Perplexity-Spectrum Calibration for Post-Training Pruning of Large Language Models

Qizhu Wang[1,2], Xingzhou Zhang[2,†], Zhihao Qu[1,†], Wenfeng Xu[1], Wentao Wang[3]
[1]College of Computer Science and Software Engineering, Hohai University, Nanjing, China
[2]Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing, China
[3]Nanjing University of Posts and Telecommunications, Nanjing, China

***Abstract*—Post-training pruning enables efficient deployment of large language models (LLMs) without costly retraining, but it relies on calibration data for weight importance estimation. Related studies analyze the impact of calibration data along two dimensions: data sources and input formats. However, once the dataset is fixed, calibration samples are typically selected by random sampling. Therefore, sample selection within the same data source remains underexplored. We propose a perplexity-spectrum calibration sample selection strategy. This strategy uses the target model's perplexity to measure sample difficulty. It sorts the candidate pool by perplexity and selects a contiguous difficulty window as the calibration set. We evaluate this strategy on three 7B models: Qwen2-7B, Mistral-7B, and Llama-2-7B. Both Wanda and SparseGPT are tested under 2:4 sparsity. Experiments show that the medium-difficulty window centered at the median achieves the best pruning results across all settings. Compared to random sampling, this strategy reduces perplexity degradation by 20%–26% on WikiText-2. On four zero-shot downstream reasoning tasks, it reduces the standard deviation by 3.4–$6.1\times$. Further analysis reveals that medium-difficulty samples induce stronger and more selective neuron activations. This improves pruning mask consistency (Jaccard similarity) from 0.74 to 0.82. The strategy requires one forward pass to score the candidate pool and a sorting step. It serves as an efficient, plug-and-play calibration optimization step.**

***Index Terms*—Large language models, post-training pruning, calibration data, model compression**

I. Introduction

Large language models (LLMs) have achieved remarkable performance across diverse natural language tasks, yet their deployment remains constrained by substantial computational costs. A 7B-parameter model requires approximately 14GB of memory at 16-bit precision, limiting deployment on consumer hardware and increasing inference latency in production systems. Model compression, particularly pruning, has emerged as a practical solution to reduce these costs while preserving model capabilities [1]–[3].

Post-training pruning is attractive because it avoids retraining, which can be prohibitively expensive for billion-parameter models. Recent work has improved pruning quality substantially. SparseGPT [4] casts pruning as layer-wise sparse regression with second-order (Hessian) information and applies weight reconstruction. Wanda [5] uses a first-order criterion that combines weight magnitude with activation norms, without updating weights. Subsequent methods further refine the scoring and sparsity allocation scheme. For example, DSnoT performs training-free mask refinement via iterative pruning and regrowing [6]. BESA explores block-wise pruning with learned sparsity allocation [7]. OWL adopts outlier-weighted layer-wise sparsity allocation [8]. Despite these algorithmic differences, all of the above methods rely on calibration data to estimate activation statistics. Moreover, 2:4 semi-structured sparsity can be accelerated on NVIDIA Ampere GPUs via dedicated hardware support, offering up to $2\times$ speedup with 50% weight reduction [9].

We adopt Wanda and SparseGPT as representative pruning methods: Wanda uses a simple importance metric with no weight updates, while SparseGPT performs layer-wise reconstruction with weight updates. Validating on both demonstrates generalization across the algorithmic spectrum.

Calibration data substantially affects pruning quality. This raises a fundamental question: **how should calibration data be selected to optimize pruning outcomes?** Existing studies have examined this question from two angles: the data source and the data format. In terms of data source, Williams and Aletras [10] observe significant performance variability across corpora, while Bandari et al. [11] highlight that C4 may be suboptimal compared to the Pile. Further, Ji et al. [12] demonstrate that data mirroring the pretraining distribution yields superior results. Accordingly, they propose the use of self-generated samples for calibration. Regarding data format, incorporating in-context learning (ICL) formatting has been shown to improve pruning outcomes [11]. However, these approaches uniformly rely on *random sampling* once the data source and format are established.

This leaves a third dimension unexplored: **which samples within a given corpus should be selected for calibration?** Even within one corpus, samples vary widely in difficulty as perceived by the model. Some sentences are highly predictable with low perplexity. Others are challenging with high perplexity. This variation matters because all post-training pruning methods compute weight importance based on calibration-induced activations. Consider Wanda's importance score $S_{ij} = |W_{ij}| \cdot \|X_j\|_2$. When easy samples produce uniformly weak activations, the $\|X_j\|_2$ terms become nearly constant across channels. This reduces the metric to magnitude-only pruning and loses activation-based discriminability. Conversely,

†Corresponding authors: zhangxingzhou@ict.ac.cn, quzhihao@hhu.edu.cn.
This work was supported in part by the National Natural Science Foundation of China (No. 62572171, No. 62402475) and the Basic Research Program of Jiangsu (No. BK20253011).

when hard samples produce erratic activations, the importance rankings become unstable across different calibration sets. This leads to inconsistent pruning decisions. Medium-difficulty samples may strike a balance, activating the model meaningfully without triggering unpredictable behavior.

Based on this insight, we propose a perplexity-spectrum calibration strategy. Perplexity is a natural difficulty proxy. It is model-specific and efficient, requiring only one forward pass per sample. It is also widely used in curriculum learning research [13]. We sort the candidate pool by perplexity and select contiguous windows at different difficulty levels. These include Easy (lowest perplexity), Hard (highest perplexity), Medium (centered at the median), and Random (uniform sampling as baseline). Experiments on three 7B models with Wanda and SparseGPT under 2:4 sparsity show that a median-centered medium-difficulty window consistently improves pruning. It reduces WikiText-2 perplexity degradation by 20%–26% relative to random sampling. It also lowers standard deviation on four zero-shot tasks by 3.4–$6.1\times$. Activation and mask analysis suggests that medium-difficulty samples yield stronger activations and more consistent masks. Specifically, the Jaccard similarity is 0.82 for Medium and 0.74 for Random.

Contributions. We make three contributions:

1) We propose a perplexity-spectrum calibration strategy. The default rule is to select the medium-difficulty window centered at the median. This strategy requires only one forward pass over the candidate pool for scoring. It is independent of the underlying corpus and can serve as a second-stage filter after any data source selection. On WikiText-2, the strategy reduces perplexity degradation by 20%–26% compared with random sampling. This holds across Qwen2, Mistral, and Llama-2 with both Wanda and SparseGPT under 2:4 sparsity.
2) We quantify the effect of calibration sample difficulty on pruning stability. The medium window significantly improves reproducibility. On four zero-shot tasks (ARC-Easy, BoolQ, HellaSwag, PIQA), it reduces the standard deviation by 3.4–$6.1\times$ compared to random sampling.
3) We analyze calibration activations and mask consistency to explain the performance and stability gains. This analysis provides mechanistic evidence and supports transferability across pruning algorithms. On Qwen2-7B, the Jaccard similarity increases from 0.74 to 0.82.

II. Related Work

A. Post-Training Pruning for LLMs

The computational demands of large language models have motivated extensive research on model compression [1]. Post-training pruning is particularly attractive as it requires no retraining. Retraining is prohibitively expensive for billion-scale models. Many post-training pruning methods estimate weight importance from calibration activations. Representative examples include Wanda and SparseGPT.

Wanda [5] assigns an importance score to each weight. The score is the product of weight magnitude and the ℓ_2 norm of the corresponding input activation:

$$S_{ij} = |W_{ij}| \cdot \|X_j\|_2, \tag{1}$$

Here W_{ij} connects input neuron j to output neuron i. $X_j \in \mathbb{R}^{n\times d}$ denotes the activations of the j-th input feature over n calibration samples. Wanda prunes weights with the smallest S_{ij} and does not update the remaining weights.

SparseGPT [4] formulates pruning as a layer-wise sparse reconstruction problem:

$$\hat{W} = \arg\min_{W' \in \mathcal{S}_k} \|WX - W'X\|_F^2, \tag{2}$$

Here X is the input activation matrix from calibration data. $\mathcal{S}_k$ denotes the set of weight matrices that satisfy the target sparsity constraint. $\|\cdot\|_F$ is the Frobenius norm. SparseGPT uses second-order information and updates the remaining weights to reduce reconstruction error.

Both methods rely on calibration activations to estimate importance. Different calibration samples can lead to different activation statistics and thus different pruning decisions. Follow-up methods refine the scoring and allocation scheme [6]–[8] or explore structured pruning [14]–[16].

B. Calibration Data for Model Compression

Post-training compression methods rely on calibration data to estimate weight importance or activation statistics. A common default is to sample 128 sequences of 2,048 tokens from C4 [17], but recent studies suggest this choice can materially affect performance.

Williams and Aletras [10] report substantial variation across calibration sources. The accuracy gap can reach 7.5% with SparseGPT. Bandari et al. [11] find that C4 is not always optimal. The Pile often yields better results, and ICL formatting tends to improve pruning outcomes. Ji et al. [12] show that calibration data closer to the pretraining distribution yields better performance. They propose self-generated calibration when pretraining data is unavailable. Other factors such as sequence length [18] and language diversity [19] also affect compression outcomes.

Calibration data also affects quantization methods such as GPTQ [20] and AWQ [21]. Our difficulty-based selection may benefit these methods as well.

C. Difficulty-Based Selection in Machine Learning

Curriculum learning [13] shows that ordering examples by difficulty can improve optimization and generalization. However, curriculum learning differs from our work in several aspects. First, it targets training dynamics, while we target one-shot importance estimation. Second, it relies on sample ordering across gradient updates, while our method affects a single forward pass with no training. Third, it typically progresses from easy to hard, while our results favor medium-difficulty samples.

Table I positions our work relative to prior studies. Existing research primarily addresses what data source or format to use.

TABLE I
COMPARISON WITH PRIOR WORK ON CALIBRATION DATA SELECTION.

Study	Dimension	Key Findings
Williams & Aletras [10]	Data source	Cross-source variance
Bandari et al. [11]	Source + format	Pile > C4; ICL helps
Ji et al. [12]	Distribution	Pretraining-similar better
This work	Sample difficulty	Medium is optimal

We explore a complementary dimension: how does sample difficulty affect pruning outcomes? Even within a single corpus, examples span a wide range of model-perceived difficulty. This difficulty distribution can systematically affect activation-based importance scores. Our method can be used alongside source-selection strategies with only a small preprocessing overhead.

III. METHOD

This section presents our perplexity-spectrum calibration framework. Fig. 1 illustrates the overall pipeline.

A. Problem Formulation

Post-training pruning methods rely on calibration data to estimate weight importance. We consider a candidate pool $\mathcal{D} = \{x_1, \ldots, x_N\}$ of N text sequences. Each sequence contains L tokens. The goal is to select a calibration subset of K samples ($K \ll N$) that yields the best pruning quality.

Existing methods draw calibration samples by uniform random sampling. This ignores the variation in sample characteristics. We propose to select samples based on their difficulty as perceived by the model.

B. Difficulty Measurement

We use model perplexity as a proxy for sample difficulty. For a sequence $x = (x_1, \ldots, x_T)$, perplexity is defined as:

$$\text{PPL}(x) = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log p_\theta(x_t|x_{<t})\right) \tag{3}$$

where $p_\theta(x_t|x_{<t})$ is the probability assigned by the model to token x_t given the preceding context.

Perplexity measures how surprised the model is by a sequence. Low values indicate predictable (easy) samples. High values indicate unpredictable (hard) samples. This metric is model-specific. The same text may have different difficulty for different models. This property is desirable because we want difficulty relative to the model being pruned.

Computing perplexity requires one forward pass per sample. For N candidates, the total overhead is $O(N)$ forward passes. This is negligible compared to the pruning phase.

C. Spectrum Slicing Strategies

Given perplexity scores $\{\text{PPL}(x_i)\}_{i=1}^{N}$, we sort candidates in ascending order.

We define four strategies (Fig. 1(b)): **Easy** selects the K lowest-perplexity samples (indices 1 to K); **Hard** selects the K highest (indices $N-K+1$ to N); **Medium** selects K samples centered at the median (indices $\lfloor\frac{N-K}{2}\rfloor+1$ to $\lfloor\frac{N-K}{2}\rfloor+K$); **Random** uniformly samples K sequences as baseline.

Algorithm 1 Perplexity-Spectrum Calibration Selection

Require: Candidate pool $\mathcal{D}$, size N; budget K; strategy s; model θ
Ensure: Calibration set $\mathcal{C}$ with $|\mathcal{C}| = K$

```
if s = Random then
    return UniformSample(D, K)
end if
for i = 1 to N do
    p_i ← PPL_θ(x_i)
end for
π ← argsort({p_i}) {ascending order}
if s = Easy then
    C ← {x_{π_j}}_{j=1}^{K}
else if s = Medium then
    m ← ⌊(N − K)/2⌋ + 1
    C ← {x_{π_j}}_{j=m}^{m+K−1}
else if s = Hard then
    C ← {x_{π_j}}_{j=N−K+1}^{N}
end if
return C
```

Algorithm 1 summarizes the procedure. The computational cost is $O(N)$ forward passes plus $O(N \log N)$ sorting.

We hypothesize that medium-difficulty samples are most informative. Easy samples produce weak, uniform activations that fail to differentiate weights. Hard samples produce noisy activations that yield unreliable importance estimates. Medium samples represent the model's typical operating regime, producing activations that are both strong and stable. Section IV tests this hypothesis; Section V provides mechanistic analysis.

IV. EXPERIMENTS

We design experiments to answer three questions: (1) Does Medium outperform Random across different models and pruning methods? (2) Does the improvement transfer to downstream tasks? (3) How does the calibration budget affect the results?

A. Experimental Setup

We conduct experiments on three representative open-source LLMs at the 7B parameter scale: Qwen2-7B-Instruct, Mistral-7B-Instruct-v0.3, and Llama-2-7b-hf. These models differ in architecture details, training corpora, and tokenization schemes. They provide diverse testbeds for evaluating our calibration strategy.

Two post-training pruning methods are employed in our experiments. Wanda computes importance scores as $|W_{ij}| \cdot \|X_j\|_2$ and prunes weights with smallest scores without any weight updates [5]. SparseGPT leverages approximate Hessian information and updates remaining weights to compensate for pruning errors [4]. Both methods operate under 2:4 semi-structured sparsity, where exactly 2 out of every 4 consecutive

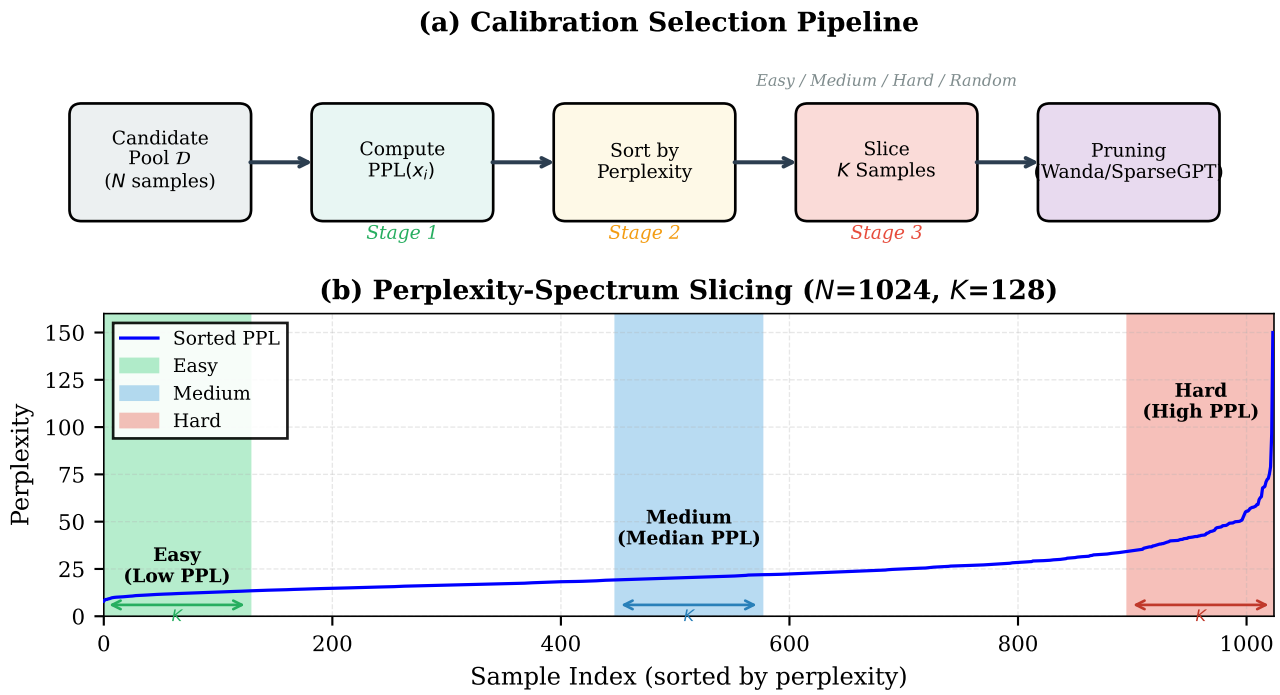

Fig. 1. Overview of perplexity-spectrum calibration. (a) Selection pipeline. (b) Spectrum slicing (N=1024, K=128).

weights are set to zero. This sparsity pattern achieves 50% effective compression while enabling hardware acceleration on NVIDIA Ampere GPUs [9].

For calibration data, we follow the established practice and use C4 as the source corpus [11], [17]. We randomly sample a candidate pool of $N = 1024$ sequences from C4, with each sequence containing $L = 2048$ tokens. The calibration budget is set to $K = 128$ samples, consistent with prior work. We compute the perplexity of each candidate using the target model. Samples are then selected according to the Easy, Medium, Hard, or Random strategy described in Section III.

We evaluate pruned models on WikiText-2 perplexity for language modeling quality. We also report zero-shot accuracy on four downstream tasks: ARC-Easy, BoolQ, HellaSwag, and PIQA. Downstream evaluation is performed using lm-evaluation-harness [22]. To assess the stability of pruning outcomes, we run each configuration with 5 different random seeds and report both mean and standard deviation.

B. Main Results

Table II presents the WikiText-2 perplexity after 2:4 pruning across all model-method combinations. We compare the proposed Medium strategy against Random sampling, which serves as the commonly used baseline in existing literature.

The results show that Medium consistently outperforms Random across all six model-method combinations. It reduces perplexity by 0.89–1.56. In terms of relative improvement, Medium reduces the pruning-induced degradation by 20%–26%. Here ΔPPL denotes the increase relative to the dense model. For instance, on Qwen2-7B with Wanda, Random increases perplexity by 5.90 (from 7.48 to 13.38), while Medium increases it by only 4.34. Both Wanda and SparseGPT benefit from Medium selection. This indicates that the advantage stems from better importance estimation rather than interaction with weight reconstruction.

TABLE II
WIKITEXT-2 PERPLEXITY (↓) AFTER 2:4 PRUNING. RESULTS AVERAGED OVER 5 SEEDS.

Model	Method	Dense	Random	Medium	Δ
Qwen2-7B	Wanda	7.48	13.38	**11.82**	−1.56
	SparseGPT	7.48	12.15	**10.94**	−1.21
Mistral-7B	Wanda	6.04	11.14	**10.08**	−1.06
	SparseGPT	6.04	10.52	**9.63**	−0.89
Llama-2-7B	Wanda	6.74	12.14	**10.94**	−1.20
	SparseGPT	6.74	11.38	**10.45**	−0.93

Table III compares all four strategies on Qwen2-7B with Wanda. A clear ordering emerges: Medium > Random > Hard > Easy. Both extremes underperform the random baseline, confirming that calibration effectiveness follows an inverted-U relationship with sample difficulty.

C. Downstream Tasks and Variance Analysis

Beyond language modeling perplexity, we evaluate whether the calibration strategy affects downstream task accuracy. Table IV reports zero-shot accuracy on four tasks, with mean and standard deviation computed over 5 random seeds.

Mean accuracy differences are negligible (less than 0.2%). However, Medium achieves 3.4–6.1× lower standard devi-

TABLE III
COMPARISON OF FOUR CALIBRATION STRATEGIES (QWEN2-7B, WANDA, 2:4 SPARSITY).

Strategy	PPL	ΔPPL	Rel. Impr.
Dense	7.48	—	—
Easy	14.21	6.73	−14.1%
Random	13.38	5.90	(baseline)
Hard	13.85	6.37	−8.0%
Medium	**11.82**	**4.34**	**+26.4%**

TABLE IV
ZERO-SHOT ACCURACY (MEAN ± STD) (QWEN2-7B, WANDA, 2:4 SPARSITY). VR IS THE STD RATIO OF RANDOM TO MEDIUM.

Task	Random	Medium	VR
ARC-Easy	44.3 ± 0.49	44.1 ± 0.08	6.1×
BoolQ	66.3 ± 0.24	66.2 ± 0.07	3.4×
HellaSwag	52.2 ± 0.35	**52.4** ± 0.09	3.9×
PIQA	74.3 ± 0.28	**74.5** ± 0.06	4.7×
Average	59.3 ± 0.34	59.3 ± 0.08	**4.5**×

ation. Levene's test confirms significance ($p < 0.01$). This indicates more reproducible pruning outcomes.

D. Effect of Calibration Budget

We vary the calibration budget $K \in \{64, 128, 256\}$ to test whether Medium remains effective under different data constraints. Table V reports the results on Qwen2-7B with Wanda under 2:4 sparsity.

Medium consistently outperforms Random across all budget sizes. The perplexity improvements are 1.85, 1.56, and 1.36 for $K = 64$, 128, and 256 respectively. The relative improvement is 28.3% at $K = 64$, 26.4% at $K = 128$, and 27.8% at $K = 256$.

The benefit of Medium is largest at $K = 64$. This suggests that difficulty-based selection is most useful when calibration data is limited. Medium still provides a meaningful improvement at $K = 256$.

V. ANALYSIS

The experiments in Section IV demonstrate that Medium achieves both lower perplexity degradation and reduced variance compared to other strategies. This section investigates the mechanistic basis for these observations through activation statistics analysis.

A. Activation Statistics

Both Wanda and SparseGPT derive weight importance from calibration activations. The choice of calibration data affects the activation distribution and the pruning behavior. We log MLP input activations during calibration forward passes on Qwen2-7B, and summarize them with three statistics.

We summarize activations with three statistics. **Magnitude** (mean ℓ_2-norm) measures overall activation strength; larger values increase $\|X_j\|_2$ and make activations more influential. **Kurtosis** (fourth standardized moment) reflects tail heaviness, indicating selective channel responses. **Mask Overlap** (Jaccard similarity across seeds) measures pruning stability; higher values mean decisions are less sensitive to the specific calibration draw.

TABLE V
EFFECT OF CALIBRATION BUDGET K (QWEN2-7B, WANDA, 2:4 SPARSITY). IMPR. DENOTES RELATIVE IMPROVEMENT OVER RANDOM.

K	Random	Medium	Δ	Impr.
64	14.02	12.17	−1.85	+28.3%
128	13.38	11.82	−1.56	+26.4%
256	12.37	11.01	−1.36	+27.8%

TABLE VI
ACTIVATION STATISTICS ON QWEN2-7B MLP INPUTS.

Strategy	Magnitude	Kurtosis	Mask Overlap
Easy	0.282	5.10	0.70
Random	0.295	5.36	0.74
Hard	0.293	5.20	0.68
Medium	**0.300**	**5.48**	**0.82**

Table VI reports these statistics averaged across layers (with Layer 16 as representative). Medium achieves the highest values on all three metrics. It reaches magnitude 0.300 (vs. 0.295 for Random), kurtosis 5.48 (vs. 5.36), and mask overlap 0.82 (vs. 0.74).

B. Mask Stability Visualization

Fig. 2 visualizes the mask overlap results. Medium achieves 0.82 Jaccard similarity, substantially higher than Random (0.74) and the two extremes (Easy: 0.70, Hard: 0.68). This 8 percentage point absolute improvement in overlap directly explains the 3.4–6.1× reduction in standard deviation on downstream tasks (Table IV).

C. Mechanistic Interpretation

These activation statistics help explain why Medium consistently outperforms the other strategies.

Easy samples under-activate the model. Easy yields the smallest magnitude (0.282). The score $S_{ij} = |W_{ij}| \cdot \|X_j\|_2$ is thus driven largely by $|W_{ij}|$. This reduces activation influence. Its low kurtosis (5.10) offers limited separation between important and unimportant channels.

Hard samples induce unstable masks. Hard has the lowest mask overlap (0.68), meaning small calibration changes lead to noticeably different sparse masks.

Medium balances strength and stability. Medium achieves the highest magnitude (0.300), kurtosis (5.48), and mask overlap (0.82). This indicates stronger, more selective, and more consistent activations. The higher overlap (0.82 vs. 0.74 for Random) directly explains the 3.4–6.1× variance reduction in downstream tasks (Table IV). More consistent masks yield more reproducible predictions. This is valuable for deployment requiring consistent behavior.

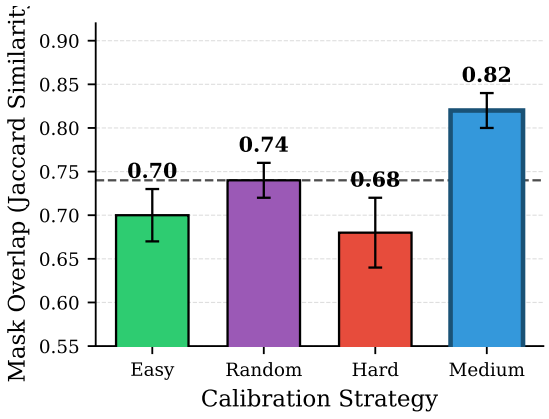

Fig. 2. Sparse mask overlap (Jaccard similarity) across 5 random seeds.

VI. Conclusion

We studied how the difficulty distribution of calibration data affects post-training pruning of LLMs. We used a perplexity-spectrum construction to form Easy, Hard, Medium, and Random calibration sets. All sets were drawn from a shared candidate pool under a fixed budget from the same data source.

Across three 7B models with Wanda and SparseGPT under 2:4 sparsity, Medium difficulty consistently outperforms random selection. It reduces pruning-induced perplexity degradation by 20%–26%. It also reduces the standard deviation by 3.4–6.1$\times$ on downstream tasks. Levene's test is significant ($p < 0.01$). Activation analysis aligns with these gains, showing stronger, more selective activations and more consistent sparse masks for Medium.

Our results suggest a simple addition to calibration pipelines. Candidates are ranked by perplexity, and a median-centered window is selected. This highlights difficulty as a complementary dimension to source and formatting choices. Future work includes testing higher sparsity ratios and larger models. We also plan to analyze attention layers and explore interactions with other calibration strategies such as self-generation and ICL formatting.

References

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

[2] W. Hu, P. Henderson, and J. Cano, "ICE-Pruning: An iterative cost-efficient pruning pipeline for deep neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2025, pp. 1–8.

[3] Z. Tang, P. Li, J. Xiao, and J. Nie, "Pruning neural networks by synchronously changing weight information," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–7.

[4] E. Frantar and D. Alistarh, "SparseGPT: Massive language models can be accurately pruned in one-shot," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR, 2023, pp. 10 323–10 337.

[5] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter, "A simple and effective pruning approach for large language models," in *International Conference on Learning Representations*, 2024.

[6] Y. Zhang *et al.*, "Dynamic sparse no training: Training-free fine-tuning for sparse llms," in *International Conference on Learning Representations (ICLR)*, 2024.

[7] P. Xu *et al.*, "Besa: Pruning large language models with blockwise parameter-efficient sparsity allocation," in *International Conference on Learning Representations (ICLR)*, 2024.

[8] L. Yin *et al.*, "Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR, 2024, pp. 57 101–57 115.

[9] J. Pool, A. Sawarkar, and J. Rodge, "Accelerating inference with sparsity using the nvidia ampere architecture and nvidia tensorrt," https://developer.nvidia.com/blog/accelerating-inference-with-sparsity-using-ampere-and-tensorrt/, Jul. 2021.

[10] M. Williams and N. Aletras, "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)*. Association for Computational Linguistics, 2024, pp. 10 100–10 118.

[11] A. Bandari *et al.*, "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 (EMNLP)*, 2024, pp. 18 089–18 099.

[12] Y. Ji *et al.*, "Beware of calibration data for pruning large language models," 2025, arXiv:2410.17711.

[13] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009, pp. 41–48.

[14] S. Ashkboos, M. L. Croci, M. G. do Nascimento, T. Hoefler, and J. Hensman, "Slicegpt: Compress large language models by deleting rows and columns," in *The Twelfth International Conference on Learning Representations*, 2024.

[15] X. Men *et al.*, "ShortGPT: Layers in large language models are more redundant than you expect," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 20 192–20 204.

[16] L. G. Mugnaini *et al.*, "Efficient llms with amp: Attention heads and mlp pruning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2025.

[17] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[18] J. Oh and D. Oh, "Beyond fixed-length calibration for post-training compression of LLMs," in *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025, pp. 19 355–19 366.

[19] H. Zeng, H. Xu, L. Chen, and K. Yu, "Multilingual brain surgeon: Large language models can be compressed leaving no language behind," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Torino, Italia: ELRA and ICCL, 2024, pp. 11 794–11 812.

[20] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "OPTQ: accurate quantization for generative pre-trained transformers," in *The Eleventh International Conference on Learning Representations*. OpenReview.net, 2023.

[21] J. Lin *et al.*, "AWQ: activation-aware weight quantization for on-device LLM compression and acceleration," in *Proceedings of Machine Learning and Systems*, 2024.

[22] EleutherAI, "lm-evaluation-harness: A framework for few-shot language model evaluation," https://github.com/EleutherAI/lm-evaluation-harness, 2023.