Adaptive Stress Testing of VLM Safety Monitors

Aleicia Zhu[1], Rory Lipkis[2], and Adrian Agogino[2]

[1] Harvey Mudd College, Claremont, CA
alezhu@hmc.edu
[2] NASA Ames Research Center, Moffett Field, CA
{rory,adrian.k.agogino}@nasa.gov

Abstract. Vision-language models (VLMs) have been proposed as high-level monitors in safety-critical vehicle applications, where their chain-of-thought (CoT) reasoning could provide a layer of redundancy in unexpected and complex scenarios. However, these models are prone to reliability issues, which manifest in sparse, unpredictable failure modes that traditional testing struggles to discover. We present an adaptive stress testing framework to validate multimodal language models against certain correctness requirements. Our approach uses Monte Carlo tree search (MCTS) to systematically and efficiently perturb the CoT process toward violations of a requirement. A margin-based reward function quantifies proximity to a decision boundary and guides the search toward failure modes within the nominal operating domain, ensuring that identified vulnerabilities are relevant. We demonstrate our approach on an 11B-parameter VLM in a simulated aircraft collision monitoring task, revealing specific vulnerabilities in its reasoning that would be obscured by random testing. This work provides a practical validation approach for multimodal language models in safety-critical domains.

1 Introduction

Recent research has explored the application of vision-language models (VLMs) as high-level controllers or monitors in safety-critical domains [15, 14, 16, 4, 3]. These models often operate within agentic frameworks which constrain them to a set of approved behaviors without fully specifying an internal decision process. In this setting, they can offer impressive reasoning capabilities with a degree of human-like "common sense" [12]. Their large-scale transformer architectures allow them to process information using ad-hoc intermediate representations. This flexibility enables them to handle corner cases and heterogeneous scenarios where traditional, component-based architectures might come apart at the seams [2, 10].

VLMs are often invoked with elaborate prompts and chain-of-thought (CoT) enhancements designed to trigger explicit intermediate reasoning steps. While these techniques have improved performance across a variety of tasks [13], VLMs still exhibit sparse and erratic failure modes that do not surface in limited testing. This has led to a development paradigm where errors are addressed reactively in production, typically through prompt engineering or targeted fine-tuning.

This sort of experimentalism is inadequate for safety-critical applications, which require a higher level of assurance.

Adaptive stress testing (AST) provides a partial solution to the challenge of validating performance. AST efficiently drives systems to their nearest failure modes, enabling systematic analysis and error detection earlier in the design stage [7, 8]. Applied to VLMs, this method can identify the likeliest missteps in the stochastic CoT process that lead to a violation, revealing sensitivities that might otherwise be missed by a fixed testing regime.

2 Adaptive stress testing

2.1 Requirement scope

We restrict our focus to a subset of correctness requirements whose satisfaction or violation is determined by a single logit in the model's output. The target logit is not required to have a fixed position in the output but must be consistently extractable (for instance, through a schema specified in the prompt). Under this constraint, our approach is applicable to most vision-language tasks involving binary and multiple-choice questions with known answers. These restrictions are broadly consistent with other automated testing schemes.

2.2 Autoregression as a decision process

AST is typically applied to systems that make decisions autonomously over time in a stochastic environment; this allows adversarial reinforcement learning to optimize against the evolution of the system. Here, although we are examining a VLM's ability to perform an assessment at a single moment in time, its autoregressive mechanism sequentially queries itself to produce a response (Fig. 1).

Because VLMs are often used with a nonzero sampling temperature to increase output diversity, the system can be considered stochastic. Even when a model is invoked deterministically, as it might be for a critical task, our approach remains relevant as a form of sensitivity analysis, since the characteristics of a sampled model can be formally related to those of a deterministic model under uncertain input [5].

We formulate adversarial stress testing of VLMs as a Markov decision process (MDP). Given a fixed system prompt and an input image with a known ground-truth label, our objective is to identify the likeliest vulnerabilities, i.e., the highest-likelihood response sequences that violate the correctness requirement. To guide the learning process toward meaningful failures, a reward function captures the confidence margin in the model's response, as measured by the difference between the highest-scoring logits.

2.3 Monte Carlo tree search

We apply Monte Carlo tree search (MCTS) to systematically explore the space of possible model responses. MCTS builds a search tree that maintains statistics

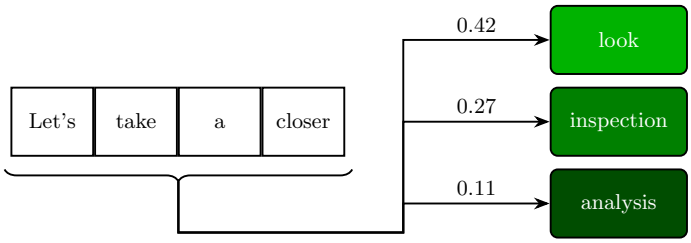

Fig. 1: Language models perform autoregressive inference to predict the next token in the sequence. The model's propensity to select non-modal tokens is controlled by the sampling temperature. AST tries to find the smallest deviation from the mode that results in an eventual correctness violation.

on promising response paths, allowing balanced exploration of the output space [6]. The model generates responses token-by-token, where each token t_i is sampled from an output distribution conditioned on all previous tokens. A complete response trajectory is defined as $\tau = (t_1, t_2, \ldots, t_n)$, where n is determined either by reaching a terminal token or the maximum token limit.

The search tree $\mathcal{T}$ is rooted at the initial state s_0. Each node $v \in \mathcal{T}$ represents a partial response sequence; it maintains a cache of child nodes sampled from the top k most likely next tokens and a set of statistics about the rewards accrued by simulations passing through v (Fig. 2). To manage the large branching factor in language generation ($|\mathcal{V}| > 10^5$ for LLaMA-3), we employ progressive widening; this allows the tree to gradually expand its branching factor as nodes accumulate visits, focusing on promising regions while maintaining tractability [1].

We constrain model output using standard prompt techniques, requesting a CoT reasoning sequence and a parsable final answer; this allows computation of the adversarial reward function. Given a complete response τ, we provide a reward bonus β for an incorrect response. Additionally, we include the logit margin $m(\tau)$, the difference between the highest logit value and the logit value corresponding to the correct answer (or, when these are the same, the difference between the two highest logit values). The margin encourages CoT executions generated with higher confidence, leading to the discovery of failure modes that are closer to nominal operation.

3 Case study: aircraft collision monitoring

We tested our framework with a medium-sized VLM [9] in the role of an aircraft collision monitor, using images from the AVOIDDS detect-and-avoid dataset [11]. The correctness requirement states that the model must determine whether an aircraft is present in an image. Although our emphasis is on false negatives because of their higher impact, the framework is equally capable of detecting false positives.

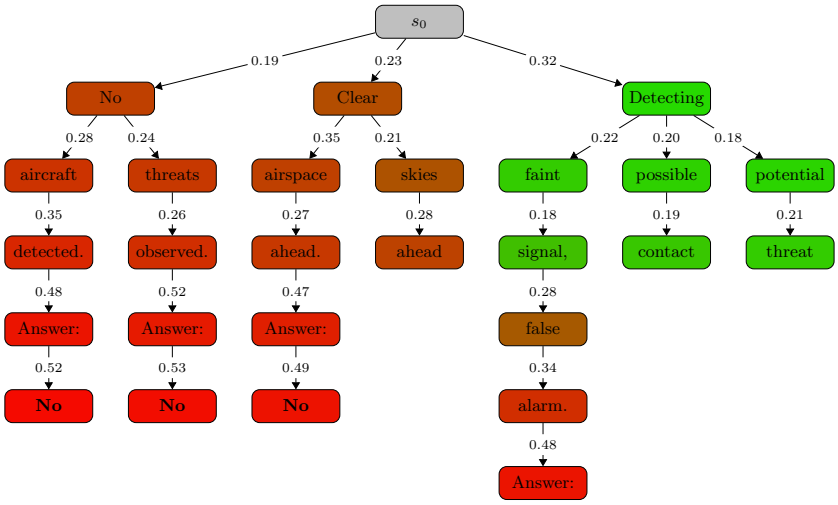

Fig. 2: Basic example of the MCTS tree mid-generation; edges are labeled with transition probabilities while nodes are colored by their reward statistics.

Fig. 3a presents a representative failure case found through AST. Despite acknowledging an apparent aircraft, the model constructs an elaborate justification for why the vehicle cannot actually be present:

> Upon examining this aerial image, it becomes apparent that there is no aircraft [...] Perspective may make small objects on the near-horizon line seem bigger in real life when an aircraft can't fit in such a position. This indicates the image likely contains no physical objects such as an aircraft, even if an impression of one exists [...] Answer: **No**

Fig. 3b demonstrates a failure pattern involving undesirable meta-awareness. As the CoT progresses, the model comes to determine that the image depicts a simulated aircraft and concludes that there is no threat to safety:

> The sunlight coming in at an extreme angle over a vast landscape implies a considerable altitude [...] it's most likely that the picture depicts a plane that's been modeled for simulation [...] with sufficient clarity to be definitively determined in all probability it's not an image of actual aircraft [...] Answer: **No**

This example highlights both the strengths and complications of VLM intelligence. The model clearly understands the context of the input beyond the assumed confines of the agentic role. While this level of insight could be critical to the development of a robust and capable system, it complicates validation, which should occur without the explicit awareness of the system under test.

(a) (b)

Fig. 3: Images from the simulated dataset. The agent's role is to monitor the presence of other aircraft. Adaptive stress testing finds the likeliest chain-of-thought executions that induce an incorrect response. In Fig. 3a, it concludes that no aircraft is present due to spurious reasoning about geometry. In Fig. 3b, it "deduces" that the aircraft is simulated and poses no actual threat.

4 Discussion

4.1 Failure mode characterization

AST strongly optimizes for likelihood according to the generative model under test. As a result, this approach avoids linguistic errors, which are too improbable for an advanced language model, shifting the focus toward subtler reasoning errors. The resulting output is fluent and seemingly competent but diverges along the model's in-distribution failure modes.

We observe three recurring classes of reasoning errors: *spatial reasoning mistakes*, where the model rejects visible aircraft with faulty geometric arguments about perspective distortion and distance-size relationships; *awareness of simulation*, where the model dismisses the detection task with meta-arguments about image authenticity; and *deference to spurious authority*, where the model contends that human pilots or existing instrumentation would have already detected a threat if one were present.

4.2 Computational performance

Like other sample-based testing approaches, the runtime of AST is proportional to model inference time, making it well-suited for testing larger models (the inference step dominates the nonlinear terms in the time complexity of the MCTS algorithm). It is straightforward to accelerate our approach with batch inference and model parallelization.

Because MCTS is theoretically guaranteed to converge to the optimal solution, AST will eventually identify the highest-likelihood violation, although the time required to do so may be prohibitive.

4.3 Future work

Expanded evaluations. We plan a more comprehensive evaluation of our approach by comparing it with other sample- and gradient-based testing baselines, examining a wider range of monitoring tasks, and performing an ablation study on the margin-based reward function. These experiments will also help address several open research questions, including whether allowing the model to express uncertainty through multiple-choice outputs can decrease the likelihood of failure.

Corrective finetuning. The ability of AST to automatically surface relevant errors makes it a generally useful technique within an iterative development cycle. Since AI models can be continually improved through retraining, it raises the possibility of integrating AST into a finetuning loop to strengthen a model's overall robustness.

Sensitivity analysis. As referenced above, certain properties of sampled inference and zero-temperature inference under uncertain input can be related through Gumbel reparameterization. This connection allows AST to be interpreted as a probe of model sensitivity. Further analysis is needed to quantify how this relationship extends to autoregressive models.

5 Acknowledgments

This work was supported by the System-Wide Safety project under the Airspace Operations and Safety Program (AOSP) of the NASA Aeronautics Research Mission Directorate (ARMD).

References

1. Coulom, R.: Efficient selectivity and backup operators in Monte-Carlo tree search. In: International Conference on Computers and Games. pp. 72–83. Springer (2006)
2. Elhafsi, A., Sinha, R., Agia, C., Schmerling, E., Nesnas, I.A., Pavone, M.: Semantic anomaly detection with large language models. Autonomous Robots **47**(8), 1035–1055 (2023)
3. Ganai, M., Sinha, R., Agia, C., Morton, D., Di Lillo, L., Pavone, M.: Real-time out-of-distribution failure prevention via multi-modal reasoning. arXiv preprint arXiv:2505.10547 (2025)
4. Guo, Z., Yagudin, Z., Lykov, A., Konenkov, M., Tsetserukou, D.: VLM-auto: VLM-based autonomous driving assistant with human-like behavior and understanding for complex road scenes. In: 2024 2nd International Conference on Foundation and Large Language Models (FLLM). pp. 501–507. IEEE (2024)
5. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with Gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
6. Kochenderfer, M.J.: Decision making under uncertainty: theory and application. MIT press (2015)

7. Lee, R., Mengshoel, O.J., Saksena, A., Gardner, R.W., Genin, D., Silbermann, J., Owen, M., Kochenderfer, M.J.: Adaptive stress testing: finding likely failure events with reinforcement learning. Journal of Artificial Intelligence Research **69**, 1165–1201 (2020)
8. Lipkis, R., Agogino, A.: Failure analysis of autonomous systems with RL-guided MCMC sampling. In: 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2025)
9. Meta: Llama-3.2-11B-Vision-Instruct. https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct (2024)
10. Sinha, R., Elhafsi, A., Agia, C., Foutter, M., Schmerling, E., Pavone, M.: Real-time anomaly detection and reactive planning with large language models. arXiv preprint arXiv:2407.08735 (2024)
11. Smyers, E.Q., Katz, S.M., Corso, A.L., Kochenderfer, M.J.: AVOIDDS: Aircraft vision-based intruder detection dataset and simulator (2023), https://arxiv.org/abs/2306.11203
12. Wang, J., Jiang, H., Liu, Y., Ma, C., Zhang, X., Pan, Y., Liu, M., Gu, P., Xia, S., Li, W., Zhang, Y., Wu, Z., Liu, Z., Zhong, T., Ge, B., Zhang, T., Qiang, N., Hu, X., Jiang, X., Zhang, X., Zhang, W., Shen, D., Liu, T., Zhang, S.: A comprehensive review of multimodal large language models: Performance and challenges across different tasks (2024), https://arxiv.org/abs/2408.01319
13. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E.H., Le, Q., Zhou, D.: Chain of thought prompting elicits reasoning in large language models. CoRR **abs/2201.11903** (2022), https://arxiv.org/abs/2201.11903
14. Xu, Z., Zhang, Y., Xie, E., Zhao, Z., Guo, Y., Wong, K.Y.K., Li, Z., Zhao, H.: DriveGPT4: Interpretable end-to-end autonomous driving via large language model (2024), https://arxiv.org/abs/2310.01412
15. Zhou, X., Knoll, A.C.: GPT-4V as traffic assistant: An in-depth look at vision language model on complex traffic events (2024), https://arxiv.org/abs/2402.02205
16. Zhou, X., Liu, M., Yurtsever, E., Zagar, B.L., Zimmer, W., Cao, H., Knoll, A.C.: Vision language models in autonomous driving: A survey and outlook. IEEE Transactions on Intelligent Vehicles pp. 1–20 (2024). https://doi.org/10.1109/TIV.2024.3402136