimal solution. In this work, we adopt the clustered variant proposed by [19], which identifies multiple activation ranges per neuron through clustering, allowing us to analyze a more complete set of concept associations for each neuron.

D Sensitivity to the Number of Activation Clusters

In this section, we examine the sensitivity of the analysis to the number of activation clusters. Our objective is not to identify an optimal number of clusters for each architecture, but to determine a shared structure among the models while providing sufficient sample support. We perform this analysis on BiLSTM because it is the architecture considered associated with the highest number of sparse neurons (i.e., neurons that activate for less than 0.005% of samples) and therefore represents the most restrictive setting for identifying interpretable activation ranges. We compare configurations with 1, 3, and 5 clusters.

Table 8 shows the number of interpretable neurons identified in each cluster under the different configurations. When using a single cluster, all neuron activations are grouped together, limiting the ability to distinguish between different activation ranges. Increasing the number of clusters improves the granularity of the analysis and reveals additional interpretable behaviors. However, with five clusters, some of the highest-activation ranges in BiLSTM do not contain the 500 samples required by the interpretability criterion (Section 4) and therefore yield no interpretable neurons. Thus, increasing the number of clusters also reduces the sample support available within each range. Based on this analysis, we use three clusters throughout the experiments. Model-specific values or alternative clustering procedures could provide different or more fine-grained characterizations. However, using the same number of clusters across architectures provides a consistent setup for cross-model comparisons, aligned with the paper's goal.

Table 7: Mean and standard deviation of accuracy across sparsity levels for LLAMA.

	LLaMA		
Sparsity (%)	LTH	Wanda	CoFi
0	0.799	0.799	0.734
25	0.833 ± 0.020	0.801	0.734 ± 0.002
43.75	0.843 ± 0.027	0.755	0.735 ± 0.001
57.81	0.840 ± 0.023	0.527	0.735 ± 0.002
68.36	0.842 ± 0.018	0.348	0.734 ± 0.002
76.27	0.842 ± 0.022	0.339	0.732 ± 0.001

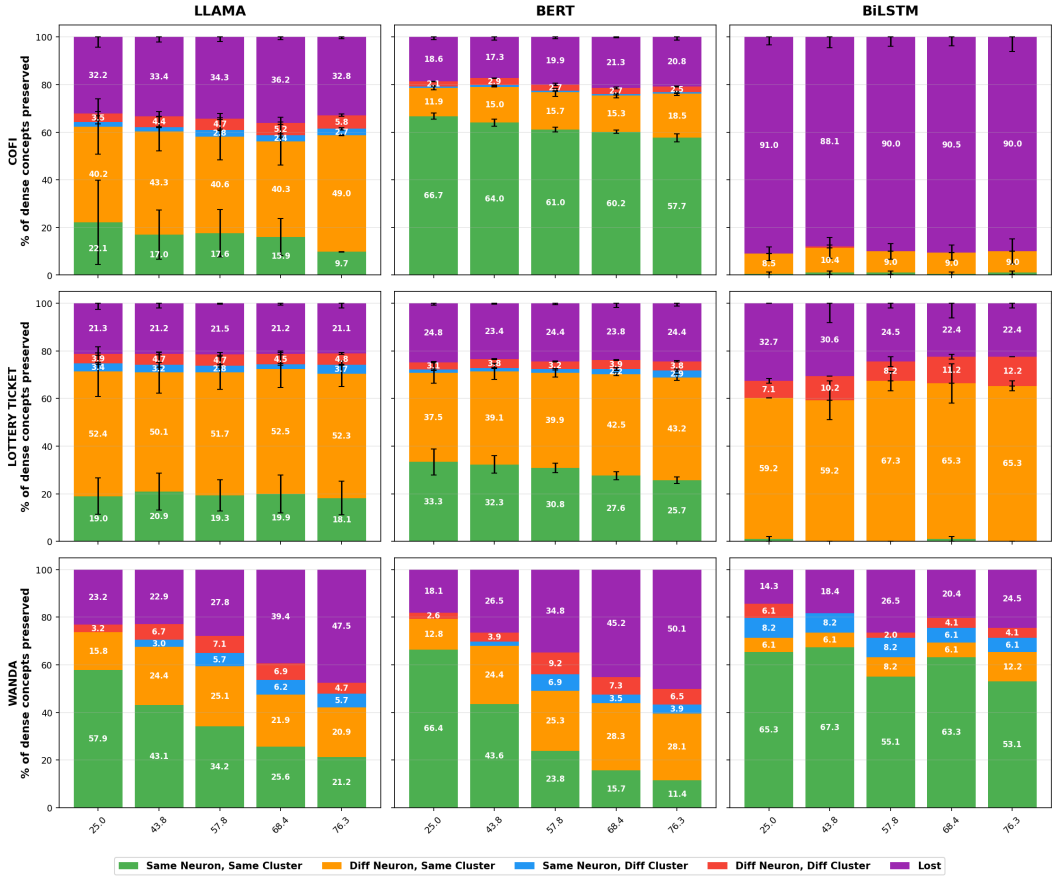

Figure 6: Cluster Stability across models and sparsity levels

E Limitations

This work provides an initial analysis of how pruning reshapes representations through neuron-level explanations. However, several limitations remain and leave open directions for future research.

Restricted Scope. Our experiments focus on the natural language inference task and on representations that can be characterized through compositional explanations and the concept space introduced by [27]. This setup enables controlled experiments and direct comparison with prior work, but only reflects a subset of the representations encoded by the models. Richer semantics, additional tasks and domains, and other modalities may exhibit different behaviors under pruning. Explanation coverage also varies across architectures. In particular, compositional explanations characterize only a small fraction of BiLSTM neurons, so results for this architecture describe this interpretable subset rather than the network as a whole.

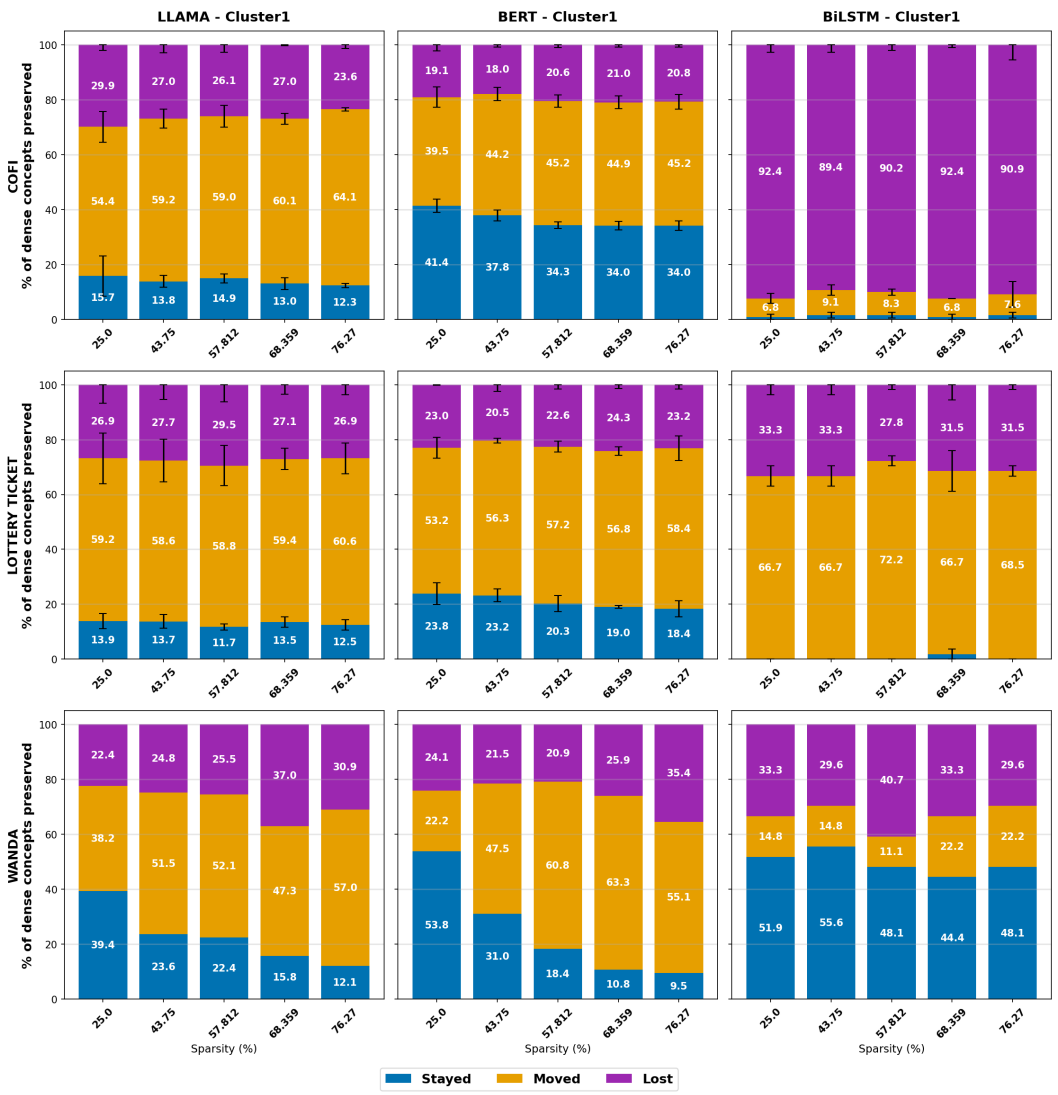

Figure 7: Neuron Stability across models and sparsity levels for Cluster 1

Coverage of Pruning Techniques. The analysis considers a limited representative subset of pruning paradigms rather than a comprehensive benchmark of pruning methods. Although the selected approaches cover structured and unstructured pruning, with and without retraining, alternative sparsity schedules, recovery procedures, or pruning criteria could lead to different concept-level dynamics.

Neuron Interactions. The proposed analysis works at the neuron and activation-range levels. This granularity allows us to study the local organization of concept associations, but it does not fully capture interactions spanning multiple neurons or layers. In transformer architectures especially, representations are highly distributed and redundant, meaning that some concept-level representations may emerge only at the level of groups of neurons rather than individual units.

Causal Analysis. The relationships identified in this work are observational. Although concept stability aligns with model performance, our experiments do not establish whether these representational changes cause downstream performance changes or instead co-occur with them as consequences of pruning. Causal interventions over neurons, concept associations, or activation regimes would be required to distinguish these possibilities.

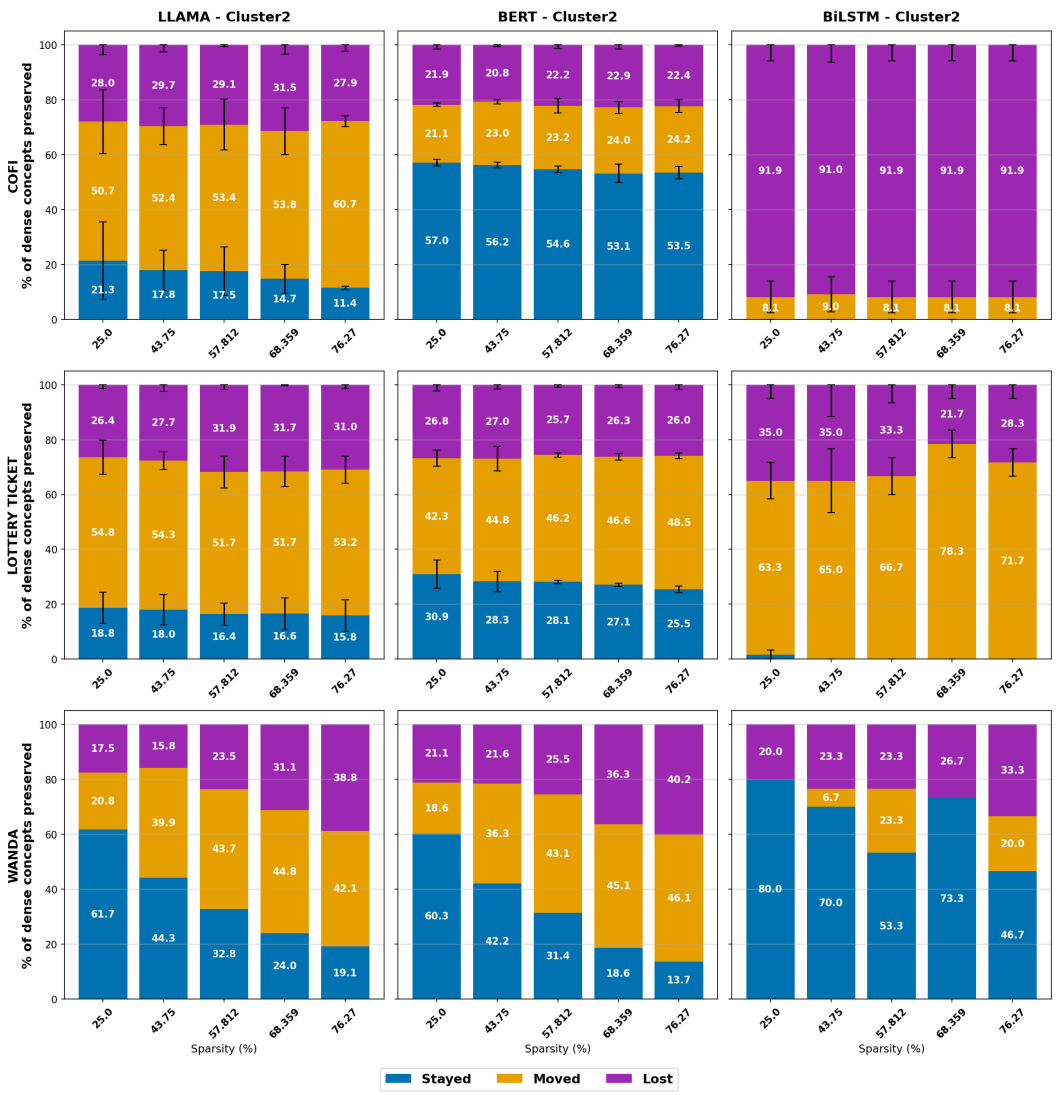

Figure 8: Neuron Stability across models and sparsity levels for Cluster 2

Analysis Configuration. Following standard practice in the area, our analysis uses a fixed concept vocabulary, minimum-support threshold, maximum formula length, beam size, and number of activation clusters. Holding these choices constant provides a common measurement configuration across models. However, alternative concept vocabularies can change which associations are observable, while the support threshold and formula-search configuration can affect which explanations are identified. The absolute values of the proposed metrics should therefore be interpreted relative to the analysis configuration used in this work.

Explanation Identifiability. Our comparisons use concepts identified by compositional explanations and depend on the mapping from activation patterns to the fixed concept vocabulary. This mapping need not be unique: semantically related concepts or alternative logical formulas may achieve similar alignment with an activation range. In addition, compositional explanations use approximate search over the formula space [27], which can select different explanations when multiple candidates obtain similar scores. As a result, some apparent concept loss or reassignment may reflect changes in the selected explanation rather than a complete disappearance of the underlying semantic pattern. Studying equivalence classes or semantic similarity between alternative explanations would provide a more representation-invariant comparison.

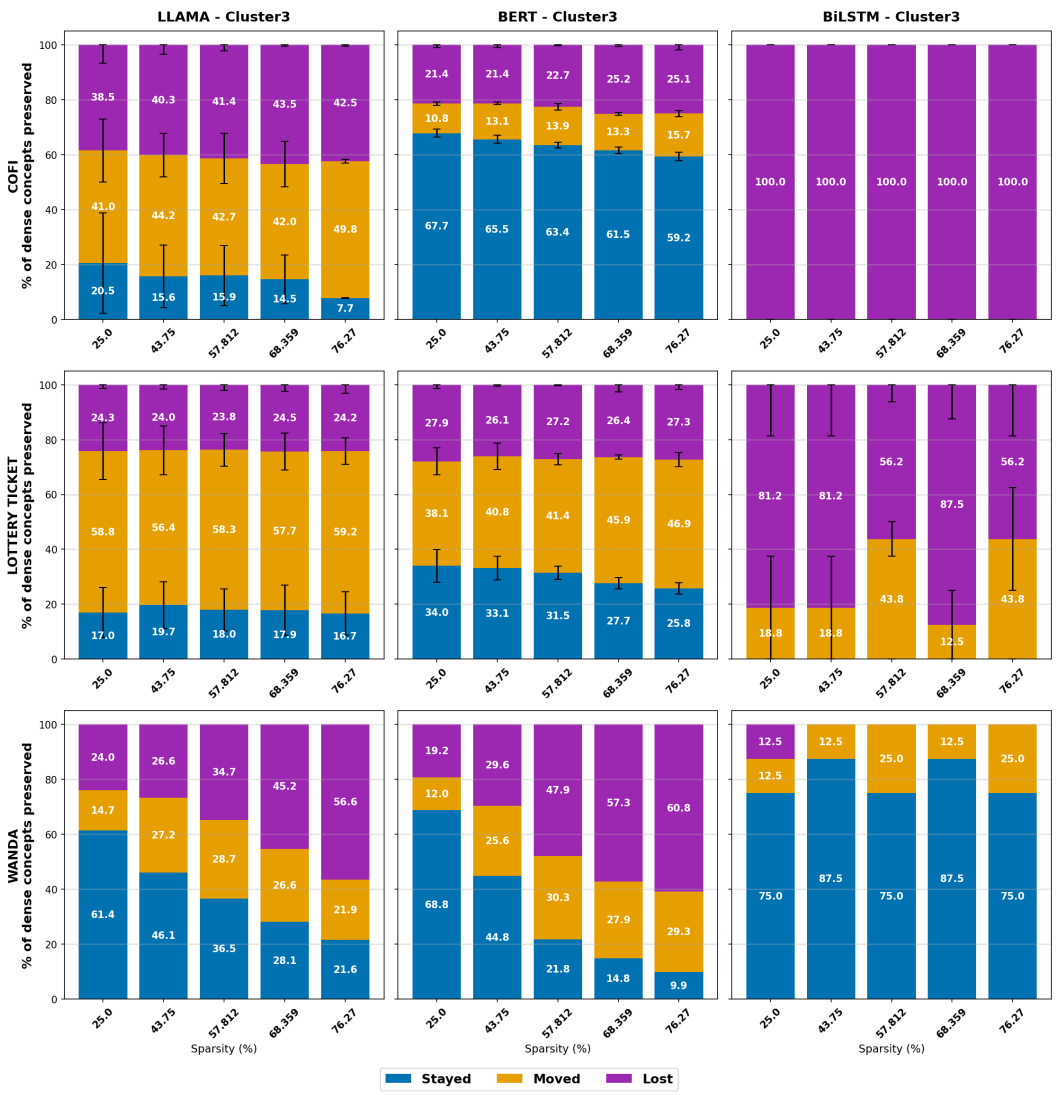

Figure 9: Neuron Stability across models and sparsity levels for Cluster 3

F Extending CoFi to MLP

CoFi is a structured pruning algorithm that follows a PDT paradigm. It prunes the underlying model at multiple structural granularities, including entire layers, attention heads, and individual neurons. CoFi associates each prunable component with a learnable soft mask. Specifically, for each structure, a mask variable z is introduced and parameterized through a sigmoid function to produce values in $[0, 1]$. These masks are applied multiplicatively during the forward pass, gating the contribution of each component. For example, given an activation h, a masked version is computed as $\tilde{h} = z \odot h$, where z controls if the corresponding unit is preserved ($z \approx 1$) or suppressed ($z \approx 0$).

The pruning process consists of two main phases. First, the model is initialized through knowledge distillation from a pretrained teacher model. Then, during pruning, the student model continues to be trained while simultaneously learning the masks. At each step, the masks are applied to the network, and both the model weights and mask parameters are updated using a combination of task loss and distillation loss. The process continues until the target sparsity is achieved, after which the model is fine-tuned to convergence. To extend CoFi to BiLSTM, BERT, and Llama on the NLI task, we introduce the following additional masks for the final MLP layer.

- $\mathbf{z}_{\text{int}}$ defines which neurons in each FFN are pruned.

Table 8: Number of interpretable neurons per cluster for 1-cluster, 3-cluster, and 5-cluster configurations.

Cluster ID	Config		
	1 Cluster	**3 Clusters**	**5 Clusters**
1	18	17	16
2	-	12	13
3	—	2	9
4	—	—	0
5	—	—	0

- $\mathbf{z_{hidden}}$ defines which hidden dimensions d_{h} are pruned.
- $\mathbf{z_{MHA}}$ defines which attention layers are pruned.
- $\mathbf{z_{head}}$ defines which heads (in BERT) or KV heads (in LLAMA) are pruned for each layer.
- $\mathbf{z_{FFN}}$ defines which feed-forward networks are pruned.

For BERT, the resulting algorithm can be expressed as follows:

$$\begin{aligned}\hat{s} = 4 \cdot d_h \sum_{i=1}^{L} \sum_{j=1}^{N_h} \sum_{k=1}^{d_{hidden}} z_{\text{MHA}}^{(i)} \cdot z_{\text{head}}^{(i,j)} \cdot z_{\text{hidden}}^{(k)} \\ + 2 \sum_{i=1}^{L} \sum_{j=1}^{d_f} \sum_{k=1}^{d_{hidden}} z_{\text{FFN}}^{(i)} \cdot z_{\text{int}}^{(i,j)} \cdot z_{\text{hidden}}^{(k)} \\ 4 \cdot \left(\sum_{j=1}^{N} z_j^{\text{hidden}} \right) \cdot \left(\sum_{i=1}^{F} z_i^{\text{pre-final}} \right) + \left(\sum_{i=1}^{F} z_i^{\text{pre-final}} \right) \cdot d_{\text{out}}\end{aligned}$$

For Llama, the algorithm can be expressed as follows:

$$\begin{aligned}\hat{s} = 2 \cdot d_{head} \cdot Q_{kv} \sum_{i=1}^{L} \sum_{j=1}^{N_{kv}} \sum_{k=1}^{d_{hidden}} z_{\text{KV}}^{(i)} \cdot z_{\text{head}}^{(i,j)} \cdot z_{\text{hidden}}^{(k)} \\ + 3 \sum_{i=1}^{L} \sum_{j=1}^{d_f} \sum_{k=1}^{d_{hidden}} z_{\text{FFN}}^{(i)} \cdot z_{\text{int}}^{(i,j)} \cdot z_{\text{hidden}}^{(k)} \\ + 4 \cdot \left(\sum_{j=1}^{N} z_j^{\text{hidden}} \right) \cdot \left(\sum_{i=1}^{F} z_i^{\text{pre-final}} \right) + \left(\sum_{i=1}^{F} z_i^{\text{pre-final}} \right) \cdot d_{\text{out}}\end{aligned}$$

Finally, for the BiLSTM:

$$\hat{s} = 4 \cdot \left(\sum_{j=1}^{N} z_j^{\text{hidden}} \right) \cdot \left(\sum_{i=1}^{F} z_i^{\text{pre-final}} \right) + \left(\sum_{i=1}^{F} z_i^{\text{pre-final}} \right) \cdot d_{\text{out}}$$

where

- L : number of attention blocks
- N_h : number of attention heads
- d_{head} : output dimension of each head
- d_{hidden} : hidden dimension
- N : number of neurons in the pre-final layer (1024)

- F : number of neurons in the final layer
- Q_{kv} : number of query parameters encompassed by one KV group (LLaMA GQA)
- N_{kv} : number of KV heads

The primary differences between LLaMA and BERT relevant to CoFi is that LLaMA employs Grouped Query Attention (GQA) rather than Multi-Head Attention (MHA). Under MHA, the $head\_zmask$ directly prunes individual attention heads with a one-to-one correspondence across Query, Key, Value, and Output projections. Under GQA, Key and Value projections operate over a smaller set of KV heads shared across multiple Query heads. We therefore redefine $head\_z$ to correspond to KV heads rather than Query heads. When emulating or applying structural pruning, we expand the KV head mask to the Query and Output projections so that the masking pattern remains consistent with the underlying grouped computation.

G Reproducibility

To ensure full reproducibility, we will release the complete codebase and all scripts required to reproduce the results presented in this paper upon acceptance. In the meantime, this section serves as a brief summary and documentation of the experimental setup used by our experiments, along with the resources required.

Dataset. In this paper we use SNLI v.1.0 accessible at: `https://nlp.stanford.edu/projects/snli/` and associated with CC-BY-SA license.

Models.

- BiLSTM
  - Dense accessible at: `http://downloads.cs.stanford.edu/nlp/data/muj/bowman_snli/6.pth`
  - License: MIT
- BERT
  - Pre-trained before finetuning accessible at: `https://huggingface.co/google-bert/bert-base-uncased`
  - License: Apache 2.0
- LLaMA
  - Pre-trained before finetuning accessible at: `https://huggingface.co/knowledgator/Llama-encoder-1.0B`
  - License: Apache 2.0

Repositories (Pruning and Compositional Explanations). Our experiments use and merge the implementations of the pruning techniques and compositional explanations code available at the following repositories:

- LTH: `https://github.com/lecode-official/pytorch-lottery-ticket-hypothesis` License: MIT
- Wanda: `https://github.com/locuslab/wanda` License: MIT
- CoFi: `https://github.com/princeton-nlp/cofipruning` License: MIT
- Compositional Explanations: `https://github.com/jayelm/compexp` License: MIT

We follow the standard settings of these repositories and use the hyperparameters mentioned in Section A to run the pruning methods.

Resources. All experiments were conducted on a workstation equipped with an NVIDIA A100 GPU, 5 CPU cores, and 125 GB of RAM. This configuration was necessary to support the most computationally demanding setting, namely LTH combined with LLaMA.

Runtime. Given the setup described above, we provide a coarse estimate of the runtime required for a single experimental run. In general, runtime depends on the specific combination of model, pruning method, sparsity level, and number of interpretable neurons. Computing compositional explanations for the final layer requires approximately 15 hours for BERT and LLaMA, and around 30 minutes for the BiLSTM model. These explanations are recomputed at each pruning iteration. Regarding pruning methods, WANDA requires approximately 2 hours per model. In contrast, LTH and CoFi are substantially more expensive and their runtime depends strongly on the architecture. On average, CoFi requires 1–2 days per model, while LTH ranges from approximately 10 hours for the BiLSTM model to up to 5 days for LLaMA.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The specific contributions are declared at the end of the introduction and are supported by experiments presented in Section 4 and 5. Specifically, contribution 1 and 2 are presented through Section 4.2 and its respective analysis on knowledge stability, and contribution 3 is supported by Section 4.2 and 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes],

   Justification: We discuss limitations in Section E. analysis.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide all the details needed to reproduce the experiments in Sections A and 3 to D.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We will release the code and models upon acceptance

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The full details about training and testing are described in Section 4 and Section A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report plots with error bars in their respective sections followed by tables recording the mean and standard deviation for all metrics analyzed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The description of resources and coarse estimate of runtime are provided in Section G.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release new data or new models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: For every model, algorithm (pruning and clustering), and dataset used, the respective authors and papers have been cited in Citations

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification:The paper does not release new assets

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.