Evaluating Tool Use in LLM-Based Autonomous Driving

Stella, William; Dr. Hongkai Yu

Department of Computer Science, Cleveland State University, Cleveland, OH

Cleveland State University

Objective

If an LLM choosing a driving trajectory is handed one deterministic tool that reports the exact distance and time-to-collision of the nearest obstacle, does it choose differently than when it has to work those numbers out from raw sensor data?

Background

- LLM decision layers are increasingly proposed for autonomous driving, and the general intuition is to give the model a tool for every quantity it might need.
- Each feature is an extra calculation, and that can stack over time to become very costly.
- Whether an exactly computed number actually changes a driving decision has not been tested directly.
- This project isolates the question by adding exactly one tool and measuring what happens.

Methods

- TransFuser generates 8 candidate 4-second trajectories per scene.
- deepseek-v4-flash picks one, given the ego state and a raw table of nearby road users' position and velocity.
- In the WITHOUT condition, the model has to derive distance and closing speed itself.
- In the WITH condition, it also gets get_nearest_agent_and_ttc, a forced-callable function returning exact distance and time-to-collision.
- 334 risk-relevant scenes, drawn from an exhaustive survey of 4,795 candidates (6.97% flagged).
- Every scene runs under both conditions, and the chosen trajectories are re-scored with NAVSIM's EPDMS suite for a paired comparison.

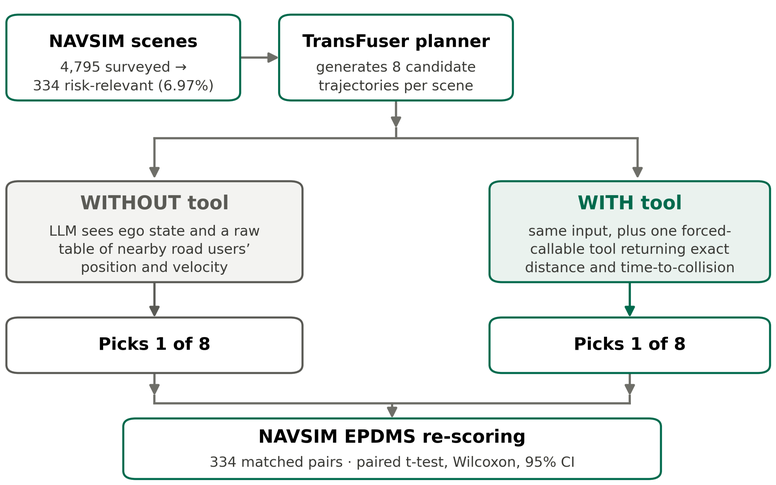

Figure 1. Experimental pipeline.

Experiment 1: Tool vs. No Tool

−0.016
change in PDM score
95% CI [−0.044, 0.012]

53%
of scenes where the tool changed the pick

97.3%
of eligible calls used the tool

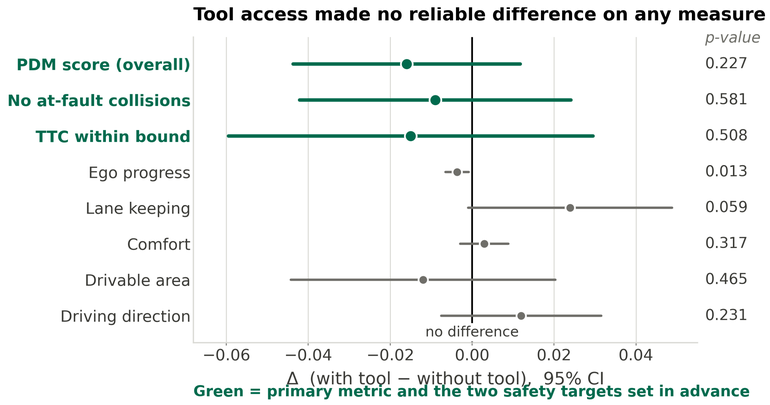

Figure 2. Change in each EPDMS outcome when the tool is added, with 95% confidence intervals.

- Each dot is how far a measure moved when the tool was added and the line is the confidence range. Most lines cross zero, meaning nothing reliably changed.
- The tool was not ignored, and the model was not already near-perfect. It called the tool almost every time and picked a different trajectory in half the scenes, yet scored the same whether or not it had the tool.
- The only row that looks significant is ego progress, but it tracks distance travelled rather than safety, and could still easily be produced by chance.

Experiment 3: Rewriting the Numbers

To test whether the model's stated numbers actually drive its decision, we edited its own reasoning to claim a very different distance and time-to-collision, then asked it to choose again. The method follows existing work on chain-of-thought faithfulness.

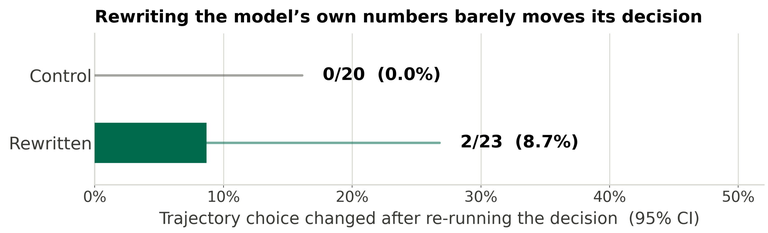

Figure 4. Choice-flip rate under rewritten reasoning

Experiment 2: Transcript Mining

If the tool changed nothing, the natural question is what the model was doing instead. We mined all 334 without-tool transcripts for the numbers it worked out on its own.

- Half the time the model committed to no distance at all, and reasoned qualitatively about the scene instead.
- When it did give a number, the median error was roughly the size of the true value itself.
- How accurate that number was bore no relationship to how well the chosen trajectory scored (r ≈ 0.00 across every metric and subset we tested).

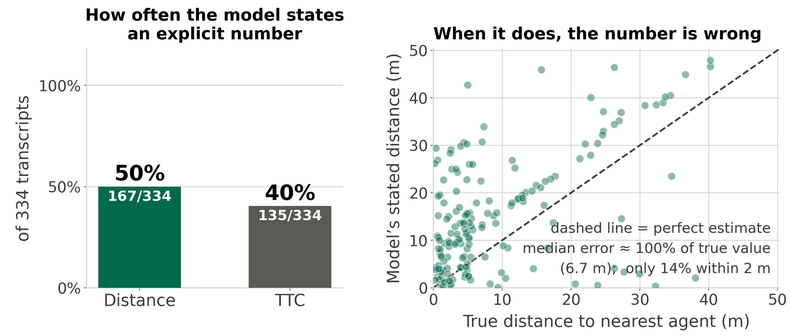

Figure 3. Left: how often the model commits to an explicit number. Right: stated distance against deterministic ground truth.

Limitations

- Experiments 1 and 2 (n = 334) are well powered. Experiment 3 (n = 29) is not, and should be read as suggestive rather than confirmatory.
- Only one model and one decision structure were tested.
- The model used was relatively weak compared to frontier LLMs.

Conclusion

This task is not bottlenecked on numeric precision. Choosing among 8 pre-filtered, already reasonable trajectories seems to be settled by a coarse qualitative read of the scene rather than by arithmetic, which is why an exact number has nothing to improve. Tooling for an agentic driving system should be chosen deliberately, because some precision is simply lost inside the model.

Acknowledgments

Thanks to Dr. Hongkai Yu and Dr. Jiacheng Guo for their guidance on this project.

Code & data