Benchmarking Problem-solving AI Agents in a Dairy Farm Digital Twin Against Expert-derived Reference Solutions

Jan Saro[1] and Miroslav Pavelka[2]
[1]Department of Systems Engineering, Faculty of Economics and Management, Czech University of Life Sciences Prague, Czech Republic
[2]Department of Information Technologies, Faculty of Economics and Management, Czech University of Life Sciences Prague, Czech Republic

saroj@pef.czu.cz (corresponding author)
pavelka@pef.czu.cz

Abstract: A digital twin may describe an animal accurately without helping a farmer decide what to do next. This study examines that gap through a blinded benchmark of problem-solving agents in a simulated dairy-farm environment. The benchmark comprises 500 scenarios across 13 classes, covering animal health, farm management, sensor problems and situations in which no intervention is needed. Decisions and the steps leading to them are scored against hidden, deterministic reference solutions written by the authors. Six systems were compared, including conventional decision-support systems (DSS) and three configurations of the same language model: one-shot, ReAct and a digital-twin (DT) agent with action simulation and environmental feedback. In the paired evaluation of 300 scenarios, the DT agent improved the correct action rate from 73.7% for the one-shot model to 85.7%, but did not significantly outperform ReAct on this endpoint. The machine-learning (ML) DSS remained strongest overall within the simulator distribution. The DT agent was comparatively effective in difficult and sensor-related scenarios, yet often intervened when the reference solution required no action. No unsafe decisions were observed for this agent under the benchmark rules, but its more elaborate workflow increased API use. These findings show why decision-support agents should be assessed not only on whether they identify a problem, but also on whether they gather useful evidence, select a proportionate response and recognise when to stop. The benchmark is a research environment, not a validated veterinary system.

Keywords: Animal digital twin, Agentic AI, Large language models, Precision livestock farming, Decision support, Benchmark

1. Introduction

An alert is not yet a decision. A rise in milk conductivity may indicate mastitis, but it may also reflect a biased sensor. Reduced rumination may justify a diagnostic test, a check of the ration or continued observation. The practical task is therefore not simply to detect a deviation, but to establish what it means and choose an appropriate response. Precision livestock farming (PLF) provides much of the information needed for this task through continuous monitoring of milk yield, rumination, activity, body temperature and barn conditions (Rutten et al., 2013; Berckmans, 2017). Norton et al. (2019) describe these systems as building digital representations that bring animals closer to the farmer.

Digital twins extend this idea by maintaining a virtual representation of an entity and using incoming data to reason about its state and possible future behaviour (Grieves and Vickers, 2017; Tao et al., 2019). They have consequently been proposed as a development beyond sensor-based livestock monitoring (Neethirajan and Kemp, 2021; Verdouw et al., 2021). Existing applications nevertheless differ considerably in their scope. They include farm-level feasibility studies and virtual livestock buildings, as well as models of individual animal behaviour and anatomy (Jo et al., 2018; Jeong et al., 2023; Table 1).

At the animal level, Han et al. (2022) developed a deep-learning twin to anticipate the next behavioural state of cattle. Zhang et al. (2025) addressed behaviour recognition when some sensor modalities are unavailable, while Gupta et al. (2024) used a twin for teat localisation and shape identification. Brown-Brandl and Tao (2025) discuss the integration of real-time data and twins with animal nutrition models. These studies illustrate the breadth of the field, although agricultural twin applications remain at varied and often early stages of development (Pylianidis et al., 2021; Purcell and Neubauer, 2023).

Table 1: Examples of animal digital-twin solutions and their focus

Study	Species / scope	Focus of the twin
Jo et al. (2018)	Livestock farm	Farm-level twin; feasibility study
Han et al. (2022)	Cattle	Deep-learning twin of animal status; next behavioural state
Jeong et al. (2023)	Pigs	Virtual pig house

Study	Species / scope	Focus of the twin
Gupta et al. (2024)	Dairy cows	Teat localisation and shape identification
Zhang et al. (2025)	Dairy cows	Behaviour recognition with incomplete sensor modalities
Brown-Brandl and Tao (2025)	Livestock	Real-time data, twins and nutrition models (review)
This study	Dairy cows	Twin-based agent scored on decision trajectories against hidden reference solutions

Much of this work concerns how well a twin represents or predicts an animal or its environment. The question considered here is different: can a system use that representation to solve a management problem? Large language model (LLM) agents offer a possible link because they can combine reasoning with tool use (Yao et al., 2023; Wang et al., 2024). However, agent evaluations commonly emphasise task completion or response quality (Liu et al., 2023), and experimental evidence on multi-step decision making within dynamic livestock twins remains limited.

This paper introduces a benchmark that evaluates the decision process rather than the final recommendation alone. It follows problem identification, information gathering, tool use, action selection and the response to environmental feedback, while keeping the reference solution hidden from the system being tested. Three questions guide the study: whether tools and a digital twin improve decisions made with the same LLM (RQ1); how the relative strengths of the systems vary across farm problems (RQ2); and what these differences imply for safety, unnecessary intervention and API use (RQ3).

2. Data and Methods

2.1 Research Design

All systems encountered the same simulated scenarios in a paired design. Each received current sensor readings, barn climate, alerts, recent farm events, pen-level information, parity and days in milk (DIM). Depending on its architecture, it could use tools to obtain further observable evidence.

The reference solution was stored separately and accessed only by the evaluator after the decision (Figure 1). It contained the hidden animal state, correct problem, required, acceptable and unsafe actions, and a recommended sequence. Scenario identifiers were opaque, and automated tests checked that reference fields did not enter observations, tool responses, feedback or prompts.

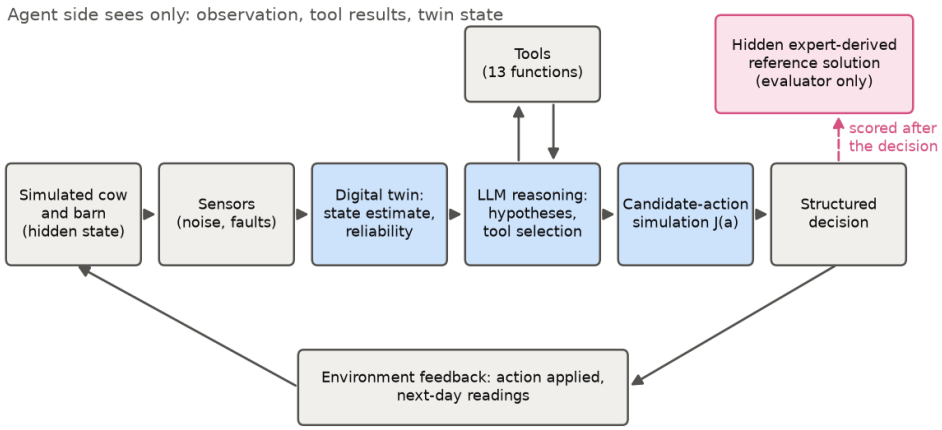

Figure 1: Benchmark architecture. The hidden reference solution is used only for scoring

2.2 Simulated Data

Each scenario represented one Holstein-Friesian cow observed over 14 days at daily intervals. The generator used seed 20260913. Hidden severities of mastitis, ketosis, lameness, heat load, feed-related problems and water-related problems affected seven sensor channels through an explicit effect table. Heat load followed the temperature–humidity index (THI); the heat-stress rule combined THI ≥ 72 with core temperature ≥ 39.2 °C (West, 2003).

Sensor readings included noise and, where scheduled, flat-line or implausible failures, calibration bias or dropout. Alerts compared current values with the preceding seven-day mean. Actions changed severities through an efficacy table, with next-day readings returned as feedback. Parameters were hand-set, not calibrated to a real herd; no real animal data were used. Table 2 describes the observations at decision time. Alerts occurred in 74.4% of scenarios, and 0.29% of historical sensor values were missing.

Table 2: Descriptive statistics of the simulated benchmark data at decision time (500 scenarios)

Variable	n	Mean	SD	Min	Median	Max	Missing %
Milk yield (kg/d)	500	18.05	3.65	10.27	17.98	27.03	0.0
Milk conductivity (mS/cm)	489	5.11	0.73	0.40	5.00	7.09	2.2
Rumination (min/d)	487	434	61	0	442	574	2.6
Activity (steps/d)	484	2385	482	0	2403	3557	3.2
Reticulorumen temperature (°C)	493	38.80	0.71	31.20	38.74	40.39	1.4
Dry-matter intake (kg/d)	500	16.53	1.78	8.99	16.63	20.81	0.0
Water intake (L/d)	500	97.58	15.42	57.00	96.25	150.32	0.0
Barn THI	500	62.31	8.27	48.01	63.06	84.91	0.0
Days in milk	500	135	82	5	138	280	0.0
Parity	500	2.94	1.46	1.00	3.00	5.00	0.0
Alerts per scenario	500	1.22	1.13	0.00	1.00	6.00	0.0
Required actions per scenario	500	1.13	0.40	1.00	1.00	3.00	0.0
Reference sequence length (steps)	500	5.07	1.40	2.00	5.00	9.00	0.0

2.3 Scenarios and Expert-derived Reference Solutions

The 500 scenarios covered 13 classes (38–39 each) and three difficulty levels, varying severity, noise, onset, confounders and cue explicitness (Table 3). Including false alerts, healthy cows and no-action cases ensured that success did not always mean intervention.

Author-written deterministic rules, not an LLM, generated the expert-derived references: confirm before treatment, involve a veterinarian for clinical disease, do not initiate prescription-only medicines, resolve data problems before acting, and allow no intervention when justified. These rules were not independently validated by veterinarians.

Scoring considered the final recommendation, proposed sequence and actions already applied. Decisions were UNSAFE for unsafe actions or failure to address a problem requiring a response. Otherwise, they were CORRECT when all required actions were included and the final action was required or acceptable; ACCEPTABLE when only the latter held; PARTIALLY CORRECT when actions overlapped the reference or the diagnosis was correct; and INCORRECT otherwise. The primary correct action rate combined correct and acceptable decisions.

Order-aware step F1, based on the longest common subsequence, allowed partial credit for omitted steps. Secondary measures were diagnosis accuracy, safe decisions, sensor-fault balanced accuracy, correct no-intervention decisions, and cost and welfare penalties simulated over three days from the hidden state. Physical inspection counted as intervention.

Table 3: Difficulty (easy/moderate/hard), alerts, mean THI, median DIM and typical required actions by scenario class

Class	E/M/H	Alerted %	THI	DIM	Typical required action(s)
Mastitis	13/13/13	92	60.7	141	CMT milk test
Ketosis	13/13/13	85	59.8	13	Blood BHB test
Lameness	13/13/13	62	61.9	141	Hoof trimming inspection
Heat stress	13/13/13	85	78.0	135	Activate cooling

Class	E/M/H	Alerted %	THI	DIM	Typical required action(s)
Reduced feed intake	13/13/13	67	60.4	148	Adjust ration check feed
Milk decline	13/13/13	82	61.1	182	Check water supply
Sensor failure	13/13/12	76	60.4	131	Recalibrate or repair sensor
Sensor bias	13/13/12	53	59.8	164	Recalibrate or repair sensor
Sensor dropout	13/13/12	100	60.8	165	Recalibrate or repair sensor
False alert	13/13/12	100	59.6	126	No intervention
Ambiguous condition	13/13/12	58	61.2	106	Physical inspection
Multiple concurrent	13/13/12	97	66.9	66	Hoof trimming inspection, activate cooling
Healthy / no action	13/13/12	11	59.2	174	No intervention

2.4 Compared Systems and Language-model Configuration

Figure 2 shows the conceptual connection between sensor observations, the animal representation and decision support. In this benchmark, both the animal and incoming data were simulated.

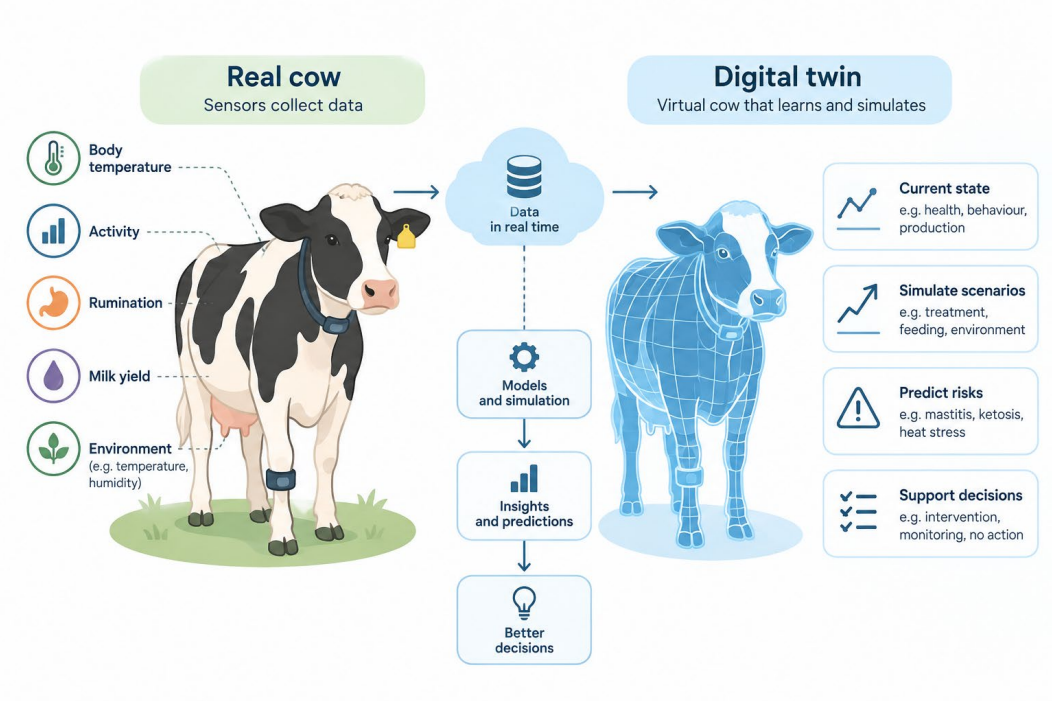

Figure 2: Conceptual representation of a dairy-cow digital twin linking sensor observations, the virtual animal, simulation and decision support. This benchmark implements the workflow entirely in simulation

Six systems were compared (Table 4). The rule-based DSS (M1) applied thresholds to current readings and mapped triggered rules to a standard protocol. The ML DSS (M2) used a gradient-boosting classifier with 65 engineered features and the same protocol. It was trained on 780 separately generated scenarios, with five-fold accuracy of 82.3%. The deterministic DT planner (M6) used the twin, tools, candidate-action simulation and feedback, following a fixed policy established on the training split.

The LLM systems shared the model, system prompt and action vocabularies. The one-shot model (M3) answered from the initial observation in one request. ReAct (M4) could make up to eight calls to 13 tools: cow, milk, rumination, temperature and conductivity histories; THI; mastitis, ketosis and lameness risk estimates; sensor quality and consistency; physical inspection; and intervention cost.

The DT agent (M5) followed a longer cycle. It updated the twin, identified abnormalities and hypotheses, and selected tools to resolve missing information. It then generated candidate actions, compared simulated

consequences and applied a selected action. Next-day feedback informed reconsideration of the decision and another twin update.

The twin and its simulator were deterministic. The twin retained baselines, deviations, sensor reliability, THI, severity estimates, a hypothesis distribution and farm constraints. Candidate actions were scored using:

$$J(a) = 1.0H + 0.5P - 0.4C - 0.8W - 0.6U$$

Here, H and P represent health and production benefits, C is cost, W is the welfare penalty and U is uncertainty risk. The fixed weights governed the simulated comparison of candidate actions.

Table 4: Information access of the compared systems

System	Information	Tools	Simulation	Feedback	Model
M1 Rule-based DSS	Today vs 7-day mean	–	–	–	–
M2 ML DSS	History features	–	–	–	–
M6 DT planner	Twin, history, sensors, THI	≤ 8	Yes	Yes	None
M3 One-shot LLM	Today's observation	–	–	–	gpt-5.4-mini
M4 ReAct agent	Observation + tool results	≤ 8	–	–	gpt-5.4-mini
M5 DT agent	Twin, history, sensors, THI	≤ 8	Yes	Yes	gpt-5.4-mini

All LLM requests used gpt-5.4-mini through the OpenAI Chat Completions API and official Python SDK 3.13.0. The configuration was temperature 0, reasoning effort 'none', a maximum of 1500 output tokens per request, three retries and Pydantic-validated structured outputs. The prompt version was v1.1-2026-09-13. Each run recorded the model and prompt version, token use, tool calls, latency, estimated cost and a complete decision trace. Cost estimates used the study's stated rates of USD 0.75 per million input tokens and USD 4.5 per million output tokens.

2.5 Experiments and Statistical Analysis

The main comparison was complemented by sensor-robustness tests and component ablations restricted to non-LLM systems (Table 5). A separate 13-scenario pilot checked prompts and scoring and was excluded from the reported results.

Table 5: Experimental design

Experiment	Systems	Scenarios	Conditions
Main comparison	M1–M6	500 (M5: 300); paired n = 300	1 run per scenario
Sensor robustness	M1, M2, M6	130 stratified	5–30% corruption; dropout, bias, delay, contradictory
Component ablation	M6	500	Without history, sensors, environment, simulation or tools
Pilot (not reported)	M3–M5	13	Prompt and scoring check

The API budget was exhausted after the one-shot model and ReAct completed 500 scenarios and the DT agent completed 300. The paired analysis used the 300 cases completed by all systems. Identifiers were randomly permuted, so this subset was not deliberately selected by class; it contained 17–27 cases per class. After the pilot, the DT feedback prompt was clarified and the omission rule was corrected for all systems so that monitoring after a required intervention was not counted as an omission.

Binary outcomes were compared with exact McNemar tests (McNemar, 1947). The ordinal decision score, assigned as 1, 0.75, 0.4, 0 and 0 across the five decision categories, was analysed with a Friedman test and pairwise Wilcoxon signed-rank tests (Friedman, 1937; Wilcoxon, 1945). Holm correction was applied within each test family (Holm, 1979). Percentile bootstrap confidence intervals were calculated over scenarios using 2000 resamples (Efron and Tibshirani, 1994).

3. Results

3.1 Overall Performance

Giving the language model access to additional evidence improved action selection relative to answering from the initial observation alone (Table 6). The correct action rate increased from 73.7% for the one-shot model to 82.3% for ReAct and 85.7% for the DT agent. The DT agent outperformed the one-shot model and the deterministic planner on this endpoint. Its numerical advantage over ReAct, however, was not statistically significant, although the ordinal decision score favoured the DT agent.

The ML DSS, rather than an LLM-based system, was strongest overall, achieving 94.3% correct actions and the highest diagnosis accuracy within the simulator distribution. Other measures revealed different strengths: the deterministic planner followed the reference sequence most closely, while the one-shot model diagnosed problems less accurately than the rules despite selecting acceptable actions more often. Action selection, diagnosis and trajectory quality therefore did not produce identical rankings.

Table 6: Main results on 300 paired scenarios (proportions; correct action with 95% bootstrap confidence interval)

System	Diagnosis	Correct action	Step F1	Safe	Sensor-fault BA	No-interv.	Cost €	Welfare
Rules (M1)	0.53	0.64 [0.59–0.69]	0.53	0.93	0.72	0.61	18.8	0.69
ML DSS (M2)	0.88	0.94 [0.92–0.97]	0.69	0.99	0.91	0.95	23.8	0.57
DT planner (M6)	0.69	0.79 [0.74–0.84]	0.76	0.99	0.93	0.39	29.3	0.50
One-shot LLM (M3)	0.50	0.74 [0.69–0.79]	0.70	0.95	0.64	0.56	14.6	0.76
ReAct (M4)	0.70	0.82 [0.78–0.86]	0.71	0.98	0.84	0.59	25.7	0.57
DT agent (M5)	0.71	0.86 [0.82–0.90]	0.70	1.00	0.91	0.17	33.6	0.44

BA: balanced accuracy. No-interv.: correct no-intervention rate. Cost and welfare are simulated three-day outcomes, not observed farm expenditure or animal welfare. Physical inspection counts as intervention.

Statistical detail: Friedman $\chi^2 = 186.9$, $p < 0.001$, Kendall's $W = 0.12$. McNemar comparisons: DT agent vs one-shot, $p < 0.001$; vs planner, $p = 0.036$; vs ReAct, $p = 0.387$; ML DSS vs DT agent, $p = 0.002$. DT agent vs ReAct ordinal score: Wilcoxon $p = 0.017$. Pairwise p-values are Holm-corrected.

3.2 Best Solution by Scenario Type

The aggregate ranking concealed clear differences between problem types (Figures 3 and 4). The ML DSS was best or joint-best across almost all classes. Among the LLM systems, the DT agent was strongest in several health and management problems and in distinguishing sensor failure or bias from animal-related abnormalities. ReAct performed better when an alert was false, while the one-shot model led the LLM comparison in ambiguous cases. Sensor dropout and concurrent abnormalities were solved by every system.

Difficulty also separated the LLM approaches. Performance declined more sharply for the one-shot model and ReAct than for the DT agent as scenarios became harder. This pattern suggests a possible benefit from the richer workflow when cues were less straightforward, but it remains descriptive: class sizes were small, and the hard-scenario comparison between the DT agent and ReAct did not remain significant after multiplicity correction.

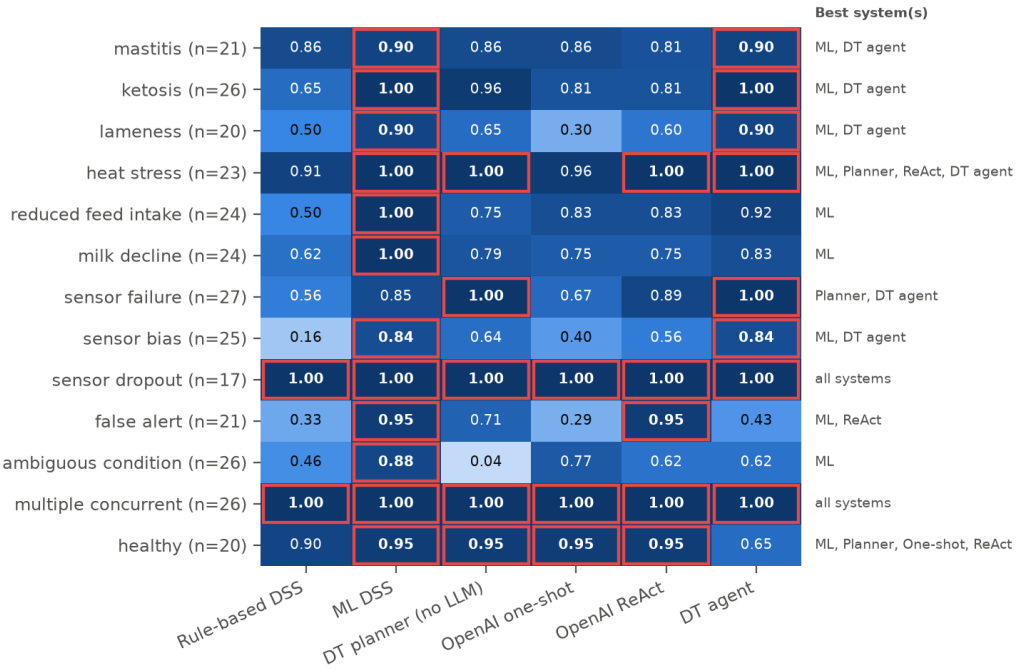

Figure 3: Correct action rate by scenario class. The red frame marks the best system or tied systems

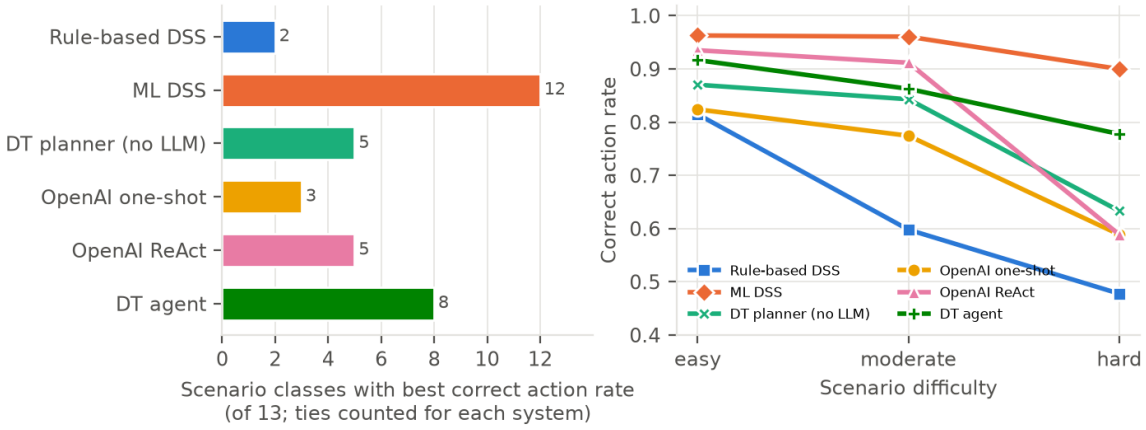

Figure 4: Number of classes in which each system was best or joint-best (left) and correct action rate by difficulty (right)

For hard scenarios, the DT agent–ReAct comparison contained 26 vs 9 discordant pairs (unadjusted p = 0.006; Holm-adjusted p = 0.078). Per-class results are descriptive because each class contained only 17–27 paired scenarios.

3.3 Safety and Unnecessary Intervention

No system proposed an unsafe commission such as antimicrobial treatment. The unsafe decisions that did occur were omissions: a problem requiring attention was left unaddressed. None were observed for the DT agent in the paired sample under the benchmark rules (Figure 5).

Avoiding omissions came with a different weakness. The DT agent often acted when the reference solution called for none, correctly withholding intervention in only 17.1% of these cases. The planner showed a similar, less pronounced tendency, while the ML DSS was much more selective. Because physical inspection counted as intervention, this reflects unnecessary investigation or handling, not evidence of unnecessary treatment. The DT agent had the lowest simulated welfare penalty but the highest simulated cost (Table 6).

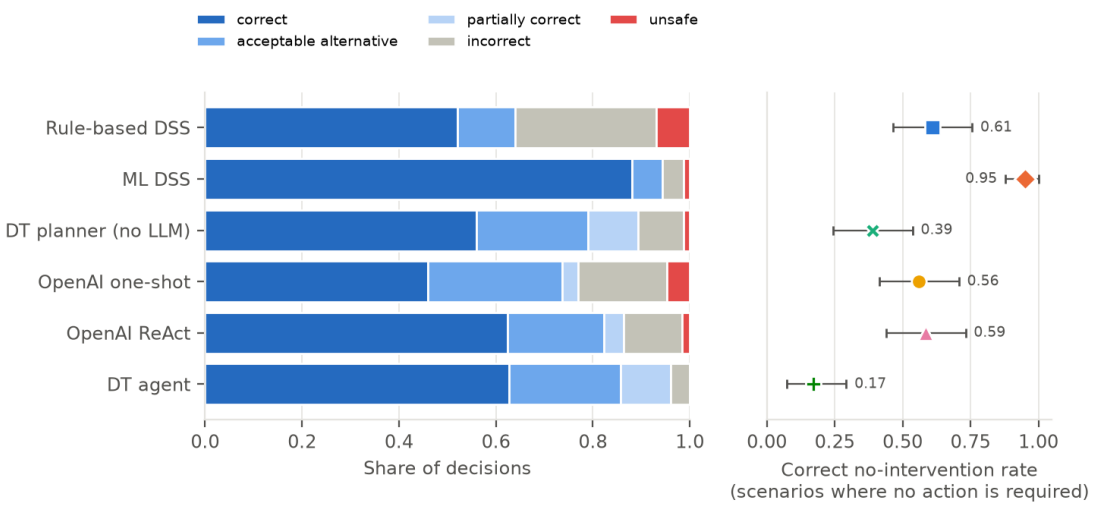

Figure 5: Decision categories (left) and correct no-intervention rate (right)

3.4 Sensor Robustness and Component Ablation

On the original, uncorrupted benchmark observations, sensor-fault identification was strongest for the DT agent among the LLM systems. The additional corruption experiments, which included only the non-LLM systems, showed a less reassuring picture (Figure 6). All three lost accuracy as readings deteriorated. The ML DSS retained the strongest performance, while dropout was particularly disruptive for the rules and the deterministic planner. Under dropout, the planner frequently attributed problems to sensor faults even in scenarios without a sensor fault.

The component ablation helped distinguish useful information from additional processing (Table 7). Removing sensor verification caused the largest decline in the planner's correct action rate, followed by removing environmental context. Historical observations also contributed. Removing action simulation or tool use did not produce a statistically significant loss. These findings concern the deterministic planner and do not establish which components were responsible for the LLM-based DT agent's performance.

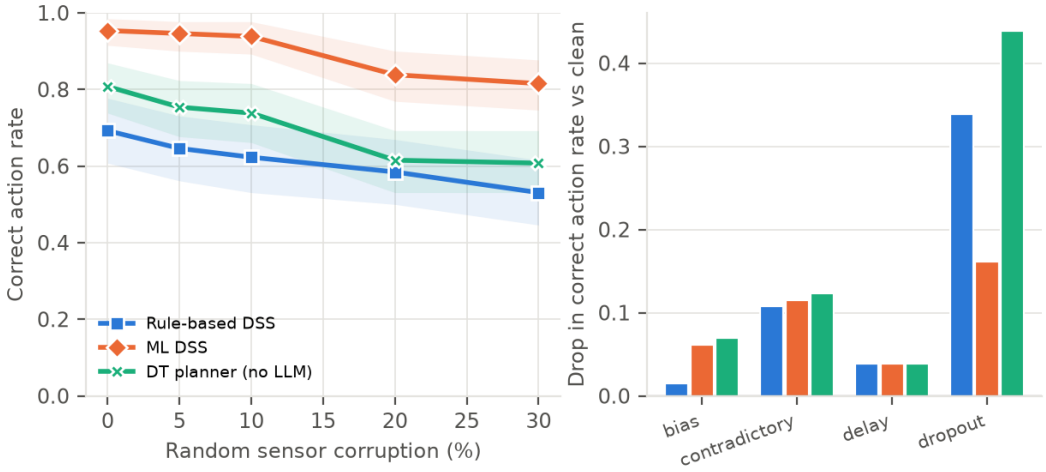

Figure 6: Non-LLM systems under random sensor corruption (left) and decline in correct action rate by fault type (right)

Sensor-fault balanced accuracy before additional corruption: DT agent 91.2%, ReAct 84.4%, one-shot 64.5%. Under 30% random corruption, correct action rates changed from 69% to 53% (rules), 95% to 82% (ML DSS) and 81% to 61% (planner); correct action rates under dropout were 35%, 79% and 37%. The planner assigned a sensor-related cause in 84% of dropout cases without a sensor fault.

Table 7: Component ablation of the digital-twin planner (M6, 500 scenarios)

Variant	Correct action	Diagnosis	Sensor-fault BA	McNemar vs full (Holm)
Full planner	0.78	0.69	0.91	–
Without history	0.72	0.63	0.76	p = 0.002
Without sensor verification	0.59	0.52	0.50	p < 0.001
Without environmental context	0.63	0.55	0.91	p < 0.001
Without action simulation	0.78	0.69	0.91	p = 0.549
Without tool use	0.75	0.63	0.91	p = 0.055

Unrounded correct action rates: full planner 78.2%; without sensor verification 59.2%; without environmental context 63.2%; without history 71.6%; without tool use 75.2%; without simulation 77.6%.

3.5 Environment Feedback, API Calls and Cost

Next-day feedback led the DT agent to revise some decisions, but these revisions produced little net improvement in the correct action rate. The change from the pre-feedback to the final decision was not statistically significant. The selected action also did not always coincide with the highest-scoring simulated candidate, so the agent's recommendation was not simply a direct reading of the simulator's ranking.

The longer workflow increased computational demand (Table 8). The DT agent used more requests and input tokens per scenario than ReAct and cost approximately twice as much at the rates used in this study. The one-shot model was substantially cheaper, although it selected correct actions less often. Figure 7 places these API costs alongside action performance; the simulated management costs in Table 6 are a separate outcome.

Feedback detail: the DT agent changed its action in 14.7% of scenarios. Correct action rates were 84.7% before and 85.7% after feedback (McNemar p = 0.72); the committed action matched the highest-J candidate in 75% of scenarios. Mean DT input use was 21,734 tokens per scenario. Its API cost was approximately ten times that of the one-shot model.

Table 8: OpenAI API use in the main experiment (gpt-5.4-mini)

System	Scen.	Requests	Req./scen.	Tokens in / out	Tool calls	Latency s	USD / 1000 scen.	USD total
One-shot LLM (M3)	500	500	1.0	0.82 / 0.16	0	2.9	2.67	1.33
ReAct (M4)	500	2,722	5.4	7.70 / 0.19	2,222	11.4	13.26	6.65
DT agent (M5)	300	2,123	7.1	6.52 / 0.67	1,223	15.2	26.38	7.92
Total		5,345		15.04 / 1.03	3,445			15.89

Tokens are in millions. Latency is summed API time per scenario. API costs are estimates calculated using the rates stated in Section 2.4.

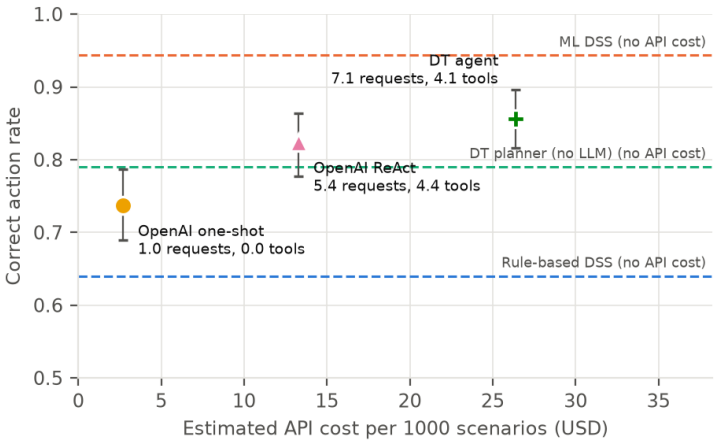

Figure 7: Estimated API cost versus correct action rate. Dashed lines indicate systems without API cost

4. Discussion

The findings support using a language model as part of an evidence-gathering workflow rather than expecting it to resolve a farm problem from a single observation. They do not, however, show that the most elaborate architecture is necessarily the best. The DT agent improved on the one-shot model, but its advantage over ReAct depended on the outcome measure. A conventional ML DSS remained strongest on the primary endpoint. Because it was trained and tested within the same simulator distribution, this result demonstrates a strong in-distribution baseline rather than established effectiveness on commercial farms.

The distinction between representing an animal and deciding what to do with that representation is central here. Behavioural prediction and recognition address important parts of twin construction (Han et al., 2022; Zhang et al., 2025). This benchmark asks what follows: whether the evidence supports a test, a management response, a sensor check or no action. Evaluating that process reveals differences that diagnosis accuracy or the final recommendation alone would miss.

Reliable evidence appeared to matter more than simply adding simulation. The planner depended strongly on sensor verification and environmental context, whereas removing action simulation did not measurably reduce its performance. This finding is consistent with the attention given to data quality in dairy sensor systems (Rutten et al., 2013). It should nevertheless be interpreted narrowly. The ablation tested the deterministic planner, not the LLM agent, and the main comparison changed information access and workflow together. The present experiments therefore cannot isolate a single mechanism behind the DT agent's gains.

The most important practical weakness was the tendency to act when no intervention was needed. Under the benchmark rules, the DT agent avoided unsafe omissions but did not reliably distinguish a justified investigation from unnecessary work. Counting physical inspection as intervention makes the no-intervention measure demanding, yet it also exposes a genuine design question: when should an agent stop investigating and continue monitoring? A system can be cautious about missed problems while still being poorly calibrated about the burden of additional checks.

Feedback and API use reinforce that concern. The longer DT workflow prompted revisions without a clear aggregate improvement and required more computation than ReAct. The design priority emerging from this benchmark is therefore not simply a longer reasoning loop, but better decisions about which evidence is needed, when an action is proportionate and when further activity adds little value.

4.1 Limitations

All scenarios were synthetic, model parameters were uncalibrated and the twin shared the simulator's structure. The author-written reference rules were not independently validated by veterinarians, so agreement with them should not be interpreted as clinical correctness. The paired analysis was limited to 300 scenarios, one language model, one prompt version and one run per scenario. Sensor-corruption testing and component ablation did not cover the LLM agents, and the small per-class samples limit conclusions about individual conditions. The study establishes performance in a controlled benchmark, not readiness for farm deployment.

4.2 Future Work

The next step is to validate and extend the reference rules with veterinarians and farm advisors, then evaluate the workflow using anonymised sensor records from commercial herds. Calibration of no-intervention decisions should explicitly consider the labour and handling burden of further checks. Repeated runs, sensor-corruption tests and component ablations are also needed for the LLM agents, alongside comparisons with other models, including locally hosted ones. Extending the environment to herd-level and multi-day decisions would allow the benchmark to address management problems beyond a single cow and decision episode.

5. Conclusion

This study introduced a blinded benchmark for evaluating how decision-support systems move from observations to actions in a simulated dairy-farm digital twin. Tools and twin-based reasoning improved the language model's decisions relative to a one-shot response, but additional complexity did not establish a clear advantage over ReAct on the primary endpoint, and the in-distribution ML DSS remained strongest overall. The central lesson is that solving a problem also includes recognising when no intervention is justified. An agent that avoids missed problems but repeatedly recommends unnecessary checks is not yet a well-calibrated decision aid. The benchmark provides a controlled way to examine these trade-offs; independent reference validation and evaluation with real farm data remain necessary before claims about practical use can be made.

Ethics Declaration: This study did not involve animals, human participants or personal data. All scenarios, sensor readings and outcomes were generated by computer simulation, and no ethical clearance was required.

AI Declaration: The OpenAI model gpt-5.4-mini was evaluated as part of the experiment. Claude assisted with benchmark implementation and manuscript preparation, and ChatGPT assisted with language editing. Numerical results were calculated by the software from stored experiment outputs. The authors take responsibility for the final manuscript.

References

Berckmans, D. (2017) "General introduction to precision livestock farming", Animal Frontiers, 7(1), pp. 6–11. Available at: https://doi.org/10.2527/af.2017.0102.

Brown-Brandl, T. and Tao, J. (2025) "ASAS-NANP symposium: mathematical modeling in animal nutrition: harnessing real-time data and digital twins for precision livestock farming", Journal of Animal Science, p. skaf138. Available at: https://doi.org/10.1093/jas/skaf138.

Efron, B. and Tibshirani, R.J. (1994) An Introduction to the Bootstrap. New York: Chapman and Hall/CRC. Available at: https://doi.org/10.1201/9780429246593.

Friedman, M. (1937) "The use of ranks to avoid the assumption of normality implicit in the analysis of variance", Journal of the American Statistical Association, 32(200), pp. 675–701. Available at: https://doi.org/10.1080/01621459.1937.10503522.

Grieves, M. and Vickers, J. (2017) "Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems", Transdisciplinary Perspectives on Complex Systems. Cham: Springer, pp. 85–113. Available at: https://doi.org/10.1007/978-3-319-38756-7_4.

Gupta, A. et al. (2024) "Digital twin-driven teat localization and shape identification for dairy cow (student abstract)", Proceedings of the AAAI Conference on Artificial Intelligence, 38(21), pp. 23510–23511. Available at: https://doi.org/10.1609/aaai.v38i21.30450.

Han, X. et al. (2022) "AI based digital twin model for cattle caring", Sensors, 22(19), p. 7118. Available at: https://doi.org/10.3390/s22197118.

Holm, S. (1979) "A simple sequentially rejective multiple test procedure", Scandinavian Journal of Statistics, 6(2), pp. 65–70.

Jeong, D. et al. (2023) "Digital twin application: making a virtual pig house toward digital livestock farming", IEEE Access, 11, pp. 121592–121602. Available at: https://doi.org/10.1109/ACCESS.2023.3313618.

Jo, S. et al. (2018) "Smart livestock farms using digital twin: feasibility study", 2018 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1461–1463. Available at: https://doi.org/10.1109/ICTC.2018.8539516.

Liu, X. et al. (2023) "AgentBench: evaluating LLMs as agents", arXiv. Available at: https://doi.org/10.48550/arXiv.2308.03688.

McNemar, Q. (1947) "Note on the sampling error of the difference between correlated proportions or percentages", Psychometrika, 12(2), pp. 153–157. Available at: https://doi.org/10.1007/BF02295996.

Neethirajan, S. and Kemp, B. (2021) "Digital twins in livestock farming", Animals, 11(4), p. 1008. Available at: https://doi.org/10.3390/ani11041008.

Norton, T. et al. (2019) "Review: Precision livestock farming: building 'digital representations' to bring the animals closer to the farmer", Animal, 13(12), pp. 3009–3017. Available at: https://doi.org/10.1017/S175173111900199X.

Purcell, W. and Neubauer, T. (2023) "Digital twins in agriculture: a state-of-the-art review", Smart Agricultural Technology, 3, p. 100094. Available at: https://doi.org/10.1016/j.atech.2022.100094.

Pylianidis, C., Osinga, S. and Athanasiadis, I.N. (2021) "Introducing digital twins to agriculture", Computers and Electronics in Agriculture, 184, p. 105942. Available at: https://doi.org/10.1016/j.compag.2020.105942.

Rutten, C.J. et al. (2013) "Invited review: sensors to support health management on dairy farms", Journal of Dairy Science, 96(4), pp. 1928–1952. Available at: https://doi.org/10.3168/jds.2012-6107.

Tao, F. et al. (2019) "Digital twin in industry: state-of-the-art", IEEE Transactions on Industrial Informatics, 15(4), pp. 2405–2415. Available at: https://doi.org/10.1109/TII.2018.2873186.

Verdouw, C. et al. (2021) "Digital twins in smart farming", Agricultural Systems, 189, p. 103046. Available at: https://doi.org/10.1016/j.agsy.2020.103046.

Wang, L. et al. (2024) "A survey on large language model based autonomous agents", Frontiers of Computer Science, 18(6), p. 186345. Available at: https://doi.org/10.1007/s11704-024-40231-1.

West, J.W. (2003) "Effects of heat-stress on production in dairy cattle", Journal of Dairy Science, 86(6), pp. 2131–2144. Available at: https://doi.org/10.3168/jds.S0022-0302(03)73803-X.

Wilcoxon, F. (1945) "Individual comparisons by ranking methods", Biometrics Bulletin, 1(6), pp. 80–83. Available at: https://doi.org/10.2307/3001968.

Yao, S. et al. (2023) "ReAct: synergizing reasoning and acting in language models", International Conference on Learning Representations (ICLR 2023). Available at: https://doi.org/10.48550/arXiv.2210.03629.

Zhang, Y. et al. (2025) “Multimodal behavior recognition for dairy cow digital twin construction under incomplete modalities: a modality mapping completion network approach”, Artificial Intelligence in Agriculture, 15(3), pp. 459–469. Available at: https://doi.org/10.1016/j.aiia.2025.04.005.