# Revisiting Pruning vs Quantization for Small Language Models

**Zihan Zhou**
Tsinghua University
zh-zhou23@mails.tsinghua.edu.cn

**Simon Kurz**
TU Dortmund University, Lamarr Institute
simon.kurz@tu-dortmund.de

**Zhixue Zhao**
University of Sheffield
zhixue.zhao@sheffield.ac.uk

## Abstract

Deploying language models on resource-constrained devices, such as mobile phones, wearables, and on-device AI assistants, demands compact, efficient models without sacrificing performance. Compressing Small Language Models (SLMs) is particularly suited for these scenarios, yet their compression dynamics remain underexplored compared to Large Language Models (LLMs). We systematically evaluate leading post-training pruning (SparseGPT, Wanda) and quantization (GPTQ, AWQ) methods across six SLMs from 0.5 to 3.8B, seven languages, and seven downstream tasks. Our results show that quantization consistently outperforms pruning in preserving model fidelity, multilingual perplexity, and reasoning accuracy. However, quantization's advantages diminish on complex knowledge and reasoning tasks like OpenBookQA, highlighting a disconnect between compression fidelity and downstream task performance. Notably, trends observed in LLMs (e.g., Wanda's competitive performance to SparseGPT) do not generalize to SLMs. For practitioners, we recommend prioritizing quantization (particularly AWQ) for SLM compression and caution against relying on a single metric.

## 1 Introduction

The pervasive demand for intelligent systems in diverse applications, from resource-constrained devices to privacy-sensitive offline deployments, necessitates compact models and efficient inference (Al-Doghman et al., 2022; Meuser et al., 2024; Gill et al., 2025). Small Language Models (SLMs) offer distinct advantages through their efficiency, faster inference, and lower memory footprint, making them well-suited for deployment in constrained environments (Wang et al., 2025; Lu et al., 2024). Despite their smaller parameter count compared to Large Language Models (LLMs), further compression is crucial and beneficial to accommodate diverse hardware constraints and extend AI applications to edge devices (Hao et al., 2024; Pujari and Pakina, 2024; Xu et al., 2024; Wang et al., 2025).

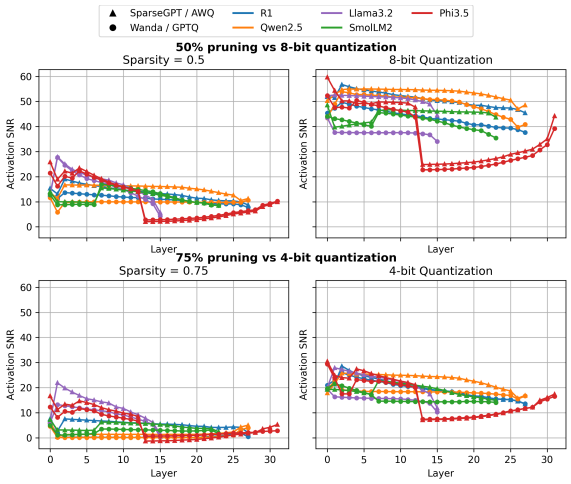


Figure 1: Layer-wise activation SNR for quantized and pruned models. Higher SNR indicates better compression fidelity. Both pruning (left column) and quantization (right column) exhibit noticeably reduced SNR at similar layers, but quantization consistently achieves higher SNR than pruning across models.

Compression techniques such as pruning and quantization have been extensively explored for LLMs; however, the effectiveness of these methods when directly applied to SLMs remains unclear. Particularly, advanced pruning methods like SparseGPT (Frantar and Alistarh, 2023a) and Wanda (Sun et al., 2024), and state-of-the-art quantization methods like GPTQ (Frantar et al., 2023) and AWQ (Lin et al., 2024), have not been thoroughly compared in the context of SLMs (Chrysostomou et al., 2024; Kurz et al., 2024; Ramesh and Zhao, 2024).[1] This gap motivates a principled study to uncover generalizable patterns, which is currently hindered by inconsistent benchmark usage and the lack of comprehensive evaluations in

[1]Detailed related work in App. A.



*Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12055–12070
November 4-9, 2025 

existing work.

We conduct extensive experiments across multiple SLMs and compression ratios, evaluating three dimensions: (1) compression fidelity to assess the activation changes after compression, Fig 1 as an example of layer-wise SNR comparison, (2) perplexity to evaluate language modeling capability for multiple languages, and (3) downstream tasks to investigate the reasoning capability. The results indicate that quantization consistently outperforms pruning across all three evaluation dimensions, with particularly strong advantages under high compression. It offers better preservation of compression fidelity and language modeling performance, while its superiority on downstream reasoning tasks is less consistent and varies across models and task types.

## 2 Methodology

We conduct a comprehensive evaluation of two state-of-the-art pruning and quantization methods each, under two commonly used compression settings (Sec. 2.1). We experiment with five popular SLMs (Sec. 2.3), spanning three-dimensional evaluations: two compression fidelity metrics, perplexity across seven languages, and accuracy on seven downstream tasks (Sec. 2.2). With the full-size baseline, this comprehensive setup results in a total of 710 evaluations across methods, models, and languages.

### 2.1 Model Compression

Given the impracticality of evaluating all compression methods, we focus on four widely adopted approaches, following the original setup unless specified otherwise.[2] We use FP16 as the standard baseline, as it is commonly used for both training and inference with negligible accuracy loss. Accordingly, we compare 50% pruning sparsity to INT8 quantization, and 75% sparsity to INT4.

**Quantization.** We consider **GPTQ** (Frantar et al., 2023) and **AWQ** (Lin et al., 2024) for 4-bit and 8-bit weight quantization. Since GPTQ was proposed before the release of Llama, we use the hyperparameters from the GPTQModel library.[3]

**Pruning.** We adopt **SparseGPT** (Frantar and Alistarh, 2023a) and **Wanda** (Sun et al., 2024), both based on unstructured pruning, which is commonly used in post-training compression studies.

### 2.2 Evaluation Metrics and Tasks

To comprehensively compare post-training pruning and quantization, we evaluate **compression fidelity** by measuring Signal-to-Noise Ratio (SNR) and compression errors of layer outputs (App. C.1). We further assess multilingual language modeling and zero-shot downstream performance. For **multilingual language modeling**, we cover seven languages of different scripts and language families: English (en), Arabic (ar), Hindi (hi), Chinese (zh), Thai (th), German (de), and Spanish (es) (256 samples with 2048 tokens each, mc4). For **zero-shot natural language understanding tasks**, we include zero-shot tasks used in the original work of GPTQ, AWQ, SparseGPT, and Wanda. These are: (1) ARC easy (ARC-e) and (2) ARC challenge (ARC-c) sets (Clark et al., 2018); (3) BoolQ (Clark et al., 2019); (4) HellaSwag (Zellers et al., 2019); (5) WinoGrande (Sakaguchi et al., 2021); (6) OpenBookQA (Banerjee et al., 2019); (7) RTE (Dagan et al., 2005). These benchmarks collectively cover multiple choice, Cloze, entailment, and Winograd-style formats, and test a range of reasoning capabilities from common sense to factual and linguistic inference. We report the evaluation set sizes in the Appendix. C.4.

### 2.3 Models

We use five popular open-source SLMs: `Llama 3.2 1B Instruct` (Llama3.2)[4], `DeepSeek R1 Distill Qwen 1.5B` (R1)[5], `Qwen2.5 1.5B Instruct` (Qwen2.5)[6], `SmolLM2 1.7B Instruct` (SmolLM2)[7], and `Phi3.5 mini instruct` (Phi3.5)[8].

### 2.4 Implementation Details

We follow Frantar et al. (2023) for hyperparameter and calibration setup[9]. To create the calibration sets, we use the publicly available version of each

[2] Complete hyperparameters are provided in App. B. We also assume no overhead on storing the sparsity mask for pruning and relegate such hardware-specific implementations to section 2.4.

[3] https://github.com/ModelCloud/GPTQModel

[4] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

[5] https://huggingface.co/deepseek-ai/R1

[6] https://huggingface.co/Qwen/Qwen2.5

[7] https://huggingface.co/HuggingFaceTB/SmolLM2

[8] https://huggingface.co/microsoft/Phi-3.5-mini-instruct

[9] For the sensitivity of the calibration dataset, we refer interested readers to Williams and Aletras (2024) for a particular investigation of the calibration dataset.

| Model | Sparsity/Bit-width | Method | SNR ↑ | Error$_{(\times 10^{-4})}$ ↓ |
|---|---|---|---|---|
| R1 | 0.50 / 8-bit | SparseGPT | 13.61 | 0.855 |
| | | Wanda | 11.11 | 1.59 |
| | | AWQ | **50.91** | **0.000** |
| | | GPTQ | 43.61 | 0.001 |
| | 0.75 / 4-bit | SparseGPT | 5.48 | 5.95 |
| | | Wanda | 0.58 | 22.39 |
| | | AWQ | **19.83** | **0.209** |
| | | GPTQ | 19.48 | 0.226 |
| Qwen2.5 | 0.50 / 8-bit | SparseGPT | 15.12 | 2.79 |
| | | Wanda | 9.81 | 11.66 |
| | | AWQ | **53.34** | **0.000** |
| | | GPTQ | 49.05 | 0.001 |
| | 0.75 / 4-bit | SparseGPT | 1.82 | 78.43 |
| | | Wanda | 0.56 | 104.12 |
| | | AWQ | **22.91** | **0.399** |
| | | GPTQ | 18.22 | 1.495 |
| Llama3.2 | 0.50 / 8-bit | SparseGPT | 17.61 | 0.915 |
| | | Wanda | 16.32 | 1.270 |
| | | AWQ | **51.09** | **0.001** |
| | | GPTQ | 37.61 | 0.015 |
| | 0.75 / 4-bit | SparseGPT | 12.59 | 2.850 |
| | | Wanda | 9.17 | 6.851 |
| | | AWQ | **22.55** | **0.303** |
| | | GPTQ | 15.39 | 2.281 |

Table 1: Layer-wise average activation SNR and compression error (Error). Full results with the other four models are in App. D. The **best** among the same compression level are in **bold** per metric.

source dataset from Hugging Face Datasets (Lhoest et al., 2021). Similarly, we use the weights and implementation of each model from Hugging Face Transformers (Wolf et al., 2020). To ensure that our model evaluations are robust and reproducible, we use the EleutherAI Language Model Evaluation Harness (Gao et al., 2024). Each model is compressed and evaluated using a single NVIDIA A100 (SXM 80GB) GPU.

## 3 Results and Analysis

### 3.1 Fidelity: SNR and Compression Error

As shown in Table 1 (see full results in App. D), **quantization consistently preserves the original signal more effectively than pruning**, as evidenced by both SNR and compression error metrics. At 50% sparsity, which corresponds to 8-bit quantization, AWQ achieves an average SNR of 50.9 on R1, and GPTQ follows closely at 43.6. In comparison, SparseGPT and Wanda drop significantly to 13.6 and 11.1 , respectively. At a higher sparsity level of 75%, pruning becomes increasingly unstable. The average SNR of Qwen2.5 at 75% sparsity falls to around 1, indicating severe degradation, while quantization continues to retain moderate fidelity in the range of 18–23. We observe the consistent pattern on the other SLMs.

Interestingly, *SparseGPT consistently outperforms Wanda* in preserving activation fidelity, which is **in contrast to the previous findings on LLMs** in Sun et al. (2024). For instance, in Qwen2.5 at 50% sparsity, SparseGPT yields 15.12 average SNR compared to 9.81 with Wanda. Between quantization methods, *AWQ outperforms GPTQ* across all models. The identical ranking on five architectures suggests that the compression method, rather than the backbone design, primarily determines fidelity. We also observe strong correlations between SNR degradation and model depth for both pruning and quantization, where SNR exhibits a roughly monotonic decline from the first layer, except for Phi3.5.

### 3.2 Language Modeling: Perplexity

As shown in Fig. 2, **quantization consistently outperforms pruning on language modeling** (full details are in App. E), particularly under high compression ratios. Quantization performance remains remarkably low and stable across different compression ratios, models and languages, while pruning sensitivity varies significantly. For instance, on the R1 model with 75% compression, AWQ yields a PPL of 32.3 for English and 185 for Arabic, while SparseGPT degrades sharply to 292.4 and 13,470, respectively. The gap is less dramatic at lower compression levels. On SLMs, SparseGPT consistently surpasses Wanda in maintaining performance across models and languages, which is also in contrast to previous findings on LLMs that Wanda outperforms SpareseGPT Sun et al. (2024). Further, similar to compression fidelity, AWQ achieves slightly better results than GPTQ in most cases.

We also find that *pruning disproportionately affects long-tail or typologically distinct-to-English languages* (e.g., Arabic, Thai, Chinese) compared

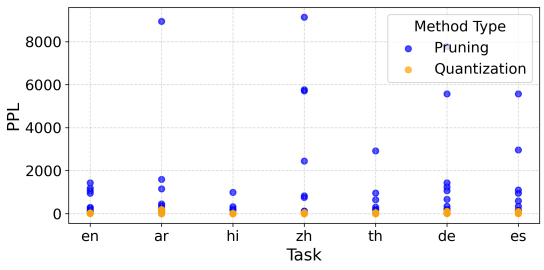


Figure 2: Comparing pruning and quantization on downstream reasoning accuracy. Each point corresponds to a specific model, method, and sparsity configuration. Quantization methods consistently yield higher retention of accuracy across tasks compared to pruning.

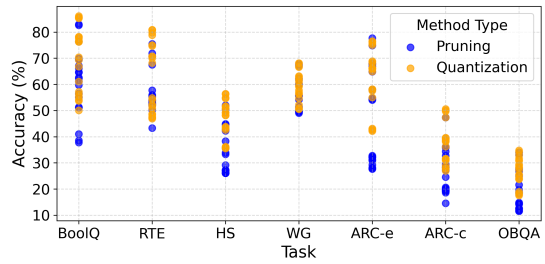


Figure 3: Comparing pruning and quantization on downstreams. Each point corresponds to a specific model, method, and sparsity configuration. Quantization methods consistently yield higher retention of accuracy across tasks compared to pruning. HS: HellaSwag, WG: WinoGrande. Details in App. G.

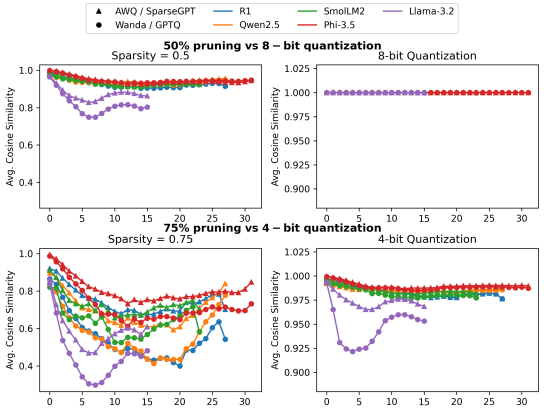


Figure 4: Layer-wise averaged cosine similarity of quantized and pruned models compared to the original model. The higher the cosine similarity, the higher the match between the compressed and the original model. Quantization with different y-axis scales is in App. H.

to resource-rich languages (Kurita et al., 2020). These languages tend to exhibit higher token-level sparsity and longer tail distributions in vocabulary usage, leading to pruning to disproportionately eliminate rare but semantically critical rows in the weight matrices (Pfeiffer et al., 2020). Quantization, by contrast, retains the full parameter structure and only introduces bounded noise, resulting in more stable performance across typologically diverse languages (Dettmers et al., 2022c). For example, under 50% sparsity in the Llama3.2 model, SparseGPT achieves a perplexity of 27.96 on English, but this rises sharply to 77.16 on Chinese and 129.15 on Arabic. In contrast, quantization methods maintain perplexities between 16 and 30 across these same languages, highlighting their robustness in multilingual settings (Lauscher et al., 2020; Dettmers et al., 2022c).

### 3.3 Downstream Tasks

As shown in Fig. 3 (full details are in App. G), **quantization consistently better preserves reasoning capabilities than pruning** across most tasks and models, especially at high compression ratio. At 50% compression, quantization retains near-original accuracy, with a maximum drop of just 1.81%, while pruning incurs significant degradation up to 16.61%. Deeper compression with 4-bit quantization only shows a maximum accuracy drop of 5.6% among all tasks, whereas 75% sparsity pruning yields double-digit losses up to 48.07%. This supports the Junk DNA hypothesis that pruning may irreversibly damage task-critical knowledge representations (Yin et al., 2024).

Another observation at 50% compression ratio is that on reading comprehension and textual entailment tasks such as BoolQ, RTE, and ARC-e, pruning methods often perform better. For example, in the R1 model, SparseGPT achieves 66.2% accuracy on BoolQ, notably outperforming AWQ at 50.1%. This may be attributed to pruning's tendency to retain high-magnitude weights, which could better preserve syntactic alignment signals. In contrast, on commonsense reasoning tasks such as HellaSwag and ARC-c, quantization outperforms pruning. On HellaSwag, for instance, AWQ yields 36.0% accuracy compared to 34.0% from SparseGPT, suggesting that the more uniform noise introduced by quantization may be more favorable for long-range reasoning.

### 3.4 Cosine Similarity

We complement the feature magnitude driven SNR and compression error analysis with an angular deviation study, which measures the average cosine similarity between token-level hidden states in the compressed and original model by layer, illustrated in Fig. 4. Across all methods, except Llama3.2, the angular similarity decreases along with layer depth increases. Notable differences persist between compression techniques: quantization more effectively preserves angular information than pruning, with AWQ performing best and Wanda worst. Deeper compression further reduces similarity, particularly for pruning-based methods.

### 3.5 Disentangling Scale vs. Compression Effects

While our work focuses on compressing SLMs under 3B parameters, it remains unclear whether the observed performance–efficiency trade-offs arise

| Model | en | ar | hi | zh | th | de | es |
|---|---|---|---|---|---|---|---|
| Uncompressed Qwen-0.5B | 15.21 | 16.29 | 6.66 | 22.62 | 6.24 | 20.89 | 18.99 |
| Llama3.2-1B (0.5 Wanda) | 35.98 | 106.94 | 23.24 | 85.30 | 26.08 | 57.45 | 51.25 |
| Llama3.2-1B (0.5 SparseGPT) | 27.96 | 129.15 | 23.20 | 77.16 | 23.13 | 45.28 | 41.38 |
| Llama3.2-1B (AWQ 8) | 16.20 | 30.31 | 9.70 | 25.23 | 11.53 | 17.13 | 18.61 |
| Llama3.2-1B (GPTQ 8) | 16.21 | 30.34 | 9.73 | 25.25 | 11.54 | 17.14 | 18.61 |
| SmolLM2-1.7B (0.75 Wanda) | 953.00 | 8939.75 | 2999.34 | 5709.18 | 652.35 | 1629.46 | 1566.07 |
| SmolLM2-1.7B (0.75 SparseGPT) | 104.97 | 1597.66 | 159.86 | 833.10 | 68.74 | 342.06 | 348.52 |
| SmolLM2-1.7B (AWQ 4) | **9.87** | **4.95** | **4.42** | **4.48** | **3.25** | **12.09** | **10.38** |
| SmolLM2-1.7B (GPTQ 4) | 10.04 | 5.23 | 4.62 | 4.59 | 3.38 | 12.54 | 10.70 |

Table 2: Multilingual perplexity of models in comparable sizes. The column-wise best is highlighted in **bold**.

primarily from the effective model size or from the compression techniques themselves. To investigate this, we compare 50% compressed Llama 3.2 1B Instruct (Llama3.2-1B) and 75% compressed SmolLM2 1.7B Instruct (SmolLM2-1.7B) with the uncompressed Qwen2.5-0.5B-Instruct (Qwen-0.5B) in Table 2 and Table 13 (the latter is provided in the Appendix). These models have comparable effective sizes. The results show that the 75% compressed SmolLM2-1.7B consistently outperforms the full-size small model Qwen-0.5B. Specifically, AWQ yields clear gains in language modeling, whereas GPTQ demonstrates greater robustness on downstream tasks. For instance, SmolLM2-1.7B quantized to 4-bit with AWQ consistently outperforms the uncompressed Qwen-0.5B across both multilingual perplexity and downstream benchmarks. We note that differences in pre-training data and methodology between Qwen-0.5B and SmolLM2-1.7B may act as potential confounders. However, fully controlling for these factors would require pre-training all models from scratch on identical corpora and with identical procedures, which is beyond the scope of this study.

Second, we compare model performance against model size in Fig. 5, where each compressed model is represented by its effective size. For example, if a 1B parameter model is pruned by 50%, it is plotted as 0.5B. The result clearly shows that compressing a larger model down to a smaller size yields better performance than using an uncompressed small model of the same size.

### 3.6 SLMs under 1B Parameters

To broaden the model size range, we include one SLM under 1B parameters, Qwen2.5-0.5B. As shown in Table 7, Table 8, and Table 9 in the Appendix, the findings for the 0.5B model align with our main conclusion: quantization, particularly AWQ, consistently outperforms pruning.

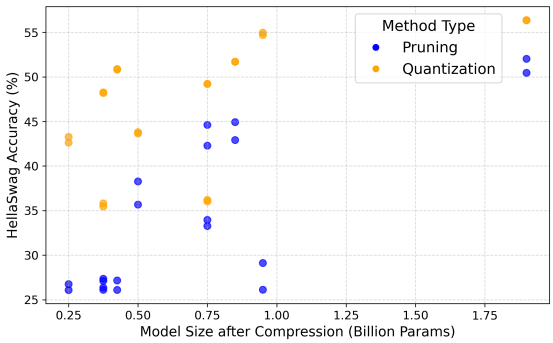


Figure 5: Comparing pruning and quantization on HellaSwag. The plot shows compressed model size (x-axis) versus performance (y-axis) across all models and compression setups.

## 4 Discussion

Our comprehensive comparison suggests the superior choice of quantization over pruning for SLMs. Although the selected pruning and quantization techniques follow a similar local activation reconstruction error minimization to retain performance (Kuzmin et al., 2023), we empirically observe that by removing entire weights, pruning impacts performance more severely than quantization. Quantization introduces small, uniform perturbations (noise) to weights, preserving overall model structure and learned representations (Jacob et al., 2018; Stock et al., 2019). Pruning, however, removes entire connections, significantly altering neuron connectivity and functionality (LeCun et al., 1989; Hassibi et al., 1993; Zhu and Gupta, 2017). While quantization reduces numerical precision, it keeps neuron connectivity intact, maintaining higher activation SNR than pruning (Kuzmin et al., 2023; Nagel et al., 2021), theoretically explaining its empirical robustness (Meuser et al., 2024).

## Limitations

While this study provides a systematic and extensive comparison of leading post-training pruning and quantization methods for Small Language Models (SLMs), it is subject to two limitations. Our empirical investigation focuses specifically on six SLMs from 0.5B to 3.8B parameters. While our findings clearly indicate the superiority of quantization over pruning for these models under aggressive compression, the direct applicability and generalization of these conclusions to other sizes of SLMs are not evaluated and remain an open question, as the dynamics of compression might differ. Additionally, we concentrate on four state-of-the-art post-training methods: SparseGPT and Wanda for pruning, and GPTQ and AWQ for quantization. This study does not cover other compression methods requiring full or partial retraining (training-aware pruning/quantization).

## References


Firas Al-Doghman, Nour Moustafa, Ibrahim Khalil, Nasrin Sohrabi, Zahir Tari, and Albert Y Zomaya. 2022. Ai-enabled secure microservices in edge computing: Opportunities and challenges. *IEEE Transactions on Services Computing*, 16(2):1485–1504.

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

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

George Chrysostomou, Zhixue Zhao, Miles Williams, and Nikolaos Aletras. 2024. Investigating hallucinations in pruned large language models for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 12:1163–1181.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022a. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022b. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022c. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

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

Elias Frantar and Dan Alistarh. 2022. Optimal brain compression: A framework for accurate post-training quantization and pruning. *arXiv preprint arXiv:2208.11580*.

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

Elias Frantar and Dan Alistarh. 2023b. Sparsegpt: Massive language models can be accurately pruned in one-shot. *arXiv preprint arXiv:2301.00774*.

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

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

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

Sukhpal Singh Gill, Muhammed Golec, Jianmin Hu, Minxian Xu, Junhui Du, Huaming Wu, Guneet Kaur Walia, Subramaniam Subramanian Murugesan, Babar Ali, Mohit Kumar, and 1 others. 2025. Edge ai: A taxonomy, systematic review and future directions. *Cluster Computing*, 28(1):1–53.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Preprint*, arXiv:1510.00149.

Zixu Hao, Huiqiang Jiang, Shiqi Jiang, Ju Ren, and Ting Cao. 2024. Hybrid slm and llm for edge-cloud collaborative inference. In *Proceedings of the Workshop on Edge and Mobile Foundation Models*, pages 36–41.

Babak Hassibi, David Stork, and Gregory Wolff. 1993. Optimal brain surgeon: Extensions and performance comparisons. In *Advances in Neural Information Processing Systems*.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. 2023. Compressing llms: The truth is rarely pure and never simple. *arXiv preprint arXiv:2310.01382*.

Sho Kurita, Jonas Pfeiffer, Ivan Vulić, Edoardo Ponti, and Anna Korhonen. 2020. A weighted approach to unsupervised multilingual transformer fine-tuning. In *Proc. of AACL-IJCNLP*.

Simon Kurz, Jian-Jia Chen, Lucie Flek, and Zhixue Zhao. 2024. Investigating language-specific calibration for pruning multilingual large language models. *arXiv preprint arXiv:2408.14398*.

Andrey Kuzmin, Markus Nagel, Mart van Baalen, Arash Behboodi, and Tijmen Blankevoort. 2023. Pruning vs quantization: Which is better? In *Advances in Neural Information Processing Systems*.

Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. 2020. Language diversity in multilingual NLP models: Current trends and limitations. In *Proc. of COLING*.

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

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

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. 2021. {BRECQ}: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*.

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

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2024. Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790*.

Tobias Meuser, Lauri Lovén, Monowar Bhuyan, Shishir G Patil, Schahram Dustdar, Atakan Aral, Suzan Bayhan, Christian Becker, Eyal De Lara, Aaron Yi Ding, and 1 others. 2024. Revisiting edge ai: Opportunities and challenges. *IEEE Internet Computing*, 28(4):49–59.

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

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. 2021. A white paper on neural network quantization. *ArXiv*, abs/2106.08295.

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

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Mangesh Pujari and Anil Kumar Pakina. 2024. Edgeai for privacy-preserving ai: The role of small llms in federated learning environments. *International Journal of Engineering and Computer Science*, 13(10):26589–26601.

Samarth N Ramesh and Zhixue Zhao. 2024. Efficient pruning of text-to-image models: Insights from pruning stable diffusion. *arXiv preprint arXiv:2411.15113*.

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

Pierre Stock, Armand Joulin, Rémi Gribonval, Benjamin Graham, and Hervé Jégou. 2019. And the bit goes down: Revisiting the quantization of neural networks. *arXiv preprint arXiv:1907.05686*.

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

Rui Wang, Zhiyong Gao, Liuyang Zhang, Shuaibing Yue, and Ziyi Gao. 2025. Empowering large language models to edge intelligence: A survey of edge efficient llms and techniques. *Computer Science Review*, 57:100755.

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

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

Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. 2024. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*.

Lu Yin, Shiwei Liu, AJAY KUMAR JAISWAL, Souvik Kundu, and Zhangyang Wang. 2024. Junk DNA hypothesis: A task-centric angle of LLM pre-trained weights through sparsity.

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

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

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

## A Related Work

Pruning and quantization are two widely adopted approaches for model compression (Gholami et al., 2021; Hoefler et al., 2021; Zhu et al., 2024). With origins in seminal work from Hassibi et al. (1993); LeCun et al. (1989), quantization achieves compression by reducing the numerical precision of model parameters, while pruning determines redundant weights for removal (Han et al., 2016). Their application to LLMs poses significant challenges, such as the importance of retaining large-magnitude outlier features, interplay between weights and inputs, and high computational requirements (Dettmers et al., 2022a). Recently, advanced pruning methods like SparseGPT (Frantar and Alistarh, 2023a) and Wanda (Sun et al., 2024), and sophisticated quantization techniques such as GPTQ (Frantar et al., 2023) and AWQ (Lin et al., 2024), have significantly advanced LLM compression in the post-training and retraining-free setting.

While quantization and pruning achieve compression through different means, they share the common optimization goal of *block-wise reconstruction error minimization* (Frantar and Alistarh, 2022; Li et al., 2021). That is, model weights get compressed in a way that best preserves the original outputs of a model block. More formally, given layer $\ell$ with weights $\mathbf{W}_\ell$ and inputs $\mathbf{X}_\ell$, the goal is to minimize $||\mathbf{W}_\ell\mathbf{X}_\ell - \widehat{\mathbf{W}}_\ell\mathbf{X}_\ell||_2^2$ with respect to the compressed weights $\widehat{\mathbf{W}}$.

While pruning and quantization are now widely used for compressing LLMs, early comparative studies (Kuzmin et al., 2023) rely on outdated techniques such as magnitude-based pruning and plain

symmetric quantization. These approaches typically minimize weight reconstruction error without accounting for activation dynamics, limiting their relevance to current methods. Foundational theoretical publications (Frantar and Alistarh, 2022; Dettmers et al., 2022b; Dettmers and Zettlemoyer, 2023) fostering later compression techniques emphasize the importance of preserving activations, reflecting a broader shift in compression objectives. This shift has yielded more effective methods. Previous work has compared pruning and quantization for LLMs, but without focusing on compression errors (Jaiswal et al., 2023). Moreover, SLMs remain underexplored in this setting. As SLMs grow in importance for efficient deployment, it is unclear whether findings from LLMs carry over.

## B Implementation Details

We follow the exact hyperparameter settings from prior work for pruning and quantization. Specifically, for Wanda, SparseGPT, and GPTQ, we sample the calibration set from the initial shard of the C4 dataset, following the methodology of Frantar et al. (2023).

As previous studies have shown that the benefit of increasing calibration samples saturates logarithmically (Frantar and Alistarh, 2023b; Sun et al., 2024), we adopt the standard protocol of randomly sampling 128 calibration examples, each containing 2,048 tokens, totaling 262,144 tokens. For further discussion on calibration set sensitivity, we refer to the analysis by Williams and Aletras (2024).

AWQ uses a smaller calibration set: 128 samples of 512 tokens each, drawn from The Pile, totaling 65,536 tokens. For AWQ, SparseGPT, and GPTQ, we set the group size to 128 and prune for unstructured sparsity.

To evaluate pruning error, SNR, and cosine similarity, we sample 128 sequences of 2,048 tokens from the C4 validation set. Metrics are computed token-wise at each layer output. For multilingual perplexity evaluations, we use the corresponding mC4 validation set for each language, maintaining the same number of samples and tokens as in the internal feature analysis.

All calibration datasets are sourced via public versions on Hugging Face Datasets (Lhoest et al., 2021), and all model architectures and configurations are loaded from Hugging Face Transformers (Wolf et al., 2020). For quantization, we use the Optimum library for GPTQ and AutoAWQ for AWQ. SparseGPT and Wanda are implemented using the official GitHub repositories provided by the respective paper authors.

For the construction of calibration corpora, we employ the publicly accessible iterations of each source dataset available through Hugging Face Datasets (Lhoest et al., 2021). Similarity, we utilize the parametric configurations and implementations of each architectural model from Hugging Face Transformers (Wolf et al., 2020).

## C Evaluation

### C.1 Quantization and Pruning Error

We refer to the definition of Kuzmin et al. (2023) for computing the Pruning Error and SNR. However, both of them relate to model weights, which does not reflect the current compression paradigm of emphasizing the importance of features. Therefore, we compute Pruning Error and SNR with respect to the outputs of each layer, i.e. the hidden states. To cope with different hidden state magnitudes per layer, all hidden states are normalized layer-wise by their average vector norm, yielding the following pruning error:

$$E[(H^{(l)} - \widetilde{H}^{(l)})^2] = \frac{1}{Nd} \sum_{i=1}^{N} \sum_{j=1}^{d} \left( \frac{h_{i,j}^{(l)} - \widetilde{h}_{i,j}^{(l)}}{\mu^{(l)}} \right)^2 \tag{1}$$

$$\mu^{(l)} = \frac{1}{N} \sum_{i=1}^{N} ||h_i^{(l)}||_2 \tag{2}$$

with $H^{(l)} \in^{N \times d}$ as the hidden states of all $N$ tokens in layer $k$ before compression and $\widetilde{H}^{(l)} \in^{N \times d}$ after compression. Individual elements of the hidden states are denoted as $h_{i,j}^{(l)}$, referring to the hidden state of the $j$-th feature element of the $i$-th token in $H^{(l)}$.

### C.2 Signal-to-noise ratio (SNR)

Following the same notation as for computing the compression error, the SNR of a single layer $l$ is computed as

$$SNR_{dB}^{(l)} = 10 log_{10} \left( \frac{E\left[ H^{(l)} \right)^2 \right]}{E\left[ \left( H^{(l)} - \widetilde{H}^{(l)} \right)^2 \right]} \right) \tag{3}$$

The final model SNR and compression error is the average over all layer-wise $SNR_{dB}^{(l)}$ or compression errors respectively.

### C.3 Practical Deployment Comparision

Given the growing need to deploy language models on resource-constrained edge devices, practical deployment-relevant statistics such as memory footprint, latency and inference speed are crucial and highly informative for practitioners. In response, We have included theoretical inference speedup from Wanda, SparseGPT, AWQ, and GPTQ papers, and also calculated FLOPs for comparison in Table 3.

### C.4 Datasets

Table 4 presents the number of examples across the evaluation tasks, extracted from their respective test or validation partitions.

## D Fidelity: SNR and Compression Error

We report average activation-level SNR and compression error across all models and configurations as shown in Table 5. Quantization methods consistently yield higher SNR and lower error than pruning methods, particularly under higher compression. AWQ achieves the strongest overall fidelity, followed by GPTQ. Among pruning methods, SparseGPT consistently outperforms Wanda. Additionally, both SNR and error exhibit a depth-wise trend, with deeper layers generally showing greater degradation. Results on English are visualized in Fig.6,Fig.7, Fig.8, Fig.9, and Fig.10.

| Method | Models & Size | Sparsity | Device | Memory Footprint | FLOPs / Ops | Performance (speed/latency) |
|---|---|---|---|---|---|---|
| Wanda | LLaMA (7B, 13B, 30B, 65B) | 50% unstructured; 2:4, 4:8 | GPU/CPU (A6000) | $\approx 0.5\times$ (weights) | $\approx 0.5\times$ (ops) | 0.54 s vs 203 s prune (7B, A6000); $\sim$1.5–1.8$\times$ for 2:4 |
| SparseGPT | OPT/BLOOM (up to 175B; LLaMA 13B) | 50-60% unstructured; 2:4, 4:8 | GPU (A100) | $\approx 0.5\times$ (weights) | $\approx 0.5\times$ (ops) | Prunes 175B in $\approx$ 4.5 h; negligible accuracy loss up to 50-60% sparsity; $\sim$1.5-1.8$\times$ speedup for 2:4 sparsity |
| AWQ | LLaMA-2/Vicuna/ MFM (up to 70B) | INT4 weights (keep$\sim$1% FP16) | GPU (RTX4090) | $\approx 0.25\times$ (weights vs FP16) | $\approx 1\times$ (ops) | 3.2-3.9$\times$ speedup vs FP16; 30 tok/s (13B on RTX4070-8GB) |
| GPTQ | GPT/OPT (up to 175B; LLaMA 7B/ 13B/30B/65B) | INT4/3 (layerwise; some INT2) | GPU (A100/ A6000) | $\approx 0.25\times$ (weights vs FP16) | $\approx 1\times$ (ops) | $\sim$3.25$\times$ (A100)–4.5$\times$ (A6000) speedup vs FP16; single-GPU 175B |

Table 3: Practical Deployment Metrics comparison summary

| Dataset | # Examples |
|---|---|
| ARC-Easy (Clark et al., 2018) | 2,376 |
| ARC-Challenge (Clark et al., 2018) | 1,172 |
| BoolQ (Clark et al., 2019) | 3,270 |
| HellaSwag (Zellers et al., 2019) | 10,042 |
| LAMBADA (Paperno et al., 2016) | 5,153 |
| OpenBookQA (Banerjee et al., 2019) | 500 |
| PIQA (Bisk et al., 2020) | 1,838 |
| RTE (Dagan et al., 2005) | 277 |
| StoryCloze (Mostafazadeh et al., 2016) | 1,511 |
| WinoGrande (Sakaguchi et al., 2021) | 1,267 |

Table 4: Number of examples for each evaluation task.

| Model | Sparsity/Bit-width | Method | SNR | Error($\times 10^{-4}$) |
|---|---|---|---|---|
| SmolLM2 | 0.50 / 8-bit | SparseGPT | 11.75 | 1.31 |
| | | Wanda | 11.87 | 1.32 |
| | | AWQ | 44.78 | 0.00 |
| | | GPTQ | 41.71 | 0.00 |
| | 0.75 / 4-bit | SparseGPT | 4.57 | 7.46 |
| | | Wanda | 2.65 | 12.51 |
| | | AWQ | 18.97 | 0.23 |
| | | GPTQ | 15.86 | 0.71 |
| Phi3.5 | 0.50 / 8-bit | SparseGPT | 10.53 | 4.82 |
| | | Wanda | 10.42 | 4.17 |
| | | AWQ | 37.47 | 0.02 |
| | | GPTQ | 34.95 | 0.04 |
| | 0.75 / 4-bit | SparseGPT | 5.28 | 11.17 |
| | | Wanda | 4.70 | 8.49 |
| | | AWQ | 15.97 | 1.47 |
| | | GPTQ | 14.86 | 1.50 |

Table 5: Full results of layer-wise average activation SNR and compression error (Error).

## E Multilingual Language Modeling

Table 6 presents token-level perplexity across seven languages for all models and compression settings. Quantization maintains stable performance across languages and compression levels, while pruning leads to greater degradation, especially at higher sparsity. The performance drop under pruning varies across languages and models, reflecting the interaction between compression sensitivity, tokenization, and pretraining distribution.

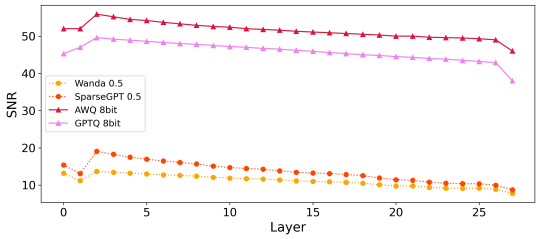


Figure 6: Layer-wise activation SNR on English tokens for R1.

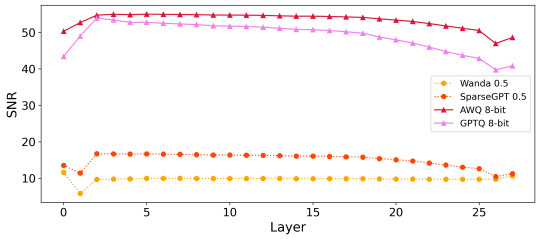


Figure 7: Layer-wise activation SNR on English tokens for Qwen2.5.

## F Small models smaller than 1B

## G Reasoning Tasks

To compare the downstream task performance, we include a wide range of reasoning tasks for zero-shot evaluation shown in Table 10. These tasks primarily assess commonsense reasoning abilities, using binary, multiple choice, Cloze, and Winograd style questions, covering different difficulty levels: OBQA (OpenBookQA) is the multi-hop reasoning task, requiring combining multiple facts from the "open book" knowledge base to answer questions.

| Model | Sparsity | Method | en | ar | hi | zh | th | de | es |
|---|---|---|---|---|---|---|---|---|---|
| R1 | 0.00 | Full Precision | 29.79 | 159.39 | 11.02 | 41.37 | 40.67 | 98.58 | 85.36 |
| | 0.50 | SparseGPT | 44.41 | 379.12 | 16.26 | 117.87 | 306.32 | 193.78 | 162.97 |
| | | Wanda | 47.75 | 451.78 | 16.85 | 114.34 | 170.08 | 206.75 | 164.00 |
| | | AWQ | 29.80 | 159.51 | 11.03 | 41.38 | 40.46 | 98.73 | 85.48 |
| | | GPTQ | 29.83 | 159.32 | 11.03 | 41.38 | 40.87 | 98.66 | 85.59 |
| | 0.75 | SparseGPT | 292.42 | 13469.27 | 996.42 | 5767.65 | 2922.67 | 5566.94 | 2966.31 |
| | | Wanda | 1180.82 | 201520.22 | 42817.80 | 9133.28 | 60985.34 | 26397.47 | 24978.16 |
| | | AWQ | 33.22 | 190.01 | 12.30 | 49.83 | 48.97 | 112.98 | 96.45 |
| | | GPTQ | 32.28 | 185.16 | 12.29 | 49.08 | 46.54 | 112.37 | 99.91 |
| Qwen2.5 | 0.00 | Full Precision | 10.98 | 11.00 | 5.42 | 12.07 | 5.35 | 13.31 | 11.45 |
| | 0.50 | SparseGPT | 15.54 | 41.40 | 11.02 | 33.20 | 15.11 | 27.49 | 22.64 |
| | | Wanda | 16.38 | 39.73 | 10.65 | 30.49 | 16.91 | 31.39 | 26.26 |
| | | AWQ | 10.98 | 11.00 | 5.42 | 12.07 | 5.35 | 13.31 | 11.46 |
| | | GPTQ | 10.98 | 11.00 | 5.42 | 12.07 | 5.35 | 13.32 | 11.46 |
| | 0.75 | SparseGPT | 155.99 | 1158.69 | 324.37 | 2424.02 | 961.63 | 1238.69 | 1102.33 |
| | | Wanda | 1436.60 | 26945.81 | 12328.96 | 69817.35 | 16434.86 | 7782.48 | 5573.80 |
| | | AWQ | 11.79 | 12.68 | 6.14 | 13.19 | 6.08 | 14.87 | 12.48 |
| | | GPTQ | 12.02 | 13.44 | 6.41 | 14.25 | 6.33 | 15.32 | 12.81 |
| Llama3.2 | 0.00 | Full Precision | 16.20 | 30.31 | 9.72 | 25.23 | 11.53 | 17.13 | 18.61 |
| | 0.50 | SparseGPT | 27.96 | 129.15 | 23.2 | 77.16 | 23.13 | 45.28 | 41.38 |
| | | Wanda | 35.98 | 106.94 | 23.24 | 85.3 | 26.08 | 57.45 | 51.25 |
| | | AWQ | 16.2 | 30.31 | 9.7 | 25.23 | 11.53 | 17.13 | 18.61 |
| | | GPTQ | 16.21 | 30.34 | 9.73 | 25.25 | 11.54 | 17.14 | 18.61 |
| | 0.75 | SparseGPT | 273.95 | 144742.47 | 53063.41 | 41727.27 | 40215.73 | 1429.28 | 1111.82 |
| | | Wanda | 10130.75 | 302363.66 | 116279.45 | 180397.61 | 189368.5 | 45676.04 | 22685.21 |
| | | AWQ | 17.53 | 36.64 | 10.95 | 29.96 | 12.94 | 19.46 | 20.72 |
| | | GPTQ | 18.9 | 46.59 | 13.22 | 37.00 | 15.44 | 22.84 | 23.29 |
| SmolLM2 | 0.00 | Full Precision | 9.27 | 4.50 | 4.00 | 4.11 | 2.94 | 10.86 | 9.57 |
| | 0.50 | SparseGPT | 13.85 | 8.98 | 6.93 | 10.70 | 4.23 | 20.07 | 18.75 |
| | | Wanda | 13.63 | 7.97 | 6.48 | 10.37 | 4.22 | 20.80 | 18.78 |
| | | AWQ | 9.27 | 4.51 | 4.01 | 4.11 | 2.94 | 10.86 | 9.57 |
| | | GPTQ | 9.27 | 4.51 | 4.01 | 4.11 | 2.94 | 10.87 | 9.57 |
| | 0.75 | SparseGPT | 104.97 | 1597.66 | 159.86 | 833.10 | 68.74 | 342.06 | 348.52 |
| | | Wanda | 953.00 | 8939.75 | 2999.34 | 5709.18 | 652.35 | 1629.46 | 1566.07 |
| | | AWQ | 9.87 | 4.95 | 4.42 | 4.48 | 3.25 | 12.09 | 10.38 |
| | | GPTQ | 10.04 | 5.23 | 4.62 | 4.59 | 3.38 | 12.54 | 10.70 |
| Phi3.5 | 0.00 | Full Precision | 7.75 | 3.20 | 3.71 | 5.04 | 3.85 | 8.61 | 8.64 |
| | 0.50 | SparseGPT | 11.34 | 9.11 | 9.90 | 15.26 | 8.69 | 15.78 | 16.49 |
| | | Wanda | 11.08 | 7.71 | 8.32 | 13.05 | 7.69 | 14.60 | 15.43 |
| | | AWQ | 7.75 | 3.20 | 3.71 | 5.04 | 3.85 | 8.61 | 8.64 |
| | | GPTQ | 7.75 | 3.20 | 3.71 | 5.04 | 3.85 | 8.61 | 8.65 |
| | 0.75 | SparseGPT | 76.53 | 280.62 | 227.38 | 749.68 | 205.68 | 669.48 | 594.91 |
| | | Wanda | 1068.58 | 1329864.75 | 2380391.50 | 3571925.75 | 9051510.00 | 14388.79 | 19762.86 |
| | | AWQ | 8.06 | 3.56 | 4.25 | 5.48 | 4.26 | 9.14 | 9.09 |
| | | GPTQ | 8.20 | 3.73 | 4.42 | 5.66 | 4.50 | 9.49 | 9.40 |

Table 6: Multilingual PPL results across models, sparsity levels, and compression methods. 4-bit quantization corresponds to approximately 75% sparsity, 8-bit to 50%.

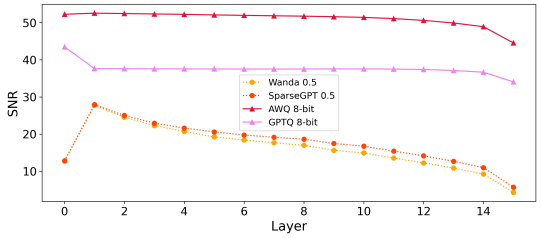


Figure 8: Layer-wise activation SNR on English tokens for Llama3.2.

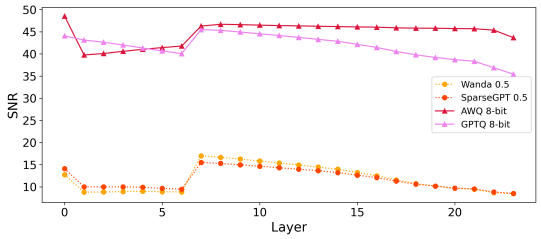


Figure 9: Layer-wise activation SNR on English tokens for SmolLM2.

Other tasks (e.g., BoolQ, RTE, HellaSwag) primarily assess single-step reasoning or commonsense understanding.

## H Cosine Similarity

We visualize layer-wise cosine similarity between the hidden states of compressed and original models in Fig.4. Quantization methods preserve higher similarity across layers and models. Pruning methods show lower similarity, particularly in deeper layers. The similarity decreases with more aggressive compression, but the relative advantage of quantization remains consistent.

## I Limitations of Wanda in SLMs

The contrast between our findings on SLMs and prior work on LLMs (where Wanda typically outperforms SparseGPT) is noteworthy and merits further discussion. We believe several factors might contribute to this divergence.

| Model | SNR | Error |
|---|---|---|
| SparseGPT (unstructured 50%) | 11.94 | 2.77e-4 |
| Wanda (unstructured 50%) | 11.66 | 2.76e-4 |
| GPTQ INT8 | 43.16 | 1.86e-7 |
| AWQ INT8 | **48.85** | **5.56e-8** |
| SparseGPT (unstructured 75%) | 4.85 | 16.47e-4 |
| Wanda (unstructured 75%) | 1.56 | 25.29e-4 |
| GPTQ INT4 | 18.91 | 0.49e-4 |
| AWQ INT4 | 18.59 | 0.53e-4 |

Table 7: Compression results for Qwen2.5-0.5B-Instructure under different compression methods. The column-wise best is highlighted in **bold**.

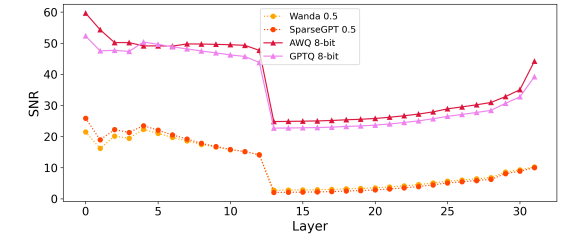


Figure 10: Layer-wise activation SNR on English tokens for Phi3.5.

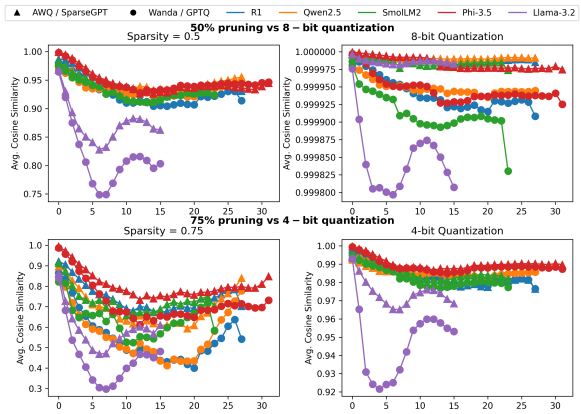


Figure 11: Layer-wise averaged cosine similarity of quantized and pruned models compared to the original model. Quantization is shown with different y-axis scales.

First, LLMs have more redundant weights, allowing pruning (even heuristic methods like Wanda) to preserve performance. Wanda's success on LLMs, e.g. Llama 7B, attests that many weights in a 7B+ model can be removed without significant impact, the remaining weights can compensate, and important features are preserved. In smaller models, by contrast, there are fewer extraneous weights. Every parameter likely has a higher relative contribution to some facet of the model's knowledge or capacity. Therefore, pruning 50% of a 1.5B model removes a much larger fraction of the model's "knowledge" than pruning 50% of a 13B model. Wanda performs better than magnitude, but it is still essentially 'a heuristic selection of weights'; its design doesn't attempt to minimize the difference before and after pruning, but rather removes weights with smaller activations. On the other hand, SparseGPT uses a second-order (Hessian-based) approximation to measure the sensitivity of the model's outputs to

| Model | en | hi | zh | th | de | ar | es |
|---|---|---|---|---|---|---|---|
| Full Size | 16.29 | 6.66 | 22.62 | 6.24 | 20.89 | 18.99 | 15.21 |
| Sparsegpt (0.5) | 22.21 | 67.18 | 14.88 | 45.19 | 16.01 | 46.40 | 36.92 |
| Wanda (0.5) | 25.62 | 50.39 | 14.12 | 47.91 | 16.23 | 55.51 | 44.83 |
| GPTQ (INT8) | **14.92** | 16.90 | **7.09** | **16.64** | **7.22** | **20.57** | **16.69** |
| AWQ (INT8) | **14.92** | **16.89** | **7.09** | **16.63** | **7.22** | **20.56** | **16.68** |
| Sparsegpt (0.75) | 322.83 | 2026.32 | 975.30 | 2555.91 | 473.33 | 1964.03 | 2108.29 |
| Wanda (0.75) | 913.84 | 7710.46 | 2358.86 | 4268.96 | 2606.93 | 5548.94 | 5131.00 |
| GPTQ (INT4) | 16.97 | 22.84 | 9.47 | 21.13 | 9.59 | 26.00 | 19.93 |
| AWQ (INT4) | 17.38 | 22.40 | 9.09 | 20.62 | 8.93 | 26.10 | 20.08 |

Table 8: New multilingual perplexity results for Qwen2.5-0.5-Instructure under different compression methods. The column-wise best is highlighted in **bold**.

| Model | BoolQ | RTE | HellaSwag | WinoGrande | ARC-e | ARC-c | OBQA |
|---|---|---|---|---|---|---|---|
| Full Size | 67.95 | 62.45 | 40.63 | 55.88 | 65.53 | 30.89 | 24.2 |
| Sparsegpt (0.5) | 41.22 | 51.62 | 35.76 | 54.06 | 57.03 | 24.83 | 20.2 |
| Wanda (0.5) | 58.29 | 53.43 | 34.44 | 52.64 | 57.79 | 25.09 | 19.4 |
| GPTQ (INT8) | 66.63 | 63.54 | 39.68 | 55.88 | 54.80 | 27.47 | 24.0 |
| AWQ (INT8) | 66.42 | 62.45 | 39.67 | **56.20** | 55.35 | 28.33 | 24.2 |
| Sparsegpt (0.75) | 43.33 | 56.32 | 27.00 | 49.57 | 33.67 | 18.69 | 16.0 |
| Wanda (0.75) | 37.83 | 52.71 | 26.53 | 49.57 | 28.96 | 18.43 | 12.6 |
| GPTQ (INT4) | 65.32 | **65.70** | 38.25 | 51.46 | 51.64 | 27.82 | 22.4 |
| AWQ (INT4) | 65.66 | 59.21 | 37.90 | 52.80 | 51.98 | 27.30 | 20.4 |

Table 9: New downstream tasks results for Qwen2.5-0.5-Instruct under different compression methods. The column-wise best is highlighted in **bold**.

each weight, which can benefit SLMs more than LLMs, as SLMs have fewer redundant weights.

Second, Wand lacks weight adjustment and error compensation. In a small model with limited redundancy, any one-shot removal of weights leaves more unrecoverable damage. SparseGPT mitigates this damage by reallocating some weight importance during its adjustment step. On the other hand, Wanda's design, which assumes large-model characteristics (many dispensable weights and obvious outlier features), is less suited to the small-model regime. Its lack of any weight adjustment leaves small models struggling to recover from pruning-induced damage.

Third, Wanda's core assumption leverages the phenomenon of outlier features, a well-documented effect in LLMs where a few dimensions in each layer's activation have disproportionately large variance or magnitude. These outlier dimensions often carry crucial semantic information (e.g. they correlate with rare but important token patterns) and must not be pruned aggressively. Wanda implicitly protects those: even a small weight on an outlier neuron won't be pruned if the activation norm is huge, because the weight×activation score would still be relatively large. In large models (like 13B, 70B), such outlier channels are prominent and Wanda's heuristic is very effective at avoiding the truly important weights. However, outlier behavior is less studied and potentially less pronounced in SLMs. If outlier-driven structure is weaker or absent, Wanda's assumptions may not hold, leading to less effective pruning.

## J 2:4 Structured Pruning

Table 11 and Table 12 report the results on 2:4 structured pruning using Wanda and SparseGPT. Overall, unstructured pruning consistently outperforms structured pruning, in line with prior findings in LLMs.

This finding has practical implications, especially for SLM practitioners: since SLMs are already small and require less compute, which lowers the barrier for applying unstructured pruning, practitioners have the flexibility to choose between structured and unstructured approaches. Our results suggest that unstructured pruning remains preferable for optimal performance.

### J.1 Models in comparable size

| Model | Sparsity | Method | BoolQ | RTE | HellaSwag | WinoGrande | ARC-e | ARC-c | OBQA |
|---|---|---|---|---|---|---|---|---|---|
| R1 | 0.00 | Full Precision | 51.10 | 46.93 | 36.13 | 51.07 | 42.93 | 27.56 | 18.60 |
| | 0.5 | SparseGPT | 66.21 | 55.96 | 33.95 | 55.72 | 54.34 | 29.44 | 19.80 |
| | | Wanda | 64.56 | 57.76 | 33.27 | 54.14 | 54.08 | 28.24 | 18.80 |
| | | AWQ | 50.06 | 48.74 | 36.03 | 50.99 | 42.55 | 27.99 | 18.40 |
| | | GPTQ | 54.16 | 47.65 | 36.20 | 50.67 | 43.10 | 27.73 | 18.80 |
| | 0.75 | SparseGPT | 61.16 | 52.35 | 27.11 | 49.80 | 27.82 | 19.71 | 12.60 |
| | | Wanda | 59.76 | 52.71 | 26.36 | 49.01 | 28.16 | 20.31 | 14.40 |
| | | AWQ | 61.07 | 52.35 | 35.81 | 50.51 | 42.21 | 28.92 | 17.60 |
| | | GPTQ | 53.67 | 54.15 | 35.46 | 52.17 | 42.89 | 26.96 | 19.00 |
| Qwen2.5 | 0.00 | Full Precision | 78.32 | 73.65 | 49.30 | 59.59 | 68.39 | 38.48 | 31.20 |
| | 0.5 | SparseGPT | 76.36 | 75.45 | 44.61 | 61.80 | 68.01 | 36.18 | 27.80 |
| | | Wanda | 69.39 | 67.51 | 42.28 | 60.06 | 66.88 | 34.38 | 24.40 |
| | | AWQ | 78.13 | 75.09 | 49.21 | 59.35 | 68.64 | 39.25 | 30.80 |
| | | GPTQ | 78.23 | 74.73 | 49.21 | 59.35 | 68.86 | 38.48 | 31.00 |
| | 0.75 | SparseGPT | 56.48 | 50.18 | 27.35 | 49.57 | 29.29 | 19.28 | 12.00 |
| | | Wanda | 51.13 | 51.99 | 26.11 | 51.07 | 27.65 | 19.71 | 11.60 |
| | | AWQ | 76.51 | 70.40 | 48.19 | 61.48 | 66.37 | 36.60 | 29.40 |
| | | GPTQ | 77.49 | 70.76 | 48.26 | 57.85 | 66.20 | 39.16 | 25.60 |
| Llama3.2 | 0.00 | Full Precision | 57.28 | 47.29 | 43.73 | 56.75 | 58.12 | 31.31 | 24.80 |
| | 0.5 | SparseGPT | 64.68 | 54.15 | 38.27 | 55.72 | 58.00 | 28.33 | 21.60 |
| | | Wanda | 62.63 | 53.79 | 35.67 | 55.8 | 54.76 | 24.57 | 18.40 |
| | | AWQ | 55.26 | 48.38 | 43.64 | 56.27 | 57.58 | 31.57 | 23.80 |
| | | GPTQ | 57.22 | 46.93 | 43.8 | 56.99 | 58.12 | 31.48 | 24.80 |
| | 0.75 | SparseGPT | 55.05 | 53.07 | 26.75 | 50.2 | 32.24 | 19.2 | 14.20 |
| | | Wanda | 41.04 | 52.71 | 26.08 | 49.49 | 27.36 | 20.05 | 12.40 |
| | | AWQ | 56.21 | 47.29 | 43.26 | 56.51 | 58.21 | 30.89 | 24.60 |
| | | GPTQ | 66.76 | 54.87 | 42.62 | 54.7 | 54.97 | 31.66 | 25.40 |
| SmolLM2 | 0.00 | Full Precision | 75.81 | 70.04 | 51.75 | 62.75 | 66.29 | 39.42 | 28.00 |
| | 0.5 | SparseGPT | 69.85 | 53.43 | 44.92 | 62.27 | 68.14 | 37.03 | 26.80 |
| | | Wanda | 67.16 | 55.23 | 42.91 | 59.83 | 67.76 | 32.76 | 25.00 |
| | | AWQ | 75.78 | 69.31 | 51.71 | 63.14 | 65.78 | 39.51 | 28.80 |
| | | GPTQ | 76.30 | 68.23 | 51.68 | 62.90 | 66.04 | 39.59 | 28.20 |
| | 0.75 | SparseGPT | 38.56 | 51.99 | 27.16 | 49.64 | 31.31 | 18.60 | 12.60 |
| | | Wanda | 37.80 | 53.79 | 26.10 | 49.72 | 28.45 | 19.54 | 14.60 |
| | | AWQ | 75.93 | 70.40 | 50.80 | 62.19 | 67.85 | 38.14 | 27.20 |
| | | GPTQ | 76.27 | 70.76 | 50.89 | 60.77 | 64.94 | 38.74 | 29.00 |
| Phi3.5 | 0.00 | Full Precision | 85.84 | 80.87 | 56.32 | 68.59 | 76.14 | 50.68 | 33.60 |
| | 0.5 | SparseGPT | 82.63 | 70.76 | 52.02 | 67.64 | 74.58 | 46.16 | 35.80 |
| | | Wanda | 83.00 | 71.84 | 50.45 | 67.01 | 77.65 | 47.35 | 33.00 |
| | | AWQ | 85.72 | 81.23 | 56.34 | 68.11 | 75.63 | 50.94 | 34.20 |
| | | GPTQ | 86.09 | 80.87 | 56.37 | 68.11 | 76.30 | 50.60 | 34.20 |
| | 0.75 | SparseGPT | 62.26 | 55.6 | 29.12 | 51.38 | 30.89 | 20.65 | 11.60 |
| | | Wanda | 51.35 | 52.35 | 26.12 | 49.88 | 28.07 | 19.11 | 14.80 |
| | | AWQ | 85.63 | 79.78 | 54.96 | 66.85 | 75.04 | 50.17 | 31.80 |
| | | GPTQ | 85.63 | 80.87 | 54.69 | 66.61 | 76.56 | 49.74 | 34.80 |

Table 10: Performance (%) on English downstream benchmarks: BoolQ, RTE, HellaSwag, WinoGrande, ARC-e, ARC-c, and OBQA, across models, sparsity levels, and pruning/quantization methods.

| Model | SNR | Error (x1e-4) | en | ar | hi | zh | th | de | es |
|---|---|---|---|---|---|---|---|---|---|
| R1 Original | - | - | 29.79 | 159.39 | 11.02 | 41.37 | 40.67 | 98.58 | 85.36 |
| Wanda (Structured 2:4) | 7.08 | 3.97 | 148.92 | 4864.43 | 54.45 | 514.27 | 666.37 | 1024.4 | 738.42 |
| **Wanda (Unstructured 0.5)** | **11.11** | **1.59** | **47.75** | **451.78** | **16.85** | **114.34** | **170.08** | **206.75** | **164.00** |
| SparseGPT (Structured 2:4) | 9.64 | 2.19 | 61.14 | 664.81 | 30.96 | 232.78 | 258.91 | 367.24 | 254.92 |
| **SparseGPT (Unstructured 0.5)** | **13.61** | **0.86** | **44.41** | **379.12** | **16.26** | **117.87** | **306.32** | **193.78** | **162.97** |

Table 11: Compression performance and multilingual perplexity of DeepSeek R1 Distil Qwen 1.5B. The better result between structured and unstructured is highlighted in **bold**. Due to rebuttal time constraints, results for the other four models will be provided in the final version.

| Model | BoolQ | RTE | HellaSwag | WinoGrande | ARC-e | ARC-c | OBQA |
|---|---|---|---|---|---|---|---|
| R1 Original | 51.10 | 46.93 | 36.13 | 51.07 | 42.93 | 27.56 | 18.60 |
| R1 Wanda (Structured 2:4) | 62.05 | 51.62 | 29.89 | 52.09 | 43.48 | 23.46 | 15.60 |
| **R1 Wanda (Unstructured)** | **64.56** | **57.76** | **33.27** | **54.14** | **54.08** | **28.24** | **18.80** |
| R1 SparseGPT (Structured 2:4) | 62.75 | 52.71 | 32.01 | 52.88 | 49.45 | 24.40 | 16.60 |
| **R1 SparseGPT (Unstructured)** | **66.21** | **55.96** | **33.95** | **55.72** | **54.34** | **29.44** | **19.80** |

Table 12: Performance (%) of DeepSeek R1 Distill Qwen 1.5B on English downstream benchmarks. The better result between structured and unstructured is highlighted in **bold**. Due to rebuttal time constraints, results for the other four models will be provided in the final version.

| Model | BoolQ | RTE | HellaSwag | WinoGrande | ARC-e | ARC-c | OBQA |
|---|---|---|---|---|---|---|---|
| Uncompressed Qwen-0.5B | 67.95 | 62.45 | 40.63 | 55.88 | 65.53 | 30.89 | 24.20 |
| Llama3.2-1B (0.5 Wanda) | 62.63 | 53.79 | 35.67 | 55.80 | 54.76 | 24.57 | 18.40 |
| Llama3.2-1B (0.5 SparseGPT) | 64.68 | 54.15 | 38.27 | 55.72 | 58.00 | 28.33 | 21.60 |
| Llama3.2-1B (AWQ 8) | 55.26 | 48.38 | 43.64 | 56.27 | 57.58 | 31.57 | 23.80 |
| Llama3.2-1B (GPTQ 8) | 57.22 | 46.93 | 43.80 | 56.99 | 58.12 | 31.48 | 24.80 |
| SmolLM2-1.7B (0.75 Wanda) | 37.80 | 53.79 | 26.10 | 49.72 | 28.45 | 19.54 | 14.60 |
| SmolLM2-1.7B (0.75 SparseGPT) | 38.56 | 51.99 | 27.16 | 49.64 | 31.31 | 18.60 | 12.60 |
| SmolLM2-1.7B (AWQ 4) | **75.93** | **70.40** | **50.80** | **62.19** | **67.85** | **38.14** | 27.20 |
| SmolLM2-1.7B (GPTQ 4) | 76.27 | 70.76 | 50.89 | 60.77 | 64.94 | 38.74 | **29.00** |

Table 13: Downstream performance of models in comparable sizes. The column-wise best is highlighted in **bold**.