SIMUVERITY: BENCHMARKING AGENTS FOR ENGINEERING-GRADE SIMULINK MODEL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing SIMULINK benchmarks mainly evaluate whether generated models compile, execute, or resemble a reference model. These criteria do not establish whether a model satisfies its engineering requirements. We introduce SIMUVERITY, a benchmark of 101 text-to-executable SIMULINK model-generation tasks across ten engineering domains. For each task, executable-system profiles ground the engineering specification and four families of native simulation scenarios. A hierarchical evaluator first checks artifact delivery, native executability, and engineering qualification, then scores qualified models across six dimensions covering accuracy, output quality, mechanistic fidelity, control and causal integrity, operating-domain robustness, and dynamic response. We evaluate six agent systems with SIMUVERITY. The best system achieves an overall score of only 42.86. The results show that structural similarity is a poor proxy for engineering performance: capability bottlenecks arise both in producing qualified implementations and in satisfying multidimensional requirements after qualification. Meanwhile, some high-scoring models still exhibit severe visual-layout disorder. SIMUVERITY provides a systematic basis for assessing agents' engineering capabilities and diagnosing failures in executable SIMULINK model generation.

1 INTRODUCTION

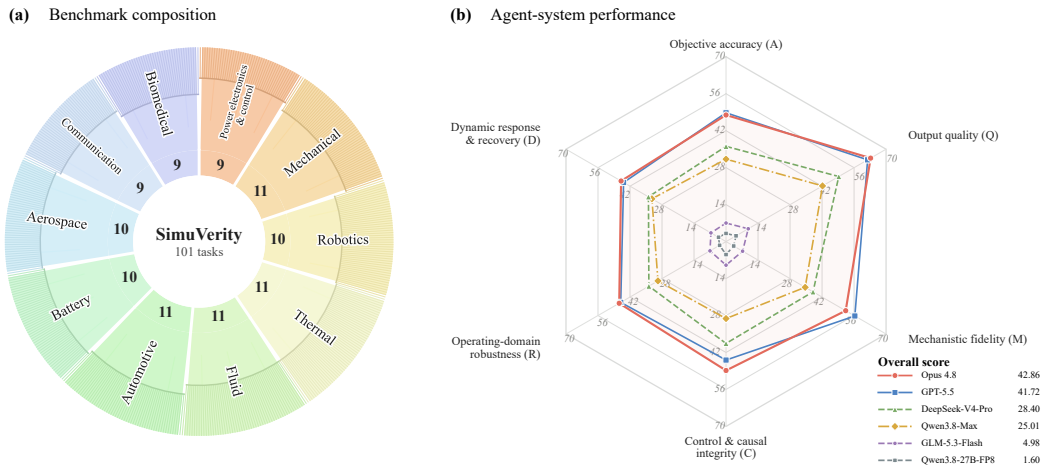

Figure 1: **Task composition and agent performance in SIMUVERITY.** (a) Task distribution across ten engineering domains; outer-ring line lengths indicate mean domain scores. (b) Six-dimensional performance and overall scores of six agents.

SIMULINK is a powerful simulation and Model-Based Design environment for system modeling, dynamic analysis, control design, and verification, with broad use across industrial engineering domains (Shrestha et al., 2022; MathWorks, 2026b). Recent LLM-based agents can translate natural-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

language requirements into executable `.slx` models, run simulations, analyze results, and iteratively refine their implementations (Abdalla et al., 2026; Liang & Zhao, 2026). Therefore, as engineering practice demands high system accuracy, performance, and robustness, an important question is how to benchmark whether these agents produce models that truly satisfy engineering requirements rather than merely execute.

However, existing benchmarks have not centered the core engineering question of whether a generated model satisfies its requirements across prescribed operating conditions. They primarily rely on two types of criteria: (i) artifact delivery and technical execution, and (ii) structural and functional similarity to a reference system (Shrestha & Csallner, 2021; Ren et al., 2025; Liang & Zhao, 2026). But the former only verifies that a model exists and runs, while the latter treats one reference implementation as the standard despite valid structural alternatives and possible errors in reference-aligned models. Some automotive studies improve on these criteria with requirement-level tests, but remain limited to software logic and control functions (Abdalla et al., 2026). A cross-domain benchmark must therefore evaluate candidates against their engineering requirements through task-specific native simulation scenarios and multidimensional performance criteria over the prescribed operating domain.

To address this gap, we introduce SIMUVERITY, an engineering-grade native-simulation benchmark with 101 tasks across ten engineering domains. For each task, domain experts build an executable-system profile from native runs of a selected, reconstructed, or independently implemented reference system. The profile captures system boundaries, interfaces, engineering relations, operating ranges, and dynamics, grounding both the natural-language specification and four families of task-specific simulation scenarios. Evaluation first checks artifact delivery, technical executability, and engineering qualification, then scores qualified candidates across A/Q/M/C/R/D. This design supports multidimensional diagnosis, valid structural diversity, and detection of reference-aligned models with poor engineering performance.

We evaluate six agent systems with SIMUVERITY. The highest-scoring system reaches only 42.86 overall, indicating limited capability on multi-domain SIMULINK engineering tasks. Its artifact-delivery, executability, and engineering-qualification rates are 94.72%, 86.47%, and 76.90%, respectively. Thus, some delivered and executable candidates do not form qualified engineering implementations, and shallow success cannot replace multidimensional evaluation. We further analyze potential deficiencies in the generated models. Experts classify eight of the 31 candidates from the highest-scoring agent system with engineering scores of at least 70 as severely disordered, showing that agents often overlook readability despite strong engineering performance. Current agents therefore lack effective cross-domain SIMULINK generation capability and require a closed loop of specification-driven modeling, native simulation validation, and multi-condition feedback correction.

In conclusion, our main contributions are:

(I) **Benchmark curation.** We introduce SIMUVERITY, a benchmark of 101 text-to-executable SIMULINK model-generation tasks across ten engineering domains. Each task includes a reconstructed or independently built reference system, an executable-system profile, and a profile-grounded engineering prompt.

(II) **Evaluation framework.** We design a hierarchical multidimensional performance evaluation framework comprising four families of task-specific native simulation scenarios and six performance dimensions. It enables hierarchical assessment and multidimensional diagnosis of candidate engineering performance.

(III) **Empirical findings.** We evaluate six agent systems and analyze the performance and limitations of their generated models from multiple engineering perspectives, laying an empirical foundation for future improvements in SIMULINK model-generation performance.

2 RELATED WORK

Benchmarks for executable engineering artifacts. Domain-specific benchmarks increasingly evaluate complete executable engineering artifacts through native execution and requirement-level evidence, moving beyond artifact formation and basic validity checks (Chen et al., 2025; Guo et al., 2025; Cui et al., 2026; Qin et al., 2026; Dong et al., 2026; Singh et al., 2026; Chen et al., 2021;

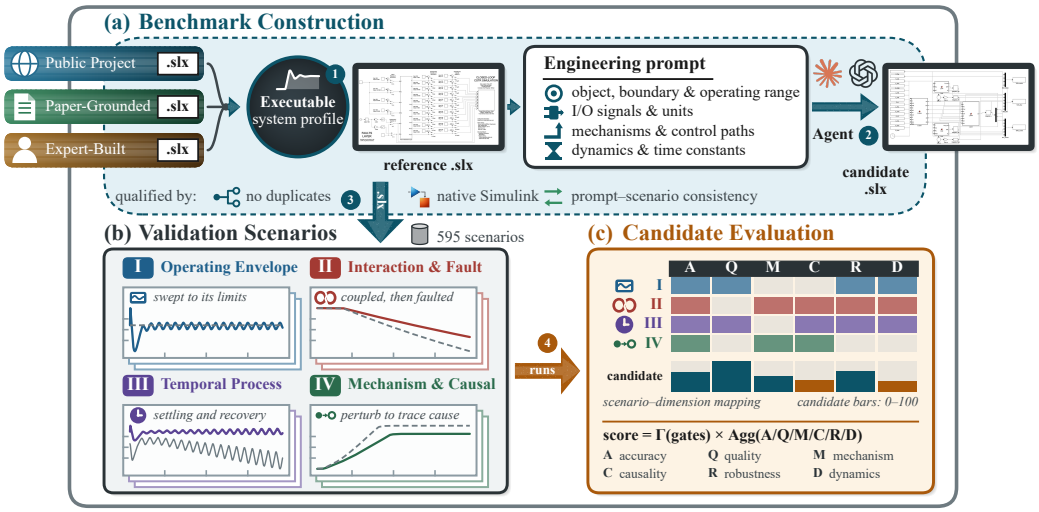

Figure 2: **The SIMUVERITY pipeline.** Benchmark construction, native simulation testing, and six-dimensional performance evaluation.

Liu et al., 2023; Lu et al., 2024). This progression raises the corresponding question for executable dynamical systems: how should native execution evidence support a hierarchical, multidimensional assessment of engineering compliance?

Agentic SIMULINK model generation and evaluation. Existing SIMULINK studies target toolchain fault discovery (SLGPT), block-diagram reconstruction with basic execution checks (SimuGen), or automotive control validation with boundary and time-series tests (ASWE-Bench) (Shrestha & Csallner, 2021; Ren et al., 2025; Abdalla et al., 2026; Kevian et al., 2024; Guo et al., 2024). SimuBench extends text-to-model generation across multiple domains, but relies mainly on artifact completeness, basic executability, and structural or functional similarity to a reference model (Liang & Zhao, 2026). Such criteria can miss structurally distinct valid implementations and engineering failures in reference-aligned candidates. Native MATLAB/SIMULINK access for agents further broadens the space of valid implementations (MathWorks, 2026a;c). SIMUVERITY addresses these gaps with executable-system-profile-grounded simulation scenarios and hierarchical multidimensional evaluation. Detailed comparisons appear in Appendix A.

3 SIMUVERITY

3.1 TASK DEFINITION AND CROSS-DOMAIN COVERAGE

Task Definition. We formulate text-to-executable SIMULINK model generation as a build-to-specification task. Given a natural-language engineering specification defining the system, interfaces, operating conditions, functional objectives, mechanisms, and dynamics, a system must deliver a complete native `.slx` model. Block choices, diagram layout, and subsystem hierarchy remain unrestricted, allowing diverse valid implementations.

Cross-Domain Coverage. SIMUVERITY comprises 101 tasks spanning ten engineering domains: power electronics and control, mechanical systems, robotics, thermal systems, fluid systems, automotive systems, battery systems, aerospace systems, communication systems, and biomedical systems. Each domain contributes 9–11 tasks, yielding a near-balanced cross-domain composition (Table 1).

3.2 DATA CURATION

Reference-System Curation and Reconstruction. We collect candidate reference systems from public examples and engineering projects, research artifacts accompanying papers, and expert-built

Table 1: **Task distribution across the ten engineering domains of SIMUVERITY.**

Engineering domains (101 tasks)			
Power electronics & control	9	Automotive systems	11
Mechanical systems	11	Battery systems	10
Robotics	10	Aerospace systems	10
Thermal systems	11	Communication systems	9
Fluid systems	11	Biomedical systems	9

models. We retain systems that can be executed in SIMULINK, represent meaningful engineering behavior, and broaden the benchmark's coverage. Each selected system is then reconstructed into a reproducible evaluation target by resolving dependencies and compatibility issues, exposing the signals needed for control and observation, and connecting it to repeatable operating conditions and a task-specific evaluator.

Executable-System Profiling and Prompt Design. Domain experts inspect and execute each reference system under a unified protocol to construct an executable-system profile grounded in observable model facts and reproducible simulation evidence. The profile captures the system boundary, interfaces, mechanisms, control relationships, operating conditions, and dynamic behavior. Based on these profiles, human experts design a unified prompt framework covering the modeled system, deliverable, input–output interfaces, operating range, functional objectives, required mechanisms, control relationships, and dynamic behavior. They use this framework to write a prompt for each task. Each prompt is then checked against its executable-system profile for engineering consistency and native SIMULINK realizability.

Quality Control. We track source lineage and apply same-source deduplication and near-duplicate screening. Task-specific test conditions, scenario combinations, thresholds, and scoring logic are withheld from the evaluated agents during model generation to reduce retrieval- and memorization-based shortcuts. A task enters the formal benchmark only after its reference system completes native loading, diagram updating, compilation, and simulation under a unified toolchain. We then verify that the observed system behavior supports the executable-system profile, the public specification accurately reflects that profile, and the independent evaluator tests the specified requirements through the correct input–output interfaces. The final benchmark contains 101 tasks.

3.3 TASK-SPECIFIC SIMULATION TEST DESIGN

This section translates each task's engineering requirements into executable simulation tests that produce the evidence needed for evaluation.

For each task, domain experts translate the requirements in the public engineering specification and executable-system profile into an executable test suite. Each scenario defines the conditions to control, the external and internal signals to observe, the engineering quantities to compute, and the criteria used to interpret them.

SIMUVERITY organizes these tests into four complementary families:

(i) **Operating-envelope scenarios** test nominal operation, boundary conditions, parameter and load variations, and external disturbances.

(ii) **Interaction and fault scenarios** test joint inputs, compound operating conditions, fault injection, and transitions between normal and faulty states.

(iii) **Temporal-process scenarios** test startup, transients, settling, event order, fault onset and clearance, and recovery.

(iv) **Mechanism and causal-check scenarios** test whether required physical paths, functional mechanisms, feedback loops, control relations, and signal provenance are active and correct.

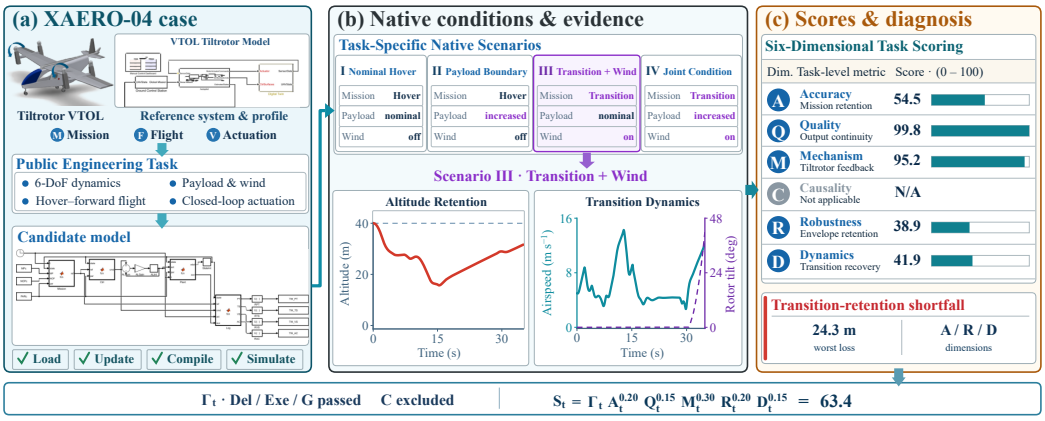

Figure 3: **Evaluation of the closed-loop tiltrotor VTOL model.** Native simulation evidence, six-dimensional scores, and failure diagnosis for the candidate model. Four of six scenarios are shown.

For each applicable scenario, the candidate model is loaded, updated, compiled, and simulated in the native MATLAB/SIMULINK runtime. The resulting execution status, trajectories, engineering quantities, and mechanism checks constitute the evidence used by the evaluation framework in Section 3.4.

3.4 HIERARCHICAL MULTIDIMENSIONAL PERFORMANCE EVALUATION FRAMEWORK

SIMUVERITY first applies three prerequisite gates and then scores qualified candidates across six engineering dimensions. For candidate model x on task t, the complete evaluation is

$$\mathcal{V}_t(x) = \left(r_t^{\mathrm{del}}, r_t^{\mathrm{exe}}, r_t^{G}, \mathbf{v}_t\right).$$

Prerequisite Gates. r_t^{del} checks whether the required `.slx` artifact is delivered, and r_t^{exe} checks whether it completes native loading, diagram updating, compilation, and simulation. The engineering-qualification gate r_t^{G} evaluates only candidates that have been delivered and can execute natively, and returns pass or fail. Its checks are specified separately for each task by human experts based on the task specification and executable-system profile: the required functional roles must be present, the required signal paths must be connected, and the candidate must produce valid evaluation evidence in the formal scenarios according to the prescribed output conventions. Candidates that pass the G gate proceed to the subsequent six-dimensional performance scoring.

Multidimensional Performance. Candidates passing all three gates receive the performance vector

$$\mathbf{v}_t = \left(v_t^{A}, v_t^{Q}, v_t^{M}, v_t^{C}, v_t^{R}, v_t^{D}\right), \qquad \mathcal{P} = \{A, Q, M, C, R, D\}.$$

The six dimensions measure **A**, objective accuracy; **Q**, output quality; **M**, mechanistic fidelity; **C**, control and causal integrity; **R**, operating-domain robustness; and **D**, dynamic response and recovery.

Evidence Mapping and Scoring. Let $\mathcal{S}_t$ be the scenario set and $\mathcal{K}_t$ the observed engineering quantities for task t. The task-specific relation

$$\mathcal{R}_t \subseteq \mathcal{S}_t \times \mathcal{K}_t \times \mathcal{P}$$

maps each quantity k observed under scenario s to a performance dimension d. This mapping is many-to-many: one scenario may support several dimensions, and one dimension may combine evidence from several scenarios.

Each observed quantity $z_{t,s,k}$ is converted to a normalized utility by a task-specific function,

$$u_{t,s,k} = \phi_{t,s,k}(z_{t,s,k}), \qquad u_{t,s,k} \in [0,1].$$

Table 2: **Overall scores and prerequisite pass rates.** Agent results are three-run means; Reference reports the mean score of 101 task-specific reference systems.

Agent system	Overall	Delivered	Executable	G pass
Opus 4.8 + Claude Code	42.86	94.72%	86.47%	76.90%
GPT-5.5 + Codex	41.72	98.68%	94.72%	75.58%
DeepSeek-V4-Pro + Claude Code	28.40	97.03%	88.12%	61.39%
Qwen3.8-Max + Claude Code	25.01	89.77%	80.86%	53.80%
GLM-5.3-Flash + Claude Code	4.98	96.04%	48.18%	13.20%
Qwen3.8-27B-FP8 + Claude Code	1.60	30.36%	13.53%	6.27%
Reference	96.01	100.00%	100.00%	100.00%

The relevant utilities are then aggregated into each dimension:

$$v_t^d = \mathrm{Agg}_{t,d}(\{u_{t,s,k} \mid (s,k,d) \in \mathcal{R}_t\}) .$$

The resulting dimension scores are reported on a 0–100 scale. The final task score is

$$S_t(x) = \Gamma_t\big(r_t^{\mathrm{del}}, r_t^{\mathrm{exe}}, r_t^{G}\big)\, H_t(\mathbf{v}_t) , \qquad H_t(\mathbf{v}_t) = \prod_{d \in \mathcal{P}_t} \big(v_t^d\big)^{w_{t,d}} ,$$

where $\mathcal{P}_t \subseteq \mathcal{P}$ is the applicable dimension set for task t, $\sum_{d \in \mathcal{P}_t} w_{t,d} = 1$, and Γ_t applies the prerequisite gates and H_t aggregates the six-dimensional performance. When required by the task, worst-condition or bottleneck-sensitive aggregation prevents critical failures from being hidden by strong results on easier conditions. The evaluator returns the prerequisite outcomes, six-dimensional performance, overall score, and the scenarios and quantities responsible for each failure. Figure 3 illustrates this evaluation chain for the closed-loop tiltrotor VTOL task.

Expert Validation of Evaluator Validity. To test whether evaluator scores reflect engineering judgment (Zhu et al., 2025), two domain experts independently rated 30 candidate models generated by Opus 4.8 Run 1. The candidates span all ten engineering domains and were rated on a five-point scale without access to evaluator scores. Inter-rater agreement was high, with weighted Cohen's $\kappa = 0.90$. The Spearman correlation between evaluator scores and mean expert ratings was $\rho = 0.94$, comparable to the inter-expert rank agreement ($\rho = 0.87$). Even when restricted to the 18 candidates with nonzero evaluator total scores, the correlation remained $\rho = 0.88$. These results indicate close agreement between evaluator rankings and domain-expert judgments. Detailed sampling and rank-correlation results appear in Appendix C.4.

4 Experiments and Results

4.1 Experimental Setup

Agent Systems. We evaluate six agent systems: Claude Opus 4.8 (Anthropic, 2026b) with Claude Code (Anthropic, 2026a), GPT-5.5 (OpenAI, 2026b) with Codex (OpenAI, 2026a), DeepSeek-V4-Pro (DeepSeek, 2026) with Claude Code, Qwen3.8-Max (Qwen Team, 2026b) with Claude Code, GLM-5.3-Flash (Z.ai, 2026) with Claude Code, and Qwen3.8-27B-FP8 (Qwen Team, 2026a) with Claude Code. Harness versions, reasoning settings, and context configurations are reported in Appendix D.1.

Common Task and Execution Conditions. All six systems are evaluated on the same 101 tasks across ten engineering domains, using identical public specifications, delivery requirements, task workspaces, and MATLAB/Simulink environments. Each run begins in a clean session with an independent working directory and access to the same core MATLAB MCP tools. These tools connect the agent to the native MATLAB/Simulink runtime for model construction, inspection, and testing (MathWorks, 2026a;c; Yang et al., 2024).

Table 3: **End-to-end and post-G performance on the six engineering dimensions.** Each dimension is averaged over its registered applicable-task set.

Agent system	Scope	A	Q	M	C	R	D
Opus 4.8 + Claude Code	End-to-end	47.95	63.24	52.34	48.82	46.72	45.90
	Post-G	62.28	82.21	65.25	58.92	60.73	58.65
GPT-5.5 + Codex	End-to-end	48.80	61.94	56.34	44.87	46.05	44.81
	Post-G	64.57	81.96	72.43	53.62	60.92	59.07
DeepSeek-V4-Pro + Claude Code	End-to-end	36.08	49.31	38.05	38.57	33.75	33.93
	Post-G	58.78	80.32	59.32	57.59	55.07	54.50
Qwen3.8-Max + Claude Code	End-to-end	31.24	42.20	34.62	29.15	29.77	32.41
	Post-G	58.03	78.46	60.57	48.47	55.40	59.14
GLM-5.3-Flash + Claude Code	End-to-end	7.03	9.72	7.25	8.92	7.02	6.60
	Post-G	54.23	75.01	50.62	55.33	53.45	52.14
Qwen3.8-27B-FP8 + Claude Code	End-to-end	3.09	4.38	3.34	4.91	2.81	3.28
	Post-G	49.51	68.50	51.81	57.25	45.14	51.00

Execution Environment and External-Resource Isolation. The execution environment is isolated from the outside: the example-model directories of the MATLAB installation, the local reference-model directories, and the scenario and scoring-script directories are invisible to the agent.

Evaluation Protocol. We impose common stopping conditions and delivery rules on all systems (Kapoor et al., 2024). Each task–system pair is run three times from a fresh session and directory, with a separate run identifier. The final `.slx` artifact from each run is evaluated using the same frozen task-specific scorer. We average the three runs to obtain the task–system score, then average across tasks for system-level and domain-level results; each A/Q/M/C/R/D dimension is averaged over its registered applicable tasks. Non-delivery, technical non-execution, and candidate-caused runtime failures count as failures. Errors attributable to the API, evaluator, or execution environment are classified as infrastructure failures and rerun under a new identifier.

4.2 Overall Performance

Overall System Performance. Table 2 summarizes the overall score and three prerequisite outcomes for all six agent systems. Claude Opus 4.8, GPT-5.5, DeepSeek-V4-Pro, Qwen3.8-Max, GLM-5.3-Flash, and Qwen3.8-27B-FP8 obtain overall scores of 42.86, 41.72, 28.40, 25.01, 4.98, and 1.60, respectively. Even the highest-scoring agent system remains far from fully satisfying the engineering requirements across the benchmark. Task-level 95% CIs are given in Table 13.

Prerequisite Gate Outcomes. Across all task–system runs, artifact delivery, native executability, and G-pass rates are 84.43%, 68.65%, and 47.85%, respectively. Thus, delivery and executability alone do not guarantee a qualified engineering implementation.

Multidimensional Performance Profiles. Table 3 reports end-to-end and post-G performance across A/Q/M/C/R/D. End-to-end results capture the full generation pipeline: a run that fails the G gate contributes zero to every applicable dimension. Post-G results isolate the performance of qualified engineering implementations by averaging the G-passing candidates within each run. Averaged equally across the six systems, end-to-end A/Q/M/C/R/D scores are 29.03, 38.46, 31.99, 29.21, 27.69, and 27.82, respectively. Output quality is the highest, and the other five dimensions are lower, with operating-domain robustness (R) and dynamic response and recovery (D) the lowest.

4.3 Structural Similarity and Engineering Performance

Section 3.4 specifies that candidate performance is adjudicated by the hierarchical multidimensional performance framework rather than scored by similarity to a reference system. We test whether

Table 4: **Structural relation and engineering performance.** Results cover 95 delivered Opus Run 1 candidates; post-G means include only candidates that pass engineering qualification.

Structural relation	n	G pass	End-to-end	Post-G	Top quartile	Zero
Reference-Aligned	39	87.18%	52.13	59.80	12	7
Partially Similar	43	83.72%	44.05	52.61	9	10
Structurally Distinct	13	69.23%	42.37	61.20	3	4

candidate–reference structural similarity can serve as a reliable criterion for engineering performance. Domain experts classify the 95 formally delivered Opus candidates from Run 1 into three mutually exclusive structural relations—**Reference-Aligned**, **Partially Similar**, and **Structurally Distinct**—using block choices and abstraction level, system topology, subsystem decomposition, and signal and feedback routing. The groups contain 39, 43, and 13 candidates, respectively; 56/95 (58.95%) are Partially Similar or Structurally Distinct.

Table 4 compares the groups. Their end-to-end means are 52.13, 44.05, and 42.37, while their post-G means are 59.80, 52.61, and 61.20 for Reference-Aligned, Partially Similar, and Structurally Distinct models, respectively. Of the 24 top-quartile candidates, 12 are Partially Similar or Structurally Distinct, whereas seven Reference-Aligned candidates receive zero, including two that pass G. Thus, the results show both valid structural diversity and deceptive structural similarity: reference proximity does not reliably order engineering performance.

The representative valid-diversity and deceptive-similarity cases are reported in Appendix E. They present two Structurally Distinct candidates that pass G and score 85.10 and 80.07, and two Reference-Aligned candidates that pass G but score zero.

4.4 Cross-Domain Capability Gaps

We analyze Claude Opus 4.8 with Claude Code across ten engineering domains by jointly considering engineering-qualification pass rates and post-G performance. The full domain profiles are reported in Appendix D (Table 18). The results reveal three capability-gap patterns: qualification-dominated domains such as power electronics and fluid systems; post-qualification bottlenecks in battery and aerospace systems; and mixed bottlenecks in mechanical and communication systems. These patterns show that progress is needed both to form genuine engineering implementations and to satisfy multidimensional requirements after qualification.

4.5 Visual Layout Quality of High-Performing Simulink Models

To examine whether high engineering performance is accompanied by readable diagrams, domain experts inspected high-scoring Run 1 candidates from Claude Opus 4.8 with Claude Code. Among the 31 candidates with engineering scores of at least 70, the experts identified several cases with substantial block overlap, difficult-to-trace signal routes, or poorly organized functional regions; eight were judged severely disordered. We then quantified these cases using block overlap, routing detours, and line-crossing density (defined in Appendix F.1), and produced layout-only reorganizations for direct before-and-after comparison, adjusting only block positions and existing line routes while keeping each model's implementation unchanged.

Across the eight cases, reorganization reduced block overlap to zero in every model, reduced line-crossing density by 22.7–85.1%, and reduced routing-detour ratio in seven cases. Figure 4 shows one representative before-and-after example, the robotic ball-balancing system; the complete measurements, intervention protocol, and remaining cases are provided in Appendix F. These results show that high engineering scores do not imply clear diagram organization.

4.6 Ablations of Native Simulation Feedback and Tool Access

Ablation Setup. To test whether an agent can still complete tasks it already solves once a tool capability is removed, we use Claude Opus 4.8 with Claude Code and select one strong-performing task under Full MCP from each of the ten engineering domains, forming a cross-domain subset.

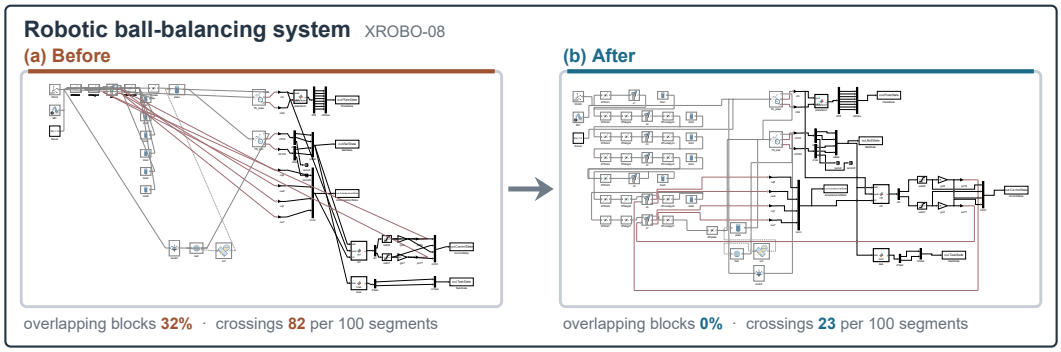

Figure 4: **Robotic ball-balancing system before and after layout optimization.** Only block positions and existing line routes are adjusted; the model implementation remains unchanged.

Table 5: **Ablations of native simulation feedback and tool access.** Results report three-run mean prerequisite pass rates and overall scores for Claude Opus 4.8 with Claude Code on ten cross-domain, high-performing tasks.

Tool condition	Artifact delivery	Native executability	G pass	Overall score
Full MCP	100.00%	100.00%	100.00%	78.46
No-Simulation MCP	53.33%	33.33%	13.33%	7.88
Batch-only	60.00%	50.00%	40.00%	18.23

The three tool conditions form two single-factor contrasts: No-Simulation MCP retains MCP model construction, querying, and compilation but disables simulation execution, differing from Full MCP only in simulation feedback; Batch-only removes the MCP interface, differing from Full MCP in the entire MCP toolset.

Results. Table 5 summarizes the three-run mean prerequisite pass rates and overall engineering scores under the three tool conditions. Full MCP versus No-Simulation MCP: artifact delivery, native executability, and engineering qualification fall from 100.00% to 53.33%, 33.33%, and 13.33%, respectively, and the overall score falls from 78.46 to 7.88; unable to verify its outputs through simulation feedback, the agent scores lower on all ten tasks. Full MCP versus Batch-only: removing MCP lowers the overall score to 18.23, with prerequisite pass rates of 60.00%, 50.00%, and 40.00%, and 8 of 10 tasks score lower.

On these ten tasks that the agent can otherwise complete, both restricted conditions degrade sharply, indicating that simulation feedback and MCP tools are critical for reliably generating executable SIMULINK models. The dimension-level ablation results are reported in Appendix G.

5 CONCLUSION

SIMUVERITY evaluates SIMULINK model generation through native simulation and six engineering dimensions across 101 tasks in ten domains. Among six agent systems, the best scores only 42.86, with bottlenecks in both engineering qualification and subsequent performance. Structural similarity does not reliably reflect engineering quality; models closely aligned with the reference can still fail to satisfy engineering requirements, and high-scoring models may still have disordered layouts. Tool ablations on ten selected tasks show substantial benefits from simulation feedback and MCP access. SIMUVERITY provides a systematic basis for assessing agents' engineering capabilities and diagnosing failures in executable model generation. The benchmark relies on MATLAB/SIMULINK, and its native simulation-based evaluation can be slow; future work will optimize the evaluation pipeline for greater efficiency and scalability.

AI USE STATEMENT

Generative AI tools were used in two supporting roles. First, they assisted literature discovery, including identifying related work, by helping formulate and refine search queries, locate candidate papers, and organize the literature around executable engineering benchmarks, SIMULINK model generation, native tool use, and evaluator validation. The authors independently checked the cited works and determined which sources support the manuscript's claims. Second, the tools assisted manuscript drafting and language polishing by proposing wording, translating and condensing author-written passages, and checking consistency in terminology, section and figure labels, and reference formatting. The authors reviewed and verified every AI-assisted output, including all claims, citations, numerical results, and final wording.

REPRODUCIBILITY STATEMENT

To support reproducibility, the paper and appendix document task construction, scoring methodology, experimental settings and results, repeated-run protocols, and ablation protocols. The supplementary material provides the benchmark repository, which contains the per-task specifications and executable-system profiles for all 101 tasks, the official simulation scenarios, the evaluator and scoring code, and the reference assets or import recipes, together with the results repository, which contains the per-run score tables of all six agent systems and the tool ablation, with their aggregation and validation scripts. The materials are anonymized and will be open-sourced after the review process.

REFERENCES

Abdelrahman Abdalla, Vincent Thie, Joschka Schaub, Markus Eisenbarth, Sung-Yong Lee, and Jakob Andert. Multi-agent software development for automotive model-based graphical programming. *IEEE Access*, 14:115016–115028, 2026. doi: 10.1109/ACCESS.2026.3711515. URL https://doi.org/10.1109/ACCESS.2026.3711515.

Anthropic. Claude Code: Overview. Official documentation, 2026a. URL https://code.claude.com/docs/en/overview. Version 2.1.238 used; accessed September 25, 2026.

Anthropic. Introducing Claude Opus 4.8. Official release announcement, May 2026b. URL https://www.anthropic.com/news/claude-opus-4-8. Released May 28, 2026; accessed September 25, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, pp. 96934–96990, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/f12b4df26344f3be803c06b555252efe-Paper-Conference.pdf.

Fan Cui, Hongyuan Hou, Zizhang Luo, Chenyun Yin, and Yun Liang. HWE-Bench: Benchmarking LLM agents on real-world hardware bug repair tasks, 2026. URL https://arxiv.org/abs/2604.14709.

DeepSeek. DeepSeek-V4-Pro GA release. Official release note, August 2026. URL https://api-docs.deepseek.com/news/news260813/. Released August 13, 2026; accessed September 25, 2026.

Xiaoyu Dong, Zhi Li, and Xiao-Ming Wu. MUSE: Benchmarking manufacturable, functional, and assemblable text-to-CAD generation, 2026. URL https://arxiv.org/abs/2605.28579.

Xingang Guo, Darioush Keivan, Usman Syed, Lianhui Qin, Huan Zhang, Geir Dullerud, Peter Seiler, and Bin Hu. ControlAgent: Automating control system design via novel integration of LLM agents and domain expertise, 2024. URL https://arxiv.org/abs/2410.19811.

Xingang Guo, Yaxin Li, XiangYi Kong, Yilan Jiang, Xiayu Zhao, Zhihua Gong, Yufan Zhang, Daixuan Li, Tianle Sang, Beixiao Zhu, Gregory Jun, Yingbing Huang, Yiqi Liu, Yuqi Xue, Rahul Dev Kundu, Qi Lim, Yizhou Zhao, Luke Granger, Mohamed Younis, Darioush Keivan, Nippun Sabharwal, Shreyanka Sinha, Prakhar Agarwal, Kojo Vandyck, Hanlin Mai, Zichen Wang, Aditya Venkatesh, Ayush Barik, Jiankun Yang, Chongying Yue, Jingjie He, Libin Wang, Licheng Xu, Hao Chen, Jinwen Wang, Liujun Xu, Rushabh Shetty, Ziheng Guo, Dahui Song, Manvi Jha, Weijie Liang, Weiman Yan, Bryan Zhang, Sahil Bhandary Karnoor, Jialiang Zhang, Rutva Pandya, Xinyi Gong, Mithesh Ganesh, Feize Shi, Ruiling Xu, Yifan Zhang, Yanfeng Ouyang, Lianhui Qin, Elyse Rosenbaum, Corey Snyder, Peter Seiler, Geir Dullerud, Xiaojia Zhang, Zuofu Cheng, Pavan Kumar Hanumolu, Jian Huang, Mayank Kulkarni, Mahdi Namazifar, Huan Zhang, and Bin Hu. Toward engineering AGI: Benchmarking the engineering design capabilities of LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2372.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI Agents That Matter, 2024. URL https://arxiv.org/abs/2407.01502.

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.

Yanchang Liang and Xiaowei Zhao. Simuagent: An LLM-based simulink modeling assistant enhanced with reinforcement learning. *arXiv preprint arXiv:2601.05187*, 2026. doi: 10.48550/arXiv.2601.05187. URL https://arxiv.org/abs/2601.05187.

Mingjie Liu, Nathaniel Pinckney, Brucek Khailany, and Haoxing Ren. VerilogEval: Evaluating large language models for Verilog code generation. In *2023 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2023. doi: 10.1109/ICCAD57390.2023.10323812. URL https://arxiv.org/abs/2309.07544.

Yao Lu, Shang Liu, Qijun Zhang, and Zhiyao Xie. RTLLM: An open-source benchmark for design RTL generation with large language models. In *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 722–727, 2024. doi: 10.1109/ASP-DAC58780.2024.10473904. URL https://doi.org/10.1109/ASP-DAC58780.2024.10473904.

MathWorks. MATLAB MCP Server. Official software release, June 2026a. URL https://github.com/matlab/matlab-mcp-server/releases/tag/v0.11.0. Released June 18, 2026; accessed August 18, 2026.

MathWorks. Simulink: Simulation and model-based design. Official product documentation, 2026b. URL https://www.mathworks.com/products/simulink.html. Accessed August 22, 2026.

MathWorks. Simulink Agentic Toolkit. Official software repository, 2026c. URL https://github.com/matlab/simulink-agentic-toolkit. Accessed August 18, 2026.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations, 2024. URL https://arxiv.org/abs/2411.00640.

OpenAI. Codex. Official software repository, 2026a. URL https://github.com/openai/codex. Version 0.149.0 used; accessed September 25, 2026.

OpenAI. Introducing GPT-5.5. Official release announcement, April 2026b. URL https://openai.com/index/introducing-gpt-5-5/. Released April 23, 2026; accessed September 25, 2026.

Sizhong Qin, Yi Gu, Yao Jiang, Ao Cai, Changjian Zhou, Shaoxuan Shuai, Jiachang Wang, Tianhao Shen, Yueqiang Li, Xinhao Li, Li Zeng, Yueshi Chen, Dachen Gao, Genrong Xu, Wenjie Liao, and Xinzheng Lu. Structureclaw: Traceable LLM agents and an executable benchmark for structural engineering workflows, 2026. URL https://arxiv.org/abs/2607.14896.

Qwen Team. Qwen3.8-27B-FP8. Model card, Hugging Face, August 2026a. URL https://huggingface.co/Qwen/Qwen3.8-27B-FP8. Released August 14, 2026; accessed September 25, 2026.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. Official blog post, August 2026b. URL https://qwen.ai/blog?id=qwen3.8. Accessed September 25, 2026.

Xinxing Ren, Qianbo Zang, and Zekun Guo. Simugen: Multi-modal agentic framework for constructing block diagram-based simulation models, 2025. URL https://arxiv.org/abs/2506.15695.

Sohil Lal Shrestha and Christoph Csallner. SLGPT: Using transfer learning to directly generate simulink model files and find bugs in the simulink toolchain. In *Evaluation and Assessment in Software Engineering*, pp. 260–265. ACM, 2021. doi: 10.1145/3463274.3463806. URL https://doi.org/10.1145/3463274.3463806.

Sohil Lal Shrestha, Shafiul Azam Chowdhury, and Christoph Csallner. SLNET: A redistributable corpus of 3rd-party simulink models. In *Proceedings of the 19th International Conference on Mining Software Repositories*, pp. 237–241. Association for Computing Machinery, 2022. doi: 10.1145/3524842.3528001.

Harmanjot Singh, Abhra Dubey, and Jorge Alejandro Amador Herrera. CADEngBench: It looks like CAD, but does it work? evaluating parametric design, assembly reasoning, and physics simulation, 2026. URL https://arxiv.org/abs/2608.09296.

Jiahui Xiang, Tong Ye, Peiyu Liu, Yinan Zhang, and Wenhai Wang. ModiGen: A large language model-based workflow for multi-task Modelica code generation, 2025. URL https://arxiv.org/abs/2503.18460.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

Z.ai. GLM-5.3-Flash: Frontier intelligence, flash cost. Official blog post, August 2026. URL https://z.ai/blog/glm-5.3-flash. Released August 26, 2026; accessed September 25, 2026.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Jasjeet Sekhon, Jacob Steinhardt, Antony Kellermann, Sarah Schwettmann, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. Establishing best practices for building rigorous agentic benchmarks. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025. URL https://arxiv.org/abs/2507.02825.

APPENDIX

A CLOSEST-WORK COMPARISON

Table 6 summarizes the scope and evaluation criteria of the most relevant studies.

Table 6: **Comparison with related executable-engineering and SIMULINK benchmarks.**

Work	Task scope	Native execution	Evaluation focus
SLGPT (Shrestha & Csallner, 2021)	SIMULINK file generation	Toolchain execution	Fault discovery
SimuGen (Ren et al., 2025)	Image-to-model reconstruction	Basic execution	Blocks and connections
ASWE-Bench (Abdalla et al., 2026)	Automotive control logic	SIMULINK simulation	Boundary and time-series tests
SimuBench (Liang & Zhao, 2026)	Multi-domain text-to-model	Basic execution	Reference similarity and completeness
VerilogEval (Liu et al., 2023)	RTL code generation	Transient simulation	Functional correctness
RTLLM (Lu et al., 2024)	RTL design generation	Compilation and simulation	Syntax, function, and design quality
ModiGen (Xiang et al., 2025)	Modelica code generation	Modelica simulation	Simulation success and model quality
SIMUVERITY	Multi-domain executable models	Loading, compilation, simulation	Requirement-grounded A/Q/M/C/R/D evaluation

B TASK EXAMPLES

XAERO-04: Closed-Loop Tiltrotor VTOL Flight	
Engineering goal	Build a VTOL model coupling flight missions, closed-loop control, tilt actuation, and six-degree-of-freedom vehicle dynamics. Support hover waypoint flight, hover-to-fixed-wing transition, and fixed-wing waypoint or orbit flight.
Inputs and outputs	Inputs are mission type, initial flight mode, mission profile, payload, wind enablement, and simulation duration. Outputs are position-tracking, transition-state, vehicle-state, and actuator-command time series.
Core constraints	Rotor propulsion, independent front-rotor tilt, aerodynamic control surfaces, and vehicle dynamics must form connected paths, with vehicle states participating in feedback control. The model must cover the specified payload and wind conditions, maintain bounded tracking errors and continuous transitions, and avoid prescribed or replayed flight trajectories.
Validation scenarios	Six scenarios: nominal hover, payload boundary, transition with wind, forward transition, combined payload and wind, and a fixed-wing turn.
Applicable dimensions	A/Q/M/R/D.

The corresponding task-level execution and scoring walkthrough is shown in Figure 3.

XAERO-07: Electric-Aircraft Mission and Powertrain	
Engineering goal	Build an aircraft powertrain coupling the flight mission, aerodynamic loads, electric propulsion, and power distribution. Evaluate how battery capacity and payload affect mission execution.
Inputs and outputs	Inputs are battery capacity and payload. Outputs are flight range, battery state, motor current, and low-voltage auxiliary-load power.
Core constraints	Battery capacity affects both stored energy and mass, while payload affects power demand. Electrical and mechanical paths must actively participate in execution.
Validation scenarios	Five scenarios: nominal design, increased capacity, increased payload, joint variation, and an intermediate design point.
Applicable dimensions	A/Q/M/C/R/D.

XAUTO-07: Electric Axle Drive with Regenerative Braking	
Engineering goal	Build a battery-powered electric axle drive coupling closed-loop motor control, the drivetrain, and longitudinal vehicle dynamics for acceleration and regenerative braking.
Inputs and outputs	Inputs scale acceleration and braking demand. Outputs include actual pedal inputs; reference, measured, and estimated torque and torque limits; three-phase currents; and vehicle speed.
Core constraints	A high-voltage source, DC bus, controlled three-phase converter, interior permanent-magnet synchronous motor, reduction gear, and vehicle load form a connected electromechanical path. Phase currents and rotor states participate in feedback control. Vehicle speed follows finite-mass vehicle dynamics, without prescribed speed or replayed torque and current trajectories.
Validation scenarios	Five scenarios: nominal startup and regeneration transition, reduced acceleration demand, reduced regenerative braking, moderate-demand cyclic operation, and asymmetric pedal excitation.
Applicable dimensions	A/Q/M/C/R/D.

C Native Scenarios and Scoring Details

C.1 Scenario Coverage

Each of the 595 formal scenarios is assigned one primary category according to its evaluation focus. Cross-category contributions are not counted twice.

Table 7: **Formal scenarios across the 101 tasks, counted by primary category.**

Primary scenario category	Count
Operating-envelope scenarios	205
Interaction and fault scenarios	189
Temporal-process scenarios	79
Mechanism and causal-check scenarios	122
Total	595

C.2 Dimension Applicability

Domain experts determine applicable dimensions from each task's engineering requirements and system characteristics. The same dimension set applies to every agent system evaluated on that task.

Table 8: **Task counts for the six applicable performance dimensions.**

Performance dimension	A	Q	M	C	R	D
Applicable tasks	101	101	93	35	101	98

Engineering-qualification gate. The G gate evaluates only candidates that have been delivered and execute natively, and returns pass or fail. Its checks are designed by human experts based on the task specification and the executable-system profile, and in general include the presence of the required functional roles, the connectivity of the required signal paths, and the production of valid evaluation evidence according to the task's output contract.

C.3 Electric-Axle Scoring Example

The five scenarios of XAUTO-07 cover all four categories. Table 9 lists their main performance associations.

Table 9: **Electric-axle scenarios, engineering checks, and main performance associations. All scenarios undergo output-completeness checks and contribute to dimension scores and cross-scenario evaluation.**

Primary category	Scenario	Engineering checks	Main dims.
Temporal process	Nominal startup and regeneration transition	Startup torque establishment, transition timing from motoring to regeneration, and torque tracking.	D
Mechanism and causal check	Reduced acceleration demand	Paired comparison with the nominal scenario of torque and speed responses to acceleration demand, together with electromechanical coupling.	M/C
Mechanism and causal check	Reduced regenerative braking	Paired comparison with the nominal scenario of generating torque, final vehicle speed, and braking-response delay.	C/D
Operating envelope	Moderate-demand cyclic operation	Torque tracking, ripple, and current balance at a moderate operating point within the input range.	A/Q
Interaction and fault	Asymmetric pedal excitation	Output quality and performance retention relative to nominal operation under asymmetric acceleration and braking demands.	R

Table 10 reports the six dimension scores and their derivation for the Opus Run 1 candidate on this task.

Table 10: **Six-dimensional performance and score derivation for the Opus Run 1 electric-axle candidate. Scores are rounded to two decimal places.**

Dim.	Engineering metrics	Score derivation for this candidate	Score
A	Torque-tracking and torque-estimation errors	Measured-to-reference tracking error and estimated-to-measured torque error are converted to utilities, then averaged across the five scenarios.	100.00
Q	Torque ripple, peak-to-average ratio, phase-current crest factor and balance, and ride smoothness	Torque and current fluctuations, three-phase current imbalance, and vehicle jerk are evaluated. Their utilities are averaged across the five scenarios.	100.00
M	Functional-path connectivity, motoring and generating operation, and electromechanical coupling	Connectivity, bidirectional operation, torque–acceleration relationships, and current–torque relationships are combined, with greater weight on weaker components. Electromechanical coupling is the main shortfall here.	78.25
C	Acceleration/braking input fidelity and corresponding vehicle responses	Checks assess whether acceleration and braking inputs are applied as required and whether torque and speed respond accordingly. Weaker input responses receive greater weight. Braking-input fidelity and braking response are weaker here.	71.88
R	Retention of accuracy, output quality, electromechanical coupling, and bidirectional operation	Performance retention relative to nominal operation is calculated for each scenario, then aggregated through a weighted combination of the minimum and mean. Stronger motoring with light braking gives the lowest retention, although the overall decline is small.	99.90
D	Startup torque-establishment and regenerative-braking delays	Three delays are checked: nominal startup, nominal braking, and entry into generating operation under reduced braking demand.	100.00

After the prerequisite checks, we use a weighted geometric mean so that all applicable dimensions influence the overall score and a high score on one dimension cannot fully compensate for a critical weakness in another. Domain experts set the weights separately for each task according to its engineering objectives, key failure risks, and dimension importance; non-applicable dimensions are excluded from aggregation. For XAUTO-07, the expert-designed weights are 0.25, 0.10, 0.25, 0.20, 0.10, and 0.10 for A/Q/M/C/R/D, respectively, and the task score follows the same form as the main text:

$$S_t(x) = \Gamma_t\big(r_t^{\mathrm{del}}, r_t^{\mathrm{exe}}, r_t^{G}\big)\, H_t(\mathbf{v}_t), \qquad H_t(\mathbf{v}_t) = A^{0.25}Q^{0.10}M^{0.25}C^{0.20}R^{0.10}D^{0.10}. \quad (1)$$

This candidate passes all three prerequisite checks, so $\Gamma_t = 1$, and obtains 88.03 when its unrounded dimension scores are aggregated.

C.4 Expert Validation of Evaluator Validity

The sample was stratified into 8 high-score candidates, 5 medium-score candidates, 5 low-but-nonzero candidates, 5 candidates that passed the qualification gate but received an overall score of zero, and 7 candidates that failed the qualification gate. For each candidate model, the experts reviewed the task specification, delivered model, and native simulation outputs, and used a five-point scale to rate overall engineering adequacy: 1 point indicates that the task requirements are clearly unmet; 2 points indicate that most requirements are unmet; 3 points indicate a borderline case or insufficient evidence for a determination; 4 points indicate that the task requirements are largely met; and 5 points indicate that the requirements are clearly met with a complete implementation.

The experts did not see evaluator scores, discuss their ratings, or participate in evaluator design. Across the 30 candidate models, the experts assigned the same score in 21 cases, differed by 1 point in 7 cases, and differed by 2 points in 2 cases, with a mean absolute difference of 0.37 points. The weighted Cohen's $\kappa = 0.90$ with a 95% confidence interval of $[0.79, 0.96]$; rank correlations for each comparison are reported in Table 11.

Table 11: **Rank correlations between evaluator scores and domain-expert ratings.**

Statistic	Comparison	n	Estimate	95% CI
Spearman ρ	Expert 1 vs. Expert 2	30	0.87	[0.71, 0.96]
Spearman ρ	Evaluator vs. Expert 1	30	0.88	[0.71, 0.96]
Spearman ρ	Evaluator vs. Expert 2	30	0.94	[0.85, 0.98]
Spearman ρ	Evaluator vs. expert mean	30	0.94	[0.88, 0.98]
Spearman ρ	Evaluator vs. expert mean (nonzero candidates)	18	0.88	[0.72, 0.94]

The table shows high agreement and rank consistency: evaluator scores correlate strongly with both individual experts and their mean rating, and the correlation remains high after excluding zero-score candidates. Thus, evaluator rankings closely agree with expert judgments and are not driven mainly by the zero-score cases.

D Experimental Settings and Supplementary Results

D.1 Implementation Settings

Table 12: **Reasoning effort and context windows. Each model uses its highest supported reasoning level and its official native context-window setting.**

Setting	Claude Opus 4.8	GPT-5.5	DeepSeek V4-Pro	Qwen3.8 Max	GLM-5.3 Flash	Qwen3.8 27B-FP8
Reasoning effort	`max`	`xhigh`	`max`	`xhigh`	`max`	`xhigh`
Context window (tokens)	1M	1,050,000	1M	1M	1,048,576	262,144

The agent frameworks are Claude Code 2.1.238 and Codex 0.149.0. Native execution uses MATLAB/Simulink R2025a and the products required by each task.

Harness pairing and stopping rule. Every model that can be called through an Anthropic-compatible interface runs in the same harness (Claude Code); GPT-5.5 uses its native Codex. Opus 4.8 and GPT-5.5 therefore both run in their vendors' native agent tools, and the unit of evaluation is the agent system (model + harness). Sampling parameters follow each harness's defaults. Each run stops when the agent declares completion.

Environment isolation. Outbound connections of the agent process and the MATLAB session are blocked, and the harness's built-in web tools are disabled. The `examples` directory under the MATLAB root, the `examples` directories of the toolboxes, and the support-package directories are removed. Reference `.slx` files, scenario definitions, and scoring scripts are stored in directories the agent cannot access and are read only by the evaluation process after the agent run ends. We searched all agent trajectories for attempts to access example directories, reference files, or the network; none succeeded.

D.2 Statistical Reliability

Following Miller (2024), we report uncertainty at the task level. Each task score is the mean over three independent runs, and the system score is the mean over 101 task scores. We obtain 95% confidence intervals from 10,000 bootstrap resamples of tasks with replacement and report percentile intervals. For pairwise comparisons, we resample paired task-level differences and compute p-values using the Wilcoxon signed-rank test. These intervals quantify uncertainty when resampling similar tasks; run-to-run variation is shown in the Run 1–3 columns.

Table 13: **Scores for the three runs, mean, task-level 95% confidence interval, and the proportion of tasks with consistent G outcomes across runs.**

System	Run 1	Run 2	Run 3	Mean	95% CI	Consistent G
Claude Opus 4.8	44.34	41.05	43.18	42.86	[37.03, 48.73]	92.08%
GPT-5.5	39.40	42.87	42.91	41.72	[35.29, 48.31]	89.11%
DeepSeek-V4-Pro	27.35	29.52	28.32	28.40	[22.01, 34.76]	82.18%
Qwen3.8-Max	24.57	25.85	24.62	25.01	[18.85, 31.37]	86.14%
GLM-5.3-Flash	3.36	6.19	5.38	4.98	[2.40, 8.04]	81.19%
Qwen3.8-27B-FP8	1.58	1.56	1.67	1.60	[0.41, 2.99]	89.11%

Across the three runs, only Claude Opus 4.8 and GPT-5.5 exchange positions. Paired task-level comparisons place the six agent systems into four tiers: $\{$Claude Opus 4.8, GPT-5.5$\}$ $>$ $\{$DeepSeek-V4-Pro, Qwen3.8-Max$\}$ $>$ GLM-5.3-Flash $>$ Qwen3.8-27B-FP8. The within-tier pairwise differences are not significant ($p = 0.86$ and 0.46), whereas all adjacent between-tier differences are significant ($p \leq 0.012$). The proportion of tasks with consistent G outcomes across runs is 81.19–92.08%.

Table 14: **Differences between adjacent agent systems. 95% confidence intervals are obtained by paired task-level bootstrap, and p-values are from the Wilcoxon signed-rank test.**

Comparison	Difference	95% CI	p
Claude Opus 4.8 – GPT-5.5	1.13	[-5.80, 7.68]	0.86
GPT-5.5 – DeepSeek-V4-Pro	13.33	[6.93, 19.89]	< 0.001
DeepSeek-V4-Pro – Qwen3.8-Max	3.38	[-3.63, 10.39]	0.46
Qwen3.8-Max – GLM-5.3-Flash	20.04	[13.73, 26.54]	< 0.001
GLM-5.3-Flash – Qwen3.8-27B-FP8	3.37	[0.92, 6.03]	0.012

System ranking is insensitive to the aggregation rule. Table 15 recomputes task scores with the G gate retained, using the arithmetic mean, equal-weight geometric mean, or minimum of applicable dimension scores. The ordering below the top pair is unchanged, and Kendall's τ with the original ranking is 0.87, 1.00, and 0.87, respectively.

Table 15: **Overall scores under the original and three alternative aggregation rules; candidates failing G receive zero.**

System	Original	Arithmetic	Equal-weight geometric	Minimum
Claude Opus 4.8	42.86	50.34	44.00	25.38
GPT-5.5	41.72	50.65	42.49	26.48
DeepSeek-V4-Pro	28.40	37.70	29.40	18.87
Qwen3.8-Max	25.01	33.01	25.61	16.06
GLM-5.3-Flash	4.98	7.37	5.38	3.20
Qwen3.8-27B-FP8	1.60	3.29	1.67	0.74

Table 16: **Scores for the three runs, mean, task-level 95% confidence interval, and tasks with scores below Full MCP on the tool-ablation subset.**

Condition	Run 1	Run 2	Run 3	Mean	95% CI	Tasks below Full MCP
Full MCP	82.19	76.04	77.16	78.46	[76.13, 80.93]	—
No-Simulation MCP	6.63	10.85	6.17	7.88	[0.00, 20.06]	10/10
Batch-only	16.81	20.67	17.22	18.23	[0.00, 40.73]	8/10

Full MCP exceeds No-Simulation MCP by 70.58 points (95% CI [58.02, 78.63], Wilcoxon $p = 0.002$) and Batch-only by 60.23 points ([37.13, 77.58], $p = 0.010$). At the task level, they are lower on 10/10 and 8/10 tasks, respectively.

In summary, the six agent systems fall into four tiers, with significant differences between adjacent tiers. In every run, the mean scores of both restricted conditions are below Full MCP. The conclusion remains consistent across the three runs and continues to hold under task-level bootstrap resampling.

D.3 Cross-System Domain Results

Table 17 reports end-to-end and post-G overall scores across ten domains for six systems, comparing overall performance and domain performance after qualification.

Table 17: **End-to-end and post-G overall scores across ten domains for six systems.** An em dash indicates no qualified candidate in any run; 0.00 denotes a defined zero score.

(a) End-to-end overall score

Domain	Claude Opus 4.8	GPT-5.5	DeepSeek V4-Pro	Qwen3.8 Max	GLM-5.3 Flash	Qwen3.8 27B-FP8
Power electronics & control	25.99	30.51	21.63	1.79	5.72	0.00
Mechanical	32.85	34.25	23.74	35.91	0.00	0.00
Robotics	55.49	49.79	47.75	36.83	0.00	0.00
Thermal	49.44	59.48	36.13	36.52	3.52	0.00
Fluid	48.92	46.35	39.79	24.81	19.48	5.87
Automotive	53.03	43.95	15.19	18.29	3.64	3.04
Battery	33.13	36.94	31.04	5.16	10.47	0.00
Aerospace	42.49	35.29	6.92	21.51	2.52	0.00
Communication	33.16	27.56	35.34	29.41	3.14	3.76
Biomedical	50.91	49.68	26.13	37.76	0.00	3.33

(b) Post-G overall score

Domain	Claude Opus 4.8	GPT-5.5	DeepSeek V4-Pro	Qwen3.8 Max	GLM-5.3 Flash	Qwen3.8 27B-FP8
Power electronics & control	58.49	54.93	64.64	16.10	77.24	0.00
Mechanical	47.46	59.24	46.18	49.38	—	—
Robotics	64.05	55.33	54.92	50.58	0.00	0.00
Thermal	62.90	69.99	56.78	57.76	26.24	0.00
Fluid	67.26	80.50	65.83	74.88	79.92	41.50
Automotive	58.33	63.47	26.09	50.29	13.35	25.38
Battery	38.29	44.99	36.00	9.75	43.38	—
Aerospace	47.21	35.29	15.60	37.73	37.87	—
Communication	50.08	35.43	43.40	36.14	28.24	38.90
Biomedical	65.45	61.18	58.78	68.72	0.00	29.96

End-to-end results show that the six systems have different domain-performance profiles. Post-G results further show substantial differences between systems across domains even after engineering qualification.

Claude Opus 4.8's cross-domain capability patterns are analyzed in Section 4.4; the following paragraph supplements the results for the other agent systems. GPT-5.5 achieves relatively higher end-to-end scores in thermal, robotics, biomedical, and fluid domains, but lower scores in communication and in power electronics and control; after engineering qualification, its stronger domains are fluid and thermal. DeepSeek-V4-Pro is relatively stronger in robotics and fluid, but substantially constrained in aerospace and automotive; its qualified candidates remain comparatively strong in power electronics and control, fluid, and biomedical domains. Qwen3.8-Max's end-to-end scores are concentrated in mechanical, robotics, thermal, and biomedical domains, while power electronics and control and battery remain low; after qualification, fluid and biomedical are stronger, whereas power electronics and control and battery remain constrained. GLM-5.3-Flash and Qwen3.8-27B-FP8 obtain near-zero end-to-end scores in most domains, and several domains lack qualified candidates, indicating that these agent systems have not yet formed stable cross-domain engineering capability.

Table 18: **Cross-domain engineering profiles for Claude Opus 4.8 with Claude Code.** "Post-G" averages include only candidates that pass engineering qualification; an em dash marks domains with no dominant bottleneck type.

Engineering domain	*N*	Overall	G pass	Post-G	Capability regime
Power electronics & control	9	25.99	44.44%	58.49	Qualification dominated
Mechanical systems	11	32.85	69.70%	47.46	Compound
Robotics	10	55.49	86.67%	64.05	—
Thermal systems	11	49.44	78.79%	62.90	—
Fluid systems	11	48.92	72.73%	67.26	Qualification dominated
Automotive systems	11	53.03	90.91%	58.33	—
Battery systems	10	33.13	86.67%	38.29	Post-G performance
Aerospace systems	10	42.49	90.00%	47.21	Post-G performance
Communication systems	9	33.16	66.67%	50.08	Compound
Biomedical systems	9	50.91	77.78%	65.45	—

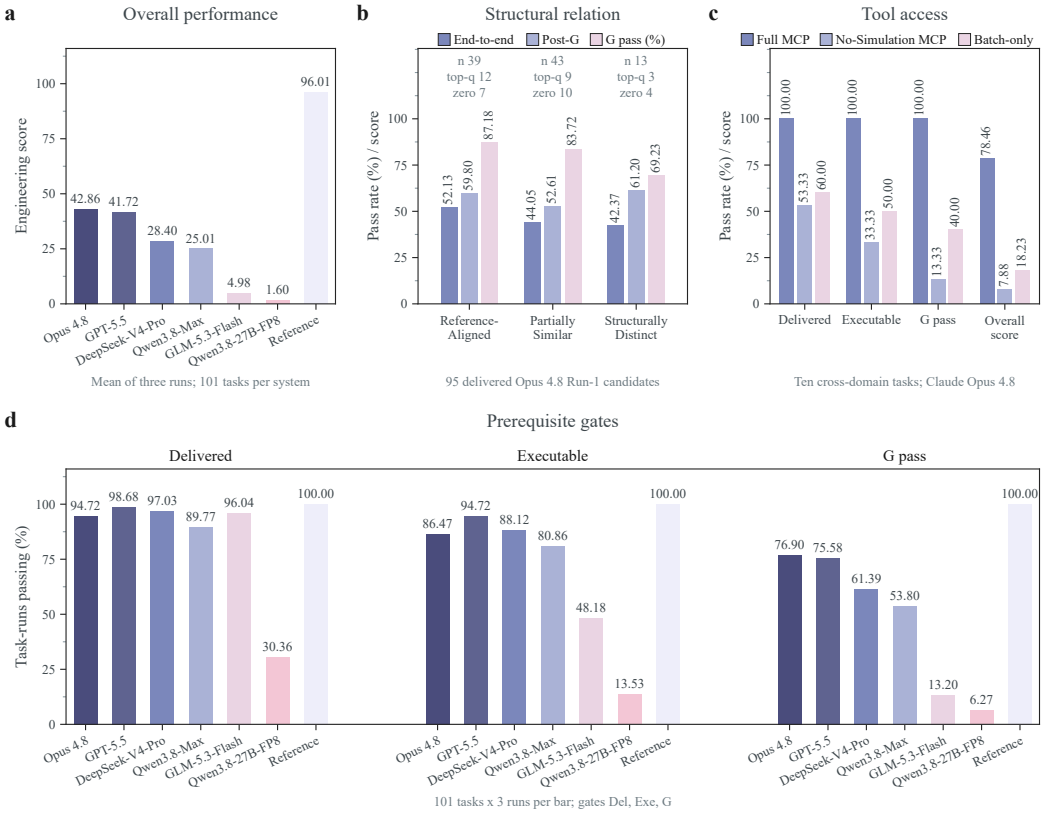

Figure 5: Summary of the experimental results. (a) Overall performance, (b) structural relation, (c) tool access, and (d) prerequisite-gate outcomes.

E STRUCTURAL RELATIONS AND REPRESENTATIVE CASES

E.1 SIX-DIMENSIONAL PERFORMANCE

Table 19: **Six-dimensional scores of the four representative candidates.** All four candidates pass the G gate; an em dash indicates a non-applicable dimension.

Case	A	Q	M	C	R	D
Wheel-loader coordination (XMECH-11)	95.55	99.89	100.00	50.00	91.40	100.00
Electric-aircraft mission and powertrain (XAERO-07)	100.00	33.33	75.00	100.00	66.67	100.00
Serial robotic arm (XMECH-10)	0.00	0.00	0.00	—	0.00	0.00
Radar system (XCOMM-01)	0.00	55.18	25.00	—	0.00	63.69

E.2 STRUCTURAL FEATURES AND ENGINEERING OUTCOMES

Structural relations are assigned qualitatively from block choices and abstraction level, system topology, subsystem decomposition, and signal and feedback routing. The three mutually exclusive relations are defined below.

Table 20: **Three-level structural relation criteria.**

Relation	Decision rule
Reference-Aligned (RA)	The candidate preserves a recognizable correspondence in major block choices and abstraction level, system topology, subsystem decomposition, and signal/feedback routing. Differences are limited mainly to local substitutions, parameterization, auxiliary blocks, or bounded reorganization.
Partially Similar (PS)	The candidate retains some central components or engineering paths, but materially changes the overall architecture, abstraction level, subsystem organization, or signal/feedback routing. Similarity is local rather than system-wide.
Structurally Distinct (SD)	The candidate differs substantially in system-level formulation, abstraction level, topology, subsystem organization, or signal/feedback routing. Shared elements are mainly task-mandated interfaces, generic domain components, or the engineering objective.

Human domain experts classify candidate results from Opus 4.8 Run 1 according to the qualitative criteria above. Layout coordinates, colors, labels, parameter values, solver settings, and block-count differences alone do not determine a label; the relation is not part of the SIMUVERITY score.

The following table gives representative valid-diversity and deceptive-similarity cases from the structural-similarity analysis in Section 4.3.

Table 21: **Representative cases illustrating valid structural diversity and deceptive structural similarity.**

Phenomenon	Task	Relation	G	Score	Engineering evidence
Valid diversity	XMECH-11	Structurally Distinct	Pass	85.10	Reference: detailed physical loader, engine, transmission, and surface subsystems; candidate: a lumped MATLAB-function plant, a different system-level structural implementation.
Valid diversity	XAERO-07	Structurally Distinct	Pass	80.07	Reference: physical Simscape aircraft distribution network; candidate: signal-level lumped equations, with different component families, topology, and routing.
Deceptive similarity	XMECH-10	Reference-Aligned	Pass	0.00	A recognizable serial youBot arm topology extends from the base pivot to the two-finger gripper, but defects in its hierarchy, actuation, and routing cause severely inadequate performance; all performance dimensions are zero.
Deceptive similarity	XCOMM-01	Reference-Aligned	Pass	0.00	The organization and output paths resemble a single-core radar, but target-range and velocity estimation are defective. Objective accuracy and operating-domain robustness are zero.

F Additional Visual Layout Interventions

This appendix provides the seven layout-only comparisons not shown in Figure 4. Within each pair, the block set, parameters, ports, connection topology, hierarchy, simulation configuration, and solver are held fixed; only top-level block and subsystem positions and the routing of existing lines are changed.

F.1 Supplementary Metric Conventions

- **Block overlap:** We count positive-area overlaps among top-level blocks, count each block once, and exclude boundary contact alone.
- **Routing detour:** We count only traceable directed signal paths, excluding paths with zero endpoint distance and undirected physical connections.
- **Line crossings:** We count only internal crossings between different line networks outside blocks, excluding shared endpoints, branches within the same network, and collinear overlap.

F.2 Layout-Only Intervention Results

For each severely disordered model, experts produced a reorganized version by modifying only top-level block positions, subsystem positions, and existing line routes. Blocks, parameters, ports, connectivity, hierarchy, simulation settings, and solver configuration were kept unchanged.

Table 22: **Layout-only optimization of eight high-performing but severely disordered SIMULINK models. Lower values are better for all three layout indicators.**

Candidate model	Domain	Score	Overlap	Detour	Crossings / 100 segments
QPSK Communication Link	Communication	83.33	0.950→0.000	1.283→1.009	71.0→19.6
Two-Body Point-Absorber Wave-Energy Converter	Fluid	84.22	0.966→0.000	5.998→1.060	295.6→70.8
Closed-Loop CPU Cooling System	Thermal	81.02	0.907→0.000	1.493→1.078	204.8→30.6
Grid-Connected Two-Level Inverter	Power electronics & control	74.81	0.240→0.000	1.085→1.080	55.9→43.2
Dual-Patient Ventilation System	Biomedical	73.46	0.952→0.000	1.388→1.013	120.1→36.7
Power-Split Hybrid-Electric Vehicle	Automotive	72.48	0.918→0.000	1.040→1.011	309.9→70.1
Robotic Ball-Balancing System	Robotics	70.89	0.324→0.000	1.000→1.035	82.4→23.2
Three-Level NPC Inverter	Power electronics & control	70.40	0.971→0.000	4.226→1.100	142.3→41.9

F.3 SEVEN SUPPLEMENTARY FIGURES

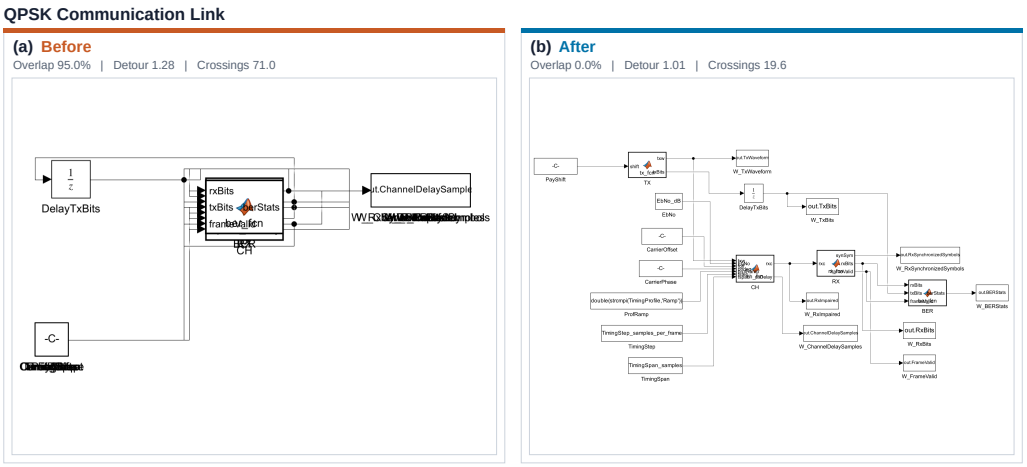

Figure 6: QPSK Communication Link

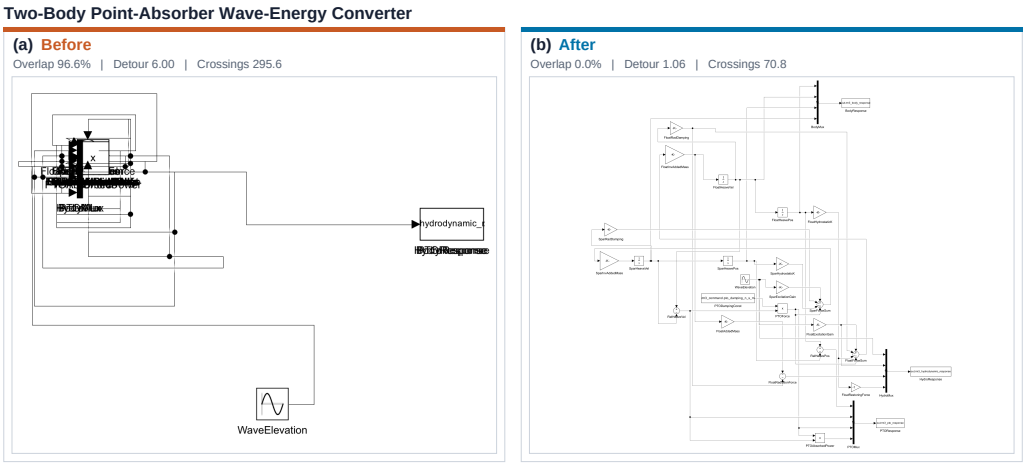

Figure 7: Two-Body Point-Absorber Wave-Energy Converter

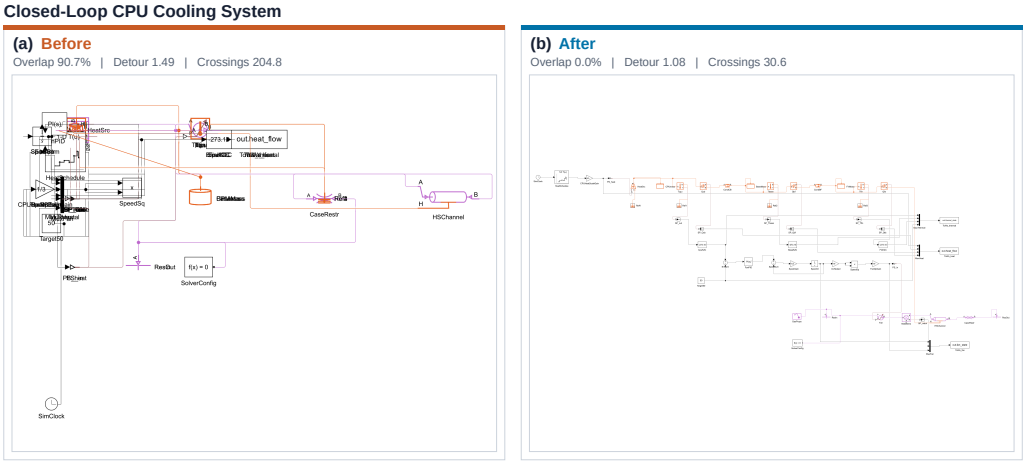

Figure 8: Closed-Loop CPU Cooling System

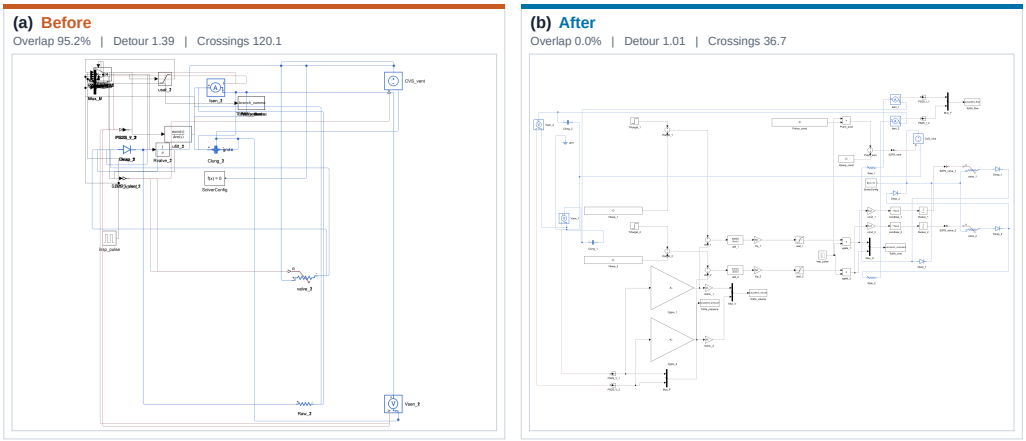

Figure 9: Dual-Patient Ventilation System

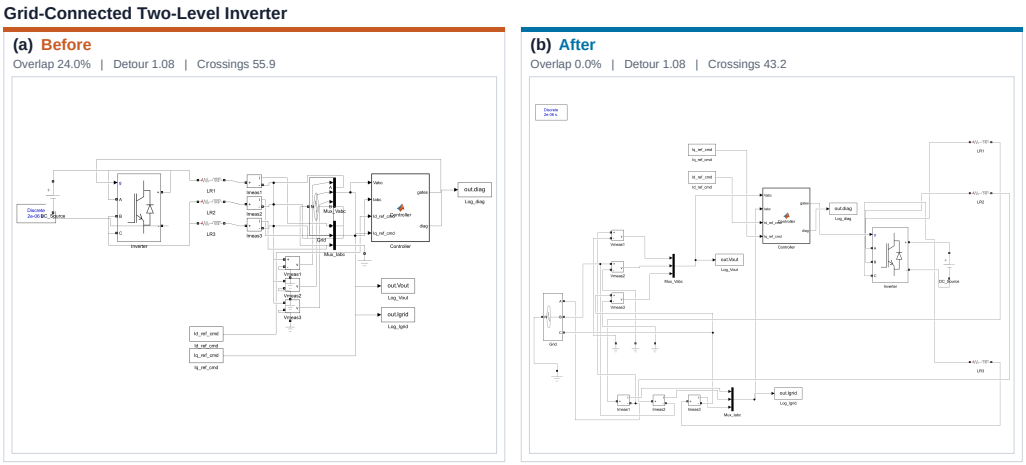

Figure 10: Grid-Connected Two-Level Inverter

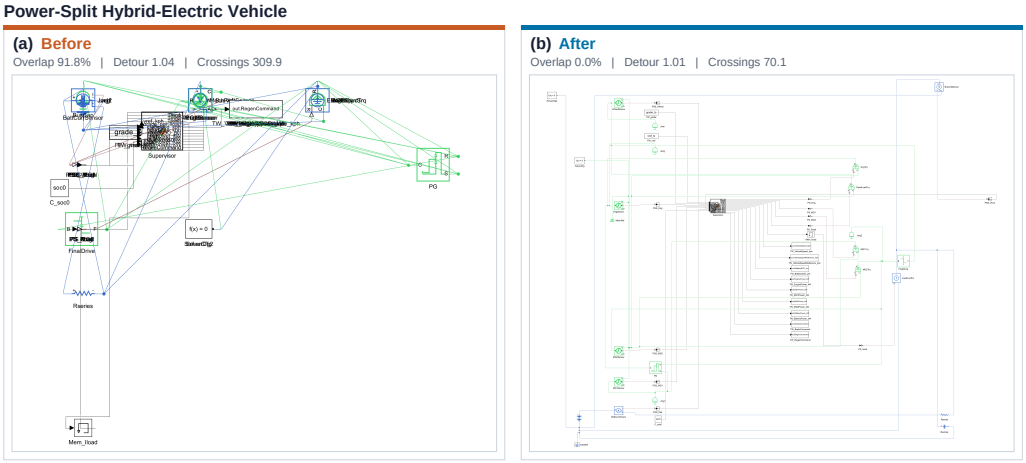

Figure 11: Power-Split Hybrid-Electric Vehicle

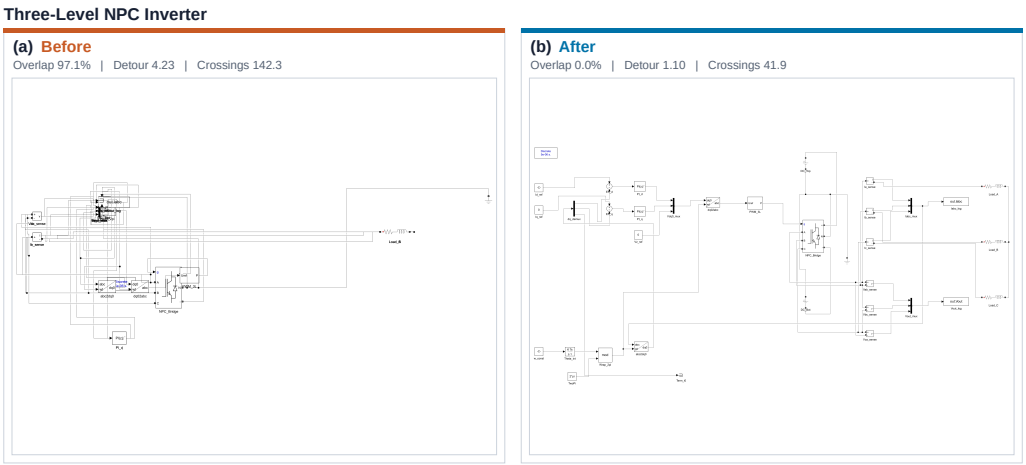

Figure 12: Three-Level NPC Inverter

G TOOL-ABLATION DETAILS

G.1 FIXED TASKS AND TOOL CONDITIONS

Full MCP: Reuses the three-run mean results of these ten tasks for Opus 4.8 from the main experiment.

No-Simulation MCP: Retains MATLAB MCP capabilities for model creation, reading, editing, querying, structural checks, updating, and compilation, but disables agent-side simulation self-checks during generation: the MCP server rejects simulation calls, including `sim` invocations issued through the code-execution tool.

Batch-only: Removes access to all MATLAB MCP tools, retains standard file and shell tools, and permits stateless MATLAB batch scripts through shell calls.

Table 23: **Task-level engineering scores under three tool conditions across ten cross-domain tasks. Each entry is the arithmetic mean over three runs.**

Task	Domain	Full MCP	No-Simulation MCP	Batch-only
XELEC-07	Power electronics & control	74.55	0.00	0.00
XAERO-07	Aerospace systems	75.35	0.00	0.00
XAUTO-07	Automotive systems	85.13	0.00	21.49
XBATT-05	Battery systems	73.33	0.00	75.01
XCOMM-06	Communication systems	79.91	60.87	0.00
XFLUID-08	Fluid systems	79.78	0.00	85.81
XMECH-06	Mechanical systems	79.75	0.00	0.00
XMED-10	Biomedical systems	78.97	0.00	0.00
XROBO-02	Robotics	83.70	17.97	0.00
XTHERM-06	Thermal systems	74.14	0.00	0.00

G.2 SUPPLEMENTARY SIX-DIMENSION PERFORMANCE

Table 24: **End-to-end six-dimensional tool-ablation results for Claude Opus 4.8 with Claude Code on ten cross-domain tasks.**

Tool condition	A	Q	M	C	R	D
Full MCP	78.09	86.82	81.35	97.08	84.50	75.58
No-Simulation MCP	6.56	9.71	14.07	0.00	8.38	5.86
Batch-only	21.70	32.89	38.53	18.76	26.08	21.12