SAGE: Scalable Automated Robustness Augmentation for LLM Knowledge Evaluation

Xiaoyuan Li[1*] **Yuzhe Wang**[1*] **Moxin Li**[3†] **Keqin Bao**[2†] **Rui Men**[2]
Yichang Zhang[2] **Dayiheng Liu**[2] **Wenjie Wang**[1] **Fuli Feng**[1†]
[1]University of Science and Technology of China [2]Alibaba Group
[3]National University of Singapore

Abstract

Large Language Models (LLMs) achieve strong performance on standard knowledge evaluation benchmarks, yet recent work shows that their knowledge capabilities remain brittle under question variants that test the same knowledge in different forms. Robustness augmentation of existing knowledge evaluation benchmarks is therefore necessary, but current LLM-assisted generate-then-verify pipelines are costly and difficult to scale due to low-yield variant generation and unreliable variant verification. We propose **SAGE** (**S**calable **A**utomated **G**eneration of Robustness B**E**nchmarks), a framework for scalable robustness augmentation of knowledge evaluation benchmarks using fine-tuned smaller models. SAGE consists of VariantQual, a rubric-based verifier trained on human-labeled seed data, and VariantGen, a variant generator initialized with supervised fine-tuning and further optimized with reinforcement learning using VariantQual as the reward model. Experiments on HellaSwag show that SAGE constructs a large-scale robustness-augmented benchmark with quality comparable to the human-annotated HellaSwag-Pro at substantially lower cost, while the fine-tuned models further generalize to MMLU without benchmark-specific fine-tuning. Code and dataset are available at: `https://github.com/Mintwyz/SAGE`.

1 Introduction

Large Language Models (LLMs) encode substantial knowledge, which enables them to perform well across a wide range of tasks. Evaluating the knowledge capabilities of LLMs is therefore critical for their trustworthy deployment. Knowledge evaluation benchmarks such as CommonsenseQA [Talmor et al., 2019] and MMLU [Hendrycks et al., 2021] have become widely used standards for LLM evaluation. Current state-of-the-art LLMs, including GPT-4o [Achiam et al., 2023] and Claude-3.5 [Anthropic, 2024], achieve over 90% accuracy on these benchmarks, demonstrating strong knowledge capabilities under standard evaluation settings.

Despite near-saturated performance on standard benchmarks, recent work shows that LLMs remain brittle to question variants [Li et al., 2025a], which preserve the tested knowledge but alter the formulation or reasoning process of the original question. For example, GPT-4o achieves about 90% accuracy on the original HellaSwag [Zellers et al., 2019] but drops to 9% under negative transformations, *i.e.*, from "The lady *will*..." to "The lady *will not*...", revealing a serious but often hidden weakness in LLM knowledge capabilities. These findings highlight the need for robustness augmentation of existing knowledge evaluation benchmarks with such question variants, enabling robust and comprehensive evaluation of LLM knowledge capabilities.

*Equal contribution.
†Corresponding authors.

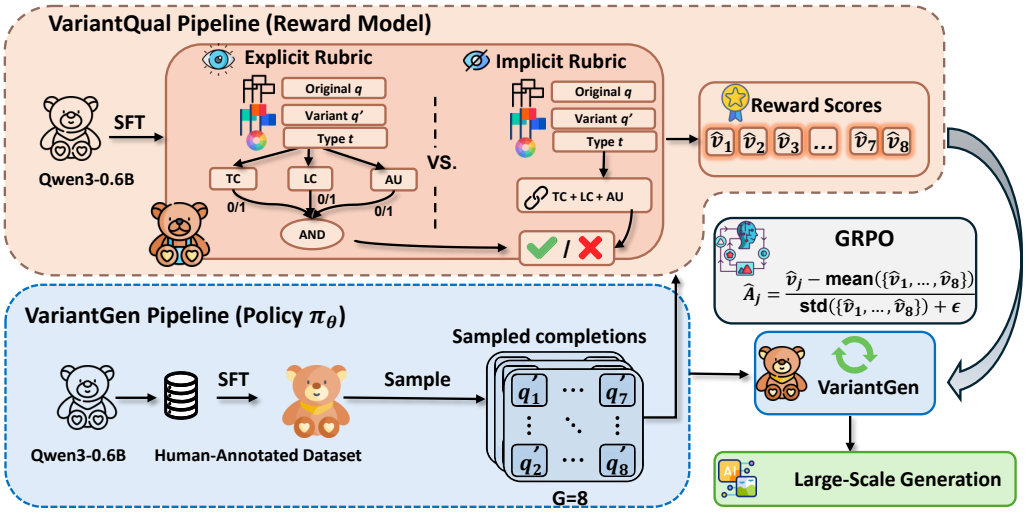

Figure 1: Overview of the SAGE framework. SAGE consists of three stages: SFT of VariantGen and VariantQual on human-annotated dataset, GRPO-based optimization of VariantGen using VariantQual as the reward model, and large-scale benchmark generation with quality filtering.

However, robustness augmentation of large-scale knowledge evaluation benchmarks is both costly and challenging. Existing efforts such as HellaSwag-Pro [Li et al., 2025a] typically adopt an LLM-assisted generate-then-verify pipeline, where strong LLMs generate variants under predefined variant types and reformulation rules, and then judge their validity to filter out invalid candidates. This pipeline faces two critical bottlenecks. First, **variant generation has low yield**: LLMs often produce invalid or low-quality variants, and only 46% of generated candidates survived quality control in HellaSwag-Pro, resulting in substantial wasted token cost. Second, **variant verification is labor-intensive**: prompted LLMs do not reliably determine whether a variant is valid, requiring extensive human annotation. These bottlenecks limit the scalability and cost-effectiveness of robustness augmentation.

To address these bottlenecks, we replace the costly pipeline of prompting strong models with fine-tuned smaller models specialized for generation and verification. Our key idea is to first build a reliable variant verifier, and then use it to improve variant generation. Specifically, we decompose variant quality into three general rubric dimensions: *type compliance, label correctness, and answer uniqueness*, and fine-tune a small verifier on human-labeled seed data to provide reliable quality judgments. Building on this verifier, we further build a small variant generator by first initializing it through supervised fine-tuning (SFT) on human-annotated seed examples, and then optimizing it with reinforcement learning (RL), using the fine-tuned verifier as the reward model to encourage high-quality generation. This design improves cost-efficiency through small-model deployment and improves accuracy through verifier-guided quality control and generator optimization.

In this paper, we propose **SAGE** (**S**calable **A**utomated **G**eneration of Robustness B**E**nchmarks), a framework for scalable automated robustness augmentation of knowledge evaluation benchmarks. SAGE consists of two components: **VariantGen** for generating question variants and **VariantQual** for evaluating their quality. Applying SAGE to HellaSwag [Zellers et al., 2019], we construct a robustness-augmented benchmark with quality comparable to HellaSwag-Pro, while requiring substantially lower cost. Furthermore, these components generalize to MMLU [Hendrycks et al., 2021] without benchmark-specific fine-tuning. Our contributions are three-fold:

- We formalize the task of scalable automated robustness augmentation for knowledge evaluation benchmarks, which generates question variants under predefined variant types.
- We propose SAGE, a framework for scalable automated robustness benchmark augmentation that replaces costly strong-model prompting with fine-tuned small-model generators and evaluators to improve both cost-efficiency and construction accuracy.
- We apply SAGE to HellaSwag to construct a large-scale robustness-augmented benchmark with quality comparable to HellaSwag-Pro at substantially lower cost, and further validate its cross-benchmark generalization on MMLU.

2 Related Work

Robustness and adversarial evaluation. Prior work on LLM robustness spans adversarial attacks on reading comprehension [Jia and Liang, 2017], universal adversarial triggers [Wallace et al., 2019, Chang et al., 2026], spurious correlation analysis [Branco et al., 2021, Geirhos et al., 2020], and consistency evaluation [Storks and Chai, 2021, Johnson and Marasović, 2023, Wang et al., 2026c, Li et al., 2025b]. Balepur et al. [2024] examined negation-based reasoning, while Wu et al. [2024] studied counterfactual task performance. HellaSwag-Pro [Li et al., 2025a] introduced a systematic framework with seven variant types grounded in Bloom's taxonomy, representing the most comprehensive robustness evaluation effort to date. However, all existing robustness benchmarks are manually constructed, limiting their scale and update frequency. SAGE addresses this bottleneck by automating the entire generation and verification pipeline.

Automated data generation. Self-Instruct [Wang et al., 2023] and Evol-Instruct [Xu et al., 2023] demonstrated that LLMs can generate training data for instruction following. Yuan et al. [2023] explored constrained language planning with LLM-generated data. However, these methods target general-purpose training data rather than the specific challenge of generating semantically valid, diverse, and adversarially effective robustness evaluation variants. SAGE differs by targeting robustness benchmark generation specifically and employing a rubric-based quality verifier to ensure quality.

Reinforcement learning for text generation. RLHF [Ouyang et al., 2022] and preference-based methods such as DPO [Rafailov et al., 2023] align LLMs with human preferences using holistic reward signals. GRPO [Shao et al., 2024] has proven particularly effective in verifiable domains like mathematics [Li et al., 2026b] and code agent with memory management [Bao et al., 2025, Wang et al., 2026a,b], where correctness can be automatically checked to provide clear reward signals. More recently, Gunjal et al. [2025], Li et al. [2026a,c] extended RL to non-verifiable domains by using structured rubrics as reward functions, showing that decomposing quality into explicit criteria can guide optimization where binary verification is unavailable. Inspired by this insight, SAGE applies rubric-guided RL to robustness benchmark generation: VariantQual's rubric decomposes variant quality into concrete, assessable dimensions and serves as the reward model for GRPO.

3 Method

3.1 Problem Definition

We define robustness augmentation for LLM knowledge evaluation benchmarks as the task of generating valid question variants under predefined variant types. Given an original multiple-choice question $q = (c, \mathcal{O}, y)$, where c is the context, $\mathcal{O} = \{o_k\}_{k=1}^{K}$ denotes the answer choices, and $y \in \{1, \ldots, K\}$ is the correct answer index, let $\mathcal{T}$ denote the set of predefined variant types. For each target type $t \in \mathcal{T}$, the generated variant $q_t = (c_t, \mathcal{O}_t, y_t)$ should remain a valid multiple-choice question that tests the same underlying knowledge as q, while changing its formulation or reasoning process according to type t. In our implementation, $\mathcal{T}$ contains seven variant types guided by Bloom's cognitive taxonomy [Krathwohl, 1973], as summarized in Table 1.

Existing work shows that prompting strong LLMs for variant construction remains costly and difficult to scale due to low-yield generation and unreliable verification. Motivated by these limitations, SAGE replaces repeated strong-model prompting with two fine-tuned small-model components: VariantQual, a rubric-based verifier for evaluating variant quality, and VariantGen, a generator conditioned on the original question and the target variant type. We first describe these two components and then present the complete benchmark construction pipeline. Figure 1 presents an overview of SAGE.

3.2 VariantQual

We identify two key requirements for fine-tuning a reliable verifier: well-defined evaluation criteria and accurate verification based on these criteria. We therefore derive a three-dimensional rubric for variant verification and fine-tune VariantQual to learn and apply this rubric.

Table 1: Seven variant types adopted in SAGE, mapped to Bloom's cognitive taxonomy [Krathwohl, 1973]. Each type transforms the original question into a different reasoning form while preserving the underlying knowledge. Refer to Figure 3 for examples.

Variant Type	Cognitive Level	Description
Problem Restatement	Understand	Rephrase context and correct choice; semantics unchanged
Causal Inference	Understand	Identify the most plausible cause of an event
Reverse Conversion	Apply	Infer the context from the outcome
Scenario Refinement	Apply	Modify context so a different choice becomes correct
Negative Transformation	Apply	Introduce negation; least plausible choice becomes correct
Sentence Ordering	Analyze	Determine the correct temporal order of shuffled sentences
Critical Testing	Evaluate	Remove key details; "None of the above" becomes correct

Rubric Design. By examining the publicly released annotation data from HellaSwag-Pro, we identify three key requirements for q_t to be valid. We summarize these requirements into three rubric dimensions. A variant is considered valid only if it passes all three dimensions.

- **Type Compliance (TC)**: Whether the generated variant preserves the original tested knowledge while satisfying the constraints of the target variant type.
- **Label Correctness (LC)**: Whether the annotated label correctly identifies the right answer given the generated context and answer choices.
- **Answer Uniqueness (AU)**: Whether the generated variant has exactly one correct answer, with the other choices being neither duplicate nor irrelevant to the question.

We transform the verification annotations from HellaSwag-Pro into seed data for training VariantQual:

$$\mathcal{D}_{\text{qual}} = \{(q, q_t, t, l_{\text{TC}}, l_{\text{LC}}, l_{\text{AU}}, v)\},$$

where $l_{\text{TC}}, l_{\text{LC}}, l_{\text{AU}} \in \{0, 1\}$ are dimension-level labels, and $v \in \{0, 1\}$ is the final validity label. Since a valid variant must satisfy all three dimensions, we define

$$v = l_{\text{TC}} \cdot l_{\text{LC}} \cdot l_{\text{AU}}.$$

The remaining question is how VariantQual should learn and aggregate these rubric dimensions. We compare two strategies: *explicit* and *implicit* rubric aggregation.

Explicit Rubric Aggregation (ERA). ERA fine-tunes VariantQual to evaluate each rubric dimension separately. Given a rubric instruction r_d for dimension $d \in \{\text{TC}, \text{LC}, \text{AU}\}$, VariantQual predicts the corresponding dimension-level label:

$$p_\phi(l_d \mid q, q_t, t, r_d),$$

where ϕ denotes the parameters of VariantQual. The SFT objective for ERA is to maximize the likelihood of all dimension-level labels:

$$\max_\phi \ \mathcal{J}_{\text{ERA}}(\phi) = \mathbb{E}_{\mathcal{D}_{\text{qual}}} \left[\sum_{d \in \{\text{TC},\text{LC},\text{AU}\}} \log p_\phi(l_d \mid q, q_t, t, r_d) \right]. \tag{1}$$

At inference time, VariantQual predicts $\hat{l}_{\text{TC}}$, $\hat{l}_{\text{LC}}$, and $\hat{l}_{\text{AU}}$ separately, and aggregates them as

$$\hat{v} = \hat{l}_{\text{TC}} \cdot \hat{l}_{\text{LC}} \cdot \hat{l}_{\text{AU}}.$$

This strategy provides fine-grained diagnostic signals, since each invalid variant can be traced to specific failed dimensions.

Implicit Rubric Aggregation (IRA). IRA fine-tunes VariantQual to directly predict the final validity label by evaluating all rubric dimensions jointly in a single verification pass:

$$p_\phi(v \mid q, q_t, t, \mathcal{R}),$$

where $\mathcal{R} = [r_{\text{TC}}, r_{\text{LC}}, r_{\text{AU}}]$ denotes the complete rubric instruction. The SFT objective for IRA is

$$\max_\phi \ \mathcal{J}_{\text{IRA}}(\phi) = \mathbb{E}_{\mathcal{D}_{\text{qual}}} \left[\log p_\phi(v \mid q, q_t, t, \mathcal{R}) \right]. \tag{2}$$

This strategy reduces inference cost and allows the model to make a joint decision over all validity requirements. We fine-tune VariantQual under both ERA and IRA, empirically compare them in Section 4.3, and adopt IRA as the default setting in the SAGE pipeline.

3.3 VariantGen

After obtaining VariantQual, we now train VariantGen to generate variants that satisfy the rubrics with a two-stage strategy. First, we apply SFT on human-annotated seed examples from HellaSwag-Pro to initialize VariantGen for q_t generation conditioned on q and t. Second, we further optimize VariantGen with GRPO [Shao et al., 2024], using VariantQual to provide verification rewards.

Stage 1: SFT. We construct the SFT data as

$$\mathcal{D}_{\text{gen}} = \{(q, t, q_t^*)\},$$

where q_t^* is a human-accepted variant of the original question q under variant type t. Let π_θ denote VariantGen. The SFT objective is to maximize the likelihood of the accepted variant:

$$\max_\theta \ \mathcal{J}_{\text{SFT}}(\theta) = \mathbb{E}_{(q,t,q_t^*)\sim\mathcal{D}_{\text{gen}}} \left[\log \pi_\theta(q_t^* \mid q, t)\right]. \tag{3}$$

Stage 2: GRPO Optimization. SFT teaches VariantGen to imitate accepted variants, but does not directly optimize the verifier-defined quality objective. We therefore apply GRPO [Shao et al., 2024] to further optimize VariantGen using VariantQual's validity prediction as the reward signal. For each input $x = (q, t)$, VariantGen samples a group of G candidate variants from the old policy $\pi_{\theta_{\text{old}}}$:

$$q'_1, \ldots, q'_G \sim \pi_{\theta_{\text{old}}}(\cdot \mid x).$$

VariantQual evaluates each candidate by taking the original question, the generated variant, the target variant type, and the complete rubric instruction as input:

$$\hat{v}_j = p_\phi(q, q'_j, t, \mathcal{R}), \quad \hat{v}_j \in \{0, 1\}. \tag{4}$$

We use $\hat{v}_j$ as the reward signal for updating VariantGen. GRPO computes the group-relative advantage by normalizing the rewards within the sampled group:

$$\hat{A}_j = \frac{\hat{v}_j - \text{mean}(\{\hat{v}_k\}_{k=1}^G)}{\text{std}(\{\hat{v}_k\}_{k=1}^G) + \epsilon}. \tag{5}$$

VariantGen is then optimized by maximizing the clipped GRPO objective:

$$\max_\theta \ \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x,\{q'_j\}_{j=1}^G} \Bigg[\frac{1}{G} \sum_{j=1}^{G} \min \left(\rho_j(\theta)\hat{A}_j, \text{clip}(\rho_j(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})\hat{A}_j \right) - \beta D_{\text{KL}} \left(\pi_\theta(\cdot \mid x) \| \pi_{\text{ref}}(\cdot \mid x)\right) \Bigg], \tag{6}$$

where

$$\rho_j(\theta) = \frac{\pi_\theta(q'_j \mid x)}{\pi_{\theta_{\text{old}}}(q'_j \mid x)}.$$

Here, $\pi_{\theta_{\text{old}}}$ is the old policy used to sample candidate variants, π_{ref} is the SFT model used as the reference policy, ϵ is a small constant for numerical stability, ϵ_{clip} is the clipping threshold, and β controls the KL penalty strength.

3.3.1 Benchmark Construction

Finally, we construct the full benchmark through generate-then-filter. For each source question, VariantGen generates candidates for all variant types, and VariantQual retains only those passing quality verification. We repeat this process until for the entire benchmark, each variant type reaches N accepted examples. The same pipeline is applied to HellaSwag and, without additional fine-tuning, to MMLU for cross-benchmark evaluation.

4 Experiments

In this section, we present a comprehensive evaluation of SAGE. Our experiments are designed to answer four key research questions: **- RQ1**: What is the effectiveness of the generator and quality checker framework in SAGE? **- RQ2**: How can the detection accuracy for VariantQual be improved? **- RQ3**: How effective is the SAGE-generated benchmark in evaluating model robustness? **- RQ4**: What is the performance of SAGE across different tasks?

4.1 Experimental Setup

Training configuration. VariantGen and VariantQual are both initialized from Qwen3-0.6B. Table 9 summarizes the training and test set sizes for each module in SAGE. All SFT training uses the LLaMA-Factory [Zheng et al., 2024] framework with LoRA adapters (rank 8, α = 16). GRPO training uses ms-swift [Zhao et al., 2025] with num generations = 8 and max completion length = 1024; rollout sampling uses temperature = 0.9 and top-p = 0.9. Full hyperparameter details are provided in Appendix A (Tables 10 and 11).

Evaluation metrics. For VariantQual we report classification accuracy (ACC), AUC, Recall, and F1 on the held-out test set. For VariantGen we report pass rate: the fraction of generated variants accepted by VariantQual. For the downstream robustness benchmark we report four metrics following HellaSwag-Pro: original accuracy (OA), average robust accuracy (ARA), robust loss accuracy (RLA), and consistent robust accuracy (CRA).

4.2 Overall Performance (RQ1)

Table 2: Overall performance of SAGE on VariantGen (%). Best results in **bold**.

Training Stage	Base	Instruct
SFT Only	70.58	90.53
GRPO Only	65.77	77.54
SFT + GRPO	69.52	**91.80**

Table 3: Per-variant-type quality comparison between prompted Qwen2.5-Max and SAGE.

Variant Type	Qwen2.5-Max	SAGE
Causal Inference	97.71	**100.00**
Critical Testing	96.85	**100.00**
Negative Transformation	85.44	**98.75**
Problem Restatement	98.61	**98.75**
Reverse Conversion	78.24	**88.75**
Scenario Refinement	45.56	**56.25**
Sentence Ordering	98.00	**100.00**

Main results. To assess the effectiveness of SAGE, we conduct experiments as shown in Table 2. VariantGen-SFT (Instruct) achieves 90.53%, establishing a strong baseline for subsequent RL optimization. Adding GRPO further improves the pass rate to 91.80%, yielding a gain of 1.27% over SFT alone. In addition, Instruct models consistently outperform their Base counterparts, suggesting that instruction-following priors transfer effectively to structured variant generation.

Per-variant comparison. In Table 3, we compare SAGE with a prompt-based Qwen2.5-Max baseline that following HellaSwag-Pro generation protocol. SAGE exceeds Qwen2.5-Max on all seven variant types. *Scenario Refinement* remains the most challenging type for both systems, but the margin indicates that SAGE is notably stronger at handling complex semantic transformations.

4.3 VariantQual Training (RQ2)

Table 4: Comparison of VariantQual models with different sizes.

Base Model	ACC	AUC	Recall	F1
Qwen3-0.6B	**90.21**	**79.10**	**94.50**	**94.30**
Qwen3-1.7B	89.23	77.70	93.70	93.70
Qwen3-4B	89.32	76.70	94.20	93.80

Table 5: Comparison of implicit and explicit rubric strategies.

Rubric Strategy	ACC	AUC	Recall	F1
IRA	**90.21**	**79.10**	94.50	**94.30**
ERA	87.90	64.60	**97.10**	93.20

Table 6: VariantQual performance before and after RL training.

VariantQual	ACC	AUC	Recall	F1
SFT	**90.21**	**79.10**	94.50	94.30
SFT + GRPO	90.12	77.70	**94.90**	94.30

Model-size comparison. Table 4 reports VariantQual performance after SFT across different model sizes. We observe that Qwen3-0.6B performs on par with, or slightly better than, the larger models. This result suggests that the VariantQual task is sufficiently constrained for a 0.6B model to capture the relevant decision boundary given expert-annotated training data, making additional model scale unnecessary. Since the smallest model already delivers competitive performance, we adopt Qwen3-0.6B as the default VariantQual backbone for the rest of our experiments. A similar trend also appears in VariantGen, where Qwen3-0.6B (90.53%) remains competitive with Qwen3-1.7B (88.06%) and Qwen3-4B (90.02%).

Rubric design comparison. We compare the implicit rubric (IRA) against the explicit rubric (ERA) described in Section 3 in Table 5. IRA achieves higher ACC (90.21% vs. 87.90%) and

Table 7: Robustness evaluation of HellaSwag on the SAGE-generated benchmark. All reported values include 95% confidence intervals. Best results are **bold**.

Model	OA (%) ↑	ARA (%) ↑	RLA (%) ↓	CRA (%) ↑
Gemma-3				
Gemma-3-4B	60.79±1.95	33.39±0.71	27.40±2.08	23.15±0.64
Gemma-3-12B	64.50±1.91	33.73±0.71	30.77±2.04	24.33±0.65
Gemma-3-27B	66.17±1.89	34.41±0.72	31.76±2.02	25.33±0.66
Llama				
Llama-3.2-1B	53.67±1.99	31.83±0.70	21.83±2.12	20.29±0.61
Llama-3.2-3B	59.54±1.96	32.84±0.71	26.70±2.09	22.57±0.63
Llama-3.1-8B	64.71±1.91	34.11±0.72	30.60±2.04	24.79±0.65
Llama-3.1-70B	71.54±1.80	35.68±0.72	35.86±1.95	27.93±0.68
Qwen3.5				
Qwen3.5-2B	54.12±1.99	32.50±0.71	21.62±2.12	20.57±0.61
Qwen3.5-4B	60.50±1.95	34.73±0.72	25.77±2.08	23.83±0.64
Qwen3.5-9B	66.21±1.89	36.73±0.73	29.48±2.03	26.86±0.67
Qwen3.5-35B-A3B	71.79±1.80	37.60±0.73	34.20±1.94	29.26±0.69
GPT				
GPT-4o	**74.17**±1.75	**65.68**±0.66	**8.49**±2.27	**50.67**±0.72

AUC (79.10% vs. 64.60%). One possible reason is that IRA better adapts to the data distribution during training, whereas ERA may over-rely on specific dimensions, leading to degraded overall performance after the logical AND aggregation.

RL stability analysis. To further improve VariantQual's performance and robustness, we conduct additional GRPO training on top of its SFT baseline. We evaluate VariantQual's performance before and after this RL training. As shown in Table 6, GRPO yields marginal and inconsistent changes relative to the SFT baseline (ACC: 90.21% → 90.12%; Recall: 94.50% → 94.90%), with no consistent improvement across metrics. We hypothesize that GRPO optimization might induce a degree of reward hacking by exploiting VariantQual's vulnerabilities, which prevents consistent performance gains. Therefore, we adopt the SFT-trained VariantQual in the final system.

4.4 Robustness Benchmark Evaluation and Analysis (RQ3)

To evaluate the effectiveness of the SAGE-generated benchmark in assessing model robustness, we deploy the full SAGE pipeline on HellaSwag to generate a large-scale robustness benchmark with 16800 samples ($N = 2400$) and evaluate 12 LLMs spanning four model families.

Main results. Table 7 reports OA, ARA, RLA, and CRA for tested LLMs. We find that: (1) **Universal performance gap**: Consistent with HellaSwag-Pro, all models exhibit a substantial OA-to-ARA drop, confirming that robustness deficiencies are pervasive across families and scales. (2) **Robustness lags capability**: For open-source LLMs, OA spans 53.67%–71.79%, yet ARA is compressed into 31.83%–37.60%, showing that scaling yields diminishing robustness returns. (3) **RLA grows with model size**: Within each family, scaling consistently increases RLA (Llama 1B→70B: 21.83%→35.86%; Qwen3.5 2B→35B: 21.62%→34.20%)—indicating that capability gains do not translate proportionally into robustness gains. (4) **CRA is uniformly low**: Even the strongest open-source model achieves only 29.26% CRA, highlighting the severity of the robustness gap.

Per-type RLA Contribution. To further analyze the impact of different variant types, we calculate the normalized contribution of each variant type to the RLA. As shown in Figure 2, the distribution is remarkably uniform across models within each family, suggesting that variant-type difficulty is a property of the perturbation strategy. *Negative Transformation* consistently contributes the largest RLA share across all model families. This may be attributed to the presence of statistical shortcuts in pre-training corpora and shortcut learning from corpus-level statistical patterns for this variant type.

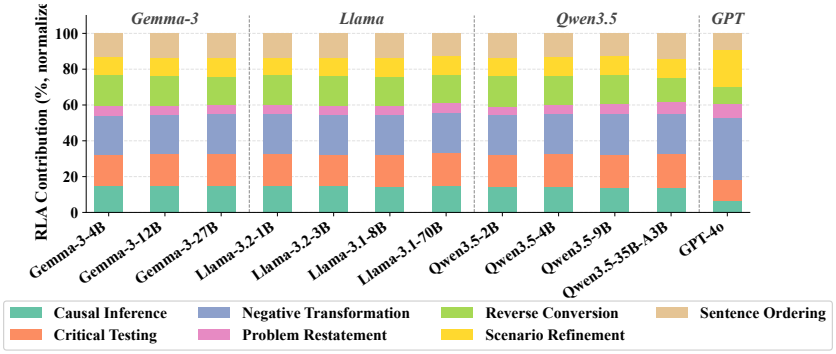

Figure 2: Normalized per-variant-type contribution to RLA across LLMs.

Table 8: Robustness evaluation of MMLU on the SAGE-generated benchmark. All reported values include 95% confidence intervals. Best results are **bold**.

Model	OA (%) ↑	ARA (%)↑	RLA (%)↓	CRA (%)↑
Gemma-3				
Gemma-3-4B	65.55±6.04	47.78±2.27	17.77±6.45	34.57±2.22
Gemma-3-12B	74.79±5.52	49.58±2.28	25.21±5.97	40.16±2.27
Gemma-3-27B	75.21±5.49	49.22±2.29	25.99±5.94	39.92±2.26
Llama				
Llama-3.2-1B	52.10±6.35	45.50±2.27	6.60±6.74	27.91±2.10
Llama-3.2-3B	62.18±6.16	47.12±2.29	15.07±6.57	32.53±2.20
Llama-3.1-8B	66.81±5.98	47.72±2.28	19.09±6.40	34.93±2.22
Llama-3.1-70B	70.17±5.81	48.80±2.28	21.37±6.24	37.21±2.24
Qwen3.5				
Qwen3.5-2B	46.22±6.33	45.86±2.28	**0.36**±6.73	24.37±2.02
Qwen3.5-4B	53.36±6.34	46.52±2.29	6.84±6.74	27.79±2.11
Qwen3.5-9B	56.72±6.29	46.88±2.28	9.84±6.70	29.95±2.15
Qwen3.5-35B-A3B	58.40±6.26	47.12±2.28	11.28±6.66	30.49±2.16
GPT				
GPT-4o	**94.12**±2.99	**67.33**±1.89	26.79±3.03	**64.76**±1.94

In contrast, *Problem Restatement* contributes the smallest RLA share, likely because it involves more surface-level transformations that are easier for models to handle.

Furthermore, to verify that the SAGE-generated benchmark maintains consistency in model ranking with the human-annotated HellaSwag-Pro reference set, we calculate the Spearman ρ correlation coefficients between these. This analysis covers OA, ARA, RLA, and CRA across the models shared by both datasets. Among these, ARA achieves $\rho = 0.857$ ($p = 0.0137$) and CRA achieves $\rho = 0.964$ ($p = 0.0005$); OA and RLA both achieve $\rho = 1.000$. These results confirm that SAGE-generated benchmarks faithfully preserve the relative ordering of model robustness established by human annotation, validating SAGE as a reliable and scalable alternative to manual benchmark construction. Detailed results are provided in Table 13.

Meanwhile, we randomly sample 350 SAGE-generated variants and manually verify whether they adhere to the intended constraints, achieving 86.6% accuracy. See Appendix D for details.

4.5 Cross-task Evaluation (RQ4)

MMLU robustness results. As shown in Table 8, the robustness degradation pattern observed on HellaSwag transfers consistently to MMLU. ARA is compressed into 45.50%–67.33% despite OA spanning 52.10%–94.12%, and scaling increases RLA within each family. Compared to HellaSwag, MMLU exhibits smaller RLA values, likely because factual knowledge questions are less susceptible

ORIGINAL *MMLU*

Context: Jacques, Hedwig, and Reyes want to form a business entity through which they can raise capital by selling equity shares to the public. Which of the following business structures should they adopt?
Choices: (A) LLC (B) PC (C) Corporation (D) LLP

---

[1] Problem Restatement *(label: C)*

Context: Jacques, Hedwig, and Reyes aim to establish a business unit where they can attract investors by offering equity shares. What type of business structure would best suit their needs?
Choices: (A) LLC (B) PC (C) Corporation (D) LLP

---

[2] Negative Transformation *(label: A)*

Context: Jacques, Hedwig, and Reyes want to form a business entity through which they can raise capital by selling equity shares to the public. Which of the following business structures should they **not** adopt?
Choices: (A) LLC (B) Corporation (C) Joint-Stock Company (D) Public Limited Company

---

[3] Scenario Refinement *(label: D)*

Context: Jacques, Hedwig, and Reyes want to form a business entity through which they can raise capital by selling equity shares to the public. However, after consulting with their advisor, they decide to prioritize a partnership-based structure that limits each partner's personal liability for the other partners' actions, rather than pursuing public equity financing. Which of the following business structures should they adopt?
Choices: (A) LLC (B) PC (C) Corporation (D) LLP

---

[4] Critical Testing *(label: E)*

Context: Jacques, Hedwig, and Reyes want to form a business entity through which they can raise capital by selling equity shares to the public. However, regulations in their jurisdiction prohibit any newly formed entity from issuing equity shares to the public within the first five years of registration. Which of the following business structures should they adopt to meet their immediate goal?
Choices: (A) LLC (B) PC (C) Corporation (D) LLP (E) None of the above four options are suitable.

---

[5] Causal Inference *(label: A)*

Context: Jacques, Hedwig, and Reyes want to form a business entity through which they can raise capital by selling equity shares to the public. They should adopt a corporation. Which could be the possible reason for this action?
Choices:

(A) A corporation can issue shares of stock to the general public, enabling large-scale capital raising.
(B) Forming a corporation eliminates all personal tax obligations for the founders.
(C) A corporation requires fewer regulatory filings and compliance procedures than other structures.
(D) Corporate structures automatically prevent any form of personal liability, including for fraudulent acts.

---

[6] Reverse Conversion *(label: A)*

Context: The corporation should be formed by Jacques, Hedwig, and Reyes. Which could be the possible context for this action?
Choices:

(A) Jacques, Hedwig, and Reyes want to form a business entity through which they can raise capital by selling equity shares to the public.
(B) A family business is planning to establish a limited liability company to protect its members' interests.
(C) A family business is looking to form a limited liability company to manage its finances more effectively.
(D) A family business is planning to form a limited liability company to ensure clear ownership and limited liability.

---

[7] Sentence Ordering *(label: A)*

Context: (1) Jacques, Hedwig, and Reyes want to form a business entity through which they can raise capital by selling equity shares to the public. (2) They decide to establish a corporation to maximize their investment potential. (3) Each shareholder elects a board of directors and has voting rights. (4) The corporation plans to launch its own business operations. Which is the correct order?
Choices: (A) 1–2–3–4 (B) 2–1–3–4 (C) 1–3–2–4 (D) 4–1–2–3

Figure 3: Case study of SAGE-generated variants from MMLU. The original question and 7 generated variants are shown, with the correct answer choice underlined.

to structural perturbations. Nevertheless, CRA remains moderate, confirming that robustness gaps persist across task domains.

Case study. Figure 3 presents a case study of SAGE applied to an MMLU question. SAGE generates seven variants from one question, covering all perturbation types, each adhering to its constraints. Notably, *Scenario Refinement* shifts the correct answer from Corporation to LLP by introducing a flexibility constraint, and *Critical Testing* invalidates all original options by adding a liquidity constraint, demonstrating SAGE's ability to generate context-sensitive variants beyond simple paraphrase.

5 Conclusion

We proposed SAGE, a scalable framework for automated robustness augmentation of knowledge evaluation benchmarks. SAGE replaces costly strong-model prompting with two fine-tuned small models trained via SFT and verifier-guided GRPO. Applied to HellaSwag, SAGE constructs a 16,800-sample robustness benchmark whose model rankings correlate near-perfectly with human-annotated references, while generalizing to MMLU without additional fine-tuning. We hope SAGE facilitates the shift from static benchmarks toward scalable, continuously updated robustness diagnostics.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Introducing the next generation of claude. `https://www.anthropic.com/news/claude-3-family`, 2024.

Nishant Balepur, Shramay Palta, and Rachel Rudinger. It's not easy being wrong: Large language models struggle with process of elimination reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10143–10166, 2024.

Keqin Bao, Nuo Chen, Xiaoyuan Li, Binyuan Hui, Bowen Yu, Fuli Feng, Xiangnan He, and Dayiheng Liu. Teaching LLM to reason: Reinforcement learning from algorithmic problems without code. *CoRR*, abs/2507.07498, 2025. doi: 10.48550/ARXIV.2507.07498. URL `https://doi.org/10.48550/arXiv.2507.07498`.

Ruben Branco, António Branco, Joao Rodrigues, and Joao Silva. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, 2021.

Shuochen Chang, Tong Bai, Xiaofeng Zhang, Qianli Ma, Qingyang Liu, Zhaohe Liao, Yibo Miao, and Li Niu. Unlocking the black box of latent reasoning: An interpretability-guided approach to intervention. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pages 34019–34032. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1568. URL `https://doi.org/10.18653/v1/2026.acl-long.1568`.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=d7KBjmI3GmQ`.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2021–2031, 2017.

Jacob K Johnson and Ana Marasović. How much consistency is your accuracy worth? In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 250–260, 2023.

David R Krathwohl. Taxonomy of educational objectives. *Affective domain*, 1973.

Xiaoyuan Li, Moxin Li, Rui Men, Yichang Zhang, Keqin Bao, Wenjie Wang, Fuli Feng, Dayiheng Liu, and Junyang Lin. Hellaswag-pro: A large-scale bilingual benchmark for evaluating the robustness of llms in commonsense reasoning. pages 9038–9072, 2025a.

Xiaoyuan Li, Moxin Li, Wenjie Wang, Rui Men, Yichang Zhang, Fuli Feng, Dayiheng Liu, and Junyang Lin. Mathopeval: A fine-grained evaluation benchmark for visual operations of mllms in mathematical reasoning. *CoRR*, abs/2507.18140, 2025b. doi: 10.48550/ARXIV.2507.18140. URL https://doi.org/10.48550/arXiv.2507.18140.

Xiaoyuan Li, Keqin Bao, Moxin Li, Yubo Ma, Yichang Zhang, Wenjie Wang, Fuli Feng, and Dayiheng Liu. ARES: automated rubric synthesis for scalable LLM reinforcement learning. *CoRR*, abs/2605.23454, 2026a. doi: 10.48550/ARXIV.2605.23454. URL https://doi.org/10.48550/arXiv.2605.23454.

Xiaoyuan Li, Yubo Ma, Chengpeng Li, Fengbin Zhu, Yiyao Yu, Keqin Bao, Wenjie Wang, Fuli Feng, and Dayiheng Liu. Unified data selection for LLM reasoning. *CoRR*, abs/2605.22389, 2026b. doi: 10.48550/ARXIV.2605.22389. URL https://doi.org/10.48550/arXiv.2605.22389.

Xiaoyuan Li, Yubo Ma, Kexin Yang, Moxin Li, Keqin Bao, Wenie Wang, Fuli Feng, and Dayiheng Liu. On predicting the post-training potential of pre-trained llms. *CoRR*, abs/2605.11978, 2026c. doi: 10.48550/ARXIV.2605.11978. URL https://doi.org/10.48550/arXiv.2605.11978.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shane Storks and Joyce Chai. Beyond the tip of the iceberg: Assessing coherence of text classifiers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3169–3177, 2021.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, 2019.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2153–2162, 2019.

Junxi Wang, Te Sun, Jiayi Zhu, Junxian Li, Haowen Xu, Zichen Wen, Xuming Hu, Zhiyu Li, and Linfeng Zhang. Streammeco: Long-term agent memory compression for efficient streaming video understanding. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pages 13234–13251. Association for Computational Linguistics, 2026a. doi: 10.18653/V1/2026.FINDINGS-ACL.647. URL https://doi.org/10.18653/v1/2026.findings-acl.647.

Junxi Wang, Te Sun, Jiayi Zhu, Chen Zhang, Siyuan Li, Xuyang Liu, Zichen Wen, Xiaobing Tu, Jinkui Ren, Xiantao Zhang, Ziqi Yuan, and Linfeng Zhang. Memforest: Efficient agent memory management via eventtree partitioning and progressive merging. *CoRR*, abs/2609.08273, 2026b. doi: 10.48550/ARXIV.2609.08273. URL https://doi.org/10.48550/arXiv.2609.08273.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. pages 13484–13508, 2023.

Yutong Wang, Yunke Wang, Tianfan Xue, Yu Qiao, Yaohui Wang, Xinyuan Chen, and Chang Xu. PARE: pruning and adaptive routing for efficient video generation. *CoRR*, abs/2605.27336, 2026c. doi: 10.48550/ARXIV.2605.27336. URL `https://doi.org/10.48550/arXiv.2605.27336`.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1819–1862, 2024.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Siyu Yuan, Jiangjie Chen, Ziquan Fu, Xuyang Ge, Soham Shah, Charles Jankowski, Yanghua Xiao, and Deqing Yang. Distilling script knowledge from large language models for constrained language planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4303–4325, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800, 2019.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29733–29735, 2025.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pages 400–410, 2024.

A Hyperparameter Details

Table 9 summarizes the training and test set sizes for each module in SAGE. Tables 10 and 11 summarize the hyperparameter configurations used for SFT and GRPO training of VariantGen and VariantQual.

Table 9: Training and test set sizes for each module in SAGE. Each dataset covers all 7 variant types, which are balanced within each set. The GRPO training set is not evaluated on, so no test set is needed.

Module	**Train**	**Test**
VariantQual	10096	1124
$\text{VariantGen}_{\text{SFT}}$	5039	561
$\text{VariantGen}_{\text{GRPO}}$	7000	-

B Compute and Cost Details

The total monetary cost of the reported experiments is approximately 284 USD. For comparison, the original human annotation process in HellaSwag-Pro cost approximately 7,064 USD, meaning SAGE achieves a $\sim$25$\times$ cost reduction while producing benchmarks of comparable quality. The main training runs use 15 A40 GPU-hours and 40 A100 GPU-hours, as summarized in Table 12.

Table 10: SFT (LoRA) hyperparameters. Both VariantGen and VariantQual share the same configuration.

Hyperparameter	Value	Description
Framework	LLaMA-Factory	SFT training framework
Base model	Qwen3-0.6B	Backbone for both modules
LoRA rank	8	Intrinsic dimension of LoRA
LoRA alpha	16	Scaling factor ($2\times$ rank)
LoRA dropout	0.0	Dropout rate on LoRA layers
Learning rate	1e-4	AdamW learning rate
Temperature	0.95	Sampling temperature during training
Max new tokens	1024	Maximum generation length

Table 11: GRPO hyperparameters.

Hyperparameter	Value	Description
Framework	ms-swift	RLHF training framework
Base model	Qwen3-0.6B	Backbone initialized from SFT checkpoint
Num generations	8	Rollout samples per prompt
Max completion length	1024	Maximum generation length per rollout
Rollout temperature	0.9	Sampling temperature during rollout
Top-p	0.9	Nucleus sampling threshold
Learning rate	1e-6	AdamW learning rate
β (KL penalty)	0.04	KL divergence coefficient
Clip range	0.2	PPO probability ratio clip bound
Num train epochs	2	Total training epochs
Per-device train batch size	2	Batch size per GPU
Gradient accumulation steps	8	Effective batch size $= 2 \times 8 = 16$
Precision	bfloat16	Mixed-precision training format
Gradient checkpointing	True	Enabled to reduce GPU memory usage

C Ranking Consistency with HellaSwag-Pro

C.1 Model Comparison

We compare model performance on HellaSwag-Pro reference set versus SAGE-generated variants across all four metrics (OA, ARA, RLA, CRA), and report Spearman ρ and Kendall τ rank correlation coefficients. Full results are shown in Table 13.

C.2 Extended Model Comparison

To provide a more comprehensive evaluation of ranking consistency, we conducted an additional experiment comparing HellaSwag-Pro and SAGE on 12 models spanning multiple model families and scales, which are reported in HellaSwag-Pro. Table 14 reports detailed performance metrics with confidence intervals for HellaSwag-Pro and corresponding SAGE results. The Spearman correlations for ARA, RLA, and CRA are 0.951, 0.944, and 0.993 respectively (all $p < 0.001$), confirming that SAGE preserves model-level robustness rankings similar to the human-curated reference benchmark across a diverse set of contemporary language models.

D Human Evaluation of Variant Quality

To verify that SAGE-generated variants adhere to their intended perturbation constraints, we randomly sample 350 variants (50 per variant type) and conduct manual evaluation. Each variant is assessed on three criteria: (1) perturbation compliance, (2) label correctness, and (3) answer uniqueness. As shown in Figure 4, the vast majority of generated variants successfully satisfy the three criteria above. This confirms that SAGE effectively produces valid, diverse variants suitable for robust evaluation.

Although we observe moderately lower strict compliance rates for the *Scenario Refinement* and *Critical Thinking* perturbation types, this is primarily due to the multi-task nature of our training

Table 12: Aggregate cost and GPU-hour usage for the reported experiments.

Item	Value
Total monetary cost	∼284 USD
A40 compute	15 GPU-hours
A100 compute	40 GPU-hours

Table 13: Comparison of model performance on HellaSwag-Pro reference set vs. SAGE-generated variants, with ranking consistency between the two evaluations. Statistical significance of ranking correlations is assessed via Spearman ρ and Kendall τ with two-sided p-values.

	HellaSwag-Pro Reference				**SAGE Variants**			
Model	OA	ARA	RLA	CRA	OA	ARA	RLA	CRA
Gemma-2								
Gemma-2-2B	59.62	39.13	20.50	24.88	58.29	33.43	24.86	22.43
Gemma-2-9B	64.88	39.80	25.08	26.91	62.38	33.14	29.24	23.45
Gemma-2-27B	71.88	40.91	30.97	30.25	69.29	34.78	34.51	26.60
Llama-3								
Llama-3-8B	66.25	40.21	26.04	27.34	65.38	34.26	31.12	25.13
Llama-3-70B	72.50	41.27	31.23	30.63	71.33	35.46	35.87	27.72
Mistral								
Mistral-7B-v0.1	67.50	41.52	25.98	28.93	66.04	35.05	30.99	25.93
Mixtral-8x7B-v0.1	69.75	41.21	28.54	29.39	69.08	35.14	33.94	26.89
Ranking Consistency (Reference vs. SAGE)								
Spearman ρ	1.000	0.857	1.000	0.964				
p-value	0.000	0.014	0.000	0.001				
Kendall τ	1.000	0.714	1.000	0.905				
p-value	0.000	0.030	0.000	0.003				

mixture where constraints can occasionally blend. However, even when a variant does not perfectly mirror the strict definition of its assigned perturbation type (e.g., failing to flawlessly "refine" a scenario), it typically still introduces a meaningful and challenging structural modification to the original question. Consequently, these variants remain highly effective for testing reasoning robustness.

E Extended Human Evaluation

To further validate SAGE's quality and address potential circular validation concerns, we conducted an extended human evaluation on 700 filtered SAGE variants (100 per category). Human annotators assessed each variant on three criteria: perturbation compliance, label correctness, and answer uniqueness. Results show that 614 out of 700 variants (87.7%) met all three strict validity criteria.

Table 15 compares human-verified validity rates between HellaSwag-Pro and SAGE across all seven variant types. SAGE achieves a higher overall validity rate (87.71% vs. 85.77%), demonstrating that small-model benchmark augmentation can produce high-quality variants comparable to human annotation.

F Robustness on Human-Verified Subset

To assess the impact of potentially invalid examples on model ranking consistency, we evaluate model performance on both the full SAGE dataset and a human-verified subset. After removing invalid examples, the Spearman correlations for RLA and CRA remain high at 0.832 and 0.989 respectively, indicating that model rankings are preserved even when restricting to manually verified variants. Table 16 reports detailed results.

Table 14: Comparison of model performance on HellaSwag-Pro reference set vs. SAGE-generated variants on an extended set of 12 models. RLA denote relative accuracy metrics computed as performance drops relative to the original accuracy. Statistical significance of ranking correlations is assessed via Spearman ρ and Kendall τ with two-sided p-values.

	HellaSwag-Pro Reference				SAGE Variants			
Model	OA	ARA	RLA	CRA	OA	ARA	RLA	CRA
Gemma-3								
Gemma-3-4B	45.50	32.66	28.22	17.05	60.79	33.39	45.07	23.15
Gemma-3-12B	47.75	32.57	31.79	17.91	64.50	33.73	47.71	24.33
Gemma-3-27B	48.62	33.25	31.62	18.62	66.17	34.41	48.00	25.33
Llama-3.1								
Llama-3.1-8B	48.50	33.20	31.55	18.21	64.71	34.11	47.29	24.79
Llama-3.1-70B	53.50	34.66	35.21	20.86	71.54	35.68	50.13	27.93
Llama-3.2								
Llama-3.2-1B	41.50	31.27	24.66	14.95	53.67	31.83	40.69	20.29
Llama-3.2-3B	45.62	32.39	29.00	16.98	59.54	32.84	44.84	22.57
Qwen3.5								
Qwen3.5-2B	40.88	32.95	19.40	15.23	54.12	32.50	39.95	20.57
Qwen3.5-4B	47.25	34.59	26.80	17.98	60.50	34.73	42.60	23.83
Qwen3.5-9B	50.75	36.20	28.68	20.16	66.21	36.73	44.52	26.86
Qwen3.5-35B-A3B	54.75	37.05	32.32	21.88	71.79	37.60	47.63	29.26
GPT-4o								
GPT-4o	88.63	70.17	20.83	63.06	74.17	65.68	11.44	50.67
Ranking Consistency (Reference vs. SAGE)								
Spearman ρ	0.972	0.951	0.944	0.993				
p-value	$<$0.001	$<$0.001	$<$0.001	$<$0.001				
Kendall τ	0.909	0.879	0.818	0.970				
p-value	$<$0.001	$<$0.001	$<$0.001	$<$0.001				

Table 15: Human-verified validity comparison between HellaSwag-Pro and SAGE.

Variant Type	**HellaSwag-Pro (%)**	**SAGE (%)**
Causal Inference	97.71	100.00
Critical Testing	96.85	75.00
Negative Transformation	85.44	97.00
Problem Restatement	98.61	98.00
Reverse Conversion	78.24	99.00
Scenario Refinement	45.56	50.00
Sentence Ordering	98.00	95.00
Overall	**85.77**	**87.71**

G GRPO Quality Improvement Analysis

We provide additional evidence that GRPO improves VariantGen quality beyond mere alignment with the verifier. After applying GRPO, the VariantQual acceptance rate on an independent human-annotated testset increased from 90.53% to 91.80%.

To verify that these gains reflect genuine quality improvements rather than reward hacking, we conducted a targeted human evaluation on 28 disagreement cases where SFT+GRPO output was accepted by VariantQual but SFT output was rejected. Manual inspection found that 21 out of 28 cases (75.0%, Wilson 95% CI: 56.6%–87.3%) represent valid variants, indicating that a substantial proportion of newly accepted cases correspond to real improvements in variant quality.

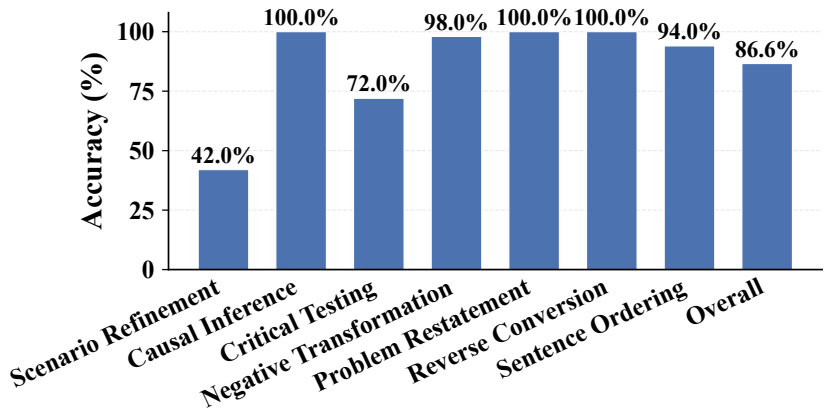

Figure 4: Human verification accuracy of SAGE-generated variants.

Table 16: Model robustness on human-verified subset versus full SAGE dataset.

Model	SUB-RLA	SUB-CRA	SAGE RLA	SAGE CRA
Gemma-3-4B	24.34	18.52	27.40	23.15
Gemma-3-12B	26.98	20.63	30.77	24.33
Gemma-3-27B	30.16	21.16	31.76	25.33
Llama-3.2-1B	24.34	17.46	21.83	20.29
Llama-3.2-3B	26.46	17.99	26.70	22.57
Llama-3.1-8B	26.98	20.63	30.60	24.79
Llama-3.1-70B	34.39	22.75	35.86	27.93
Qwen3.5-2B	24.87	17.46	21.62	20.57
Qwen3.5-4B	26.98	20.63	25.77	23.83
Qwen3.5-9B	28.57	22.22	29.48	26.86
Qwen3.5-35B-A3B	36.51	25.40	34.20	29.26
Spearman ρ	0.832	0.989		
Spearman p-value	0.0015	$<$0.001		

H MMLU Transfer Evaluation

To assess SAGE's transfer capability beyond HellaSwag, we apply the same pipeline to MMLU without benchmark-specific fine-tuning. The pipeline successfully generates all seven variant types on MMLU. Human evaluation on 70 MMLU-derived variants shows that 72.86% meet validity criteria, demonstrating effective cross-benchmark transfer. This result confirms that the learned perturbation patterns are not tied to HellaSwag-specific content and can be applied to diverse multiple-choice knowledge evaluation tasks.

I Limitations

Scope of supported benchmarks and variants. SAGE is evaluated on multiple-choice knowledge evaluation benchmarks, with HellaSwag as the primary construction setting and MMLU as a cross-task case study. The current formulation assumes that each source question has a well-defined correct answer and a fixed set of answer choices. Consequently, the conclusions may not directly transfer to free-form generation, multi-answer questions, multimodal evaluation, or tasks whose validity depends on long-form explanation rather than selecting one option. In addition, SAGE generates variants under seven predefined perturbation types. These types cover a range of reasoning forms, but they do not exhaust all possible robustness failures; extending SAGE to new domains may require designing new perturbation types and collecting a small amount of domain-specific verification data.

Dependence on verifier quality. SAGE relies on VariantQual both for filtering generated candidates and for providing rewards during GRPO optimization. Errors or biases in VariantQual can therefore

propagate into the final benchmark, and a generator optimized against this verifier may exploit verifier-specific blind spots. Although our human evaluation shows high overall quality, it also reveals that some perturbation types, especially Scenario Refinement and Critical Testing, have lower strict compliance rates than simpler transformations. This indicates that automatic filtering does not fully replace human auditing for the most semantically complex variants.

Interpretation of robustness scores. The SAGE-generated benchmark is intended as a diagnostic tool for measuring consistency under controlled question variants, not as a definitive measure of whether a model possesses the underlying knowledge. Generated variants can change question difficulty in ways that are not perfectly captured by the original-variant pairing, and some variants may introduce artifacts that affect model behavior independently of the intended perturbation. Therefore, robustness metrics such as ARA, RLA, and CRA should be interpreted together with per-type analyses and manual quality checks.

Data and deployment considerations. Because SAGE augments existing benchmarks, it inherits their coverage limitations, annotation artifacts, and potential social or cultural biases. When applying SAGE to sensitive domains such as medicine, law, finance, or safety evaluation, additional expert review is needed to verify label correctness, fairness, and domain appropriateness. Moreover, while SAGE reduces reliance on repeated prompting of strong proprietary models, large-scale benchmark construction still requires model fine-tuning, rollout generation, and verification compute. These costs should be considered when scaling the framework to substantially larger or continuously updated benchmark suites.

J Broader Impact

Positive impact. SAGE democratizes the construction of robust evaluation benchmarks by replacing expensive, large-scale human annotation with a lightweight, reproducible pipeline built on small open-source models. This substantially lowers the financial and logistical barriers to creating high-quality benchmarks, enabling resource-limited research groups and underrepresented communities to participate in LLM evaluation research. By systematically probing models under controlled perturbations, SAGE encourages the development of more robust and reliable language models, which benefits downstream applications in education, healthcare, legal reasoning, and other high-stakes domains where consistent model behavior is critical.

Potential risks and mitigation. Because SAGE can automatically generate large numbers of benchmark variants, there is a risk that the generated variants could be used to overfit models to specific evaluation patterns, thereby undermining the purpose of benchmarking. We mitigate this by clearly stating that generated variants are intended solely for evaluation, not for training. Additionally, since SAGE inherits content from existing benchmarks, it may propagate biases present in the source data. Practitioners applying SAGE to sensitive domains should conduct domain-specific expert review before deployment. We do not anticipate direct negative societal consequences from the benchmark augmentation methodology itself, as it does not generate harmful content, collect personal data, or enable surveillance capabilities.

K Asset Documentation

K.1 Existing Assets

Table 17 summarizes the existing datasets, models, and software frameworks used in this work. We use these assets for research evaluation and model fine-tuning, cite the corresponding papers or project pages in the main text, and do not redistribute third-party model weights or source datasets beyond their original terms. For assets whose license or terms are managed by an external provider, users should consult the original resource page before reuse.

K.2 New Assets

This paper introduces two types of new research assets. First, SAGE generates robustness-augmented multiple-choice benchmark variants from existing source questions. Each generated example contains

Table 17: Existing assets used in SAGE.

Asset	Use in this work	License / terms	Source
HellaSwag	Source benchmark for large-scale robustness augmentation and evaluation	MIT License	https://github.com/rowanz/hellaswag
HellaSwag-Pro	Human-annotated seed data and reference benchmark for variant generation and verification	CC BY 4.0	https://arxiv.org/abs/2502.11393
MMLU	Cross-task case study for transfer evaluation	MIT License	https://huggingface.co/datasets/cais/mmlu
Qwen3-0.6B	Backbone for VariantGen and VariantQual	Apache-2.0	https://huggingface.co/Qwen/Qwen3-0.6B
Qwen2.5-Max	Prompt-based generation baseline via API access	Provider terms	https://www.alibabacloud.com/help/en/model-studio/user-guide/model/
LLaMA-Factory	SFT training framework	Apache-2.0	https://github.com/hiyouga/LLaMA-Factory
ms-swift	GRPO training framework	Apache-2.0	https://github.com/modelscope/ms-swift

the original question identifier, target variant type, generated context, answer choices, and correct label. The intended use of these generated variants is robustness evaluation of LLM knowledge capabilities, not model training on test answers. The construction pipeline, variant types, filtering criteria, prompt templates, training data sizes, human evaluation protocol, and limitations are documented in Sections 3 and 4, Table 1, Appendix A, Appendix D, Appendix I, and Appendix L.

Second, SAGE trains task-specific VariantGen and VariantQual components initialized from Qwen3-0.6B. The model documentation includes the backbone model, training frameworks, LoRA configuration, SFT and GRPO hyperparameters, compute usage, and prompt templates in Appendix A, Appendix B, and Appendix L. Because the generated benchmark is derived from existing public benchmarks, any release of the derived examples should preserve attribution to the source datasets and follow their licenses or terms. No additional consent is required for human subjects, since the new assets are generated from public benchmark items and do not contain newly collected personal data.

L Prompt Templates

Figures 5–9 show the complete prompt templates used in SAGE for the VariantGen and VariantQual components.

M Use of Large Language Models

This work involves the use of LLMs in two capacities. First, LLMs are integral to the research methodology itself: SAGE employs LLM-based components (VariantGen and VariantQual) for benchmark variant generation and quality verification, and uses proprietary LLMs such as Qwen2.5-Max as baselines for comparison experiments. These usages are documented in detail throughout Sections 3 and 4. Second, LLM-based writing assistants were used to polish and improve the clarity of the manuscript text. The authors have carefully reviewed all content and take full responsibility for the correctness, originality, and integrity of the entire paper.

VariantGen BASE TEMPLATE (all variant types)

System role:

Suppose you are a case generator. Given original_context, original_choices, and original_label, your goal is to generate context, choices, label, and explanation according to perturbation_type. Your output should be a dictionary whose keys are original_context, original_choices, original_label, perturbation_type, context, choices, label, and explanation.
I will provide some examples, and you should imitate my case generation process.

[3 in-context example cases]

[Perturbation-specific instructions]

So, how to convert the following case?
original_context: [INPUT]
original_choices: [INPUT]
original_label: [INPUT]
perturbation_type: [INPUT]

Remember to only output the dictionary, not other content.

Figure 5: VariantGen base prompt template. The same template is used for all variant types.

PERTURBATION-SPECIFIC INSTRUCTIONS

Causal Inference

I want you to strictly follow the steps below to generate:

Step 1: Concatenate the original_context and original_choices corresponding to the original_label into a complete paragraph and turn it into context, ending with "Which could be the possible reason for this action?"

Step 2: Generate the reason for such choice containing common sense as the correct option in choices. The correct option should be as concise as possible, and generate 9 other obviously wrong options according to the format and length of this option. The wrong options should contain wrong common sense. Put the correct option in the first position of choices and mark the label as 0. Context and choices should be fluent.

Note: The conversion process hopes to infer possible reasons through the context and choices. The capitalization of context must be correct.

Figure 6: Perturbation instructions for VariantGen. Causal Inference is shown as an example.

Implicit VariantQual Templete

You are an expert case evaluator. Your task is to evaluate whether the conversion from original case to new case is correct based on the given perturbation type.

Evaluation Criteria:

1. **Perturbation Compliance: [perturbation-specific requirement]**
2. **Label Correctness: Evaluate if the label is correct for the converted case.**
3. **Answer Uniqueness: Check if there is only one unique correct option among the choices.**

Scoring:

Score 1 — Meets all three criteria.
Score 0 — Fails to meet any of the criteria.

Output format:

Score: [0 or 1]
Explanation: [one of the templates]

Figure 7: VariantQual implicit rubric prompt template.

Implicit VariantQual Explanation Templates

Score 1:
"The conversion is correct: meets variant requirements, label is accurate, and there is only one unique correct option."
Score 0 (compliance only):
"The conversion is incorrect: does not meet variant requirements."
Score 0 (label only):
"The conversion is incorrect: label is incorrect."
Score 0 (uniqueness only):
"The conversion is incorrect: multiple correct options exist."
Score 0 (compliance + label):
"The conversion is incorrect: does not meet variant requirements, label is incorrect."
Score 0 (compliance + uniqueness):
"The conversion is incorrect: does not meet variant requirements, multiple correct options exist."
Score 0 (label + uniqueness):
"The conversion is incorrect: label is incorrect, multiple correct options exist."
Score 0 (all):
"The conversion is incorrect: does not meet variant requirements, label is incorrect, multiple correct options exist."

Please evaluate the following conversion:
[Original and Converted Case Details]

Provide ONLY the score and explanation in the specified format.

Figure 8: VariantQual implicit explanation prompt template.

Explicit VariantQual Templete

Criterion 1: Variant Compliance

You are an expert case evaluator. Evaluate the conversion based solely on Perturbation Compliance.

Criterion: [perturbation-specific requirement]

Scoring:
Score 1 — Meets perturbation compliance requirements.
Score 0 — Fails to meet perturbation compliance requirements.

Output format:
Score: [0 or 1]
Explanation:
Score 1: "The conversion is correct: meets variant requirements."
Score 0: "The conversion is incorrect: does not meet variant requirements."

[Case Details]
Provide ONLY the score and explanation in the specified format.

Figure 9: VariantQual explicit rubric prompt template.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Introduction state the paper's contributions and scope: SAGE for scalable robustness augmentation of multiple-choice knowledge benchmarks, evaluated mainly on HellaSwag with cross-task evidence on MMLU.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3 and 4 describe the SAGE pipeline, model backbones, datasets, metrics, and evaluation protocol. Appendix A and Appendix H provide training hyperparameters, data sizes, and prompt templates needed to reproduce or verify the main results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide methodological details, hyperparameters, prompt templates, and asset documentation in Sections 3 and 4 and Appendix, and release the code and SAGE-generated datasets in supplementary materials.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 4.1 reports the model backbone, training frameworks, data split sizes, and evaluation metrics. Appendix A gives SFT and GRPO hyperparameters, including LoRA settings, learning rates, rollout settings, batch size, precision, and KL/clip parameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Table 2 reports 95% confidence intervals for the downstream robustness metrics and states how they are computed. Appendix C additionally reports Spearman and Kendall rank correlations, with significance values discussed in Section 4.4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B reports the aggregate monetary cost and GPU-hour usage of the reported experiments, including A40 and A100 compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public NLP benchmarks and generated text variants for model evaluation, and we are not aware of any deviation from the NeurIPS Code of Ethics. Appendix E and Appendix F discuss inherited benchmark biases, potential risks, and the need for expert review in sensitive domains.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Appendix F provides a dedicated Broader Impact discussion covering both positive societal impacts (democratizing benchmark construction, promoting robust model development) and potential risks (benchmark overfitting, inherited biases), along with corresponding mitigation strategies.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk pretrained model, image generator, or scraped dataset.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix G.1 lists the existing datasets, models, and software frameworks used in this work, together with their licenses or provider terms and source pages.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Appendix G.2 documents the SAGE-generated benchmark variants and trained VariantGen/VariantQual components, including intended use, data fields, training details, quality checks, limitations, and attribution requirements.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not conduct crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not involve crowdsourcing or human-subject experiments, so IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Appendix I provides a dedicated statement on LLM usage covering both experimental use (VariantGen, VariantQual, and baseline models) and writing assistance, with the authors taking full responsibility for all content.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.