Overview of Programming Project: Replication and Extension of SAC3

Introduction:

Detecting hallucinations in large language models (LLMs) is crucial for improving their trustworthiness and reliability, especially in applications requiring factual accuracy such as question-answering. The SAC3 method (i.e., semantic-aware cross-check consistency), proposed by Zhang et al. (2023), introduces a novel approach that expands on the self-consistency checking by integrating semantically equivalent question perturbation and cross-model response consistency checking.

This project replicated the SAC3 method as specified in the original paper and extended it to a broader range of models with varying parameter sizes, assessing its scalability and robustness.

Key Work

1. **Replication**

I began by replicating SAC3 on binary classification tasks (prime number and senator search) and open-domain QA tasks (HotpotQA-halu), using GPT-3.5-turbo, with AUROC and accuracy metrics for evaluation. The results closely matched those reported in the original paper, confirming SAC3's reliability in detecting hallucinations. Testing with GPT-4.0 yielded similar results, further validating the method.

SAC3 consistently outperformed self-check consistency (SC2) across various settings, including different numbers of self-responses (3, 5, 10, and 15) and semantically perturbed questions (5 and 10). While performance improved with larger sample sizes, returns were diminishing beyond certain thresholds, consistent with theoretical expectations.

Method	AUROC Score (SC2)
Reported SC2 (self-checking)	74.2
SC2 for 3 self-responses	74.4
SC2 for 5 self-responses	75.36
SC2 for 10 self-responses	74.08
SC2 for 15 self-responses	78.24

Table 1: Self-Checking AUROC Score Comparison illustrates the scores obtained when varying the number of self-responses in the self-checking setup.

Method	AUROC Score (SC2)
Reported SAC3 (cross-checking) for 10 perturbed questions	81.3
SAC3 for 5 perturbed questions	81.5
SAC3 for 10 perturbed questions	86.96

Table 2: Cross-Checking AUROC Score Comparison shows the outcomes when varying the number of perturbed questions in the cross-checking setup.

2. **Extension**

The SAC3 method was extended to open-source LLMs with fewer than 10B parameters, including Llama-3-8B, Gemma-7B, Mistral-7B, Phi-3-mini, and Qwen1.5-7B, tested with the HotpotQA-halu dataset. Initially, testing was limited to 10 data points due to memory constraints, and custom prompt and code modifications were made to improve the integration of perturbed questions in cross-checking. Across all models, SAC3 consistently outperformed SC2, achieving higher AUROC scores.

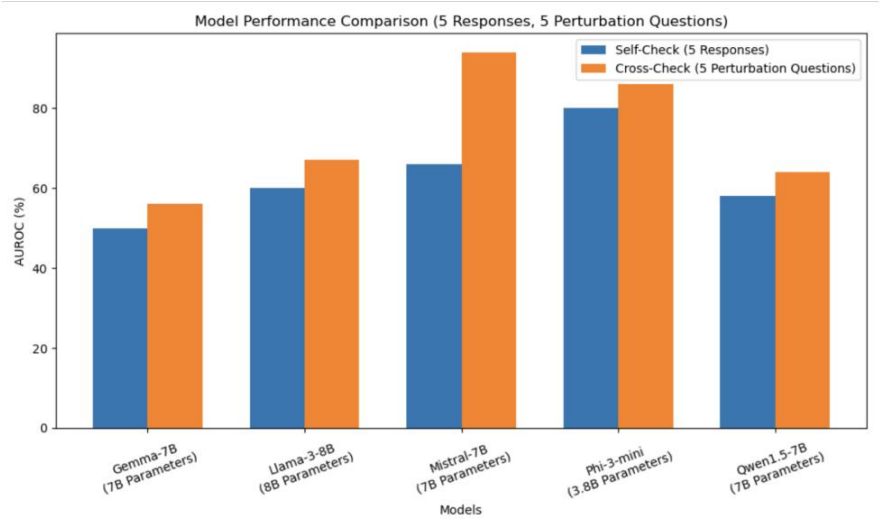

Chart 1: Comparing AUROC scores for 5 responses (self-checking) vs. 5 perturbed questions (cross-checking)

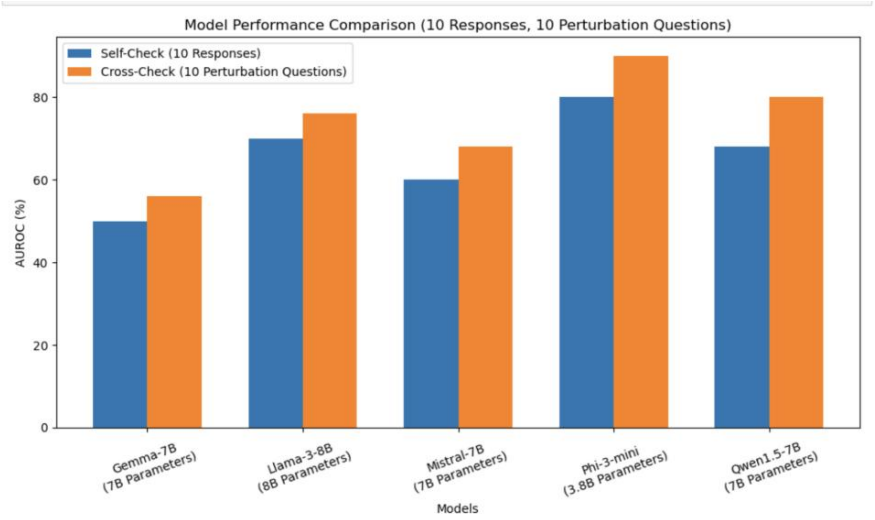

Chart 2: Comparing AUROC scores for 10 responses (self-checking) vs. 10 perturbed questions (cross-checking)

To obtain more reliable results, I implemented bootstrap sampling, testing each model with random samples of 10 questions repeated five times to aggregate 50 data points. The cross-checking method consistently outperformed self-checking across all tested models, demonstrating SAC3's robustness and scalability.

Model	Self-Checking (SC2)	Standard Deviation	Cross-Checking (SAC3)	Standard Deviation
Mistral-7B-Instruct-v0.2	55.20	0.1107	60.80	0.1468
Phi-3-mini-128k-instruct	46.00	0.1625	78.00	0.1513
Gemma-7B-it	62.00	0.0748	69.20	0.0943
Meta-Llama-3-8B-Instruct	64.00	0.1824	70.00	0.1397
Qwen1.5-7B-Chat	61.20	0.1603	65.20	0.1542

Table 3: Comparing AUROC scores for 5 responses (self-checking) vs. 5 perturbed questions (cross-checking) with Bootstrap

Model	Self-Checking (SC2)	Standard Deviation	Cross-Checking (SAC3)	Standard Deviation
Mistral-7B-Instruct-v0.2	62.80	0.2160	70.80	0.1129
Phi-3-mini-128k-instruct	58.00	0.0400	68.80	0.1024
Gemma-7B-it	46.00	0.1625	58.80	0.1001
Meta-Llama-3-8B-Instruct	59.20	0.1281	61.00	0.1130
Qwen1.5-7B-Chat	50.00	0.2067	58.40	0.1015

Table 4: Comparing AUROC scores for 10 responses (self-checking) vs. 10 perturbed questions (cross-checking) with Bootstrap

Conclusion

The SAC3 method has shown strong potentials in detecting hallucinations across a variety of LLMs, confirming its robustness. Cross-checking consistently outperforms self-checking, proving SAC3's adaptability to different models and conditions. However, challenges such as performance variability and hardware limitations indicate the need for further testing with larger datasets and more optimized computational strategies.

Future work should focus on further validating SAC3 with more extensive datasets to ensure consistent performance across a wider range of models and improving computational efficiency.