

# More Agents Improve Math Problem Solving but Adversarial Robustness Gap Persists

Khashayar Alavi<sup>\*†</sup> Zhastay Yeltay<sup>\*†</sup> Lucie Flek<sup>♣♠</sup> Akbar Karimi<sup>♣♠</sup>

<sup>♣</sup>Bonn-Aachen International Center for Information Technology, University of Bonn, Germany

<sup>♠</sup>Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

{s76kalav, s06zyelt}@uni-bonn.de

ak@bit.uni-bonn.de

## Abstract

When LLM agents work together, they seem to be more powerful than a single LLM in mathematical question answering. However, are they also more robust to adversarial inputs? We investigate this question using adversarially perturbed math questions. These perturbations include punctuation noise with three intensities (10%, 30%, 50%), plus real-world and human-like typos (WikiTypo, R<sup>2</sup>ATA). Using a unified sampling-and-voting framework (Agent Forest), we evaluate six open-source models (Qwen3-4B/14B, Llama3.1-8B, Mistral-7B, Gemma3-4B/12B) across four benchmarks (GSM8K, MATH, MMLU–Math, MultiArith), with various numbers of agents  $n \in \{1, 2, 5, 10, 15, 20, 25\}$ . Our findings show that 1) Noise type matters: punctuation noise harm scales with its severity, and the human typos remain the dominant bottleneck, yielding the largest gaps to Clean accuracy and the highest attack success rate (ASR) even with a large number of agents; 2) Collaboration reliably improves accuracy as the number of agents,  $n$ , increases, with the largest gains from  $n=1$  to  $n=5$  and diminishing returns beyond  $n \approx 10$ . However, the adversarial robustness gap persists regardless of the agent count. Code is available on GitHub<sup>1</sup>.

## 1 Introduction

Large Language Models (LLMs) have made remarkable progress in solving a variety of reasoning-intensive tasks (Zhao et al., 2025), including mathematical problem solving. Beyond simply scaling model size, recent work shows that treating multiple LLM instances as *agents* and combining their outputs through debate, voting, or self-consistency can substantially boost performance. Li et al. (2024a) demonstrate that even a simple

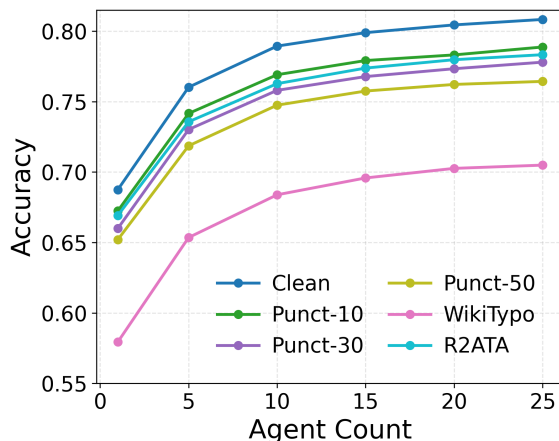


Figure 1: Average accuracy across datasets and models as a function of agent count. Lines denote noise types (Clean, Punct-10/30/50, WikiTypo, R<sup>2</sup>ATA). Markers show mean accuracy across datasets.

sampling-and-voting strategy, essentially allowing multiple agents to propose answers and then aggregating them, can rival more sophisticated collaborative frameworks. These findings suggest that collaboration between agents is a powerful and relatively simple way to enhance reasoning accuracy. At the same time, another body of research highlights a major weakness of LLMs: their fragility when inputs are perturbed. While the semantic meaning of a math problem may remain unchanged, minor surface-level changes, such as synonym replacements (Shi et al., 2023a), typos (Gemma Team et al., 2024), or even random punctuation insertions, can significantly reduce model accuracy. For instance, Abedin et al. (2025) show that inserting punctuation noise, despite leaving the problem content intact, can cause accuracy drops on well-established benchmarks like GSM8K (Cobbe et al., 2021) and MultiArith (Roy and Roth, 2016). These two perspectives converge on an important open question: *If multiple agents can enhance reasoning performance on Clean inputs, can they also*

<sup>\*</sup>Equal contribution

<sup>1</sup><https://github.com/caisa-lab/multi-agent-adversarial-robustness>

*improve robustness under noisy conditions?* Understanding whether ensemble-based multi-agent setups show resilience to noise would provide valuable insights for designing more reliable and trustworthy LLM systems.

To address this gap, we investigate the robustness of Agent Forest (a multi-LLM-agent setup) to adversarially perturbed math questions. We develop a taxonomy of noise types for adversarial modification and systematically introduce them across multiple reasoning datasets for mathematical question answering, enabling a controlled evaluation of their impact. We use ArithmAttack (Abedin et al., 2025) to generate synthetic noise, while for real-world noise we rely on natural typos from Wikipedia (Aliakbarzadeh et al., 2025) and adversarial perturbations introduced by the Adversarial Typo Attack (Gan et al., 2024). We assess the impact of input noise perturbations on open-source LLMs, specifically Qwen (Yang et al., 2025), Llama (Dubey et al., 2024), Mistral (Jiang et al., 2023), and Gemma (Gemma Team et al., 2025), across multiple parameter scales. By doing so, we evaluate not only how noise affects individual models but also whether multi-agent collaboration enhances resilience and preserves reasoning accuracy in the presence of such perturbations.

Figure 1 gives an overview of the results of our investigation. Across all experiments, we observe a consistent, monotonic gain from increasing the number of collaborating agents. Averaged over datasets and models, accuracy rises from 0.6579 with a single agent to 0.7740 with 25 agents, with the largest jump occurring between *one* and *five* agents and clear diminishing returns beyond 10.

The relative ordering of noise conditions is stable: With the Clean performance at the top, Punct-10 (i.e., inserting punctuation noise in as many spaces as 10 percent of the question length) causes a consistent accuracy drop, then even larger drops are caused by Punct-30 and Punct-50, while WikiTypo is consistently the most challenging, causing a drop of around 8% on average across all the settings with different numbers of agents. R<sup>2</sup>ATA causes a slightly larger drop than Punct-10 in our aggregate.

## 2 Related Work

Recent work has shown that current open-source large language models (LLMs) exhibit limited resilience to naturally occurring input noise, such

as spelling mistakes, typographical errors, and minor punctuation artifacts, raising serious concerns about their reliability in real-world deployments (Dong et al., 2023; Abedin et al., 2025). A substantial body of research has analyzed the impact of input perturbations across multiple granularities. At the character level, studies examine misspellings and typos (Gao et al., 2018; Li et al., 2018); at the word level, attacks involve synonym substitution or word shuffling (Garg and Ramakrishnan, 2020; Moradi and Samwald, 2021); and at the sentence level, robustness is tested via paraphrasing or insertion of irrelevant context (Shi et al., 2023b; Latham et al., 2023; Arora et al., 2025). Even seemingly innocuous changes, such as random punctuation insertion, have been shown to significantly degrade performance (Abedin et al., 2025). These perturbations have been evaluated across a range of tasks and settings. Prior analyses of LLMs under noisy inputs include text classification (Eger and Benz, 2020), natural language inference (Formento et al., 2021, 2023), and multilingual scenarios (Aliakbarzadeh et al., 2025), offering valuable insights into task-specific vulnerabilities.

In parallel, a growing line of work (Talebirad and Nadiri, 2023; Seo et al., 2025; Wang et al., 2025) demonstrates that multi-agent ensembles can substantially boost reasoning performance on Clean inputs. These systems often form complex networks of LLMs, coordinating via peer-to-peer or distributed configurations (Tran et al., 2025). For example, multi-agent debates allow models to iteratively propose and critique answers (Du et al., 2024), a process that can be further optimized using sparse communication networks to reduce error propagation (Li et al., 2024b). Despite the sophistication of such topologies, Lu et al. (2024) show that simple strategies such as majority voting often rival complex debate frameworks, allowing ensembles of smaller models to match much larger single models. However, the robustness of these collaborative setups remains an open question. Helmi (2025) provide preliminary evidence that multi-agent reasoning degrades under high noise, highlighting a critical vulnerability.

Together, these observations reveal a severe gap. While both noise robustness and multi-agent collaboration are active research areas, their intersection has not been systematically explored. Although recent frameworks continue to introduce highly complex, multi-round network topologies (Becker et al., 2025; Asad et al., 2025), understanding their

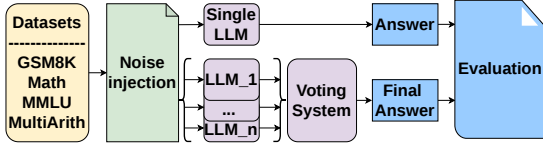


Figure 2: Overview of the experimental framework for evaluating multi-LLM-agent robustness. The system processes input questions from four datasets under Clean and noisy conditions. For each condition, a single LLM generates multiple independent answers ( $LLM_1$  to  $LLM_n$ ), which are then aggregated by majority voting to produce the final answer.

fundamental vulnerabilities first requires a solid, isolated baseline. By focusing on a sampling-and-voting topology, our work directly addresses this gap, establishing a baseline for evaluating the adversarial robustness of multi-LLM-agent systems.

### 3 Methodology

Figure 2 depicts our pipeline for evaluating the adversarial robustness of LLMs from noisy dataset generation to answer extraction.

#### 3.1 Noisy Dataset Creation

To systematically analyze the influence of noise on multi-agent LLMs, we introduce a taxonomy that distinguishes between synthetic noise and human-like noise. This taxonomy enables a structured evaluation of model performance under both artificially generated perturbations and noise patterns that more closely resemble human input.

**Synthetic noise.** Following ArithmAttack (Abedin et al., 2025), we implement synthetic noise with *punctuation insertion*, where symbols from the set  $\{".", ",", "!", "?", ";", ":"\}$  are randomly added to sentences. To study the effect of perturbation strength, each noise type is applied at three intensity levels (10%, 30%, and 50%), allowing a controlled analysis of model performance under progressively noisier conditions.

**Human-like noise.** To mimic realistic human errors, we consider two complementary approaches. The first is the Adversarial Typo Attack (ATA) (Gan et al., 2024), which evaluates the robustness of LLMs to typographical errors by iteratively introducing common character-level mistakes typically caused by keyboard usage. These include letter duplication, character substitution, deletion, or insertion, and removal of spaces, all targeted toward words that are important for the query, with

the goal of degrading model performance. For this, we rely on the  $R^2ATA$  benchmark, which applies ATA to GSM8K, BBH, and MMLU tasks. The second approach leverages naturally occurring spelling mistakes extracted from Wikipedia edit histories (Aliakbarzadeh et al., 2025). Specifically, we use the WikiTypo dictionary, which provides a collection of real-world human spelling errors, to replace words in the datasets with authentic typo variants.

#### 3.2 Agent Forest

We evaluate multi-agent performance using *Agent Forest* (Li et al., 2024a), which consists of two phases: sampling and voting. For each question  $x$  from a dataset, we query the same base model  $\mathcal{M}$  multiple times with independent random seeds and fixed decoding settings. Each call acts as one agent and produces one candidate solution  $s_i$ . After  $N$  calls we obtain the set of candidates  $S = \{s_1, \dots, s_N\}$ .

**Answer extraction and normalization.** For arithmetic datasets, we extract the final numeric answer from each  $s_i$  using simple patterns that look for common markers such as Answer or `\boxed{\}`. We then extract the value by removing formatting artifacts, converting textual numbers to numerals, and standardizing whitespace. This yields a canonical answer  $\text{canon}(s_i)$  for each sample.

**Aggregation by voting.** To specify the final answer, we aggregate candidates by majority vote over canonical answers.

**Protocol and replication.** For every model and dataset, we generate exactly 25 independent single-agent outputs per question with fixed decoding parameters. The threshold for the number of agents was based on a preliminary study, indicating that variance stabilizes well before this point. To study different agent counts  $n \in \{1, 2, 5, 10, 15, 20, 25\}$ , we partition them into  $\lfloor 25/n \rfloor$  disjoint groups of size  $n$  in their generation order. Each group forms one independent  $n$  agent ensemble whose vote yields one prediction for that question. Thus we obtain 25 independent predictions for  $n = 1, 12$  for  $n = 2, 5$  for  $n = 5$ , and 2 for  $n = 10$ . For  $n = 15, n = 20$ , and  $n = 25$ , only one ensemble is available, so we use the first  $n$  outputs. At the dataset level, we compute accuracy for each independent group and report the mean and the standard deviation across groups when multiple groups exist. No

fine-tuning or task-specific supervision is used, and decoding settings are identical for Clean and noisy inputs so that any change in accuracy is attributable to aggregation by Agent Forest.

### 3.3 Evaluation Metrics

We report accuracy and Attack Success Rate (ASR). Let  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$  be a data set with gold answers  $y_i$ . Let  $f_n$  denote the  $n$  agent ensemble predictor that maps an input  $x$  to a single answer by Agent Forest voting as defined in Section 3.2. The metric is:

**Accuracy.** Clean accuracy for  $n$  agents:

$$\text{Acc}(n) = \frac{1}{m} \sum_{i=1}^m \mathbf{1}[f_n(x_i) = y_i].$$

**Attack Success Rate (ASR).** ASR measures the fraction of originally correct predictions that are flipped to incorrect after applying the noise. Formally,

$$\text{ASR}(n) = \frac{\sum_{i=1}^m \mathbf{1}[f_n(A(x_i)) \neq y_i]}{\sum_{i=1}^m \mathbf{1}[f_n(x_i) = y_i]}.$$

ASR is thus defined on the subset of items that the model with  $n$  agents solves correctly in the Clean setting and takes values in  $[0, 1]$ . A lower value indicates better robustness.

**Reporting.** As described in Section 3.2, for each model and dataset, we generate 25 independent single-agent outputs per question and form disjoint groups to realize different agent counts  $n \in \{1, 2, 5, 10, 15, 20, 25\}$ . For  $n \in \{1, 2, 5, 10\}$  this yields multiple independent groups; we compute Acc and ASR for each group and report the mean and the standard deviation across groups. For  $n \in \{15, 20, 25\}$  only one group is available; we report the single value.

## 4 Experiments

All experiments were executed on HPC GPU Cluster 8 x NVIDIA GA100 [A100] GPUs with 80GB, running under CUDA 12.2. We used Python 3.10 with the PyTorch 2.2 and HuggingFace Transformers (Wolf et al., 2020) libraries. For efficient inference, we employed vLLM (Kwon et al., 2023), which allows fast batched decoding for large-scale evaluation.

### 4.1 Datasets

Evaluation of mathematical reasoning capabilities is conducted across four established benchmarks: **GSM8K** (Cobbe et al., 2021), **MATH** (Hendrycks et al., 2021b), the mathematics subset of **MMLU** (Hendrycks et al., 2021a), and **MultiArith** (Roy and Roth, 2016). This selection provides a comprehensive assessment spanning difficulty levels from basic arithmetic to advanced competition problems while incorporating both open-ended and multiple-choice formats.

**GSM8K** (Cobbe et al., 2021) contains 8.5K linguistically diverse grade school math word problems, with evaluation performed on the standard 1.3K test questions requiring multi-step arithmetic reasoning. Its broad adoption as a mathematical problem-solving benchmark offers a standard reference point for studying reasoning robustness under input noise.

The **MATH** dataset (Hendrycks et al., 2021b) consists of 12.5K challenging mathematics competition problems covering advanced topics, including algebra, geometry, and number theory. These problems require symbolic manipulation and multi-step derivations, enabling assessment of how adversarial noise affects complex problem-solving capabilities compared to elementary-level reasoning.

Evaluation using **MMLU** (Hendrycks et al., 2021a) focuses specifically on the mathematics subsets, which provide multiple-choice questions testing factual recall and mathematical reasoning. This format complements the open-ended generation tasks from other datasets.

**MultiArith** (Roy and Roth, 2016) consists of 600 arithmetic word problems requiring multi-step operations, with evaluation on the standard 180-question test set. This dataset serves as a lightweight benchmark for studying noise effects on simple yet compositional reasoning tasks.

### 4.2 Models

To evaluate a diverse range of capabilities while maintaining computational efficiency, a selection of modern open-source language models of varying parameter sizes was chosen for evaluation to analyze the impact of model scale. The models comprise Qwen3-4B-Instruct and Qwen3-14B-Instruct (Yang et al., 2025), Llama-3.1-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), along with Gemma3-4B-IT and Gemma3-

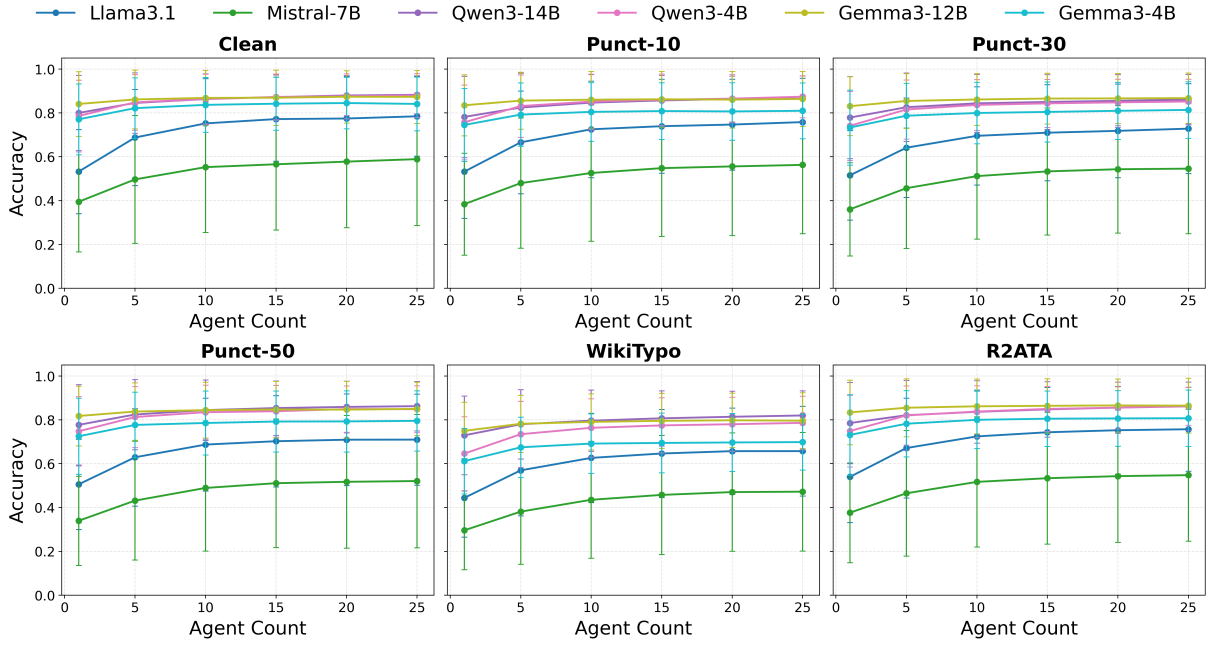


Figure 3: Accuracy averaged across datasets by noise type (Clean, Punct-10/30/50, WikiTypo, R<sup>2</sup>ATA) as a function of agent count. Larger models (Gemma3-12B and Qwen3-14B) maintain higher absolute accuracy across conditions, while smaller (and older) models (Llama3.1 and Mistral-7B) show steeper relative gains with more agents. New small models (Gemma3-4B and Qwen3-4B) show similar performances to their larger counterparts.

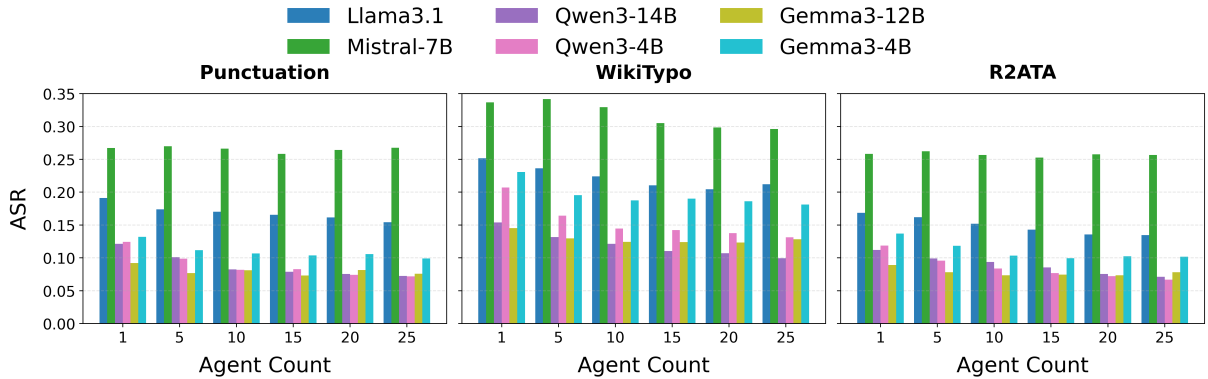


Figure 4: Attack Success Rate averaged across datasets, shown by noise category (Punctuation, WikiTypo, R<sup>2</sup>ATA). Bars represent different models at varying agent counts. Punctuation noise vulnerability is most effectively reduced with more agents. WikiTypo remains the most challenging across all models, and R<sup>2</sup>ATA lies in between, initially harmful for single agents but increasingly mitigated through collaboration.

12B-IT (Gemma Team et al., 2025). These models will be subsequently referred to as **Qwen3-4B**, **Qwen3-14B**, **Llama3.1**, **Mistral-7B**, **Gemma3-4B**, and **Gemma3-12B**, respectively.

## 5 Results

The overall trends, regarding the impact of each adversarial perturbation method, show themselves when we average the results over datasets and models. This can be seen in Figure 1, depicting that all the methods cause a consistent robustness gap, with WikiTypo having the largest one. In the fol-

lowing, we report the results and the trends across the adversarial models, datasets, and adversarial noise, respectively.

### 5.1 Models

Figure 3 (also Table 1 in detail in Appendix B) presents the impact of input perturbations across models and agent counts. Increasing the number of agents reliably improves performance under all noise types. The largest improvements occur between  $n=1$  and  $n=5$ , with diminishing returns beyond  $n \geq 10$ . As the agent count approaches  $n=15$ ,

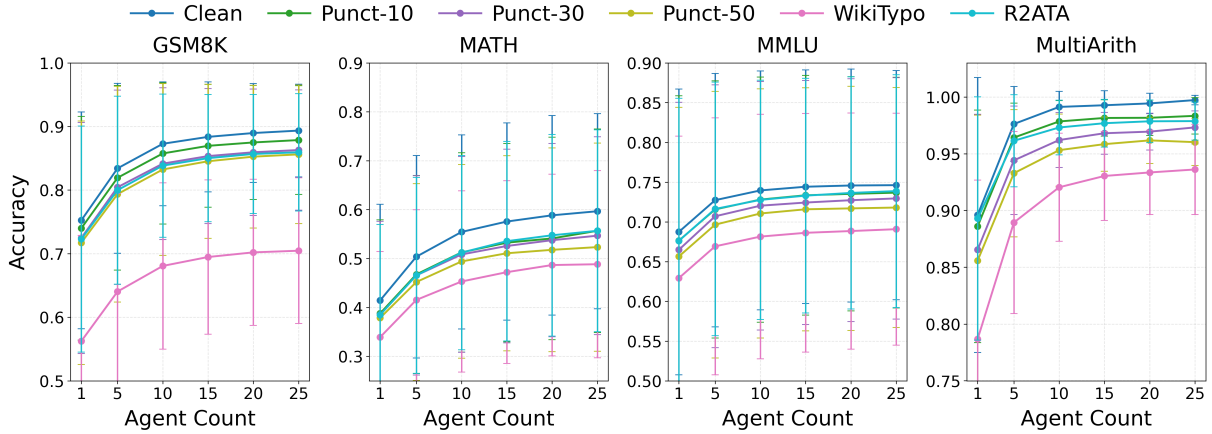


Figure 5: Accuracy averaged across models, shown separately for each dataset as a function of agent count. Collaboration improves accuracy under all noise types, with the largest gains on MATH, MultiArith, and GSM8K. In contrast, MMLU shows smaller absolute drops under noise and saturates more quickly with increasing agent count.

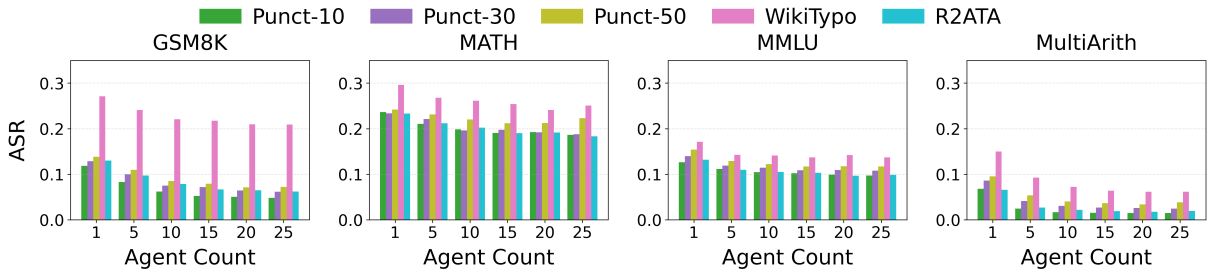


Figure 6: Attack Success Rate averaged across models, shown separately for each dataset. Noise impact differs by task: more difficult datasets, such as MATH and MMLU, exhibit higher ASR, which remains stable with the increase of agents, while easier datasets (GSM8K and MultiArith) show lower ASR values and more benefits from increasing agent count.

the accuracy curves smooth out (we also see this with agent numbers up to 50 as shown in the extended ablation in Figure 8 in Appendix B). The performance delta among  $n \in \{25, 30, 40, 50\}$  remains negligible and is comparable to the minor fluctuations observed across runs, indicating that the ensemble effectively reaches a steady state and justifying our focus on agent counts up to  $n=25$  for the primary evaluation.

As shown in Figure 3, accuracy decreases as punctuation intensity rises from Punct-10 to Punct-50. This degradation is visible at the single-agent level ( $n=1$ ), where models lose between 5 to 15 percentage points compared to the Clean baseline. However, collaboration rapidly compensates for these losses by  $n=5$ , the performance gap between Punct-10 and Clean inputs becomes negligible for most model families, and even Punct-30 and Punct-50 recover a substantial share of their accuracy.

Despite the accuracy gains from collaboration, noise affects models similarly. As we can see in Figure 4, the ASR values stay almost the same when increasing the number of agents. This indicates that under simple surface-level perturbations, the collaboration between agents is as vulnerable to noise as a single LLM.

## 5.2 Datasets

Figures 5 and 6 reveal distinct robustness profiles by dataset. MATH is the most difficult benchmark in absolute terms. Accuracy starts lowest and improves with more agents, yet gaps to the Clean baseline remain under all types of adversarial inputs even at large agent counts. Its ASR is also the highest among the datasets and hardly declines with increasing  $n$ , indicating that long, multi-step solutions are sensitive to both synthetic punctuation and human-like typos.

For the GSM8K dataset, moving from one to five agents recovers much of the loss from punctuation

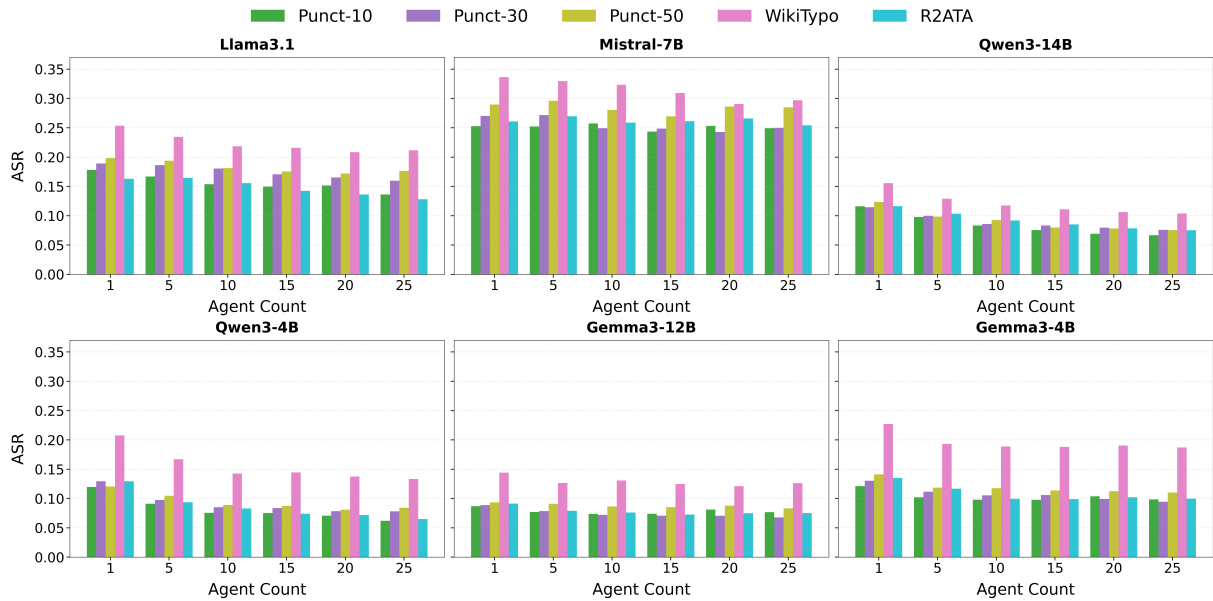


Figure 7: Attack Success Rate averaged across datasets, shown per model family and agent count. Generally, all the investigated noise types consistently decrease the model performance across different numbers of agents. WikiTypo, however, seems to be the strongest across the models and agent counts.

noise, and ASR drops steadily with  $n$ . WikiTypo consistently induces the largest degradation at a fixed agent count, suggesting heterogeneous human errors are more disruptive than punctuation insertions on these short, compositional problems. MMLU (math subset) is comparatively stable. Accuracies cluster tightly and rise smoothly with agent count, but its ASR is slightly higher than in GSM8K across noise types.

MultiArith is near the ceiling. Performance in this dataset is higher than other datasets both with and without noise. In addition, ASR is the lowest, which can be attributed to the fact that the dataset is less difficult compared to others. The short, structured arithmetic chains offer limited headroom for noise to reduce performance or for collaboration to yield large additional gains.

Overall, easier datasets with simpler tasks (GSM8K, MultiArith) are less affected by noise, which is reflected in their lower ASR values. Any performance degradation can be compensated for more easily through multi-agent collaboration. In contrast, more challenging datasets (MATH and MMLU) suffer greater absolute performance drops under noise, and even with larger agent counts, the recovery remains incomplete, highlighting that task difficulty amplifies the impact of input perturbations.

### 5.3 Adversarial noise types

Figure 7 presents the Attack Success Rate (ASR) aggregated across datasets, broken down by model family and agent count. Two major patterns emerge. Across all conditions, we observe an almost stable ordering of impact: WikiTypo  $>$  Punct-50  $\geq$  Punct-30  $\geq$  R<sup>2</sup>ATA  $\geq$  Punct-10, with R<sup>2</sup>ATA typically close to the Clean baseline. This ranking can be more visibly seen in Figure 1.

However, the punctuation noise types as well as R<sup>2</sup>ATA seem to generally have a similar negative impact on the performance of all the investigated models. This impact is also somewhat stable as we increase the number of agents. By contrast, WikiTypo remains the dominant bottleneck. Even at  $n=25$ , ASR typically stays above 0.10, far higher than for punctuation. Human typos introduce heterogeneous and context-dependent distortions that majority voting cannot fully absorb, making them the most persistent source of errors.

## 6 Discussion

Our results reveal two distinct failures in multi-agent reasoning under adversarial perturbations:

**Semantic vs. Morphological Noise.** We attribute the disproportionate impact of human spelling errors (WikiTypo) compared to punctuation noise to the degradation of semantic integrity during tokenization. Punctuation noise, while visu-

ally distracting, often preserves the morphological root of keywords, allowing the model’s attention mechanism to recover the original intent. In contrast, real-world typos fundamentally alter the input’s representation in the embedding space. This causes a drastic divergence from the target semantic vector, a gap that the smaller models in our study (Mistral-7B, Llama-3.1-8B) cannot bridge as the perturbed tokens might map to unrelated or out-of-vocabulary ones.

**The Consensus Paradox.** While classic ensemble theory assumes that agent errors are statistically independent (Condorcet, 1785; Theisen et al., 2023), adversarial noise introduces a systematic bias. This bias forces the underlying model into specific, repeatable failure modes. Consequently, the agents do not exhibit the random variance required for self-correction through majority voting. Instead, they converge on the same erroneous output, causing the consensus mechanism to replicate the systematic failure rather than mitigate it. This could explain why the robustness gap remains nearly constant despite increasing the number of agents.

**Trend Analysis.** By aggregating these results, we move beyond simple repetition of dataset performance, observing that collaboration boosts the baseline accuracy but fails to compensate for the reasoning failure of the system against adversarial distortions. The consistency of this gap across six different model families suggests a fundamental limitation in how LLM agents process non-standard orthography.

## 7 Conclusion

This work provides a systematic study of multi-LLM-agent robustness for question answering under synthetic and human-like input noise. Using a unified sampling-and-voting framework across four datasets (GSM8K, MATH, MMLU–Math, Multi-Arith) and six open-source models, we examined accuracy and ASR as a function of the number of collaborating agents. Three findings emerge: First, collaboration reliably improves accuracy, with the largest gains between one and five agents and diminishing returns beyond ten. However, ASR remains almost stable as the agent count increases. This pattern holds across models and datasets, indicating that multi-agent collaboration via majority voting is effective in improving accuracy, but the

adversarial robustness gap persists.

Second, the noise type matters. Punctuation corruption degrades performance roughly in proportion to its intensity, yet is largely neutralized by five to ten agents. Human typos (WikiTypo) remain the dominant bottleneck: WikiTypo consistently induces the largest gaps to Clean accuracy and the highest ASR, even at high agent counts. R<sup>2</sup>ATA (an adversarial perturbation method) occupies a middle ground. Finally, model-level comparisons show that robustness does not strictly correlate with parameter count. While the midsize models show the highest attack success rates, both the larger and the smaller ones show more robustness. Within families such as Qwen and Gemma, larger models consistently achieve lower ASR values and thus demonstrate stronger inherent robustness. Their smaller counterparts show similar robustness while benefiting more from collaboration. Promising directions include noise-aware sampling and aggregation objectives, verifier- or tool-assisted agents, and training-time augmentation targeted at human typos. Extending beyond our scope, open-source models, and character-level corruptions to other domains, languages, and real-world pipelines will test the breadth of these gains. Finally, our framework can also be used to evaluate the robustness of different agentic topologies to adversarial noise.

## 8 Limitations

This study has several limitations. First, we restrict evaluation to English, math-reasoning benchmarks, and character-level corruptions; we do not test multilingual data, formatting/layout perturbations (e.g., LaTeX/OCR noise), or adaptive adversaries that react to model outputs. Second, our multi-agent scheme is a single instantiation-independent sampling from the same base model with fixed prompts/decoding and simple majority voting, so we do not compare against debate, tool-augmented agents, verifier-guided aggregation, diversity-promoting sampling, or cost/latency-aware strategies; the efficiency–robustness frontier therefore remains underexplored. Third, answer extraction is based on rule-based numeric parsing. Residual parsing errors can inflate ASR or deflate accuracy. Fourth, all agents share the same base model, so errors can be correlated, and our analysis does not disentangle correlation structure from the benefits of voting. Fifth, limited computational and financial resources constrained us from scaling the full eval-

uation to  $n=100$  or sweeping various hyperparameters. However, we submit that our protocol remains robust for the claims made. A pilot study on a data subset scaling to  $n=50$  confirmed that the adversarial robustness gap remains constant once the ensemble variance stabilizes at lower agent counts.

## 9 Ethical Considerations

The persistence of the adversarial robustness gap has significant implications for the deployment of LLM agents in safety-critical, high-stakes environments. As these systems are integrated into financial modeling, medical coding, or legal reasoning, their susceptibility to simple typographical errors represents a severe reliability risk. Our findings emphasize that "scaling agents" is not a substitute for "improving base-model robustness." We highlight the necessity for character-robust tokenization or pre-processing guardrails before deploying multi-agent systems in environments where human-entered data, which is naturally noisy, is the primary input.

In addition to these safety concerns, we have conducted this research following standard ethical guidelines. We evaluate only open-source LLMs on public math benchmarks (GSM8K, MATH, MMLU–Math, MultiArith), which contain no personal or sensitive data. No human subjects were involved. Practitioners must assess risks carefully when transferring these methods beyond research. To polish the language and code, we used AI assistants, manually controlling the generated content to avoid producing harmful or biased material.

## Acknowledgments

This work was supported by the BMFTR and the state of North Rhine-Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence and the AALearning Project, funded under the BMFTR grant proposal No. 05D25PD2.

## References

Zain Ul Abedin, Shahzeb Qamar, Lucie Flek, and Akbar Karimi. 2025. Arithmattack: Evaluating robustness of llms to noisy context in math problem solving. *arXiv preprint arXiv:2501.08203*. <https://arxiv.org/abs/2501.08203>.

Amirhossein Aliakbarzadeh, Lucie Flek, and Akbar Karimi. 2025. [Exploring robustness of multilin-](#)

[gual llms on real-world noisy data](#). *Preprint*, arXiv:2501.08322.

- Pulkit Arora, Akbar Karimi, and Lucie Flek. 2025. Exploring robustness of llms to paraphrasing based on sociodemographic factors. *arXiv preprint arXiv:2501.08276*.
- Ali Asad and 1 others. 2025. Reddebate: Safer responses through multi-agent red teaming debates. *arXiv preprint arXiv:2506.11083*.
- Jonas Becker and 1 others. 2025. Mallm: Multi-agent large language models framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Marquis de Condorcet. 1785. Essay on the application of analysis to the probability of majority decisions. *Paris: Imprimerie Royale*, page 1785.
- Guanting Dong, Jinxu Zhao, Tingfeng Hui, Daichi Guo, Wenlong Wan, Boqi Feng, Yueyan Qiu, Zhuoma Gongque, Keqing He, Zechen Wang, and Weiran Xu. 2023. [Revisit input perturbation problems for llms: A unified robustness evaluation framework for noisy slot filling task](#). *Preprint*, arXiv:2310.06504.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.
- Steffen Eger and Yannik Benz. 2020. From hero to z\`eroe: A benchmark of low-level adversarial attacks. *arXiv preprint arXiv:2010.05648*.
- Brian Formento, Chuan-Sheng Foo, Luu Anh Tuan, and See Kiong Ng. 2023. Using punctuation as an adversarial attack on deep learning-based nlp systems: An empirical study. In *Findings of the association for computational linguistics: EACL 2023*, pages 1–34.
- Brian Formento, See-Kiong Ng, and Chuan-Sheng Foo. 2021. Special symbol attacks on nlp systems. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Esther Gan, Yiran Zhao, Liying Cheng, Yancan Mao, Anirudh Goyal, Kenji Kawaguchi, Min-Yen Kan, and Michael Shieh. 2024. [Reasoning robustness of llms to adversarial typographical errors](#). *Preprint*, arXiv:2411.05345.

- Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. [Black-box generation of adversarial text sequences to evade deep learning classifiers](#). In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.
- Siddhant Garg and Goutham Ramakrishnan. 2020. [BAE: BERT-based adversarial examples for text classification](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 2 others. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, and Léonard Hussenot. 2024. [Gemma 2: Improving open language models at a practical size](#). *Preprint*, arXiv:2408.00118.
- Tooraj Helmi. 2025. [Modeling response consistency in multi-agent llm systems: A comparative analysis of shared and separate context approaches](#). *Preprint*, arXiv:2504.07303.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. [Measuring massive multitask language understanding](#). *Preprint*, arXiv:2009.03300.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. [Measuring mathematical problem solving with the math dataset](#). *Preprint*, arXiv:2103.03874.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiušė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. [Measuring faithfulness in chain-of-thought reasoning](#). *Preprint*, arXiv:2307.13702.
- Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. [Textbugger: Generating adversarial text against real-world applications](#). *arXiv preprint arXiv:1812.05271*.
- Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024a. [More agents is all you need](#). *Preprint*, arXiv:2402.05120.
- Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024b. [Improving multi-agent debate with sparse communication topology](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7281–7294.
- Xiaoding Lu, Zongyi Liu, Adian Liusie, Vyas Raina, Vineet Mudupalli, Yuwen Zhang, and William Beauchamp. 2024. [Blending is all you need: Cheaper, better alternative to trillion-parameters llm](#). *Preprint*, arXiv:2401.02994.
- Milad Moradi and Matthias Samwald. 2021. [Evaluating the robustness of neural language models to input perturbations](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Subhro Roy and Dan Roth. 2016. [Solving general arithmetic word problems](#). *Preprint*, arXiv:1608.01413.
- Wonduk Seo, Juhyeon Lee, and Yi Bu. 2025. [Spio: Ensemble and selective strategies via llm-based multi-agent planning in automated data science](#). *Preprint*, arXiv:2503.23314.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023a. [Large language models can be easily distracted by irrelevant context](#). *Preprint*, arXiv:2302.00093.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023b. [Large language models can be easily distracted by irrelevant context](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.
- Yashar Talebirad and Amirhossein Nadiri. 2023. [Multi-agent collaboration: Harnessing the power of intelligent llm agents](#). *Preprint*, arXiv:2306.03314.
- Ryan Theisen, Hyunsuk Kim, Yaoqing Yang, Liam Hodgkinson, and Michael W Mahoney. 2023. [When are ensembles really effective?](#) *Advances in neural information processing systems*, 36:15015–15026.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D. Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of LLMs. *arXiv preprint arXiv:2501.06322*.

Yaoxiang Wang, Zhiyong Wu, Junfeng Yao, and Jinsong Su. 2025. Tdag: A multi-agent framework based on dynamic task decomposition and agent generation. *Preprint*, arXiv:2402.10178.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Hugging-face’s transformers: State-of-the-art natural language processing](#). *Preprint*, arXiv:1910.03771.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025. [A survey of large language models](#). *Preprint*, arXiv:2303.18223.

## A Qualitative Failure Cases

The following are some of the failure cases under different attacks:

Dataset: MATH

Noise type: punctuation\_10

Ground truth:  $\frac{9}{4}, -\frac{9}{4}$

Question:

Here is a math problem written in LaTeX: Find all real numbers  $k$  such that  $x^4 + kx^3 + x^2 + 4kx + 16 = 0$  is true for exactly , one real number  $x = r$ . Enter all the possible values of !  $k$ , separated by commas. Solve it step by step. - If the correct answer has units, write the units OUTSIDE the box in plain text. - Inside `\boxed{...}` put only the Latex style exact numeric expression (with units) (e.g., rationalized denominator, reduced fractions, simplified radicals). At the very end of your response, output exactly one line:

`\boxed{<simplest_exact_value>}`

If units are required, add as a `\text{ unit }` and then the units in words after the box, e.g.: `\boxed{ 12 \text{ square inches } }` Do not add any extra commentary after that line.

**Output:**

Let  $P(x) = x^4 + kx^3 + x^2 + 4kx + 16$ . We require  $P(x) = 0$  to have exactly one real root  $x = r$ . We can rewrite  $P(x)$  as:  $P(x) = x^4 + x^2 + 16 + kx^3 + 4kx = (x^2 + 1)^2 + kx(x^2 + 4) = 0$ .

If  $k = 0$ , then  $P(x) = x^4 + x^2 + 16$ . The discriminant of  $x^4 + x^2 + 16$  is  $16 - 4(16) = 16 - 64 = -48 < 0$ . Since the leading coefficient is positive,  $P(x) > 0$  for all real  $x$ . Therefore,  $P(x) = 0$  has no real roots.

Consider  $k \neq 0$ . **Suppose  $r$  is a repeated root of  $P(x)$ . Then  $P(r) = 0$  and  $P'(r) = 0$ .**  $P'(x) = 4x^3 + 3kx^2 + 2x + 4k$ .  $P(r) = r^4 + kr^3 + r^2 + 4kr + 16 = 0$   $P'(r) = 4r^3 + 3kr^2 + 2r + 4k = 0$   $4r^3 + 3kr^2 + 2r + 4k = 0 \Rightarrow k(3r^2 + 4) = -4r^3 - 2r$  If  $3r^2 + 4 \neq 0$ , then  $k = \frac{-4r^3 - 2r}{3r^2 + 4}$ . Substituting this into

$P(r) = 0$ , we get  $r^4 + \frac{-4r^3 - 2r}{3r^2 + 4}r^3 + r^2 + 4\frac{-4r^3 - 2r}{3r^2 + 4}r + 16 = 0$   $r^4 + \frac{-4r^6 - 2r^4}{3r^2 + 4} + r^2 + \frac{-16r^4 - 8r^2}{3r^2 + 4} + 16 = 0$  Multiplying by  $3r^2 + 4$ , we get  $r^4(3r^2 + 4) - 4r^6 - 2r^4 + r^2(3r^2 + 4) - 16r^4 - 8r^2 + 16(3r^2 + 4) = 0$   $3r^6 + 4r^4 - 4r^6 - 2r^4 + 3r^4 + 4r^2 - 16r^4 - 8r^2 + 48r^2 + 64 = 0$   $-r^6 - 11r^4 + 44r^2 + 64 = 0$   $r^6 + 11r^4 - 44r^2 - 64 = 0$  Let  $y = r^2$ . Then  $y^3 + 11y^2 - 44y - 64 = 0$ . Since  $r$  is real,  $y = r^2 \geq 0$ .

However, we are seeking for exactly one real root. Consider the case where  $x^4 + kx^3 + x^2 + 4kx + 16 = (x^2 + ax + b)(x^2 + cx + d)$  where  $a, b, c, d$  are real numbers. Since  $P(x)$  has degree 4, we have  $x^4 + kx^3 + x^2 + 4kx + 16 = (x^2 + ax + b)(x^2 + cx + d) = x^4 + (a + c)x^3 + (ac + b + d)x^2 + (ad + bc)x + bd$  Comparing coefficients, we have  $a + c = k$   $ac + b + d = 1$   $ad + bc = 4k$   $bd = 16$  If  $b = d = 4$ , then  $ac + 8 = 1$ , so  $ac = -7$ .  $4a + 4c = 4k$ , so  $a + c = k$ . So  $a + c = k$

and  $ac = -7$ .  $a$  and  $c$  are roots of the equation  $t^2 - kt - 7 = 0$ . The discriminant is  $k^2 + 28$ . For  $a, c$  to be real,  $k^2 + 28 \geq 0$ , which is always true.  $t = \frac{k \pm \sqrt{k^2 + 28}}{2}$ . Thus  $a = \frac{k + \sqrt{k^2 + 28}}{2}$  and  $c = \frac{k - \sqrt{k^2 + 28}}{2}$ . Then  $P(x) = (x^2 + (\frac{k + \sqrt{k^2 + 28}}{2})x + 4)(x^2 + (\frac{k - \sqrt{k^2 + 28}}{2})x + 4)$  We want exactly one real root. We need  $x^2 + (\frac{k - \sqrt{k^2 + 28}}{2})x + 4 = 0$  to have no real roots, so  $(\frac{k - \sqrt{k^2 + 28}}{2})^2 - 4 \cdot 1 < 0$ .  $\frac{k^2 - 2k\sqrt{k^2 + 28} + k^2 + 28}{4} - 4 < 0$   $2k^2 + 28 - 2k\sqrt{k^2 + 28} < -16$   $2k^2 + 44 < 2k\sqrt{k^2 + 28}$   $k^2 + 22 < k\sqrt{k^2 + 28}$  Squaring both sides:  $(k^2 + 22)^2 < k^2(k^2 + 28)$   $k^4 + 44k^2 + 484 < k^4 + 28k^2$   $16k^2 < -484$ , which is impossible since  $k$  is real. If  $b = d = 16$ , then  $ac + 16 + 16 = 1$ , so  $ac = -31$ .  $16a + 16c = 4k$ , so  $a + c = \frac{1}{4}k$ . Then  $a$  and  $c$  are roots of  $t^2 - \frac{k}{4}t - 31 = 0$ .  $t = \frac{\frac{k}{4} \pm \sqrt{\frac{k^2}{16} + 124}}{2} = \frac{k \pm \sqrt{k^2 + 1984}}{8}$ . Then we require  $\frac{k^2}{16} + 124 < 0$ , which is impossible. Let  $k = 0$ . Then  $x^4 + x^2 + 16 = 0$ . This has no real roots. Let  $P(x) = 0$ . If  $P(x)$  has only one real root then  $P'(x) = 0$  at that root. Thus  $x = r$  must be a double root. Then  $P(r) = 0$  and  $P'(r) = 0$ . Let  $k = 2\sqrt{2}$ .  $P(x) = x^4 + 2\sqrt{2}x^3 + x^2 + 4\sqrt{2}x + 16$ . Final Answer: The final answer is

**Takeaway:** This example shows how punctuation noise can disrupt the model's reasoning in complex mathematical problems. The model incorrectly assumes that exactly one real root must be a repeated root, leading to a flawed approach and incorrect conclusion. This highlights that noise can cause the model to make invalid assumptions and fail to apply correct mathematical principles.

Dataset: GSM8K

Noise type: Wikitypo

Ground truth: 18

Question:

Can you solve the following math problem? Janet's ducks lay 16 eggs por day. The eats three por breakfast every morning Rand bakes muffins por heir friends every days

Width four. The sells Then remainder nat Then farmers' markets daily por \$2 por fresh dock egg. How much im dollars does The made every days nat Then farmers' market? Explain your reasoning. Your final answer should be a single numerical number, in the form , at the end of your response.

**Output:**

Here's how to break down the problem and solve it step-by-step: 1. **\*\*Eggs laid per day:\*\*** Janet's ducks lay 16 eggs per day. 2. **\*\*Eggs eaten:\*\*** Each day, the ducks eat 3 eggs. 3. **\*\*Eggs remaining after being eaten:\*\*** 16 eggs - 3 eggs = 13 eggs 4. **\*\*Muffins baked per day:\*\*** Janet bakes 4 muffins per day. This information is irrelevant to the final result. 5. **\*\*Eggs sold:\*\*** Janet sells the remaining 13 eggs. 6. **\*\*Income from egg sales:\*\*** She sells each egg for \$2, so she makes 13 eggs \* \$2/egg = \$26. 7. **\*\*Total earnings:\*\*** Janet makes \$26 per day at the farmers' market.

**Takeaway:** This example illustrates how Wikitypo noise (intentional typos/misspellings) can cause the model to overlook critical arithmetic operations. The noisy text changed "eats three for breakfast" to "eats three por breakfast" and introduced other distracting misspellings like "por" instead of "for," "nat" instead of "at," and "im" instead of "in." While the model correctly extracted the key numbers (16 eggs, 3 eaten, \$2 per egg), it failed to recognize that the muffins information was entirely irrelevant noise and incorrectly performed the subtraction (16-3=13) to get \$26, when the correct interpretation should have been that all 16 eggs are sold directly without any subtraction for breakfast consumption. The typos obscured the logical relationships between quantities, leading to an incorrect arithmetic operation and final answer.

Dataset: MultiArith

Noise type: punctuation\_50

Ground truth: 72

Question:

Can you solve the following math problem? Megan . ; earned ? ; ? 8 . points ? for

each bag of cans she recycled. . If . , she had , 14 bags, but didn't recycle 5 : of them, how ; many points ! would she have earned? Explain your reasoning. Your final answer should be a single numerical number, in the form `answer`, at the end of your response.

**Output:**

First, let's figure out the points Megan earned for recycling each bag of cans. She earned 8 points for each bag.

Maybe we can add that up for all the bags she recycled. That would be 8 points / bag times 14 bags = `112`

However, she didn't recycle 5 bags, so we can't add anything for those un-recycled bags.

Therefore, she would have earned `112` points before she recycled only 14 bags of cans.

**Output:**

Statement 1 is False. In  $R^2$ , we can find two vectors that are linearly dependent (e.g.,  $v_1 = (1, 0)$  and  $v_2 = (2, 0)$ ). **Statement 2 is True.** If  $V = \text{span}(v_1, \dots, v_k)$  and  $v_1, \dots, v_k$  are linearly independent, then the vectors  $v_1, \dots, v_k$  form a basis for  $V$ , and the dimension of  $V$  is equal to the number of vectors in the set, which is  $k$ . `(B)`

**Takeaway:** This example demonstrates how character substitution noise ( $R^2$ ATA) can lead to logical inconsistencies in the model's reasoning. The model correctly identifies Statement 1 as false, but then contradicts itself by stating Statement 2 is true while selecting option (B) which requires both statements to be false, showing confusion in maintaining logical consistency throughout the response.

**Takeaway:** This example demonstrates how inserting punctuation noise can lead to task misunderstanding. The original problem intended that Megan earned 112 points for 14 bags, it noted the 5 un-recycled bags but the model failed to subtract them from the total, resulting in an incorrect calculation.

Dataset: MMLU

Noise type:  $R^2$ ATA

Ground truth: (B)

**Question:**

Can you answer the following question as accurately as possible? Question: Statement 1 | Any set of two vectors in  $R^2$  is linearly independent. Statement 2 | If  $V = \text{span}(v_1, \dots, v_k)$  and  $v_1, \dots, v_k$  are linearly independent, then  $\dim(V) = k$ . : Choices: A) True, True, B) False, False, C) True, False, D) False, True. Briefly explain your reasoning. Choose only one of A, B, C, or D. At the end, output only one final answer like `(A)`, `(B)`, `(C)`, or `(D)`. Do not add extra commentary after the answer. Do not include anything else on that final line.

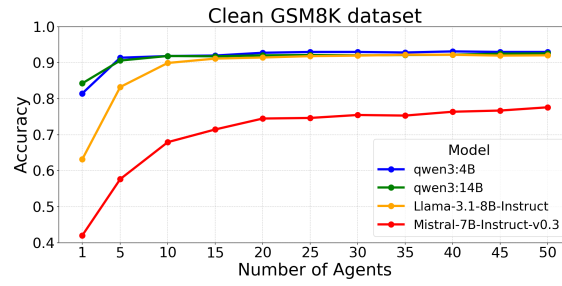


Figure 8: Model accuracy trend by the number of agents (Noise type: clean). The graph shows the performance of four different models (qwen3:4B, qwen3:14B, Llama-3.1-8B-Instruct, and Mistral-7B-Instruct-v0.3) as the number of agents increases from 1 to 50. All models demonstrate improved accuracy with more agents, with qwen3:14B achieving the highest performance, followed closely by qwen3:4B and Llama-3.1-8B-Instruct. Mistral-7B-Instruct-v0.3 shows lower overall accuracy but still benefits from the multi-agent approach.

## B Extended Results

Model	Agent Count	Clean	Punct-10	Punct-30	Punct-50	WikiTypo	R <sup>2</sup> ATA	Avg
Llama-3.1	1	0.532 ± 0.192	0.532 ± 0.213	0.515 ± 0.204	0.505 ± 0.206	0.444 ± 0.177	0.540 ± 0.208	0.511
	5	0.688 ± 0.219	0.666 ± 0.234	0.641 ± 0.227	0.626 ± 0.220	0.570 ± 0.206	0.671 ± 0.227	0.644
	10	0.752 ± 0.205	0.725 ± 0.220	0.695 ± 0.224	0.686 ± 0.211	0.626 ± 0.203	0.724 ± 0.212	0.701
	20	0.775 ± 0.192	0.747 ± 0.207	0.718 ± 0.213	0.709 ± 0.209	0.657 ± 0.197	0.753 ± 0.200	0.727
Mistral-7B	1	0.395 ± 0.228	0.384 ± 0.233	0.360 ± 0.213	0.340 ± 0.203	0.296 ± 0.178	0.376 ± 0.228	0.359
	5	0.497 ± 0.292	0.480 ± 0.297	0.457 ± 0.274	0.431 ± 0.271	0.381 ± 0.239	0.465 ± 0.287	0.452
	10	0.553 ± 0.298	0.526 ± 0.311	0.512 ± 0.287	0.489 ± 0.287	0.435 ± 0.266	0.517 ± 0.297	0.505
	20	0.578 ± 0.301	0.556 ± 0.316	0.543 ± 0.291	0.517 ± 0.303	0.470 ± 0.270	0.543 ± 0.302	0.535
Qwen3-14B	1	0.800 ± 0.172	0.782 ± 0.186	0.779 ± 0.188	0.777 ± 0.184	0.730 ± 0.179	0.785 ± 0.185	0.776
	5	0.846 ± 0.138	0.824 ± 0.157	0.826 ± 0.156	0.824 ± 0.160	0.780 ± 0.158	0.821 ± 0.160	0.820
	10	0.863 ± 0.116	0.847 ± 0.128	0.844 ± 0.132	0.844 ± 0.138	0.796 ± 0.140	0.836 ± 0.142	0.838
	20	0.880 ± 0.091	0.864 ± 0.105	0.855 ± 0.120	0.859 ± 0.118	0.814 ± 0.117	0.856 ± 0.117	0.855
Qwen3-4B	1	0.786 ± 0.164	0.757 ± 0.170	0.741 ± 0.157	0.748 ± 0.159	0.646 ± 0.167	0.749 ± 0.164	0.738
	5	0.848 ± 0.127	0.831 ± 0.142	0.816 ± 0.135	0.813 ± 0.140	0.735 ± 0.150	0.820 ± 0.135	0.811
	10	0.864 ± 0.114	0.852 ± 0.126	0.836 ± 0.116	0.835 ± 0.123	0.763 ± 0.133	0.838 ± 0.117	0.831
	20	0.876 ± 0.103	0.865 ± 0.110	0.848 ± 0.105	0.849 ± 0.106	0.780 ± 0.123	0.855 ± 0.096	0.846
gemma-3-12b	1	0.841 ± 0.147	0.835 ± 0.139	0.831 ± 0.134	0.818 ± 0.136	0.750 ± 0.130	0.834 ± 0.147	0.818
	5	0.862 ± 0.134	0.856 ± 0.130	0.855 ± 0.125	0.838 ± 0.131	0.782 ± 0.130	0.855 ± 0.132	0.841
	10	0.868 ± 0.126	0.860 ± 0.129	0.861 ± 0.118	0.844 ± 0.127	0.791 ± 0.127	0.862 ± 0.125	0.848
	20	<b>0.874 ± 0.119</b>	0.861 ± 0.129	<b>0.866 ± 0.115</b>	0.846 ± 0.130	0.798 ± 0.124	<b>0.866 ± 0.121</b>	0.852
gemma-3-4b	1	0.771 ± 0.161	0.745 ± 0.166	0.733 ± 0.171	0.725 ± 0.174	0.611 ± 0.148	0.731 ± 0.184	0.719
	5	0.822 ± 0.138	0.792 ± 0.144	0.787 ± 0.148	0.777 ± 0.149	0.675 ± 0.137	0.782 ± 0.151	0.773
	10	0.837 ± 0.125	0.805 ± 0.134	0.800 ± 0.140	0.786 ± 0.146	0.692 ± 0.135	0.800 ± 0.131	0.787
	20	0.845 ± 0.118	0.807 ± 0.134	0.810 ± 0.131	0.793 ± 0.146	0.697 ± 0.132	0.807 ± 0.131	0.793
<b>Average</b>	1	0.688 ± 0.243	0.672 ± 0.246	0.660 ± 0.245	0.652 ± 0.248	0.580 ± 0.161	0.669 ± 0.247	0.654
	5	0.760 ± 0.227	0.741 ± 0.235	0.730 ± 0.233	0.718 ± 0.236	0.654 ± 0.142	0.736 ± 0.233	0.723
	10	0.789 ± 0.210	0.769 ± 0.222	0.758 ± 0.219	0.747 ± 0.222	0.684 ± 0.126	0.763 ± 0.218	0.752
	20	0.805 ± 0.202	0.793 ± 0.213	0.773 ± 0.210	0.773 ± 0.219	0.703 ± 0.118	0.780 ± 0.209	0.771

Table 1: Averaged accuracy (mean ± std) across datasets for each model and agent count under different noise types. Larger models (e.g., Qwen3-14B, Gemma3-12B) achieve the highest overall accuracies, while smaller variants (Qwen3-4B, Gemma3-4B) show steeper gains with increasing agent count. Mistral-7B remains lowest across all settings, highlighting model-specific limitations. Bold values indicate the best accuracy per column.

Model	Agent Count	Clean	Punct-10	Punct-30	Punct-50	WikiTypo	R <sup>2</sup> ATA	Avg
<b>Llama-3.1</b>	1	0.518 ± 0.015	0.490 ± 0.016	0.469 ± 0.014	0.451 ± 0.013	0.441 ± 0.014	0.501 ± 0.015	0.478
	5	0.623 ± 0.011	0.588 ± 0.012	0.574 ± 0.008	0.553 ± 0.010	0.537 ± 0.010	0.603 ± 0.010	0.580
	10	0.662 ± 0.010	0.630 ± 0.011	0.620 ± 0.008	0.597 ± 0.007	0.575 ± 0.008	0.639 ± 0.009	0.621
	15	0.679 ± 0.004	0.650 ± 0.010	0.626 ± 0.005	0.611 ± 0.009	0.591 ± 0.004	0.656 ± 0.005	0.636
	20	0.683 ± 0.003	0.658 ± 0.007	0.635 ± 0.004	0.621 ± 0.004	0.603 ± 0.004	0.665 ± 0.004	0.644
	25	0.686 ± 0.000	0.663 ± 0.000	0.635 ± 0.000	0.624 ± 0.000	0.607 ± 0.000	0.672 ± 0.000	0.648
<b>Mistral-7B</b>	1	0.374 ± 0.013	0.369 ± 0.015	0.358 ± 0.012	0.350 ± 0.010	0.339 ± 0.013	0.370 ± 0.015	0.360
	5	0.421 ± 0.010	0.418 ± 0.012	0.405 ± 0.010	0.394 ± 0.009	0.380 ± 0.010	0.418 ± 0.009	0.406
	10	0.442 ± 0.006	0.431 ± 0.006	0.421 ± 0.008	0.416 ± 0.009	0.395 ± 0.008	0.437 ± 0.006	0.424
	15	0.448 ± 0.004	0.437 ± 0.005	0.430 ± 0.005	0.425 ± 0.007	0.402 ± 0.006	0.444 ± 0.006	0.431
	20	0.449 ± 0.006	0.444 ± 0.005	0.432 ± 0.005	0.420 ± 0.003	0.403 ± 0.003	0.448 ± 0.006	0.433
	25	0.455 ± 0.000	0.449 ± 0.000	0.437 ± 0.000	0.426 ± 0.000	0.411 ± 0.000	0.447 ± 0.000	0.437
<b>Qwen3-14B</b>	1	0.822 ± 0.007	0.814 ± 0.006	0.808 ± 0.006	0.804 ± 0.007	0.788 ± 0.009	0.807 ± 0.007	0.807
	5	0.836 ± 0.005	0.828 ± 0.005	0.822 ± 0.006	0.823 ± 0.004	0.803 ± 0.006	0.823 ± 0.004	0.822
	10	0.836 ± 0.003	0.830 ± 0.004	0.821 ± 0.004	0.824 ± 0.004	0.806 ± 0.004	0.824 ± 0.003	0.824
	15	0.833 ± 0.002	0.829 ± 0.003	0.822 ± 0.002	0.823 ± 0.003	0.806 ± 0.005	0.825 ± 0.004	0.823
	20	0.834 ± 0.003	0.829 ± 0.002	0.825 ± 0.002	0.823 ± 0.003	0.804 ± 0.002	0.826 ± 0.001	0.824
	25	0.832 ± 0.000	0.830 ± 0.000	0.830 ± 0.000	0.821 ± 0.000	0.805 ± 0.000	0.827 ± 0.000	0.824
<b>Qwen3-4B</b>	1	0.829 ± 0.006	0.817 ± 0.009	0.802 ± 0.006	0.797 ± 0.009	0.765 ± 0.007	0.815 ± 0.007	0.804
	5	0.850 ± 0.004	0.842 ± 0.006	0.824 ± 0.004	0.815 ± 0.004	0.792 ± 0.006	0.840 ± 0.005	0.827
	10	0.853 ± 0.003	0.845 ± 0.004	0.829 ± 0.005	0.819 ± 0.004	0.795 ± 0.005	0.841 ± 0.004	0.830
	15	0.856 ± 0.003	0.847 ± 0.002	0.829 ± 0.003	0.819 ± 0.002	0.795 ± 0.002	0.843 ± 0.002	0.832
	20	0.858 ± 0.002	0.845 ± 0.002	0.830 ± 0.003	0.821 ± 0.003	0.793 ± 0.002	0.845 ± 0.003	0.832
	25	0.858 ± 0.000	0.840 ± 0.000	0.831 ± 0.000	0.818 ± 0.000	0.795 ± 0.000	0.844 ± 0.000	0.831
<b>Gemma-3-12B</b>	1	0.850 ± 0.005	0.845 ± 0.008	0.843 ± 0.008	0.829 ± 0.006	0.793 ± 0.008	0.850 ± 0.006	0.835
	5	0.866 ± 0.004	0.864 ± 0.005	0.870 ± 0.004	0.851 ± 0.005	0.817 ± 0.005	0.868 ± 0.004	0.856
	10	0.870 ± 0.003	0.867 ± 0.004	0.873 ± 0.003	0.856 ± 0.003	0.823 ± 0.003	0.872 ± 0.003	0.860
	15	0.870 ± 0.001	0.868 ± 0.004	0.878 ± 0.002	0.860 ± 0.003	0.825 ± 0.002	0.874 ± 0.002	0.862
	20	0.870 ± 0.001	0.868 ± 0.002	0.878 ± 0.002	0.859 ± 0.001	0.828 ± 0.002	0.874 ± 0.002	0.863
	25	<b>0.871 ± 0.000</b>	<b>0.868 ± 0.000</b>	<b>0.879 ± 0.000</b>	<b>0.860 ± 0.000</b>	<b>0.826 ± 0.000</b>	<b>0.880 ± 0.000</b>	<b>0.864</b>
<b>Gemma-3-4B</b>	1	0.731 ± 0.008	0.723 ± 0.010	0.709 ± 0.011	0.707 ± 0.008	0.650 ± 0.009	0.713 ± 0.009	0.706
	5	0.767 ± 0.006	0.754 ± 0.006	0.748 ± 0.007	0.743 ± 0.008	0.686 ± 0.006	0.745 ± 0.006	0.740
	10	0.774 ± 0.004	0.765 ± 0.004	0.757 ± 0.005	0.751 ± 0.007	0.695 ± 0.005	0.753 ± 0.005	0.749
	15	0.779 ± 0.003	0.769 ± 0.003	0.760 ± 0.003	0.756 ± 0.004	0.699 ± 0.004	0.757 ± 0.004	0.753
	20	0.781 ± 0.001	0.767 ± 0.003	0.763 ± 0.002	0.758 ± 0.003	0.699 ± 0.002	0.760 ± 0.002	0.755
	25	0.775 ± 0.000	0.771 ± 0.000	0.765 ± 0.000	0.759 ± 0.000	0.701 ± 0.000	0.762 ± 0.000	0.755

Table 2: Aggregated accuracy (mean ± std) on the **MMLU** benchmark for each model and agent count under different noise conditions. Only the global best values per column are bolded. Larger models (e.g., Qwen3-14B, Gemma-3-12B) achieve the highest accuracies, while smaller ones (e.g., Qwen3-4B, Gemma-3-4B) show steeper gains with increasing collaboration. Mistral-7B performs lowest across all settings.

Model	Agent Count	Clean	Punct-10	Punct-30	Punct-50	WikiTypo	R <sup>2</sup> ATA	Avg
<b>Llama-3.1</b>	1	0.233 ± 0.029	0.215 ± 0.027	0.211 ± 0.028	0.203 ± 0.032	0.185 ± 0.036	0.237 ± 0.024	0.214
	5	0.370 ± 0.027	0.330 ± 0.021	0.311 ± 0.025	0.308 ± 0.030	0.287 ± 0.029	0.348 ± 0.022	0.326
	10	0.466 ± 0.016	0.416 ± 0.017	0.367 ± 0.021	0.386 ± 0.020	0.348 ± 0.023	0.427 ± 0.021	0.402
	15	0.506 ± 0.019	0.435 ± 0.011	0.393 ± 0.015	0.407 ± 0.017	0.370 ± 0.019	0.457 ± 0.019	0.428
	20	0.508 ± 0.020	0.453 ± 0.011	0.410 ± 0.010	0.410 ± 0.016	0.385 ± 0.014	0.473 ± 0.014	0.440
25	0.540 ± 0.000	0.480 ± 0.000	0.440 ± 0.000	0.410 ± 0.000	0.370 ± 0.000	0.490 ± 0.000	0.455	
<b>Mistral-7B</b>	1	0.068 ± 0.020	0.057 ± 0.019	0.058 ± 0.019	0.049 ± 0.023	0.042 ± 0.017	0.055 ± 0.016	0.055
	5	0.100 ± 0.022	0.078 ± 0.020	0.084 ± 0.018	0.069 ± 0.026	0.066 ± 0.023	0.073 ± 0.016	0.078
	10	0.148 ± 0.017	0.099 ± 0.022	0.125 ± 0.015	0.101 ± 0.027	0.089 ± 0.019	0.106 ± 0.017	0.111
	15	0.156 ± 0.009	0.123 ± 0.011	0.143 ± 0.015	0.112 ± 0.022	0.105 ± 0.011	0.115 ± 0.020	0.126
	20	0.168 ± 0.013	0.120 ± 0.008	0.152 ± 0.004	0.103 ± 0.014	0.123 ± 0.009	0.122 ± 0.009	0.131
25	0.180 ± 0.000	0.130 ± 0.000	0.140 ± 0.000	0.100 ± 0.000	0.120 ± 0.000	0.130 ± 0.000	0.133	
<b>Qwen3-14B</b>	1	0.522 ± 0.032	0.476 ± 0.027	0.472 ± 0.029	0.477 ± 0.031	0.444 ± 0.025	0.486 ± 0.021	0.479
	5	0.629 ± 0.026	0.570 ± 0.021	0.577 ± 0.023	0.567 ± 0.021	0.534 ± 0.018	0.565 ± 0.018	0.574
	10	0.689 ± 0.016	0.646 ± 0.017	0.640 ± 0.014	0.631 ± 0.018	0.582 ± 0.011	0.613 ± 0.020	0.633
	15	0.728 ± 0.011	0.685 ± 0.028	0.661 ± 0.012	0.666 ± 0.009	0.619 ± 0.011	0.655 ± 0.013	0.669
	20	0.757 ± 0.011	0.710 ± 0.013	0.675 ± 0.016	0.685 ± 0.013	0.645 ± 0.005	0.682 ± 0.012	0.692
25	<b>0.770 ± 0.000</b>	<b>0.710 ± 0.000</b>	<b>0.690 ± 0.000</b>	<b>0.700 ± 0.000</b>	<b>0.660 ± 0.000</b>	<b>0.700 ± 0.000</b>	<b>0.705</b>	
<b>Qwen3-4B</b>	1	0.524 ± 0.035	0.481 ± 0.034	0.484 ± 0.038	0.489 ± 0.027	0.410 ± 0.037	0.488 ± 0.025	0.479
	5	0.649 ± 0.017	0.607 ± 0.027	0.602 ± 0.033	0.592 ± 0.016	0.535 ± 0.023	0.610 ± 0.023	0.599
	10	0.688 ± 0.012	0.657 ± 0.023	0.656 ± 0.018	0.645 ± 0.012	0.597 ± 0.013	0.657 ± 0.013	0.650
	15	0.714 ± 0.009	0.680 ± 0.013	0.676 ± 0.015	0.658 ± 0.018	0.625 ± 0.007	0.698 ± 0.022	0.675
	20	0.722 ± 0.011	0.702 ± 0.020	0.688 ± 0.011	0.692 ± 0.011	0.638 ± 0.004	0.715 ± 0.014	0.693
25	0.730 ± 0.000	<b>0.740 ± 0.000</b>	0.700 ± 0.000	0.690 ± 0.000	0.650 ± 0.000	<b>0.740 ± 0.000</b>	0.708	
<b>Gemma-3-12B</b>	1	0.604 ± 0.031	0.609 ± 0.024	0.612 ± 0.028	0.595 ± 0.027	0.553 ± 0.033	0.597 ± 0.033	0.595
	5	0.645 ± 0.024	0.644 ± 0.017	0.650 ± 0.025	0.622 ± 0.020	0.582 ± 0.022	0.641 ± 0.030	0.631
	10	0.665 ± 0.017	0.651 ± 0.019	0.668 ± 0.016	0.635 ± 0.011	0.594 ± 0.010	0.659 ± 0.014	0.645
	15	0.667 ± 0.014	0.656 ± 0.014	0.676 ± 0.010	0.635 ± 0.015	0.603 ± 0.009	0.664 ± 0.019	0.650
	20	0.683 ± 0.016	0.650 ± 0.010	0.678 ± 0.007	0.633 ± 0.016	0.608 ± 0.007	0.670 ± 0.016	0.654
25	<b>0.680 ± 0.000</b>	0.660 ± 0.000	<b>0.680 ± 0.000</b>	<b>0.650 ± 0.000</b>	<b>0.600 ± 0.000</b>	0.660 ± 0.000	<b>0.655</b>	
<b>Gemma-3-4B</b>	1	0.534 ± 0.033	0.488 ± 0.026	0.473 ± 0.030	0.458 ± 0.026	0.400 ± 0.037	0.447 ± 0.038	0.467
	5	0.629 ± 0.026	0.575 ± 0.018	0.568 ± 0.028	0.554 ± 0.021	0.488 ± 0.033	0.556 ± 0.024	0.562
	10	0.670 ± 0.013	0.606 ± 0.018	0.594 ± 0.016	0.567 ± 0.017	0.509 ± 0.018	0.611 ± 0.017	0.593
	15	0.682 ± 0.009	0.616 ± 0.012	0.605 ± 0.011	0.585 ± 0.019	0.510 ± 0.012	0.624 ± 0.014	0.604
	20	0.692 ± 0.007	0.612 ± 0.007	0.622 ± 0.012	0.583 ± 0.007	0.520 ± 0.014	0.623 ± 0.009	0.609
25	0.680 ± 0.000	0.620 ± 0.000	0.630 ± 0.000	0.590 ± 0.000	0.530 ± 0.000	0.620 ± 0.000	0.612	

Table 3: Aggregated accuracy (mean ± std) on the **Math** benchmark for each model and agent count under different noise conditions. Only the single highest value per column is bolded. Overall, Qwen3-14B achieves the top accuracy in most categories, followed closely by Gemma-3-12B.

Model	Agent Count	Clean	Punct-10	Punct-30	Punct-50	WikiTypo	R <sup>2</sup> ATA	Avg
<b>Llama-3.1</b>	1	0.632 ± 0.011	0.637 ± 0.011	0.624 ± 0.010	0.615 ± 0.010	0.466 ± 0.012	0.608 ± 0.013	0.597
	5	0.799 ± 0.018	0.788 ± 0.013	0.763 ± 0.013	0.763 ± 0.009	0.588 ± 0.016	0.766 ± 0.020	0.745
	10	0.886 ± 0.004	0.869 ± 0.005	0.846 ± 0.008	0.839 ± 0.006	0.668 ± 0.007	0.843 ± 0.007	0.825
	15	0.905 ± 0.003	0.886 ± 0.002	0.866 ± 0.005	0.858 ± 0.003	0.695 ± 0.005	0.865 ± 0.004	0.846
	20	0.911 ± 0.001	0.891 ± 0.002	0.874 ± 0.002	0.869 ± 0.003	0.708 ± 0.003	0.877 ± 0.003	0.855
	25	0.913 ± 0.000	0.901 ± 0.000	0.877 ± 0.000	0.871 ± 0.000	0.713 ± 0.000	0.877 ± 0.000	0.859
<b>Mistral-7B</b>	1	0.428 ± 0.010	0.397 ± 0.009	0.368 ± 0.011	0.338 ± 0.015	0.264 ± 0.010	0.385 ± 0.013	0.363
	5	0.553 ± 0.019	0.512 ± 0.021	0.484 ± 0.024	0.433 ± 0.018	0.341 ± 0.013	0.490 ± 0.019	0.469
	10	0.659 ± 0.009	0.614 ± 0.008	0.582 ± 0.013	0.538 ± 0.005	0.420 ± 0.007	0.594 ± 0.008	0.568
	15	0.692 ± 0.004	0.657 ± 0.003	0.621 ± 0.008	0.581 ± 0.005	0.454 ± 0.004	0.632 ± 0.006	0.606
	20	0.718 ± 0.002	0.678 ± 0.007	0.641 ± 0.003	0.607 ± 0.003	0.476 ± 0.006	0.652 ± 0.004	0.629
	25	0.732 ± 0.000	0.690 ± 0.000	0.656 ± 0.000	0.618 ± 0.000	0.480 ± 0.000	0.657 ± 0.000	0.639
<b>Qwen3-14B</b>	1	0.875 ± 0.005	0.868 ± 0.006	0.861 ± 0.006	0.862 ± 0.005	0.750 ± 0.009	0.863 ± 0.006	0.846
	5	0.921 ± 0.003	0.917 ± 0.003	0.915 ± 0.004	0.918 ± 0.004	0.808 ± 0.006	0.906 ± 0.004	0.898
	10	0.931 ± 0.003	0.928 ± 0.003	0.924 ± 0.002	0.929 ± 0.002	0.824 ± 0.003	0.918 ± 0.002	0.909
	15	0.933 ± 0.002	0.931 ± 0.001	0.926 ± 0.002	0.932 ± 0.002	0.830 ± 0.002	0.920 ± 0.001	0.912
	20	0.934 ± 0.001	0.932 ± 0.001	0.927 ± 0.001	0.933 ± 0.002	0.833 ± 0.003	0.921 ± 0.002	0.913
	25	0.934 ± 0.000	<b>0.934 ± 0.000</b>	0.929 ± 0.000	<b>0.934 ± 0.000</b>	<b>0.837 ± 0.000</b>	0.924 ± 0.000	<b>0.915</b>
<b>Qwen3-4B</b>	1	0.819 ± 0.007	0.790 ± 0.009	0.777 ± 0.009	0.788 ± 0.008	0.573 ± 0.010	0.761 ± 0.007	0.751
	5	0.898 ± 0.006	0.880 ± 0.006	0.869 ± 0.007	0.872 ± 0.007	0.670 ± 0.010	0.846 ± 0.010	0.839
	10	0.916 ± 0.002	0.905 ± 0.005	0.890 ± 0.004	0.899 ± 0.004	0.702 ± 0.004	0.868 ± 0.006	0.863
	15	0.922 ± 0.002	0.913 ± 0.003	0.895 ± 0.003	0.906 ± 0.002	0.712 ± 0.005	0.873 ± 0.005	0.870
	20	0.925 ± 0.002	0.914 ± 0.001	0.900 ± 0.002	0.910 ± 0.002	0.717 ± 0.003	0.878 ± 0.002	0.874
	25	0.930 ± 0.000	0.916 ± 0.000	0.899 ± 0.000	0.914 ± 0.000	0.722 ± 0.000	0.882 ± 0.000	0.877
<b>Gemma-3-12B</b>	1	0.911 ± 0.004	0.907 ± 0.005	0.900 ± 0.005	0.897 ± 0.005	0.740 ± 0.007	0.898 ± 0.004	0.875
	5	0.935 ± 0.002	0.930 ± 0.002	0.921 ± 0.004	0.918 ± 0.004	0.783 ± 0.006	0.919 ± 0.004	0.901
	10	0.938 ± 0.002	0.933 ± 0.002	0.924 ± 0.002	0.921 ± 0.002	0.796 ± 0.004	0.923 ± 0.002	0.906
	15	0.941 ± 0.002	0.933 ± 0.001	0.925 ± 0.003	0.923 ± 0.002	0.801 ± 0.004	0.924 ± 0.002	0.908
	20	0.941 ± 0.001	0.935 ± 0.001	0.927 ± 0.001	0.922 ± 0.001	0.802 ± 0.001	0.924 ± 0.001	0.909
	25	<b>0.942 ± 0.000</b>	0.933 ± 0.000	<b>0.928 ± 0.000</b>	0.923 ± 0.000	0.802 ± 0.000	<b>0.924 ± 0.000</b>	<b>0.909</b>
<b>Gemma-3-4B</b>	1	0.849 ± 0.006	0.840 ± 0.007	0.817 ± 0.009	0.803 ± 0.006	0.582 ± 0.007	0.823 ± 0.008	0.786
	5	0.898 ± 0.004	0.889 ± 0.005	0.874 ± 0.005	0.857 ± 0.006	0.651 ± 0.005	0.873 ± 0.004	0.840
	10	0.906 ± 0.002	0.896 ± 0.003	0.882 ± 0.003	0.869 ± 0.004	0.672 ± 0.006	0.883 ± 0.003	0.852
	15	0.908 ± 0.001	0.897 ± 0.001	0.887 ± 0.002	0.871 ± 0.003	0.676 ± 0.003	0.886 ± 0.003	0.854
	20	0.909 ± 0.001	0.899 ± 0.001	0.888 ± 0.001	0.873 ± 0.002	0.676 ± 0.002	0.888 ± 0.001	0.855
	25	0.909 ± 0.000	0.898 ± 0.000	0.888 ± 0.000	0.876 ± 0.000	0.674 ± 0.000	0.892 ± 0.000	0.856

Table 4: Aggregated accuracy (mean ± std) on the **GSM8K** benchmark for each model and agent count under different noise conditions. Only the single best value per column is bolded. Gemma-3-12B achieves the highest performance overall, particularly under noisy settings.

Model	Agent Count	Clean	Punct-10	Punct-30	Punct-50	WikiTypo	R <sup>2</sup> ATA	Avg
<b>Llama-3.1</b>	1	0.747 ± 0.029	0.787 ± 0.034	0.758 ± 0.024	0.753 ± 0.027	0.683 ± 0.038	0.812 ± 0.018	0.757
	5	0.959 ± 0.019	0.957 ± 0.015	0.917 ± 0.013	0.893 ± 0.018	0.866 ± 0.023	0.968 ± 0.012	0.927
	10	0.995 ± 0.003	0.986 ± 0.003	0.949 ± 0.007	0.926 ± 0.010	0.914 ± 0.011	0.987 ± 0.008	0.960
	15	0.997 ± 0.003	0.987 ± 0.002	0.954 ± 0.004	0.933 ± 0.004	0.929 ± 0.010	<b>0.994 ± 0.004</b>	0.966
	20	0.996 ± 0.003	0.985 ± 0.003	0.955 ± 0.004	0.937 ± 0.003	0.932 ± 0.006	<b>0.994 ± 0.000</b>	0.967
	25	<b>1.000 ± 0.000</b>	0.989 ± 0.000	0.961 ± 0.000	0.933 ± 0.000	0.939 ± 0.000	0.989 ± 0.000	0.969
<b>Mistral-7B</b>	1	0.709 ± 0.026	0.712 ± 0.029	0.657 ± 0.030	0.620 ± 0.034	0.540 ± 0.038	0.696 ± 0.030	0.656
	5	0.913 ± 0.012	0.911 ± 0.013	0.853 ± 0.020	0.830 ± 0.020	0.738 ± 0.033	0.879 ± 0.019	0.854
	10	0.962 ± 0.008	0.959 ± 0.015	0.919 ± 0.011	0.902 ± 0.012	0.836 ± 0.015	0.930 ± 0.007	0.918
	15	0.966 ± 0.011	0.975 ± 0.010	0.939 ± 0.007	0.926 ± 0.010	0.868 ± 0.014	0.943 ± 0.008	0.936
	20	0.976 ± 0.007	0.981 ± 0.003	0.947 ± 0.007	0.939 ± 0.003	0.879 ± 0.007	0.951 ± 0.005	0.946
	25	0.989 ± 0.000	0.983 ± 0.000	0.950 ± 0.000	0.939 ± 0.000	0.878 ± 0.000	0.956 ± 0.000	0.949
<b>Qwen3-14B</b>	1	0.982 ± 0.007	0.969 ± 0.011	0.974 ± 0.011	0.965 ± 0.011	0.936 ± 0.017	0.986 ± 0.005	0.969
	5	0.996 ± 0.002	0.983 ± 0.004	0.990 ± 0.005	0.989 ± 0.007	0.974 ± 0.008	0.990 ± 0.003	0.987
	10	0.996 ± 0.002	0.984 ± 0.002	0.990 ± 0.005	<b>0.994 ± 0.004</b>	0.973 ± 0.005	0.991 ± 0.003	0.988
	15	0.995 ± 0.002	0.983 ± 0.000	0.992 ± 0.004	0.993 ± 0.003	0.973 ± 0.003	0.991 ± 0.003	0.988
	20	0.994 ± 0.000	0.983 ± 0.000	<b>0.994 ± 0.002</b>	<b>0.994 ± 0.000</b>	0.973 ± 0.004	<b>0.994 ± 0.002</b>	<b>0.989</b>
	25	0.994 ± 0.000	0.983 ± 0.000	<b>0.994 ± 0.000</b>	<b>0.994 ± 0.000</b>	<b>0.978 ± 0.000</b>	<b>0.994 ± 0.000</b>	<b>0.989</b>
<b>Qwen3-4B</b>	1	0.972 ± 0.009	0.940 ± 0.019	0.902 ± 0.021	0.916 ± 0.017	0.836 ± 0.027	0.931 ± 0.017	0.916
	5	0.996 ± 0.003	0.997 ± 0.004	0.967 ± 0.007	0.972 ± 0.006	0.941 ± 0.013	0.984 ± 0.005	0.976
	10	0.997 ± 0.003	<b>1.000 ± 0.001</b>	0.970 ± 0.006	0.977 ± 0.006	0.959 ± 0.005	0.984 ± 0.002	0.981
	15	0.999 ± 0.002	<b>1.000 ± 0.000</b>	0.974 ± 0.003	0.973 ± 0.002	0.966 ± 0.004	0.984 ± 0.002	0.983
	20	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>	0.974 ± 0.003	0.973 ± 0.002	0.971 ± 0.004	0.983 ± 0.000	0.984
	25	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>	0.978 ± 0.000	0.972 ± 0.000	<b>0.978 ± 0.000</b>	0.983 ± 0.000	0.985
<b>Gemma-3-12B</b>	1	0.998 ± 0.003	0.979 ± 0.009	0.967 ± 0.009	0.950 ± 0.012	0.913 ± 0.012	0.991 ± 0.004	0.966
	5	<b>1.000 ± 0.000</b>	0.986 ± 0.006	0.978 ± 0.004	0.960 ± 0.009	0.945 ± 0.007	0.993 ± 0.003	0.977
	10	<b>1.000 ± 0.000</b>	0.990 ± 0.003	0.980 ± 0.003	0.965 ± 0.008	0.949 ± 0.004	0.993 ± 0.002	0.979
	15	<b>1.000 ± 0.000</b>	0.991 ± 0.004	0.983 ± 0.002	0.970 ± 0.005	0.954 ± 0.003	<b>0.994 ± 0.000</b>	0.982
	20	<b>1.000 ± 0.000</b>	0.990 ± 0.004	0.981 ± 0.003	0.971 ± 0.004	0.954 ± 0.003	<b>0.994 ± 0.000</b>	0.982
	25	<b>1.000 ± 0.000</b>	0.994 ± 0.000	0.983 ± 0.000	0.967 ± 0.000	0.956 ± 0.000	<b>0.994 ± 0.000</b>	0.982
<b>Gemma-3-4B</b>	1	0.969 ± 0.010	0.929 ± 0.013	0.934 ± 0.012	0.930 ± 0.012	0.813 ± 0.021	0.942 ± 0.011	0.920
	5	0.993 ± 0.005	0.951 ± 0.008	0.960 ± 0.007	0.953 ± 0.005	0.873 ± 0.012	0.954 ± 0.006	0.947
	10	0.997 ± 0.003	0.952 ± 0.004	0.966 ± 0.007	0.955 ± 0.001	0.890 ± 0.008	0.952 ± 0.006	0.952
	15	0.998 ± 0.002	0.952 ± 0.002	0.967 ± 0.008	0.956 ± 0.002	0.893 ± 0.006	0.954 ± 0.004	0.953
	20	0.999 ± 0.002	0.950 ± 0.000	0.966 ± 0.004	0.956 ± 0.000	0.892 ± 0.004	0.955 ± 0.004	0.953
	25	<b>1.000 ± 0.000</b>	0.950 ± 0.000	0.972 ± 0.000	0.956 ± 0.000	0.889 ± 0.000	0.956 ± 0.000	0.954

Table 5: Aggregated accuracy (mean ± std) on the **MultiArith** benchmark for each model and agent count under different noise conditions. Only the single highest value per column is bolded. Qwen3-14B achieves the top accuracy in the WikiTypo and average columns, while Gemma-3-12B leads in the Clean column.