Conformal Risk Control for Sparsity Selection in Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Model pruning can effectively reduce the deployment cost of large language models (LLMs), while selecting an appropriate sparsity level is critical to balancing compression gains against performance preservation. Existing methods typically assess a prespecified sparsity level on the basis of its empirical performance on held-out evaluation data. However, this strategy lacks statistical reliability and may lead to uncontrolled performance degradation in real-world deployment. Therefore, we extend split conformal prediction, which can provide statistically valid guarantees for unseen samples under the exchangeability assumption, to pruning sparsity selection. Meanwhile, we propose CRCSS, a performance-degradation risk control framework for sparsity selection in LLM pruning. Specifically, we transform non-monotonic performance degradation trajectories into nondecreasing degradation prefix to determine sample-specific maximum safe sparsity boundaries, thereby formulating pruning sparsity selection as a conformal calibration problem. Extensive experiments across multiple LLMs, pruning methods and language understanding benchmarks demonstrate that CRCSS achieves empirical guarantee rates consistent with the prespecified reliability targets, while enabling a controllable trade-off between reliability and compression.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in tasks such as language understanding and text generation, much of this progress driven by the continued growth of model scale and training computation (Brown et al., 2020; Hoffmann et al., 2022). However, scaling model size also incurs substantial memory consumption, computational overhead, and hardware requirements, increasing the deployment cost of modern LLMs and limiting their widespread adoption in resource-constrained environments (Zhu et al., 2024). Thus, improving inference and deployment efficiency while preserving the general capabilities of pretrained LLMs as much as possible has become an important research direction.

As research in this area has progressed, model compression techniques, such as quantization (Xiao et al., 2023), knowledge distillation (Gu et al., 2024), and network pruning, have been extensively developed (Zhu et al., 2024). Among them, network pruning, which removes redundant parameters or structures from pretrained models, has become an important approach to compressing LLMs. Based on the granularity of the removed units, existing methods can be categorized into unstructured, semi-structured, and structured pruning. Unstructured pruning operates at the level of individual weights and can achieve high sparsity without additional fine-tuning by exploiting activation statistics (Sun et al., 2024), Hessian information, or local reconstruction objectives (Frantar & Alistarh, 2023). Semi-structured pruning retains N nonzero weights in each group of M (i.e., $N{:}M$ sparsity), balancing the flexibility of fine-grained pruning with hardware efficiency (Zhou et al., 2021). Structured pruning directly removes entire structures, including neurons, channels, attention heads, or model dimensions, to achieve more hardware-friendly compression (Ma et al., 2023; Ashkboos et al., 2024). Building on these approaches, recent studies have further improved model performance at high sparsity levels through layer-wise sensitivity analysis and non-uniform sparsity allocation (Yin et al., 2024; Bai et al., 2024; Xu et al., 2024). Collectively, these studies have substantially advanced the compression efficiency, performance preservation,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

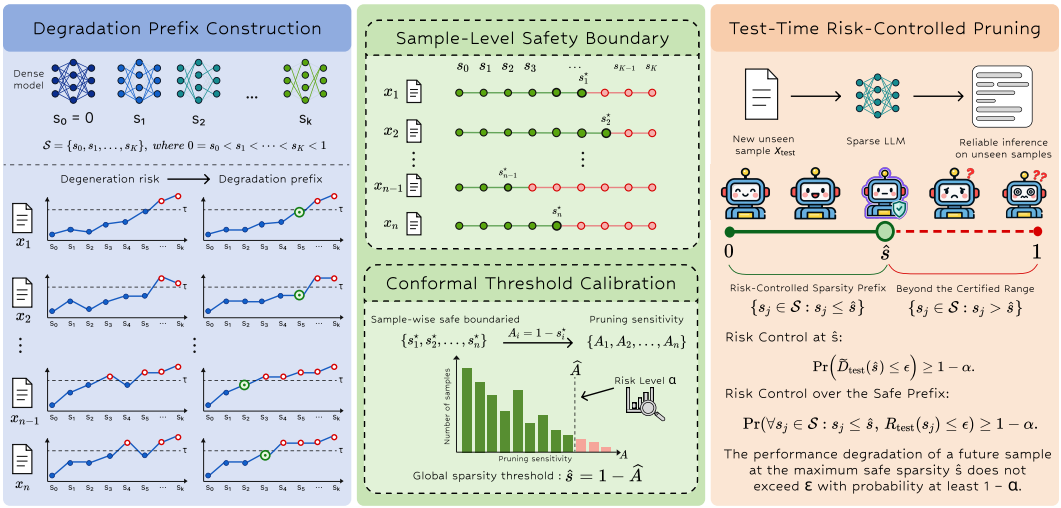


Figure 1: The overview of the proposed conformal risk-controlled sparsity selection framework, CRCSS, for LLM pruning.

and hardware applicability of LLM pruning. However, they primarily focus on how to prune and leave a critical question unanswered: what sparsity level should be selected along the pruning path to satisfy the desired performance degradation requirements on unseen data?

Answering this question requires reframing sparsity selection from a model configuration problem based on empirical performance into a risk-control problem with statistical guarantees. Toward this goal, existing research has mainly explored two categories of statistical frameworks. The first is the Learn-then-Test (LTT) framework (Angelopoulos et al., 2025). Alvarez (Alvarez, 2024), which formulates candidate sparsity levels as hypotheses concerning population expected risk, computes valid p-values using calibration data, and applies multiple testing with family-wise error rate control to select sparsity levels whose risks do not exceed a given threshold. However, LTT mainly controls the population expected risk at each candidate sparsity level and does not characterize the heterogeneous degradation boundaries of individual samples along the pruning path. The second category is based on conformal prediction, which provides finite-sample guarantees under the assumption of exchangeability (Angelopoulos et al., 2024; Angelopoulos & Bates, 2023) and has been used to calibrate the reliability of LLM outputs (Quach et al., 2024; Wang et al., 2025; 2026; Mohri & Hashimoto, 2024). Unfortunately, these conformal prediction methods (Zhao & Bellotti, 2022; Zhao et al., 2025) focus on evaluating the coverage validity and set efficiency of prediction sets produced by pruned models, rather than directly calibrating a sparsity level that satisfies a performance degradation constraint. Consequently, existing research has yet to address how to calibrate the entire candidate pruning path based on sample-level performance degradation and provide a finite-sample marginal guarantee for the maximum acceptable sparsity level.

To bridge the aforementioned gap, we propose a novel **C**onformal **R**isk **C**ontrol framework for **S**parsity **S**election, namely CRCSS, in large language model pruning. For clarity, we provide the overview of the proposed framework in Figure 1. Specifically, CRCSS consists of three major stages: (i) converting the original sample-level performance degradation into a nondecreasing degradation prefix; (ii) determining the sample-level maximum safe boundary; (iii) calibrating the threshold. Consequently, under the exchangeability assumption, CRCSS provides a finite-sample marginal guarantee for unseen samples and ensures that the entire candidate sparsity prefix from the dense model to the selected threshold satisfies the prespecified performance degradation constraint.

To evaluate the statistical validity and applicability of CRCSS, we conduct experiments on three language understanding benchmarks—CommonsenseQA for closed-ended commonsense question answering (Talmor et al., 2019), TriviaQA (Joshi et al., 2017) for open-domain

question answering, and MMLU for multi-domain knowledge evaluation (Hendrycks et al., 2021)—covering five LLMs and three widely used pruning methods. The experimental results demonstrate that CRCSS can select the maximum safe sparsity level under a user-specified performance degradation tolerance while satisfying the prespecified empirical reliability target. Moreover, because the prefix maximum risk is monotonically non-decreasing with respect to sparsity, all candidate sparsity levels within the selected boundary satisfy the same prefix safety criterion, thereby enabling flexible and safe sparsity selection for deployment under different resource constraints.

Our main contributions are summarized as follows:

- We propose CRCSS, a conformal risk calibration framework for LLM pruning, to determine the maximum safe prefix sparsity under a prespecified degradation tolerance and risk level.
- We develop a degradation prefix calibration strategy for non-monotonic degradation trajectories, and establish a finite-sample marginal guarantee for the entire candidate sparsity prefix within the selected boundary.
- Extensive experiments demonstrate that CRCSS achieves statistical validity while exhibiting a stable trade-off between reliability and compression.

2 Related Work

2.1 LLM Pruning

Network pruning is an important model compression technique for reducing model storage and computational costs and has been widely applied to large language models in recent years (Zhu et al., 2024; An et al., 2024). Based on their sparsity patterns, existing LLM pruning methods can be broadly categorized into three groups: unstructured (Zhang et al., 2024b), semi-structured (Zhang et al., 2024a) and structured (Li et al., 2024a). Unstructured pruning directly removes individual weights. For example, SparseGPT achieves high sparsity through approximate second-order information and layer-wise weight reconstruction (Frantar & Alistarh, 2023), while Wanda measures parameter importance using weight magnitudes and input activation norms (Sun et al., 2024). Semi-structured pruning retains N nonzero weights within each group of M. For instance, MaskLLM models $N:M$ masks as learnable distributions to optimize semi-structured sparse models (Fang et al., 2024). Structured pruning methods directly reduce model width or depth. For example, LLM-Pruner (Ma et al., 2023) and SliceGPT (Ashkboos et al., 2024) compress models by removing complete structures or model dimensions.

In addition to sparsity patterns based methods, recent studies including OWL (Yin et al., 2024), SparseLLM (Bai et al., 2024) and BESA (Xu et al., 2024) further adaptively allocate the pruning load according to inter-layer outlier distributions, global parameter dependencies, or block-wise reconstruction errors, thereby improving performance preservation at high sparsity levels. Although these methods have substantially enhanced the performance preservation and compression efficiency of LLM pruning, they do not employ statistical calibration to satisfy user-specified performance risk constraints on unseen data, which often limits their reliability in high-risk scenarios.

2.2 Split Conformal Prediction

Split conformal prediction provides finite-sample marginal coverage guarantees for unseen samples under the exchangeability assumption (Angelopoulos & Bates, 2023). In recent years, it has been widely used to calibrate LLM-generated outputs (Li et al., 2024b; Kang et al., 2024; Jiang et al., 2025). Conformal Language Modeling constructs reliable candidate sets by calibrating sampling-based stopping and rejection rules (Quach et al., 2024). Another study (Su et al., 2024) exploits sampling frequencies and semantic similarity to guarantee prediction-set coverage without requiring access to logits for black-box LLMs. Subsequent studies have extended conformal methods to factuality calibration (Mohri & Hashimoto, 2024; Cherian et al., 2024), multimodal risk control (Wang et al., 2025), and selective question

answering (Wang et al., 2026). Furthermore, conformal risk control generalizes classical coverage guarantees to monotone bounded losses, thereby expanding the range of risks that can be controlled (Angelopoulos et al., 2024).

Despite these advances, the application of conformal prediction to model pruning is very scarce. To our knowledge, one CP-efficiency-based pruning method (Zhao & Bellotti, 2022) is the only prior approach that directly integrates conformal prediction into neural network pruning. It removes neural units according to their effect on conformal prediction-set efficiency, but does not calibrate the candidate pruning path based on sample-level performance degradation. Therefore, it remains an insufficiently explored problem to determine the maximum safe sparsity level with statistical guarantees under a prespecified performance degradation tolerance.

3 Method

3.1 Problem Formulation

Given a pretrained large language model, we consider a risk calibration set $\mathcal{D}_{\text{cal}} = \{Z_i = (x_i, y_i)\}_{i=1}^{n}$ that is independent of both the model training data and the pruning calibration data, where x_i denotes the i-th task input and y_i denotes its corresponding ground-truth label or reference answer. Let $Z_{\text{test}} = (x_{\text{test}}, y_{\text{test}})$ denote a future test sample, and assume that the calibration samples $Z_1, \ldots, Z_n$ and Z_{test} are exchangeable (Angelopoulos & Bates, 2023). We further consider a finite candidate grid $\mathcal{S} = \{s_0, s_1, \ldots, s_K\}$ ordered by increasing sparsity, where $0 = s_0 < s_1 < \cdots < s_K < 1$, and $s_0 = 0$ corresponds to the original dense model. For any sample Z_i and candidate sparsity level s, we use a scalar function $M(x_i, y_i, s)$ to characterize the model's performance on that sample and define the sample-level performance degradation as:

$$R_i(s) = M(x_i, y_i, s) - M(x_i, y_i, 0), \tag{1}$$

where $M(\cdot)$ corresponds to sample-level performance metrics for which a larger value indicates worse performance. Given a user-specified performance-degradation tolerance $\epsilon \geq 0$, we seek to determine how far the model can be safely pruned via statistical calibration, such that the performance degradation of a future sample at the selected sparsity does not exceed ϵ with probability at least $1 - \alpha$, i.e.,

$$\Pr(R(s) \leq \epsilon) \geq 1 - \alpha, \tag{2}$$

where $\alpha \in (0, 1)$ denotes the allowable miscoverage level, and $1 - \alpha$ corresponds to the target reliability level.

Beyond a single sparsity level, we further consider the entire candidate prefix from the dense model to the selected sparsity. Accordingly, we require this prefix to satisfy the same performance-degradation tolerance:

$$\Pr(V(Z, s) = 0) \geq 1 - \alpha, \tag{3}$$

where $V(\cdot)$ is an indicator of prefix safety: $V(\cdot) = 0$ if the performance degradation at every sparsity level in the candidate set does not exceed ϵ; otherwise $V(\cdot) = 1$. Consequently, a calibrated maximum safe sparsity simultaneously determines a set of usable candidate sparsity levels preceding it, providing flexibility for model deployment under different resource constraints.

3.2 Risk-Constrained Sparsity Selection

Degradation Prefix Construction. For a calibration sample Z_i, its performance-degradation trajectory over the candidate sparsity levels can be obtained along the predefined sparsity grid. Given a tolerance ϵ, a direct approach is to regard candidate sparsity levels satisfying the degradation constraint as acceptable and select the largest one:

$$s_i = \max\{s_k \in \mathcal{S} : R_i(s_k) \leq \epsilon\}. \tag{4}$$

However, sample-level performance degradation of an LLM along a discrete pruning path is generally non-monotonic. Therefore, a larger s_i only indicates that a higher-sparsity point may satisfy the constraint again and does not imply that all preceding candidate points are safe. More specifically, because local risk peaks may occur, the event $s_i \geq s_j$ is not equivalent to $R_i(s_j) \leq \epsilon$. Consequently, directly performing quantile calibration on s_i cannot guarantee that the entire candidate prefix from the dense model to that sparsity satisfies the constraint.

To address this issue, we adopt a stricter definition of prefix safety. For a candidate sparsity s_k, we define the degradation prefix of sample Z_i as:

$$\widetilde{D}_i(s_k) = \max_{0 \leq j \leq k} R_i(s_j). \tag{5}$$

This quantity represents the maximum performance degradation encountered over the entire candidate pruning prefix from s_0 to s_k. As k increases, the candidate set over which the maximum is taken continually expands; therefore, the degradation prefix is naturally non-decreasing with sparsity. This construction converts "pointwise acceptability" into a prefix-level criterion, which requires every candidate sparsity up to the selected level to be acceptable, without imposing additional monotonicity assumptions on the original performance-degradation trajectory.

Sample-Level Safety Boundary. Building on the monotonic structure of the degradation prefix, we define the maximum safe prefix sparsity for sample Z_i as:

$$s_i^\star = \max\{s_k \in \mathcal{S} : \widetilde{D}_i(s_k) \leq \epsilon\}. \tag{6}$$

Thus, $s_i^\star$ denotes the maximum candidate sparsity at which the degradation prefix of sample i does not exceed ϵ. For the calibration set $\mathcal{D}_{\text{cal}}$, this yields a collection of sample-level safety boundaries $\{s_1^\star, \ldots, s_n^\star\}$, which captures the heterogeneous sensitivity of different samples to pruning.

To align the sample-level safety boundary with the standard SCP convention that a larger score indicates higher risk, we define the pruning sensitivity score as follows:

$$A_i = 1 - s_i^\star. \tag{7}$$

Thus, safe sparsity selection is thereby reduced to quantile calibration of the pruning sensitivity scores $\{A_1, \ldots, A_n\}$.

Conformal Threshold Calibration. We then perform quantile calibration to obtain an upper bound on the pruning sensitivity of a future sample and map the resulting threshold back to the sparsity space. The following proposition provides a finite-sample guarantee for the safe sparsity prefix obtained through this procedure.

Proposition 1. ***(Finite-Sample Guarantee for the Safe Sparsity Prefix).*** *Assume that the calibration samples $Z_1, \ldots, Z_n$ and a future test sample Z_{test} are exchangeable conditional on a fixed candidate pruning path. Let $A_1, \ldots, A_n, A_{\text{test}}$ denote their corresponding pruning sensitivity scores, and let $\widehat{A}$ be the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile defined as: $\widehat{A} = A_{\lceil (n+1)(1-\alpha) \rceil}$. Let $\widehat{s} = 1 - \widehat{A}$, and then we have:*

$$\Pr\Big(R_{\text{test}}(\hat{s}) \leq \widetilde{D}_{\text{test}}(\hat{s}) \leq \epsilon\Big) \geq 1 - \alpha. \tag{8}$$

The complete proof of Proposition 1 is provided in Appendix A. Moreover, because the prefix degradation risk is monotonically non-decreasing with respect to sparsity, the above result can be further extended to the entire candidate sparsity prefix within the selected boundary:

$$\begin{aligned} &\Pr\Big(\forall s_j \in \mathcal{S} : s_j \leq \hat{s},\, \widetilde{D}_{\text{test}}(s_j) \leq \epsilon\Big) \geq 1 - \alpha \\ &\quad \Longrightarrow \Pr(\forall s_j \in \mathcal{S} : s_j \leq \hat{s},\, R_{\text{test}}(s_j) \leq \epsilon) \geq 1 - \alpha. \end{aligned} \tag{9}$$

Therefore, CRCSS guarantees that the performance degradation of a future sample at the maximum safe sparsity $\hat{s}$ does not exceed ϵ with probability at least $1 - \alpha$. Meanwhile, by the definition of the degradation prefix, the entire candidate sparsity prefix from the dense model to $\hat{s}$ satisfies the same performance-degradation constraint.

4 Experiments

4.1 Experimental Setup

Models and Datasets. We consider five LLMs spanning different model families and scales, including Qwen-2.5-3B-Instruct, Qwen-2.5-7B-Instruct (Yang et al., 2024), LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), Vicuna-13B-v1.5 (Zheng et al., 2023), and Qwen-2.5-14B-Instruct. We evaluate CRCSS on three widely used language understanding benchmarks: CommonsenseQA (CQA) (Talmor et al., 2019), TriviaQA (Joshi et al., 2017), and MMLU (Hendrycks et al., 2021). Additional details are provided in Appendix B.1.

Pruning Methods. To evaluate the applicability of CRCSS across different underlying pruning algorithms, we consider three representative post-training pruning methods: Wanda (Sun et al., 2024), SparseGPT (Frantar & Alistarh, 2023), and OWL (Yin et al., 2024). Following standard post-training pruning settings, we adopt C4 (Raffel et al., 2020) as the pruning calibration corpus for methods that require calibration data. For each target sparsity level, we sample 128 sequences from C4 using a fixed random seed and truncate each sequence to 2,048 tokens.

Evaluation Metrics. We adopt the Empirical Violation Rate (EVR) to evaluate the statistical validity of CRCSS, with its theoretical guarantee targeting a population-level miscoverage probability of at most α. To parallel the notion of efficiency in conformal prediction, we further evaluate compression efficiency, measured by the maximum safe prefix sparsity level selected subject to the risk-control constraint.

Calibration Protocol. For each combination of model, pruning method, and risk level, we independently repeat the complete conformal calibration and testing procedure 100 times using different random data splits. By default, the calibration and test sets are split at a ratio of 1:1. The candidate sparsity grid is set to $\{0, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6, 0.7\}$. Additional experimental details are provided in Appendix B.2 and Appendix B.3.

4.2 Statistical Validity

We first verify the statistical validity of CRCSS, i.e., whether it can control the probability that an unseen sample violates the performance degradation constraint at or below α. On CommonsenseQA and TriviaQA, we utilize the number of attempts required to generate the first correct answer as the sample-level performance metric. We set the maximum number of attempts to 10, so as to standardize the inference budget and prevent persistently incorrect samples from producing unbounded counts. Moreover, because the number of attempts is a discrete metric with a minimum nonzero change of 1, we set the performance degradation tolerance to $\epsilon = 1$. This setting accommodates minor fluctuations arising from the sampling process while preserving sensitivity to further performance degradation.

Figure 2 shows the experimental results, where each data point represents the mean EVR over 100 repeated experiments, and the dashed line denotes the target risk level α. For both datasets, the mean EVR remains at or below the corresponding target level across all evaluated combinations of models and pruning methods, and increases accordingly as α rises from 0.05 to 0.30. For instance, for Qwen-2.5-3B-Instruct on TriviaQA, the mean EVR increases from 0.0219 to 0.2593 as α increases from 0.05 to 0.30. Moreover, consistent trends are observed across different models and pruning methods on both datasets, further indicating that the risk-control behavior of CRCSS is not driven by any particular dataset, model, or pruning method. Results of additional pruning methods are provided in Appendix C.2.

4.3 Compression Efficiency

After validating its statistical validity, we further evaluate the compression efficiency of CRCSS, defined as the maximum safe prefix sparsity level that can be selected while satisfying the prespecified risk constraint. Figure 3 presents the selected sparsity levels under different configurations. As we can see, when α increases, the selected sparsity generally rises, indicating that moderately relaxing the risk constraint enables a higher compression ratio. This trend remains consistent across different models, datasets, and pruning methods, further

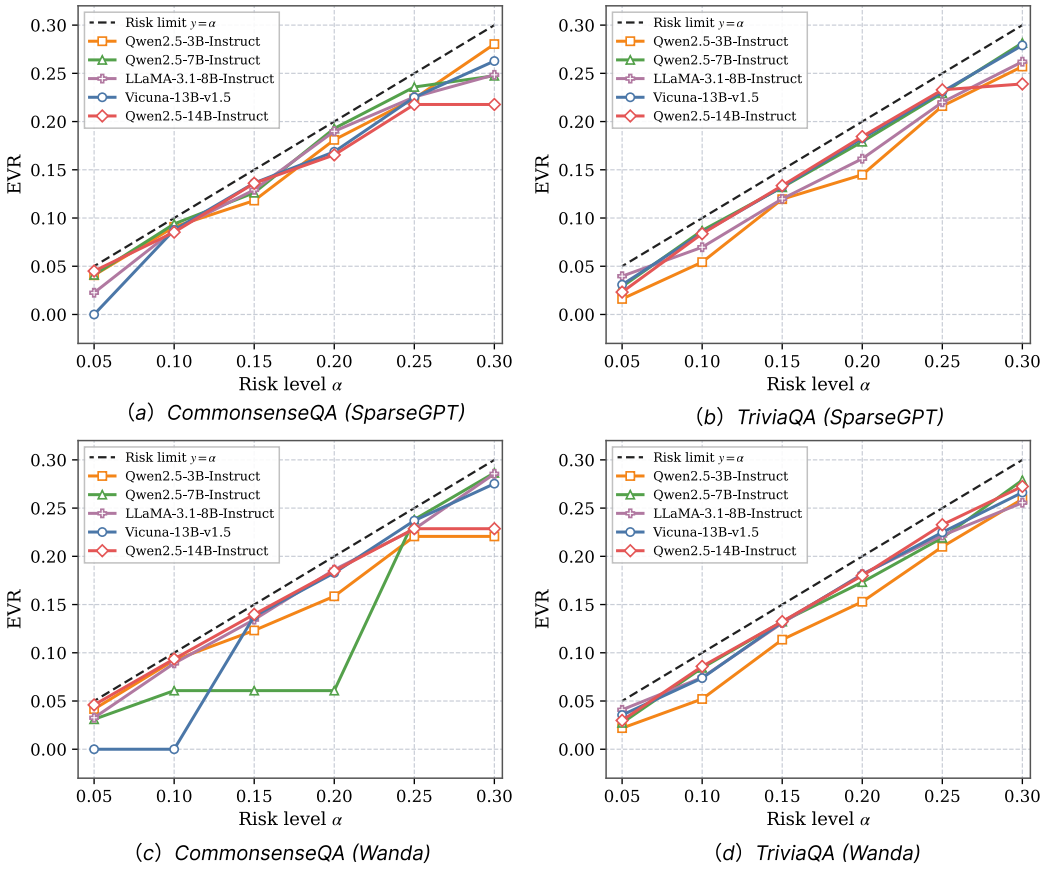


Figure 2: Empirical violation rate (EVR) of CRCSS on CommonsenseQA and TriviaQA with five LLMs and two pruning methods (SparseGPT and Wanda) under different risk levels.

demonstrating that CRCSS can determine an appropriate safe sparsity level for each specific configuration and target risk level.

Additionally, to evaluate the compression advantage of CRCSS over an existing risk-control framework, we follow the risk-controlled pruning setting in (Alvarez, 2024) and compare CRCSS with Learn-then-Test (LTT) (Angelopoulos et al., 2025). LTT treats candidate sparsity levels as hypotheses to be tested and utilizes statistical tests to select pruning levels that satisfy the risk constraint. In our experiments, the confidence parameter of LTT is set to $\delta = 0.01$, while all other settings are kept identical to those of CRCSS. The results on TriviaQA are presented in Figure 3. Across all 36 configurations, CRCSS selects a higher sparsity level than LTT. These results demonstrate that, while satisfying the target empirical reliability, CRCSS can select higher safe sparsity levels than LTT, thereby enabling more effective model compression.Additional experimental results are provided in Appendix C.3.

4.4 Generalization to Different Risk Metrics

To evaluate the applicability of CRCSS to different performance degradation metrics, we further use confidence drop as the risk metric on MMLU. To impose a relatively stringent performance degradation constraint, we set the degradation tolerance to $\epsilon = 0.05$. The EVR results are presented in Figure 4. Across all evaluated models and pruning methods, the mean EVR does not exceed the corresponding target level α, demonstrating that CRCSS continues to provide effective risk control under the confidence-drop metric. Additional experimental results are provided in Appendix C.4.

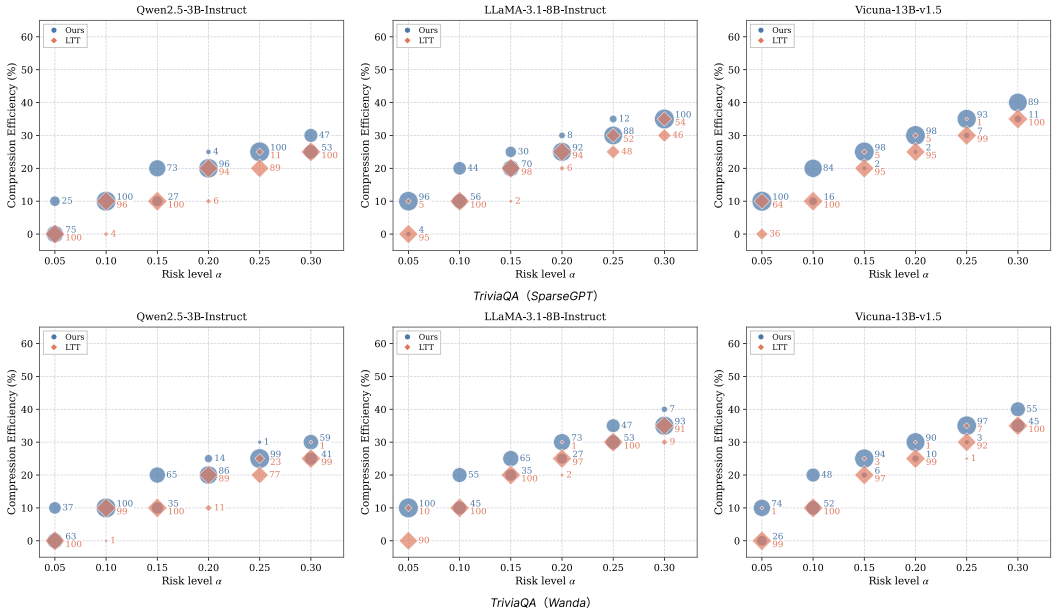


Figure 3: Compression efficiency comparison between CRCSS and LTT on TriviaQA with Qwen-2.5-3B-Instruct, LLaMA-3.1-8B-Instruct, and Vicuna-13B-v1.5.

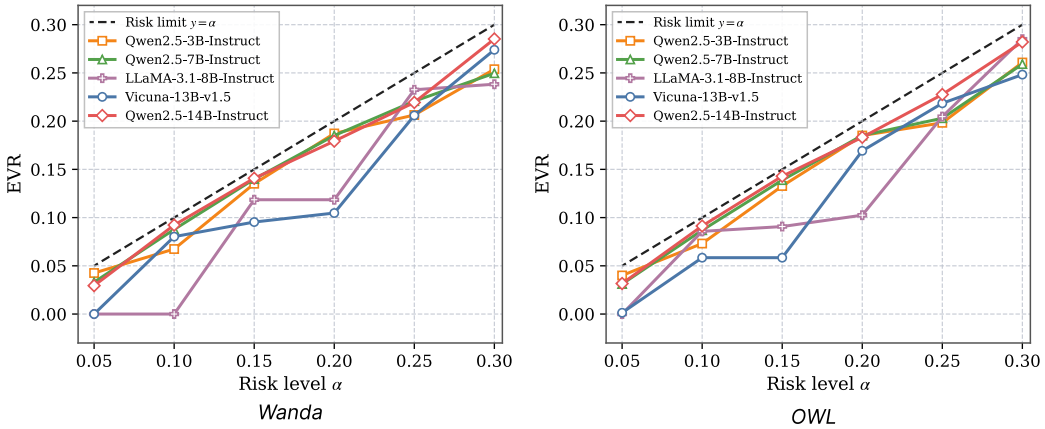


Figure 4: EVR results of CRCSS on MMLU under confidence degradation tolerance $\epsilon = 0.05$ across five LLMs and different risk levels.

Notably, the risk metric considered here and that used in Section 4.2 differ substantially in form: the number of answer attempts measures discrete degradation in task performance, whereas confidence drop captures continuous changes in predictive confidence. CRCSS accommodates both types of sample-level risk without modifying its calibration procedure, indicating that it does not depend on any particular performance metric and can calibrate the corresponding safe sparsity boundary for a user-defined performance degradation metric.

4.5 Sensitivity to Calibration Size

In practical applications, large-scale labeled calibration data may be difficult to obtain. To investigate the sensitivity of CRCSS to the number of calibration samples, we adjust the default calibration-to-test split ratio from 5:5 to 3:7, 2:8, and 1:9, and repeat each configuration 100 times.

Table 1 reports the mean EVRs under different data split ratios. Across all evaluated risk levels, models, and pruning configurations, the mean EVRs under the 3:7, 2:8, and 1:9 splits

Table 1: Robustness evaluation of CRCSS with Wanda under different calibration:test split ratios on CommonsenseQA and TriviaQA.

Dataset	Model	Risk level α					
		0.05	0.10	0.15	0.20	0.25	0.30
Calibration:test = 3:7							
CommonsenseQA	Qwen-2.5-3B-Instruct	0.0415	0.0890	0.1245	0.1588	0.2208	0.2208
	Qwen-2.5-7B-Instruct	0.0310	0.0606	0.0606	0.0606	0.2165	0.2881
	LLaMA-3.1-8B-Instruct	0.0248	0.0893	0.1354	0.1858	0.2282	0.2746
	Vicuna-13B-v1.5	0.0000	0.0000	0.1345	0.1837	0.2358	0.2781
	Qwen-2.5-14B-Instruct	0.0457	0.0929	0.1390	0.1849	0.2287	0.2287
TriviaQA	Qwen-2.5-3B-Instruct	0.0199	0.0521	0.1079	0.1618	0.2104	0.2503
	Qwen-2.5-7B-Instruct	0.0278	0.0847	0.1358	0.1808	0.2265	0.2732
	LLaMA-3.1-8B-Instruct	0.0373	0.0705	0.1268	0.1755	0.2210	0.2671
	Vicuna-13B-v1.5	0.0253	0.0729	0.1292	0.1771	0.2236	0.2619
	Qwen-2.5-14B-Instruct	0.0300	0.0844	0.1299	0.1766	0.2264	0.2718
Calibration:test = 2:8							
CommonsenseQA	Qwen-2.5-3B-Instruct	0.0409	0.0882	0.1291	0.1596	0.2209	0.2209
	Qwen-2.5-7B-Instruct	0.0319	0.0605	0.0605	0.0605	0.1944	0.2877
	LLaMA-3.1-8B-Instruct	0.0211	0.0886	0.1371	0.1843	0.2277	0.2691
	Vicuna-13B-v1.5	0.0000	0.0011	0.1337	0.1847	0.2366	0.2821
	Qwen-2.5-14B-Instruct	0.0438	0.0909	0.1350	0.1841	0.2283	0.2288
TriviaQA	Qwen-2.5-3B-Instruct	0.0152	0.0516	0.0899	0.1571	0.2067	0.2501
	Qwen-2.5-7B-Instruct	0.0269	0.0797	0.1338	0.1765	0.2232	0.2679
	LLaMA-3.1-8B-Instruct	0.0323	0.0662	0.1235	0.1741	0.2207	0.2689
	Vicuna-13B-v1.5	0.0232	0.0694	0.1265	0.1771	0.2218	0.2614
	Qwen-2.5-14B-Instruct	0.0324	0.0839	0.1288	0.1770	0.2250	0.2685
Calibration:test = 1:9							
CommonsenseQA	Qwen-2.5-3B-Instruct	0.0337	0.0606	0.0606	0.0606	0.1950	0.2883
	Qwen-2.5-7B-Instruct	0.0421	0.0899	0.1356	0.1816	0.2279	0.2288
	LLaMA-3.1-8B-Instruct	0.0000	0.0090	0.1298	0.1842	0.2326	0.2820
	Vicuna-13B-v1.5	0.0372	0.0884	0.1316	0.1607	0.2202	0.2208
	Qwen-2.5-14B-Instruct	0.0225	0.0866	0.1368	0.1832	0.2254	0.2254
TriviaQA	Qwen-2.5-3B-Instruct	0.0138	0.0517	0.0976	0.1600	0.2039	0.2417
	Qwen-2.5-7B-Instruct	0.0274	0.0776	0.1294	0.1680	0.2195	0.2627
	LLaMA-3.1-8B-Instruct	0.0269	0.0684	0.1205	0.1720	0.2192	0.2693
	Vicuna-13B-v1.5	0.0213	0.0714	0.1143	0.1698	0.2169	0.2586
	Qwen-2.5-14B-Instruct	0.0300	0.0828	0.1255	0.1800	0.2223	0.2667

remain below the target level α, demonstrating that CRCSS maintains effective risk control even when the number of calibration samples is substantially reduced. The corresponding results for the maximum safe prefix sparsity levels are provided in Appendix C.5.

5 Conclusion

In this work, we propose CRCSS, a conformal sparsity selection framework for controlling performance degradation in LLM pruning while providing a finite-sample marginal guarantee. CRCSS comprises three stages: degradation-prefix construction converts non-monotonic degradation trajectories into nondecreasing risks; sample-level safe-boundary identification characterizes heterogeneous sensitivity to pruning; and conformal threshold calibration determines a global sparsity threshold, which controls the entire candidate pruning prefix at the target risk level. Experiments across three language understanding benchmarks, five LLMs, three pruning methods, and two types of performance-degradation metrics demonstrate that CRCSS keeps the EVR within the target level while enabling an effective trade-off between reliability and compression efficiency. Notably, the current framework is limited by the exchangeability assumption and calibrates only one performance risk at a time. Therefore, in future work, we will extend CRCSS to jointly handle distribution shifts and control multiple performance risks.

AI use statement

In this work, we used generative AI tools to assist with writing and language polishing, literature search and the identification of relevant prior work, and the verification of mathematical proofs. All AI-assisted text was carefully reviewed and revised by the authors. Literature recommended by generative AI tools was verified against the original sources before being cited. Suggestions concerning the rigor of mathematical derivations were independently examined by the authors and verified for correctness.

We take responsibility for the final content of this work, including all AI-assisted text, literature found through AI-assisted searches, and mathematical arguments.

Reproducibility Statement

We have taken several steps to support the reproducibility of our work. The experimental settings, including the datasets, models, pruning methods, candidate sparsity levels, risk metrics, and calibration procedures, are described in Section 4 and further detailed in the appendix. The appendix also provides additional experimental results and the complete proof of Proposition 1, together with the assumptions required for the theoretical guarantee. All datasets and pretrained models used in our experiments are publicly available, and the corresponding sources are cited in the paper.

References

J. Alvarez. Confident magnitude-based neural network pruning. *arXiv preprint arXiv:2408.04759*, 2024.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 10, pp. 10865–10873, 2024.

A. N. Angelopoulos and S. Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.

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 *International Conference on Learning Representations*, 2024.

G. Bai, Y. Li, C. Ling, K. Kim, and L. Zhao. SparseLLM: Towards global pruning of pre-trained language models. In *Advances in Neural Information Processing Systems*, 2024.

Tom B. Brown, Benjamin Mann, Ryder, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pp. 1877–1901, 2020.

J. J. Cherian, I. Gibbs, and E. J. Candès. Large language model validity via enhanced conformal prediction methods. In *Advances in Neural Information Processing Systems*, 2024.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5050–5063. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.276.

Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. MaskLLM: Learnable semi-structured sparsity for large language models. In *Advances in Neural Information Processing Systems*, 2024.

E. Frantar and D. Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the International Conference on Machine Learning*, 2023.

A. Grattafiori, A. Dubey, A. Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, 2022.

Zhengping Jiang, Anqi Liu, and Benjamin Van Durme. Conformal linguistic calibration: Trading-off between factuality and specificity. In *Advances in Neural Information Processing Systems*, 2025.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017.

Mintong Kang, Nezihe Merve Gürel, Ning Yu, Dawn Song, and Bo Li. C-rag: Certified generation risks for retrieval-augmented language models. In *Proceedings of the International Conference on Machine Learning*, pp. 22963–23000, 2024.

Guangyan Li, Yongqiang Tang, and Wensheng Zhang. Lorap: Transformer sub-layers deserve differentiated structured compression for large language models. In *Proceedings of the International Conference on Machine Learning*, pp. 28657–28672, 2024a.

Shuo Li, Sangdon Park, Insup Lee, and Osbert Bastani. Traq: Trustworthy retrieval augmented question answering via conformal prediction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3799–3821. Association for Computational Linguistics, 2024b.

X. Ma, G. Fang, and X. Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, 2023.

C. Mohri and T. Hashimoto. Language models with conformal factuality guarantees. In *Proceedings of the International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 36029–36047, 2024.

V. Quach, A. Fisch, T. Schuster, A. Yala, J. H. Sohn, T. Jaakkola, and R. Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

J. Su, J. Luo, H. Wang, and L. Cheng. API is enough: Conformal prediction for large language models without logit-access. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 979–995, 2024.

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.

A. Talmor, J. Herzig, N. Lourie, and J. Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, pp. 4149–4158, 2019.

Q. Wang, T. Geng, Z. Wang, T. Wang, B. Fu, and F. Zheng. Sample then identify: A general framework for risk control and assessment in multimodal large language models. In *International Conference on Learning Representations*, 2025.

Z. Wang, J. Duan, Q. Wang, X. Zhu, T. Chen, X. Shi, and K. Xu. COIN: Uncertainty-guarding selective question answering for foundation models with provable risk guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 40, pp. 33764–33772, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the International Conference on Machine Learning*, pp. 38087–38099, 2023.

P. Xu, W. Shao, M. Chen, S. Tang, K. Zhang, P. Gao, F. An, Y. Qiao, and P. Luo. BESA: Pruning large language models with blockwise parameter-efficient sparsity allocation. In *International Conference on Learning Representations*, 2024.

A. Yang, B. Yang, B. Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *Proceedings of the International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 57101–57115, 2024.

Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. Plug-and-play: An efficient post-training pruning method for large language models. In *International Conference on Learning Representations*, 2024a.

Yuxin Zhang, Lirui Zhao, Mingbao Lin, Yunyun Sun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. Dynamic sparse no training: Training-free fine-tuning for sparse llms. In *International Conference on Learning Representations*, 2024b.

X. Zhao and A. Bellotti. Pruning neural networks for inductive conformal prediction. In *Proceedings of the Symposium on Conformal and Probabilistic Prediction with Applications*, Proceedings of Machine Learning Research, pp. 273–293, 2022.

X. Zhao, A. Farjudian, and A. Bellotti. Pruning convolutional neural networks for inductive conformal prediction. *Neurocomputing*, 611:128704, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, 2023. Datasets and Benchmarks Track.

Aojun Zhou, Yukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. Learning n:m fine-grained structured sparse neural networks from scratch. In *International Conference on Learning Representations*, 2021.

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

A Proof of Proposition 1

For each sample in the calibration set, we obtain a set of scores $A_1, A_2, \ldots, A_n$ according to Eq. equation 7. We sort these $A_{i=1,\ldots,n}$ values in ascending order to obtain $A_{(i=1,\ldots,n)}$, and then compute the empirical quantile as

$$\hat{A} = \inf\left\{ A : \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{A_i \le A\} \ge \frac{\lceil (n+1)(1-\alpha) \rceil}{n} \right\} = A_{(\lceil (n+1)(1-\alpha) \rceil)}. \tag{10}$$

Since $(Z_1, Z_2, \ldots, Z_n)$ satisfy the exchangeability assumption, we have

$$\Pr\left(A_{\text{test}} \le A_{(i)}\right) \ge \frac{i}{n+1}. \tag{11}$$

According to the definition of the maximum safe prefix sparsity in Eq. equation 6, the following two events are equivalent:

$$\left\{ \widetilde{D}_{\text{test}}(\hat{s}) \le \epsilon \right\} = \left\{ s_{\text{test}} \ge \hat{s} \right\}. \tag{12}$$

Moreover, according to the transformation of the pruning sensitivity score in Eq. equation 7, we have

$$\left\{ s_{\text{test}} \ge \hat{s} \right\} = \left\{ A_{\text{test}} \le \hat{A} \right\}. \tag{13}$$

Consequently, at the $1-\alpha$ confidence level, we have

$$\begin{aligned} \Pr\left(\widetilde{D}_{\text{test}}(\hat{s}) \le \epsilon \right) &= \Pr\left(s_{\text{test}} \ge \hat{s} \right) \\ &= \Pr\left(A_{\text{test}} \le \hat{A} \right) \\ &= \Pr\left(A_{\text{test}} \le A_{(\lceil (n+1)(1-\alpha) \rceil)} \right) \\ &\ge \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1} \\ &\ge 1-\alpha. \end{aligned} \tag{14}$$

Furthermore, the degradation prefix constructed in Eq. equation 5 satisfies $R_{\text{test}}(\hat{s}) \le \widetilde{D}_{\text{test}}(\hat{s})$. It therefore follows that

$$\Pr\left(R_{\text{test}}(\hat{s}) \le \widetilde{D}_{\text{test}}(\hat{s}) \le \epsilon \right) \ge 1-\alpha. \tag{15}$$

B Additional Experimental Details

B.1 Dataset Details

CommonsenseQA (Talmor et al., 2019) is a multiple-choice question-answering dataset designed for commonsense reasoning. Each sample consists of a natural-language question and five candidate answers, including one correct option and four distractors. For all samples, we use a unified task-level prompt template to ensure a consistent input format across different samples. A complete example of the task-level prompt is shown in Figure 5.

TriviaQA (Joshi et al., 2017) is an open-domain question-answering dataset covering a broad range of knowledge domains. Following prior work (Duan et al., 2024), we use the `rc.nocontext` configuration of TriviaQA and randomly sample 2,000 question–answer pairs from its validation set for our experiments. The corresponding task-level prompt is illustrated in Figure 6.

MMLU (Hendrycks et al., 2021) is a multiple-choice benchmark spanning a wide range of knowledge domains and is commonly used to evaluate the knowledge understanding and reasoning capabilities of language models across different subjects. Each sample consists of a question and four candidate answers, denoted by A, B, C, and D. We randomly select 5,000 samples from the complete MMLU test set for our experiments. The corresponding task-level prompt is shown in Figure 7.

CommonsenseQA

Answer the following multiple-choice question.
Choose the single best answer. Respond with only the answer letter.

Question: Where are you if you have to order a cup of coffee?
Choices:
A. mexico
B. tired
C. table
D. coffee shop
E. desk

Final answer:

Figure 5: An example of the task-level prompt used for CommonsenseQA.

TriviaQA

Answer the following question with only the short answer. Do not explain.
Question: Of which tribe of Red Indians was Geronimo a chief
Final answer:

Figure 6: An example of the task-level prompt used for TriviaQA.

MMLU

Answer the following multiple-choice question.
Choose the single best answer. Respond with only A, B, C, or D.

Subject: world religions
Question: In Islam, what is the qiblah?
Choices:
A. Almsgiving
B. Pilgrimage
C. Direction of prayer
D. Revelation

Answer:

Figure 7: An example of the task-level prompt used for MMLU.

B.1.1 Model Input Format

Since different instruction-tuned models may use different native input formats, we process each task-level prompt according to the tokenizer configuration of the corresponding model before feeding it into the model. Specifically, when the tokenizer provides a native chat template, we wrap the task-level prompt as a user message and use the model's built-in chat template to construct the final input, thereby maintaining consistency with the input format

used during instruction tuning. If no chat template is provided by the tokenizer, we directly tokenize the original task-level prompt and feed the resulting token sequence into the model.

B.2 Sampling Details

For CommonsenseQA and TriviaQA, we use the number of attempts required to generate the first correct answer as the performance degradation risk metric. We set the generation parameters to temperature = 0.7, top-p = 0.95, top-k = 50, and maximum new tokens = 32. In addition, to minimize the effect of stochastic sampling fluctuations when comparing different sparsity levels, we pre-generate a set of random seeds for each question, with each seed corresponding to one generation attempt. Specifically, for the same question, the j-th generation attempt at different sparsity levels uses the same j-th random seed, thereby establishing paired stochastic generation conditions across sparsity levels.

For MMLU, we directly extract the next-token logits at the answer position and compute normalized probabilities over the four candidate labels A, B, C, and D. The probability assigned to the ground-truth answer is used as the correct-answer confidence.

B.3 Decision Rules

CommonsenseQA. We first attempt to parse the final option label from the generated text. If the model directly outputs the text of a candidate answer, we further perform normalized matching between the generated output and the candidate answer texts. A response is considered correct only when the final parsed option matches the ground-truth answer key.

TriviaQA. We use exact match as the answer evaluation criterion. Specifically, we apply the same text normalization procedure to the model output and all valid reference answers, including lowercasing, removing punctuation, removing the articles a, an, and the, and normalizing whitespace. A generation is considered correct if the normalized model output exactly matches any valid reference answer.

MMLU. Predictions are made based on the normalized probabilities over the four candidate answers A, B, C, and D. The probability assigned to the ground-truth answer is used as the correct-answer confidence, and its negative value is used as the sample-level performance metric to ensure consistency with the metric direction defined in Eq. equation 1.

C Additional Experimental Results

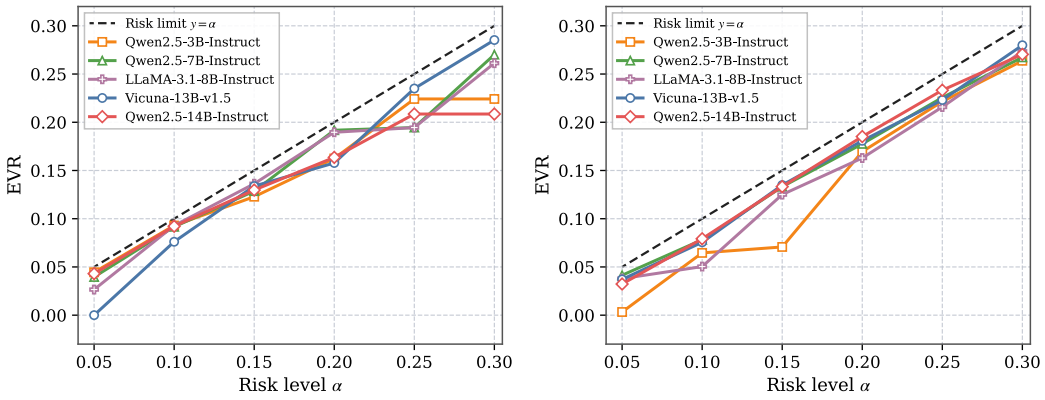
C.1 Ablation Study

To further validate the role and importance of degradation-prefix construction in CRCSS, we conduct additional ablation experiments on CommonsenseQA. Specifically, while keeping the subsequent safe-boundary identification and conformal threshold calibration procedures unchanged, we remove the degradation prefix construction in the first stage and directly perform sparsity selection based on the original performance degradation at each candidate sparsity level. We evaluate the risk-control capability of the resulting sparsity threshold from two perspectives. *Pointwise* evaluation examines whether the performance degradation constraint is satisfied only at the selected sparsity $\hat{s}$, whereas *Prefix-wide* evaluation further examines whether the constraint is satisfied throughout the entire candidate sparsity prefix from s_0 to $\hat{s}$. The results are reported in Table 2.

The results show that CRCSS keeps the EVR within the corresponding target risk level α across all 30 configurations. In contrast, after removing degradation-prefix construction, the EVR exceeds the target risk level in 24 configurations under Pointwise evaluation, and this number further increases to 28 under Prefix-wide evaluation. Moreover, the EVR is substantially higher overall without degradation-prefix construction, with the increase being more pronounced under Prefix-wide evaluation. These results indicate that directly calibrating based on the original performance degradation makes it difficult to achieve stable risk control. The underlying reason is that the original degradation trajectory is not necessarily monotonic with respect to sparsity: even if a higher sparsity level satisfies

Table 2: Ablation results of degradation-prefix construction, where green indicates that the EVR does not exceed the target risk level, while red indicates a violation.

Model	Method	Risk level α					
		0.05	0.10	0.15	0.20	0.25	0.30
Qwen-2.5-3B-Instruct	w/o degradation prefix (Pointwise)	0.1120	0.1635	0.2737	0.2737	0.3185	0.4309
	w/o degradation prefix (Prefix-wide)	0.1586	0.2219	0.3442	0.3442	0.4096	0.5758
	CRCSS	0.0416	0.0923	0.1232	0.1586	0.2207	0.2207
Qwen-2.5-7B-Instruct	w/o degradation prefix (Pointwise)	0.0947	0.1740	0.2063	0.2092	0.2092	0.2092
	w/o degradation prefix (Prefix-wide)	0.3083	0.4049	0.4496	0.4539	0.4539	0.4539
	CRCSS	0.0310	0.0608	0.0608	0.0608	0.2381	0.2865
LLaMA-3.1-8B-Instruct	w/o degradation prefix (Pointwise)	0.1139	0.1605	0.1872	0.2546	0.2546	0.3581
	w/o degradation prefix (Prefix-wide)	0.1874	0.2535	0.2953	0.3784	0.3784	0.4904
	CRCSS	0.0326	0.0891	0.1344	0.1861	0.2282	0.2856
Vicuna-13B-v1.5	w/o degradation prefix (Pointwise)	0.1483	0.1765	0.2028	0.2402	0.2625	0.2625
	w/o degradation prefix (Prefix-wide)	0.2753	0.3379	0.3798	0.4246	0.4525	0.4525
	CRCSS	0.0000	0.0000	0.1379	0.1828	0.2370	0.2753
Qwen-2.5-14B-Instruct	w/o degradation prefix (Pointwise)	0.0887	0.1316	0.1617	0.1617	0.1617	0.1617
	w/o degradation prefix (Prefix-wide)	0.1308	0.1848	0.2286	0.2286	0.2286	0.2286
	CRCSS	0.0460	0.0936	0.1398	0.1848	0.2286	0.2286



(a) *CommonsenseQA (OWL)* (b) *TriviaQA (OWL)*

Figure 8: Empirical violation rate (EVR) of CRCSS on CommonsenseQA and TriviaQA with OWL under different risk levels.

the performance degradation constraint, some preceding candidate sparsity levels may still exceed the predefined degradation tolerance.

Therefore, transforming the original degradation trajectory into a nondecreasing degradation prefix is a key construction that enables CRCSS to provide prefix-level risk guarantees. This transformation extends pointwise acceptability at the selected sparsity to risk control over the entire candidate sparsity prefix from the dense model to the selected sparsity level.

C.2 Additional Statistical Validity Results

To further evaluate the statistical validity and applicability of CRCSS across different pruning methods, we additionally consider OWL (Yin et al., 2024). Unlike conventional pruning approaches that apply a uniform sparsity level across layers, OWL assigns different sparsity levels to individual layers according to their characteristics, thereby providing a pruning configuration that differs from those used in the main-text experiments. The experimental results of CRCSS with OWL are presented in Figure 8. As shown, the mean EVR does not exceed the corresponding target risk level α across both datasets and all evaluated models. These results further demonstrate that CRCSS can accommodate different types of pruning methods while maintaining its statistical validity.

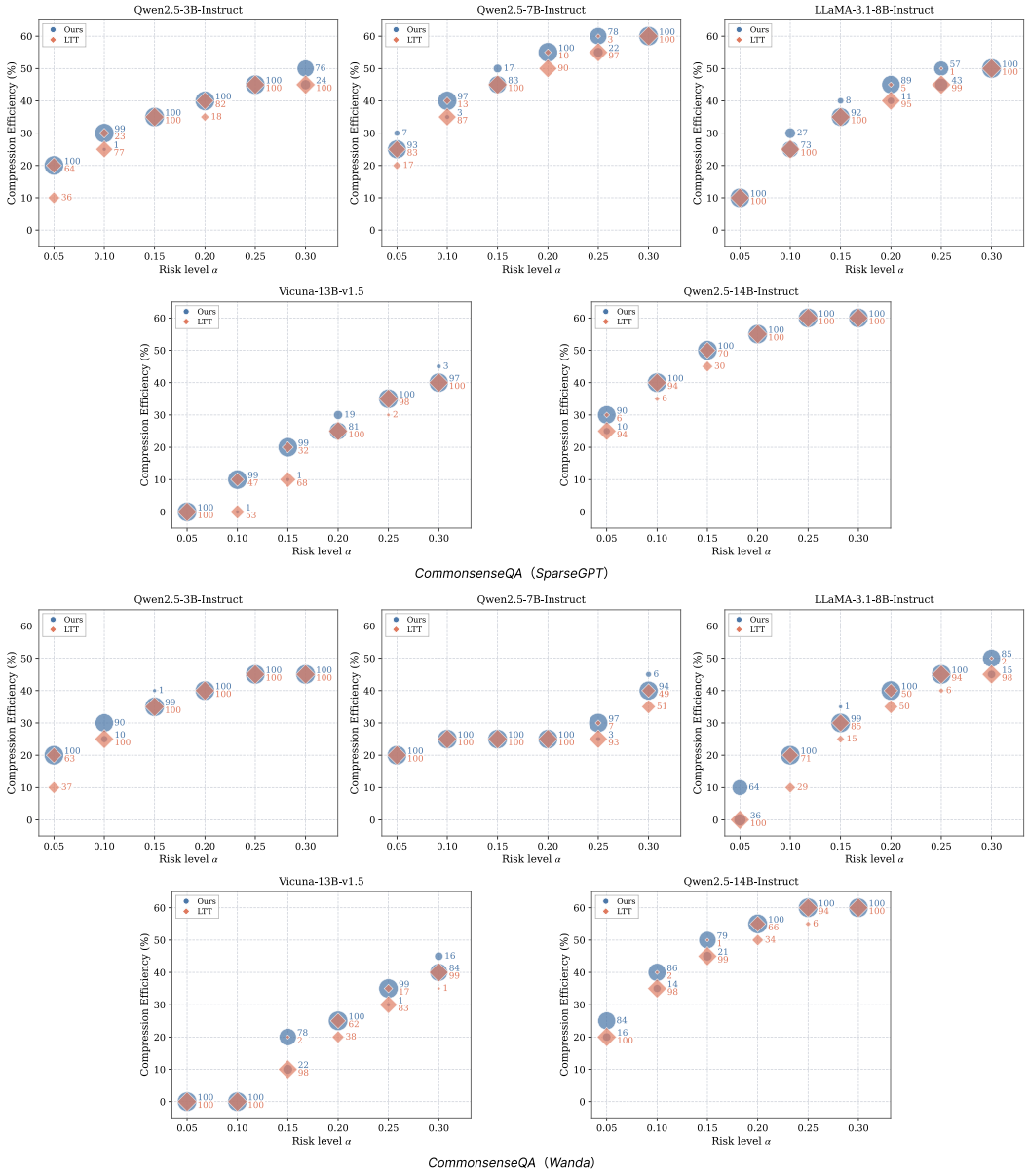


Figure 9: Compression efficiency comparison between CRCSS and LTT on CommonsenseQA.

C.3 Additional Comparison with Learn-then-Test

To further evaluate the compression efficiency of CRCSS, we conduct an additional comprehensive comparison with LTT on CommonsenseQA, where δ is set consistently with Section 4.3. The results are shown in Figure 9. Among the 60 configurations, CRCSS selects a higher sparsity level in 41 configurations, while the remaining 19 configurations achieve the same sparsity level as LTT. In addition, we include more model configurations on TriviaQA and compare CRCSS with LTT. Considering that the choice of the confidence parameter δ in LTT may lead to conservative sparsity selection, we set $\delta = 0.1$ to mitigate the conservativeness induced by a stringent confidence requirement. The results are shown in Figure 10. Among the 24 additional configurations, CRCSS selects a higher safe sparsity level in 21 configurations, while the remaining 3 configurations achieve the same sparsity level as LTT. These results show that, CRCSS selects a higher safe sparsity level in most configurations, thereby achieving higher compression efficiency. However, it should be noted

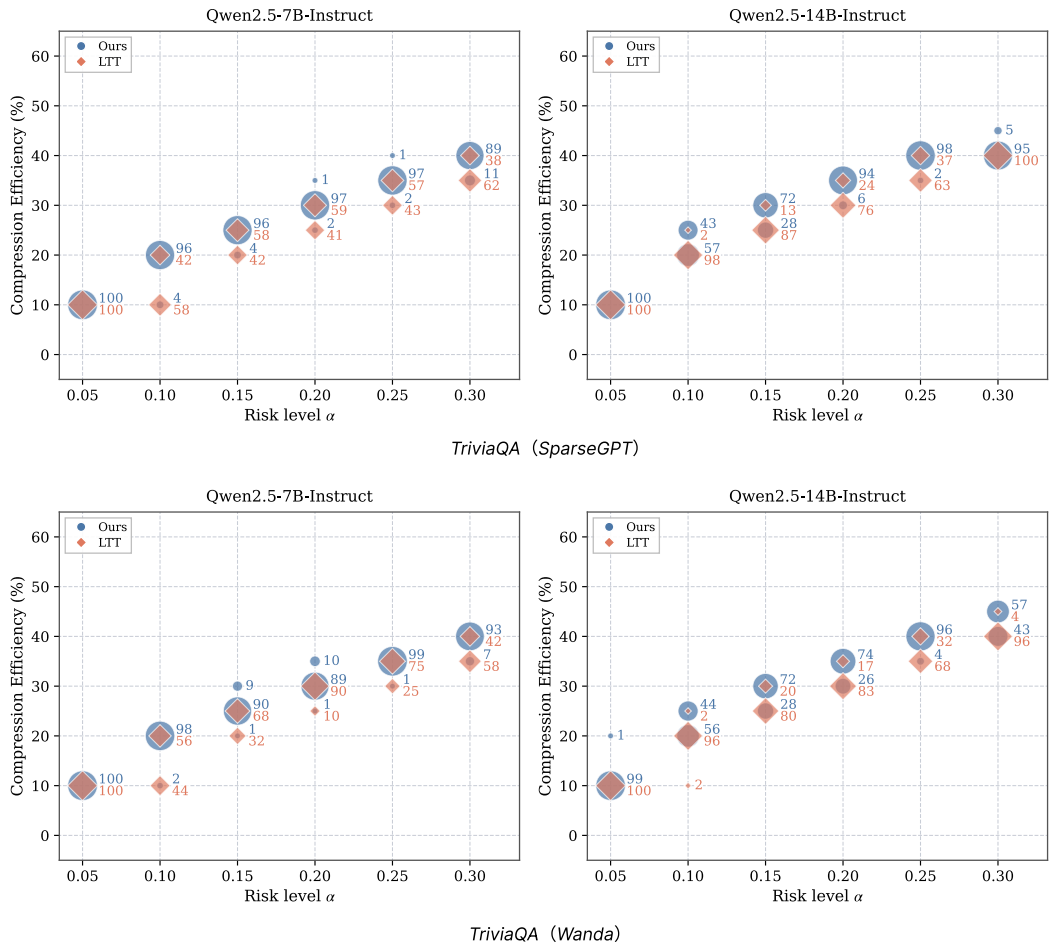


Figure 10: Compression efficiency comparison between CRCSS and LTT on TriviaQA with Qwen-2.5-7B-Instruct, Qwen-2.5-14B-Instruct.

that CRCSS and LTT provide different forms of statistical guarantees: LTT controls the population expected risk, whereas CRCSS provides a finite-sample marginal guarantee. Therefore, this comparison is intended to examine the sparsity selection behavior of the two methods under their respective risk-control mechanisms.

In addition, the selected safe sparsity levels vary noticeably across datasets. Taking Wanda as an example, when $\alpha = 0.05$, Qwen-2.5-7B-Instruct selects a sparsity level of 10% on TriviaQA, compared with 20% on CommonsenseQA. This reflects differences across tasks in their sensitivity to pruning-induced performance degradation. Therefore, in practical deployment, using a uniform or predefined sparsity level may not adequately reflect the performance risk on the target data. In contrast, calibrating the sparsity level on the target data under explicit risk control may be more meaningful in practical deployment.

C.4 Additional Compression Efficiency Results under Confidence Drop

We further analyze the compression efficiency of CRCSS under the confidence-drop risk metric. Figure 11 shows the distributions of selected sparsity levels over 100 repeated experiments for different models and pruning methods on MMLU. As the target risk level α increases, CRCSS tends to select higher safe sparsity levels while satisfying the prescribed risk constraint. This trend is consistent with the results in Section 4.3: moderately relaxing the risk constraint allows for a higher degree of compression, thereby illustrating the trade-off between reliability and compression efficiency.

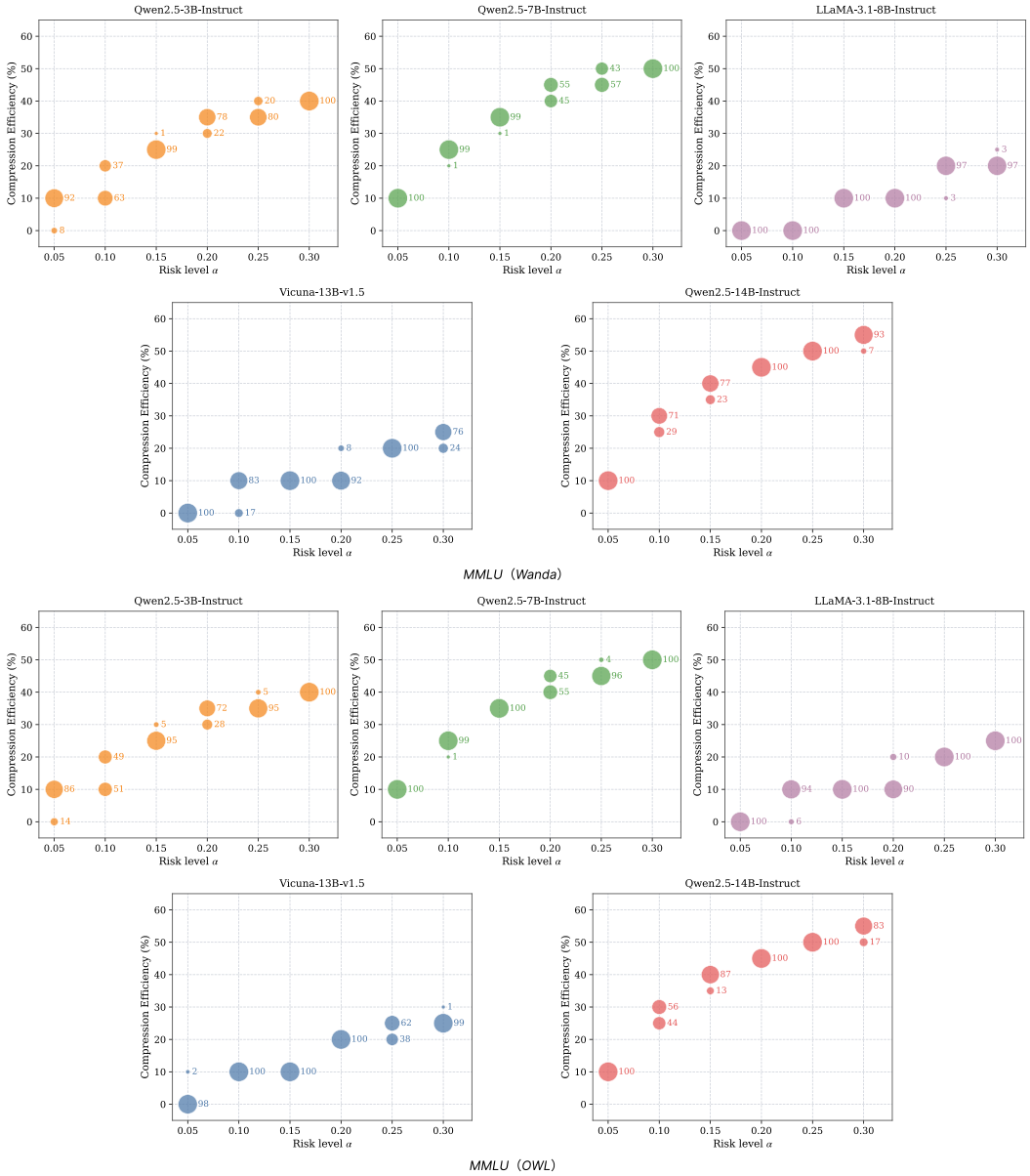


Figure 11: Distribution of sparsity levels selected by CRCSS on MMLU under different configurations.

Notably, the distributions of selected sparsity levels vary substantially across different models. Taking Wanda as an example, when $\alpha = 0.25$, Qwen-2.5-14B-Instruct selects a sparsity level of 50% in all 100 repeated experiments, whereas Vicuna-13B-v1.5 consistently selects a sparsity level of 20%. This discrepancy reflects substantial differences in pruning sensitivity across models. It further indicates that, in LLM pruning, adaptively selecting the sparsity level under a risk constraint can better account for model-specific pruning tolerance than using a uniform or predefined sparsity level.

C.5 Additional Results on Calibration Size

To further complement the analysis of sensitivity to calibration-set size, we report detailed results on TriviaQA under different calibration:test split ratios, as shown in Table 3. The results show that the risk control provided by CRCSS does not rely on a large calibration set.

Even when the number of calibration samples is substantially reduced, the empirical violation rate remains within the corresponding target risk level. These results further demonstrate the robustness of CRCSS to variations in calibration-set size.

Table 3: Empirical violation rates of CRCSS with OWL under different calibration:test split ratios on CommonsenseQA and TriviaQA.

Dataset	Model	Risk level α					
		0.05	0.10	0.15	0.20	0.25	0.30
Calibration:test = 3:7							
CommonsenseQA	Qwen-2.5-3B-Instruct	0.0438	0.0911	0.1234	0.1628	0.2243	0.2243
	Qwen-2.5-7B-Instruct	0.0401	0.0910	0.1277	0.1862	0.1937	0.2693
	LLaMA-3.1-8B-Instruct	0.0285	0.0923	0.1366	0.1843	0.1980	0.2615
	Vicuna-13B-v1.5	0.0000	0.0764	0.1349	0.1581	0.2363	0.2827
	Qwen-2.5-14B-Instruct	0.0435	0.0905	0.1305	0.1670	0.2085	0.2085
TriviaQA	Qwen-2.5-3B-Instruct	0.0082	0.0651	0.0809	0.1690	0.2179	0.2599
	Qwen-2.5-7B-Instruct	0.0380	0.0786	0.1335	0.1774	0.2266	0.2711
	LLaMA-3.1-8B-Instruct	0.0354	0.0580	0.1297	0.1684	0.2228	0.2767
	Vicuna-13B-v1.5	0.0327	0.0676	0.1280	0.1764	0.2198	0.2688
	Qwen-2.5-14B-Instruct	0.0319	0.0806	0.1308	0.1795	0.2273	0.2695
Calibration:test = 2:8							
CommonsenseQA	Qwen-2.5-3B-Instruct	0.0407	0.0888	0.1251	0.1634	0.2237	0.2243
	Qwen-2.5-7B-Instruct	0.0398	0.0898	0.1282	0.1825	0.1944	0.2693
	LLaMA-3.1-8B-Instruct	0.0309	0.0903	0.1353	0.1820	0.1978	0.2613
	Vicuna-13B-v1.5	0.0000	0.0765	0.1371	0.1582	0.2376	0.2818
	Qwen-2.5-14B-Instruct	0.0449	0.0906	0.1313	0.1712	0.2085	0.2085
TriviaQA	Qwen-2.5-3B-Instruct	0.0057	0.0651	0.0774	0.1618	0.2163	0.2568
	Qwen-2.5-7B-Instruct	0.0337	0.0765	0.1313	0.1729	0.2288	0.2690
	LLaMA-3.1-8B-Instruct	0.0279	0.0561	0.1244	0.1674	0.2184	0.2735
	Vicuna-13B-v1.5	0.0299	0.0664	0.1257	0.1779	0.2200	0.2671
	Qwen-2.5-14B-Instruct	0.0320	0.0826	0.1314	0.1795	0.2251	0.2676
Calibration:test = 1:9							
CommonsenseQA	Qwen-2.5-3B-Instruct	0.0375	0.0881	0.1281	0.1646	0.2223	0.2241
	Qwen-2.5-7B-Instruct	0.0396	0.0874	0.1281	0.1760	0.1966	0.2669
	LLaMA-3.1-8B-Instruct	0.0314	0.0880	0.1336	0.1762	0.2041	0.2615
	Vicuna-13B-v1.5	0.0000	0.0766	0.1325	0.1625	0.2382	0.2772
	Qwen-2.5-14B-Instruct	0.0446	0.0896	0.1313	0.1757	0.2085	0.2085
TriviaQA	Qwen-2.5-3B-Instruct	0.0076	0.0615	0.0863	0.1559	0.2119	0.2438
	Qwen-2.5-7B-Instruct	0.0192	0.0714	0.1253	0.1670	0.2216	0.2693
	LLaMA-3.1-8B-Instruct	0.0260	0.0648	0.1206	0.1700	0.2109	0.2664
	Vicuna-13B-v1.5	0.0230	0.0696	0.1197	0.1714	0.2173	0.2611
	Qwen-2.5-14B-Instruct	0.0285	0.0764	0.1270	0.1805	0.2203	0.2709