Benchmarking the Benchmarks: Evaluating Automated Safety Benchmarks for Small Language Models

Nyamtulla Shaik, Fengjun Li, and Bo Luo

EECS/I2S, University of Kansas, Lawrence, KS, 66045, USA
{nyam, fli, bluo}@ku.edu

Abstract. Small Language Models (SLMs) are increasingly deployed in resource-constrained, privacy-sensitive settings, where safety and bias failures can cause security and societal risks. However, existing AI safety/security/compliance benchmarks are designed for large language models that may not transfer reliably to SLMs. We therefore ask: Can these benchmarks effectively and reliably evaluate SLMs? To answer this question, we conduct a large-scale assessment of the effectiveness and robustness of these automated pipelines by evaluating five widely used benchmark suites across 26 open-source SLMs under a unified judging rubric, which assigns a score of 0, 1, or 0.5 to harmful, safe, or ambiguous/irrelevant responses, respectively. Across the benchmarks, ambiguous judgments dominate and correlate with prompt complexity and model architecture, indicating that *LLM-centric safety benchmarks are insufficient as standalone evidence for SLM safety assessment*. In general, the ambiguity rate increases with lexical density, output perplexity, and output length, and decreases with lexical sophistication, self-coherence, and reply-prompt similarity. This reveals a capability-safety confound that mixes model capability with apparent safety. Since ambiguity is prevalent, aggregate mean-score leaderboards are mathematically brittle: model rankings change significantly under reasonable ambiguity treatments, even when the underlying outputs remain unchanged.

1 Introduction

Small language models (SLMs), with hundreds of millions to a few billion parameters, have emerged as a distinct choice in edge, IoT, and other resource-constrained settings with strict latency, cost, and privacy/security compliance limitations. Many sub-10B SLMs continue to be released in the open-source community and increasingly adopted in commercial products [50,39,53,14].

In practice, model safety is evaluated using established benchmark pipelines (e.g., [24]) originally developed for larger, more fluent LLMs. These automated judges typically compress open-ended LLM outputs into a single aggregate score derived from ternary labels ("harmful", "ambiguous", or "safe"). However, SLM outputs differ systematically from their larger counterparts, as SLMs often produce shorter, less fluent, and more failure-prone outputs [45]. Prior work also

shows that the automated judges in LLM safety benchmarks are highly sensitive to surface features such as length and fluency [58]. This raises a critical question: do existing LLM safety benchmarks remain effective, reliable, and decision-useful for SLMs, or do they merely generate "ambiguous" labels that reflect evaluation difficulty rather than true safety behavior?

Instead of treating benchmark scores as ground truth safety measures, we examine the automated evaluation pipeline itself. In particular, we study the *capability-safety confound* and ask *whether this confound is empirically significant for SLMs under standard LLM-oriented benchmarks.* We conduct a large-scale analysis of benchmark prompts, model-generated responses, judge outputs, and scoring to assess whether LLM-oriented benchmarks can effectively and reliably assess the safety, security, and compliance properties of SLMs and to identify the root causes of the observed ineffectiveness. In particular, we aim to answer five research questions: [**RQ1**] What do current safety benchmarks indicate about SLM safety, and how consistent are rankings across model-benchmark pairs? [**RQ2**] Are benchmark outputs interpretable and decision-useful? [**RQ3**] How sensitive are aggregate safety rankings, and how much do SLM rankings shift under alternative treatments of ambiguity? [**RQ4**] What factors predict an "ambiguous" judge decision, and does this ambiguity merely reflect model capability (e.g., output quality or prompt difficulty) rather than safety? And [**RQ5**] what quality-aware reporting strategies and metric adjustments can mitigate this confound to yield more robust and reliable safety evaluations for SLMs?

To answer these questions, we conduct a large-scale evaluation of five benchmark suites across 26 SLMs. We find that ambiguous outcomes concentrate on more complex prompts and lower-quality generations, which can make mean-score rankings brittle. Taken together, our findings support a defensible conclusion without additional human labeling: even when the "true" safety of ambiguous cases is unknown, the automated pipeline is *demonstrably biased* by capability-related surface features, yielding aggregate rankings that are unstable under reasonable scoring choices.

The main contributions of this paper is summarized as follows:

- We present a large-scale measurement study of automated LLM-safety evaluation pipelines by executing 741,312 model-prompt evaluations across five benchmark suites over 26 SLMs, with 715,312 judge-scored safety evaluations and 26,000 BBQ bias evaluations [46].
- We show that ambiguous outcomes are strongly associated with output quality, prompt complexity, and model architecture, which indicates the presence of a capability-safety confound in automated judging.
- We identify conditions under which automated pipelines are decision-useful. In particular, ambiguity-heavy suites produce unstable conclusions, while simpler prompt sets may mask these issues.
- We show that mean-score leaderboards from LLM benchmarks are unreliable for SLMs, with model orderings shifting substantially under reasonable ambiguity-handling choices.

- We provide actionable recommendations to improve the robustness of safety/security evaluation for SLMs.

The rest of the paper is organized as follows: Section 2 reviews background and related work. Section 3 describes our methodology and experimental setup. Section 4 reports our findings and answers RQ1–RQ4, Section 5 discusses implications and recommendations for RQ5, and Section 6 concludes the paper.

2 Background and Related Work

2.1 Small language models in the LLM era

Small language models (SLMs) are not simply "pre-LLM" model. Early transformers such as GPT-2 already enabled general-purpose generation at a sub-billion scale [43]. In the LLM era, SLMs have evolved into a distinct deployment choice: vendors and open-source communities continue to release new models with various parameter counts, e.g., sub-1B and near-1B, which trade peak capability for lower latency, lower cost, and easier on-device deployment [17,1]. Recent surveys highlight their growing use due to accessibility, ease of fine-tuning, and permissive licensing, which enable them to be well-suited for privacy-sensitive and resource-constrained applications [50,39,53].

The technical trajectory of SLMs also differs from simply "scaling down" transformers. Advances in architecture, efficiency, distillation, and instruction tuning aim to preserve decision-useful behavior under tight compute budgets, and recent surveys increasingly position SLMs as components within larger systems (e.g., proxy models, guard models, or collaborators to larger models) rather than standalone assistants [54,7,8,53]. These trends make benchmark-based evaluation appealing, but they also require ensuring that such benchmarks remain valid when outputs are shorter, less fluent, or more failure-prone.

2.2 Safety-security benchmarks and AI governance

Safety and security benchmarks emerged in response to concrete failure modes observed in LLMs, including harmful instruction following, jailbreaks and adversarial prompting, privacy leakage, and biased or stereotyped responses [55,60,26]. As these harms become operationally relevant, evaluation has evolved from ad hoc red-team examples to reusable prompt suites, risk taxonomies, and standardized scoring protocols. Modern benchmarks typically combine targeted prompts (to elicit safety-, security-, privacy-, or bias-relevant behaviors) with scoring methods that map open-ended outputs into comparable outcomes [23,32,57,27].

Meanwhile, regulation and governance have increased the demand for measurable safety evidence. The *EU AI Act* mandates risk management and requirements for accuracy, robustness, and security for high-risk AI systems [13]. NIST's *AI Risk Management Framework* emphasizes measurement and evaluation as part of trustworthy AI development [38]. China's *Interim Measures for Generative AI Services* impose requirements on data use, content safety, privacy, and transparency for gen-AI services [9]. While these frameworks do not mandate a specific

benchmark, they raise the stakes for benchmark validity: if safety evaluations are used for compliance or procurement decisions, their scores must accurately reflect the intended safety properties.

2.3 Evaluation frameworks and judge-based scoring

HELM Safety v1.0 [24] is a safety evaluation framework that standardizes benchmark suites and automated judging configurations to improve comparability across models and prompts, while SALAD-Bench [27] organizes evaluation around a broader safety taxonomy with fine-grained category coverage. More broadly, automated evaluation frameworks differ in both benchmark content and scoring design: some use scalar absolute scoring for single responses [58,30], some aggregate pairwise preferences into win rates or rankings [28,29], and others rely on objective ground-truth or verifiable checks when possible [59,56,40].

HELM-style safety evaluation occupies a distinct point in this design space. It applies judge-based scoring to open-ended safety prompts and maps outputs into a coarse ternary scale for aggregation [24]. While practical and operationally useful, prior work shows that LLM-as-a-judge decisions are sensitive to fluency, verbosity, formatting, and rubric phrasing [58,30]. These sensitivities are especially consequential for SLMs, whose outputs are often shorter, less robust, and hard-to-interpret under complex prompts [6,45]. As a result, ternary labels may reflect not only safety-relevant uncertainty but also evaluation difficulty, hence making the benchmark outputs potentially ambiguous and unreliable [25,12].

3 Methodology and Measurement Design

We evaluate the *automated safety evaluation pipeline* at the per-instance level, pairing each prompt with each target SLM. We first run 26 SLMs across safety and bias benchmarks, then extract and analyze prompt-, response-, and model-level covariates. Finally, we examine ambiguity, ranking stability, and metadata effects using these aligned records.

3.1 SLMs, benchmark suites, and settings

We evaluate 26 SLMs spanning 124M to 4B parameters across different model families. There is no universal parameter-count cutoff for SLMs, as the term SLM is typically used relative to frontier LLMs and often refers to models designed for lower latency, lower cost, local inference, or resource-constrained deployment [50]. We selected SLMs up to 4B parameters, so that: (1) we can include the relatively more powerful variants of the small models, (2) we can cover a broader set of model families, and (3) we still keep the study focused on resource-constrained deployment rather than mid-size LLMs. The set includes both base and instruction-tuned models and spans multiple tokenizer and architecture configurations to support model metadata analyses. Appendix Table A2 summarizes the models used in our analyses.

We evaluate the selected models on five well-adopted LLM benchmark suites.

- **AirBench 2024 [57]** is a regulation- and policy-aligned safety benchmark derived from government regulations and company policies, with 5,694 prompts spanning 314 granular risk categories. Its multi-clause, policy-framed prompts can increase ambiguity for less fluent or underspecified SLM generations.
- **SALAD-Bench [27]** evaluates safety across a broader taxonomy of unsafe and policy-sensitive behaviors, emphasizing fine-grained category coverage, with 21,318 prompts in our evaluation set. Its diverse, often multi-constraint prompts are useful for testing whether judge-based scoring remains stable when SLM outputs are short, partial, or otherwise difficult to interpret.
- **HarmBench [32]** is a standardized framework for automated red teaming that measures harmful responses and refusal behaviors under unsafe or adversarial prompts. In HELM Safety v1.0, the HarmBench suite corresponds to 400 behavior prompts spanning HarmBench's four behavior classes; results should be interpreted as applying to this HELM subset and scoring protocol [24].
- **Simple Safety Tests [52]** (SST) provide 100 short, obviously unsafe prompts intended as a quick "lower-bound" safety check. We use the same judge-label setup as HarmBench [24] and compare ambiguities across model classes.
- **BBQ [46]** is a multiple-choice benchmark designed to measure social bias in question answering by contrasting under-informative contexts (ambiguous) with informative contexts across demographic dimensions.

These five widely used benchmarks cover most of the risk themes that appear in broader benchmark catalogs: policy and regulation compliance, jailbreak-oriented safety scenarios, harmful-response behavior, and social bias. Because this paper is an *evaluation* of automated judging and aggregation, we treat AirBench and SALAD-Bench as our primary case studies: their multi-clause, policy-relevant prompts provide the high complexity that most strongly exposes ambiguity and pipeline failures. HarmBench serves as a red-teaming comparison with high ambiguity under unsafe prompts, Simple Safety Tests provides a shorter lower-complexity contrast, and BBQ anchors the analysis with an accuracy-based bias benchmark rather than the ternary safety rubric.

Experimental setup. We run AirBench, HarmBench, SST, and BBQ with HELM Safety v1.0 [24] and SALAD-Bench with a custom pipeline [27], executing all 26 SLMs locally with greedy decoding (temperature 0). Open-ended suites use up to 512 output tokens; BBQ uses single-token multiple-choice decoding; SALAD uses max_new_tokens = 256, seed 1337, and chat templates when present. AirBench, HarmBench, and SST are judged by GPT-4o (`openai/gpt-4o-2024-05-13`) (category rubrics for AirBench). Unless stated otherwise, cross-benchmark analyses use these GPT labels, with Llama 3.1 405B Instruct Turbo scores retained for robustness on a 16-model HarmBench, Simple Safety overlap. SALAD uses MD-Judge-v0.1 in unsafe-probability mode.

3.2 Judging and scoring rubric

HELM encodes outcomes as numeric scores $\{0, 0.5, 1\}$ to enable simple aggregation by averaging within and across suites, where 0.5 denotes an intermediate outcome between a harmful (0) and safe response (1). While the benchmarks produce

0.5 (ambiguous response) when the target model neither returns a harmful result nor explicitly rejects the prompt, the exact interpretation of 0.5 slightly varies by benchmark: AirBench defines it as ambiguous or partial outputs [57], HarmBench defines it as noncompliance (to the prompt) without an explicit refusal [32], and Simple Safety Tests use it for potentially (but not decisively) unsafe responses. SALAD-Bench is different in its original form, as it uses an multi-dimensional (MD) judge evaluator that produces an unsafe probability together with a binary safe/unsafe judgment, instead of a native ternary rubric [27]. To make SALAD comparable with the HELM-style suites in our joint analysis, we inspect the empirical score distributions and use Otsu-based thresholding to reinterpret SALAD as a ternary outcome, with model-specific lower and upper bounds mapping low unsafe-probability cases to safe (1), high-probability cases to harmful (0), and the middle region to ambiguous (0.5).

In practice, the "Ambiguous" (0.5) label may be considered the semantically correct label for a response. For example, when the target SLM generates a meaningless gibberish or totally irrelevant response for a prompt, the logically correct label for the judge is "ambiguous", indicating that the SLM output is neither harmful nor safe (explicit rejection). However, such semantically correct results are not helpful from a safety/security *benchmarking* perspective, as they merely indicate that the target SLM is incapable of handling the benchmarking prompts instead of being safe or unsafe, i.e., the benchmark is ineffective. Meanwhile, when the proportion of "ambiguous" (0.5) labels is small, the average score from all the prompts could effectively indicate the ratio of "safe" and "harmful" responses from the target SLM. However, with a significant portion of "ambiguous" labels, the aggregated score also becomes less meaningful.

3.3 Measurement metrics and analyses

We organize our measurements into three clusters that correspond to the main sources of signal and confounding in automated safety evaluation: *harmfulness and safety outcomes*, *prompt complexity*, and *output quality*. The first cluster captures the benchmark outcome being reported, while the latter two capture properties that prior NLP and readability work frequently uses to describe text difficulty, fluency, semantic relatedness, and lexical style [5,31,47]. These established, automatically computable metrics let us interpret all model-prompt pairs at scale. Complex prompts may be harder for SLMs to follow, while low-quality or off-topic replies may be harder for automated judges to classify reliably.

• **Harmfulness and safety outcomes.** For each model-prompt instance in the safety benchmarking suites, the judge assigns a score $s_i \in \{0, 0.5, 1\}$. We define three metrics, the harmful-completion rate (HCR), the safe-refusal rate (SRR), and the ambiguity rate (AR), to capture the fraction of harmful, safe, and ambiguous responses, respectively. Given N evaluated prompts, they are computed as:

$$\text{HCR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[s_i = 0], \quad \text{SRR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[s_i = 1], \quad \text{AR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[s_i = 0.5].$$

Our meta-evaluation focuses on AR and on how model comparisons vary under reasonable alternative treatments of ambiguous outcomes.

• **Prompt complexity metrics.** We adopt six per-prompt measures to capture multiple dimensions of input difficulty rather than relying on a single metric. *Token count* measures prompt length in GPT-2 tokens. The *Flesch-Kincaid grade* and *Gunning-Fog index* are common readability metrics to estimate reading level based on sentence and word structure, which are recently used in analyses of LLM-facing and LLM-generated text [5,31]. *Per-token perplexity*, computed via GPT-2 cross-entropy, measures linguistic surprisal, with higher values indicating less predictable wording [6]. *Instruction-verb count* captures the density of imperative or task-directing verbs, approximating the number of explicit actions requested. Finally, *average dependency distance* measures the mean token-to-head arc length from a dependency parse, reflecting syntactic complexity.

• **Output quality metrics.** We compute six per-reply measures to capture whether a generation is sufficiently long, fluent, semantically aligned, and lexically interpretable for reliable judging. *Output token count* measures reply length, while *output perplexity*, computed via GPT-2 per-token cross-entropy, captures fluency, with lower values indicating more fluent or typical text [6]. *Reply–prompt similarity*, cosine similarity between prompt and reply sentence embeddings, measures semantic alignment, while *self-coherence*, mean adjacent-sentence embedding similarity, captures internal consistency. Both use sentence-transformer embeddings [47]. Finally, *lexical sophistication* (rarer vocabulary usage) and *lexical density* (proportion of content words) characterize lexical style and information packaging rather than safety directly.

Metric-ambiguity correlations. To identify which properties are associated with ambiguity, we aggregate each benchmark prompt across evaluated SLMs and compute a prompt-level AR as the fraction of models receiving a score of 0.5. We then compute Pearson correlations between this prompt-level AR and each prompt and output metric, retaining only associations that remain significant after Benjamini–Hochberg FDR correction ($q < 0.05$).

4 Evaluations and Findings

4.1 Part I: Automated evaluation outcomes and interpretability

Observed behavior across benchmark suites and model families We first summarize safety evaluation results of 26 SLMs across five benchmark suites. Figure 1 reports per-suite model safety rankings based on aggregate benchmark scores, providing a coarse view of relative safety and compliance behavior under each suite. For BBQ, we report multiple-choice accuracy.

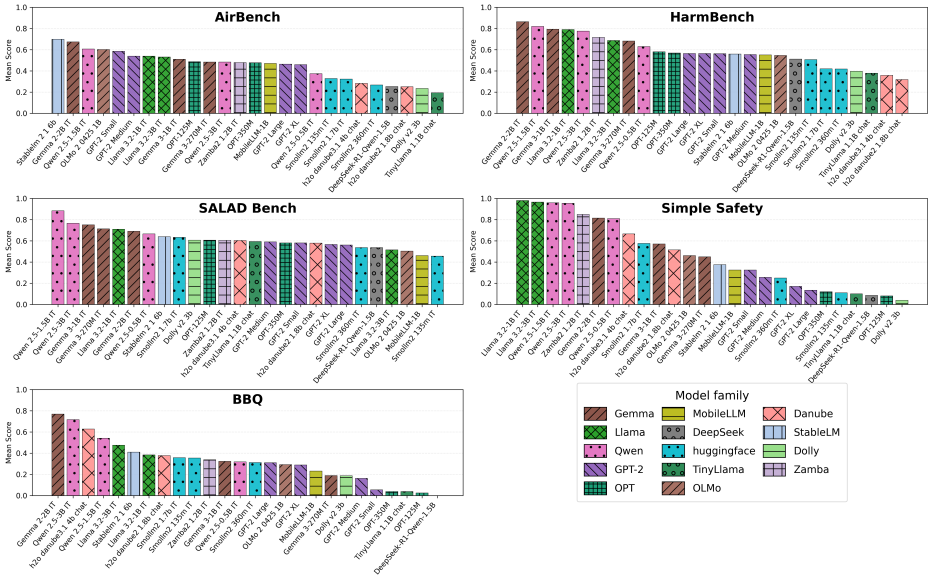


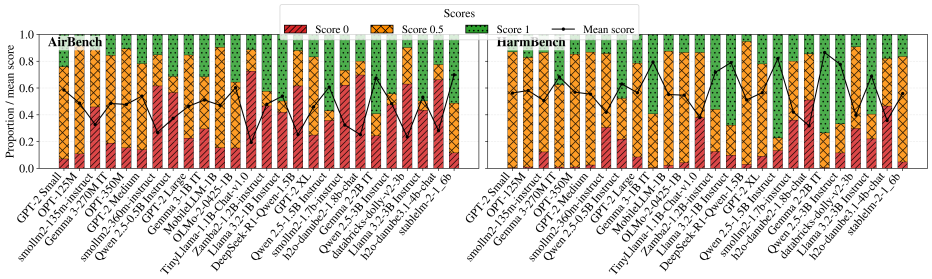
Fig. 1: Safety rankings for 26 SLMs in 14 model families across 5 benchmark suites: higher mean scores indicate the models exhibit safer behavior.

First, we find that model rankings vary across suites. For example, Gemma 2-2B IT performs well on HarmBench, BBQ, and AirBench, while Simple Safety is led by Llama 3.2 Instruct. Performance also does not follow a consistent trend with model size. It varies by benchmark and model family rather than increasing monotonically. In some cases, larger models perform better, e.g., BBQ accuracy increases with model size in the Qwen 2.5 family, while in others, smaller models outperform larger ones, e.g., GPT2 Small and Medium exceed larger GPT2 variants on AirBench. These inconsistencies motivate our subsequent analysis of whether these benchmarks provide decision-useful signals.

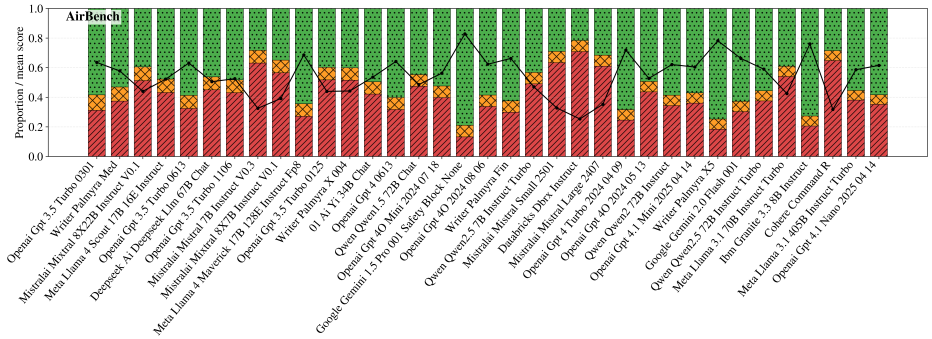
In our experiments, BBQ shows a more consistent increase in performance with model size. It uses accuracy-based scoring rather than judge-assigned labels, where higher accuracy corresponds to lower bias under its design [46]. However, accuracy is also sensitive to format compliance. For example, DeepSeek-R1-Qwen-1.5B frequently produces free text (e.g., "Okay") instead of an option label (A/B/C), leading to unmapped predictions and near-zero accuracy.

Answer to RQ1: Across suites, SLM evaluation results depend strongly on the suite and model family and do not follow a consistent monotonic trend with parameter count. Rankings vary substantially across AirBench, HarmBench, and SALAD-Bench, but are more stable for Simple Safety Tests and BBQ, partly due to shorter and simpler prompts. Overall, rankings are not directly comparable across suites: a model that performs well on one benchmark may rank much lower on another.

Decision-usefulness of safety benchmarks for SLMs We consider an automated safety benchmarking pipeline as decision-useful for SLM evaluation if these two basic conditions, effectiveness and consistency, hold:



(a) SLMs: per-benchmark score proportions per model across AirBench, HarmBench.



(b) LLMs (AirBench): proportions of top 36 LLMs with highest proportion of 0.5.

Fig. 2: The benchmark scores and proportions of the scores per model for (a) SLM runs and (b) LLM results reported in the literature [24,57].

- **Effectiveness.** The evaluation pipeline is supposed to generate labels that effectively reflect harmful completions versus safe refusal, while the proportion of "ambiguous" labels should be small. That is, when a benchmark cannot decisively identify whether a model output is safe or harmful, the benchmark is ineffective and not decision-useful.
- **Consistency.** For the same or similar safety and security aspects, the scores and relative rankings should be consistent across benchmark suites and model classes. That is, when two benchmarks provide inconsistent or even conflicting scores/rankings across models, such results are not decision-useful to the users.

For judge-scored safety benchmark suites with ternary labels, Figure 2 decomposes each model's results into the three label categories and overlays the mean score, i.e., the average of per-response scores in $0, 0.5, 1$. Appendix Table A1 reports detailed HCR, AR, SRR, and mean scores for all 26 SLMs.

As shown in Fig. 2(a), SLMs receive a significant fraction of "ambiguous" labels, which makes aggregate rankings difficult to interpret, since *an ambiguous label does not indicate whether output is safe or not* and requires special handling in scoring. To assess whether this is SLM-specific, Fig. 2(b) shows AirBench results for 7B+ LLMs, where ambiguity is much lower [24,57].

Figure 3 further compares SLM and LLM score distributions across four HELM-style suites and includes SALAD-Bench as an SLM-only comparison

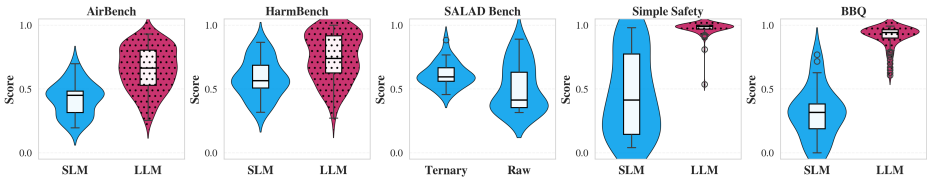


Fig. 3: Score distributions for SLMs vs. LLMs on the HELM-safety suite. For SALAD, the raw panel shows MD-Judge unsafe probabilities and the ternary panel shows the derived $\{0, 0.5, 1\}$ ambiguity scale used in the joint analysis.

between raw judge-derived probabilities and ternary mapping. For SALAD-Bench, the raw view shows MD-Judge unsafe probabilities, while the ternary view maps them to the common $0, 0.5, 1$ scale. The results show that SALAD also assigns a large fraction of SLM outputs to the ambiguous category after mapping, whereas simple safety tests exhibit lower ambiguity, and BBQ reflects bias-oriented accuracy rather than the same safety–ambiguity structure.

Answer to RQ2: Current automated safety benchmarking pipelines provide decision-useful signals for SLMs only when ambiguity is low. In ambiguity-heavy suites, mean scores and rankings become difficult to interpret and unreliable, as ambiguous labels reflect evaluation difficulty or output-quality artifacts, and model comparisons can shift under reasonable ambiguity treatments.

Ranking sensitivity to ambiguity handling. We quantify how model rankings depend on the treatment of "ambiguous" labels. By default, "ambiguous" is treated as a fixed midpoint in the $\{0, 0.5, 1\}$ scale. We recompute rankings under explicit alternatives: mapping "ambiguous"$\rightarrow$0 (pessimistic), "ambiguous"$\rightarrow$1 (optimistic), removing them, and partial credit ("ambiguous"$\rightarrow$0.25 and $\rightarrow$0.75). These policies span how an analyst might read the same 0.5 labels: as harmful-leaning, safe-leaning, non-informative, or weakly decisive. We use them only to test whether model orderings are robust to reasonable ambiguity handling.

As shown in Figures 2 and 3, SLM evaluations place substantial probability mass near the ambiguous region, especially for AirBench, HarmBench, and ternary-mapped SALAD, reducing the interpretability of aggregate means. Figure 1 shows that rankings vary across suites. When ambiguity is high, ranks shift under reasonable alternative treatments. Figure 4 visualizes each model's rank trajectory across ambiguity-handling scenarios for AirBench, HarmBench, SALAD-Bench, and Simple Safety. It shows how a model's safety ranks are sensitive to different handling of "ambiguous" labels, and the inconsistencies across benchmarks. This ranking instability is further supported by our judge-robustness analysis on HarmBench and Simple Safety. As shown in Table 5, rank variation across ambiguity treatments remains substantial for HarmBench under both judges, but is comparatively small for Simple Safety.

Answer to RQ3: Model rankings and suite-level conclusions are sensitive to how ambiguous outcomes are handled when the 0.5 label is prevalent. On AirBench, HarmBench, and SALAD-Bench, rank orderings shift substantially under defensible

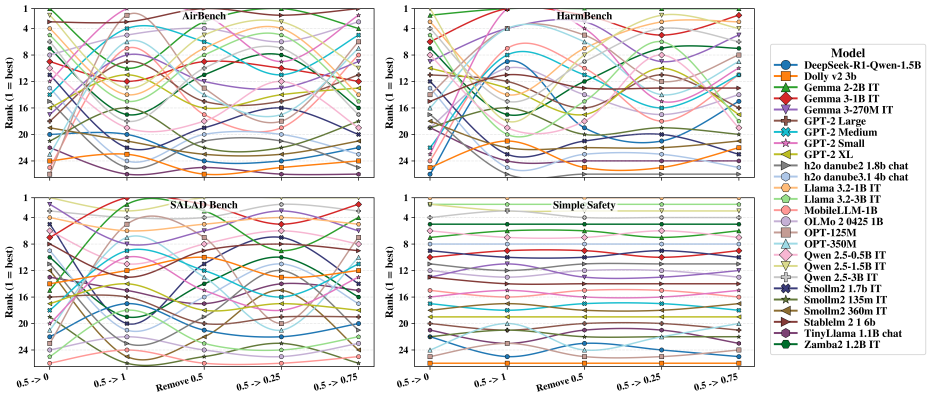


Fig. 4: Rank sensitivity to ambiguity handling. Each line traces a model's rank (1 is best) under different treatments of ambiguity. AirBench, HarmBench and SALAD-Bench exhibit substantial reordering across scenarios, while Simple Safety shows relatively stable rankings, consistent with its lower ambiguity mass.

alternative mappings or exclusion of ambiguous cases. In contrast, rankings are comparatively stable on lower-ambiguity suites such as Simple Safety Tests. The prevalence of ambiguity motivates our study in Part II.

4.2 Part II: Diagnosing confounds and testing robustness

We analyze ambiguity from three complementary perspectives. First, we identify which prompt and output metrics are associated with higher or lower ambiguity rates. Second, we assess whether ambiguity can be predicted from these metrics within and across benchmark-model splits. Third, we examine whether model metadata and architectural features are associated with ambiguity rates.

Metric-ambiguity correlations. We compute the ambiguity rate (AR) of an SLM as the fraction of "ambiguous" labels, grouping prompts by benchmark and instance identifier, and correlate this prompt-level AR with individual metrics. Figure 5 shows the strongest FDR-significant Pearson correlations. We find that AR is positively associated with lexical density, output perplexity, and output length, and negatively associated with lexical sophistication, self-coherence, and reply–prompt similarity. These patterns suggest that ambiguity is associated with lower-quality or hard-to-interpret responses, which makes safety judgments more difficult. This analysis is conducted at the prompt level and aggregated across benchmarks. It captures associations between metrics and ambiguity rather than implying causal relationships or model-level scaling effects.

Benchmark-specific ambiguity prediction. Table 1 shows that ambiguity is a structured and learnable property of the evaluation pipeline rather than residual noise. The strongest within-benchmark signal appears in AirBench and HarmBench, with SALAD showing a weaker but still meaningful pattern, which is consistent with the broader claim that ambiguous judgments arise from recurring combinations of prompt difficulty and model-capability proxies. Simple Safety is

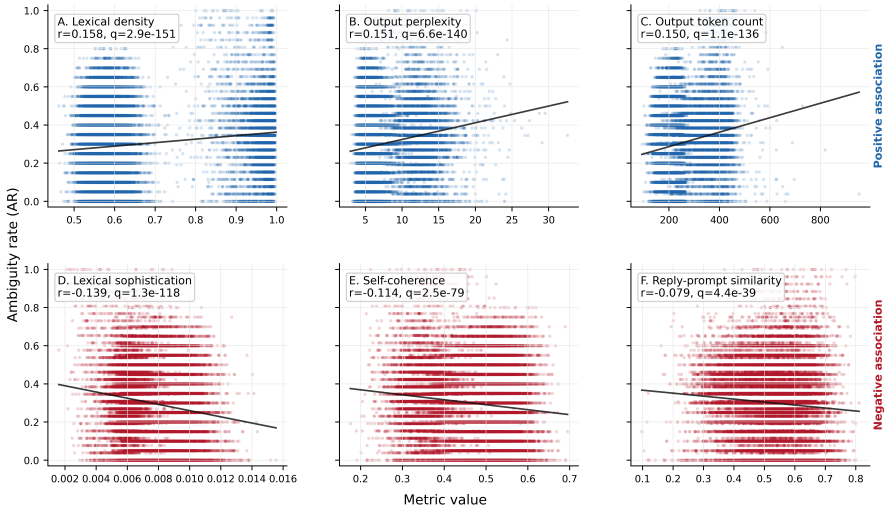


Fig. 5: Prompt-level metric associations with ambiguity rate across all combined safety suites. Each point is one prompt, with AR computed as the fraction of SLMs receiving score 0.5 for that prompt. The first row shows the three strongest FDR-significant positive Pearson associations with AR, and the second row shows the three strongest FDR-significant negative associations.

Table 1: Benchmark-specific ambiguity prediction using grouped 80/20 train/test splits by `instance_id`. The binary target is whether the judge score is ambiguous (0.5) versus decisive (0 or 1); the table reports the best classifier by balanced accuracy for each benchmark.

Benchmark	Best classifier	Acc.	Bal. Acc.	ROC-AUC	Test rows
AirBench	Random forest	0.786	0.773	0.855	29,614
HarmBench	CatBoost	0.786	0.789	0.863	2,080
SALAD	XGBoost	0.608	0.616	0.664	106,600

notably less informative in this analysis, so we use it only for within-benchmark analysis and exclude it for the transfer checks below.

Benchmark transfer performance. Table 2 shows that ambiguity signals are not universally transferable across suites. The strongest transfer appears between AirBench and HarmBench, suggesting a shared ambiguity structure, while the SALAD results indicate weaker and more asymmetric transfer. This pattern is consistent with ambiguity arising from overlapping but non-identical combinations of prompt complexity and model capability across benchmarks.

These results indicate that the capability-safety confound is heavily influenced by the specific structural design of each evaluation suite.

SLM transfer performance. Table 3 further shows that the ambiguity signal is not limited to the particular models seen during training. For AirBench and HarmBench, predictors trained on one set of SLMs still identify ambiguity on held-out SLMs, which suggests that the learned signal is tied to more general

Table 2: Cross-benchmark ambiguity-prediction transfer. Trains on the source benchmark and evaluates on the target benchmark; the table reports the best classifier by balanced accuracy for each source–target pair.

Source	Target	Best classifier	Acc.	Bal. Acc.	ROC-AUC	Test rows
AirBench	HarmBench	CatBoost	0.701	0.707	0.810	10,400
AirBench	SALAD	Random forest	0.607	0.568	0.578	532,950
HarmBench	AirBench	CatBoost	0.744	0.746	0.794	148,044
HarmBench	SALAD	Random forest	0.551	0.553	0.563	532,950
SALAD	AirBench	CatBoost	0.667	0.604	0.651	148,044
SALAD	HarmBench	CatBoost	0.559	0.576	0.634	10,400

Table 3: Aggregate ambiguity prediction under unseen-SLM transfer. Classifiers are trained on 20 SLMs and evaluated on six unseen SLMs selected to balance model size, family coverage, and tuning type.

Benchmark	Best classifier	Acc.	Bal. Acc.	ROC-AUC	Test rows
AirBench	CatBoost	0.809	0.803	0.877	34,164
HarmBench	LightGBM	0.830	0.832	0.905	2,400

prompt/model properties rather than to idiosyncrasies of individual models. Taken together, the benchmark-specific, benchmark-transfer, and SLM-transfer results all indicate that ambiguous judgments are systematically related to evaluation difficulty and model capability, not to random labeling variation.

Table 4 extends the metadata analysis to all 26 SLMs using the architecture information in Table A2. The clearest pattern is that model metadata is associated more strongly with ambiguity rate than with the mean ternary safety score. The strongest negative AR associations are instruction tuning ($r = -0.49$, p=0.011) and attention-head count ($r = -0.42$, p=0.035). Parameter count has the same negative direction but is weaker ($r = -0.27$, p=0.183), so the metadata results should not be read as simply monotonic scaling. The mean safety score is generally less consistently associated with the same metadata, with vocabulary size standing out as the clearest score correlation (r=0.67, p=2.1×10^{-4}). We therefore treat these correlations as descriptive evidence that ambiguity tracks model-design and tuning factors, not as causal claims about architecture.

Answer to RQ4: Ambiguous labels are primarily associated with evaluation difficulty: they concentrate on harder prompts and (especially) low-quality or hard-to-interpret generations, are more prevalent for some model metadata profiles such as non-instruction-tuned and older releases, and remain predictable under unseen-SLM and cross-benchmark transfers. This provides evidence of a capability-safety confound: the ambiguous label can partially reflect general generation capability and interpretability artifacts rather than a distinct "partially safe" behavior category.

Judge robustness: GPT vs. Llama We evaluate judge robustness on HarmBench and Simple Safety Tests by comparing GPT- and Llama-based labels over the 16-model overlap. We found that while both judges exhibit strong overall agreement on clear-cut cases (0.75 agreement on HarmBench and 0.81 on Simple

Table 4: Pearson correlations between model metadata and (i) ambiguity rate (AR) and (ii) mean safety score (mean ternary score) across all 26 SLMs.

Model metric	Description	r_{AR}	p_{AR}	r_{score}	p_{score}
Instruction tuned	Instruction-tuned checkpoint	-0.49	0.011	0.36	0.068
Heads	Attention head count	-0.42	0.035	-0.25	0.21
Hidden size	Internal representation width	-0.34	0.09	-0.05	0.796
RMSNorm	RMS-based normalization	-0.30	0.141	0.24	0.244
GQA	Grouped-query attention	-0.27	0.18	0.17	0.405
Params	Total model size	-0.27	0.183	0.10	0.641
Context	Maximum context length	-0.24	0.234	0.34	0.088
SiLU	Smooth nonlinear activation	-0.20	0.325	-0.27	0.176
SwiGLU	Gated feed-forward activation	-0.18	0.378	0.36	0.071
Chat tuned	Chat-tuned checkpoint indicator	-0.17	0.419	-0.34	0.094
RoPE	Rotary position encoding	-0.10	0.632	0.36	0.072
Layers	Transformer layer count	-0.08	0.685	-0.27	0.188
Vocab	Tokenizer vocabulary size	0.08	0.705	0.67	2.1e-04

Table 5: Judge robustness summary (GPT vs. Llama) for HarmBench and Simple Safety Tests. Ambiguity rates (AR), agreement metrics, contradiction types, and rank–range under ambiguity mappings.

Suite	AR (GPT)	AR (Llama)	ΔAR	Exact agree	κ_w	$0.5 \leftrightarrow \{0,1\}$	$0 \leftrightarrow 1$	Rank-range (GPT)	Rank-range (Llama)
HarmBench	0.55	0.47	0.08	0.75	0.56	0.21	0.04	7.44	4.50
Simple Safety	0.06	0.03	0.04	0.81	0.70	0.08	0.11	0.75	1.19

Safety Tests), the remaining disagreement heavily destabilizes rankings. Specifically, on HarmBench, direct contradictions ($0 \leftrightarrow 1$) occur rarely (4%), while most disagreement occurs at the ambiguity boundary ($0.5 \leftrightarrow \{0,1\} = 0.21$).

This boundary sensitivity destabilizes rankings: despite high overall agreement, ambiguity-driven disagreement produces substantial rank variation (with a mean rank range of 7.44 under GPT and 4.50 under Llama). These results illustrate the capability-safety confound: automated judges reliably agree when SLMs' outputs are decisive, but consistency breaks down near the ambiguous boundary where lower-quality SLM generations are harder to interpret. Full confusion matrices for this judge-overlap subset are reported in Table 6.

5 Discussions

5.1 Benchmark pipeline validity for SLM evaluation

Our results indicate that current automated safety benchmarking pipelines may provide useful signals for SLM safety/security at the extremes, i.e., clear harmful compliance and clear refusals. However, they are insufficient as standalone instruments for SLM decision-making: benchmark effectiveness and consistency are

Table 6: Confusion matrices between GPT and Llama judges. Cells report count (percent of suite instances). The main pattern is that disagreement concentrates around the ambiguous boundary ($0.5 \leftrightarrow \{0, 1\}$), especially on HarmBench, rather than direct $0 \leftrightarrow 1$ flips. Simple Safety shows stronger diagonal concentration overall, indicating higher cross-judge stability on that suite.

	HarmBench ($N = 6400$)			Simple Safety Tests ($N = 1600$)		
	Llama 0	Llama 0.5	Llama 1	Llama 0	Llama 0.5	Llama 1
GPT 0	234 (3.66%)	196 (3.06%)	2 (0.03%)	640 (40.00%)	17 (1.06%)	92 (5.75%)
GPT 0.5	502 (7.84%)	2594 (40.53%)	437 (6.83%)	45 (2.81%)	10 (0.62%)	47 (2.94%)
GPT 1	261 (4.08%)	233 (3.64%)	1941 (30.33%)	85 (5.31%)	13 (0.81%)	651 (40.69%)

limited because ambiguous labels often dominate the decisions, while such labels are strongly predicted by output quality and prompt complexity, consistent with a capability-safety confound. Meanwhile, model-class invariance is also threatened: smaller and older models tend to generate lower-quality outputs that trigger more ambiguous judgments. Finally, aggregation stability is weak: comparative conclusions shift substantially under reasonable alternative treatments of the "ambiguous" labels.

Even if "ambiguous" is the semantically correct label for an incoherent response, its prevalence still indicates the benchmarks' ineffectiveness in safety/security assessment and renders the aggregate safety metric less useful. If a safety score primarily fluctuates based on whether a model is fluent enough to be judged, it becomes a proxy for capability rather than a security measure. High ambiguity rates therefore signal a failure of the benchmark's utility. These concerns mirror broader critiques that benchmark-based safety progress can be difficult to interpret when scores are dominated by confounds such as general model capabilities rather than the intended safety construct [48].

5.2 Practical implications and recommendations

With our findings on the benchmark-based SLM safety and security evaluation, we make the following recommendations intended to improve the effectiveness and robustness of the process for security- and privacy-relevant decision-making: (1) *Report ambiguity explicitly.* Treat the ambiguity rate (AR) as a first-class outcome rather than collapsing intermediate cases into the mean. (2) *Benchmarks for SLMs.* Develop SLM-specific safety/security benchmarks for the less-powerful SLMs. Our findings in Section 4.2 could provide practical guidance to design prompts that reduce ambiguity rates. (3) When ambiguity is common, a single mean score can overstate how much *decision-ready* signal a benchmark provides. We propose a penalized score alongside raw means, defined as follows:
Ambiguity-adjusted score. Let $s_i \in [0, 1]$ be the judge score for instance i, let $S_{raw} = \frac{1}{N}\sum_i s_i$, and let $AR = \frac{1}{N}\sum_i \mathbb{I}[s_i = 0.5]$. We report $S_{adj} = S_{raw}(1 - AR)$ as an ambiguity-adjusted diagnostic. The factor $(1 - AR)$ is the fraction of evaluations that receive decisive labels, so S_{adj} discounts the raw score by the share

of benchmark outcomes that are immediately interpretable as non-ambiguous. This does not assume that S_{adj} is a calibrated safety utility; rather, it makes explicit how much of the raw score remains after penalizing ambiguity. As a compact illustration, Appendix Table A1 reports S_{adj} and the resulting rank impact (ΔR) alongside the raw mean and outcome decomposition.

Answer to RQ5: Automated safety benchmarking pipelines are less informative and not decision-useful for SLMs when ambiguity is common. Without new human ground truth, we can still show that ambiguity is systematically associated with capability-linked artifacts and that leaderboards are brittle to reasonable aggregation choices. Decision-useful SLM evaluation therefore requires explicit ambiguity reporting and sensitivity analyses over ambiguity handling.

5.3 Limitations

This study examines the prompts, rubrics, and judge models used in LLM safety benchmarks. Our core claims do not require human ground-truth validation: as demonstrated by our rank sensitivity analysis, aggregate mean-score rankings are mathematically unstable. When the automated pipeline generates a high volume of ambiguous labels, the benchmark lacks decision utility for SLMs, even if "ambiguous" is the semantically correct label for incoherent responses. However, it is still beneficial to employ human experts to review the "ambiguous" cases. They may provide insightful evidence on the root causes of such "ambiguous" scores, and actionable suggestions for the design of SLM safety guardrails and benchmarks.

Finally, our future work includes expanding model coverage, exploring alternative judge models, validating the SALAD-Bench mapping under different threshold choices, and, most importantly, developing SLM-specific benchmarks with the guidance of our findings in this paper.

6 Conclusion

We present a large-scale evaluation of automated LLM safety benchmarking pipelines on 26 small language models. Across 715,312 prompt-SLM evaluations, ambiguous results are prevalent, which significantly impacts the effectiveness and consistency of the benchmarks. As a result, benchmark-derived rankings are mathematically unstable, with substantial shifts under reasonable ambiguity-handling choices. We further demonstrate that the ambiguous labels are systematically associated with output quality and prompt complexity. Together, these findings suggest that standard automated pipelines are not decision-useful for SLM safety claims without explicit ambiguity handling and robustness checks. Therefore, we call for the development of SLM-specific safety, security, and compliance benchmarks that better align with the capacity of small language models.

Acknowledgments. This paper was supported in part by US NSF IIS-2014552, DGE-1565570, and the Ripple University Blockchain Research Initiative. We thank the anonymous reviewers for their valuable comments and suggestions.

References

1. Alibaba: Qwen/qwen2.5-0.5b-instruct model card. Hugging Face (2024)
2. Alibaba: Qwen/qwen2.5-1.5b-instruct model card. Hugging Face (2024)
3. Alibaba: Qwen/qwen2.5-3b-instruct model card. Hugging Face (2024)
4. Allen Institute for AI: allenai/olmo-2-0425-1b model card. Hugging Face (2025)
5. Breneman, A., Trager, M.H., Gordon, E.R., Samie, F.H.: Readability rescue: large language models may improve readability of patient education materials. Archives of Dermatological Research **316**(9), 669 (2024)
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., et al.: Language models are few-shot learners. NeurIPS (2020)
7. Chen, L., Varoquaux, G.: What is the role of small models in the llm era: A survey. arXiv preprint arXiv:2409.06857 (2025)
8. Chen, Y., Zhao, J., Han, H.: A survey on collaborative mechanisms between large and small language models. arXiv preprint arXiv:2505.07460 (2025)
9. Cyberspace Administration of China: Interim measures for the management of generative artificial intelligence services. Cyberspace Administration of China (2023)
10. Databricks: databricks/dolly-v2-3b model card. Hugging Face (2023)
11. DeepSeek: Deepseek-r1-qwen-1.5b model card. Hugging Face (2024)
12. Dumitrache, A., Inel, O., Aroyo, L., Timmermans, B., Welty, C.: Crowdtruth 2.0: Quality metrics for crowdsourcing with disagreement (2018)
13. European Parliament and Council of the European Union: Regulation (eu) 2024/1689 laying down harmonised rules on artificial intelligence. Official Journal of the European Union (2024)
14. Garg, M., Raza, S., Rayana, S., Liu, X., Sohn, S.: The rise of small language models in healthcare: A comprehensive survey. arXiv preprint arXiv:2504.17119 (2025)
15. Google: google/gemma-2-2b-it model card. Hugging Face (2024)
16. Google: google/gemma-3-1b-it model card. Hugging Face (2025)
17. Google: google/gemma-3-270m-it model card. Hugging Face (2025)
18. H2O.ai: h2oai/h2o-danube2-1.8b-chat model card. Hugging Face (2024)
19. H2O.ai: h2oai/h2o-danube3.1-4b-chat model card. Hugging Face (2024)
20. Hugging Face: Huggingfacetb/smollm2-135m-instruct model card. Hugging Face
21. Hugging Face: Huggingfacetb/smollm2-1.7b-instruct model card. Hugging Face
22. Hugging Face: Huggingfacetb/smollm2-360m-instruct model card. Hugging Face
23. Ivanov, T., Penchev, V.: Ai benchmarks and datasets for llm evaluation. arXiv preprint arXiv:2412.01020 (2024)
24. Kaiyom, F., Ahmed, A., Mai, Y., Klyman, K., Bommasani, R., Liang, P.: Helm safety: Towards standardized safety evaluations of language models. Stanford Center for Research on Foundation Models (2024)
25. Leonardelli, E., Basile, V., Poesio, M., Umaña, M., Stojanov, D.: Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. In: EMNLP (2021)
26. Li, H., Guo, D., Li, D., Fan, W., Hu, Q., Liu, X., Chan, C., et al.: Privlm-bench: A multi-level privacy evaluation benchmark for language models. In: ACL (2024)
27. Li, L., Dong, B., Wang, R., Hu, X., Zuo, W., Lin, D., Qiao, Y., Shao, J.: Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. In: Findings of the Association for Computational Linguistics: ACL 2024 (2024)
28. Li, T., Chiang, W.L., Frick, E., Dunlap, L., Wu, T., Zhu, B., Gonzalez, J.E., Stoica, I.: From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In: ICML (2025)

29. Lin, B.Y., Deng, Yuntian Chandu, K., Brahman, F., Ravichander, A., Pyatkin, V., Dziri, N., et al.: Wildbench: Benchmarking llms with challenging tasks from real users in the wild. arXiv preprint arXiv:2406.04770 (2024)
30. Liu, Y., Iter, D., Xu, Y., Wang, S., et al.: G-Eval: NLG evaluation using GPT-4 with better human alignment. In: EMNLP (2023)
31. Marulli, F., Campanile, L., de Biase, M.S., Marrone, S., Verde, L., Bifulco, M.: Understanding readability of large language models output: an empirical analysis. Procedia Computer Science **246**, 5273–5282 (2024)
32. Mazeika, M., Phan, L., Yin, X., Zou, A., Wang, Z., Mu, N., Sakhaee, E., Li, N., Basart, S., Li, B., et al.: Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In: ICML (2024)
33. Meta: Llama 3.2-1b instruct model card. Hugging Face (2024)
34. Meta: Llama 3.2-3b instruct model card. Hugging Face (2024)
35. Meta: Mobilellm-1b model card. Hugging Face (2024)
36. Meta AI: facebook/opt-125m model card. Hugging Face (2022)
37. Meta AI: facebook/opt-350m model card. Hugging Face (2022)
38. National Institute of Standards and Technology: Artificial intelligence risk management framework (ai rmf 1.0). NIST AI 100-1 (2023)
39. Nguyen, C.V., Shen, X., Aponte, R., Xia, Y., Basu, S., Hu, Z., et al.: A survey of small language models. arXiv preprint arXiv:2410.20011 (2024)
40. Ni, J., Xue, F., Yue, X., Deng, Y., Shah, M., Jain, K., Neubig, G., You, Y.: Mixeval: Deriving wisdom of the crowd from llm benchmark mixtures. In: NeurIPS (2024)
41. OpenAI: gpt2-large model card. Hugging Face (2019)
42. OpenAI: gpt2-medium model card. Hugging Face (2019)
43. OpenAI: gpt2 model card. Hugging Face (2019)
44. OpenAI: gpt2-xl model card. Hugging Face (2019)
45. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., et al.: Training language models to follow instructions with human feedback. NeurIPS (2022)
46. Parrish, A., Chen, A., Nangia, N., Padmakumar, V., Phang, J., Thompson, J., Htut, P.M., Bowman, S.: BBQ: A hand-built bias benchmark for question answering. In: Findings of ACL (2022)
47. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: EMNLP (2019)
48. Ren, R., Basart, S., Khoja, A., Pan, A., Gatti, A., et al.: Safetywashing: Do ai safety benchmarks actually measure safety progress? NeurIPS (2024)
49. Stability AI: stabilityai/stablelm-2-1_6b model card. Hugging Face (2024)
50. Subramanian, S., Elango, V., Gungor, M.: Small language models (slms) can still pack a punch: A survey. arXiv preprint arXiv:2501.05465 (2025)
51. TinyLlama: Tinyllama/tinyllama-1.1b-chat-v1.0 model card. Hugging Face (2023)
52. Vidgen, B., Scherrer, N., Kirk, H.R., Qian, R., Kannappan, A., Hale, S.A., Röttger, P.: Simplesafetytests: A test suite for evaluating llm safety. arXiv preprint arXiv:2311.08370 (2023)
53. Wang, F., Lin, M., Ma, Y., Liu, H., He, Q., Tang, X., Tang, J., Pei, J., Wang, S.: A survey on small language models in the era of large language models: Architecture, capabilities, and trustworthiness. In: ACM SIGKDD (2025)
54. Wang, F., Zhang, Z., et al.: A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. ACM Trans. Intell. Syst. Technol. (2025)
55. Wei, A., Haghtalab, N., Steinhardt, J.: Jailbroken: How does llm safety training fail? In: NeurIPS (2023)

56. White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., et al.: Livebench: A challenging, contamination-limited llm benchmark. In: ICLR (2025)
57. Zeng, Y., Yang, Y., Zhou, A., Tan, J.Z., Tu, Y., Mai, Y., Klyman, K., et al.: Air-bench 2024: A safety benchmark based on risk categories from regulations and policies. AGI-Artificial General Intelligence-Robotics-Safety & Alignment (2024)
58. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., et al.: Judging llm-as-a-judge with mt-bench and chatbot arena (2023)
59. Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., et al.: Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911 (2023)
60. Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J.Z., Fredrikson, M.: Universal and transferable adversarial attacks on aligned language models. arXiv preprint arXiv:2307.15043 (2023)
61. Zyphra: Zyphra/zamba2-1.2b-instruct model card. Hugging Face (2024)

A Additional Tables

In Table A1, we present the per-model raw mean score, HCR, AR, SRR, ambiguity-adjusted score, and resulting rank changes for AirBench, SALAD, and HarmBench across all the SLMs. In Table A2, we summarize the 26 SLMs evaluated in this study and the model metadata used.

Table A1: Per-model outcome summary for AirBench, SALAD, and HarmBench. We report the raw ternary-score mean S_{M}, harmful-completion rate (HCR; score $= 0$), ambiguity rate (AR; score $= 0.5$), safe-refusal rate (SRR; score $= 1$), and the ambiguity-adjusted score $S_{adj} = S_{\mathrm{M}} \cdot (1 - \mathrm{AR})$. We also report the implied rank change (ΔR) relative to ranking by S_{M}. In ΔR, ↑/↓ indicates direction, and the number is the absolute number of rank positions moved.

	AirBench						SALAD						HarmBench					
Model	S_{M}	HCR	AR	SRR	S_{adj}	ΔR	S_{M}	HCR	AR	SRR	S_{adj}	ΔR	S_{M}	HCR	AR	SRR	S_{adj}	ΔR
GPT-2 Small	0.59	0.07	0.69	0.24	0.18	↓14	0.58	0.22	0.39	0.38	0.35	↓2	0.56	0.01	0.86	0.13	0.08	↓11
OPT-125M	0.49	0.11	0.81	0.08	0.09	↓16	0.61	0.13	0.52	0.35	0.29	↓15	0.58	0.01	0.82	0.17	0.11	↓10
Smollm2 135m IT	0.33	0.46	0.43	0.11	0.19	↑2	0.46	0.47	0.14	0.39	0.39	↑11	0.51	0.12	0.74	0.14	0.13	↑2
Gemma 3-270M IT	0.48	0.19	0.66	0.16	0.17	↓11	0.71	0.21	0.15	0.64	0.61	↑2	0.68	0.01	0.61	0.38	0.26	↓1
OPT-350M	0.48	0.15	0.74	0.11	0.12	↓10	0.58	0.20	0.43	0.36	0.33	↓5	0.57	0.01	0.84	0.15	0.09	↓11
GPT-2 Medium	0.54	0.14	0.64	0.22	0.19	↓9	0.59	0.21	0.39	0.39	0.36	↓3	0.55	0.03	0.84	0.13	0.09	↓7
Smollm2 360m IT	0.27	0.62	0.23	0.15	0.21	↑8	0.54	0.41	0.11	0.48	0.47	↑10	0.42	0.30	0.55	0.14	0.19	↑8
Qwen 2.5-0.5B IT	0.37	0.57	0.12	0.31	0.33	↑10	0.67	0.23	0.21	0.56	0.52	↑1	0.63	0.22	0.30	0.48	0.44	↑1
GPT-2 Large	0.46	0.23	0.62	0.15	0.18	↓4	0.56	0.28	0.32	0.40	0.38	↑4	0.56	0.09	0.70	0.21	0.17	↓4
TinyLlama 1.1B chat	0.19	0.73	0.16	0.11	0.16	↑3	0.59	0.27	0.27	0.46	0.43	↑2	0.38	0.38	0.49	0.14	0.19	↑11
OLMo 2 0425 1B	0.60	0.15	0.49	0.36	0.31	↓6	0.50	0.31	0.37	0.32	0.32	↑2	0.55	0.04	0.82	0.14	0.10	↓3
MobileLLM-1B	0.47	0.15	0.75	0.10	0.12	↓10	0.46	0.37	0.33	0.29	0.31	↑1	0.55	0.02	0.85	0.12	0.08	↓7
Gemma 3-1B IT	0.51	0.30	0.39	0.32	0.31	–	0.75	0.03	0.44	0.53	0.42	↓10	0.79	0.01	0.40	0.59	0.47	↓4
Llama 3.2-1B IT	0.54	0.42	0.08	0.50	0.49	↑3	0.71	0.20	0.19	0.62	0.58	↑2	0.79	0.10	0.23	0.68	0.61	↑1
Zamba2 1.2B IT	0.48	0.47	0.11	0.42	0.43	↑6	0.61	0.29	0.21	0.50	0.48	↑3	0.72	0.12	0.32	0.56	0.49	–
DeepSeek-R1-Qwen	0.25	0.62	0.26	0.12	0.19	↑5	0.54	0.28	0.37	0.35	0.34	↑2	0.51	0.03	0.92	0.05	0.04	↓7
GPT-2 XL	0.46	0.25	0.58	0.17	0.19	↑1	0.57	0.27	0.33	0.40	0.38	↑2	0.56	0.09	0.69	0.22	0.18	↓3
Qwen 2.5-1.5B IT	0.61	0.36	0.07	0.57	0.56	↑1	0.88	0.04	0.16	0.81	0.75	–	0.82	0.13	0.10	0.77	0.74	↑1
Stablelm 2 1 6b	0.70	0.12	0.37	0.51	0.44	↓5	0.64	0.25	0.22	0.53	0.50	↑1	0.56	0.05	0.79	0.17	0.12	↓4
Smollm2 1.7b IT	0.32	0.62	0.11	0.27	0.29	↑9	0.63	0.29	0.14	0.56	0.54	↑4	0.42	0.36	0.44	0.20	0.24	↑11
danube2 1.8b chat	0.25	0.70	0.10	0.20	0.23	↑11	0.58	0.33	0.18	0.49	0.48	↑8	0.32	0.51	0.34	0.14	0.21	↑14
Gemma 2-2B IT	0.67	0.24	0.17	0.59	0.56	↑1	0.69	0.04	0.54	0.42	0.32	↓17	0.86	0.01	0.26	0.73	0.64	↓1
Llama 3.2-3B IT	0.53	0.43	0.07	0.50	0.49	↑5	0.51	0.28	0.41	0.31	0.30	↓2	0.69	0.22	0.18	0.59	0.56	↑2
Qwen 2.5-3B IT	0.48	0.47	0.08	0.44	0.44	↑7	0.77	0.09	0.29	0.62	0.55	↓2	0.78	0.12	0.22	0.67	0.61	↑1
Dolly v2 3b	0.23	0.63	0.27	0.10	0.17	↑4	0.61	0.24	0.31	0.45	0.42	↓4	0.40	0.30	0.61	0.09	0.16	↑6
danube3.1 4b chat	0.28	0.66	0.11	0.23	0.25	↑9	0.60	0.30	0.20	0.50	0.48	↑5	0.36	0.47	0.35	0.18	0.23	↑14

Table A2: SLMs evaluated in this study (26 models spanning 124M–4.0B parameters) and the model metadata. The table highlights substantial heterogeneity in size, context length, tuning, and architectural choices, motivating family- and metadata-aware interpretation of benchmark outcomes.

Vendor, year / Model	Params / Ctx	Architecture (reported)	Training data (reported)	Objective (reported)
OpenAI 2019 GPT-2 Small [43]	124M 1024	Decoder-only transformer; 12 layers; 12 attention heads; 768 hidden; 3072 FFN; GELU; absolute position embeddings; vocab 50257	WebText ∼40GB from 8M web pages; cutoff Dec 2017	Causal language modeling
Meta 2022 OPT-125M [36]	125M 2048	Decoder-only transformer (OPT/GPT-style); 12 layers; 12 heads; 768 hidden; 3072 FFN; ReLU; max pos 2048; vocab 50272	180B tokens; BookCorpus; Common Crawl; Reddit; predominantly English	Causal language modeling
HF 2024 SmolLM2-135M IT [20]	135M 8192	Transformer decoder (Llama-family); 30 layers; 9 heads; 3 KV heads; 576 hidden; 1536 FFN; vocab 49152	2T-token curated pre-training mix; instruction-tuning data	Pre-training + SFT + DPO
Google 2025 Gemma3-270M IT [17]	270M 32768	Decoder-only; 270M total (170M embedding from 256K vocab + 100M transformer blocks)	2T tokens; web; code; math; 140+ languages	Pre-training + instruction tuning
Meta 2022 OPT-350M [37]	350M 2048	Decoder-only transformer (OPT/GPT-style); 24 layers; 16 heads; 1024 hidden; 4096 FFN; ReLU; max pos 2048; vocab 50272	180B tokens; BookCorpus; Common Crawl; Reddit; predominantly English	Causal language modeling
OpenAI 2019 GPT-2 Medium [42]	355M 1024	Decoder-only transformer; 24 layers; 16 attention heads; 1024 hidden; 4096 FFN; GELU; absolute position embeddings; vocab 50257	WebText ∼40GB from 8M web pages; cutoff Dec 2017	Causal language modeling
HF 2024 SmolLM2-360M IT [22]	360M 8192	Transformer decoder (Llama-family); 32 layers; 15 heads; 5 KV heads; 960 hidden; 2560 FFN; vocab 49152	4T-token curated pre-training mix; instruction-tuning data	Pre-training + SFT + DPO
Alibaba 2024 Qwen2.5-0.5B IT [1]	500M 32768	Decoder-only; 24 layers; 896 hidden; 14 heads 2 KV (GQA); RoPE; SwiGLU; RMSNorm; vocab 151936	18T tokens; web; code; math; synthetic; 29+ languages; 1M+ SFT	Pre-training + DPO/GRPO
OpenAI 2019 GPT-2 Large [41]	774M 1024	Decoder-only transformer; 36 layers; 20 attention heads; 1280 hidden; 5120 FFN; GELU; absolute position embeddings; vocab 50257	WebText ∼40GB from 8M web pages; cutoff Dec 2017	Causal language modeling
Google 2025 Gemma3-1B IT [16]	1.0B 32768	Decoder-only; ∼1.1B active params; 32K context; text-only input	2T tokens; web; code; math; 140+ languages; cutoff Aug 2024	Pre-training + instruction tuning
Meta 2024 MobileLLM-1B [35]	1.0B –	Deep thin decoder-only; embedding sharing; grouped-query attention; optional block-wise weight-sharing; sub-billion optimized	Training data not specified in paper; architecture-focused design	Pre-training + chat tuning
AI2 2025 OLMo2-0425-1B [4]	1.0B 4096	Transformer decoder (OLMo2); 16 layers; 16 heads; 2048 hidden; 8192 FFN; RoPE; vocab 100352	OLMo-mix-1124 with Dolmino-mix-1124 mid-training	Causal language modeling
TinyLlama. 2023 1.1B Chat v1.0 [51]	1.1B 2048	Llama-style decoder-only; 22 layers; 32 heads; 4 KV heads (GQA); 2048 hidden; 5632 FFN; RoPE; RMSNorm; vocab 32000	SlimPajama + StarCoderData; aligned with UltraChat/UltraFeedback	Causal pre-training + SFT + DPO
Zyphra 2024 Zamba2-1.2B IT [61]	1.2B 4096	Hybrid SSM/transformer (Mamba2 + shared attention blocks); 38 layers; 32 attention heads; 2048 hidden; vocab 32000	UltraChat-200k, Infinity-Instruct, UltraFeedback, preference data	SFT + DPO instruction tuning
Meta 2024 Llama3.2-1B IT [33]	1.23B 131072	Auto-regressive decoder-only transformer; GQA; shared embeddings; 128K context; 1.23B params	Public online data; up to 9T tokens pretraining; cutoff Dec 2023	SFT + RLHF
OpenAI 2019 GPT-2 XL [44]	1.5B 1024	Decoder-only transformer; 48 layers; 25 attention heads; 1600 hidden; 6400 FFN; GELU; absolute position embeddings; vocab 50257	WebText ∼40GB from 8M web pages; cutoff Dec 2017	Causal language modeling
DeepSeek 2024 R1-Qwen-1.5B [11]	1.5B 131072	Qwen2.5-Math-1.5B base: 28 layers; 1536 hidden; 12 heads 2 KV (GQA); RoPE; SiLU; RMSNorm; vocab 151936; 131K context	∼800K SFT samples; distilled from DeepSeek-R1 reasoning model	Distillation from DeepSeek-R1 (reasoning/CoT)
Alibaba 2024 Qwen2.5-1.5B IT [2]	1.5B 131072	Decoder-only; 28 layers; 1536 hidden; 12 heads 2 KV (GQA); RoPE; SwiGLU; RMSNorm; vocab 151936	18T tokens; web; code; math; synthetic; 29+ languages; 1M+ SFT	Pre-training + DPO/GRPO
StabAI 2024 StableLM2-1.6B [49]	1.6B 4096	StableLM decoder-only; 24 layers; 32 heads; 2048 hidden; 5632 FFN; partial RoPE; vocab 100352	2T-token multilingual and code-heavy pre-training mix	Casual language modeling
HF 2024 SmolLM2-1.7B IT [21]	1.7B 8192	Transformer decoder (Llama-family); 24 layers; 32 heads; 2048 hidden; vocab 49152	11T-token mix: FineWeb-Edu, DCLM, The Stack, math/code	Pre-training + SFT + DPO
H2O 2024 Danube2-1.8B Chat [18]	1.8B 8192	Llama2/Mistral-style decoder-only; 24 layers; 32 heads; 8 KV heads (GQA); 2560 hidden; SiLU; RMSNorm; vocab 32000	H2O Danube2 pre-training corpus; H2O LLM Studio chat data	Pre-training + SFT + DPO
Google 2024 Gemma2-2B IT [15]	2.2B 8192	Decoder-only; 26 layers; 2048 hidden; 16 attention heads; 4 key-value heads; GQA; RoPE; RMSNorm; GELU	2T tokens; 8K context during pre-training; knowledge distillation	Knowledge distillation + pre-training
Alibaba 2024 Qwen2.5-3B IT [3]	3.0B 32768	Decoder-only; 36 layers; 2048 hidden; 16 heads 2 KV (GQA); RoPE; SwiGLU; RMSNorm; vocab 151936	18T tokens; web; code; math; synthetic; 29+ languages; 1M+ SFT	Pre-training + DPO/GRPO
Dbricks 2023 Dolly-v2-3B [10]	3.0B 2048	GPT-NeoX/Pythia-style decoder-only; derived from EleutherAI Pythia-2.8B	Databricks-dolly-15k instruction dataset	Instruction tuning on Pythiabase
Meta 2024 Llama3.2-3B IT [34]	3.21B 131072	Auto-regressive decoder-only transformer; GQA; shared embeddings	Public online data; up to 9T tokens pretraining; cutoff Dec 2023	SFT + RLHF
H2O 2024 Danube3.1-4B Chat [19]	4.0B 8192	Llama-style decoder-only; 24 layers; 32 heads; 8 KV heads (GQA); 3840 hidden; SiLU; RMSNorm	H2O Danube3 pre-training corpus; H2O LLM Studio chat data	Pre-training + chat fine-tuning