Frequency Matters: Fast Model-Agnostic Data Curation for Pruning and Quantization

Francesco Pio Monaco **Elia Cunegatti** **Flavio Vella** **Giovanni Iacca**
University of Trento
{francescopio.monaco, elia.cunegatti, flavio.vella, giovanni.iacca}@unitn.it

arXiv:2603.16105v4 [cs.CL] 27 Aug 2026

Abstract

Post-training model compression is essential for enhancing the portability of Large Language Models (LLMs) while preserving their performance. While several compression approaches have been proposed, less emphasis has been placed on selecting the most suitable set of data (the so-called *calibration data*) for finding the compressed model configuration. The choice of calibration data is a critical step in preserving model capabilities both intra- and inter-tasks. In this work, we address the challenge of identifying high-performance calibration sets for both pruning and quantization by analyzing intrinsic data properties rather than model-specific signals. We introduce **ZipCal**, a model-agnostic data curation strategy that maximizes lexical diversity based on Zipfian power laws. Experiments demonstrate that our method outperforms standard uniform random sampling across various pruning benchmarks. Notably, it also performs on par, in terms of downstream performance, with a state-of-the-art method that relies on model perplexity. The latter becomes prohibitively expensive for large-scale models and datasets, while **ZipCal** is on average $\sim$240$\times$ faster due to its tractable linear complexity[1].

1 Introduction

In recent years, considerable effort has been put into alleviating the significant computational requirements of Large Language Models (LLMs) (Xia et al., 2024; Muralidharan et al., 2024; Lee et al., 2025). To facilitate the deployment of such models, researchers have developed various post-training compression techniques that reduce memory and compute overhead while striving to preserve the original model's capabilities. In this work, we mainly focus on two such compression techniques, which are pruning and quantization. The former compresses the model by removing part of its parameters, with earlier approaches requiring expensive retraining (Han et al., 2015; Frankle and Carbin, 2019), and more modern approaches simply removing weights in a gradient-free manner (Frantar and Alistarh, 2023; Sun et al., 2024b), or enforcing hardware-friendly sparsity patterns (Sun and Sakuma, 2026) without the need for parameter updates (Zhang et al., 2024b; Cunegatti et al., 2025). Quantization, on the other hand, compresses the model by reducing the numerical precision of weights and activations (Frantar et al., 2023; Lin et al., 2024; Huang et al., 2025). The latest approaches are based on ternary quantized models (Wang et al., 2023; Ma et al., 2024) or efficient binary MatMul techniques (Dehghankar et al., 2025).

Beyond the choice of the compression approach, recent works (Bandari et al., 2024; Williams and Aletras, 2024; Oh and Oh, 2025) explored how the selection of the data used for gathering model statistics during the compression process, called *calibration data*, can influence the process and, as a result, the final compressed model capabilities. While the majority of compression techniques (Frantar and Alistarh, 2023; Lin et al., 2024) rely on general-purpose datasets, such as C4 (Raffel et al., 2020) or Pile (Gao et al., 2020), it has been investigated that domain-specific calibration has significant impact on the downstream performances of the model (Bandari et al., 2024; Zhang et al., 2024a; Williams et al., 2026). However, a compressed model can be deployed across multiple, and at times unforeseen, downstream tasks, raising the question of whether a calibration strategy can be optimal on average across a spread of domains rather than tailored to any single one. A question that, to our knowledge, remains unaddressed. We investigate this via a Multi-Domain extension of our method that aggregates candidate calibration data across domains without requiring prior knowledge of the specific deployment task. We further explore an

[1]We make the code and the experiments available at https://github.com/FrancescoMonaco/ZipCal.

orthogonal and underexplored axis of calibration data selection: language. While prior work considers task or data alignment between calibration and deployment data, whether alignment in language between calibration and downstream data is similarly necessary has not been studied.

To analyze the impact of selected data for compression, we leverage several studies on statistical linguistics that show that human languages show a structure in the frequency distribution of words (Zipf, 1932; Piantadosi, 2014). Specifically, we investigate whether this statistical observation can be exploited for data curation of calibration sets for model compression. We hypothesize that a set of data samples maximizing lexical diversity, capturing the sparse *tail* of the Zipfian distribution, can provide much richer and more representative information for compression algorithms. By focusing on these intrinsic linguistic properties, we propose a sampling strategy that identifies high-utility, *model-agnostic* data with negligible computational overhead. Our results show that this linguistically informed approach performs on par with more expensive, model-dependent curation methods, offering a scalable and robust solution for billion-parameter model compression that generalizes across diverse downstream tasks.

Data Curation Goals An ideal technique for data curation must be *scalable* **G1**, i.e., capable of processing massive corpora with minimal computational overhead; *model-agnostic* **G2**, i.e., capable of identifying the most informative examples from a corpus without relying on expensive model passes; and address *inter-domain generalization* **G3**, i.e., being capable of synthesizing both Single-Domain and Multi-Domain corpora settings by design. Existing data curation techniques only fulfill a subset of these goals, and, to our best knowledge, none have yet explored **G3**.

Core Contributions We propose a sampling strategy rooted in Zipfian statistics that identifies high-utility calibration samples by maximizing lexical diversity. This approach sidesteps the need for model-dependent metrics (e.g., perplexity or gradient information), achieving goals **G1** and **G2**. Moreover, we provide a framework for extracting representative samples from Multi-Domain datasets, ensuring the calibration set is balanced, meeting goal **G3**. We empirically evaluate our proposed method, called **`ZipCal`**, against SoTA techniques for data curation for compression algorithms. We conduct a comprehensive analysis against baselines based on alternative data properties, showing that capturing lexical diversity is a stable and scalable proxy for data curation.

2 Related Work

Model Compression Pruning methods aim to remove parameters to reduce the size of the models. The first approaches to pruning involved estimating the contribution of each neuron to the final loss using the magnitude of the weights and gradient information (Molchanov et al., 2019; Ma et al., 2023). More recently, different gradient-free no-retraining pruning modalities have been explored. Unstructured approaches zero-out individual weights (Frantar and Alistarh, 2023; Sun et al., 2024b; Yang et al., 2025b), while semi-structured methods enforce specific sparsity patterns to ensure hardware compatibility (Zhou et al., 2021). Structured approaches, instead, remove entire architectural components, such as attention heads or layer rows/columns (Ashkboos et al., 2024; Sandri et al., 2025; Guo et al., 2025).

On the other hand, quantization methods reduce the numerical precision of weights and/or activations to reduce the memory footprint and accelerate inference. These methods range from Round-To-Nearest (RTN) (Nagel et al., 2020) to more sophisticated error-minimization strategies (Nahshan et al., 2021; Chen et al., 2025). Other works focus on reducing degradation in extreme quantization scenarios (Dettmers et al., 2022), employ optimized kernels like LUT-GEMM (Park et al., 2024), or activation-aware scaling (Lin et al., 2024).

Calibration Data Almost every compression algorithm relies on a small, representative *calibration set* to estimate the information flow through the network. These statistics guide the compression process, determining quantization thresholds (Lin et al., 2024) or pruning scores (Sun et al., 2024b). Recent studies have shown that the choice of the calibration source significantly impacts the performance of the compressed model (Williams and Aletras, 2024), revealing that general-purpose corpora like C4 (Raffel et al., 2020) are not the optimal choice for downstream tasks (Bandari et al., 2024). This suggests that calibration data should mirror the target domain to prevent activation distribution shifts, which can lead to suboptimal quantization thresholds or pruning masks.

Researchers have proposed more sophisticated

calibration data curation strategies. Marion et al. (Marion et al., 2023) demonstrate that perplexity serves as a robust metric to rank and select the most impactful samples for pruning, essentially using the model's own likelihood as a proxy for quality. Extending this logic, COLA (He et al., 2025) introduces a hybrid approach that selects samples by balancing the magnitude of model activations with intrinsic data statistics. While effective, these methods are inherently model-dependent and computationally intensive. Moreover, these techniques are typically evaluated on single-source datasets and do not address the challenge of heterogeneous composability **G3**, while it would be desirable to have stable, cross-task performance post-compression.

3 Zipf Sampling

Natural languages are characterized by a Zipfian distribution (Piantadosi, 2014), where a small number of words appear with high frequency, while most of the vocabulary resides in an increasingly sparse long tail. Not acknowledging this sparsity implies potentially omitting the rare tokens and diverse semantic contexts that trigger critical activation outliers in LLMs.

3.1 Single-Domain Sampling

We propose to sample calibration data from a dataset by maximizing the *lexical diversity* of the calibration set within a constrained number of samples. Specifically, we apply a sanitization pass to the dataset where tokens are converted to lowercase to form a vocabulary V and special tokens (i.e., EOS) are removed. We then employ a randomized greedy selection heuristic to iteratively populate the calibration set. In each iteration, we select a sample s from a candidate pool P that maximizes the gain of the sample's vocabulary:

$$s^* = \arg\max_{s \in P} |V(s) \setminus \mathcal{V}_{covered}| \quad (1)$$

where $\mathcal{V}_{covered}$ is the vocabulary of sanitized tokens already present in previously selected samples. In the event of a tie, we prioritize the sample with the highest total number of unique tokens to maximize information density. This approach ensures that the resulting calibration data provides a high-fidelity approximation of the full dataset's vocabulary manifold. We present the procedure, named **ZipCal**, in Algorithm 1.

Algorithm 1: ZipCal

Input: Dataset D; Number of samples k
Output: Set of calibration samples S

1 S,$V_{covered} \leftarrow \emptyset$;
// Precalculate the full vocabulary
2 **foreach** *sample* $s \in \mathcal{D}$ **do**
3 $\quad V_s \leftarrow \{\text{sanitize}(t) \mid t \in s, t \notin \text{SpecialTokens}\}$;
4 **for** $i = 1$ **to** k **do**
$\quad$ // Select sample with maximum marginal vocabulary gain
5 $\quad s^* \leftarrow \arg\max_{s \in \mathcal{D} \setminus S} |V_s \setminus \mathcal{V}_{covered}|$;
6 $\quad S \leftarrow S \cup \{s^*\}$;
7 $\quad V_{covered} \leftarrow V_{covered} \cup V_{s^*}$;
8 **return** S ;

Lemma 3.1. *When* **ZipCal** *is used to extract a calibration set of k samples on dataset $\mathcal{D}$ of n elements, it completes the procedure in $O(nk)$ time.*

Proof. The selection process, lines 4 to 7 of Algorithm 1, consists of k iterations to construct the calibration set. At any iteration $i \in (1, k)$, we have to carry out $T_i = N - i + 1$ evaluations. To find out the total number of evaluations required to construct a calibration set of k samples, we have:

$$T_{tot} = \sum_{i=1}^{k} T_i = k \cdot n - \frac{k(k+1)}{2} + k \quad (2)$$

During each evaluation, we compute the set difference between the candidate's vocabulary V_s and the cumulative vocabulary $V_{covered}$. This cost scales with the number of unique elements in a sample, which is bounded by the context window size w. Since w is fixed a priori, its contribution to the asymptotic complexity is constant. Consequently, summing these contributions, the overall complexity is $O(kn)$. □

3.2 Multi-Domain Sampling

When calibrating models for general-purpose use or Multi-Domain applications, a single source of data is often insufficient. To address goal **G3**, we extend our approach to support heterogeneous multi-domain settings. Simply concatenating datasets and applying **ZipCal** over the joint dataset is suboptimal, as a single large or linguistically dense corpus might dominate the selection

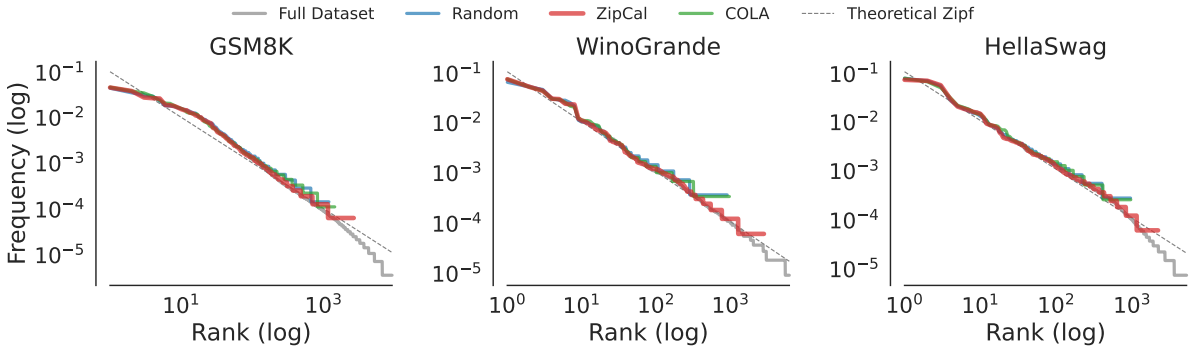


Figure 1: Token frequency distribution of the original datasets and the random, COLA, and **ZipCal** calibration sets.

process. To address this, we propose a hierarchical selection strategy. First, we apply **ZipCal** to each dataset $D_i \in \mathbb{D}$, to extract a local representative pool P_i of size k. This ensures each domain's unique vocabulary is captured. Second, we consolidate these pools into a candidate set $\mathcal{P} = \bigcup P_i$ and apply a greedy k-centers selection algorithm. By representing each sample using a lightweight embedding, the k-centers objective selects a final set S that maximizes the distance between samples, ensuring the calibration data is semantically spread across all provided domains.

Lemma 3.2. *The Multi-Domain* **ZipCal** *procedure extracts k samples from m datasets, each of length n_m, in $O(mNk)$ time, where $N = \max_m(n_m)$.*

4 Experiments

We present below the details of the experimental setup and the experimental results.

4.1 Experimental Setup

Post-training Compression We validate **ZipCal** across a number of post-training compression techniques that rely on calibration data. For pruning, we consider Wanda (Sun et al., 2024b), an unstructured approach which scores weight importance via the product of magnitudes and input activation norms $|W_{ij}| \cdot \|X_j\|_2$, and 2SSP (Sandri et al., 2025), a two-stage framework for structured pruning that balances width and depth reduction. For quantization, we evaluate GPTQ (Frantar et al., 2023), which minimizes layer-wise reconstruction error $\|\mathbf{W}X - \hat{\mathbf{W}}X\|_2^2$, and AWQ (Lin et al., 2024), which preserves salient weights critical to model performance by scaling them according to activation magnitude $|\mathbf{X}|$.

Experimental Environment We use two LLMs to perform our evaluation of downstream tasks: `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024) and `Gemma-2-9B-it` (Team et al., 2024). For language modeling evaluation, we use two base models, namely `Llama-3.1-8B` and `Gemma-2-9B` [2]. Unless otherwise specified, we set the context length to $w = 2048$ and the number of calibration samples $k = 128$.

Baselines We benchmark **ZipCal** firstly against random sampling, the approach used by almost any compression algorithm (Ashkboos et al., 2024; Sun et al., 2024b; Lin et al., 2024). Then, we evaluate whether our lightweight, model-agnostic approach can match or exceed the performance of a computationally expensive, model-dependent technique. Hence, we extensively compare against COLA (He et al., 2025), a recent state-of-the-art data curation method for compression algorithms that relies on both activation influence and data diversity metrics. To further validate **ZipCal**, we show the comparison with a self-generative approach (Ji et al., 2025) in Section E. We compare the performance evaluation of both intra- (i.e., when the evaluation tasks *match* with the calibration domain) and inter- (i.e., when the evaluation tasks *do not match* with the calibration domain) tasks, as well as the runtime between **ZipCal** and COLA.

Evaluation To assess downstream performance post-compression, we use the LM-Evaluation-Harness framework (Gao et al., 2023) across five functional domains and three languages (EN, ES, and ZH). Results are reported on the subset of datasets that support standardized zero- or few-shot metrics. We categorize the datasets in functional domains as follows. **(i) Language Modeling**: zero-shot perplexity is measured on WikiText (Merity

[2] Due to the number of combinations, executing the experiments over these models requires $\approx$ 1200 GPU hours.

Model	Task	Dense	COLA Calibration Category						ZipCal Calibration Category						Δ
			LangMod	Math	CommQA	NLI	KnowTran	Mean	LangMod	Math	CommQA	NLI	KnowTran	Mean	
Llama-3.1-8B-Instruct	MMLU-M	24.44	23.33	24.44	23.01	21.85	23.33	**23.19**	23.01	26.29	21.24	22.02	23.01	23.11	-0.08
	GSM8k	78.09	76.55	75.70	74.40	76.54	75.89	75.82	76.88	76.27	75.82	75.70	76.61	**76.25**	**+0.43**
	HellaSwag	71.71	72.03	72.09	71.76	71.67	71.86	**71.88**	71.78	71.90	71.83	71.59	71.80	71.78	-0.10
	WinoGr.	69.46	68.93	68.51	67.93	68.03	68.07	**68.30**	67.88	68.43	68.82	68.15	68.03	68.26	-0.04
	OBQA	47.80	46.67	47.30	46.67	46.20	47.10	46.79	47.90	47.50	48.10	47.30	47.80	**47.72**	**+0.93**
	BoolQ	84.46	84.99	84.74	84.41	84.85	84.91	**84.78**	84.50	84.45	84.83	84.60	84.48	84.57	-0.21
	RTE	74.73	72.32	72.20	71.96	72.38	72.38	72.25	72.20	72.38	72.74	72.74	73.10	**72.64**	**+0.39**
	ANLI	58.40	56.37	59.00	59.33	57.10	57.80	57.92	61.50	58.50	59.80	59.65	61.95	**60.28**	**+2.36**
	ARC-C	51.71	51.54	50.30	50.28	51.24	50.77	50.82	51.45	50.98	51.58	51.32	51.11	**51.29**	**+0.47**
	ARC-E	73.86	73.27	73.46	73.13	73.21	73.74	73.36	73.86	74.05	73.74	73.84	73.93	**73.88**	**+0.52**
	MMLU-K	62.28	62.08	62.51	62.04	61.94	61.56	62.02	62.23	62.68	62.70	62.44	62.58	**62.53**	**+0.51**
	Mean	63.36	62.55	62.75	62.27	62.27	62.49	62.47 ±0.19	63.02	63.04	62.84	62.67	63.13	**62.94** ±0.17	**+0.47**
	Runtime		5400s	36s	3240s	2160s	1380s	2443s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**228**×
gemma-2-9b-it	MMLU-M	21.48	24.44	23.01	22.02	21.24	21.85	**22.51**	27.40	24.44	26.29	28.79	24.44	**26.27**	**+3.76**
	GSM8k	75.44	74.05	74.34	74.98	74.30	73.88	74.31	74.32	74.41	75.09	74.26	74.37	**74.49**	**+0.18**
	HellaSwag	67.24	67.11	67.23	67.08	67.06	67.22	67.14	67.27	67.15	67.23	67.34	67.34	**67.27**	**+0.13**
	WinoGr.	70.48	69.24	69.06	68.82	69.26	69.65	**69.21**	69.11	68.63	69.18	69.34	68.82	69.02	-0.19
	OBQA	45.40	46.07	46.10	46.40	45.70	46.30	**46.11**	45.33	45.60	45.40	45.20	45.40	45.39	-0.72
	BoolQ	88.59	88.92	88.62	88.65	88.81	88.64	88.73	88.80	88.79	88.79	88.72	88.98	**88.81**	**+0.08**
	RTE	78.34	78.22	78.88	77.80	78.34	78.70	78.39	78.46	78.34	78.52	78.52	78.34	**78.44**	**+0.05**
	ANLI	72.80	72.37	73.30	73.05	72.40	72.15	72.65	73.27	73.15	72.60	73.05	73.00	**73.01**	**+0.36**
	ARC-C	51.79	52.13	51.37	51.71	51.62	51.83	**51.73**	51.48	51.54	51.54	51.79	51.32	51.53	-0.20
	ARC-E	66.79	67.51	67.09	66.94	67.28	67.23	**67.21**	66.72	66.75	66.90	66.96	66.90	66.85	-0.36
	MMLU-K	33.83	35.90	33.53	32.94	36.80	33.94	34.62	36.95	36.07	36.42	35.88	35.63	**36.19**	**+1.57**
	Mean	61.11	61.45	61.14	60.94	61.16	61.04	61.15 ±0.81	61.74	61.35	61.63	61.80	61.32	**61.57** ±0.82	**+0.42**
	Runtime (sec)		6231s	149s	3400s	2671s	1500s	2790s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**260**×

Table 1: Comparison of COLA vs. **ZipCal** calibration data under Wanda unstructured pruning at 25% sparsity.

et al., 2016), C4 (Raffel et al., 2020), and Pile (Gao et al., 2020), which are general datasets that have commonly been used for model compression. **(ii) Mathematical Reasoning**: evaluated via GSM8k (5-shot) (Cobbe et al., 2021), SVAMP (Patel et al., 2021), and MMLU-M the subset of math tasks in MMLU. **(iii) Commonsense Reasoning & QA**: zero-shot assessed using WinoGrande (Keisuke et al., 2021), XWinograd (Tikhonov and Ryabinin, 2021), CommonsenseQA (Talmor et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018) and XCOPA (Ponti et al., 2020). **(iv) Natural Language Inference (NLI)**: tested in zero-shot on RTE (Wang et al., 2018) and the adversarial ANLI (Nie et al., 2020) benchmarks. **(v) Knowledge & Translation**: general world knowledge and reasoning are measured via MMLU-K, MMLU-ES, and MMLU-ZH, the MMLU corpus with math tasks excluded (Hendrycks et al., 2021; Singh et al., 2025), XQuAD (Artetxe et al., 2020), XNLI (Conneau et al., 2018), and ARC (Clark et al., 2018), while translation capabilities are tested on WMT14 (Bojar et al., 2014).

4.2 Experimental Results

Better than Random The purpose of this experiment is to verify the hypothesis that **ZipCal** identifies higher-utility data compared to standard uniform random sampling. Random sampling is statistically prone to overrepresent high-frequency tokens while failing to capture the *Zipfian tail* composed of tokens that are an integral part of the corpora, as shown in Figure 1. Moreover, as numerically shown in Table 2, **ZipCal** wins 24 of 33 comparisons against random selection across the different compression techniques. On the few occasions where random selection performs best, it does so by a small margin ($< 1\%$). Most notably, we observe significant gains in reasoning-intensive tasks: on ANLI, our method improves accuracy by 7.4% for GPTQ and 2.7% for Wanda. Similarly, in GSM8K, we achieve a 2.1% boost in the pruning

Task	Dense	Wanda (25%)		GPTQ (W4A16)		AWQ (W4A16)	
		Rand.	ZipCal	Rand.	ZipCal	Rand.	ZipCal
MMLU-M	24.44	23.33	**24.14**	23.33	**24.07**	23.12	**24.07**
GSM8K	78.08	74.98	**77.17**	70.12	**70.63**	36.13	**38.46**
HellaSwag	71.70	**64.49**	64.28	**64.23**	64.20	**64.85**	**64.85**
WinoGr.	73.56	67.86	**68.07**	67.21	**67.39**	67.48	**68.71**
OBQA	47.80	41.83	**42.42**	**42.02**	41.10	**40.90**	40.45
BoolQ	85.41	**84.80**	84.64	**83.53**	83.44	83.98	**84.36**
RTE	74.37	72.39	**72.77**	71.75	**73.06**	**74.73**	73.47
ANLI	58.40	57.90	**60.64**	46.50	**53.90**	51.10	**53.20**
ARC-C	53.75	50.33	**50.89**	50.03	**50.37**	51.41	**51.54**
ARC-E	82.24	76.48	**76.67**	**76.27**	76.22	76.47	**76.49**
MMLU-K	62.28	62.24	**62.46**	57.73	**58.08**	56.51	**57.49**
Mean	64.73	61.51	**62.20**	59.34	**60.22**	56.97	**57.55**

Table 2: Performance comparison against random sampling across different tasks and compression techniques for `Meta-Llama-3.1-8B-Instruct`.

setting with Wanda. This increased lexical representation correlates directly with performance; by forcing the calibration set to cover a broader vocabulary manifold, Zipf Sampling provides the compression algorithms with a more representative set of activation outliers (Sun et al., 2024a; An et al., 2025). This leads to more robust statistics, effectively bridging the gap between the original dense model and its compressed counterpart.

Evaluating Single-Domain Data Curation We now discuss the evaluation against COLA on models compressed using a Single-Domain calibration source (i.e, the calibration data are extracted from a single domain). Tables 1 and 3 shows how our proposed approach achieves on-par performance w.r.t. the baselines. Considering only the mean across all possible <task,domain> pairs, **ZipCal** performs better than COLA in 3 out of 4 cases. More importantly, as stated at the beginning of the paper, it reduces the data curation bottleneck from over an hour to mere seconds, yielding a $228-260\times$ acceleration while preserving downstream capabilities on par with computationally expensive baselines.

Furthermore, the results in Tables 1 and 3 demonstrate that Single-Domain compression is highly sensitive to the choice of calibration data. Contrary to intuition, matching the calibration source domain to the task domain (e.g., using Commonsense Reasoning & QA for BoolQ) does not universally hold the best performance (best is underlined in Tables 1 and 3, while domain-task match cells are colored). The results are also in line with (Bandari et al., 2024), where the authors showed how Language Modeling corpora are not inherently the optimal calibration source for diverse downstream tasks. This domain sensitivity finding introduces a form of intra-task sub-optimality that limits the possibility of deploying a unique compressed model that can achieve reasonable performance on a predefined downstream domain task.

Evaluating Multi-Domain Data Curation We tested our Multi-Domain approach of **ZipCal**, Section 3.2, to see if it can effectively bypass the aforementioned intra-task sub-optimality, as well as the inter-task limitations of real-world model deployment, where the specific downstream task domain that will be used is unknown *a priori*. By aggregating Zipfian subsets from multiple domains, we produce a single general calibration set. We show in Table 4 that models compressed using a Multi-Domain source perform better overall on average across different tasks. Most notably, Multi-Domain calibration delivers a model that achieves a score of 64.48 for `Llama-3.1-8B` with Wanda pruning, which is a higher score than any individual calibration domain within the Single-Domain compressed models within **ZipCal** and COLA groups. These results highlight that the Multi-Domain version of **ZipCal** can solve both the intra-suboptimality issue as well as the inter-task deployment.

Evaluating Single-Domain Single-Language Data Curation Tables A5 to A8 show the results among all the compression techniques in our testbed. In 5 out of 8 model-compression configurations, models compressed with **ZipCal**-curated data outperform those compressed using COLA-curated data. In the remaining cases, the performance degradation is marginal (-0.30 on average, -0.40 at most). Furthermore, we emphasize that **ZipCal** achieves these results while retrieving the curation data in mere seconds, compared to the significantly longer processing time required by COLA (which averages 17 minutes per dataset).

We previously discussed that there is little to no correlation between the calibration data domain and higher performance on downstream tasks of that same domain. The multilingual results allow us to further expand this claim: counterintuitively, utilizing calibration data in a specific language does not guarantee optimal compression for downstream tasks in that identical language. We observe that performance on MMLU-ES is actually higher when the model is compressed using Chinese calibration data. Similarly, MMLU-ZH and XNLI-ZH frequently reach peak performance when models are compressed using Spanish data. This pattern of mismatches repeats consistently across the different compression techniques. We can confidently develop the observation from Bandari et al. (Bandari et al., 2024) about the mismatch between the calibration data domain and the higher performance in downstream tasks of the same domain to languages. The optimal calibration data for a downstream task in a certain language does not necessarily coincide with the data from that same language.

Evaluating Multi-Domain Multi-Language Data Curation We have seen that language alignment between calibration data and evaluation tasks is neither a necessary nor sufficient condition for preserving performance. This mismatch adds another layer of complexity to the manual selection of calibration data, as the natural intuition on match-

Model	Task	Dense	COLA Calibration Category LangMod	Math	CommQA	NLI	KnowTran	COLA Mean	ZipCal Calibration Category LangMod	Math	CommQA	NLI	KnowTran	ZipCal Mean	Δ
Llama-3.1-8B-Instruct	MMLU-M	24.44	23.33	24.44	23.01	21.85	23.33	**23.19**	23.01	26.29	21.24	22.02	23.01	23.11	-0.08
	GSM8k	78.09	71.75	68.46	68.84	63.99	65.01	67.61	71.99	74.60	65.13	70.96	67.32	**70.00**	**+2.39**
	HellaSwag	71.71	72.07	72.14	71.91	72.06	71.54	71.94	71.84	72.71	71.46	71.89	72.37	**72.05**	**+0.11**
	WinoGr.	69.46	66.75	68.15	66.34	66.30	66.73	66.85	66.34	67.40	66.69	67.05	67.32	**66.96**	**+0.11**
	OBQA	47.80	45.60	46.40	46.20	43.90	45.90	45.60	45.20	47.20	45.60	47.50	46.20	**46.34**	**+0.74**
	BoolQ	84.46	84.70	84.08	84.08	84.74	83.53	**84.23**	82.63	82.68	83.79	83.91	83.35	83.27	-0.96
	RTE	74.73	74.61	75.63	71.30	74.91	74.37	**74.16**	72.38	70.76	71.84	73.10	72.92	72.20	-1.96
	ANLI	58.40	57.47	52.85	53.55	54.40	58.25	**55.30**	53.45	51.90	55.20	56.25	55.05	54.37	-0.93
	ARC-C	51.71	50.14	50.38	50.21	50.77	50.64	50.43	50.21	50.34	50.17	49.66	51.96	**50.47**	**+0.04**
	ARC-E	73.86	73.55	73.76	72.62	73.30	73.88	73.42	72.73	74.07	74.33	72.90	75.48	**73.90**	**+0.48**
	MMLU-K	62.28	57.46	60.45	57.48	58.45	58.35	**58.44**	54.42	58.64	57.40	56.96	58.12	57.11	-1.33
	Mean	63.36	61.58	61.52	60.50	60.42	61.05	**61.02** ±0.67	60.38	61.51	60.26	61.11	61.19	60.89 ±0.63	-0.13
	Runtime		5400s	36s	3240s	2160s	1380s	2443s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**228×**
gemma-2-9b-it	MMLU-M	21.48	24.44	23.01	22.02	21.24	21.85	22.44	27.40	24.44	26.29	28.79	24.44	**26.27**	**+3.83**
	GSM8k	75.44	74.12	74.30	74.41	74.18	73.81	**74.16**	73.67	73.92	73.39	73.81	73.54	73.66	-0.50
	HellaSwag	67.24	67.43	67.56	68.23	67.43	67.48	67.63	68.02	67.61	67.81	67.36	67.37	67.63	+0.00
	WinoGr.	70.48	69.93	70.01	69.69	69.34	69.61	69.72	69.69	70.60	70.38	70.01	69.02	**69.92**	**+0.20**
	OBQA	45.40	45.47	44.50	44.20	44.10	45.40	44.73	45.07	44.90	45.20	45.00	45.00	**45.03**	**+0.30**
	BoolQ	88.59	88.66	88.62	88.32	88.50	88.76	88.57	88.64	88.62	88.58	89.07	88.49	**88.68**	**+0.11**
	RTE	78.34	78.10	76.53	78.34	77.26	77.62	77.57	77.26	77.80	77.80	78.16	78.52	**77.91**	**+0.34**
	ANLI	72.80	71.07	71.75	71.80	71.90	72.00	71.70	72.20	72.90	70.30	72.55	71.95	**71.98**	**+0.28**
	ARC-C	51.79	52.28	50.77	51.79	51.32	51.41	51.51	52.22	52.09	52.43	51.45	52.52	**52.14**	**+0.63**
	ARC-E	66.79	67.31	66.50	66.73	66.22	67.66	66.88	68.83	67.82	68.81	66.41	68.22	**68.02**	**+1.14**
	MMLU-K	33.83	33.23	26.74	29.17	29.70	37.46	31.26	36.76	30.64	34.03	28.79	41.04	**34.25**	**+2.99**
	Mean	61.11	61.09	60.03	60.43	60.11	61.19	60.56 ±0.86	61.80	61.03	61.37	61.04	61.83	**61.41** ±0.85	**+0.85**
	Runtime		6231s	149s	3400s	2671s	1500s	2790s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**260×**

Table 3: Comparison of COLA vs. **ZipCal** calibration data under GPTQ W4A16 quantization.

ing languages falls short. We recall that **ZipCal** was developed with property **G3** in mind. We now test the Multi-Domain approach of **ZipCal** under this more difficult setup, to see if the models compressed with aggregated Zipfian subsets can achieve both intra-task sub-optimality and cross-lingual sub-optimality. The results, reported in Table A9, highlight that this aggregation across different languages and tasks yields compressed models that, on average, outperform their single-task, single-language counterparts. While our previous evaluations on strictly English tasks demonstrated relatively stable and uniform improvements, the multilingual, multi-domain setting exhibits a noticeably higher skewness in the performance deltas. **ZipCal** yields a positive mean Δ in 6 out of 8 compression scenarios and proves itself more robust than relying on naive language-matching.

Evaluation on instruction-tuned benchmarks To assess generation quality beyond likelihood-based benchmarks, we evaluate on AlpacaEval (Li et al., 2023), reporting both win rate and the length-controlled (LC) win rate (Dubois et al., 2024), which also accounts for the length of the outputs. Due to computational cost limitations, we use Llama-3.1-8B-Instruct itself as a local judge rather than GPT-4, which introduces a self-evaluation bias: the judge may systematically favor generations that resemble its own output distribution, independent of true quality. Table 5 shows that **ZipCal** matches or slightly exceeds COLA across all four compression methods (Δ ranging from -0.1 to +1.6 points), these values indicate comparable generation quality. Because the same judge and prompt set are applied identically to both COLA and **ZipCal** calibrated models, any self-evaluation bias is shared across conditions and should not favor one calibration method over the other.

Evaluating Scalability We evaluated the efficiency of **ZipCal** against COLA across various dataset scales, to assess its scalability **G1**. While already in Tables 1 and 3 the runtime superiority of **ZipCal** is clear, we also tried gradually subsampling two datasets, namely ARC-C and WinoGrande, to better understand the computational complexity trade-off between **ZipCal** and COLA. Results are reported in Figure 2. On a small dataset like ARC-C, COLA needs from a few seconds with a small 3B model to some minutes using a large 70B model, with the forward pass becoming a bottleneck even on a dataset of this size. We observe a similar trend with WinoGrande, which is an order of magnitude larger than ARC-C. COLA with a 70B model requires over 3 hours, while **ZipCal** takes just 9 seconds. Overall, **ZipCal** is 104× faster over the 3B model, and achieves up to 1330×

Model	Task	Dense	Wanda (25%)			2SSP (25%)			GPTQ (W4A16)			AWQ (W4A16)		
			COLA	ZipCal	Δ	COLA	ZipCal	Δ	COLA	ZipCal	Δ	COLA	ZipCal	Δ
Llama-3.1-8B-Instruct	MMLU-M	24.44	23.19	**24.00**	+0.81	22.22	**22.54**	+0.32	**23.19**	23.01	-0.18	24.44	**24.44**	0.00
	GSM8k	78.09	**75.82**	75.80	-0.02	**4.88**	4.13	-0.75	67.61	**74.07**	+6.46	68.98	**70.50**	+1.52
	HellaSwag	71.71	**71.88**	71.86	-0.02	**59.41**	59.16	-0.25	**71.94**	71.73	-0.21	71.68	**71.95**	+0.27
	WinoGr.	69.46	68.30	**68.82**	+0.52	**60.03**	59.68	-0.35	66.85	**67.95**	+1.10	66.75	**68.10**	+1.35
	OBQA	47.80	46.79	**47.90**	+1.11	40.25	**40.92**	+0.67	**45.60**	44.40	-1.20	45.90	45.90	0.00
	BoolQ	84.46	84.78	**85.23**	+0.45	72.57	**73.12**	+0.55	84.23	**85.38**	+1.15	84.07	**84.60**	+0.53
	RTE	74.73	72.25	**72.82**	+0.57	67.04	**68.02**	+0.98	74.16	**74.72**	+0.56	73.10	**74.50**	+1.40
	ANLI	58.40	57.92	**61.50**	+3.58	**44.50**	43.87	-0.63	55.30	**56.90**	+1.60	53.56	**55.00**	+1.44
	ARC-C	51.71	50.82	**51.11**	+0.29	39.71	**39.75**	+0.04	**50.43**	49.31	-1.12	50.21	**50.30**	+0.09
	ARC-E	73.86	73.36	**73.86**	+0.50	**63.21**	62.47	-0.74	**73.42**	72.97	-0.45	72.55	**72.80**	+0.25
	MMLU-K	62.28	62.02	**76.37**	+14.35	35.59	**36.07**	+0.48	**58.44**	55.86	-2.58	58.52	**58.60**	+0.08
	Mean	63.36	62.47	**64.48**	**+2.01**	46.31	**46.34**	**+0.03**	61.02	**61.48**	**+0.46**	60.89	**61.52**	**+0.63**
gemma-2-9b-it	MMLU-M	21.48	22.51	**27.40**	+4.89	19.26	**23.33**	+4.07	22.44	**26.29**	+3.85	**21.40**	21.30	-0.10
	GSM8k	75.44	74.31	**74.75**	+0.44	3.30	**3.51**	+0.21	**74.16**	73.61	-0.55	**74.22**	74.10	-0.12
	HellaSwag	67.24	67.14	**67.25**	+0.11	**55.48**	54.73	-0.75	**67.63**	67.55	-0.08	**67.58**	67.50	-0.08
	WinoGr.	70.48	**69.21**	68.98	-0.23	**59.06**	58.66	-0.40	**69.72**	69.61	-0.11	**69.58**	69.40	-0.18
	OBQA	45.40	**46.11**	45.60	-0.51	38.14	**38.64**	+0.50	44.73	**45.40**	+0.67	45.17	**46.10**	+0.93
	BoolQ	88.59	88.73	**89.08**	+0.35	72.78	**73.07**	+0.29	**88.57**	88.04	-0.53	88.57	**89.00**	+0.43
	RTE	78.34	78.39	**79.06**	+0.67	**66.00**	65.64	-0.36	77.57	**79.06**	+1.49	77.58	**78.50**	+0.92
	ANLI	72.80	**72.65**	72.60	-0.05	51.90	**52.28**	+0.38	71.70	**72.90**	+1.20	71.76	**72.10**	+0.34
	ARC-C	51.79	51.73	**52.21**	+0.48	39.80	**40.62**	+0.82	**51.51**	51.11	-0.40	**51.28**	51.20	-0.08
	ARC-E	66.79	67.21	**67.84**	+0.63	**59.22**	58.43	-0.79	66.88	**68.22**	+1.34	66.66	**66.80**	+0.14
	MMLU-K	33.83	34.62	**36.96**	+2.34	**22.77**	22.41	-0.36	31.26	**35.19**	+3.93	32.00	**34.00**	+2.00
	Mean	61.11	61.15	**61.98**	**+0.83**	44.34	**44.67**	**+0.33**	60.56	**61.54**	**+0.98**	60.53	**60.91**	**+0.38**

Table 4: Comparison of COLA vs. **ZipCal** (Multi-Domain) performance across Wanda and 2SSP at 25% sparsity, and GPTQ and AWQ using W4A16 compression scheme.

Model	Method	COLA	ZipCal-SD	ZipCal-MD	Δ
Llama-3.1-8B	Dense	50.0/28.9			
	Wanda (25%)	41.3/27.4	40.8/27.1	41.3/27.4	0.0
	2SSP (25%)	16.7/12.2	10.0/7.2	18.3/12.5	+1.6
	GPTQ (W4A16)	48.8/32.2	48.6/32.0	49.1/32.6	+0.3
	AWQ (W4A16)	49.1/30.6	48.8/32.4	49.0/30.4	-0.1

Table 5: AlpacaEval win rate / length-controlled win rate (%) over 805 instructions, judged locally with Llama-3.1-8B-Instruct, comparing COLA and **ZipCal** (SD/MD) calibration across compression methods on Llama-3.1-8B-Instruct.

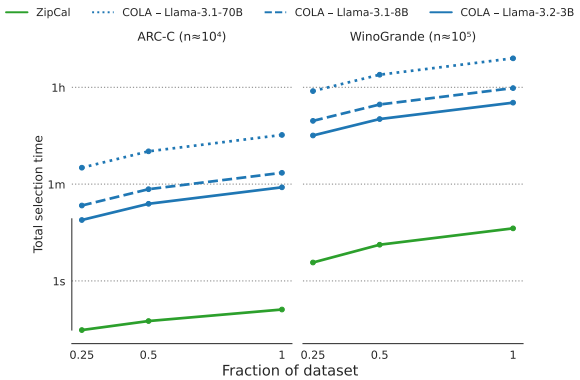


Figure 2: Running time (log-scale) for calibration data selection. The COLA baseline is run for models of different sizes; whereas **ZipCal** is model-agnostic, thus we report the measurement of the only run.

speedup over the 70B model.

Remarks To conclude, the results highlight that while model-based curation becomes computationally prohibitive as LLMs scale in size, our model-agnostic approach **G1** is a near-zero-overhead solution **G2** that maintains high-fidelity calibration without requiring inference. Crucially, the efficiency gain is model-agnostic; once a calibration set is computed, it is fixed and reusable across different models and compression techniques. Moreover, the Multi-Domain version of **ZipCal** overcomes the necessity of a *lucky* pick from a plethora of datasets and samples. The proposed sampling approach provides a compressed model that, on average, outperforms models compressed separately on single domains and single languages **G3**.

5 Further Insights

Ablation on Length-Controlled Samples **ZipCal**'s selected samples are longer on average than those chosen by COLA. For example, on HellaSwag, the median length of the samples picked by **ZipCal** is 30 tokens, compared to 23 for those picked by COLA. This difference is exacerbated on more complex datasets, such as BoolQ, where **ZipCal**'s samples have a median length of 171 tokens, compared to 76 for COLA's samples. One could think that length could be driving the quality of the results rather than lexical representation. To isolate the effect of linguistic diversity from the effect of length, we truncate

Model	Method	COLA	ZipCal	Δ
Llama-3.1-8B	Wanda (25%)	62.96 ± 0.15	60.06 ± 0.17	-2.90
	2SSP (25%)	40.39 ± 0.12	42.86 ± 0.12	2.47
	GPTQ (W4A16)	58.40 ± 0.15	57.97 ± 0.15	-0.43
	AWQ (W4A16)	58.01 ± 0.15	60.57 ± 0.15	2.56

Table 6: Comparison of COLA vs. **ZipCal** across compression methods on Llama-3.1-8B-Instruct using samples truncated at $l = 64$ tokens.

every sample in the source corpus C to $l = 64$ tokens, creating C'. We run both calibration methods over C' to select the calibration samples S, then compression is performed as always. Table 6 reports the results, **ZipCal** retains an advantage on 2SSP and AWQ even under this strict length control, and remains competitive for the other compression techniques.

Mechanistic Analysis To understand why lexical diversity correlates with more robust compressed models, we recall that **ZipCal** represents a larger vocabulary space (Figure 1). This results in a more accurate representation of the original dataset's long tail, increasing entropy in the activations and gathering richer statistics for otherwise dormant parts of the network that would be compressed. This comes with a measurable trade-off: decomposing distributional fidelity into head- and tail-frequency regions, **ZipCal** substantially improves tail-region KL divergence at the cost of increased head-region divergence, confirming that its benefit is concentrated in restoring exposure to the long tail rather than uniformly matching the frequency profile. We define the *overlap* between two models as the ratio between the number of weights compressed in both models and the total number of weights. The overlap between random and **ZipCal**-compressed models is around 90%, compared to 96% between random and COLA, with overlap dropping further to 85–94% in the attention output projection (`o_proj`) and the feed-forward down-projection (`down_proj`), the layers where each sub-block's output rejoins the stream, evidence that **ZipCal** rescues weight pathways that random and COLA calibration leave underactivated. Beyond raw rare-token counts, which we find only weakly predictive, *calibration entropy* is the dominant predictor of post-compression resilience, correlating strongly with performance retention ($\rho = 0.94$, $p \leq 0.005$). For an extended analysis, see Section D.

Effects of Compression on Perplexity Along with the downstream evaluation over Instruct models, we also evaluate **ZipCal** against COLA over Language Modeling tasks (i.e., perplexity). The results reported in Table 7 show that Single-Domain and Multi-Domain **ZipCal** performances are once again on par with COLA. We want to highlight that the calibration sets extracted by **ZipCal** are shared across different models without adaptation, confirming that lexically diverse calibration sets preserve compression quality.

Model	Method	COLA	ZipCal-SD	ZipCal-MD	Δ
Llama-3.1-8B	Dense	7.15	7.15	7.15	
	Wanda (25%)	7.77	7.76	**7.67**	-0.10
	2SSP (25%)	16.32	**15.81**	18.52	-0.51
	GPTQ (W4A16)	**8.09**	**8.09**	8.67	+0.00
	AWQ (W4A16)	**7.35**	7.40	7.40	+0.05
gemma-2-9B	Dense	7.93	7.93	7.93	
	Wanda (25%)	**8.13**	8.22	**8.13**	0.00
	2SSP (25%)	12.74	**12.41**	14.18	-0.33
	GPTQ (W4A16)	**8.27**	**8.27**	8.67	0.00
	AWQ (W4A16)	15.84	15.75	**15.73**	-0.11

Table 7: Language modeling perplexity (↓). Comparison between COLA, **ZipCal**, and **ZipCal** (Multi-Domain). Δ indicates the difference between our best and COLA on Avg. over C4, WikiText, and Pile.

6 Conclusion

In this paper, we highlighted the limitations of current calibration data selection strategies for compression algorithms, which are either based on pure random sampling or use expensive, model- and task- dependent calibration strategies, limiting performance across multiple models and tasks. To address these issues, we proposed **ZipCal**, a model-agnostic, computationally cheap data curation strategy for both pruning and quantization approaches, which selects calibration data by following a linguistics-principled Zipfian distribution. Results show that the proposed method performs better than random on majority of the evaluation benchmarks, and more importantly, on par with state-of-the-art data curation approaches while requiring minimal overhead. Furthermore, we also introduced a multi-domain version of **ZipCal**, which applies Zipf sampling hierarchically across different calibration datasets. Results show that the resulting unified multi-domain calibration dataset allows for outperforming single-domain calibration, proving a solution to the aforementioned problem of inter-task sub-optimality.

Limitations

While we provide extended experimental insights on choosing calibration data based on linguistic diversity, we acknowledge the following limitations to the current study. We mainly relied on English calibration data and evaluation tasks for the main paper, with an accompanying extension to Spanish and Chinese. Although Zipf's Law is a cross-linguistic phenomenon (Piantadosi, 2014), the specific "sanitization" process (e.g., lowercase conversion and subword marker stripping) may require tuning for morphologically rich languages or non-alphabetic scripts. Our experiments focus on three specific families of LLMs (Llama-3.1-8B, Gemma-2-9B, and Qwen3-8B) and four compression methods. It remains to be seen how lexical diversity as a curation proxy behaves for Mixture-of-Experts (MoE) models, where activation routing might necessitate a different balance of data to ensure all "experts" are adequately calibrated. Similarly, extending this approach to Multimodal Large Language Models (MLLMs) would require a multimodal analogue of lexical diversity that captures the distributional properties of non-textual tokens, such as visual, audio, or video modality.

Ethics Statement

This work focuses on efficient data curation for model compression. All models, datasets, and benchmarks used in this research are publicly available with appropriate licenses. We credit original authors throughout the manuscript and acknowledge their contributions. We acknowledge that data selection processes can inadvertently amplify existing biases in the source data, and we have reported detailed performance statistics across our benchmarks. We also recognize that efficient techniques for deploying LLMs accelerate AI adoption, potentially leading to misuse. However, our focus remains on the foundational technical aspects of data curation rather than specific downstream applications of AI.

Acknowledgments

This work was supported in part by the UE GA n. 101167870 "ARCHitectures based on unconventional accelerators for dependable/energY efficienT AI Systems" - ARCHYTAS. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

References

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2025. Systematic Outliers in Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. SliceGPT: Compress Large Language Models by Deleting Rows and Columns. In *The Twelfth International Conference on Learning Representations*.

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

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. 2025. EfficientQAT: Efficient Quantization-Aware Training for Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10081–10100, Vienna, Austria. 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 [cs].

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint*. ArXiv:2110.14168 [cs].

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Elia Cunegatti, Leonardo Lucio Custode, and Giovanni Iacca. 2025. Zeroth-Order Adaptive Neuron Alignment Based Pruning without Re-Training. *Transactions on Machine Learning Research*.

Mohsen Dehghankar, Mahdi Erfanian, and Abolfazl Asudeh. 2025. An Efficient Matrix Multiplication Algorithm for Accelerating Inference in Binary and Ternary Neural Networks. *arXiv preprint*. ArXiv:2411.06360 [cs].

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. *arXiv preprint arXiv:2404.04475*.

Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive Language Models Can be Accurately Pruned in One-Shot. In *Proceedings of the 40th International Conference on Machine Learning*, pages 10323–10337. PMLR. ISSN: 2640-3498.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. *arXiv preprint*. ArXiv:2210.17323 [cs].

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint*. ArXiv:2101.00027 [cs].

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. 2023. A framework for few-shot language model evaluation.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 181 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ADS Bibcode: 2024arXiv240721783G.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. 2025. SlimLLM: Accurate Structured Pruning for Large Language Models. *arXiv preprint*. ArXiv:2505.22689 [cs].

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Bowei He, Lihao Yin, Huiling Zhen, Shuqi Liu, Han Wu, Xiaokun Zhang, Mingxuan Yuan, and Chen Ma. 2025. Preserving LLM Capabilities through Calibration Data Curation: From Analysis to Optimization. *arXiv preprint*. ArXiv:2510.10618 [cs].

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv preprint*. ArXiv:2103.03874 [cs].

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. 2025. SliM-LLM: Salience-Driven Mixed-Precision Quantization for Large Language Models. In *International Conference on Machine Learning*, pages 25672–25692. PMLR.

Yixin Ji, Yang Xiang, Juntao Li, Qingrong Xia, Ping Li, Xinyu Duan, Zhefeng Wang, and Min Zhang. 2025. Beware of calibration data for pruning large language models. In *International Conference on Learning Representations*, volume 2025, pages 18604–18620.

Sakaguchi Keisuke, BrasRonan Le, BhagavatulaChandra, and ChoiYejin. 2021. WinoGrande. *Communications of the ACM*. Publisher: ACMPUB27New York, NY, USA.

Banseok Lee, Dongkyu Kim, Youngcheon you, and Young-Min Kim. 2025. LittleBit: Ultra Low-Bit Quantization via Latent Factorization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models. `https://github.com/tatsu-lab/alpaca_eval`.

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. *Proceedings of Machine Learning and Systems*, 6:87–100.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. 2024. The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits. *arXiv preprint*. ArXiv:2402.17764 [cs].

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When Less is More: Investigating Data Pruning for Pretraining LLMs at Scale. *arXiv preprint*. ArXiv:2309.04564 [cs].

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer Sentinel Mixture Models. *arXiv preprint*. ArXiv:1609.07843 [cs].

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Bhuminand Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact Language Models via Pruning and Knowledge Distillation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or Down? Adaptive Rounding for Post-Training Quantization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7197–7206. PMLR.

Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M. Bronstein, and Avi Mendelson. 2021. Loss aware post-training quantization. *Machine Learning*, 110(11):3245–3262.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Jaehoon Oh and Dokwan Oh. 2025. Beyond Fixed-Length Calibration for Post-Training Compression of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19355–19366, Suzhou, China. Association for Computational Linguistics.

Gunho Park, Baeseong Park, Minsub Kim, Sungjae Lee, Jeonghoon Kim, Beomseok Kwon, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. 2024. LUT-GEMM: Quantized Matrix Multiplication based on LUTs for Efficient Inference in Large-Scale Generative Language Models. *arXiv preprint*. ArXiv:2206.09557 [cs].

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP Models really able to Solve Simple Math Word Problems? *arXiv preprint*. ArXiv:2103.07191 [cs].

Steven T. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A Multilingual Dataset for Causal Commonsense Reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Fabrizio Sandri, Elia Cunegatti, and Giovanni Iacca. 2025. 2SSP: A Two-Stage Framework for Structured Pruning of LLMs. *Transactions on Machine Learning Research*.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, and 4 others. 2025. Global MMLU: Understanding and Addressing Cultural and Linguistic Biases in Multilingual Evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Lu Sun and Jun Sakuma. 2026. Learning Semi-Structured Sparsity for LLMs via Shared and Context-Aware Hypernetwork. In *The Fourteenth International Conference on Learning Representations*.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024a. Massive Activations in Large Language Models. In *First Conference on Language Modeling*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024b. A Simple and Effective Pruning Approach for Large Language Models. *arXiv preprint*. ArXiv:2306.11695 [cs].

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *arXiv preprint*. ArXiv:2408.00118 [cs].

Alexey Tikhonov and Max Ryabinin. 2021. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3534–3546, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. 2023. BitNet: Scaling 1-bit Transformers for Large Language Models. *arXiv preprint*. ArXiv:2310.11453 [cs].

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 5776–5788, Red Hook, NY, USA. Curran Associates Inc.

Miles Williams and Nikolaos Aletras. 2024. On the Impact of Calibration Data in Post-training Quantization and Pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118, Bangkok, Thailand. Association for Computational Linguistics.

Miles Williams, George Chrysostomou, Vitor Amancio Jeronymo, and Nikolaos Aletras. 2026. Compressing Language Models for Specialized Domains. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7393–7415, Rabat, Morocco. Association for Computational Linguistics.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024. Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning. In *The Twelfth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 Technical Report. *arXiv preprint*. ArXiv:2505.09388 [cs].

Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M. Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, Nathan Susanj, Zheng Zhang, Jack FitzGerald, and Abhishek Kumar. 2025b. Wanda++: Pruning Large Language Models via Regional Gradients. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4321–4333, Vienna, Austria. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? *arXiv preprint*. ArXiv:1905.07830 [cs].

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

Yuxin Zhang, Lirui Zhao, Mingbao Lin, Sun Yunyun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. 2024b. Dynamic Sparse No Training: Training-Free Fine-tuning for Sparse LLMs. In *The Twelfth International Conference on Learning Representations*.

Aojun Zhou, Yukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. 2021. Learning N:M Fine-grained Structured Sparse Neural Networks From Scratch. In *International Conference on Learning Representations*.

G.K. Zipf. 1932. *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press.

Appendix Table of Contents

A Detailed Algorithms and Proofs 14

B Complementary Results to the Main Matter 14

C Assessing Zipf Coverage 15

D Extended Mechanistic Analysis 16

E Comparison with a Generative Data Curation Baseline 18

A Detailed Algorithms and Proofs

Multi-Domain Sampling We present here the pseudocode for the Multi-Domain selection procedure, Algorithm 2, and prove Theorem 3.2, the time complexity of **ZipCal** in this setting. For the lightweight embedding used for all our experiments, we relied on `all-MiniLM-L6-v2` (Wang et al., 2020).

Proof. Given m datasets $D_i : i \in (1, m)$. In the first stage, Single-Domain **ZipCal** is executed for each dataset. Let n_i be the size of dataset $\mathcal{D}_i$. From Theorem 3.1, the complexity for each source is $O(n_i k)$. An upper bound to this operation is mkN, where $N = \max_{D_i}(n_i)$ The second stage involves the k-centers greedy selection on a pool of size mk. At each of the k iterations, we calculate the distance between the remaining candidates and the current set S. The cost of this stage is $O(k \cdot (mk) \cdot d)$, where d is the dimensionality of the sample representation. Since m and k are typically $<<$ N, the term mkd is negligible. Thus, the overall complexity is dominated by the initial **ZipCal** pass, yielding $O(mkN)$. □

B Complementary Results to the Main Matter

We report here additional tables that were omitted from the main paper due to length constraints. In particular, Tables A1 and A2 show the comparison between COLA and Single-Domain **ZipCal** under AWQ and 2SSP compression, respectively. Standard deviations on all tables are derived from 3 seeded runs.

Algorithm 2: Multi-Domain **ZipCal**

Input: Collection of datasets $\mathbb{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$; Final budget k

Output: Multi-Domain calibration set S

$\mathcal{P} \leftarrow \emptyset$;
foreach $\mathcal{D}_i \in \mathbb{D}$ **do**
  `// Step 1: Single-Domain ZipCal`
  $P_i \leftarrow$ **ZipCal**$(\mathcal{D}_i, k)$;
  $\mathcal{P} \leftarrow \mathcal{P} \cup P_i$;
`// Step 2: Global K-Centers Greedy Selection`
$\forall s \in P$ create a lightweight embedding;
$s_1 \leftarrow$ select random $s \in \mathcal{P}$;
$S \leftarrow \{s_1\}$;
for $j = 2$ **to** k **do**
  $s^* \leftarrow \arg\max_{s \in \mathcal{P} \setminus S} \left(\min_{z \in S} \text{dist}(s, z)\right)$;
  $S \leftarrow S \cup \{s^*\}$;
return S ;

Expanding the Model-Agnostic Claim We extend our single-domain evaluation to Qwen3 (Yang et al., 2025a), an architecturally distinct third family (QK-norm, unlike Llama's pre-norm-only (Grattafiori et al., 2024) or Gemma 2's pre+post-norm design (Team et al., 2024)). Results in Table A10 confirm that our main claims hold: **ZipCal** matches or exceeds COLA across compression methods, despite using calibration data selected in a fraction of the time.

Effects of Context Length We investigate the effect of calibration sequence length by testing across a set of values $w \in [16, 2048]$ and setting the number of samples $k = 128$. As shown in Figure A1, we find that for most compression techniques, no significant improvement is observed by increasing the context window. Performance is flat even at $w = 16$, which suggests that lexically diverse calibration sets work even at small scales. This trend seems to be characterized by the compression technique itself rather than the length of the calibration set. For instance, 2SSP shows extreme sensitivity and strictly requires the longest context length to recover performance. The reason is that 2SSP removes entire attention submodules, which requires capturing long-range relationships; at small context length, this information is removed and leads to biased pruning. Our results are in line with the comprehensive analysis of (Oh and Oh, 2025).

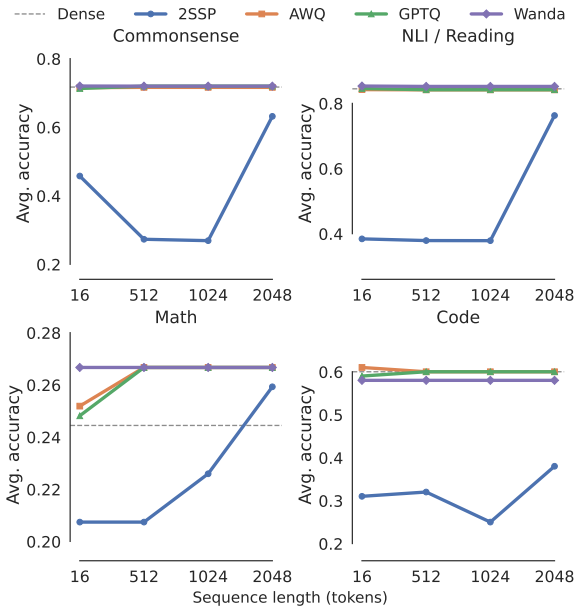


Figure A1: Effect of calibration data context length on model capabilities across compression techniques for `LLaMA-3.1-8B-Instruct`.

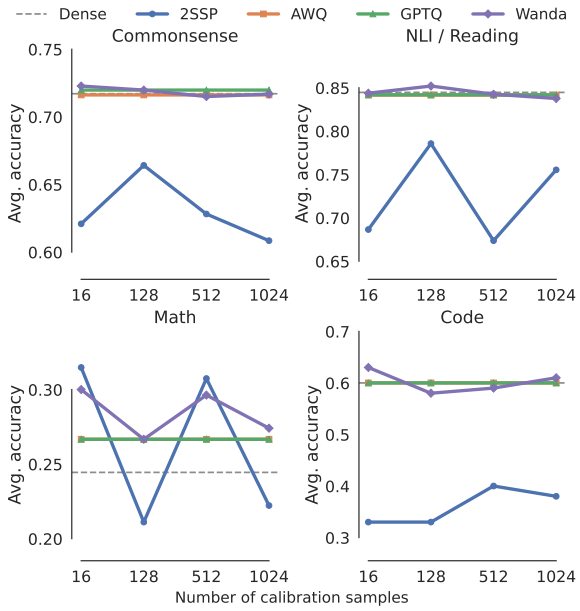


Figure A2: Effect of the number of calibration data samples on model capabilities across compression techniques for `LLaMA-3.1-8B-Instruct`.

Effects of Calibration Sample Size To further highlight the efficiency of **ZipCal**, we analyze the impact of the number of samples in the calibration set by testing $k \in [16, 2048]$ with context length $w = 2048$. The results in Figure A2 show great stability among capabilities and the number of samples. Once again, 2SSP shows greater sensitivity to sample count. Surprisingly, Code and Math capabilities actually benefit from a smaller number of samples (38% vs. 31% using 16 vs. 1024 samples, respectively, for Math capabilities after 2SSP), finding a negligible decrease. Compared to baselines such as COLA (He et al., 2025), which exhibit some capability shifts at larger sample sizes, **ZipCal** provides a much more stable and predictable function.

Scalability to Larger Language Models To investigate if **ZipCal** data curation is effective on larger models, we repeat the set of experiments about Single-Domain and Multi-Domain data curation on Llama-3.1-70B-Instruct. For Single-Domain data curation, we can see in Table A11 that the patterns seen on smaller models hold even in this case, with performances that are on par with COLA but require much less time to extract the curation set of samples **G1**, increasing the $188\times$ speedup of the 8B model to $404\times$. On Multi-Domain curation, the advantages of **ZipCal** are even more pronounced, Table A12. For the retention of mathematical capabilities, **ZipCal** is able to protect the information flow with performance that is very close to the dense model. On the other hand, COLA completely fumbles these tasks, leading to a net delta on the overall quality of the results that averages $+3.38$.

C Assessing Zipf Coverage

As illustrated in Figure 1, **ZipCal** achieves a vocabulary coverage that better reflects the original dataset. When restricted to a standard budget of 128 samples, **ZipCal** identifies by construction the subset that yields the longest possible Zipfian tail. On the other hand, random sampling would need a significantly higher number of samples in order to cover the same vocabulary space. COLA's selection criterion leads to a similar distribution to random sampling. This advantage in coverage is especially noticeable in lexically rich corpora like WinoGrande and HellaSwag, where the *gap* between **ZipCal** and competing baselines reaches up to half an order of magnitude. At extremely tiny samples (i.e., 16), Figure A3, the difference is even more pronounced, with **ZipCal** being the only one capable of covering more than half of the original vocabulary. For large amounts of samples, Figure A4, **ZipCal** actually covers the whole vocabulary space compared to the other baselines. A qualitative analysis on the selected samples reveals that

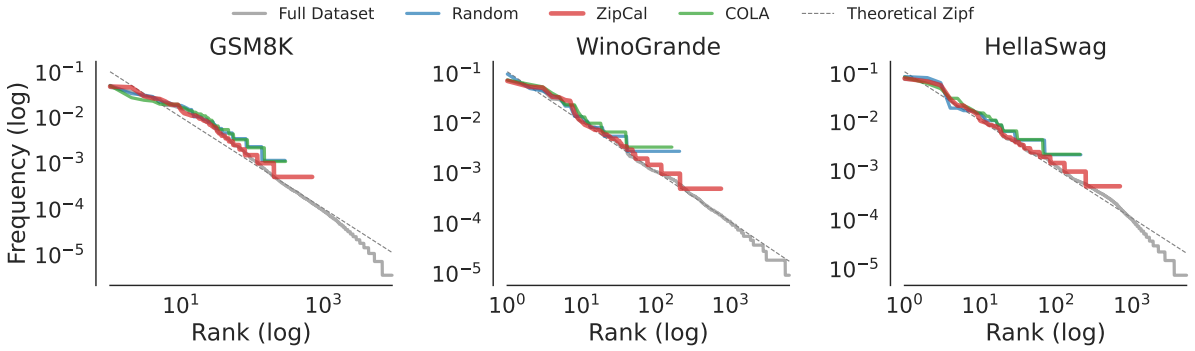


Figure A3: Token frequency distribution of the original datasets and the random, COLA, and **ZipCal** sampling calibration sets using 16 samples.

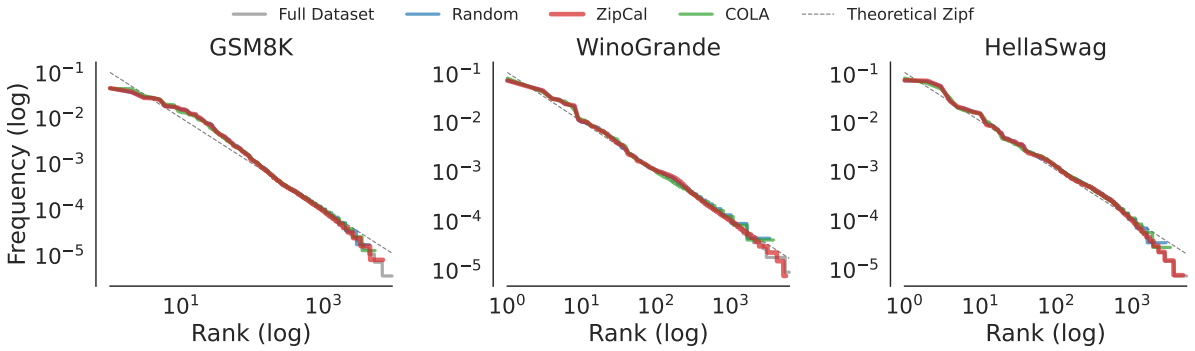


Figure A4: Token frequency distribution of the original datasets and the random, COLA, and **ZipCal** sampling calibration sets using 1024 samples.

Calibration Samples C.1: HellaSwag

ZipCal *Median Length: 30.0, Unique Words: 1455*

- A man has shaving cream on his face and reads a label on a pressurized can and discusses it. a man Shaving
- Afterwards, he does a light sanding of the table. With a tack rug, he removes the dust. Wearing gloves, he applies a stain to the table. finally Polishing forniture
- Two men walked over the football table, one man counted the dice on the side. The players twist and push and pull the poles as they play on the table. the camera men Table soccer

COLA *Median Length: 23.0, Unique Words: 927*

- There's a man in a gym using an elliptical machine. he Elliptical trainer
- There's a man in a gym using an elliptical machine. He is wearing a pair of black track pants, black shirt and a cap. he Elliptical trainer
- This girl is doing a video of how to make your white shoes become sparkly clean. you Cleaning shoes

ZipCal selects samples with significantly higher median lengths and Type-Token Ratios (TTR). This advantage is especially noticeable in lexically rich corpora like WinoGrande and HellaSwag, where the gap between **ZipCal** and competing baselines reaches half an order of magnitude, capturing the encyclopedic context that others ignore. We provide some examples in Boxes C.1 and C.2.

D Extended Mechanistic Analysis

To mechanistically understand why maximizing lexical diversity yields a more robust compressed model, we must examine the phenomenon of small-sample vocabulary collapse. In natural language, tokens follow a strict Zipfian distribution. When extracting a purely random, small calibration set (e.g., 128 samples), the long-tail parameters are probabilistically excluded. This leads to a calibration subset heavily biased towards highly frequent stop-words. Consequently, compression models evaluate weight importance on a flattened latent space, pruning the sub-networks tied to rare sequences.

Our proposed method, **ZipCal**, intentionally mitigates this collapse by selecting the sample set to mimic the macroscopic Zipfian structure of the en-

Calibration Samples C.2: BoolQ

ZipCal *Median Length: 171.5, Unique Words: 8451*

- The black turtle bean is a small, shiny variety of the common bean (Phaseolus vulgaris), especially popular in Latin American cuisine, though it can also be found in Cajun and Creole cuisines of south Louisiana. Like most common beans, they are native to the Americas, but have been introduced around the world. They are also used in East Indian cooking, Punjabi cuisine and are referred to as black beans and in Maharshtrian cuisine known as "Kala Ghevada". They are used interchangeably with vigna mungo in countries such as the US. They are often simply called black beans (frijoles negros, zaragoza, judía negra, poroto negro, caraota o habichuela negra in Spanish, and feijão preto in Portuguese), although this can cause confusion with other black beans. are black beans the same as turtle beans.
- The story begins as a girl named Sally and her brother, who serves as the narrator of the book, sit alone in their house on a cold, rainy day, staring wistfully out the window. Then they hear a loud bump which is quickly followed by the arrival of the Cat in the Hat, a tall anthropomorphic cat in a red and white striped hat and a red bow tie. The Cat proposes to entertain the children with some tricks that he knows. The children's pet fish refuses, insisting that the Cat should leave. The Cat responds by balancing the fish on the tip of his umbrella. The game quickly becomes increasingly trickier, as the Cat balances himself on a ball and tries to balance lots of household items on his limbs until he falls on his head, dropping everything he was holding. The fish admonishes him again, but the Cat in the Hat just proposes another game. does the cat in the hat have a name

COLA *Median Length: 76.0, Unique Words: 4204*

- The Stanley Cup playoffs consists of four rounds of best-of-seven series. Each series is played in a 2–2–1–1–1 format...
- United States Army Basic Training (also known as Initial Entry Training, IET) is the recruit training program of physical and mental preparation...

tire dataset, exposing the compression technique to the long-tail tokens. We validate this mechanistically through three distinct empirical observations on Llama-3.1-8B-Instruct.

- **Distributional Fidelity (KL Divergence)**: We measured the Kullback-Leibler (KL) Divergence between the calibration samples ($k = 128$) and the full datasets, decomposed by frequency region: the head (top 10% most frequent tokens) and the tail (remaining 90%). We measure this for 5 datasets (BoolQ, WinoGrande, ARC-C, GSM8K, HellaSwag). **ZipCal** shows a consistent trade-off: substantially improved tail-region matching (KL(tail) drops by 3.3–61.0%, e.g., from 16.40 to 7.87 on HellaSwag), paired with increased head-region divergence (KL(head) rises by 62.1–168.1% across datasets, e.g., from 1.02 to 2.73 on HellaSwag). This confirms that **ZipCal**'s benefit is concentrated in restoring exposure to the long tail of the distribution, the intended design goal and that this concentration comes at a cost to head-region fidelity. Since **ZipCal** reallocates probability mass toward tail tokens to counteract their underrepresentation, it necessarily samples head-region tokens less exhaustively than random sampling would, which by construction saturates the head almost immediately.
- **Pruning Mask Overlap:** To verify how this distributional shift physically alters the pruning decisions, we computed the overlap between the masks generated by our 128-sample sets and both a small and large-scale mask derived from 128 and 1024 random samples. Because the datasets consist of high-frequency stop-words, both a small random sample (128) and a large random sample (1024) are dominated by the exact same "head" of the distribution. As a result, the 128-sample random mask is virtually identical to the 1024-sample mask (98-99% overlap). This proves that simply increasing the random sample size, or using a random subset, just repeatedly reinforces the same weights tied to highly frequent tokens, ignoring the tail. In contrast, **ZipCal** intentionally diverges from this frequency-biased consensus, with mask overlaps dropping to 85%-94% (especially evident in the `o_proj` and `down_proj` layers). This systematic divergence confirms that **ZipCal** rescues distinct pathways associated with rare concepts that standard sampling would otherwise discard.
- **Calibration Entropy as the Dominant Predictor:** Finally, we correlated various distributional properties of the calibration sets with the post-compression performance drop (Δ). Interestingly, simply counting the inclusion of rare tokens is not significantly predictive. Instead, the absolute **Calibration Entropy** exhibits an overwhelming correlation with post-compression resilience (Pearson's $\rho = 0.94, p \leq 0.005$).

In conclusion, our mechanistic analysis proves that the resilience of the **ZipCal**-compressed model is not just a byproduct of encountering more words, but stems from restoring the entropy of the original corpus in miniature. By forcing a globally representative lexical space during calibration, intermediate layer activations maintain their fidelity across the true latent space, preserving the capabilities of the pruned network.

E Comparison with a Generative Data Curation Baseline

To further validate the quality of the calibration data selected by **ZipCal**, we compare against DSnoT (Ji et al., 2025), a self-generative baseline that combines perplexity-based sample selection with a self-generation phase, in which a portion of tokens are masked from each selected sample and regenerated by the model itself. Since DSnoT's publicly released implementation only supports C4, WikiText, and Pile as calibration sources, we restrict this comparison to those datasets; adapting DSnoT to the full set of calibration datasets considered in our work is beyond the scope of this paper. Results are reported in Table A4.

Model	Task	Dense	COLA Calibration Category					COLA	ZipCal Calibration Category					ZipCal	Δ
			LangMod	Math	CommQA	NLI	KnowTran	Mean	LangMod	Math	CommQA	NLI	KnowTran	Mean	
Llama-3.1-8B-Instruct	MMLU-M	24.44	24.40	24.60	24.30	24.40	24.50	**24.44**	24.50	24.30	24.40	24.20	24.70	24.42	**-0.02**
	GSM8k	78.09	72.35	62.51	69.67	69.60	70.77	**68.98**	71.80	63.10	68.90	70.10	70.50	68.88	**-0.10**
	HellaSwag	71.71	72.20	71.38	71.28	72.10	71.45	**71.68**	72.40	71.10	71.50	71.80	71.60	**71.68**	**0.00**
	WinoGr.	69.46	67.19	67.05	67.52	65.98	66.02	**66.75**	66.90	67.30	67.20	66.50	65.80	66.74	**-0.01**
	OBQA	47.80	46.60	44.90	46.50	45.20	46.30	**45.90**	46.80	44.50	46.10	45.50	46.50	45.88	**-0.02**
	BoolQ	84.46	83.92	84.28	84.20	83.70	84.24	**84.07**	83.50	84.40	83.80	84.10	84.00	83.96	**-0.11**
	RTE	74.73	74.01	72.92	73.47	72.02	73.10	73.10	73.80	73.10	73.00	72.50	73.30	**73.14**	**+0.04**
	ANLI	58.40	56.33	50.95	53.40	53.30	53.80	**53.56**	55.80	51.50	53.10	53.90	53.40	53.54	**-0.02**
	ARC-C	51.71	50.03	51.41	49.62	50.04	49.96	50.21	50.50	50.80	49.90	49.70	50.30	**50.24**	**+0.03**
	ARC-E	73.86	73.36	71.72	72.14	72.69	72.85	**72.55**	72.90	71.90	71.80	72.80	72.50	72.38	**-0.17**
	MMLU-K	62.28	58.52	58.52	58.31	58.11	59.12	**58.52**	58.10	58.80	58.15	58.30	58.90	58.45	**-0.07**
	Mean	63.36	60.81	59.97	60.84	60.72	61.13	**60.89** ±0.98	60.64	60.07	60.67	60.85	61.17	**60.85** ±0.75	**-0.04**
	Runtime		5400s	36s	3240s	2160s	1380s	2443s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**228×**
gemma-2-9b-it	MMLU-M	21.48	21.40	21.40	21.35	21.45	21.40	**21.40**	21.40	21.35	21.30	21.45	21.45	**21.39**	**-0.01**
	GSM8k	75.44	74.17	74.91	74.07	73.96	74.00	**74.22**	74.50	74.50	73.80	74.20	73.70	74.14	**-0.08**
	HellaSwag	67.24	67.25	67.81	68.04	67.39	67.42	**67.58**	66.90	67.90	67.50	67.50	67.20	67.40	**-0.18**
	WinoGr.	70.48	69.53	69.65	69.73	69.34	69.65	**69.58**	69.80	69.20	69.90	69.10	69.40	69.48	**-0.10**
	OBQA	45.40	45.47	44.70	45.20	45.50	45.00	**45.17**	45.10	45.00	44.90	45.80	44.80	45.12	**-0.05**
	BoolQ	88.59	88.53	88.53	88.61	88.58	88.59	**88.57**	88.70	88.40	88.50	88.70	88.40	88.54	**-0.03**
	RTE	78.34	79.06	77.26	76.90	77.26	77.44	**77.58**	78.50	77.50	76.50	77.00	77.60	77.42	**-0.16**
	ANLI	72.80	71.77	72.10	71.90	70.70	72.35	**71.76**	72.00	71.80	72.10	70.50	72.10	71.70	**-0.06**
	ARC-C	51.79	51.11	51.49	51.41	51.58	50.81	**51.28**	50.80	51.70	51.20	51.80	50.50	51.20	**-0.08**
	ARC-E	66.79	66.40	66.35	67.09	66.88	66.58	**66.66**	66.60	66.10	66.80	67.10	66.30	66.58	**-0.08**
	MMLU-K	33.83	35.73	28.90	30.12	31.79	33.46	**32.00**	35.10	29.50	29.80	32.10	33.10	31.92	**-0.08**
	Mean	61.11	60.95	60.26	60.22	60.40	60.61	**60.53** ±0.84	61.03	60.27	60.21	60.48	60.41	**60.44** ±0.86	**-0.09**
	Runtime		6231s	149s	3400s	2671s	1500s	2790s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**260×**

Table A1: Comparison of COLA vs. **ZipCal** calibration data under AWQ W4A16 compression.

Model	Task	Dense	COLA Calibration Category					COLA	ZipCal Calibration Category					ZipCal	Δ
			LangMod	Math	CommQA	NLI	KnowTran	Mean	LangMod	Math	CommQA	NLI	KnowTran	Mean	
Llama-3.1-8B-Instruct	MMLU-M	24.44	22.30	23.40	21.80	21.10	22.50	**22.22**	22.10	22.90	21.50	21.30	22.00	21.96	**-0.26**
	GSM8k	78.09	6.34	13.68	0.00	0.00	4.36	**4.88**	4.98	8.42	0.23	3.87	5.65	4.63	**-0.25**
	HellaSwag	71.71	61.32	60.53	60.66	57.32	57.21	59.41	60.26	61.29	58.73	59.30	59.23	**59.76**	**+0.35**
	WinoGr.	69.46	60.69	58.68	59.04	60.18	61.56	60.03	61.69	59.63	60.62	61.80	61.72	**61.09**	**+1.06**
	OBQA	47.80	41.93	38.10	41.80	40.50	38.90	40.25	40.60	39.90	41.60	40.80	40.70	**40.72**	**+0.47**
	BoolQ	84.46	75.64	71.36	69.36	71.64	74.85	**72.57**	74.01	72.66	67.06	75.46	70.60	71.96	**-0.61**
	RTE	74.73	66.43	65.52	63.00	71.12	69.13	**67.04**	67.99	59.93	64.98	65.34	71.12	65.87	**-1.17**
	ANLI	58.40	47.63	39.75	45.00	45.85	44.25	**44.50**	40.53	35.10	37.30	36.05	39.05	37.61	**-6.89**
	ARC-C	51.71	39.51	39.04	42.15	39.16	38.69	**39.71**	38.88	37.03	39.51	36.22	38.44	38.01	**-1.70**
	ARC-E	73.86	61.91	62.61	65.85	63.97	61.70	**63.21**	63.13	58.92	64.27	57.47	60.71	60.90	**-2.31**
	MMLU-K	62.28	39.87	34.59	33.25	37.68	32.54	**35.59**	38.27	26.98	29.89	29.85	31.30	31.26	**-4.33**
	Mean	63.36	47.60	46.12	45.63	46.23	45.97	**46.31** ±0.39	46.59	43.89	44.15	44.31	45.50	44.89 ±0.40	**-1.42**
	Runtime		5400s	36s	3240s	2160s	1380s	2443s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**228×**
gemma-2-9b-it	MMLU-M	21.48	19.50	20.40	18.90	18.40	19.10	**19.26**	19.10	20.00	18.50	18.70	18.80	19.02	**-0.24**
	GSM8k	75.44	4.50	7.80	1.20	0.50	2.50	**3.30**	3.20	5.40	0.00	2.10	3.10	2.76	**-0.54**
	HellaSwag	67.24	56.40	55.20	57.10	54.80	53.90	**55.48**	54.10	55.60	53.20	55.00	54.80	54.54	**-0.94**
	WinoGr.	70.48	59.50	57.80	58.40	59.10	60.50	59.06	60.20	58.10	59.30	60.50	59.90	**59.60**	**+0.54**
	OBQA	45.40	39.50	36.40	38.20	39.10	37.50	38.14	38.10	37.50	39.20	38.60	38.90	**38.46**	**+0.32**
	BoolQ	88.59	76.20	72.10	69.50	71.80	74.30	**72.78**	74.50	73.20	68.10	75.10	71.20	72.42	**-0.36**
	RTE	78.34	67.50	65.20	62.80	68.40	66.10	66.00	68.10	63.40	65.70	66.80	67.50	66.30	**+0.30**
	ANLI	72.80	54.20	48.50	51.30	53.10	52.40	**51.90**	49.50	45.20	47.80	46.90	48.50	47.58	**-4.32**
	ARC-C	51.79	40.10	39.50	41.20	39.80	38.40	**39.80**	39.20	38.10	40.50	37.60	39.10	38.90	**-0.90**
	ARC-E	66.79	59.20	58.40	60.50	59.80	58.20	**59.22**	60.10	57.50	61.20	56.40	59.30	58.90	**-0.32**
	MMLU-K	33.83	24.12	21.45	22.10	25.30	20.90	**22.77**	23.10	19.80	20.45	21.00	21.50	21.17	**-1.60**
	Mean	65.07	45.52	43.88	43.75	44.55	43.98	**44.34** ±0.56	44.47	43.07	43.10	43.52	43.88	43.61 ±0.58	**-0.73**
	Runtime		6231s	149s	3400s	2671s	1500s	2790s	15.2s	2.3s	12.3s	9.3s	14.5s	**10.7s**	**260×**

Table A2: Comparison of COLA vs. **ZipCal** under 2SSP compression at 25% sparsity.

Model	Method	COLA				ZipCal				ZipCal (Multi-Domain)				Δ Avg
		Wiki	C4	Pile	Avg	Wiki	C4	Pile	Avg	Wiki	C4	Pile	Avg	
Llama-3.1-8B	Dense	7.03	9.71	4.71	7.15	7.03	9.71	4.71	7.15	7.03	9.71	4.71	7.15	
	Wanda (25%)	7.62	10.70	4.98	7.77	7.60	10.71	4.98	7.76	**7.52**	**10.54**	**4.94**	**7.67**	-0.10
	2SSP (25%)	18.53	21.09	9.36	16.32	**17.34**	21.09	**9.00**	**15.81**	19.53	25.80	10.22	18.52	+2.20
	GPTQ (W4A16)	7.71	11.42	5.13	8.09	7.71	11.40	5.15	8.09	8.59	12.08	5.34	8.67	+0.58
	AWQ (W4A16)	**7.21**	**10.04**	**4.79**	**7.35**	7.31	10.09	4.81	7.40	7.32	10.07	4.82	7.40	+0.05
Gemma-2-9B	Dense	7.63	11.53	4.63	7.93	7.63	11.53	4.63	7.93	7.63	11.53	4.63	7.93	
	Wanda (25%)	**7.87**	**11.78**	**4.74**	**8.13**	7.92	11.99	4.76	8.22	**7.86**	11.82	**4.72**	**8.13**	0.00
	2SSP (25%)	12.56	19.20	6.47	12.74	**11.68**	**19.08**	6.47	**12.41**	14.14	21.55	6.86	14.18	+1.44
	GPTQ (W4A16)	7.93	12.05	4.82	8.27	7.93	12.06	4.81	8.27	8.44	12.59	4.99	8.67	+0.40
	AWQ (W4A16)	16.01	21.87	9.64	15.84	16.01	21.54	9.71	15.75	**16.01**	**21.51**	**9.67**	**15.73**	-0.11

Table A3: Language modeling perplexity (↓). Comparison between standard COLA, Single-Domain **ZipCal**, and Multi-Domain **ZipCal**. Δ indicates the difference between **ZipCal** (Multi-Domain) and COLA on Avg.

		Wanda (25%)			GPTQ (W4A16)			AWQ (W4A16)		
Task	Dense	DSnoT	ZipCal	Δ	DSnoT	ZipCal	Δ	DSnoT	ZipCal	Δ
MMLU-M	24.44	24.04	**24.14**	+0.10	23.33	**24.07**	+0.74	23.12	**24.07**	+0.95
GSM8K	78.08	76.15	**77.17**	+1.02	70.12	**70.63**	+0.51	36.13	**38.46**	+2.33
HellaSwag	71.70	63.12	**64.28**	+1.16	**64.23**	64.20	-0.03	64.20	**64.85**	+0.65
WinoGr.	73.56	67.20	**68.07**	+0.87	**67.39**	**67.39**	+0.00	66.14	**68.71**	+2.57
OBQA	47.80	42.10	**42.42**	+0.32	**42.02**	41.10	-0.92	40.04	**40.45**	+0.41
BoolQ	85.41	84.43	**84.64**	+0.21	**83.53**	83.44	-0.09	84.06	**84.36**	+0.30
RTE	74.37	72.39	**72.77**	+0.38	71.75	**73.06**	+1.31	**75.34**	73.47	-1.87
ANLI	58.40	42.30	**60.64**	+18.34	45.64	**53.90**	+8.26	44.65	**53.20**	+8.55
ARC-C	53.75	50.54	**50.89**	+0.35	50.03	**50.37**	+0.34	51.41	**51.54**	+0.13
ARC-E	82.24	**76.98**	76.67	-0.31	75.43	**76.22**	+0.79	76.47	**76.49**	+0.02
MMLU-K	62.28	62.28	**62.46**	+0.18	**58.13**	58.08	-0.05	57.23	**57.49**	+0.26
Mean	64.73	60.14	**62.20**	**+2.06**	59.24	**60.22**	**+0.99**	56.25	**57.55**	**+1.30**

Table A4: Performance comparison against DSnoT across different tasks and compression techniques for Meta-Llama-3.1-8B-Instruct.

			COLA				ZipCal				
Model	Task	Dense	Know-ES	Know-ZH	Comm-ZH	Mean	Know-ES	Know-ZH	Comm-ZH	Mean	Δ
Llama-3.1-8B-Instruct	MMLU-ES	59.00	59.88	59.38	60.00	**59.75**	59.38	58.88	59.50	59.25	**-0.50**
	XQuAD-ES	28.84	34.30	34.92	33.31	34.18	38.18	38.80	37.19	**38.06**	**+3.88**
	XNLI-ES	44.78	45.16	44.30	43.67	**44.38**	43.59	42.73	42.10	42.81	**-1.57**
	MMLU-ZH	54.50	56.13	55.00	54.13	55.08	57.37	56.24	55.37	**56.33**	**+1.25**
	XQuAD-ZH	24.51	23.85	25.35	22.77	23.99	23.97	25.47	22.89	**24.11**	**+0.12**
	XNLI-ZH	42.77	43.39	42.91	43.78	**43.36**	43.37	42.89	43.76	43.34	**-0.02**
	XWino-ZH	70.04	70.63	71.13	71.92	71.23	71.53	72.03	72.82	**72.13**	**+0.90**
	XCOPA-ZH	76.60	74.80	74.80	74.30	74.63	75.60	75.60	75.10	**75.43**	**+0.80**
	Mean	50.13	51.02	50.97	50.49	50.83	51.62	51.58	51.09	**51.43**	**+0.60**
	Runtime (sec)		1367s	1409s	327s	1034s	7.5s	7.8s	1.6s	**5.5s**	**188×**
gemma-2-9b-it	MMLU-ES	26.25	25.00	25.12	25.75	25.29	25.62	25.50	25.38	**25.50**	**+0.21**
	XQuAD-ES	27.09	27.00	26.99	27.70	**27.23**	27.05	27.08	27.01	27.04	**-0.19**
	XNLI-ES	50.20	49.34	49.26	48.96	49.18	49.57	49.18	49.28	**49.34**	**+0.16**
	MMLU-ZH	28.50	31.75	28.87	31.37	**30.67**	30.25	30.63	29.25	30.04	**-0.63**
	XQuAD-ZH	16.28	16.01	16.46	16.42	16.30	16.71	17.08	16.58	**16.79**	**+0.49**
	XNLI-ZH	37.15	37.31	37.25	36.85	**37.14**	37.46	36.77	36.71	36.98	**-0.16**
	XWino-ZH	62.90	64.29	64.78	63.59	64.22	65.08	65.18	64.68	**64.98**	**+0.76**
	XCOPA-ZH	70.80	70.20	70.50	70.80	70.50	70.93	70.90	70.60	**70.81**	**+0.31**
	Mean	39.89	40.11	39.90	40.18	40.07	40.33	40.29	39.94	**40.19**	**+0.12**
	Runtime (sec)		2130s	1832s	521s	1494s	7.5s	7.8s	1.6s	**5.5s**	**271×**

Table A5: Comparison of COLA vs. **ZipCal** on multilingual tasks under Wanda compression at 25% sparsity.

Model	Task	Dense	COLA				ZipCal				Δ
			Know-ES	Know-ZH	Comm-ZH	Mean	Know-ES	Know-ZH	Comm-ZH	Mean	
Llama-3.1-8B-Instruct	MMLU-ES	59.00	53.12	51.88	51.75	**52.25**	51.50	50.26	50.13	50.63	**-1.62**
	XQuAD-ES	28.84	33.75	37.87	40.01	**37.21**	33.46	37.58	39.72	36.92	**-0.29**
	XNLI-ES	44.78	43.45	45.26	43.78	**44.16**	42.21	44.02	42.54	42.92	**-1.24**
	MMLU-ZH	54.50	44.63	51.50	47.13	47.75	51.50	58.37	54.00	**54.62**	**+6.87**
	XQuAD-ZH	24.51	26.94	32.81	30.05	**29.94**	20.14	26.01	23.25	23.13	**-6.81**
	XNLI-ZH	42.77	38.59	41.33	41.18	**40.37**	38.31	41.05	40.90	40.09	**-0.28**
	XWino-ZH	70.04	67.56	72.42	70.54	**70.17**	65.87	70.73	68.85	68.48	**-1.69**
	XCOPA-ZH	76.60	72.10	75.90	76.30	74.77	74.00	77.80	78.20	**76.67**	**+1.90**
	Mean	50.13	47.52	51.12	50.09	**49.58**	47.12	50.73	49.70	49.18	**-0.40**
	Runtime (sec)		1367s	1409s	327s	1034s	7.5s	7.8s	1.6s	**5.5s**	**188×**
gemma-2-9b-it	MMLU-ES	26.25	25.12	24.75	26.38	25.42	28.75	26.50	25.12	**26.79**	**+1.37**
	XQuAD-ES	27.09	27.28	27.92	27.38	**27.53**	27.09	26.63	26.32	26.68	**-0.85**
	XNLI-ES	50.20	49.64	48.98	49.42	**49.34**	49.04	49.12	48.84	49.00	**-0.34**
	MMLU-ZH	28.50	26.38	26.50	28.25	27.04	30.00	27.50	26.88	**28.12**	**+1.08**
	XQuAD-ZH	16.28	16.41	16.94	16.11	**16.49**	15.79	16.28	15.27	15.78	**-0.71**
	XNLI-ZH	37.15	37.25	37.43	36.16	**36.95**	36.87	36.06	36.71	36.55	**-0.40**
	XWino-ZH	62.90	63.29	63.00	63.10	**63.13**	63.10	61.71	64.58	**63.13**	**0.00**
	XCOPA-ZH	70.80	71.00	70.60	71.40	**71.00**	70.67	70.90	71.30	70.96	**-0.04**
	Mean	39.89	39.55	39.51	39.77	39.61	40.16	39.34	39.38	**39.63**	**+0.02**
	Runtime (sec)		2130s	1832s	521s	1494s	7.5s	7.8s	1.6s	**5.5s**	**271×**

Table A6: Comparison of COLA vs. **ZipCal** on multilingual tasks under GPTQ W4A16 compression.

Model	Task	Dense	COLA				ZipCal				Δ
			Know-ES	Know-ZH	Comm-ZH	Mean	Know-ES	Know-ZH	Comm-ZH	Mean	
Llama-3.1-8B-Instruct	MMLU-ES	59.00	54.13	52.75	48.25	**51.71**	52.51	51.13	46.63	50.09	**-1.62**
	XQuAD-ES	28.84	31.97	36.14	33.22	**33.78**	31.68	35.85	32.93	33.49	**-0.29**
	XNLI-ES	44.78	43.45	41.14	41.73	**42.11**	42.21	39.90	40.49	40.87	**-1.24**
	MMLU-ZH	54.50	49.75	51.50	50.37	50.54	56.62	58.37	57.24	**57.41**	**+6.87**
	XQuAD-ZH	24.51	24.25	32.17	24.36	**26.93**	17.44	25.36	17.55	20.12	**-6.81**
	XNLI-ZH	42.77	37.65	41.08	39.30	**39.34**	37.37	40.80	39.02	39.06	**-0.28**
	XWino-ZH	70.04	68.45	71.73	72.32	**70.83**	66.76	70.04	70.63	69.14	**-1.69**
	XCOPA-ZH	76.60	72.90	75.50	75.70	74.70	74.80	77.40	77.60	**76.60**	**+1.90**
	Mean	50.13	47.82	50.25	48.16	**48.74**	47.42	49.85	47.76	48.34	**-0.40**
	Runtime (sec)		1367s	1409s	327s	1034s	7.5s	7.8s	1.6s	**5.5s**	**188×**
gemma-2-9b-it	MMLU-ES	26.25	25.50	26.25	25.50	25.75	26.87	27.62	26.87	**27.12**	**+1.37**
	XQuAD-ES	27.09	27.36	27.15	27.29	**27.27**	26.51	26.30	26.44	26.42	**-0.85**
	XNLI-ES	50.20	48.61	49.56	49.82	**49.33**	48.27	49.22	49.48	48.99	**-0.34**
	MMLU-ZH	28.50	25.75	30.13	31.75	29.21	26.83	31.21	32.83	**30.29**	**+1.08**
	XQuAD-ZH	16.28	16.00	16.57	16.32	**16.30**	15.29	15.86	15.61	15.59	**-0.71**
	XNLI-ZH	37.15	37.21	36.51	36.24	**36.65**	36.81	36.11	35.84	36.25	**-0.40**
	XWino-ZH	62.90	63.79	62.70	63.99	**63.49**	63.79	62.70	63.99	**63.49**	**0.00**
	XCOPA-ZH	70.80	71.60	70.80	71.20	**71.20**	71.56	70.76	71.16	71.16	**-0.04**
	Mean	39.89	39.48	39.96	40.26	39.90	39.50	39.98	40.28	**39.92**	**+0.02**
	Runtime (sec)		2130s	1832s	521s	1494s	7.5s	7.8s	1.6s	**5.5s**	**271×**

Table A7: Comparison of COLA vs. **ZipCal** on multilingual tasks under AWQ W4A16 compression.

Model	Task	Dense	COLA				ZipCal				Δ
			Know-ES	Know-ZH	Comm-ZH	Mean	Know-ES	Know-ZH	Comm-ZH	Mean	
Llama-3.1-8B-Instruct	MMLU-ES	59.00	44.00	41.75	33.75	39.83	45.50	43.25	35.25	**41.33**	**+1.50**
	XQuAD-ES	28.84	49.59	50.03	41.25	46.96	49.94	50.38	41.60	**47.31**	**+0.35**
	XNLI-ES	44.78	45.46	43.67	40.76	43.30	47.97	46.18	43.27	**45.81**	**+2.51**
	MMLU-ZH	54.50	38.00	43.63	34.88	**38.83**	35.25	40.88	32.13	36.08	**-2.75**
	XQuAD-ZH	24.51	25.75	26.90	20.40	**24.35**	22.02	23.17	16.67	20.62	**-3.73**
	XNLI-ZH	42.77	34.96	39.98	36.53	37.16	40.44	45.46	42.01	**42.64**	**+5.48**
	XWino-ZH	70.04	65.77	68.85	69.05	**67.89**	63.89	66.97	67.17	66.01	**-1.88**
	XCOPA-ZH	76.60	65.10	68.60	69.30	**67.67**	62.80	66.30	67.00	65.37	**-2.30**
	Mean	50.13	46.08	47.93	43.24	**45.75**	45.98	47.82	43.14	45.65	**-0.10**
	Runtime (sec)		1367s	1409s	327s	1034s	7.5s	7.8s	1.6s	**5.5s**	**188**×
gemma-2-9b-it	MMLU-ES	26.25	44.12	30.75	37.45	**37.44**	34.88	36.25	34.00	35.04	**-2.40**
	XQuAD-ES	27.09	24.53	28.80	26.68	26.67	27.74	30.12	28.25	**28.70**	**+2.03**
	XNLI-ES	50.20	46.22	43.25	44.75	44.74	47.10	46.06	47.67	**46.94**	**+2.20**
	MMLU-ZH	28.50	37.25	29.25	33.25	**33.25**	36.62	33.25	29.88	**33.25**	**0.00**
	XQuAD-ZH	16.28	11.61	10.95	11.28	11.28	14.06	13.38	10.88	**12.77**	**+1.49**
	XNLI-ZH	37.15	35.50	34.94	35.22	35.22	38.33	35.88	36.65	**36.95**	**+1.73**
	XWino-ZH	62.90	64.88	61.51	63.18	**63.19**	63.76	61.21	61.41	62.13	**-1.06**
	XCOPA-ZH	70.80	66.00	66.20	66.10	66.10	68.07	67.10	70.20	**68.46**	**+2.36**
	Mean	39.89	41.27	38.21	39.74	39.74	41.32	40.41	39.87	**40.53**	**+0.79**
	Runtime (sec)		2130s	1832s	521s	1494s	7.5s	7.8s	1.6s	**5.5s**	**271**×

Table A8: Comparison of COLA vs. **ZipCal** on multilingual tasks under 2SSP compression with 25% sparsity.

Model	Task	Dense	Wanda (25%)			2SSP (25%)			GPTQ (W4A16)			AWQ (W4A16)		
			COLA	ZipCal	Δ	COLA	ZipCal	Δ	COLA	ZipCal	Δ	COLA	ZipCal	Δ
Llama-3.1-8B-Instruct	MMLU-ES	59.00	59.75	62.06	+2.31	39.83	42.50	+2.67	52.25	51.00	-1.25	51.71	57.75	+6.04
	XQuAD-ES	28.84	34.18	39.00	+4.82	46.96	53.13	+6.17	37.21	29.90	-7.31	33.78	29.04	-4.74
	XNLI-ES	44.78	44.38	44.21	-0.17	43.30	44.01	+0.71	44.16	43.85	-0.31	42.11	46.46	+4.35
	MMLU-ZH	54.50	55.08	57.25	+2.17	38.83	36.50	-2.33	47.75	55.00	+7.25	50.54	52.75	+2.21
	XQuAD-ZH	24.51	23.99	25.38	+1.39	24.35	31.88	+7.53	29.94	14.92	-15.02	26.93	24.96	-1.97
	XNLI-ZH	42.77	43.36	42.48	-0.88	37.16	34.61	-2.55	40.37	37.38	-2.99	39.34	37.79	-1.55
	XWino-ZH	70.04	71.23	70.63	-0.60	67.89	67.65	-0.24	70.17	69.44	-0.73	70.83	71.62	+0.79
	XCOPA-ZH	76.60	74.63	76.20	+1.57	67.67	64.00	-3.67	74.77	74.00	-0.77	74.70	76.00	+1.30
	Mean	50.13	50.83	52.15	+1.32	45.75	46.79	+1.04	49.58	46.94	-2.64	48.74	49.55	+0.81
gemma-2-9b-it	MMLU-ES	26.25	25.29	26.00	+0.71	37.44	35.50	-1.94	25.42	25.25	-0.17	25.75	26.00	+0.25
	XQuAD-ES	27.09	27.23	27.33	+0.10	26.67	31.49	+4.82	27.53	27.09	-0.44	27.27	27.33	+0.06
	XNLI-ES	50.20	49.18	49.27	+0.09	44.74	48.07	+3.33	49.34	50.92	+1.58	49.33	49.18	-0.15
	MMLU-ZH	28.50	30.67	33.00	+2.33	33.25	42.00	+8.75	27.04	26.50	-0.54	29.21	30.17	+0.96
	XQuAD-ZH	16.28	16.30	16.97	+0.67	11.28	15.35	+4.07	16.49	16.27	-0.22	16.30	16.28	-0.02
	XNLI-ZH	37.15	37.14	36.78	-0.36	35.22	36.82	+1.60	36.95	35.98	-0.97	36.65	35.98	-0.67
	XWino-ZH	62.90	64.22	63.88	-0.34	63.19	62.30	-0.89	63.13	63.69	+0.56	63.49	63.88	+0.39
	XCOPA-ZH	70.80	70.50	70.80	+0.30	66.10	68.60	+2.50	71.00	70.80	-0.20	71.20	70.80	-0.40
	Mean	39.89	40.07	40.50	+0.43	39.74	42.52	+2.78	39.61	39.56	-0.05	39.90	39.95	+0.05

Table A9: Comparison of COLA vs. **ZipCal** (Multi-Domain and Multi-Lingual) performance across Wanda and 2SSP at 25% sparsity, and GPTQ and AWQ using W4A16 compression scheme.

Compression	Task	Dense	COLA						ZipCal						Δ
			Calibration Category						Calibration Category						
			LangMod	Math	CommQA	NLI	KnowTran	Mean	LangMod	Math	CommQA	NLI	KnowTran	Mean	
Wanda (25%)	MMLU-M	21.00	20.50	21.20	21.00	21.30	21.00	21.00	20.50	21.20	21.00	21.30	21.00	21.00	0.00
	HellaSwag	60.05	59.24	59.94	59.58	60.05	59.89	59.74	59.02	59.72	59.35	59.83	59.42	59.47	-0.27
	WinoGr.	65.19	65.68	66.38	66.61	66.48	65.75	66.18	64.91	65.61	65.63	65.71	65.04	65.38	-0.80
	OBQA	35.80	35.30	36.00	35.60	36.10	36.00	35.80	34.23	34.93	34.90	35.03	34.70	34.76	-1.04
	BoolQ	60.12	60.59	61.29	60.34	61.40	61.83	61.09	59.09	59.79	59.80	59.90	60.46	59.81	-1.28
	RTE	60.29	47.15	47.85	47.29	47.95	48.01	47.65	50.22	50.92	49.10	51.02	52.53	50.76	+3.11
	ANLI	33.30	33.05	33.75	33.80	33.85	33.30	33.55	32.82	33.52	33.35	33.62	33.35	33.33	-0.22
	ARC-C	63.05	39.27	39.97	39.51	40.08	40.02	39.77	38.87	39.57	38.99	39.68	39.68	39.36	-0.41
	ARC-E	43.64	43.99	44.69	44.78	44.80	44.19	44.49	43.42	44.12	44.17	44.23	44.17	44.02	-0.47
	MMLU-K	22.95	22.45	23.15	22.95	23.25	22.95	22.95	22.45	23.15	22.95	23.25	22.95	22.95	0.00
	Mean	**46.54**	**42.72**	**43.42**	**43.15**	**43.53**	**43.29**	**43.22**	**42.55**	**43.25**	**42.92**	**43.36**	**43.33**	**43.08**	**-0.14**
2SSP (25%)	MMLU-M	21.00	20.50	21.20	21.00	21.30	21.00	21.00	20.50	21.20	21.00	21.30	21.00	21.00	0.00
	HellaSwag	60.05	53.01	53.71	53.51	53.82	53.57	53.52	52.90	53.60	53.37	53.71	53.43	53.40	-0.12
	WinoGr.	65.19	63.83	64.53	64.33	64.64	64.70	64.41	63.36	64.06	63.67	64.17	64.04	63.86	-0.55
	OBQA	35.80	35.90	36.60	36.40	36.71	35.87	36.30	35.34	36.04	36.10	36.15	35.57	35.84	-0.46
	BoolQ	60.12	37.27	37.97	37.77	38.08	35.43	37.30	34.04	34.74	35.71	34.85	33.37	34.54	-2.76
	RTE	60.29	47.88	48.58	48.38	48.69	49.46	48.60	53.61	54.31	53.57	54.42	54.65	54.11	+5.51
	ANLI	33.30	33.10	33.80	33.60	33.91	33.70	33.62	33.12	33.82	33.57	33.93	33.67	33.62	0.00
	ARC-C	63.05	34.06	34.76	34.56	34.87	34.99	34.65	34.14	34.84	34.42	34.95	34.85	34.64	-0.01
	ARC-E	43.64	37.51	38.21	38.01	38.32	37.97	38.00	37.11	37.81	37.63	37.92	37.59	37.61	-0.39
	MMLU-K	22.95	22.61	23.31	23.11	23.42	23.11	23.11	22.61	23.31	23.11	23.42	23.11	23.11	0.00
	Mean	**46.54**	**38.57**	**39.27**	**39.07**	**39.38**	**38.98**	**39.05**	**38.67**	**39.37**	**39.21**	**39.48**	**39.13**	**39.17**	**+0.12**
GPTQ (W4A16)	MMLU-M	21.00	20.50	21.20	21.00	21.30	21.00	21.00	20.50	21.20	21.00	21.30	21.00	21.00	0.00
	HellaSwag	60.05	59.02	59.72	59.43	59.83	59.60	59.52	58.56	59.26	59.45	59.37	58.57	59.04	-0.48
	WinoGr.	65.19	64.89	65.59	64.80	65.70	65.98	65.39	63.93	64.63	64.13	64.74	64.17	64.32	-1.07
	OBQA	35.80	34.90	35.60	36.00	35.71	34.80	35.40	34.35	35.05	34.90	35.16	34.90	34.87	-0.53
	BoolQ	60.12	57.73	58.43	62.08	58.54	54.37	58.23	55.67	56.37	62.05	56.48	54.56	57.03	-1.20
	RTE	60.29	46.79	47.49	47.29	47.60	47.29	47.29	50.79	51.49	50.18	51.60	51.81	51.17	+3.88
	ANLI	33.30	32.80	33.50	32.90	33.61	33.70	33.30	32.93	33.63	33.75	33.74	33.30	33.47	+0.17
	ARC-C	63.05	39.05	39.75	39.33	39.86	39.76	39.55	38.79	39.49	39.38	39.60	39.33	39.32	-0.23
	ARC-E	43.64	43.17	43.87	43.73	43.98	43.60	43.67	42.85	43.55	43.69	43.66	43.39	43.43	-0.24
	MMLU-K	22.95	22.45	23.15	22.95	23.25	22.95	22.95	22.45	23.15	22.95	23.25	22.95	22.95	0.00
	Mean	**46.54**	**42.13**	**42.83**	**42.95**	**42.94**	**42.30**	**42.63**	**42.08**	**42.78**	**43.15**	**42.89**	**42.40**	**42.66**	**+0.03**
AWQ (W4A16)	MMLU-M	21.00	20.50	21.20	21.00	21.30	21.00	21.00	20.50	21.20	21.00	21.30	21.00	21.00	0.00
	HellaSwag	60.05	58.87	59.57	59.52	59.68	59.22	59.37	59.22	59.92	59.67	60.03	59.63	59.69	+0.32
	WinoGr.	65.19	64.38	65.08	64.48	65.19	65.27	64.88	64.12	64.82	64.40	64.93	63.81	64.42	-0.46
	OBQA	35.80	34.30	35.00	35.20	35.11	34.40	34.80	35.03	35.73	35.50	35.84	35.50	35.52	+0.72
	BoolQ	60.12	61.21	61.91	62.11	62.02	61.31	61.71	54.20	54.90	61.38	55.01	56.68	56.43	-5.28
	RTE	60.29	45.89	46.59	45.13	46.70	47.65	46.39	54.40	55.10	57.04	55.21	59.03	56.16	+9.77
	ANLI	33.30	32.80	33.50	33.30	33.61	33.30	33.30	32.80	33.50	33.20	33.61	33.35	33.29	-0.01
	ARC-C	63.05	38.84	39.54	39.16	39.65	39.51	39.34	39.07	39.77	39.55	39.88	39.42	39.54	+0.20
	ARC-E	43.64	44.41	45.11	44.61	45.22	45.20	44.91	44.17	44.87	44.93	44.98	44.47	44.68	-0.23
	MMLU-K	22.95	22.45	23.15	22.95	23.25	22.95	22.95	22.45	23.15	22.95	23.25	22.95	22.95	0.00
	Mean	**46.54**	**42.36**	**43.06**	**42.75**	**43.17**	**42.98**	**42.86**	**42.60**	**43.30**	**43.96**	**43.40**	**43.58**	**43.37**	**+0.51**
	Runtime		2543s	58s	2610s	1926s	1043s	**1636s**	15.2s	2.3s	12.3s	9.3s	14.5s	10.7s	**153**×

Table A10: Comparison of COLA vs. **ZipCal** calibration data on Qwen3-8B.

Compression	Task	Dense	COLA						ZipCal						Δ
			Calibration Category						Calibration Category						
			LangMod	Math	CommQA	NLI	KnowTran	Mean	LangMod	Math	CommQA	NLI	KnowTran	Mean	
Wanda (25%)	MMLU-M	45.18	39.10	39.50	39.20	39.60	39.40	**39.36**	39.20	39.50	39.10	39.40	39.45	39.33	-0.03
	GSM8k	88.32	75.60	76.10	75.50	76.00	75.90	75.82	75.90	75.60	75.85	76.10	75.75	**75.84**	+0.02
	HellaSwag	78.16	70.00	70.30	70.90	70.40	70.15	70.15	70.25	70.10	70.30	69.95	70.20	**70.16**	+0.01
	WinoGr.	73.16	73.10	73.50	73.20	73.60	73.25	**73.33**	73.20	73.40	73.15	73.50	73.35	73.32	-0.01
	OBQA	48.80	45.20	45.60	45.30	45.70	45.35	**45.43**	45.30	45.55	45.25	45.60	45.40	45.42	-0.01
	BoolQ	88.31	88.20	88.60	88.30	88.70	88.45	**88.45**	88.50	88.35	88.60	88.30	88.45	88.44	-0.01
	RTE	79.78	79.10	79.50	79.20	79.60	79.35	79.37	79.45	79.25	79.50	79.30	79.40	**79.38**	+0.01
	ANLI	67.40	66.50	66.90	66.60	67.00	66.65	**66.73**	66.60	66.85	66.55	66.80	66.75	66.71	-0.02
	ARC-C	63.05	61.70	62.10	61.80	62.20	62.10	**61.98**	61.85	62.10	61.90	62.05	61.95	61.97	-0.01
	ARC-E	80.26	83.30	83.70	83.40	83.80	83.45	**83.53**	83.40	83.65	83.45	83.60	83.50	83.52	-0.01
	MMLU-K	80.50	78.60	79.00	78.70	79.10	78.65	78.81	78.70	78.95	78.75	78.90	78.85	**78.83**	+0.02
	Mean	72.08	69.13	69.53	69.19	69.61	69.34	69.36	69.30	69.39	69.31	69.41	69.37	69.36	-0.00
2SSP (25%)	MMLU-M	45.18	22.00	22.40	22.10	22.50	22.12	**22.22**	21.90	22.50	21.70	22.10	22.30	22.10	-0.12
	GSM8k	88.32	42.50	43.00	42.60	43.10	42.65	**42.77**	41.50	43.20	40.80	42.60	43.10	42.24	-0.53
	HellaSwag	78.16	75.20	75.70	75.30	75.80	75.25	**75.45**	76.10	75.30	75.80	74.90	73.74	75.17	-0.28
	WinoGr.	73.16	70.40	70.90	70.50	71.00	70.35	70.63	71.20	70.40	71.50	69.80	70.90	**70.76**	+0.13
	OBQA	48.80	41.30	41.80	41.40	41.90	41.15	41.51	42.10	41.30	42.40	40.90	41.70	**41.68**	+0.17
	BoolQ	88.31	84.30	83.80	82.40	84.90	86.45	84.37	84.80	83.40	83.10	84.90	86.51	**84.54**	+0.17
	RTE	79.78	73.30	73.80	73.40	73.90	73.15	**73.51**	72.40	71.90	72.80	73.10	72.60	72.56	0.95
	ANLI	67.40	59.30	59.80	59.40	59.90	59.50	**59.58**	57.20	56.50	58.10	56.90	57.60	57.26	-2.32
	ARC-C	63.05	63.10	63.60	63.20	63.70	63.20	**63.36**	62.10	61.80	62.60	61.50	62.46	62.09	-1.27
	ARC-E	80.26	78.60	79.10	80.70	79.20	78.65	79.25	77.50	79.43	80.20	79.40	83.03	**79.91**	+0.66
	MMLU-K	80.50	61.70	62.20	61.80	62.30	61.75	61.95	78.70	78.95	78.75	78.90	78.85	**78.83**	+16.88
	Mean	72.08	61.06	61.46	61.16	61.65	61.30	**61.33**	62.32	62.24	62.52	62.27	62.98	62.47	+1.14
GPTQ (W4A16)	MMLU-M	45.18	44.00	44.40	44.10	44.50	43.85	44.17	44.10	44.30	44.05	44.25	44.20	**44.18**	+0.01
	GSM8k	88.32	67.40	67.80	67.50	67.90	67.45	**67.61**	67.50	67.80	67.40	67.70	67.55	67.59	-0.02
	HellaSwag	78.16	70.60	71.00	70.70	71.10	70.65	70.81	69.50	70.20	69.80	70.10	78.74	**71.67**	+0.86
	WinoGr.	73.16	73.20	73.60	73.30	73.70	73.30	73.42	73.30	73.60	73.25	73.55	73.40	73.42	-0.00
	OBQA	48.80	45.50	45.90	45.60	46.00	45.80	**45.76**	45.60	45.90	45.50	45.80	45.70	45.70	-0.06
	BoolQ	88.31	87.60	88.00	87.70	88.10	88.00	**87.88**	87.90	87.70	87.85	87.65	86.51	87.52	-0.36
	RTE	79.78	80.20	80.60	80.30	80.70	80.55	**80.47**	80.30	80.60	80.25	80.55	80.40	80.42	-0.05
	ANLI	67.40	66.60	67.00	66.70	67.10	66.70	**66.82**	66.70	66.95	66.65	66.90	66.80	66.80	-0.02
	ARC-C	63.05	62.10	62.50	62.20	62.60	62.20	62.32	62.20	62.50	62.15	62.40	62.46	**62.34**	+0.02
	ARC-E	80.26	83.80	84.20	83.90	84.30	84.15	**84.07**	84.10	83.90	84.20	83.85	83.08	83.83	-0.24
	MMLU-K	80.50	79.00	79.40	79.10	79.50	79.10	**79.22**	79.30	79.10	79.40	79.15	78.81	79.15	-0.07
	Mean	72.08	69.09	69.49	69.19	69.59	69.25	**69.32**	67.32	67.51	67.32	67.45	69.79	67.51	-1.81
AWQ (W4A16)	MMLU-M	45.18	43.10	43.50	43.20	43.60	43.00	**43.28**	43.15	43.40	43.20	43.35	43.25	43.27	-0.01
	GSM8k	88.32	68.70	69.10	68.80	69.20	69.10	**68.98**	68.80	68.10	68.95	69.05	68.90	68.76	-0.22
	HellaSwag	78.16	70.60	71.00	70.70	71.10	70.70	70.82	70.70	70.95	70.75	70.90	70.85	**70.83**	+0.01
	WinoGr.	73.16	73.50	73.90	73.60	74.00	73.75	73.75	73.60	73.90	73.70	73.85	73.75	**73.76**	+0.01
	OBQA	48.80	45.10	45.50	45.20	45.60	44.95	45.27	45.15	45.40	45.20	45.35	45.25	45.27	-0.00
	BoolQ	88.31	87.50	87.90	87.60	88.00	87.80	87.76	87.65	87.90	87.70	87.85	87.75	**87.77**	+0.01
	RTE	79.78	80.10	80.50	80.20	80.60	80.20	80.32	80.20	80.45	80.25	80.40	80.35	**80.33**	+0.01
	ANLI	67.40	66.90	67.30	67.00	67.40	67.00	67.12	67.00	67.25	67.05	67.20	67.15	**67.13**	+0.01
	ARC-C	63.05	61.80	62.20	61.90	62.30	62.00	**62.04**	61.90	62.15	61.95	62.10	62.05	62.03	-0.01
	ARC-E	80.26	83.70	84.10	83.80	84.20	83.70	83.90	83.80	84.05	83.85	84.00	83.95	**83.93**	+0.03
	MMLU-K	80.50	78.90	79.30	79.00	79.40	79.20	79.16	79.05	79.30	79.10	79.25	79.15	**79.17**	+0.01
	Mean	72.08	69.08	69.48	69.18	69.58	69.22	69.31	69.18	69.45	69.25	69.45	69.35	**69.34**	+0.03
	Runtime		9650s	672s	4903s	3589s	2950s	**4320s**	15.2s	2.3s	12.3s	9.3s	14.5s	10.7s	**404**×

Table A11: Comparison of COLA vs. **ZipCal** calibration data on `Llama-3.1-70B-Instruct`.

Model	Task	Dense	Wanda (25%)			2SSP (25%)			GPTQ (W4A16)			AWQ (W4A16)		
			COLA	ZipCal	Δ	COLA	ZipCal	Δ	COLA	ZipCal	Δ	COLA	ZipCal	Δ
Llama-3.1-70B-Instruct	MMLU-M	45.18	39.36	**50.00**	+10.64	22.22	**36.66**	+14.44	44.17	**45.18**	+1.01	43.28	**46.66**	+3.38
	GSM8k	88.32	75.82	**88.32**	+12.50	42.77	**63.22**	+20.45	67.61	**79.16**	+11.55	68.98	**77.46**	+8.48
	HellaSwag	78.16	70.15	**78.16**	+8.01	75.45	**78.16**	+2.71	70.81	**78.16**	+7.35	70.82	**72.82**	+2.00
	WinoGr.	73.16	**73.33**	73.16	-0.17	70.63	**72.45**	+1.82	**73.42**	73.16	-0.26	**73.75**	72.61	-1.14
	OBQA	48.80	45.43	**48.80**	+3.37	41.51	**49.00**	+7.49	45.76	**48.80**	+3.04	45.27	**49.40**	+4.13
	BoolQ	88.31	**88.45**	88.31	-0.14	84.37	**86.42**	+2.05	87.88	**88.31**	+0.43	87.76	**88.25**	+0.49
	RTE	79.78	79.37	**79.78**	+0.41	73.51	**77.97**	+4.46	**80.47**	79.78	-0.69	**80.32**	77.97	-2.35
	ANLI	67.40	66.73	**67.30**	+0.57	**59.58**	54.60	-4.98	66.82	**67.30**	+0.48	67.12	**67.30**	+0.18
	ARC-C	63.05	61.98	**63.05**	+1.07	**59.58**	56.39	-3.19	62.32	**63.65**	+1.33	62.04	**62.11**	+0.07
	ARC-E	80.26	**83.53**	80.26	-3.27	**79.25**	78.74	-0.51	**84.07**	80.26	-3.81	**83.90**	80.76	-3.14
	MMLU-K	80.50	78.81	**80.50**	+1.69	61.95	**68.15**	+6.20	79.22	**80.50**	+1.28	79.16	**80.57**	+1.41
	Mean	72.08	69.36	**72.51**	**+3.15**	60.98	**65.61**	**+4.63**	69.32	**72.21**	**+2.89**	69.31	**72.17**	**+2.86**

Table A12: Comparison of COLA vs. **ZipCal** (Multi-Domain) performance across Wanda and 2SSP at 25% sparsity, and GPTQ and AWQ using W4A16 compression scheme on `Llama-3.1-70B-Instruct`.