Structured Outputs Are Not Controllers: Evaluating LLM Roles in Constrained Greenhouse Control

Tahsin Özgür Koç

Abstract

Large language models are increasingly proposed as general-purpose components for decision-making systems, including domains that require structured outputs and safety constraints. This paper studies a narrow but important question: in a low-entropy greenhouse control task, should an LLM be used as a direct actuator controller, or is it more useful as a high-level supervisor and goal interpreter? We build a reproducible grow-box control benchmark with classical rule-based, hysteresis, and random-forest controllers; direct LLM controllers with structured output; LLM supervisor variants; hybrid safety gates; and an ambiguous human-goal interpretation task. Across the evaluated settings, direct Mistral control did not outperform classical baselines. A rules-aware direct LLM reached 0.28 exact match on 50 sampled static decisions, while a random forest reached 0.555 on the broader static test frame and a calibrated rule controller reached 0.84 on the sampled supervisor set. Structured output eliminated JSON/schema failures in later prompts, but did not guarantee semantic control quality. Supervisor and goal-interpretation results were more promising but still did not beat calibrated or keyword baselines. The central lesson is that structured outputs are interface guarantees, not control guarantees; LLMs should be separated from low-level actuation unless their signals are validated, gated, and audited.

Code and data availability. The experiment code, generated datasets, result tables, and reproducibility scripts are available at `https://github.com/tahsinkoc/llm-structured-control`.

1 Introduction

Control systems often reward properties that are not natural strengths of large language models: determinism, low latency, stable behavior under small state changes, direct grounding in sensor values, and predictable behavior under constraints. At the same time, LLMs are attractive for tasks that sit around the control loop: interpreting human goals, producing structured reports, flagging anomalies, explaining risks, and translating ambiguous instructions into explicit constraints.

This tension motivates the question studied here:

> In structured, constrained, low-level control problems, should LLMs act as direct controllers, or should they be used as higher-level supervisory components?

We investigate this question in a greenhouse grow-box simulator. The setting is deliberately small: the controller observes environmental state variables such as temperature, humidity, soil moisture, time, and light, then chooses actuator actions such as irrigation duration, LED level, ventilation, heating, and shade. The simplicity is intentional. If an LLM is not competitive in a low-entropy control problem with explicit action schemas and target ranges, that is evidence against using it as a default actuator policy in more demanding settings.

The study evaluates five LLM roles: direct actuator controller, direct controller with validator/retry, high-level supervisor, hybrid controller coupled to rule-based actuation, and ambiguous human-goal interpreter.

The main finding is negative but useful. The LLM was able to produce valid structured outputs, especially after prompt and schema adjustments. However, valid JSON did not imply correct control decisions. Classical controllers remained stronger low-level baselines, and calibrated classical control explained most of the improvement observed in hybrid systems. The LLM showed more promise as a high-level interpreter and risk reporter, but even there prompt v2 nearly matched rather than exceeded a simple keyword baseline.

2 Contributions

This paper makes four contributions.

First, it provides a role-based evaluation of LLMs in a constrained greenhouse control benchmark. Rather than asking whether an LLM "works," the benchmark distinguishes direct actuation, structured-output generation, supervision, hybrid gating, and human-goal interpretation.

Second, it shows empirically that structured output reliability is not the same as semantic control quality. Mistral direct and supervisor v2 runs reached zero invalid-output and zero fallback rates, while exact-match and risk-flag performance remained limited.

Third, it evaluates whether LLM supervisor signals improve classical control when connected to actuator policies. Aggressive supervisor coupling degraded performance, while conservative safety gates avoided harm but did not improve over calibrated rule control.

Fourth, it adds an ambiguous human-goal benchmark. Prompt v2 substantially improved Mistral's goal interpretation, raising the weighted rubric score from 0.72712 to 0.82888, but it still narrowly trailed the keyword baseline at 0.83757.

3 Related Work

This benchmark sits at the intersection of LLM-based embodied decision making, structured output generation, safe supervisory control, and greenhouse climate-control benchmarks.

LLMs for planning and embodied control. Prior work has explored LLMs as planners, tool users, or policy components in robotics and embodied tasks. SayCan grounds high-level language-model knowledge in pretrained robotic skills and affordance/value functions rather than letting the language model act alone [1]. Grounded Decoding similarly argues that embodied use of LLMs requires decoding outputs that are both likely under the language model and realizable under grounded environmental objectives [5]. ReAct studies interleaved reasoning and acting, where language-model actions gather information from tools or environments rather than remaining pure text generation [10]. Robotics surveys frame LLM use across perception, decision-making, planning, control, and interaction, while emphasizing that embodiment creates grounding and safety challenges beyond language-only benchmarks [11]. These works motivate our distinction between direct actuator control and high-level supervisory or interpretive roles.

LLMs in control engineering. ControlBench evaluates frontier LLMs on undergraduate-level classical control problems and shows that control engineering is a meaningful test of mathematical reasoning and design explanation [6]. Our work asks a different question: not whether LLMs can

solve control homework problems, but whether a schema-constrained LLM should emit low-level actuator decisions inside a closed-loop control benchmark.

Structured output and constrained decoding. JSON schemas and constrained decoding improve machine-readability and reduce parse failures. JSONSchemaBench provides a large benchmark for evaluating constrained decoding across real-world JSON schemas and emphasizes dimensions such as constraint compliance, coverage of schema features, efficiency, and output quality [4]. Our results focus on the control-specific gap that remains after syntax succeeds: schema-valid outputs can still make semantically poor control decisions.

Safe control and supervisory architectures. Classical safety work motivates keeping learned or uncertain policies behind validation and shielding layers. Shielded reinforcement learning synthesizes a shield that restricts or corrects unsafe actions according to temporal-logic safety specifications [2]. Control barrier function methods formalize safety constraints as forward-invariance conditions and combine them with performance objectives through real-time quadratic programs [3]. These lines of work support our use of fallback, validation, and sensor-confirmed gates around LLM outputs.

Greenhouse and crop-control benchmarks. GreenLight-Gym provides a fast open-source RL benchmark environment for greenhouse crop-production control, built on the GreenLight model and designed for standardized controller evaluation [9]. Other greenhouse-control work compares reinforcement learning and model predictive control, noting that MPC is a common greenhouse climate-control baseline and that RL is increasingly studied [8]. Recent RL-based MPC work also studies greenhouse climate control under prediction uncertainty and physical constraints [7]. The present grow-box simulator is smaller than these domain benchmarks; its purpose is controlled role comparison rather than full greenhouse realism.

4 Benchmark and Experimental Setup

The benchmark is a structured grow-box control problem. The simulator exposes environmental state variables and applies controller outputs to update the system. The action space includes irrigation duration, LED level, ventilation level, heating, and shade. Expert-labeled static datasets and closed-loop trajectories provide the reference behavior.

The evaluation uses five condition families: normal, boundary, noisy, missing sensor, and OOD weather. Classical baselines include a rule-based controller, a rule-plus-hysteresis controller, and a random forest trained to imitate expert policy decisions. Later experiments add calibrated rule variants after observing that actuator thresholds, not LLM reasoning, were a major source of error.

The LLM experiments use Mistral Small 2603 through a structured-output interface where possible, with local parsing and schema validation retained as the final source of truth. All paid API runs are intentionally small and logged with raw response, parsed output, final output, validation errors, fallback usage, latency, and token counts.

Table 1: Experiment protocol summary.

Component	Protocol detail
State variables	Temperature, humidity, soil moisture, time of day, outside temperature, and outside light.
Action variables	Irrigation duration, LED level, ventilation level, heating enablement, and shade enablement.
Test conditions	Normal, boundary, noisy, missing-sensor, and OOD-weather cases.
Classical baselines	Rule-based, rule+hysteresis, random forest, calibrated rule, and calibrated safety-gate variants.
LLM protocols	Direct controller, validator/retry controller, high-level supervisor, hybrid gate, and H5 goal interpreter.
Logging	Raw response, parsed output, final output, validation errors, fallback use, latency, token counts, and estimated cost.
Sampling caveat	Broad classical baselines use 1275 static cases; paid LLM and hybrid analyses use smaller sampled sets because of API cost and rate limits.

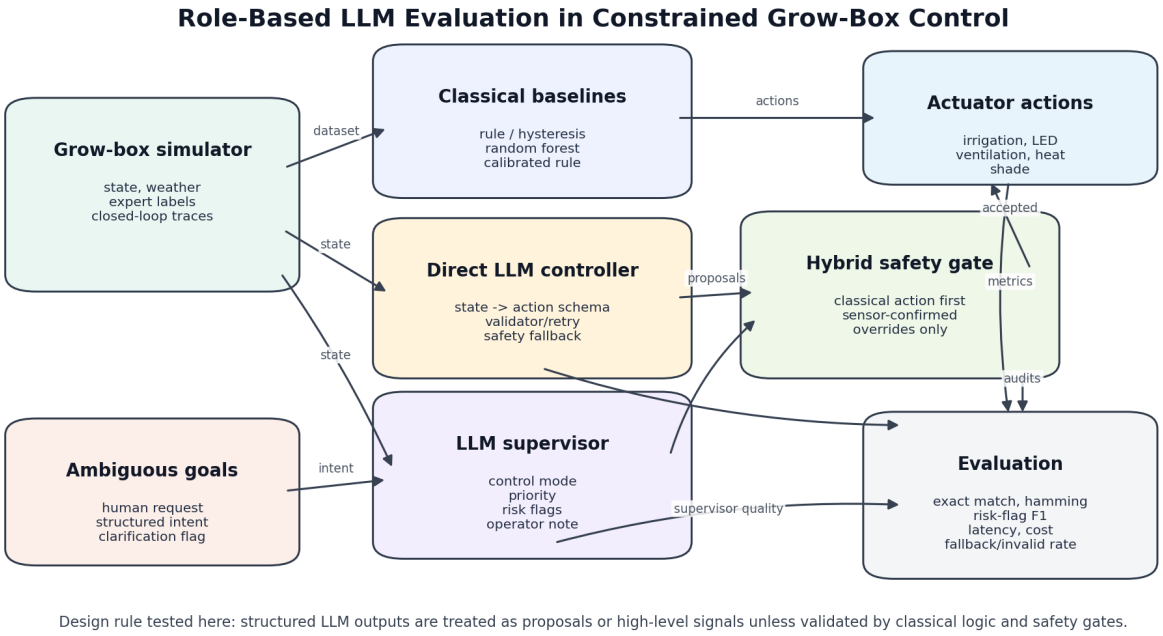

Figure 1: Role-based experimental architecture. The benchmark separates classical baselines, direct LLM actuator control, LLM supervision, hybrid safety gates, ambiguous-goal interpretation, and evaluation metrics.

5 Controller and Supervisor Variants

Direct LLM controller. The LLM receives state and objective information and returns actuator commands in a fixed action schema. Two prompt conditions are evaluated: naive and rules-aware.

Validator/retry controller. The same direct-control setting is wrapped in parsing, schema validation, semantic validation, retry, and safety fallback logic.

LLM supervisor. The LLM does not emit actuator commands. It emits high-level fields: control mode, priority, target adjustments, risk flags, and an operator note.

Hybrid controllers. Supervisor decisions are connected to classical actuation in two ways: an aggressive hybrid that maps supervisor outputs into actuator thresholds, and a conservative safety gate that starts from rule-based action and applies only sensor-confirmed overrides.

Calibrated hybrid controllers. After observing low-level calibration errors, a calibrated rule controller and a calibrated supervisor safety gate are evaluated on the same sampled set.

Goal interpreter. For H5, the LLM translates ambiguous human goals into structured high-level intent: control mode, primary priority, secondary priorities, hard constraints, allowed tradeoffs, and clarification requirement.

6 Metrics

Static decision quality is measured using exact match, hamming loss, per-actuator accuracy, and macro F1 where applicable. Closed-loop quality is measured using total cost, constraint-violation minutes, water use, energy proxy, and actuator switching. LLM reliability is measured using invalid-output rate, fallback-usage rate, retry rate, latency, input tokens, output tokens, and estimated token cost. Supervisor quality is measured using exact match for structured supervisor decisions and set accuracy/Jaccard/F1 for risk flags.

H5 goal interpretation is measured using both strict set matching and a weighted semantic rubric. The rubric scores health-guard preservation, goal alignment, clarification behavior, constraint overlap, and tradeoff overlap. This rubric avoids treating every set mismatch as equally severe while preserving safety-critical errors such as missing clarification.

7 Results

Table 2: Main result summary by evaluated role.

Role	Model/controller	Samples	Main score	Interpretation
Broad static baseline	Random forest	1275	0.555 exact	Strongest broad low-level baseline.
Direct LLM control	Mistral rules-aware	50	0.280 exact	Improved over naive prompting, but below classical baselines.
LLM supervisor	Mistral supervisor v2	50	0.240 exact	Schema-valid, but limited semantic supervisor quality.
Hybrid actuation	Supervisor rule hybrid	50	0.400 exact	Aggressive coupling degraded rule-based control.
Calibrated control	Calibrated rule	50	0.840 exact	Most improvement came from classical calibration.
Calibrated hybrid	Calibrated safety gate	50	0.840 exact	Matched calibrated rule with zero action changes.
Goal interpretation	Mistral H5 v2	16	0.829 rubric	Nearly matched, but did not exceed, keyword baseline at 0.838.

7.1 Classical Baselines

On the broad static test frame, the random forest reached the strongest exact-match score:

Table 3: Broad static baseline results.

Controller	Samples	Exact match	Hamming loss
random_forest	1275	0.555	0.119
rule_based	1275	0.392	0.148
rule_hysteresis	1275	0.340	0.181

The baseline results establish that simple classical controllers are nontrivial competitors. They are also far faster than the LLM calls, with rule-based decisions occurring at approximately millisecond or sub-millisecond scale while Mistral calls are around one second or more depending on prompt and task.

7.2 Direct LLM Control

The naive Mistral prompt produced valid JSON but poor action matching. A rules-aware prompt improved direct-control exact match to 0.28 on 50 sampled static cases, while validator+retry reached 0.26. Both had invalid-output and fallback-usage rates of 0.00 in that run.

This result supports the first main conclusion: prompt quality matters, but even improved direct LLM control did not beat classical baselines. The best broad static baseline, random forest, reached 0.555 exact match on 1275 samples. The calibrated rule controller later reached 0.84 exact match on the sampled supervisor set.

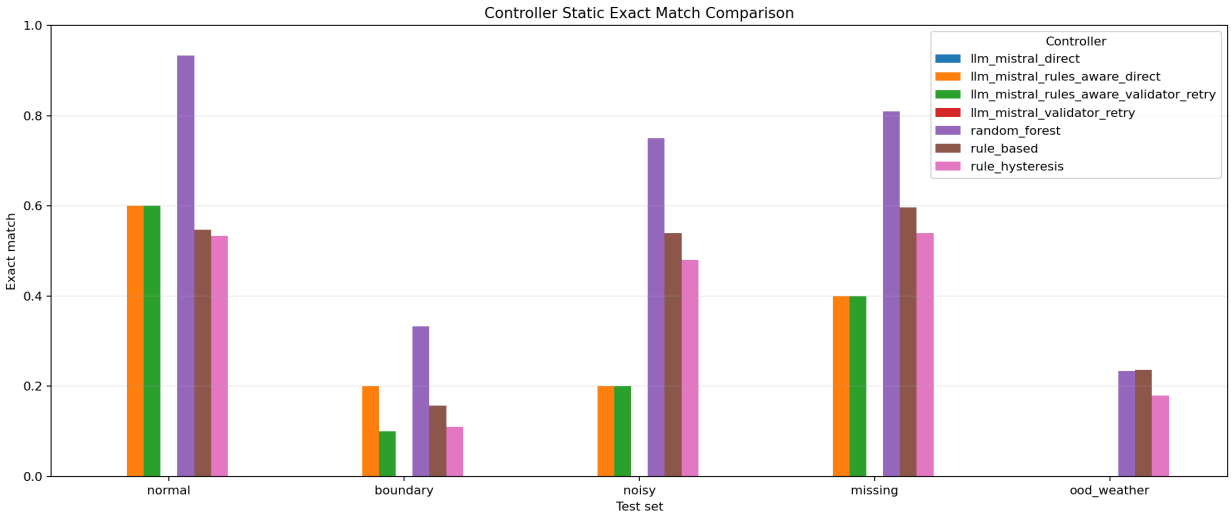

Figure 2: Static exact-match comparison across classical baselines and direct LLM controller variants. Paid LLM runs use smaller sampled sets, so broad baseline values and LLM sampled values should be interpreted with the sampling caveat described in the limitations.

7.3 Structured Output Reliability Is Not Semantic Reliability

Structured output and schema adjustments substantially reduced parse and schema errors. Direct rules-aware control and supervisor v2 both reached invalid-output rate 0.00. However, supervisor v2 exact match was 0.24 and risk-flags set accuracy was 0.46. Direct rules-aware exact match was 0.28.

This distinction is central: a structured output interface can make the model easier to parse without making it a good controller. Schema validity should therefore be treated as a necessary interface condition, not as evidence of safe or correct behavior.

7.4 Supervisor and Hybrid Control

Supervisor v1 initially appeared strong only when fallback-inflated final metrics were considered. After separating raw LLM performance from fallback behavior, the valid/fallback distinction showed that supervisor v1 had substantial schema failures. Prompt v2 fixed the structural issue:

- valid-output rate: 1.00;
- invalid-output rate: 0.00;
- fallback-usage rate: 0.00;
- exact match: 0.24;
- control-mode accuracy: 0.76;
- priority accuracy: 0.80;
- risk-flags set accuracy: 0.46.

When connected to actuator decisions, the aggressive supervisor-rule hybrid underperformed rule-based control:

Table 4: Hybrid controller results on the sampled supervisor set.

Controller	Samples	Exact match	Hamming loss
Rule-based	50	0.46	0.124
Supervisor rule hybrid	50	0.40	0.160
Supervisor safety gate	50	0.46	0.124
Calibrated rule	50	0.84	0.040
Calibrated supervisor safety gate	50	0.84	0.040

The calibrated safety gate matched the calibrated rule baseline and produced zero action changes relative to it. This means the improvement came from classical actuator calibration, not from the LLM supervisor.

7.5 Risk Flag Audit

The risk-flag audit shows a mixed pattern. Some flags were useful: high_humidity reached F1 0.91, while heat_stress and dry_soil each reached F1 0.80. Other flags were weak: low_light reached F1 0.35, ood_weather reached F1 0.31, and missing_sensor produced 8 false positives where the truth set contained no positives.

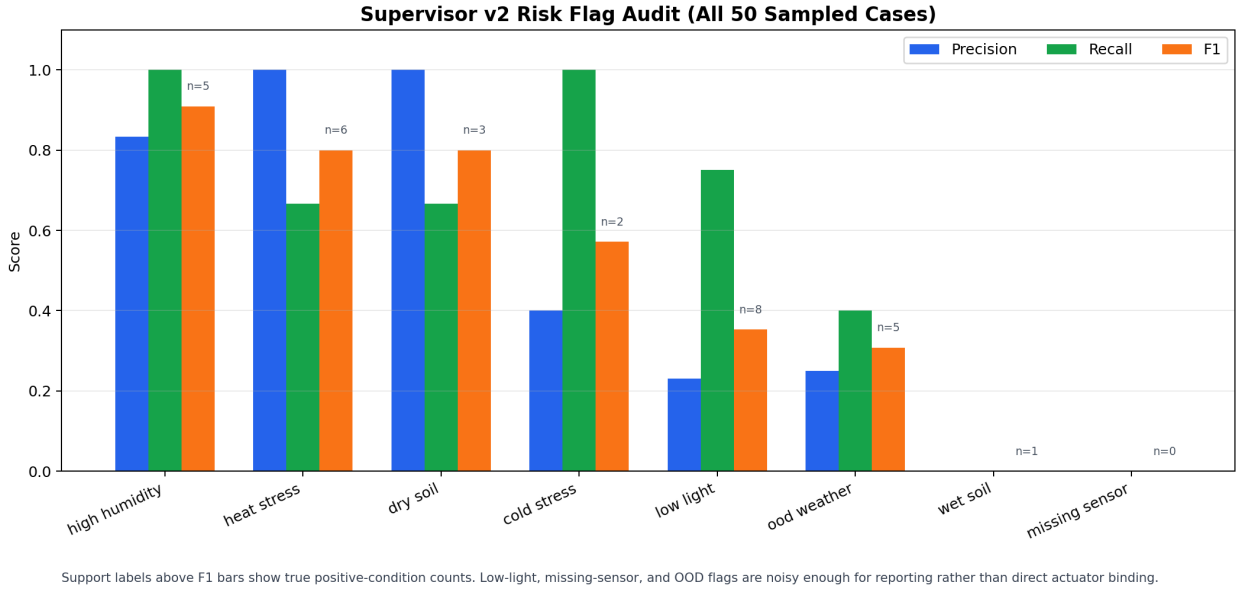

Figure 3: Risk-flag precision, recall, and F1 for supervisor v2 across all 50 sampled cases. High-humidity, heat-stress, and dry-soil flags are useful; low-light, OOD-weather, wet-soil, and missing-sensor behavior should remain advisory unless improved.

The action-impact audit reinforces the same point. Aggressive hybrid use caused 14 action changes, with 1 helpful and 9 harmful. Conservative safety gates avoided degradation but did not

improve calibrated control. A practical design should keep weak flags as operator-facing reports and connect only sensor-confirmed, high-precision flags to control gates.

7.6 Ambiguous Human Goals

H5 evaluates whether an LLM adds value when the task is not direct actuator control, but interpretation of ambiguous or conflicting human goals. The benchmark contains 16 scenarios across clear, tradeoff, conflicting, and underspecified cases.

Prompt v1 produced valid structured output but was weaker than the keyword baseline. Prompt v2 improved substantially:

Table 5: H5 goal interpretation comparison.

Interpreter	Strict field match	Rubric score	Clarification accuracy	Invalid output
keyword_goal_baseline	0.59375	0.83757	1.0000	0.00
Mistral goal interpreter v1	0.43750	0.72712	0.8125	0.00
Mistral goal interpreter v2	0.55208	0.82888	0.9375	0.00

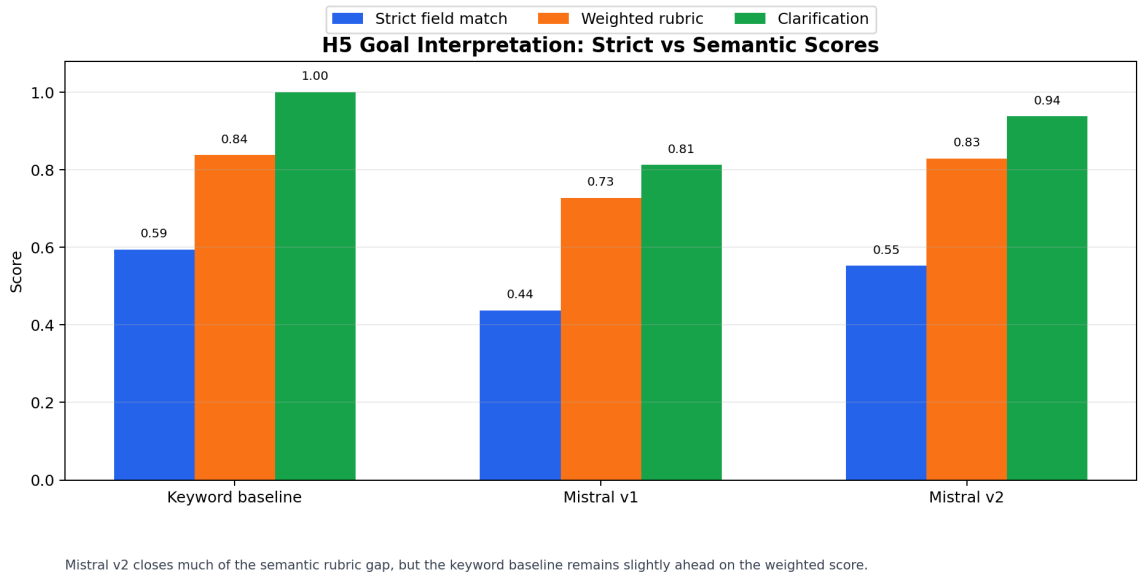

Figure 4: H5 goal-interpretation scores. Mistral v2 improves over v1 and nearly reaches the keyword baseline on the weighted semantic rubric, but it does not surpass it.

Mistral v2 nearly closed the rubric gap but did not exceed the keyword baseline. It eliminated clarification false negatives and improved health-guard and goal-alignment scores. The result should therefore be interpreted as a prompt-sensitive near miss rather than a clear LLM win.

7.7 Cost and Latency

Using the Mistral Small 2603 price assumptions recorded in the cost table, the total estimated token cost across Mistral experiments was approximately 0.0386 USD. H5 v2 cost approximately 0.00185

USD for 16 samples. Token cost was small in this benchmark, but latency and operational delays from rate limits remain important for real control settings.

The practical point is not that the API cost was large in this experiment. It is that a direct LLM actuator path pays latency, rate-limit, and reliability costs for decisions where classical controllers are cheaper, faster, and stronger.

Table 6: Estimated Mistral API token cost for the paid runs.

Run	Samples	Input tokens	Output tokens	Estimated USD
Direct naive	50	15306	2248	0.00364
Direct validator/retry	50	15358	2256	0.00366
Direct rules-aware	50	30406	2185	0.00587
Rules-aware validator/retry	50	30481	2177	0.00588
Supervisor v1	50	28806	4882	0.00725
Supervisor v2	50	37506	4469	0.00831
H5 v1	16	3163	2796	0.00215
H5 v2	16	5339	1751	0.00185
Total	332	166365	24764	0.03861

8 Discussion

The benchmark supports a conservative design rule: keep LLMs away from low-level actuation unless they beat strong classical baselines and pass action-impact audits. In this study, they did not.

At the same time, the results do not imply that LLMs are useless in control systems. Their more plausible role is outside the inner control loop. The supervisor and H5 experiments show that LLMs can produce structured, useful high-level signals after careful prompting. However, those signals should be treated as advisory or operator-facing unless they are grounded in sensor checks and validated by classical logic.

The strongest positive evidence for LLM use is prompt sensitivity in H5. Prompt v2 substantially improved clarification behavior and semantic rubric score. This suggests that high-level goal interpretation may be a productive direction, but the evidence is not yet strong enough to claim superiority over simple baselines.

9 Limitations

The benchmark is small and simulated. No physical grow-box experiment was run. Paid API runs used sampled subsets, while some classical baselines were evaluated on broader frames. Exact values should therefore be compared with the sampling caveat.

The simulator is simpler than greenhouse benchmarks such as GreenLight-Gym and does not include full crop growth dynamics. The H5 benchmark is handcrafted and contains only 16 scenarios. The weighted semantic rubric is useful for analysis but should be validated with human ratings or a larger scenario set. Only Mistral Small 2603 was used for the paid LLM runs reported here. Results may differ with other models.

10 Conclusion

This study tested LLMs across several roles in a structured greenhouse control benchmark. The main result is that LLMs did not outperform classical baselines as low-level actuator controllers, even when structured outputs were valid. Supervisor mode improved format reliability but did not improve calibrated actuator control. Ambiguous human-goal interpretation was more promising, especially with prompt v2, but still did not beat a simple keyword baseline.

The broader lesson is that structured outputs are not controllers. They make LLM outputs easier to consume, but they do not guarantee correct, safe, or useful control decisions. In constrained control systems, LLMs are best treated as high-level interpreters or reporters whose outputs are validated, gated, and audited before they influence actuators.

A H5 Weighted Semantic Rubric

The strict H5 metrics compare structured fields directly. This is useful for reproducibility, but too brittle for ambiguous human-goal interpretation. A prediction can miss one set element while preserving the important safety intent, or it can match a surface field while missing a critical clarification requirement. The weighted semantic rubric therefore scores five components:

$$\begin{aligned}\text{rubric\_score} = {} & 0.30 \cdot \text{health\_guard} + 0.20 \cdot \text{goal\_alignment} \\ & + 0.25 \cdot \text{clarification} + 0.15 \cdot \text{constraint\_overlap} \\ & + 0.10 \cdot \text{tradeoff\_overlap}.\end{aligned} \tag{1}$$

The rubric pass threshold is 0.70. Health guard score gives most credit for preserving plant-health protection and critical constraints. Goal alignment gives full credit when both control mode and primary priority match, with partial credit for safety-conservative near misses. Clarification is binary and intentionally strict because false negatives on underspecified or conflicting goals can send an uncertain request into the control stack without operator review. Constraint and tradeoff overlap scores use Jaccard overlap.

Table 7: H5 weighted semantic rubric summary.

Interpreter	Rubric	Pass	Health	Goal	Clarification	Field match
keyword baseline	0.838	0.812	0.916	0.747	1.000	0.594
Mistral v1	0.727	0.688	0.928	0.794	0.812	0.438
Mistral v2	0.829	0.938	0.934	0.847	0.938	0.552

Mistral v2 almost closes the rubric gap to the keyword baseline, but does not beat it. The main remaining weakness is tradeoff representation: Mistral v2 has 0.427 tradeoff overlap versus the keyword baseline at 0.771.

Table 8: Mistral v2 H5 scores by ambiguity level.

Ambiguity level	Samples	Rubric score	Strict field match	Pass rate
clear	4	0.900	0.792	1.000
conflicting	4	0.735	0.250	0.750
tradeoff	4	0.810	0.500	1.000
underspecified	4	0.871	0.667	1.000

The hardest group is conflicting goals. Mistral v2 asks for clarification, but strict field match is only 0.250 and the average rubric score is 0.735. This supports the cautious interpretation of H5 as a prompt-sensitive near miss rather than a decisive LLM win.

B Supervisor Risk-Flag Audit

Supervisor v2 produced valid structured output on all 50 sampled cases, but risk flags varied sharply in quality. This matters because a risk flag can be used either as an operator-facing report or as an input to actuator gating.

Table 9: Supervisor v2 risk-flag metrics across all 50 sampled cases.

Flag	True support	Predicted support	TP	FP	FN	F1
high_humidity	5	6	5	1	0	0.91
heat_stress	6	4	4	0	2	0.80
dry_soil	3	2	2	0	1	0.80
cold_stress	2	5	2	3	0	0.57
low_light	8	26	6	20	2	0.35
ood_weather	5	8	2	6	3	0.31
wet_soil	1	0	0	0	1	0.00
missing_sensor	0	8	0	8	0	0.00

High-humidity, heat-stress, and dry-soil are candidates for sensor-confirmed gating. Low-light, OOD-weather, missing-sensor, and wet-soil should remain advisory until improved. Missing-sensor is especially problematic because the truth set contained no positives, but the supervisor predicted it 8 times.

C Action-Impact Audit

The action-impact audit compares hybrid controller outputs against their classical baseline on the same 50 sampled cases.

Table 10: Hybrid action-impact audit.

Controller	Changes	Helped	Hurt	Neutral	Baseline exact	Hybrid exact
calibrated safety gate	0	0	0	50	0.84	0.84
supervisor safety gate	0	0	0	50	0.46	0.46
supervisor rule hybrid	14	1	9	40	0.46	0.40

Aggressive coupling of supervisor signals to actuator thresholds degraded performance. Conservative gates avoided harm, but in this run they also made no action changes. After actuator calibration, the supervisor safety gate exactly matched the calibrated rule controller; the improvement came from classical calibration, not from the LLM supervisor.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. URL https://arxiv.org/abs/2204.01691.

[2] Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding, 2017. URL https://arxiv.org/abs/1708.08611.

[3] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017. doi: 10.1109/TAC.2016.2638961. URL https://authors.library.caltech.edu/records/jnhr0-1ww05.

[4] Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. Jsonschemabench: A rigorous benchmark of structured outputs for language models, 2025. URL https://arxiv.org/abs/2501.10868.

[5] Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, and Brian Ichter. Grounded decoding: Guiding text generation with grounded models for embodied agents, 2023. URL https://arxiv.org/abs/2303.00855.

[6] Darioush Kevian, Usman Syed, Xingang Guo, Aaron Havens, Geir Dullerud, Peter Seiler, Lianhui Qin, and Bin Hu. Capabilities of large language models in control engineering: A benchmark study on gpt-4, claude 3 opus, and gemini 1.0 ultra, 2024. URL https://arxiv.org/abs/2404.03647.

[7] Samuel Mallick, Filippo Airaldi, Azita Dabiri, Congcong Sun, and Bart De Schutter. Reinforcement learning-based model predictive control for greenhouse climate control, 2025. URL https://arxiv.org/abs/2409.12789.

[8] Bernardo Morcego, Wenjie Yin, Sjoerd Boersma, Eldert van Henten, Vicenç Puig, and Congcong Sun. Reinforcement learning versus model predictive control on greenhouse climate control, 2023. URL https://arxiv.org/abs/2303.06110.

[9] Bart van Laatum, Eldert J. van Henten, and Sjoerd Boersma. Greenlight-gym: Reinforcement learning benchmark environment for control of greenhouse production systems, 2025. URL https://arxiv.org/abs/2410.05336.

[10] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

[11] Fanlong Zeng, Wensheng Gan, Zezheng Huai, Lichao Sun, Hechang Chen, Yongheng Wang, Ning Liu, and Philip S. Yu. Large language models for robotics: A survey, 2025. URL https://arxiv.org/abs/2311.07226.