Automotive-ENV: Benchmarking Multimodal Models in Automotive Cockpit Environments

Junfeng Yan*[1] Biao Wu*[1] Meng Fang[2] Ling Chen[1]

[1] AAII, University of Technology Sydney [2] University of Liverpool





UNIVERSITY OF LIVERPOOL



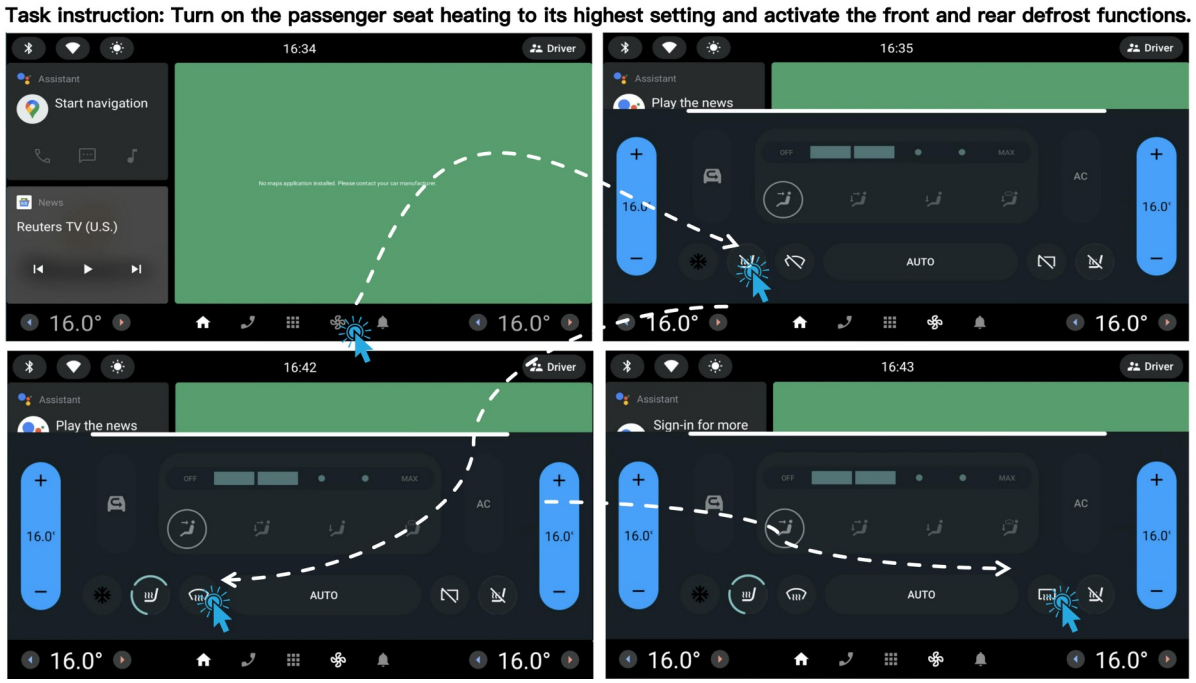
Performance in cockpit systems remains largely unexplored

Multimodal models have demonstrated strong capabilities in web, desktop, and mobile GUI environments.

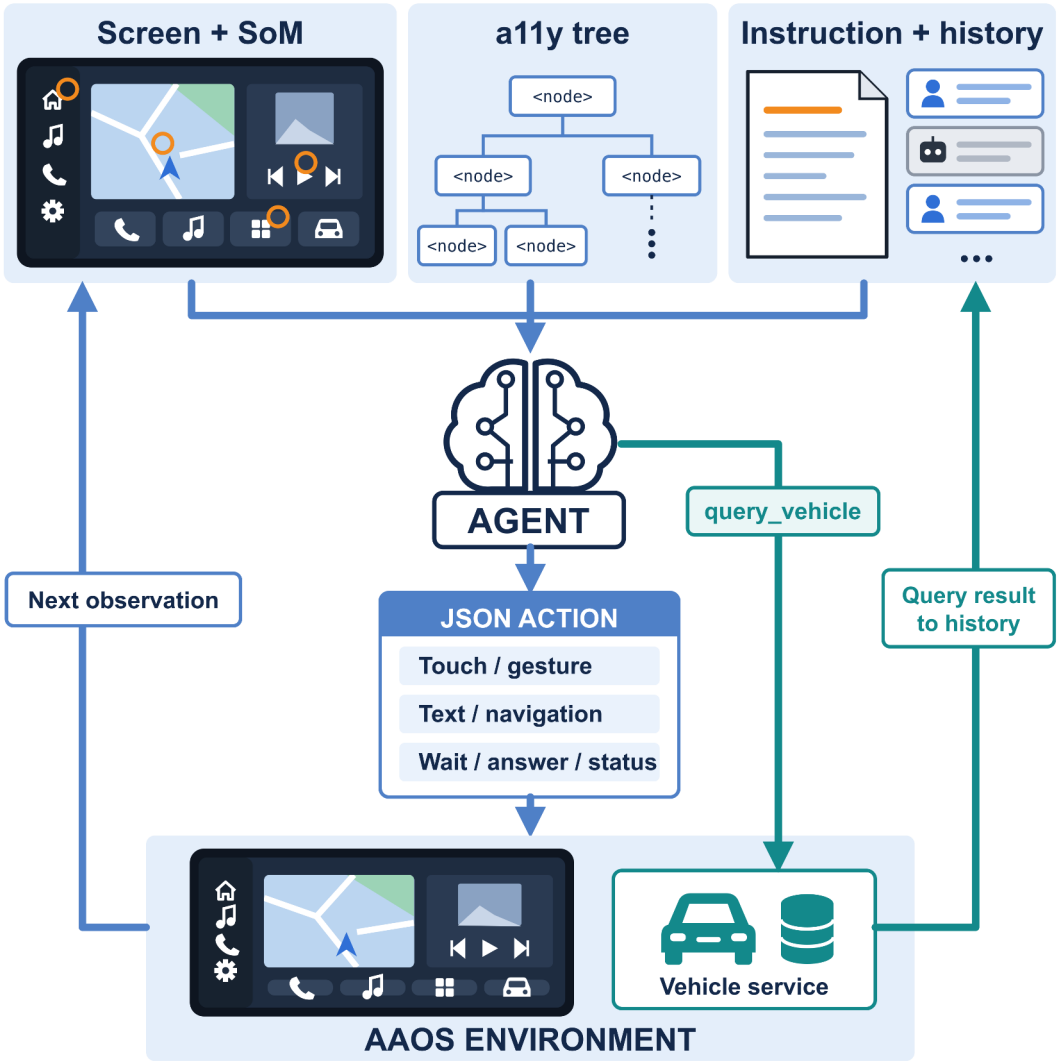
In-vehicle GUIs introduce unique challenges

- Highly dynamic and safety-critical contexts
- Brief, ambiguous, or incomplete commands, because drivers must prioritize road attention
- Mistakes can have immediate safety implications

“I can’t see through the windshield, it’s all fogged up.”



Automotive-ENV: built on Android Automotive OS



Observation

Task instruction, cockpit screenshot, Set-of-Mark annotated screenshot, structured UI metadata, and a compact summary of previous interaction history

Vehicle-state signals

Including HVAC status, speed, fuel level, tire pressure, seatbelt state, and ambient temperature

Action

Structured JSON actions: tapping, long pressing, scrolling, swiping, text input, application navigation, querying vehicle state, textual responses, and terminating an episode

420 tasks with deterministic state-based validation

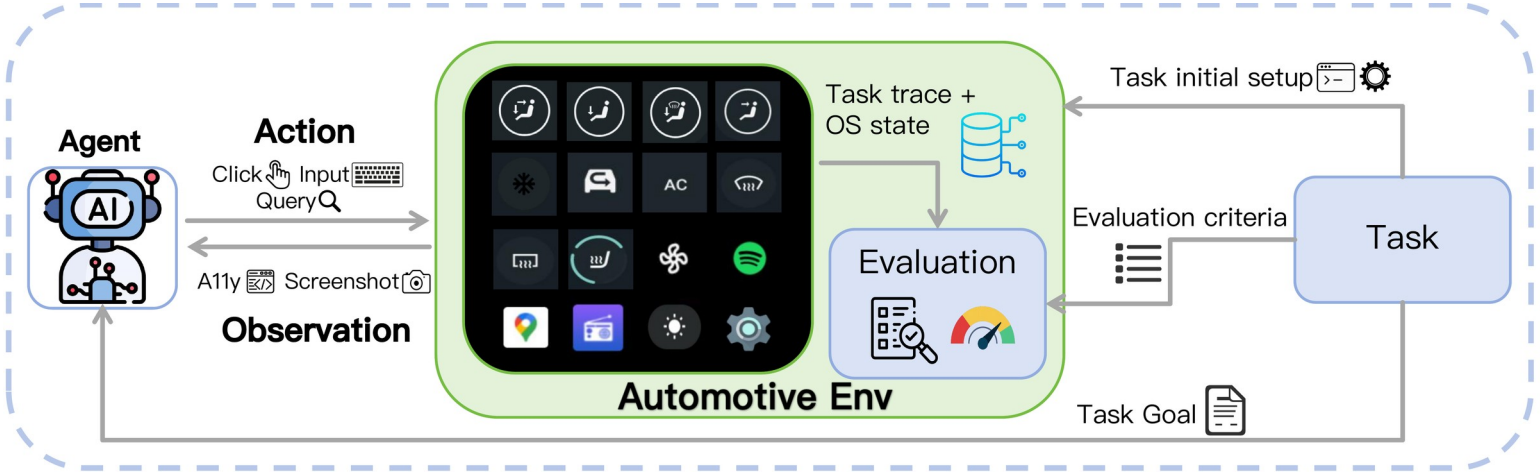
Subset	#	Target
General Explicit	57	Direct control
General Implicit	83	Intent inference
Safety-Critical Explicit	80	Explicit hazard
Safety-Critical Implicit	200	Latent hazard
Total	**420**	—

Deterministic state-based evaluation

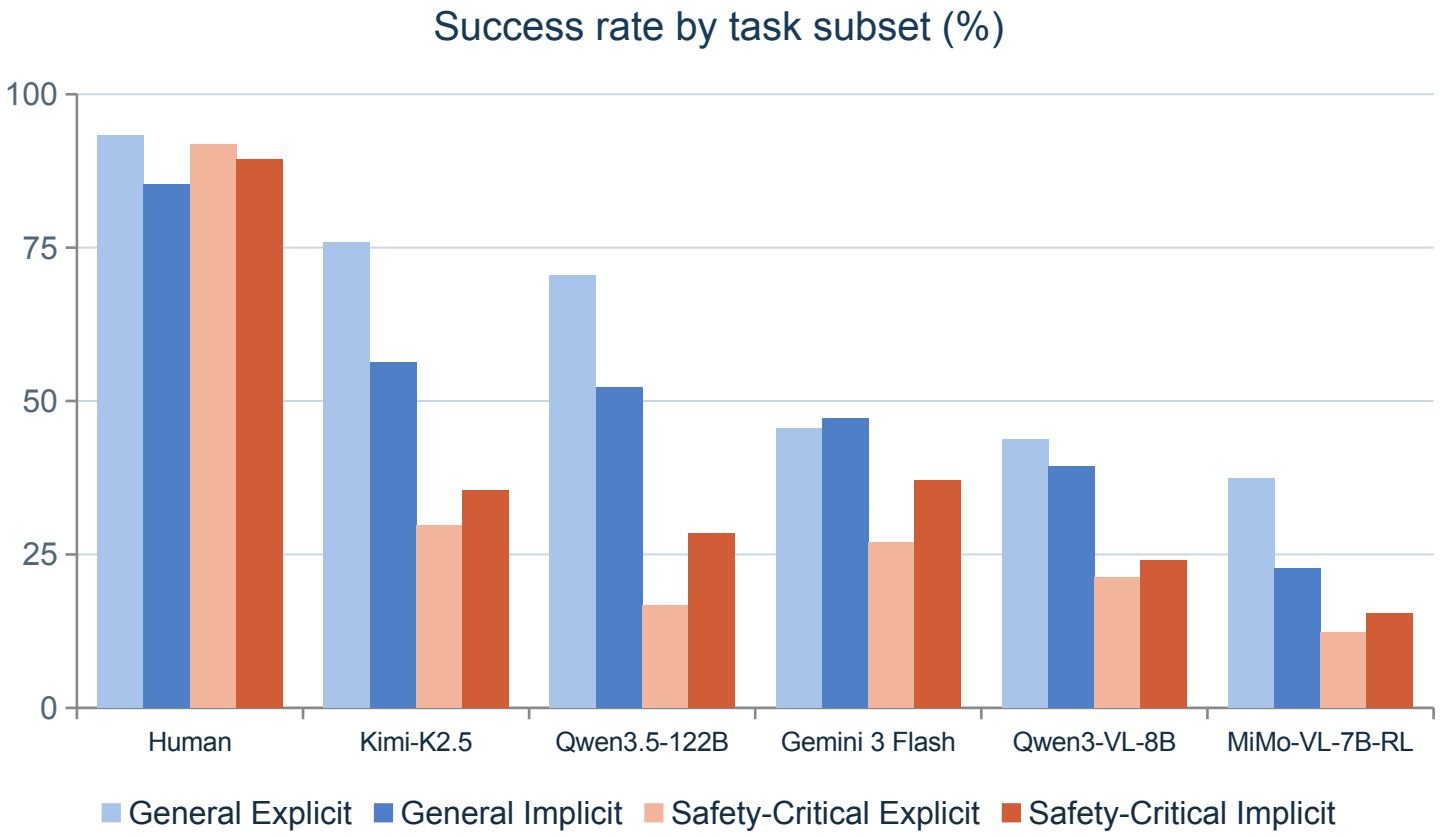
Unlike prior GUI benchmarks that rely on action matching or LLM-based judging, Automotive-ENV uses reproducible validators over UI states, vehicle signals, and safety-related conditions.

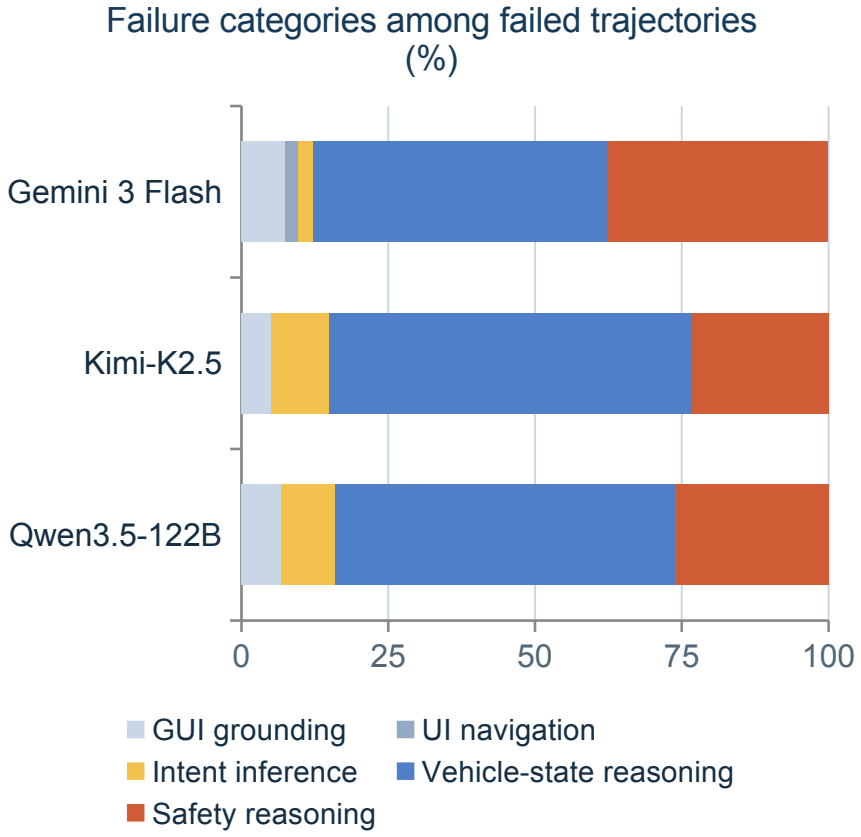
Set the fan speed to off.

```
vhal_equals(HVAC_FAN_SPEED, 0)
```



Current models remain far from reliable





Large multimodal models perform better on General tasks, but their performance drops substantially on implicit and safety-critical tasks. GUI-specialized models show limited transfer.

Failure modes and conclusion

Consistent failure modes

- Unsafe literal obedience
- Missing vehicle-control common sense
- Failures in implicit user-intent inference

Conclusion

Automotive-ENV exposes these limitations and provides a reproducible foundation for studying cockpit grounding, vehicle-state reasoning, and developing safety-aware agents.

Future work

Extend Automotive-ENV toward more industrial-grade simulators and broader in-vehicle software stacks.



Paper



Code



Project Page