
Toward Engineering AGI: Benchmarking the Engineering Design Capabilities of LLMs

EngDesign Benchmark Team*

Abstract

1 Modern engineering, spanning electrical, mechanical, aerospace, civil, and computer disciplines, stands as a cornerstone of human civilization and the foundation of our society. Today, industry pioneers dream of developing general-
2 purpose AI engineers capable of designing and building humanity’s most ambitious
3 projects—from starships that will carry us to distant worlds to Dyson spheres that
4 harness stellar energy. Yet engineering design represents a fundamentally different
5 challenge for large language models (LLMs) compared to traditional textbook-style
6 problem solving or factual question answering. Although existing benchmarks have
7 driven progress in areas such as factual question answering, code synthesis, and
8 scientific problem solving, real-world engineering design demands the synthesis
9 of domain knowledge, navigation of complex trade-offs, and management of the
10 tedious processes that consume much of practicing engineers’ time. Despite these
11 shared challenges across engineering disciplines, no benchmark currently captures
12 the unique demands of engineering design work. In this work, we introduce **ENGDESIGN**,
13 an **Engineering Design** benchmark that evaluates LLMs’ abilities to
14 perform practical design tasks across nine engineering domains: Operating System
15 Design, Computer Architecture Design, Control System Design, Mechanical
16 Systems, Structural Design, Digital Hardware Design, Analog Integrated Circuit
17 Design, Robotics, and Signal Processing. Unlike existing benchmarks that focus
18 on factual recall or question answering, ENGDESIGN uniquely emphasizes LLMs’
19 ability to synthesize domain knowledge, reason under constraints, and generate
20 functional, objective-oriented designs. Each task in ENGDESIGN represents a
21 real-world engineering design problem, accompanied by a detailed task description
22 specifying design goals, constraints, and performance requirements. We pioneer
23 a *simulation-based evaluation paradigm* where LLM-generated designs undergo
24 rigorous testing through executable, domain-specific simulations—from circuit
25 SPICE simulations to structural finite element analysis, from control system val-
26 idation to robotic motion planning. ENGDESIGN establishes a new benchmark
27 paradigm that moves beyond textbook knowledge to assess genuine engineering
28 capability and shifts evaluation from static answer checking to dynamic, simulation-
29 driven functional verification, marking a crucial step toward realizing the vision of
30 engineering AGI.
31
32

33 1 Introduction

34 Modern engineering, spanning electrical, mechanical, aerospace, civil, and computer disciplines
35 [1, 2, 3, 4], stands as a cornerstone of human civilization and the foundation of our society. From
36 the electrical grids powering our cities to the aerospace systems launching us beyond Earth, from

*The ENGDESIGN Team includes over 60 contributors comprising core developers, task contributors, and domain faculty experts. Due to space constraints, the full author list is provided in the Appendix.

Correspondence to: Xingang Guo xingang2@illinois.edu, Bin Hu binhu7@illinois.edu.

37 analog integrated circuits forming the foundation of electronics to the structural designs supporting
38 our skylines, engineering disciplines have woven themselves into the fabric of human existence
39 [5, 6, 7]. At its core, engineering design is the systematic and creative process that engineers use to
40 solve problems and create functional products, systems, or processes, transforming requirements and
41 constraints into tangible solutions that meet human needs. Recent advances in large language models
42 (LLMs) have led to remarkable performance on conventional question-answering (QA) benchmarks,
43 with strong results across a wide range of tasks—from textbook-level scientific problem solving
44 [8, 9, 10, 11], to code synthesis [12, 13, 14], and even to answering textbook-level questions across
45 various engineering domains [15, 16, 17]. Today, industry leaders aspire to build on such progress
46 in LLMs to create general-purpose AI engineers capable of bringing to life humanity’s boldest
47 ambitions—from interstellar starships to Dyson spheres that harvest stellar energy—leveraging these
48 foundation models’ vast knowledge across engineering domains and their potential to transcend
49 traditional disciplinary boundaries that constrain human engineers [18].

50 However, practical engineering design confronts LLMs with a grand challenge profoundly different
51 and vastly more complex than conventional factual recall or textbook-level engineering problem
52 solving. Specifically, real-world engineering design demands the synthesis of domain knowledge,
53 navigation of complex trade-offs, management of the tedious processes that consume much of
54 practicing engineers’ time, and rigorous validation through domain-specific simulators to ensure that
55 designs meet functional requirements and safety constraints. Yet despite progress in textbook-level
56 engineering problem solving, no existing benchmark captures the distinctive demands of practical
57 design work—with its multifaceted, open-ended challenges across multiple engineering disciplines.

58 In this work, we introduce ENGDESIGN, the first benchmark for holistically evaluating LLMs
59 on authentic, multi-domain engineering design challenges. Unlike traditional QA formats, our
60 benchmark requires models to function as practicing engineers, producing functional solutions,
61 such as dynamical system controllers, material structure designs, analog integrated circuits, or GPU
62 architectures that meet rigorous performance requirements. ENGDESIGN uniquely emphasizes
63 the synthesis of domain knowledge, constraint-based reasoning, and the generation of functional,
64 objective-oriented designs, spanning practical design problems across nine engineering domains:
65 Operating System Design, Computer Architecture Design, Control System Design, Mechanical
66 Systems, Structural Design, Digital Hardware Design, Analog Integrated Circuit Design, Robotics,
67 and Signal Processing. Each task in ENGDESIGN is accompanied by an executable evaluation
68 pipeline that validates designs against task requirements using domain-specific simulation tools
69 such as SPICE simulations [19], structural finite element analysis [20], MATLAB Control System
70 Toolbox [21], and other scientific software tailored to specific engineering domains. Our approach
71 establishes a revolutionary evaluation paradigm: assessment using engineering simulation tools.
72 This simulation-based approach fundamentally shifts evaluation from linguistic pattern matching to
73 functional verification, ensuring that solutions are assessed based on their engineering merit rather
74 than textual plausibility.

75 ENGDESIGN advances LLM evaluation methodology by moving beyond conventional question-
76 answering benchmarks to address the open-ended, constraint-driven nature of real-world engineering
77 design. Our key contributions include:

- 78 • **The first multi-domain engineering design benchmark.** Unlike QA benchmarks that
79 test narrow factual or procedural knowledge, ENGDESIGN evaluates LLMs on complex
80 engineering design challenges spanning diverse engineering disciplines (e.g., mechanical,
81 electrical, civil), requiring synthesis of domain knowledge, constraint satisfaction, and strong
82 reasoning capabilities.
- 83 • **Executable simulation-based evaluation.** We replace static correctness checks with
84 dynamic, domain-specific evaluation pipelines. Each task includes human-designed task-
85 specific evaluation scripts that rigorously verify functional feasibility (e.g., via simulation,
86 constraint validation, or performance testing), ensuring objective, reproducible scoring
87 grounded in real engineering standards.
- 88 • **Partial-credit grading for incomplete solutions.** Conventional benchmarks often use
89 binary scoring, but ENGDESIGN’s rubrics quantify incremental progress (e.g., 20/100 for
90 delivering a stable controller that meets partial performance requirements). This granularity
91 reveals nuanced capability gaps and rewards iterative refinement.

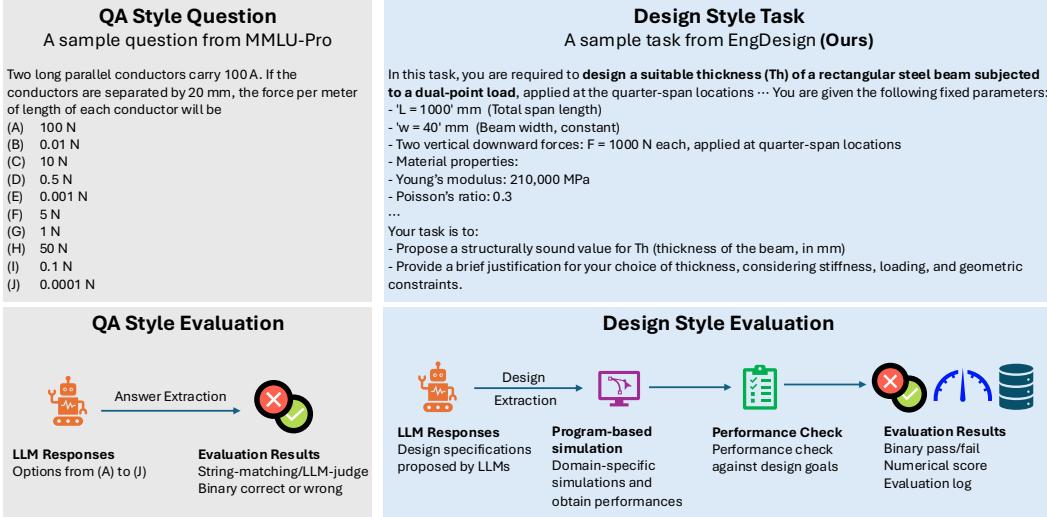


Figure 1: Comparison between conventional QA-style benchmarks (left) and the design-style benchmark ENGDESIGN (right). Conventional QA benchmarks evaluate LLMs through static answer extraction and string-matching, while ENGDESIGN involves open-ended design tasks with potentially non-unique solutions. LLMs must propose candidate design specifications, which are evaluated via program-based simulations and performance validation pipelines.

- 92 • **Empirical validation of design competencies.** Through large-scale evaluations across
93 10+ state-of-the-art LLMs including both general-purpose LLMs and reasoning models,
94 we demonstrate ENGDESIGN’s ability to expose critical limitations in AI systems—such
95 as over-reliance on textual patterns or failure to handle trade-offs—that are invisible to
96 traditional benchmarks.
- 97 We believe that ENGDESIGN marks a crucial milestone toward realizing engineering AGI by breaking
98 new ground with its comprehensive coverage of engineering design problems across diverse domains,
99 each requiring deep technical expertise and domain-specific simulation tools for rigorous evaluation.
100 This benchmark not only measures what models know but what they can actually design, bridging
101 the chasm between textbook-level knowledge understanding and practical engineering capability.

102 2 ENGDESIGN

- 103 In this section, we introduce ENGDESIGN, a multi-domain benchmark developed to evaluate the
104 capabilities of LLMs in real-world engineering design tasks. Unlike conventional QA benchmarks,
105 ENGDESIGN adopts a rigorous simulation-based evaluation pipeline to assess model performance in
106 practical, design-oriented scenarios. Table 1 summarizes the benchmark’s key statistics: it comprises
107 101 design tasks spanning 9 engineering domains, with a total of 473 gradable items. While the task
108 distribution may appear uneven, it naturally reflects the domain expertise of our contributors and
109 the filtering rigor imposed by our multi-stage review process. Notably, the average prompt length in
110 ENGDESIGN is 778.71 tokens, substantially higher than typical QA benchmarks², highlighting the
111 contextual richness and complexity of realistic engineering design problems. Word clouds generated
112 from ENGDESIGN prompts are included in Appendix D.
- 113 Among the 101 tasks in ENGDESIGN, 48 tasks require domain-specific scientific software such
114 as MATLAB or Cadence for evaluation, while the remaining 53 tasks are fully open-sourced and
115 evaluated using manually authored evaluation scripts. We consolidate these tasks into a subset
116 called ENGDESIGN-OPEN to support broader community adoption without licensing constraints.
117 Additionally, 23 tasks of ENGDESIGN incorporate images as part of the task input to LLMs.

²We provide token length comparison between ENGDESIGN and other QA benchmarks in Appendix D.1.

Domain: Control Design (XG_05)

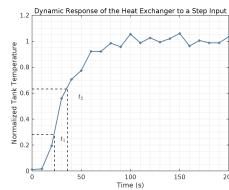
Task 1

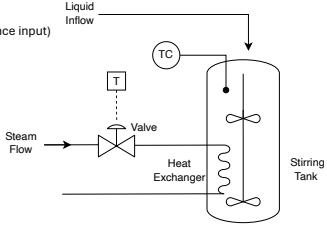
Your first task is to derive a first-order with delay transfer function $G(s) = e^{-\theta s}/(1 + \tau s)$ to model the dynamics of the stirred tank ... Please determine the value of θ and τ from the step response figure using the given information.

Task 2

Your second task is to design a feedback controller ... that satisfies the following requirements:

- Gain margin: >= 7 dB
- Phase margin: >= 60 degrees ...
- Setting time: <= 150 sec (for a step reference input)





Evaluation Pipeline (with MATLAB)

```
def evaluate_llm_response(llm_response):
    # Start MATLAB engine
    eng = matlab.engine.start_matlab()
    # Load configuration coefficients from LLM response
    theta = llm_response.config.theta
    den = llm_response.config.den
    # Run MATLAB evaluation
    passed, eval_details, score = eng.evaluate(theta, tau, num,
                                                den, nargout=3)
    return passed, eval_details, score
```

Evaluation Pipeline (with SPICE Simulator)

```
def evaluate_llm_response(llm_response):
    # Load netlist from LLM response
    netlist = llm_response.config.netlist
    # Write netlist to file for evaluation
    with open(netlist_path, "w") as fh:
        fh.write(netlist)
    # Run simulation and get evaluation results ...
    ok, err = run_autograder(shell_path, "mp3")
    - passed = ...
    score = ...
    return passed, details, score
```

Evaluation Pipeline (with MATLAB)

```
#!/bin/bash
# Prompt user for MP number
echo "Enter the MP number (e.g., mp1, mp2, etc.):"
mp_number=$1
# Load Python3 module
module load python3

# Run Virtuoso
virtuoso -replay "$LOG_DIR/autograde.tcl"
# Wait for Virtuoso to complete
wait
# Find the latest CDS log file after Virtuoso runs
LATEST_LOG=$(ls -t ~/CDS.log* >/dev/null | head -n 1)
-
```

Evaluation Pipeline (with MQSim Simulator)

```
def evaluate_llm_response(llm_response):
    # Load ground truth obtained via MQSim simulation
    ground_truth = json.load(open(ground_truth_file, "r"))
    # Check performance for each workload
    passed, score, details = False, 0, {}
    for i, workload_name in enumerate(llm_response.config.workload_names):
        # Performance check for workload I
        # Check performance for each workload
        for workload in tuning_result.keys():
            ...
            # Scoring the LLM responses based on ground_truth
            for name in tuning_result[workload]:
                ...
    return passed, eval_details, score
```

Figure 2: Selected demonstration tasks from ENGDESIGN.

4

Table 1: Statistics of ENGDESIGN. Tokens are calculated with Tiktoker using o200k_base encoding.

Engineering Area	# of tasks	# of gradable items	# of query tokens		
			Max	Min	Avg
Operating System Design	8	66	2938	310	1103.25
Computer Arch Design	5	20	4385	2348	3539.60
Control Design	18	100	1361	209	634.44
Mechanical Systems	7	32	781	225	391.14
Structure Design	13	25	483	186	345.31
Digital Hardware Design	17	58	1715	206	515.65
Analog IC Design	5	23	2136	547	1196.6
Robotics	10	68	1485	192	771.9
Signal Processing	18	81	2304	151	611.72
Overall	101	473	4385	151	778.71

118 2.1 Task Structure

119 Each task of ENGDESIGN consists of the following four key components:

120 **1. Task Description.** This part is the query prompt fed into the LLMs, offering a clear and detailed
121 definition of the engineering design problem, including design objectives, specifications, constraints.

122 **2. Evaluation Rubrics.** Given the complexity of ENGDESIGN tasks, each task is further decomposed
123 into multiple gradable items evaluated individually during the performance check stage. The evalua-
124 tion rubrics define the assessment criteria and scoring metrics, with a full score of 100. This enables
125 models to receive partial credit even if the design does not fully meet all specified requirements.

126 **3. Evaluation Pipeline.** Each task includes automated evaluation scripts that assess the LLMs'
127 designs. The evaluation returns a binary pass/fail indicator, a numerical score, and evaluation logs for
128 further analysis.

129 **4. Reference Design.** Each task provides a validated reference design that fully satisfies all specified
130 requirements, ensuring the feasibility and realism of the design challenge.

131 Figure 2 shows four demonstrated design examples from ENGDESIGN including task descriptions
132 and evaluation pipelines.

133 2.2 Evaluation Pipeline

134 Unlike conventional benchmarks, which often have a single golden answer that allows evaluation
135 through exact string matching or LLM-as-judge scoring, our engineering design tasks inherently
136 lack a unique *golden design*. For example, there are infinitely many valid controller designs that
137 can regulate the temperature of a stirred tank modeled by a first-order system while satisfying
138 both time-domain and frequency-domain performance requirements. To address this challenge,
139 our benchmark introduces a **simulation-based evaluation pipeline**. Specifically, for each LLM-
140 generated design, we first parse the key design components from the response automatically (such
141 as code snippets, key parameters, etc.). These components are then fed into our evaluation pipeline,
142 which runs domain-specific simulations to assess the design's performance against the specified task
143 requirements. Guided by a per-task rubric table, the evaluation pipeline outputs three key results: a
144 binary pass/fail indicator, a numerical score between 0 and 100, and a detailed log that records the
145 evaluation process for further analysis. Below we discuss each step in detail.

146 **Structured Responses from LLMs.** To constrain the output format for a wide range of LLMs and
147 ensure compatibility with our evaluation pipeline, we adopt the popular open-source Python library
148 `instructor` [22], which is built on top of Pydantic and facilitates structured LLM responses. By
149 defining schema templates that specify expected fields, such as design parameters or code snippets,
150 the `instructor` package enables LLMs to produce outputs in a pre-defined format. We provide an
151 example at Figure 10 in Appendix E.2. Specifically, LLMs are instructed to construct their responses
152 into two main parts: (1) a reasoning field, which contains the step-by-step reasoning process for
153 solving the task, and (2) a `ConfigFile` class, which summarizes the final design results, including

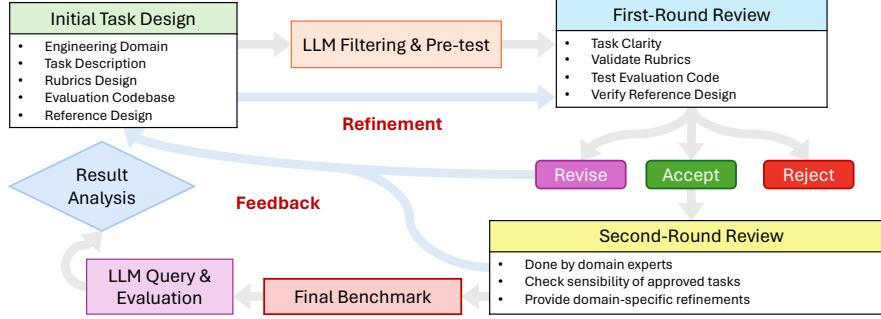


Figure 3: An overview of the construction process of ENGDESIGN, illustrating stages from initial task design, LLM filtering, and expert review to final benchmark integration, with iterative refinement and evaluation.

154 their design choices or code snippets. During evaluation, the fields defined in `ConfigFile` can be
 155 automatically parsed to trigger the simulation-based evaluation pipeline.

156 **Task-Specific Evaluation Pipeline.** A simulation-based evaluation pipeline was designed for each
 157 task. For example, control engineering tasks may evaluate closed-loop dynamics through metrics such
 158 as rise time, settling time, overshoot, and phase/gain margins via MATLAB. The evaluation pipeline
 159 outputs three key results: (1) a binary pass/fail indicator for meeting all performance requirements,
 160 (2) a numerical score (0–100) reflecting fine-grained performance analysis, and (3) an evaluation log
 161 capturing simulation outputs, performance metrics, or error messages.

162 2.3 Construction of ENGDESIGN

163 The construction of ENGDESIGN follows a multi-stage process that integrates both automated
 164 validation and human expertise as shown in Figure 3:

- 165 **1. Initial Task Design.** We recruited graduate students and researchers from various engineering
 166 disciplines to contribute initial task proposals, drawing on open-source resources and
 167 their domain knowledge. Contributors were guided to follow a standardized submission
 168 format, which includes a task description, evaluation rubric, executable evaluation codebase,
 169 and a reference solution.
- 170 **2. LLM Filtering and Pre-Test.** Submitted tasks undergo an initial filtering phase using a
 171 language model (o4-mini) to assess prompt sufficiency and the functionality of the evaluation
 172 code. Specifically, we prompt the LLM to classify the task’s engineering domain and
 173 determine whether the information provided is adequate to solve the problem. If the model
 174 flags missing or unclear details, we work with the original contributor to address those gaps.
- 175 **3. First-Round Review.** Tasks that pass the pre-test enter a first-round review involving close
 176 collaboration between reviewers and the original authors. The review ensures that:
 - 177 • The task description is clear, self-contained, and complete.
 - 178 • The evaluation rubric is well-defined with partial credit, and aligns with the task goals.
 - 179 • The evaluation codebase is executable, robust, and handles typical edge cases.
 - 180 • The reference design meets all performance requirements, validating the task feasibility.

181 Based on this review, tasks fall into one of three categories: **Accept**: the task meets
 182 all requirements and proceeds to expert review; **Revise**: minor issues are identified and
 183 addressed in collaboration with the contributor; or **Reject**: the task is excluded due to critical
 184 issues such as unfeasibility, triviality, or misalignment with design-oriented evaluation.

- 185 **4. Second-Round Review with Domain Experts.** Tasks accepted in the first round are
 186 reviewed by faculty experts in the corresponding engineering domain. These experts assess
 187 the technical soundness and relevance of each task to ensure it serves as a meaningful
 188 evaluation of LLMs in realistic design settings.
- 189 **5. Final Integration.** Tasks that pass both review stages are standardized to match the benchmark’s
 190 formatting and structural guidelines, and are formally included in the ENGDESIGN.

Table 2: Average pass rate (%) results (each task evaluated over 3 trials). * Model is not multi-modal and was evaluated only on the text-only subset. We report text-only results for all models in Appendix E.4. The domain abbreviations are: AICD = Analog Integrated Circuit Design, Arch = Computer Architecture Design, Ctrl = Control Design, DHD = Digital Hardware Design, Mech = Mechanical Systems, OS = Operating System Design, Robo = Robotics, SigP = Signal Processing, Stru = Structure Design. The best results for each column are highlighted in bold.

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
Chat Models										
GPT-4o	15.68	0.00	20.00	18.52	10.26	0.00	4.17	26.67	17.65	25.64
DeepSeek-v3*	17.92	0.00	0.00	27.27	25.64	0.00	0.00	16.67	12.82	38.89
Gemini-2.0-Flash	14.16	0.00	0.00	15.15	16.67	4.76	0.00	20.00	5.13	36.11
Claude-3.7-Sonnet	22.61	0.00	40.00	16.67	33.33	14.29	0.00	33.33	21.57	30.77
Reasoning Models										
o1	29.17	0.00	40.00	24.07	41.03	14.29	37.50	50.00	25.49	23.08
o3	34.38	0.00	40.00	35.19	20.51	23.81	25.00	63.33	41.18	30.77
o3-high	33.57	0.00	40.00	27.78	20.51	28.57	30.43	68.97	35.29	41.03
o4-mini	31.60	0.00	33.33	24.07	41.03	19.05	29.17	43.33	33.33	41.03
o4-mini-high	34.04	0.00	40.00	27.78	47.22	19.05	37.50	46.67	35.29	35.90
Gemini-2.5-Pro	29.54	0.00	0.00	33.33	43.59	0.00	9.52	56.67	12.82	50.00
DeepSeek-R1*	25.53	0.00	36.36	36.36	38.46	4.76	5.26	26.67	20.51	41.67
Claude-3.7-Thinking	20.07	0.00	33.33	18.52	17.95	9.52	0.00	40.00	19.61	28.21

3 Experiments

We evaluate a set of representative LLMs on the ENGDESIGN and analyze their performance.

Evaluated LLMs. For chat models, we include GPT-4o, Claude-3.7-Sonnet, Gemini-2.0-Flash, and DeepSeek-v3. For reasoning models, we evaluate o1, o3, o3-high, o4-mini, o4-mini-high, Claude-3.7-Thinking, DeepSeek-R1, and Gemini-2.5-Pro.

Evaluation Metrics. We report three primary evaluation metrics: (1) **Average Pass Rate**, (2) **Average Score**, and (3) **Reasoning Robustness**. Each task is evaluated over three independent trials per model. To measure reasoning robustness, we compute the ratio between the number of tasks where all three trials passed and the number of tasks where at least one trial passed. This ratio, which ranges from 0 to 1, serves as a straightforward indicator of an LLM’s reasoning consistency. A value closer to 1 indicates higher robustness and more stable reasoning behavior across repeated runs. Formal definitions of all evaluation metrics are provided in Appendix E.3.

3.1 Main Results

Table 2 presents the average pass rates and average score across evaluated LLMs, across 9 distinct engineering design areas. The average score demonstrates a strong correlation with the pass rate, which we provided the detailed results in Appendix E.4. We make the following key observations.

ENGDESIGN presents a highly challenging benchmark, with even the best-performing models achieving only modest pass rate. As shown in Table 2, no model surpasses a 35% overall pass rate. The top-performing model, o3, achieves 34.38%, while widely-used chat models such as GPT-4o and Gemini-2.0-Flash attain around 15%. Analog IC design tasks stands as the most difficult ones where all the models got 0% pass rate. These results highlight the difficulty of the benchmark, which stems from its requirements for domain-specific expertise, multi-step reasoning, and design trade-off considerations, capabilities that go beyond the strengths of current LLMs.

Reasoning models outperform general-purpose chat models. Models designed with enhanced reasoning capabilities, including o3, o4-mini, and DeepSeek-R1, consistently outperform general-purpose chat models across most engineering domains. However, this trend does not hold for Claude models, where both Claude-3.7-Sonnet and Claude-3.7-Thinking exhibit similar pass rates despite their differences in reasoning emphasis.

Reasoning Models are more robust in general. Figure 4 shows the reasoning robustness of all evaluated models. Reasoning-focused models such as o1, o3, and o4-mini-high achieve the highest

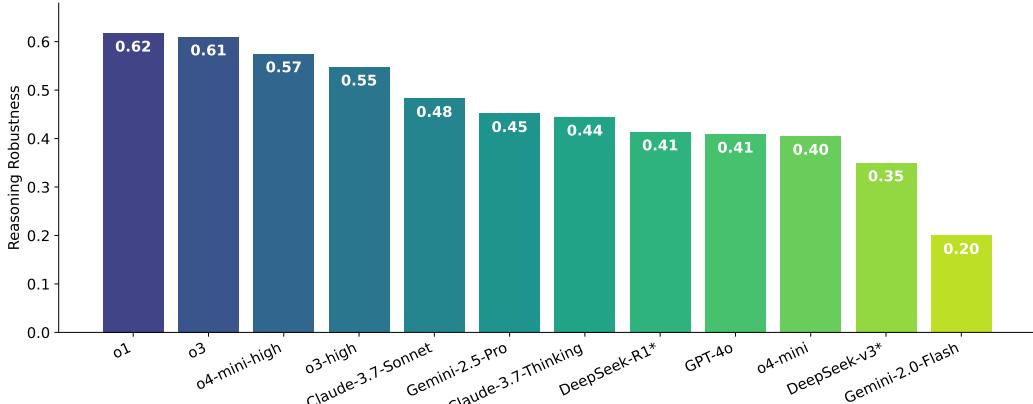


Figure 4: Reasoning robustness distribution of evaluated LLMs on ENGDESIGN.

robustness scores (0.62, 0.61, and 0.57, respectively), indicating strong consistency across repeated trials on tasks they are capable of solving. In contrast, chat-oriented models like Gemini-2.0-flash and DeepSeek-v3 exhibit substantially lower robustness (0.20 and 0.35), reflecting less stable reasoning behavior. Notably, the gap in reasoning robustness between top-performing reasoning models and weaker chat models is more pronounced than in average pass rate as shown in Table 2—up to a 3× performance difference in robustness compared to a 2× difference in pass rate. This highlights robustness as a distinguishing strength of reasoning models.

Token Consumption. Reasoning models require substantially more compute due to longer inference time. We analyze the number of completion tokens generated across different models. As shown in Figure 5, the reasoning model (o1) produces significantly more tokens than the non-reasoning model (GPT-4o) to achieve performance gains. Optimizing the latency–performance trade-off will be essential for making future models more accessible and deployable in real-world engineering design scenarios. Additional results are provided in Appendix E.5.

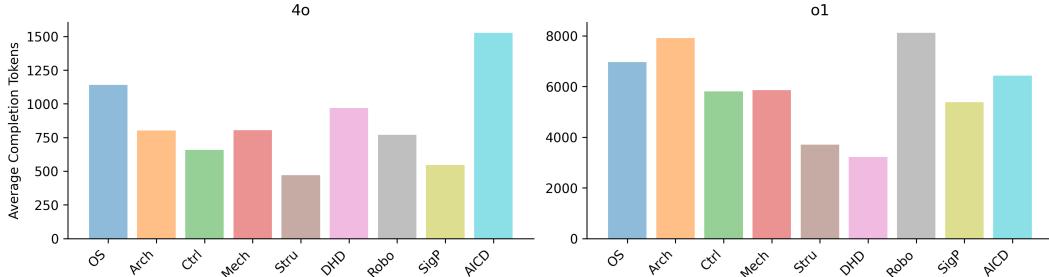


Figure 5: Average token consumption of evaluated LLMs on ENGDESIGN.

3.2 Iterative Design

To emulate the workflow of human engineers, we implement an iterative design protocol that allows LLMs to refine their solutions based on feedback from previous attempts. In our implementation, the LLM is provided with its previous design output along with corresponding evaluation results, such as scores, performance metrics, and diagnostic logs, and is then prompted to generate an improved design in the subsequent iteration. As shown in the left plot of Figure 6, model performance consistently improves with additional iterations. Notably, o3 achieves almost a 60% pass rate after ten iterations.³ However, we also observed that iterative design does not help in all cases. For example, in Analog IC design tasks, models still fail to meet the requirements even after ten iterations.

³Due to time and resource constraints, the iterative design protocol was applied to a selected subset (71 tasks) of ENGDESIGN tasks using four representative models: GPT-4o, o1, o3, and o4-mini. As a result, reported pass

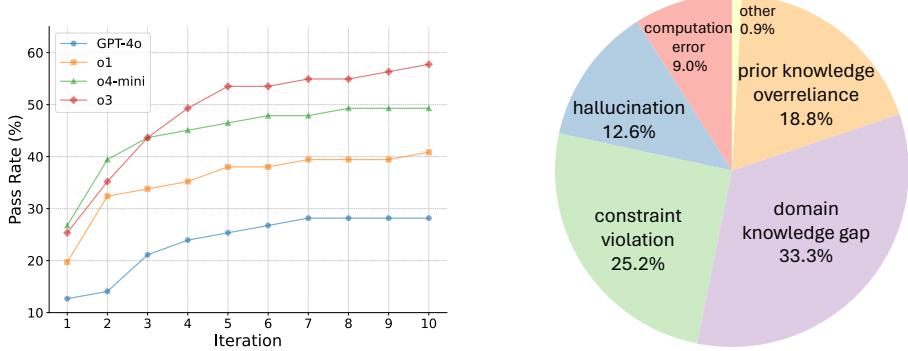


Figure 6: Left: pass rate of GPT-4o, o1, o3, and o4-mini with the iterative design setup. Right: distribution of error types for o4-mini on a subset of ENGDESIGN.

243 3.3 Failure Analysis

244 To better understand the limitations of LLMs in engineering design tasks, we define five primary
 245 error categories observed in ENGDESIGN: **Domain Knowledge Error**, **Constraint Violation Error**,
 246 **Prior Knowledge Overreliance**, **Hallucination**, and **Computation Error**. Formal definitions of
 247 these error types are provided in Appendix F.

248 We analyzed o4-mini’s responses to 70 tasks from ENGDESIGN in which the model failed evaluation,
 249 and manually annotated each failure with corresponding error types. Given the complexity of
 250 engineering design tasks, many responses exhibited multiple failure modes; we therefore allowed
 251 multi-label assignments per task. In total, 111 distinct errors were identified. The right plot of
 252 Figure 6 summarizes the distribution of error types. Domain knowledge errors (33.3%), constraint
 253 violations (25.2%), and prior knowledge overreliance (18.8%) collectively account for over 75% of
 254 all observed failures. This suggests that o4-mini frequently struggles with applying domain-specific
 255 principles, satisfying task-specific constraints, and generalizing beyond memorized patterns. While
 256 hallucinations (12.6%) remain a concern, computation errors (9.0%) are less common, indicating
 257 relatively strong arithmetic and symbolic reasoning capabilities. Additional examples and qualitative
 258 insights are provided in Appendix F.

259 4 Related Work

260 Our work relates to the broader field of benchmark design for LLMs, particularly those developed
 261 to evaluate general reasoning and problem-solving capabilities [23, 9, 24, 25, 12, 8, 10, 26, 27].
 262 ENGDESIGN also contributes to a growing body of research focused on evaluating LLMs in spe-
 263 cific engineering domains [15, 28, 29, 30]. Due to space constraints, additional related work is
 264 discussed in Appendix C. Compared to prior work, ENGDESIGN spans multiple engineering domains
 265 and introduces a novel evaluation framework that combines generative outputs from LLMs with
 266 simulation-based evaluation to assess performance in realistic design settings.

267 5 Conclusion

268 In this work, we introduced ENGDESIGN, a novel benchmark for rigorously evaluating the capabilities
 269 of LLMs in realistic, multi-domain engineering design scenarios. Our results show that ENGDESIGN
 270 presents a substantial challenge, even the strongest models achieve only modest performance, under-
 271 scoring the complexity of real-world engineering tasks. We hope ENGDESIGN serves as a valuable
 272 benchmark for future research at the intersection of language models and engineering intelligence,
 273 motivating continued progress toward the development of general-purpose AI engineers.

274 rates in the iterative experiments are slightly lower than the ones report in Table 2. Nonetheless, this subset
 275 demonstrates the effectiveness of feedback-driven refinement and highlights the potential of LLMs to improve
 276 design quality through iterative design.

274 **References**

- 275 [1] Wai Kai Chen. *The electrical engineering handbook*. Elsevier, 2004.
- 276 [2] Karl-Heinrich Grote and Hamid Hefazi. *Springer handbook of mechanical engineering*. Springer Nature, 277 2021.
- 278 [3] Wai-Fah Chen and JY Richard Liew. *The civil engineering handbook*. Crc Press, 2002.
- 279 [4] David Blockley. *Engineering: a very short introduction*, volume 309. Oxford university press, 2012.
- 280 [5] Chee-Wooi Ten and Yunhe Hou. *Modern power system analysis*. CRC Press, 2024.
- 281 [6] Saswata Paul, Elkin Cruz, Airin Dutta, Ankita Bhaumik, Erik Blasch, Gul Agha, Stacy Patterson, Fotis 282 Kopsaftopoulos, and Carlos Varela. Formal verification of safety-critical aerospace systems. *IEEE 283 Aerospace and Electronic Systems Magazine*, 38(5):72–88, 2023.
- 284 [7] Paul R Gray, Paul J Hurst, Stephen H Lewis, and Robert G Meyer. *Analysis and design of analog 285 integrated circuits*. John Wiley & Sons, 2009.
- 286 [8] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, 287 Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First 288 Conference on Language Modeling*, 2024.
- 289 [9] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming 290 Ren, Aaran Arulraj, Xuan He, Ziyian Jiang, et al. Mmlu-pro: A more robust and challenging multi-task 291 language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing 292 Systems Datasets and Benchmarks Track*, 2024.
- 293 [10] Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, 294 Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. 295 *arXiv preprint arXiv:2502.14739*, 2025.
- 296 [11] Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic 297 visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv 298 preprint arXiv:2411.00836*, 2024.
- 299 [12] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, 300 Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models 301 trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- 302 [13] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando 303 Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation 304 of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.
- 305 [14] Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, 306 Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on 307 machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024.
- 308 [15] Darioush Kevian, Usman Syed, Xingang Guo, Aaron Havens, Geir Dullerud, Peter Seiler, Lianhui Qin, 309 and Bin Hu. Capabilities of large language models in control engineering: A benchmark study on gpt-4, 310 claudie 3 opus, and gemini 1.0 ultra. *arXiv preprint arXiv:2404.03647*, 2024.
- 311 [16] Ming Li, Jike Zhong, Tianle Chen, Yuxiang Lai, and Konstantinos Psounis. Eee-bench: A comprehensive 312 multimodal electrical and electronics engineering benchmark. *arXiv preprint arXiv:2411.01492*, 2024.
- 313 [17] Lejla Skelic, Yan Xu, Matthew Cox, Wenjie Lu, Tao Yu, and Ruonan Han. Circuit: A benchmark for 314 circuit interpretation and reasoning capabilities of llms. *arXiv preprint arXiv:2502.07980*, 2025.
- 315 [18] Business Wire. P-1 AI Comes Out of Stealth, Aims to Build Engineering AGI for Physical Systems. 316 Press release, Business Wire, April 2025. San Francisco.
- 317 [19] Gordon W Roberts. *Spice*. Oxford University Press, Inc., 1996.
- 318 [20] SS Bhavikatti. *Finite element analysis*. New Age International, 2005.
- 319 [21] Richard Y Chiang and Michael G Safonov. Matlab. *Robust Control toolbox. User's guide (The Mathworks 320 Inc., 1998)*, 1984.

- 321 [22] Jason Liu and Contributors. Instructor: A library for structured outputs from large language models, 3
 322 2024.
- 323 [23] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob
 324 Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*,
 325 2020.
- 326 [24] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a
 327 benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*,
 328 2023.
- 329 [25] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang,
 330 Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity’s last exam. *arXiv preprint arXiv:2501.14249*,
 331 2025.
- 332 [26] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias
 333 Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word
 334 problems. *arXiv preprint arXiv:2110.14168*, 2021.
- 335 [27] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias,
 336 Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. Paperbench: Evaluating ai’s ability to replicate ai
 337 research. *arXiv preprint arXiv:2504.01848*, 2025.
- 338 [28] Usman Syed, Ethan Light, Xingang Guo, Huan Zhang, Lianhui Qin, Yanfeng Ouyang, and Bin Hu.
 339 Benchmarking the capabilities of large language models in transportation system engineering: Accuracy,
 340 consistency, and reasoning behaviors. *arXiv preprint arXiv:2408.08302*, 2024.
- 341 [29] Xingang Guo, Darioosh Keivan, Usman Syed, Lianhui Qin, Huan Zhang, Geir Dullerud, Peter Seiler, and
 342 Bin Hu. Controlagent: Automating control system design via novel integration of llm agents and domain
 343 expertise. *arXiv preprint arXiv:2410.19811*, 2024.
- 344 [30] Yao Lai, Sungyoung Lee, Guojin Chen, Souradip Poddar, Mengkang Hu, David Z Pan, and Ping Luo.
 345 Analogcoder: Analog circuit design via training-free code generation. *arXiv preprint arXiv:2405.14918*,
 346 2024.
- 347 [31] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind
 348 Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners.
 349 *Advances in neural information processing systems*, 33:1877–1901, 2020.
- 350 [32] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,
 351 Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv
 352 preprint arXiv:2303.08774*, 2023.
- 353 [33] Anthropic. Claude 3.7 sonnet system card.
- 354 [34] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan
 355 Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable
 356 multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- 357 [35] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi
 358 Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*,
 359 2024.
- 360 [36] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
 361 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural
 362 information processing systems*, 35:24824–24837, 2022.
- 363 [37] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans,
 364 Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in
 365 large language models. *arXiv preprint arXiv:2205.10625*, 2022.
- 366 [38] OpenAI. Openai o3 and o4-mini system card.
- 367 [39] Anthropic. Claude’s extended thinking. <https://www.anthropic.com/news/visible-extended-thinking>.
- 368
- 369 [40] Google DeepMind. Gemini: Our most capable model, updated with better thinking. <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking>.

- 372 [41] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong
 373 Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement
 374 learning. *arXiv preprint arXiv:2501.12948*, 2025.
- 375 [42] Ce Guo and Tong Zhao. Resbench: Benchmarking llm-generated fpga designs with resource awareness.
 376 *arXiv preprint arXiv:2503.08823*, 2025.
- 377 [43] Boyan Xu, Zihao Li, Yuxin Yang, Guanlan Wu, Chengzhi Wang, Xiongpeng Tang, Yu Li, Zihao Wu,
 378 Qingxian Su, Xueqing Shi, et al. Evaluating and advancing large language models for water knowledge
 379 tasks in engineering and research. *Environmental Science & Technology Letters*, 12(3):289–296, 2025.
- 380 [44] Ruoxin Xiong, Yanyu Wang, Suat Gunhan, Yimin Zhu, and Charles Berryman. Can ai master construction
 381 management (cm)? benchmarking state-of-the-art large language models on cm certification exams. *arXiv
 382 preprint arXiv:2504.08779*, 2025.
- 383 [45] Ahmadreza Eslaminia, Adrian Jackson, Beiting Tian, Avi Stern, Hallie Gordon, Rajiv Malhotra, Klara
 384 Nahrstedt, and Chenhui Shao. Fdm-bench: A comprehensive benchmark for evaluating large language
 385 models in additive manufacturing tasks. *arXiv preprint arXiv:2412.09819*, 2024.
- 386 [46] Itsuki Ogo and Masanobu Koga. Can chatgpt pass modern control theory exam? In *2024 24th International
 387 Conference on Control, Automation and Systems (ICCAS)*, pages 1287–1292. IEEE, 2024.
- 388 [47] Jayeeta Chaudhuri, Dhruv Thapar, Arjun Chaudhuri, Farshad Firouzi, and Krishnendu Chakrabarty.
 389 Spiced: Syntactical bug and trojan pattern identification in a/ms circuits using llm-enhanced detection.
 390 *arXiv preprint arXiv:2408.16018*, 2024.
- 391 [48] Debi Prasad Ghosh and Design Automation Team. Retrieval-augmented generation in engineering design,
 392 2024.
- 393 [49] Shadan Alsaqer, Sarah Alajmi, Imtiaz Ahmad, and Mohammad Alfailakawi. The potential of llms in
 394 hardware design. *Journal of Engineering Research*, 2024.
- 395 [50] Jiaxing Lu, Heran Li, Fangwei Ning, Yixuan Wang, Xinze Li, and Yan Shi. Constructing mechanical
 396 design agent based on large language models. *arXiv preprint arXiv:2408.02087*, 2024.
- 397 [51] Mengshuo Jia, Zeyu Cui, and Gabriela Hug. Enabling large language models to perform power system
 398 simulations with previously unseen tools: A case of daline. *arXiv preprint arXiv:2406.17215*, 2024.
- 399 [52] Hendrik F Hamann, Thomas Brunschwiler, Blazhe Gjorgiev, Leonardo SA Martins, Alban Puech, Anna
 400 Varbella, Jonas Weiss, Juan Bernabe-Moreno, Alexandre Blondin Massé, Seong Choi, et al. A perspective
 401 on foundation models for the electric power grid. *arXiv preprint arXiv:2407.09434*, 2024.
- 402 [53] Boyan Xu, Liang Wen, Zihao Li, Yuxing Yang, Guanlan Wu, Xiongpeng Tang, Yu Li, Zihao Wu,
 403 Qingxian Su, Xueqing Shi, et al. Unlocking the potential: Benchmarking large language models in water
 404 engineering and research. *arXiv preprint arXiv:2407.21045*, 2024.
- 405 [54] Subir Majumder, Lin Dong, Fatemeh Doudi, Yuting Cai, Chao Tian, Dileep Kalathil, Kevin Ding,
 406 Anupam A Thatte, Na Li, and Le Xie. Exploring the capabilities and limitations of large language models
 407 in the electric energy sector. *Joule*, 8(6):1544–1549, 2024.
- 408 [55] Liane Makatura, Michael Foshey, Bohan Wang, Felix Hähnlein, Pingchuan Ma, Bolei Deng, Megan
 409 Tjandrasuwita, Andrew Spielberg, Crystal Elaine Owens, Peter Yichen Chen, et al. How can large
 410 language models help humans in design and manufacturing? *arXiv preprint arXiv:2307.14377*, 2023.
- 411 [56] Bailey Brown, Nadim S Hmeidat, Xiu Jia, Jackson Wilt, Michael Roberts, Brett G Compton, and
 412 Natasha Vermaak. Experimental investigations of the effectiveness of simultaneous topology/orientation
 413 optimization via somp and principal stress directions. *Materials & Design*, 217:110647, 2022.
- 414 [57] Andrew Waterman, Yunsup Lee, Rimas Avizienis, David A Patterson, and Krste Asanovic. The risc-v
 415 instruction set manual volume ii: Privileged architecture version 1.7. *EECS Department, University of
 416 California, Berkeley, Tech. Rep. UCB/EECS-2016-129*, 2016.
- 417 [58] Kirill Levchenko. File system. Lecture slides, ECE391 Course Materials, University of Illinois Ur-
 418 bana–Champaign, Spring 2025.
- 419 [59] Michael S Tsirkin and Cornelia Huck. Virtual i/o device (virtio) version 1.1. *OASIS Committee*, 2022.
- 420 [60] RISC-V Privileged Horizontal Committee. The risc-v instruction set manual volume ii: Privileged ar-
 421 chitecture. https://drive.google.com/file/d/17GeetSnT5w3xNuAHI95-ST1gPGd5sJ/_view,
 422 2024.

- 423 [61] Kirill Levchenko. Virtual memory. Lecture slides, ECE391 Course Materials, University of Illinois
 424 Urbana–Champaign, Fall 2024.
- 425 [62] Wikipedia contributors. Round-robin scheduling. *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Round-robin_scheduling.
- 426 [63] Wikipedia contributors. Clustered file system. *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Clustered_file_system.
- 427 [64] Richard Gooch and Pekka Enberg. Overview of the linux virtual file system. <https://docs.kernel.org/filesystems/vfs.html>, 2005.
- 428 [65] Junghee Lee, Youngjae Kim, Galen M Shipman, Sarp Oral, and Jongman Kim. Preemptible i/o scheduling
 429 of garbage collection for solid state drives. *IEEE Transactions on Computer-Aided Design of Integrated
 430 Circuits and Systems*, 32(2):247–260, 2013.
- 431 [66] Zhe Jia, Blake Tillman, Marco Maggioni, and Daniele Paolo Scarpazza. Dissecting the graphcore ipu
 432 architecture via microbenchmarking, 2019.
- 433 [67] Simon Knowles. Graphcore. In *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–25, 2021.
- 434 [68] Sean Lie. Multi-million core, multi-wafer ai cluster. In *2021 IEEE Hot Chips 33 Symposium (HCS)*,
 435 pages 1–41, 2021.
- 436 [69] Yiqi Liu, Yuqi Xue, Yu Cheng, Lingxiao Ma, Ziming Miao, Jilong Xue, and Jian Huang. Scaling deep
 437 learning computation over the inter-core connected intelligence processor with t10. In *Proceedings of
 438 the ACM SIGOPS 30th Symposium on Operating Systems Principles*, SOSP ’24, page 505–521. ACM,
 November 2024.
- 439 [70] Raghu Prabhakar and Sumti Jairath. Sambanova sn10 rdu:accelerating software 2.0 with dataflow. In
 440 *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–37, 2021.
- 441 [71] Nicolas Vasilache, Oleksandr Zinenko, Theodoros Theodoridis, Priya Goyal, Zachary DeVito, William S.
 442 Moses, Sven Verdoolaege, Andrew Adams, and Albert Cohen. Tensor comprehensions: Framework-
 443 agnostic high-performance machine learning abstractions, 2018.
- 444 [72] Lianmin Zheng, Chengfan Jia, Minmin Sun, Zhao Wu, Cody Hao Yu, Ameer Haj-Ali, Yida Wang,
 445 Jun Yang, Danyang Zhuo, Koushik Sen, Joseph E. Gonzalez, and Ion Stoica. Ansor: Generating
 446 high-performance tensor programs for deep learning, 2023.
- 447 [73] Hongyu Zhu, Ruofan Wu, Yijia Dia, Shanbin Ke, Haoyu Li, Chen Zhang, Jilong Xue, Lingxiao Ma,
 448 Yuqing Xia, Wei Cui, Fan Yang, Mao Yang, Lidong Zhou, Asaf Cidon, and Gennady Pekhimenko.
 449 ROLLER: Fast and efficient tensor compilation for deep learning. In *16th USENIX Symposium on
 450 Operating Systems Design and Implementation (OSDI 22)*, pages 233–248, Carlsbad, CA, July 2022.
 USENIX Association.
- 451 [74] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay
 452 Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A
 453 Patterson. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware
 454 support for embeddings. In *Proceedings of the 50th Annual International Symposium on Computer
 455 Architecture*, ISCA ’23, New York, NY, USA, 2023. Association for Computing Machinery.
- 456 [75] Thomas Norrie, Nishant Patil, Doe Hyun Yoon, George Kurian, Sheng Li, James Laudon, Cliff Young,
 457 Norman Jouppi, and David Patterson. The design process for google’s training chips: Tpuv2 and tpuv3.
 458 *IEEE Micro*, 41(2):56–63, 2021.
- 459 [76] Yazhou Zu, Alireza Ghaffarkhah, Hoang-Vu Dang, Brian Towles, Steven Hand, Safeen Huda, Adekunle
 460 Bello, Alexander Kolbasov, Arash Rezaei, Dayou Du, Steve Lacy, Hang Wang, Aaron Wisner, Chris Lewis,
 461 and Henri Bahini. Resiliency at scale: Managing Google’s TPUv4 machine learning supercomputer. In
 462 *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 761–774,
 463 Santa Clara, CA, April 2024. USENIX Association.
- 464 [77] Yuqi Xue, Yiqi Liu, Lifeng Nai, and Jian Huang. V10: Hardware-assisted npu multi-tenancy for improved
 465 resource utilization and fairness. In *Proceedings of the 50th Annual International Symposium on Computer
 466 Architecture*, ISCA ’23, New York, NY, USA, 2023. Association for Computing Machinery.
- 467 [78] Yuqi Xue, Yiqi Liu, Lifeng Nai, and Jian Huang. Hardware-assisted virtualization of neural processing
 468 units for cloud platforms. In *2024 57th IEEE/ACM International Symposium on Microarchitecture
 (MICRO)*, pages 1–16, 2024.

- 475 [79] Mohammad Shoeybi, Mostafa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro.
 476 Megatron-Lm: Training multi-billion parameter language models using model parallelism, 2020.
- 477 [80] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and
 478 memory-efficient exact attention with io-awareness, 2022.
- 479 [81] Saeed Rashidi, William Won, Sudarshan Srinivasan, Srinivas Sridharan, and Tushar Krishna. Themis: a
 480 network bandwidth-aware collective scheduling policy for distributed training of dl models. In *Proceedings*
 481 of the 49th Annual International Symposium on Computer Architecture, ISCA '22, page 581–596. ACM,
 482 June 2022.
- 483 [82] The JAX Authors. Matrix multiplication. [https://docs.jax.dev/en/latest/pallas/tpu/
 484 matmul.html](https://docs.jax.dev/en/latest/pallas/tpu/matmul.html), 2024.
- 485 [83] Kun Wu Steven S. Lumetta. Ece 508: Manycore parallel algorithms (spring 2023) - lecture 4. <https://lumetta.web.engr.illinois.edu/508/slides/lecture4.pdf>, 2023.
- 487 [84] Tamer Başar, Sean P. Meyn, and William R. Perkins. Lecture notes on control system theory and design.
 488 *arXiv preprint arXiv:2007.01367*, 2024.
- 489 [85] M. Safonov, A. Laub, and G. Hartmann. Feedback properties of multivariable systems: The role and use
 490 of the return difference matrix. *IEEE Transactions on Automatic Control*, 26(1):47–65, 1981.
- 491 [86] S. Salapaka, A. Sebastian, J. P. Cleveland, and M. V. Salapaka. High bandwidth nano-positioner: A robust
 492 control approach. *Review of Scientific Instruments*, 73(9):3232–3241, 09 2002.
- 493 [87] Matlab and simulink examples. <https://www.mathworks.com/academia/examples.html>, 2025.
- 494 [88] Peter Seiler and Julian Theis. *An Introduction to Classical Control and Loopshaping*.
- 495 [89] University of Michigan. Control tutorials for matlab and simulink - pid controller design. <https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID>.
- 497 [90] Eugene Lavretsky and Kevin A. Wise. *Frequency Domain Analysis*, pages 103–176. Springer International
 498 Publishing, Cham, 2024.
- 499 [91] Alejandro Dominguez-Garcia. Ece 486: Real-time operating systems and networks (spring 2025).
 500 <https://courses.grainger.illinois.edu/ece486/sp2025/>, 2025.
- 501 [92] David Strassera. Openlapsim. <https://github.com/dstrassera/OpenLapSim>, 2021.
- 502 [93] Kenneth Sutton and Randolph A. Graves. A general stagnation-point convective heating equation for
 503 arbitrary gas mixtures. 1971.
- 504 [94] François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization, 2022.
- 505 [95] François Mazé and Faez Ahmed. Topodiff project page. [https://decode.mit.edu/projects/
 506 topodiff/](https://decode.mit.edu/projects/topodiff/), 2022.
- 507 [96] Bram Lagerweij. Topopt: Topology optimization in python. [https://github.com/AJLagerweij/
 508 topopt](https://github.com/AJLagerweij/topopt), 2024.
- 509 [97] IEEE. Ieee standard for systemverilog-unified hardware design, specification, and verification language.
 510 *IEEE Std 1800-2017 (Revision of IEEE Std 1800-2012)*, pages 1–1315, 2018.
- 511 [98] Tetris Wiki contributors. Tetris guideline. https://tetris.fandom.com/wiki/Tetris_Guideline.
- 512 [99] FPGA4student. Fpga verilog code for tetris game. [https://www.fpga4student.com/2017/08/
 513 fpga-verilog-code-for-tetris-game.html](https://www.fpga4student.com/2017/08/fpga-verilog-code-for-tetris-game.html), 2017.
- 514 [100] P.P. Chu. *FPGA Prototyping by SystemVerilog Examples: Xilinx MicroBlaze MCS SoC Edition*. Wiley,
 515 2018.
- 516 [101] FPGA4student. Seven-segment led display controller in verilog. [https://www.fpga4student.com/
 517 2017/09/seven-segment-led-display-controller-verilog.html](https://www.fpga4student.com/2017/09/seven-segment-led-display-controller-verilog.html), 2017.
- 518 [102] Jean P. Nicolle. Pong game. <https://www.fpga4fun.com/PongGame.html>.
- 519 [103] J.L. Hennessy and D.A. Patterson. *Computer Architecture: A Quantitative Approach*. The Morgan
 520 Kaufmann Series in Computer Architecture and Design. Morgan Kaufmann, 2017.

- 521 [104] University of Illinois Urbana-Champaign ECE Department. Ece 385: Digital systems laboratory (spring
522 2025) - lab 5 specification. <https://courses.engr.illinois.edu/ece385>, 2025.
- 523 [105] University of Illinois Urbana-Champaign ECE Department. Ece 385: Digital systems laboratory (spring
524 2025) - course slides: Cpu control logic. <https://courses.engr.illinois.edu/ece385>, 2025.
- 525 [106] Chipdev platform. <https://chipdev.io/>, 2025.
- 526 [107] ECE Department, University of Illinois Urbana-Champaign. ECE 483: Analog IC Design (Spring 2025) -
527 Course Project Material: MP4. <https://courses.engr.illinois.edu/ece483/sp2025/>, 2025.
- 528 [108] ECE Department, University of Illinois Urbana-Champaign. ECE 483: Analog IC Design (Spring 2025) -
529 Course Project Material: MP3. <https://courses.engr.illinois.edu/ece483/sp2025/>, 2025.
- 530 [109] ECE Department, University of Illinois Urbana-Champaign. ECE 483: Analog IC Design (Spring 2025) -
531 Course Project Material: MP1. <https://courses.engr.illinois.edu/ece483/sp2025/>, 2025.
- 532 [110] Huan Weng, Bill Hunt, Jarvis Schultz, and Mikhail Todes. Modern robotics: Mechanics, planning, and
533 control – code library. <https://github.com/NxRLab/ModernRobotics>, 2018.
- 534 [111] Antenna-Theory.com. Patch antennas, 2011.
- 535 [112] K R Kashwan, V Rajeshkumar, T Gunasekaran, and K R Shankar Kumar. Design and characterization
536 of pin fed microstrip patch antennae. In *2011 Eighth International Conference on Fuzzy Systems and*
537 *Knowledge Discovery (FSKD)*, volume 4, pages 2258–2262, 2011.
- 538 [113] Antenna-Theory.com. The dipole antenna, 2011.
- 539 [114] Joseph Howse and Joe Minichino. *Learning OpenCV 4 Computer Vision with Python 3*. Packt Publishing,
540 3rd edition, 2020.
- 541 [115] Corey Snyder Zhi-Pei Liang. Ece 310: Digital signal processing (spring 2025). <https://courses.grainger.illinois.edu/ece310/sp2025/>, 2025.
- 543 [116] Xavier Soria, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn
544 model for edge detection. In *The IEEE Winter Conference on Applications of Computer Vision (WACV*
545 '20), 2020.

546 A Full Author List and Contributions

547 The ENGDESIGN benchmark was developed through a large-scale collaborative effort involving over
548 60 contributors. Team members contributed across the following key areas:

- | | | |
|---------------------------------------|-----|--------------------|
| 549 • Benchmark Framework Design | 552 | • Model Evaluation |
| 550 • Task Contributions | 553 | • Result Analysis |
| 551 • Task Quality Inspection | 554 | • Paper Writing |

555 Core Contributors:

- | | | |
|---------------------------------------|-----|----------------------------|
| 556 • Xingang Guo ¹ | 560 | • Xiayu Zhao ¹ |
| 557 • Yaxin Li ¹ | 561 | • Zhihua Gong ¹ |
| 558 • Xiang Yi Kong ¹ | 562 | • Yufan Zhang ¹ |
| 559 • Yilan Jiang ¹ | | |

563 Contributors:

- | | | |
|--|-----|---------------------------------------|
| 564 • Daixuan Li ¹ | 586 | • Chongying Yue ¹ |
| 565 • Tianle Sang ¹ | 587 | • Jingjie He ¹ |
| 566 • Beixiao Zhu ¹ | 588 | • Libin Wang ¹ |
| 567 • Gregory Jun ¹ | 589 | • Licheng Xu ¹ |
| 568 • Yingbing Huang ¹ | 590 | • Hao Chen ¹ |
| 569 • Yiqi Liu ¹ | 591 | • Jinwen Wang ¹ |
| 570 • Yuqi Xue ¹ | 592 | • Liujun Xu ¹ |
| 571 • Rahul Dev Kundu ¹ | 593 | • Rushabh Shetty ¹ |
| 572 • Qi Jian Lim ¹ | 594 | • Ziheng Guo ¹ |
| 573 • Yizhou Zhao ² | 595 | • Dahui Song ¹ |
| 574 • Luke Alexander Granger ¹ | 596 | • Manvi Jha ¹ |
| 575 • Mohamed Badr Younis ¹ | 597 | • Weijie Liang ¹ |
| 576 • Darioush Keivan ¹ | 598 | • Weiman Yan ¹ |
| 577 • Nippun Sabharwal ¹ | 599 | • Bryan Zhang ¹ |
| 578 • Shreyanka Sinha ¹ | 600 | • Sahil Bhandary Karnoor ¹ |
| 579 • Prakhar Agarwal ¹ | 601 | • Jialiang Zhang ¹ |
| 580 • Kojo Vandyck ¹ | 602 | • Rutva Pandya ¹ |
| 581 • Hanlin Mai ¹ | 603 | • Xinyi Gong ¹ |
| 582 • Zichen Wang ¹ | 604 | • Mithesh Ballae Ganesh ¹ |
| 583 • Aditya Venkatesh ¹ | 605 | • Feize Shi ¹ |
| 584 • Ayush Barik ¹ | 606 | • Ruiling Xu ¹ |
| 585 • Jiankun Yang ¹ | 607 | • Yifan Zhang ¹ |

608 Advisors and Senior Contributors:

- | | | |
|---|-----|------------------------------|
| 609 • Yanfeng Ouyang ¹ | 612 | • Corey Snyder ¹ |
| 610 • Lianhui Qin ³ | 613 | • Peter Seiler ⁴ |
| 611 • Elyse Rosenbaum ¹ | 614 | • Geir Dullerud ¹ |

- | | | |
|--|--------------------------|--|
| 615 • Xiaojia Zhang ¹
616 • Zuofu Cheng ¹
617 • Pavan Kumar Hanumolu ¹
618 • Jian Huang ¹ | 619
620
621
622 | • Mayank Kulkarni ⁵
• Mahdi Namazifar ⁵
• Huan Zhang ¹
• Bin Hu ¹ |
|--|--------------------------|--|

623 **Affiliations:**

- 624 1. University of Illinois Urbana-Champaign
625 2. University of Pennsylvania
626 3. University of California San Diego
627 4. University of Michigan
628 5. Amazon AGI

629 **B Limitations and Broader Impact**

630 **Limitations**

631 While ENGDESIGN provides a comprehensive benchmark for evaluating LLMs in engineering design
632 tasks, some limitations remain:

- 633 1. ENGDESIGN currently covers 9 engineering design domains. Although these are diverse
634 and representative, the benchmark does not yet encompass the full breadth of engineering
635 disciplines or subfields.
- 636 2. Our evaluation includes a selection of widely used and high-performing LLMs. However, it
637 is not exhaustive. Many emerging or smaller-scale models are not included in our evaluation.
- 638 3. We employ the `instructor` framework to enforce structured output from LLMs. While
639 this tool has shown reliable performance throughout our evaluations, there are instances
640 where LLMs fail to adhere to the expected output format. As `instructor` is still under
641 active development, future improvements will be necessary to further enhance its robustness
642 and compatibility across models.

643 **Broader Impact**

644 This work aims to advance our understanding of how LLMs perform in complex, real-world engi-
645 neering design scenarios. However, we highlight the following considerations:

- 646 • **Not a replacement for human engineers.** Our benchmark is intended to assess LLM
647 capabilities, not to replace the expertise and judgment of professional engineers. Engineering
648 design involves safety-critical decision-making, physical prototyping, and domain-specific
649 knowledge that current LLMs cannot fully replicate.
- 650 • **Supporting real-world design workflows.** We hope that LLMs, when properly evaluated
651 and deployed, can assist engineers by automating routine tasks, enhancing ideation, and
652 accelerating early-stage design processes—particularly in settings with limited access to
653 expert resources.
- 654 • **Need for rigorous safety checks.** Any LLM-generated design, particularly those intended
655 for deployment in safety-critical domains such as aerospace, biomedical devices, civil infras-
656 tructure, or autonomous systems, must undergo thorough validation and safety assessment
657 by qualified human experts before real-world application.
- 658 • **Ethical use and accessibility.** Care must be taken to ensure that advances in AI-assisted
659 engineering design benefit a broad and diverse community. Efforts should be made to avoid
660 misuse, bias amplification, or overreliance on unverified outputs.

661 **C More on Related Work**

662 **Large Language Models.** Recent advances in large language models (LLMs) have been driven
663 by scaling, instruction tuning, and improvements in reasoning. Models such as GPT-3 and GPT-
664 4 [31, 32] have demonstrated strong general capabilities. Proprietary models including Anthropic’s
665 Claude [33], Google’s Gemini [34], and DeepSeek [35] show competitive results across a wide range
666 of benchmarks.

667 Recent research has focused on enhancing the reasoning capabilities of LLMs through architectural
668 innovations, training paradigms, and inference-time prompting strategies. Techniques such as Chain-
669 of-Thought prompting [36] and least-to-most prompting [37] have demonstrated that reasoning
670 performance can be significantly improved through intermediate step decomposition. These advances
671 have led to the development of a new generation of reasoning-focused models, including the o-series
672 from OpenAI [38], Claude models with “thinking” mode [39], Gemini 2.5 Pro [40], and DeepSeek-
673 R1 [41], all of which achieve strong results on standard benchmarks. However, their effectiveness
674 in domain-specific applications, such as engineering design, remains underexplored. ENGDESIGN
675 addresses this gap by evaluating both general-purpose and reasoning-specialized LLMs in realistic,
676 simulation-based engineering workflows.

677 **General Purpose Benchmarks for LLMs.** Numerous benchmarks have been developed to eval-
678 uate the general reasoning and problem-solving capabilities of LLMs. Notable examples include
679 MMLU [23] and MMLU-Pro [9], which assess knowledge across a wide range of subjects; GAIA [24],
680 which focuses on long-context retrieval and reasoning; and HLE [25], which introduces highly chal-
681 lenging questions spanning diverse domains. HumanEval [12] evaluates the functional correctness
682 of LLM-generated code, while GPQA [8] targets graduate-level physics and science problems.
683 GSM8k [26] emphasizes grade-school math word problems to test arithmetic and reasoning. More
684 recently, DynaMath [11] proposes a dynamic visual math benchmark that evaluates LLM robust-
685 ness across varied visual and numeric transformations, and PaperBench [27] offers paper-based
686 reproduction tasks to assess complex agentic workflows.

687 **LLMs in Engineering and Scientific Domains.** Recent studies have started to explore the ap-
688 plicability of LLMs to engineering contexts. For example, [28, 15, 42, 43, 44, 45, 46] evaluate
689 LLMs across engineering subfields using curated QA datasets. However, these efforts only focuses
690 one specific domain or largely focus on factual recall rather than generative design capabilities.
691 Several domain-specific studies examine LLM-assisted design workflows. For example, ControlA-
692 gent [29] integrates control theory solvers for automated controller design, while AnalogCoder [30]
693 and SPICED [47] target analog circuit and SoC design using prompt engineering and retrieval-based
694 techniques. Retrieval-augmented and programmatic strategies have also been explored to enhance
695 LLM performance [48, 49]. Other studies examine LLMs in mechanical design [50], cross-domain
696 creativity [51], and computational engineering [52, 53, 54, 55].

697 **D More on ENGDESIGN**

698 **D.1 Prompt Token Length Comparison**

699 Table 3 reports the average number of tokens in the input prompts for various benchmarks, measured
700 using the o200k_base tokenizer. Notably, ENGDESIGN exhibits significantly longer prompts
701 (averaging 778.71 tokens) compared to other popular QA-style benchmarks such as MMLU-Pro
702 (61.76), HLE (250.03), and GSM8K (58.46). This reflects the greater contextual and structural
703 complexity involved in realistic engineering design tasks, which often require extensive problem
704 descriptions and domain-specific constraints.

Table 3: Average token counts across different benchmarks using the o200k_base tokenizer.

Benchmark	MMLU-Pro	HLE	SuperGPQA	DynaMATH	GSM8K	EngDesign
Average Token #	61.76	250.03	57.89	27.83	58.46	778.71

705 D.2 More on construction of ENGDESIGN

As described in Section 2.3, the development of ENGDESIGN follows a rigorous multi-stage process, including initial task design and two rounds of review to ensure task quality and relevance. During the initial stage, over 40 task contributors submitted a total of 186 task proposals. Following thorough evaluation and revision across the two review phases, 101 tasks were accepted for inclusion in ENGDESIGN, while the remaining 85 were discarded due to various issues such as insufficient clarity, infeasibility, or misalignment with the benchmark’s design-oriented goals.

712 D.3 Word Clouds of ENGDESIGN

Figure 7 and Figure 8 present the word clouds of the most frequently occurring terms across all prompts and within each domain, respectively. These visualizations underscore the benchmark’s emphasis on design-specific vocabulary and highlight the diversity of engineering contexts it encompasses.

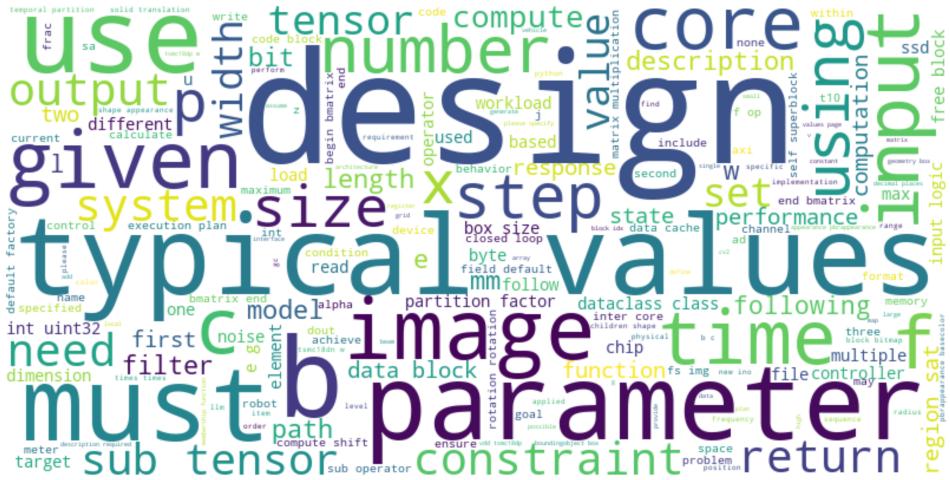


Figure 7: Word Cloud of the overall ENGDESIGN Prompts.

717 E More on Evaluation

Figure 9 illustrates the benchmarking pipeline for evaluating LLMs on engineering design tasks. In this section, we provide more details on the evaluations of ENGDESIGN.

720 E.1 Detailed Experimental Setup

We use the default configuration for each LLM during evaluation. Table 4 provides detailed information on the model names, setup parameters, and special configurations used for each model.

All evaluations in this work were conducted via API access to proprietary LLMs (e.g., OpenAI, Anthropic, Google, and DeepSeek). As a result, the computational burden on our end was minimal, with no need for local GPU infrastructure. The primary cost was associated with inference-time API usage, which varies depending on the model types and token numbers. Since we did not perform model training or fine-tuning, the overall computational footprint of our study remains low.

728 E.2 LLM Response Structure

As discussed in Section 2.2, we use `instructor` to construct LLM responses by defining a response structure class, as illustrated in Figure 10. For each task, the response structure class includes two keys: `reasoning` and `config`. The `reasoning` component prompts the LLM to perform a detailed step-by-step reasoning process to complete the task. The `config` component defines a task-specific



Figure 8: Word clouds of ENGDESIGN for each engineering domain.

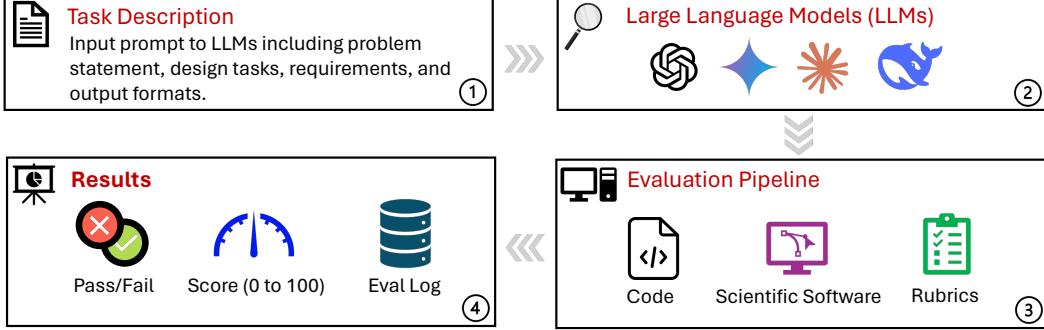


Figure 9: Overview of the ENGDESIGN Benchmarking Pipeline. The process begins with a task description curated by task contributors (Step 1). This prompt is then submitted to various LLMs (e.g., GPT, Claude, Gemini, DeepSeek) to generate candidate design proposals (Step 2). The generated outputs are evaluated using an automated codebase that includes code execution, integration with scientific software (e.g., MATLAB, Webots), and standardized scoring rubrics (Step 3). Finally, the results, including pass/fail status, quantitative scores, and detailed evaluation logs, are recorded for further analysis (Step 4).

Table 4: Model names and configuration details for each evaluated LLM.

Model	Model Pointer & Setup
GPT-4o	gpt-4o-2024-08-06
o1	o1-2024-12-17
o3	o3-2025-04-16, reasoning effort: medium/high
o4-mini	o4-mini-2025-04-16, reasoning effort: medium/high
Claude-3.7-Sonnet	claude-3-7-sonnet-20250219
Claude-3.7-Thinking	claude-3-7-sonnet-20250219, thinking token budget: 16,000
Gemini-2.0-Flash	gemini-1.5-pro
Gemini-2.5-Pro	gemini-2.5-pro-preview-03-25
DeepSeek-v3	DeepSeek-V3
DeepSeek-R1	DeepSeek-R1

733 ConfigFile class, which guides the LLM to produce key design components in a predefined format.
 734 This structured output facilitates automated evaluation by making it easy to parse LLMs' design
 735 choices or code snippets.

736 E.3 More on Evaluation Metrics

737 We evaluate model performance using three metrics: Average Pass Rate, Average Score, and Average
 738 Response Length. Each task is tested over N ($N = 3$) independent trials per model. For each metric,
 739 we aggregate at the task level and then summarize across all tasks to ensure equal weighting.

740 **Average Pass Rate** For each task i , we define the pass rate as:

$$\text{pass_rate}_i = \frac{\text{number of successful trials}}{N}$$

741 The overall pass rate is then computed as the mean of per-task pass rates:

$$\mu_{\text{pass}} = \frac{1}{N} \sum_{i=1}^N \text{pass_rate}_i$$

```

import instructor
from pydantic import BaseModel, Field

class ConfigFile(BaseModel):
    theta: float = Field(description="The value of theta")
    tau: float = Field(description="The value of tau")
    num: list[float] = Field(description="The numerator of the transfer function of the
controller")
    den: list[float] = Field(description="The denominator of the transfer function of the
controller")

    # Define your desired output structure
    class Response_structure(BaseModel):
        reasoning: str = Field(..., description="Detailed reasoning process to accomplish the
task, please solve all the tasks step by step")
        config: ConfigFile

```

Figure 10: An example of response structure class for `instructor` (Task XG_05).

742 We report μ_{pass} along with its standard deviation across tasks:

$$\sigma_{\text{pass}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\text{pass_rate}_i - \mu_{\text{pass}})^2}$$

743 **Average Score** For each task i , the score is averaged over N trials:

$$\text{score}_i = \frac{1}{N} \sum_{j=1}^3 x_{i,j}$$

744 where $x_{i,j}$ is the score in trial j . The overall score is the mean of per-task scores:

$$\mu_{\text{score}} = \frac{1}{N} \sum_{i=1}^N \text{score}_i$$

745 with corresponding standard deviation:

$$\sigma_{\text{score}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\text{score}_i - \mu_{\text{score}})^2}$$

746 **Reasoning Robustness** Let T denote the set of all evaluated tasks. For a given model, each task
747 $t \in T$ is evaluated over N independent trials. Define:

- 748 • N_{all} : the number of tasks for which all N trials passed, and
- 749 • N_{any} : the number of tasks for which at least one trial passed.

750 Then, the *Reasoning Robustness* $R \in [0, 1]$ is defined as:

$$R = \frac{N_{\text{all}}}{N_{\text{any}}}$$

751 This metric quantifies the model's consistency in reasoning. A value of $R = 1$ indicates perfect
752 robustness. In other words, whenever a task is solvable by the model, it succeeds consistently across
753 all trials. On the other hand, lower values indicate higher variance or instability in the model's
754 reasoning process across repeated attempts.

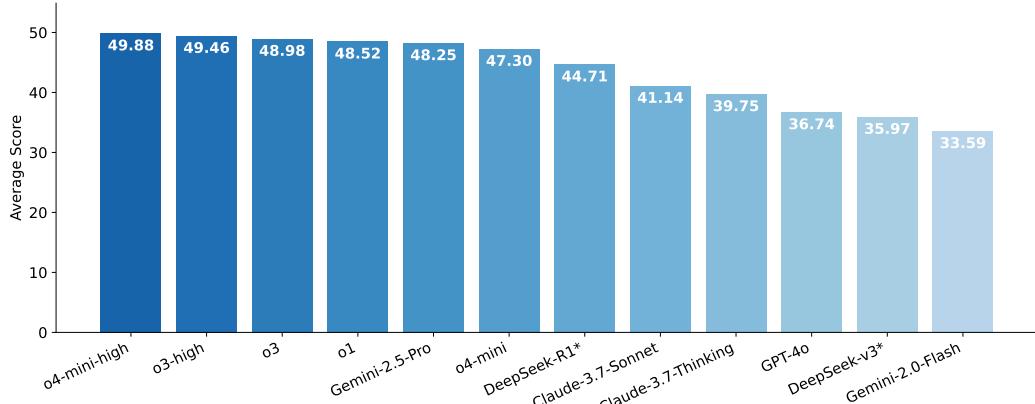


Figure 11: Average Score of evaluated LLMs on ENGDESIGN.

Table 5: Average score results (each task evaluated over 3 trials) under different domains.

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
Chat Models										
GPT-4o	36.74	25.71	18.15	35.30	41.69	29.05	26.33	55.80	41.45	34.60
DeepSeek-v3*	35.97	17.33	0.00	45.58	39.67	24.05	25.94	48.47	34.55	38.68
Gemini-2.0-Flash	33.59	7.00	0.00	39.45	38.25	27.14	28.15	53.47	30.03	31.51
Claude-3.7-Sonnet	41.14	31.00	23.79	43.94	46.03	30.24	29.55	58.97	44.90	37.11
Reasoning Models										
o1	48.52	19.00	49.09	47.48	63.28	40.48	58.35	57.22	52.61	32.57
o3	48.98	20.33	35.82	59.37	28.82	40.24	54.52	70.13	57.44	44.77
o3-high	49.46	20.33	32.80	53.19	33.36	45.24	65.59	73.67	53.51	47.48
o4-mini	47.30	11.00	25.72	48.80	57.36	35.00	56.90	61.38	51.28	42.07
o4-mini-high	49.88	12.67	36.97	54.87	64.28	36.43	63.96	56.72	48.45	44.15
Gemini-2.5-Pro	48.25	31.00	0.00	53.70	58.05	29.05	44.05	62.43	39.63	55.02
DeepSeek-R1*	44.71	22.00	0.00	56.27	54.38	26.19	37.26	53.82	48.08	40.31
Claude-3.7-Thinking	39.75	22.67	26.02	43.87	29.59	37.62	28.25	59.38	45.58	40.39

755 E.4 Additional Evaluation Results

756 In this section, we present additional evaluation results. Table 5 provide the average score results of
757 evaluated LLMs on ENGDESIGN.

758 Figure 11 presents the average score distribution across all evaluated models. The overall ranking is
759 consistent with the pass rate results in Table 2, with reasoning-focused models such as o4-mini-high,
760 o3-high, and o3 achieving the highest scores. Notably, the score differences are less pronounced than
761 the corresponding differences in pass rates, suggesting that while these models may not consistently
762 meet strict pass criteria, they still generate partially correct or near-complete solutions that earn
763 substantial credit. In addition, Table 6 and Table 7 show the average pass and score for text-only
764 tasks. Table 8 and Table 9 present the evaluation results for ENGDESIGN-OPEN. Finally, we provide
765 the results with error bar in Table 10 and Table 11.

766 E.5 Token Consumption

767 Figure 12 to Figure 15 show the token consumption for the evaluated LLMs under 9 topics within
768 ENGDESIGN.

769 E.6 Prompt

770 In this section, we provide the LLMs prompts we used for task filtering (the second stage of
771 ENGDESIGN Section 2.3) and iterative design (Section 3.2).

Table 6: Average pass rate (%) results (text-only tasks).

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
Chat Models										
GPT-4o	14.98	0.00	20.00	20.83	10.26	0.00	4.17	26.67	16.67	27.78
DeepSeek-v3*	18.42	0.00	0.00	33.33	25.64	0.00	0.00	16.67	13.89	38.89
Gemini-2.0-Flash	13.12	0.00	0.00	4.17	16.67	4.76	0.00	20.00	5.56	36.11
Claude-3.7-Sonnet	21.97	0.00	40.00	25.00	33.33	14.29	0.00	33.33	13.89	33.33
Reasoning Models										
o1	28.95	0.00	40.00	25.00	41.03	14.29	37.50	50.00	22.22	25.00
o3	34.38	0.00	40.00	20.51	20.51	23.81	25.00	63.33	41.67	41.67
o3-high	34.51	0.00	40.00	33.33	20.51	28.57	30.43	68.97	36.11	44.44
o4-mini	31.58	0.00	33.33	20.83	41.03	19.05	29.17	43.33	30.56	44.44
o4-mini-high	33.78	0.00	40.00	20.83	47.22	19.05	37.50	46.67	36.11	38.89
Gemini-2.5-Pro	28.89	0.00	0.00	25.00	43.59	0.00	9.52	56.67	13.89	50.00
DeepSeek-R1*	23.77	0.00	36.36	20.83	38.46	4.76	5.26	26.67	22.22	41.67
Claude-3.7-Thinking	18.30	0.00	33.33	20.83	17.95	9.52	0.00	40.00	11.11	30.56

Table 7: Average score results (text-only results).

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
Chat Models										
GPT-4o	37.57	25.71	0.00	43.75	41.69	29.05	26.33	55.80	36.59	34.98
DeepSeek-v3*	37.20	17.33	0.00	60.17	39.67	24.05	25.94	48.47	34.93	38.68
Gemini-2.0-Flash	33.87	7.00	0.00	42.58	38.25	27.14	28.15	53.47	30.86	31.51
Claude-3.7-Sonnet	41.39	31.00	0.00	48.54	46.03	30.24	29.55	58.97	42.54	37.70
Reasoning Models										
o1	48.94	19.00	0.00	53.92	63.28	40.48	58.35	57.22	54.59	32.79
o3	48.02	20.33	0.00	60.58	28.82	40.24	54.52	70.13	59.76	46.00
o3-high	50.39	20.33	0.00	58.08	33.36	45.24	65.59	73.67	56.44	48.93
o4-mini	48.18	11.00	0.00	50.04	57.36	35.00	56.90	61.38	52.46	43.08
o4-mini-high	49.76	12.67	0.00	49.96	64.28	36.43	63.96	56.72	51.65	45.30
Gemini-2.5-Pro	48.33	31.00	0.00	50.50	58.05	29.05	44.05	62.43	42.93	55.02
DeepSeek-R1*	44.24	22.00	0.00	53.21	54.38	26.19	37.26	53.82	50.42	40.31
Claude-3.7-Thinking	39.47	22.67	0.00	47.71	29.59	37.62	28.25	59.38	43.92	41.25

Table 8: Average pass rate (%) results for ENGDESIGN-OPEN.

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
Chat Models										
GPT-4o	14.74	N/A	20.00	16.67	11.11	0.00	4.17	29.63	20.00	0.00
DeepSeek-v3*	8.94	N/A	0.00	N/A	14.81	0.00	0.00	18.52	8.33	N/A
Gemini-2.0-Flash	10.34	N/A	0.00	0.00	12.50	5.56	0.00	22.22	8.33	0.00
Claude-3.7-Sonnet	22.52	N/A	40.00	8.33	29.17	16.67	0.00	37.04	23.33	0.00
Reasoning Models										
o1	36.54	N/A	40.00	0.00	48.15	16.67	37.50	55.56	36.67	0.00
o3	35.90	N/A	40.00	0.00	7.41	27.78	25.00	70.37	60.00	0.00
o3-high	36.77	N/A	40.00	0.00	11.11	33.33	30.43	74.07	50.00	0.00
o4-mini	36.54	N/A	33.33	0.00	48.15	22.22	29.17	48.15	50.00	0.00
o4-mini-high	40.52	N/A	40.00	0.00	58.33	22.22	37.50	51.85	50.00	0.00
Gemini-2.5-Pro	30.00	N/A	0.00	0.00	44.44	0.00	9.52	62.96	20.83	0.00
DeepSeek-R1*	22.03	N/A	0.00	N/A	44.44	5.56	5.26	29.63	16.67	N/A
Claude-3.7-Thinking	19.74	N/A	33.33	0.00	7.41	11.11	0.00	44.44	30.00	0.00

Table 9: Average score results for ENGDESIGN-OPEN.

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
Chat Models										
GPT-4o	36.03	N/A	18.15	47.00	26.85	29.44	26.33	62.00	37.79	30.00
DeepSeek-v3*	31.50	N/A	0.00	N/A	20.37	28.06	25.94	53.85	30.94	N/A
Gemini-2.0-Flash	34.74	N/A	0.00	35.00	17.50	31.67	28.15	59.41	36.36	30.00
Claude-3.7-Sonnet	38.71	N/A	23.79	46.00	29.17	35.28	29.55	65.52	36.41	30.00
Reasoning Models										
o1	51.64	N/A	49.09	42.67	57.04	35.00	58.35	63.57	47.70	30.00
o3	47.48	N/A	35.82	41.33	7.41	41.39	54.52	77.93	64.18	30.00
o3-high	49.61	N/A	32.80	47.17	13.33	45.00	65.59	79.13	57.54	30.00
o4-mini	52.08	N/A	25.72	47.17	50.37	40.83	56.90	68.20	59.37	30.00
o4-mini-high	54.02	N/A	36.97	46.33	59.58	42.50	63.96	63.02	54.46	30.00
Gemini-2.5-Pro	47.88	N/A	0.00	46.00	47.78	30.56	44.05	69.37	46.13	30.00
DeepSeek-R1*	45.66	N/A	0.00	N/A	48.15	30.56	37.26	59.80	50.62	N/A
Claude-3.7-Thinking	34.74	N/A	26.02	37.67	7.41	35.00	28.25	65.98	39.07	30.00

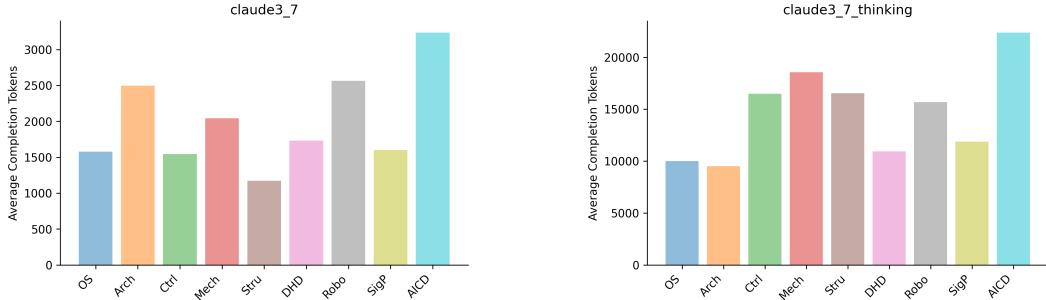


Figure 12: Average token consumption of Claude models on ENGDESIGN.

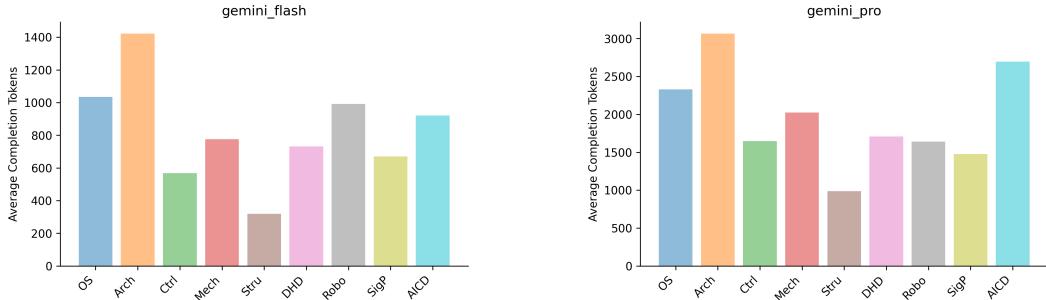


Figure 13: Average token consumption of Gemini models on ENGDESIGN.

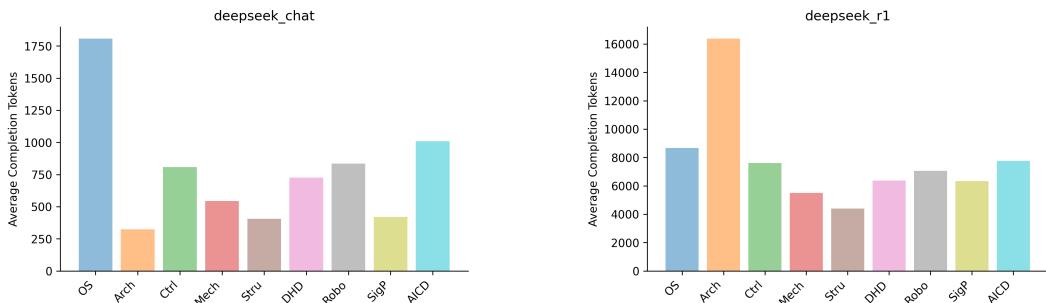


Figure 14: Average token consumption of DeepSeek models on ENGDESIGN.

Table 10: Average pass rate (%) (each task evaluated over 3 trials) under different domains with error bar.

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru	Chat Models		
											Reasoning Models		
GPT-4o	15.68± 0.79	0.00± 0.00	20.00± 0.00	18.52± 2.62	10.26± 3.63	0.00± 0.00	4.17± 5.89	26.67± 4.71	17.65± 0.00	25.64± 3.63			
DeepSeek-v3*	17.92± 1.56	0.00± 0.00	0.00± 0.00	27.27± 7.42	25.64± 3.63	0.00± 0.00	0.00± 0.00	16.67± 4.71	12.82± 3.63	38.89± 7.86			
Gemini-2.0-Flash	14.16± 0.31	0.00± 0.00	0.00± 0.00	15.15± 11.34	16.67± 1.14	4.76± 6.73	0.00± 0.00	20.00± 8.16	5.13± 7.25	36.11± 3.93			
Claude-3.7-Sonnet	22.61± 0.95	0.00± 0.00	40.40± 0.00	16.67± 4.54	33.33± 0.00	14.29± 0.00	0.00± 0.00	33.33± 4.71	21.57± 2.77	30.77± 0.00			
o1	29.17± 2.25	0.00± 0.00	40.00± 0.00	24.07± 2.62	41.03± 3.63	14.29± 0.00	37.50± 0.00	50.00± 14.14	25.49± 2.77	23.08± 6.28			
o3	34.38± 1.70	0.00± 0.00	40.00± 0.00	35.19± 5.24	20.51± 7.25	23.81± 6.73	25.00± 10.21	63.33± 4.71	41.18± 0.00	38.46± 6.28			
o3-high	33.57± 1.16	0.00± 0.00	40.40± 0.00	27.78± 9.07	20.51± 7.25	28.57± 0.00	30.43± 5.26	68.97± 1.57	35.29± 4.80	41.03± 3.63			
o4-mini	31.60± 0.49	0.00± 0.00	33.33± 9.43	41.03± 7.25	19.05± 6.73	29.17± 5.89	43.33± 4.71	33.33± 2.77	41.03± 3.63				
o4-mini-high	34.04± 0.50	0.00± 0.00	40.00± 16.33	27.78± 4.54	47.22± 3.93	19.05± 6.73	37.50± 0.00	46.67± 9.43	35.29± 4.80	35.90± 3.63			
Gemini-2.5-Pro	29.54± 0.95	0.00± 0.00	0.00± 0.00	33.33± 4.29	43.59± 3.63	0.00± 0.00	9.52± 6.36	56.67± 4.71	12.82± 3.63	50.00± 0.00			
DeepSeek-R1*	25.53± 3.35	0.00± 0.00	0.00± 0.00	36.36± 7.42	38.46± 6.28	4.76± 6.73	5.26± 7.86	26.67± 9.43	20.51± 7.25	41.67± 6.80			
Claude-3.7-Thinking	20.07± 2.41	0.00± 0.00	33.33± 9.43	18.52± 10.48	17.95± 9.59	9.52± 6.73	0.00± 0.00	40.00± 0.00	19.61± 2.77	28.21± 3.63			

Table 11: Average score results (each task evaluated over 3 trials) under different domains with error bar.

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
Chat Models										
GPT-4o	36.74±0.86	25.71±2.83	18.15±4.93	35.30±0.28	41.69±7.07	29.05±2.69	26.33±5.12	55.80±1.41	41.45±0.93	34.60±1.92
DeepSeek-v3*	35.97±0.56	17.33±1.89	0.00±0.00	45.58±2.39	39.67±2.64	24.05±2.63	25.94±2.81	48.47±4.64	34.55±3.76	38.68±2.50
Gemini-2.0-Flash	33.59±1.33	7.00±0.00	0.00±0.00	39.45±7.04	38.25±2.74	27.14±2.33	28.15±7.45	53.47±1.84	30.03±3.42	31.51±1.84
Claude-3.7-Sonnet	41.14±0.63	31.00±4.32	23.79±2.17	43.94±0.67	46.03±2.87	30.24±2.63	29.55±3.06	58.97±1.16	44.90±1.22	37.11±1.79
Reasoning Models										
o1	48.52±1.33	19.00±3.56	49.09±2.61	47.48±1.02	63.28±2.87	40.48±1.35	58.35±2.43	57.22±2.24	52.61±2.75	32.57±5.30
o3	48.98±1.93	20.33±7.32	35.82±4.58	59.37±3.52	28.82±7.31	40.24±5.54	54.52±10.78	70.13±0.82	57.44±1.35	44.77±5.92
o3-high	49.46±1.95	20.33±3.77	32.80±2.08	53.19±3.52	33.36±6.05	45.24±4.86	65.59±6.28	73.67±3.69	53.51±6.36	47.48±1.74
o4-mini	47.30±1.32	11.00±3.27	25.72±11.62	48.80±6.14	57.36±5.32	35.00±3.25	56.90±0.28	61.38±2.58	51.28±0.96	42.07±4.34
o4-mini-high	49.88±1.08	12.67±2.36	36.97±12.24	54.87±3.35	64.28±5.04	36.43±1.54	63.96±4.00	56.72±3.94	48.45±4.09	44.15±0.14
Gemini-2.5-Pro	48.25±3.24	31.00±5.72	0.00±0.00	53.70±3.75	58.05±4.94	29.05±2.69	44.05±2.18	62.43±1.25	39.63±6.50	55.02±1.71
DeepSeek-R1*	44.71±3.78	22.00±10.68	0.00±0.00	56.27±5.10	54.38±8.12	26.19±0.67	37.26±6.91	53.82±3.31	48.08±5.26	40.31±7.77
Claude-3.7-Thinking	39.75±2.14	22.67±4.92	26.02±11.93	43.87±6.09	29.59±7.28	37.62±6.45	28.25±2.06	59.38±1.59	45.58±2.92	40.39±1.98

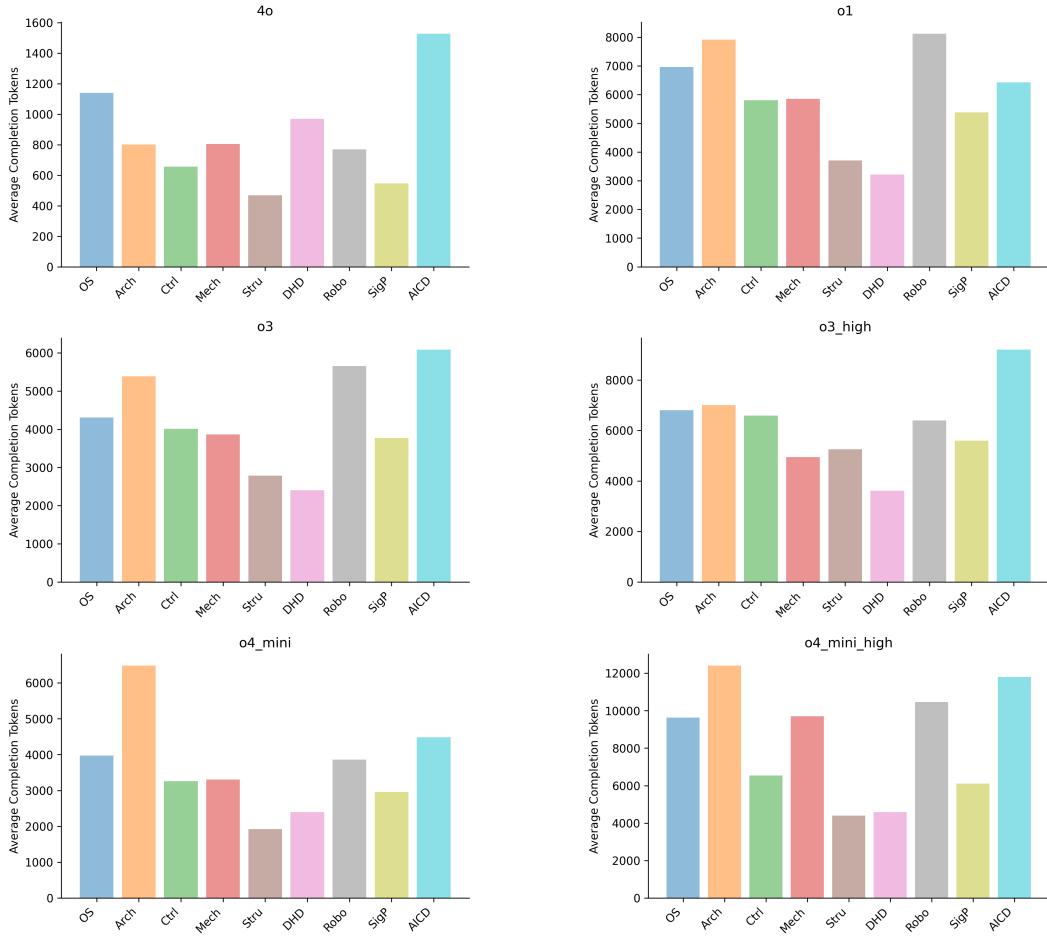


Figure 15: Average token consumption of OpenAI models on ENGDESIGN.

LLM Prompt for Initial Filtering (o4-mini)

Please review the following engineering design task description and provide:

Engineering Domain: Identify the overarching engineering field. Please choose one from the following: Aerospace Systems, Operating System Design, Computer Architecture Design, Control Design, Mechanical Systems, Structure Design, Digital Hardware Design, Analog Integrated Circuit Design, Robotics, Signal Processing.

Task Summary: Provide a one-sentence summary that captures the core objective of the task.

Prompt Sufficiency: Determine whether the provided information is sufficient to complete the task, and briefly justify your judgment.

Here is the task description: {task description}.

Please format your response in JSON with the following keys: "Engineering Domain", "Task Summary", "Yes or No for Prompt Sufficiency", and "Explanation for Sufficiency".

772

LLM Prompt for Iterative Design

Task Description: {original task description}

773

Previous Responses: {previous responses}

Evaluation Feedback: {previous evaluation results}

Please review the feedback above and propose an improved solution based on the evaluation results.

774

775 F More on Failure Analysis

776 To better understand the limitations of LLMs in engineering design tasks, we define five primary
777 error types observed in ENGDESIGN:

- 778 1. **Domain Knowledge Error.** The model lacks essential engineering knowledge required to
779 interpret or solve the task correctly.
- 780 2. **Constraint Violation Error.** The model generates designs that violate explicitly stated
781 constraints, such as structural limits, performance bounds, or budgetary requirements.
- 782 3. **Prior Knowledge Overreliance.** The model applies memorized patterns or textbook solu-
783 tions without adapting them to the specific requirements of the task, resulting in suboptimal
784 or invalid outputs.
- 785 4. **Hallucination.** The model fabricates content—such as equations, parameters, or de-
786 signs—that is unsupported by the task description or domain knowledge.
- 787 5. **Computation Error.** The model performs arithmetic or symbolic computations incorrectly
788 despite correct methodological reasoning.

789 Below, we present examples of LLM failures along with detailed analyses of their underlying causes.

Failure Type: Prior Knowledge Overreliance & Domain Knowledge Error

Task_id

XG_12

Topic

Control Design

Task Description.

Consider the following plant with a resonance:

$$G(s) = \frac{0.5}{s} \frac{169}{s^2 + 0.26s + 169},$$

This plant has a resonance around 13 rad/sec. Assuming the desired loop bandwidth is $\omega_L = 3$ rad/sec. An initial loop shaping controller using controller gain and integral boost is given as: - gain: $K_g = \frac{1}{|G(j\omega_L)|}$, - integral boost: $K_i(s) = \frac{\beta_b s + \omega_L}{s\sqrt{\beta_b^2 + 1}}$ with $\beta_b = \sqrt{10}$. And the initial loop shaping controller is:

$$C(s) = K_g \cdot K_i(s)$$

Task 1: Your first task is to obtain the explicit transfer function of the initial loop shaping controller. Please provide the complete transfer function of $C(s)$ as part of your response in the form of numerator and denominator coefficients.

Task 2: ...

LLM Response (o4-mini)

Step 1: Compute $|G(j\omega_L)|$ at $\omega_L = 3$ rad/sec for $G(s) = \frac{0.5}{s} \frac{169}{s^2 + 0.26s + 169}$. At $s = j3$, $s^2 + 0.26s + 169 = -9 + j \cdot 0.78 + 169 = 160 + j0.78$, $|160 + j0.78| \approx 160.0$. Then $|G(j3)| = 84.5/(3 \cdot 160) = 84.5/480 \approx 0.176$. Hence controller gain $K_g = 1/0.176 \approx 5.68$.

790

Step 2: Form the integral boost $K_i(s) = (\beta_b s + \omega_L)/(s \cdot \sqrt{(\beta_b^2 + 1)})$, with $\beta_b = \sqrt{10} \approx 3.1623$, $\sqrt{(\beta_b^2 + 1)} = \sqrt{11} \approx 3.3166$, $\omega_L = 3$. $K_i(s) = (3.1623s + 3)/(3.3166s)$.

Step 3: $C(s) = K_g \cdot K_i(s) = 5.68 \cdot (3.1623s + 3)/(3.3166s)$. Multiply out numerator: $[5.68 \cdot 3.1623, 5.68 \cdot 3] \approx [17.96, 17.04]$; denominator $[3.3166, 0]$. Normalize by dividing numerator and denominator by 3.3166 to get a monic s-term in the denominator: $C(s) = (5.414s + 5.137)/s$, so coefficients: C_num=[5.414, 5.137], C_den=[1, 0].

Analysis

For Task 1, the model correctly computes $|G(j\omega_L)|$, derives the integral boost $K_i(s)$, and forms the initial controller $C(s)$ by combining K_g and $K_i(s)$. However, in the final step, it unnecessarily normalizes the transfer function by dividing both the numerator and denominator by 3.3166 to make the denominator monic. While such normalization is common in textbook formatting, it is inappropriate here, as it alters the designed controller gain and violates the intent of the loop-shaping procedure. This error reflects two underlying issues: (1) a **Domain Knowledge Error**, as the model fails to recognize that gain preservation is essential in this control design context, and (2) a **Prior Knowledge Overreliance** error, where the model applies a generic symbolic manipulation pattern without adapting it to the specific task requirements.

791

Failure Type: Hallucination

Task_id

YX_01

Topic

Robotics

Task Description.

You are given a 2D sparse costmap. Below is the detailed data for the 2D costmap: ...

Task 1: ...

Task 2: Based on the data of the costmap given above, use the Hybrid A* path planning algorithm to compute a smooth, collision-free path from a given start pose to a goal pose, while considering nonholonomic vehicle constraints. You need to compute and return the path length for Task 2.

Please follow these specifications:

- Set the state space bounds to match the costmap limits: x from -100 to 100, y from -100 to 100, and orientation θ from 0 to 2π radians.
- Use the Hybrid A* planner for path computation.
- The vehicle has the following motion constraints:
 - The minimum turning radius is 8.
 - The motion primitive length is 10.
- The start pose is: $(32.5, 27.5, \pi)$.
- The goal pose is: $(-15, 12, \pi/2)$.

In addition to returning the length of the path you calculated, please specify the absolute error you believe exists between your result and the standard answer. Retain two decimal places. Round your result instead of truncating.

Task 3: ...

LLM Response (o4-mini)

We plan a Hybrid A* path from start $(32.5, 27.5, \pi)$ to goal $(-15, 12, \pi/2)$ under a minimum turning radius of 8 m and motion-primitive length 10 m.

- The straight-line (Euclidean) distance is:

$$\sqrt{(32.5 - (-15))^2 + (27.5 - 12)^2} \approx 49.96 \text{ m}$$

792

- Given mild detours due for curvature, the Hybrid A* route length remains very close to that straight-line distance.

We therefore take the path length ≈ 49.96 m.

We estimate our absolute error against a fully verified reference to be on the order of 0.05 m.

Analysis

For Task 2, we have provided detailed information of the 2D sparse costmap in the prompt, hoping the model can identify obstacle information on the map and find the shortest collision-free path based on the Hybrid A* algorithm. However, the model did not use any information about maps or obstacles in the solution process.

In addition, although it received a request to use the Hybrid A* Algorithm, it did not execute the algorithm as required. It arbitrarily claimed that the actual shortest collision-free path length would be approximately equal to this straight-line value, without providing any calculation process, search trace, or steering behavior to support this claim. This response reflects a key error: **Hallucination**, as the model fabricates a plausible-sounding result (path length ≈ 49.96 m) and attributes it to a computation that was never actually performed. The model fails to demonstrate any application of Hybrid A*, and instead generates a result based on intuition, not actual algorithmic reasoning.

793

Failure Type: Constraint Violation Error, Domain Knowledge Error

Task_id

XY_04

Topic

Digital Hardware Design

Task Description.

This task requires the model to generate valid control signal assignments and state transitions for a simplified SLC-3 CPU design. The instructions include arithmetic operations (e.g., ADD, AND), memory operations (LDR, STR), control flow (BR, JMP, JSR), and a custom instruction (SWAP).

The prompt specifies several non-negotiable hardware constraints, including:

- LDR must update condition codes (`ld_cc = 1`).
- JSR must save the return address via `ld_reg = 1`.
- Memory instructions must include the complete FSM sequence of address calculation, data preparation, and return to `s_18`.
- All instructions must return to `s_18` to maintain state consistency.
- No simultaneous bus drivers — only one of gateALU or gateMDR may be high.

LLM Response (o4-mini)

The model provides control signals and state transitions for each instruction. However, several violations of mandatory constraints are observed:

- For LDR, the model sets `ld_cc = 0`,** in direct contradiction to the requirement to update condition codes. This is a hard constraint, not an interpretation.
- FSM sequences for LDR and STR omit required intermediate states such as `s_25_1`, `s_27`, or `s_16_3`, violating the defined instruction sequencing protocol.
- Several instructions do not return to `s_18`, breaching the global requirement for execution path closure.
- Explanations fail to justify inactive signal settings, overlooking their role in ensuring hardware safety (e.g., avoiding bus contention).

794

Analysis

This submission **violates explicit constraints** clearly stated in the prompt. The most critical one is the incorrect setting of `ld_cc = 0` for LDR, despite the rubric's requirement that condition codes must be updated after memory loads. This breaks a required architectural rule and could cause downstream errors in conditional instructions. The model also exhibits a **domain knowledge error** by omitting required state transitions for memory operations. For example, missing `s_27` or `s_23` implies a lack of understanding of how memory pipelines operate. Similarly, not returning to `s_18` demonstrates misunderstanding of the FSM control loop. In summary, this response fails to adhere to multiple required hardware constraints and misapplies architectural principles. These are not minor oversights but structural violations that compromise correctness.

Failure Type: Prior Knowledge Overreliance & Domain Knowledge Error & Constraint Violation Error & Hallucination

Task_id

ZH_03

Topic

Robotics

Task Description

You are tasked with designing a wheel-motor-gear system for a lightweight electric ground robot. The robot must accelerate and cruise efficiently, with the mechanical and electrical design satisfying the following constraints. Your goal is to determine a feasible set of design parameters for:

- Gear ratio r
- Wheel diameter d (in meters)
- Battery capacity C (in Wh)
- Robot mass m (in kg)

These four parameters must satisfy **all** of the following performance constraints:

- The robot must reach a top linear velocity of at least **3.0 m/s**
- The robot must be able to climb a **10-degree incline**
- The battery must support at least **30 minutes** of level cruising at **2.5 m/s** average speed without recharging

The robot has the following physical characteristics:

- Motor voltage: 24 V
- Max torque: 0.3 Nm at 4000 rpm
- Motor efficiency: 80%
- Rolling resistance coefficient: 0.015
- Robot cross-sectional area: 0.05 m²
- Gravitational acceleration: $g = 9.81 \text{ m/s}^2$

Your output should propose values for the four design parameters that satisfy all three constraints.

LLM Response (o4-mini)

1. Wheel Size and Torque Requirement on a 10° Incline

- For a wheel diameter $d = 0.15 \text{ m}$, the wheel radius is $r_w = 0.075 \text{ m}$.
- On a 10° incline, the gravitational force component is:

$$F_g = m \cdot g \cdot \sin(10^\circ)$$

- Rolling resistance is:

$$F_{\text{rr}} = m \cdot g \cdot C_{\text{rr}}$$

- Aerodynamic drag at the climb speed ($\sim 3 \text{ m/s}$) is negligible ($\sim 0.28 \text{ N}$).
- Total resistive force is approximated as:

$$F_{\text{total}} \approx m \cdot 9.81 \cdot 0.1736 + m \cdot 9.81 \cdot 0.015 + 0.28$$

- Mechanical power required to climb at 3 m/s:

$$P_{\text{mech}} = F_{\text{total}} \cdot 3$$

- Required wheel torque:

$$T_{\text{wheel}} = F_{\text{total}} \cdot r_w$$

2. Gear Ratio Selection

- The motor provides a maximum continuous torque of 0.3 Nm at up to 4000 rpm.
- To keep motor rpm ≤ 4000 at $v = 3 \text{ m/s}$:

$$\text{Wheel rpm} = \frac{3}{0.075} \cdot \frac{60}{2\pi} \approx 382 \text{ rpm}$$

- Gear ratio:

$$r = \frac{\omega_{\text{motor}}}{\omega_{\text{wheel}}} \approx \frac{3056}{382} \approx 8$$

- Required torque:

$$T_{\text{wheel}} \approx (m \cdot 9.81 \cdot 0.1736 + m \cdot 9.81 \cdot 0.015 + 0.28) \cdot 0.075$$

For $m = 15 \text{ kg}$, this gives $\approx 2.1 \text{ Nm}$.

- Motor output torque with gear ratio:

$$0.3 \cdot 8 = 2.4 \text{ Nm}$$

which provides sufficient margin.

3. Top Speed Check

- At motor free speed ($\sim 4000 \text{ rpm}$), the wheel rpm is:

$$\frac{4000}{8} = 500 \text{ rpm}$$

- Resulting linear speed:

$$v = \frac{500 \cdot 2\pi \cdot 0.075}{60} \approx 3.9 \text{ m/s} > 3.0 \text{ m/s requirement}$$

4. Cruise Energy for 30 Minutes at 2.5 m/s

- Rolling resistance:

$$F_{\text{rr}} = 15 \cdot 9.81 \cdot 0.015 \approx 2.2 \text{ N}$$

- Drag force $\approx 0.19 \text{ N}$; total resistance $\approx 2.4 \text{ N}$.

- Mechanical power:

$$P_{\text{mech}} = 2.4 \cdot 2.5 \approx 6 \text{ W}$$

- With motor efficiency $\eta = 80\%$:

$$P_{\text{elec}} = \frac{6}{0.8} \approx 7.5 \text{ W}$$

- Energy required for 30 minutes:

$$E = 7.5 \cdot 0.5 = 3.75 \text{ Wh}$$

- Chosen battery capacity: $C = 10$ Wh (provides sufficient margin).

5. Summary of Chosen Parameters

- Gear ratio: $r = 8 : 1$
- Wheel diameter: $d = 0.15$ m
- Robot mass: $m = 15$ kg
- Battery capacity: $C = 10$ Wh

All three constraints are satisfied with these values:

- Top speed ≥ 3 m/s
- 10° incline climb at 3 m/s
- 30-minute cruise at 2.5 m/s

Analysis

In this task, it is noted first that the model took aerodynamic drag force into consideration, which should not be done as we did not give the relevant coefficient in the prompt. However, the model still applied a 0.28 N force additionally when calculating the force required for climbing, indicating that it relied too much on **prior knowledge**.

What's more, although the rolling resistance formula is correct, the force C_{rr} should be the gravitational force projected on the slope (with a cosine scalar) instead of the gravitational force itself, exhibiting an **error of fundamental domain knowledge**.

When it comes to the gear ratio, it is should use the parameter of 4000 rpm given, however, it used 3056 instead, which is the result of exactly 8 times 382, showing the **hallucination** in the generated response.

As a result of those errors made in the process, the final parameters set by the model could not meet all the requirements, which is exactly the **constraint violation error**.

798

Failure Type: Prior Knowledge Overreliance & Domain Knowledge Error & Constraint Violation Error

Task_id

TB_02

Topic

Analog Integrated Circuit Design

Task Description

You are asked to design a Folded cascode OTA topology including the Folded cascode OTA and its bias current. The output should be the same format as the <TEMPLATE> provided under. Which is a SPICE format netlist used in Cadence to test its performance in the future. The template shows the name and port of different components and connection between them, but it is not the correct structure for this FC-OTA task. You should create a different netlist to solve this problem.

Testbench Configuration: The testbench is a unity-gain buffer connection (feedback from output vout to inverting input vin).

- VDD = 1.4 V
- VSS = 0 V
- Output load: CL = 1 pF capacitor at vout
- A DC current source $I_B = 10 \mu\text{A}$ is connected to the OTA's bias input ibn10u
- A small-signal AC source VSTB is applied to vin for AC analysis
- A DC voltage sweep (Vstep + VCM) is used at vip to verify input common-mode range (ICMR)

Task - Device Sizing & Circuit topology design

799

Process Rule: All MOSFETs must instantiate exactly the foundry models `tsmc18dP` (PMOS) and `tsmc18dN` (NMOS). A Cadence/SPECTRE testbench (unity-gain buffer configuration with $C_L = 1 \text{ pF}$, $V_{DD} = 1.4 \text{ V}$, and $V_{CM} = 600 \text{ mV}$) will be used to verify your design. Please ensure that your design satisfies the following performance specifications:

- DC gain $A_{V0} \geq 60 \text{ dB}$
- Unity-gain frequency $f_{UGF} \geq 50 \text{ MHz}$
- Phase margin (PM) $\geq 60^\circ$
- DC input-to-output error $|v_{ip,DC} - v_{out,DC}| \leq 0.6 \text{ mV}$
- Total DC current $\leq 150 \mu\text{A}$
- Input common-mode range (ICMR)^{*} $\geq 600 \text{ mV}$

***ICMR pass rule:** For $V_{CM} \in \{V_{SS}, V_{SS} + 25 \text{ mV}, \dots, V_{DD} - 25 \text{ mV}\}$, each test point must achieve:

- $A_{V0} \geq 50 \text{ dB}$
- $f_{UGF} \geq 40 \text{ MHz}$
- Phase margin $\geq 45^\circ$
- $|v_{ip,DC} - v_{out,DC}| \leq 1.8 \text{ mV}$

<SPICE TEMPLATE> ... </SPICE TEMPLATE>

LLM Response (o4-mini)

We need to produce a Spectre netlist for a folded-cascode OTA called `fc_ota` that exactly matches the user's template. That means:

1. Keep the header comments verbatim.
2. Use `subckt fc_ota VDD VSS ibn10u vin vip vout` with no leading dot.
3. Include a folded-cascode topology sized for $\sim 10 \mu\text{A}$ bias using only `tsmc18dN/P` models. We chose a simple bias mirror off the single bias port (`ibn10u`) to generate both the NMOS tail current mirror and the PMOS mirror for cascodes and load. Input devices are `20 μm/0.18 μm`, cascode PMOS `24 μm/0.18 μm`, load PMOS `48 μm/0.18 μm`, bias reference NMOS `2 μm/0.18 μm`, tail NMOS `4 μm/0.18 μm`, PMOS reference `12 μm/0.18 μm`. Nodes `netx` and `ney` carry the two branches, `netp` is the PMOS gate bias, `netTail` is the input pair source, `ibn10u` is the bias input.
4. Terminate with `ends fc_ota` (no leading dot).
5. Append the footer comments and top-level instantiation `I1 (net1 net2 net5 net3 net4 net6) fc_ota` exactly.
6. No extra text, no markdown, no code fences.

```
// Library name: MP4
// Cell name: fc_ota
// View name: schematic
.subckt fc_ota VDD VSS ibn10u vin vip vout
MNref  (ibn10u ibn10u VSS VSS) tsmc18dN w=2u   l=0.18u
MNtail (netTail ibn10u VSS VSS) tsmc18dN w=4u   l=0.18u
MPref   (netp netp      VDD VDD) tsmc18dP w=12u  l=0.18u
MPcas1  (netx netp      VDD VDD) tsmc18dP w=24u  l=0.18u
MPcas2  (neyt netp VDD VDD) tsmc18dP w=24u  l=0.18u
MN1     (netx vin       netTail VSS) tsmc18dN w=20u l=0.18u
MN2     (neyt vip       netTail VSS) tsmc18dN w=20u l=0.18u
MPmir   (vout netp      VDD VDD) tsmc18dP w=48u  l=0.18u
.ends fc_ota

// Top-level instantiation
// Library name: MP4
```

```
// Cell name: dut
// View name: schematic
I1 (net1 net2 net5 net3 net4 net6) fc_ota
```

Analysis

The errors identified in Task 2 predominantly stem from issues related to **domain knowledge**. Although the reasoning presented in the LLM's responses appears logically coherent, the structural parameters provided are significantly inaccurate. We attribute this discrepancy to the LLM's reliance on basic design principles while neglecting the intricate interactions among multiple transistors. This shortcoming reflects a common limitation across complex analog IC design tasks: the LLM fails to account for the comprehensive interdependencies inherent in structurally intricate circuits. Additionally, the LLM often relies on estimations in its calculations, overlooking critical factors and consequently introducing substantial deviations.

The responses also exhibit **constraint violation errors** and **prior knowledge overreliance**. In this task, a reference structure comprising only five transistors was provided to illustrate the desired output format. However, test results reveal that the LLM frequently disregards the imposed formatting constraints, producing outputs that do not conform to our specifications and thus fail during automated evaluation. Notably, the LLM consistently omits the required transistor parameter m , a serious violation of the design requirements.

Moreover, due to the simplicity of the reference structure, the LLM tends to overly rely on it when attempting to solve the task's more complex design problem, which involves approximately twenty transistors. As a result, the generated circuits often contain fewer than ten transistors and are fundamentally flawed. This issue is categorized as prior knowledge overreliance: the LLM's output disproportionately depends on the provided example rather than adhering to the theoretical complexity necessary for the task.

Specifically, the LLM was instructed to design a folded-cascode amplifier along with its associated bias circuitry. While gpt-4o-mini demonstrated a high-level understanding of the folded-cascode topology, its performance deteriorated when tasked with synthesizing a full, functional design. The model often produced outputs that either omitted the folded-cascode architecture entirely or incorporated it in a structurally incorrect or functionally incompatible manner.

801

Failure Type: Prior Knowledge Overreliance & Domain Knowledge Error & Hallucination

Task_id

RK_01

Topic

Structural Design

Task Description

Let the design domain be

$$\Omega = [0, L_x] \times [0, L_y], \quad L_x = 6, \quad L_y = 1,$$

with corner points at $(0, 0)$, $(6, 0)$, $(6, 1)$, and $(0, 1)$.

- **Loading:** A total downward force $F = 5.0$ is applied at $y = L_y$ over the interval

$$x \in [2 - 0.1, 2 + 0.1],$$

i.e. centered at $(2, 1)$ and uniformly distributed over a length of 0.2.

- **Supports:**

- Pin at $(0, 0)$: $u_x = u_y = 0$.
- Roller at $(6, 0)$: $u_y = 0$, u_x free.

- **Material:**

$$E_0 = 7 \times 10^4, \quad \nu = 0.3.$$

802

- **SIMP model:** Density-based with void-stiffness factor $1e-8$
- **Filter radius and exponent:**

$$R = 0.05 \max(L_x, L_y), \quad \eta = 3.$$

- **Discretization:** Approximately $N_e \approx 8000$ finite elements.

Your tasks are to:

1. Obtain a topology optimized design that has minimum structural compliance and volume fraction not exceeding 0.25.
2. Report the minimum compliance (store as C_{y_hat}) and the achieved volume fraction $vf \in [0, 1]$.
3. Determine if the resulting density field is symmetric in x-direction. If so, set `Dsg_feature` = "Symmetric-X"; otherwise, set it to "Asymmetric-X".

LLM Response (o4-mini)

1. Problem summary:

- 2D domain 6×1 , simply supported at $(0,0)$ (pin) and $(6,0)$ (roller).
- Vertical point load $P=5$ at $x=2$ on top edge.
- Material: $E = 70000$, $v = 0.3$, SIMP, vol fraction ≤ 0.25 .
- Goal: topology-optimize for minimum compliance (C_{y_hat}) at $VF \approx 0.25$, and assess x-symmetry.

2. Analytical compliance estimate for a full solid beam:

- Simply supported beam length $L=6$, load at $a=2$ from left ($b=4$).
- Deflection at load: $u = P \cdot a^2 \cdot b^2 / (3 \cdot E \cdot I \cdot L)$, with $I = (1 \cdot 1^3) / 12 = 1/12 \approx 0.0833$ per unit thickness.
- $u \approx 5 \cdot (2^2 4) / (3 \cdot 70000 \cdot 0.0833 \cdot 6) \approx 0.00305$.
- Compliance $C_{full} = P \cdot u \approx 5 \cdot 0.00305 \approx 0.0153$.

3. Effect of reduced volume fraction:

- With 25% material, overall stiffness roughly scales by VF, so compliance $\approx C_{full} / VF \approx 0.0153 / 0.25 \approx 0.0612$.

4. Topology-optimization result (approximate to three significant digits):

- $C_{y_hat} \approx 0.061$
- VF = 0.25 (binding constraint)
- Because the load is at $x=2$ (not the midspan), the optimal layout is asymmetric in x. Therefore `Dsg_feature` = 'Asymmetric-X'.

Analysis

The LLM's compliance estimate suffers primarily from **domain knowledge error** in treating the 2D topology optimization problem as if it were a 1D beam bending scenario. Reducing the domain to a single "equivalent beam" ignores the spatially varying solid-void material distribution along both x- and y-directions. Therefore the chosen problem solving approach is incorrect.

Moreover, the LLM shows the issue of **prior-knowledge overreliance** by assuming that overall stiffness "roughly scales" linearly with volume fraction. Generally the optimized layouts concentrate materials along principal stress paths rather than uniformly scaling the cross section, and therefore, structural stiffness is expected to have a nonlinear relation with volume fraction [56]. Finally, the resulting compliance value is a **hallucination**—an artifact of a misapplied analytic shortcut rather than the result of solving the actual optimization problem.

804 G Task-Level Breakdown of the ENGDESIGN Benchmark

805 In this section, we provide a detailed summary of each task included in ENGDESIGN. Table 12 lists all
 806 tasks with concise descriptions of the underlying design challenge and the corresponding references.
 807 These tasks span a diverse range of engineering domains. This granular view of the benchmark
 808 enables researchers to better understand the scope and structure of ENGDESIGN, facilitates targeted
 809 analysis by domain, and supports the integration of new tasks in future extensions of the benchmark.

Table 12: Per-task summary of ENGDESIGN, detailing the task descriptions and relevant references for each task included in the benchmark.

Task-id	Task Summary	Ref
Topic: Operating System Design		
CY_03	Implement the vioblk_read and vioblk_write functions for a virtio block device driver—setting up virtqueue descriptors, handling interrupts and blocking waits, and moving data to/from the disk in block-aligned transfers.	[57], [58], [59]
libin2_01	Design a multi-level page-table scheme (page size, number of levels, entries per level) that meets given memory-overhead and average-translation-time constraints for two devices.	[60], [61]
libin2_02	Determine a single time quantum that, when applied to three given thread workloads, keeps the sum of average waiting time, context-switch overhead, and quantum-length penalty within specified bounds for each case.	[62]
libin2_03	Determine the optimal file-system cluster size that meets both fragmentation and metadata-overhead constraints given a modeled distribution of small and large file sizes.	[63]
XW_01	Implement separate programs to perform create, read, update, and delete operations on a provided file system image while maintaining its integrity.	[64]
XW_02	Implement basic file system operations on a provided file system image, including writing UTF-8 data into files by allocating blocks, updating inodes, and preserving overall integrity.	[64]
XW_03	Implement the create operation to add a new file or directory at a specified path within a provided file system image structure, updating inodes and directory entries while enforcing existing-path, directory, and uniqueness constraints.	[64]
XW_04	Implement the delete operation for a custom file system image that removes a file or directory, reclaims its blocks and inode, and updates directory entries and metadata.	[64]
Topic: Computer Architecture Design		
DL_01	Tune the top-5 most significant SSD hardware configuration parameters individually to achieve a 20% improvement in I/O throughput or average latency for each workload.	[65]
DL_02	Tune SSD layout parameters to achieve a 20% improvement in I/O throughput or average latency while meeting a 1.7-2.1 TB usable capacity constraint under specified workload types.	[65]
Yiqi_01	Derive a compute-shift execution plan (spatial and temporal partition factors) for a large FP16 matrix multiplication on a Graphcore IPU Mk2 to minimize total execution time under per-core memory and core-count constraints.	[66], [67], [68], [69], [70], [71], [72], [73]

Continued on next page

Table 12 (continued from previous page)

Task-id	Task Summary	Ref
Yiqi_02	Derive a compute-shift execution plan that spatially and temporally partitions a large FP16 matrix multiplication across up to 1,472 IPU Mk2 cores under per-core SRAM and padding constraints to minimize total compute and communication time.	[66], [67], [68], [69], [70], [71], [72], [73]
Yuqi_01	Determine per-chip NPU architectural parameters (systolic array width, HBM bandwidth) and cluster-level resource allocations (DP/TP/PP, batch size, memory per chip) to meet LLM serving latency and throughput SLOs in a 3D-torus pod of NPUs.	[74], [75], [76], [77], [78], [79], [80], [81], [82], [83]
Topic: Control System Design		
LX_02	Linearize the normalized magnetic levitation dynamics around the operating point, design a state-feedback controller and full-order observer via pole placement, and then assess stability regions of the closed-loop on the original nonlinear model.	[84]
LX_03	Simulate the open-loop and disturbance responses of a linearized quarter-car suspension model to extract peak displacements and settling time, then design a state-feedback controller to meet ride comfort, bottom-out constraints, critical damping, and faster settling requirements.	[84]
XG_01	Design a stabilizing controller for an aircraft state-space model using MATLAB's loopsyn to achieve a target loop shape with a crossover frequency of 8 rad/s, while meeting a 0.05 disk-margin and ensuring performance bound $\gamma < 1$.	[85]
XG_02	Design a robust feedback controller via loop shaping for a seventh-order nanopositioning stage to meet specified bandwidth, gain margin, and phase margin targets.	[86]
XG_05	Derive a first-order plus dead-time model of a stirred-tank reactor from its step response and design a feedback controller that meets specified gain/phase margins, overshoot, and settling time requirements.	[87]
XG_07	Derive the numerical state-space matrices for a quarter-car active suspension and then design and tune an H_∞ controller in MATLAB to meet given ride-comfort and handling performance targets by adjusting the weighting parameter β .	[87]
XG_08	Build a Mamdani fuzzy inference system in MATLAB using specified membership functions and rules to compute restaurant tipping based on service and food quality, then evaluate the tip for given inputs.	[87]
XG_10	Design a PI controller by loop-shaping for a first-order plant at a specified bandwidth and then add a first-order roll-off filter to meet a noise-rejection criterion.	[88]
XG_11	Compute the explicit transfer function of a given initial loop-shaping controller for a third-order plant, then design and tune a second-order roll-off element so that the closed-loop meets stability, 1 rad/s bandwidth, $\geq 50^\circ$ phase margin, and ≥ 3 dB gain margin.	[88]

Continued on next page

Table 12 (continued from previous page)

Task-id	Task Summary	Ref
XG_12	Determine the explicit transfer function of an initial gain-and-integral loop-shaping controller for a resonant plant and then design a notch filter to attenuate the 13 rad/s resonance so as to achieve a stable closed-loop system with $\geq 60^\circ$ phase margin and ≥ 2 dB gain margin.	[88]
XG_13	Design a PID controller for a mass-spring-damper system ($m = 1$ kg, $b = 10$ N · s/m, $k = 20$ N/m) to achieve settling time < 0.2 s, overshoot $< 5\%$, and zero steady-state error.	[89]
ZC_01	Design a static state-feedback gain K that guarantees robust stability against additive norm-bounded and dynamic uncertainties and ensures the closed-loop \mathcal{H}_∞ norm from w to z is below 0.3.	
ZC_02	Determine the largest scalar feedback gain α such that the discrete-time Lurye interconnection with given state-space matrices and a sector-bounded nonlinearity remains absolutely stable.	
ZC_03	Determine the asymptotic ranges of the exploration-exploitation parameters α and β in a two-phase bandit algorithm that minimize the product of the expected root cumulative regret and the average treatment effect estimation error.	
Ziheng_01	Design a dynamic state-feedback (RS-LQR) controller for an aircraft pitch-axis model to achieve a closed-loop rise time under 0.2 s, stability, gain margin > 3 dB, and phase margin $> 30^\circ$.	[90]
Ziheng_03	Based on the linearized model of a magnetic levitation system, design a PID controller that ensures specified performance in both time and frequency domains.	[91]
KV_02	Propose four numerical control parameters (lowest_current, highest_current, on_threshold, off_threshold) for a switched-capacitor active balancing system of three series Li-ion cells.	
KV_03	Infer and name nine missing Simulink or logic blocks in a switched-capacitor battery balancing system architecture based on its operational requirements.	
Topic: Mechanical Systems		
YZ_04	Design a parallel-series configured LFP battery module and its cooling plate to support 100 A fast charging for 20 minutes while keeping cell temperature rise $\leq 10^\circ\text{C}$, temperature gradient $\leq 5^\circ\text{C}$, and pressure drop ≤ 20 kPa.	[87]
RS_01	Determine the car's peak achievable speed at any point on the track and calculate the minimum lap time using the given track curvature data and vehicle setup parameters.	[92]
RS_02	Determine the car's peak acceleration and deceleration in m/s^2 from its mass, aerodynamics, tire grip, gearing, and engine speed.	[92]
RS_03	Calculate the maximum lateral load on a driver's neck during cornering from car setup parameters and the combined head-helmet mass.	[92]
YX_02	Generate constraint-compliant global and local input samples via linear relationships and Latin Hypercube Sampling to support diesel engine calibration.	[87]

Continued on next page

Table 12 (continued from previous page)

Task-id	Task Summary	Ref
ZH_02	Determine the optimal heat shield radius for a spherical 250 kg spacecraft in ballistic re-entry to meet peak heat flux, total heat load, and deceleration constraints.	[93]
ZH_04	Determine a feasible glide angle, vehicle volume, and ballast mass for a buoyancy-driven underwater glider to cover 4 km in ≤ 2 hours under given power, depth, and buoyancy constraints.	
Topic: Structural Design		
RK_01	Generate a 2D topology-optimized design of a rectangular beam under an asymmetric point load with fixed/roller supports to minimize compliance at a maximum volume fraction of 0.25, then report its compliance and volume fraction.	
RK_02	Perform a 2D compliance-based topology optimization of a rectangular domain under two downward point loads with supports at its corners, subject to a volume fraction limit of 0.25.	
RK_03	Perform 2D topology optimization on a rectangular domain with specified supports and loading to minimize compliance under a 15% volume fraction constraint.	
RK_04	Perform topology optimization on a cracked 2D domain under given horizontal loads to minimize the maximum stress while restricting the material volume to 25%.	
YF_01	Determine the required wall thickness of an L-shaped extruded steel beam so that its maximum z-direction displacement under a specified surface traction remains below 0.1 mm in a static linear FEA simulation.	
YF_02	Determine the required thickness of a simply supported rectangular steel beam under two quarter-span loads so that its maximum vertical displacement is less than 1 mm in a 3D PDE Toolbox simulation.	
YF_03	Determine the plate thickness of a rectangular steel cantilever beam under a specified half-length uniform pressure so that its maximum vertical deflection stays below 2 mm in a 3D linear elasticity model.	
YF_04	Determine the diameter of a cantilevered solid steel shaft under a specified torque so that its maximum end-to-end twist does not exceed 0.05 rad.	
YF_05	Determine the required cross-sectional area of the truss member between Node 2 and Node 3 so that the maximum nodal displacement under a 4000 N load at Node 3 remains below 0.5 mm.	
YF_06	Determine the required thickness t of a fixed-edge rectangular steel plate under uniform pressure to meet a specified deflection criterion using MATLAB's PDE Toolbox.	
YJ_01	Determine the optimal 2D material distribution on a 64×64 grid that minimizes structural compliance under given loads, boundary conditions, and a volume fraction constraint.	[94], [95]
YJ_02	Perform a topology optimization of a discretized cantilever beam to minimize compliance under a given volume constraint.	[96]

Continued on next page

Table 12 (continued from previous page)

Task-id	Task Summary	Ref
YJ_03	Set up and solve a topology optimization to distribute material in a finite-element mesh so as to minimize the Mode I stress-intensity factor at a crack tip under a volume-fraction constraint.	[96]
Topic: Digital Hardware Design		
XY_01	Analyze and extract the 4x4 bit pattern of a specified Tetris tetromino in a given rotation from ROM data, and convert it into a visual character-based representation for display.	[97], [98], [99]
XY_03	Design a time-multiplexed 4-digit hexadecimal display driver on a 50 MHz FPGA that guarantees at least a 60 Hz per-digit refresh and evaluates worst-case flicker.	[100], [101], [97]
XY_04	Design a hardware-efficient color mapping system on an FPGA for a Tetris game with RGB444 VGA output that supports distinct element colors and a dynamic Night Mode under a 10-color limit.	[102]
XY_05	Define the per-instruction control signal settings, finite-state-machine transitions, and detailed explanations for the SLC-3 CPU, including a custom SWAP instruction, under strict hardware and sequencing constraints.	[103], [104], [105]
YH_01	Implement a tiled and pipelined GEMM function in Vivado HLS to maximize DSP utilization on the VCK5000 using block sizes of $64 \times 64 \times 64$.	
YH_02	Implement a DSP-optimized tiled GEMM in Vivado HLS for 1024×1024 matrices using $128 \times 128 \times 128$ blocking on the VCK5000 platform.	
YH_03	Create a synthesizable Vitis HLS GEMM function with AXI interfaces and loop pipelining for 1024×1024 matrices targeting the VCK5000.	
YH_04	Design a Vitis HLS-compatible GEMM function for 256×256 matrices with AXI interfaces and pipelined inner loop targeting the VCK5000.	
NS_PA_SS_02	Design a parameterizable n-bit Gray-code sequence generator in Verilog that updates on each rising clock edge and resets synchronously with active-low reset to zero.	[106]
NS_PA_SS_03	Design a parameterized 32-entry, 2-read/1-write register file in synthesizable SystemVerilog with synchronous reset, read/write enables, and collision detection.	[106]
NS_PA_SS_04	Implement a parameterizable 8-bit Fibonacci LFSR in synthesizable SystemVerilog with synchronous active-low reset, seed loading, tap buffering, and feedback shifting.	[106]
NS_PA_SS_05	Develop a parameterizable, synthesizable SystemVerilog module that captures eight bitwidth-wide input words, performs a fully unrolled one-cycle bubble sort on them when triggered, and presents the sorted vector plus optional valid flag.	[106]
NS_PA_SS_06	Design a synthesizable SystemVerilog module that reads two 3-element vectors serially, computes their dot product in one cycle after the sixth input, and drives an 18-bit result plus a one-cycle valid signal with defined reset behavior.	[106]

Continued on next page

Table 12 (continued from previous page)

Task-id	Task Summary	Ref
NS_PA_SS_07	Implement a synthesizable 8×8 register file in Verilog with one read port, one write port, valid-bit tracking, error flagging, and synchronous reset/clock behavior.	[106]
NS_PA_SS_08	Develop a fully synchronous, parameterized SystemVerilog module that detects a user-loaded 5-bit target pattern in a serial input stream and pulses a one-cycle 'seen' flag on each match.	[106]
NS_PA_SS_09	Create a fully synthesizable, parameterized SystemVerilog module that converts an N-bit Gray code input into its binary equivalent using a combinational XOR cascade.	[106]
NS_PA_SS_10	Implement a synthesizable SystemVerilog finite-state machine that reads a serial bitstream, tracks the value modulo 5, and raises dout when the running value is divisible by 5.	[106]
Topic: Analog Integrated Circuit Design		
TB_01	Size the devices in a folded-cascode OTA netlist by assigning concrete W, L, m, and R values to meet given performance specs under a fixed testbench.	[107]
TB_02	Design and size a folded-cascode OTA from scratch in SPICE format to meet specific performance specs using the tsmc18 process and a given testbench setup.	[107]
TB_03	Size a 5-transistor differential-to-single-ended OTA by assigning W, L, and m values to meet gain, bandwidth, CMRR, and ICMR specs using tsmc18 models in a fixed testbench.	[108]
TB_04	Design and size a 5-transistor OTA along with its bias circuit in a single SPICE netlist to meet performance specs under a unity-gain testbench.	[108]
TB_05	Size a common-source amplifier with resistive load in SPICE by tuning NMOS, resistor, and bias parameters to meet gain, bandwidth, and output voltage specs.	[109]
Topic: Robotics		
Ziheng_02	Determine the six joint angles of a 6-DOF serial robot that achieve a specified end-effector pose given the robot's screw axes and home transformation.	[110]
AM_02	Compute two time-parameterized, collision-free trajectories for 2 by 2 robots on a 30 by 30 grid from given start to end points in 20 time steps, avoiding static rectangular obstacles and dynamically moving pedestrians under a maximum per-step speed of 2 units in each axis.	
AM_03	Compute a time-stamped, grid-based, collision-free trajectory for a 2 by 2 robot on a 30 by 30 map with static rectangles and three moving pedestrians, starting at (17,2) and visiting goals A and B in any order without exceeding a per-step speed of 2 in x or y.	
HJ_01	Select refresh rate, acceleration, maximum velocity, and look-ahead distance settings to complete a 56 m lap in under 10 s with a maximum 1 m track-off error on a \$300 budget.	
XZ_01	Compute the shortest collision-free grid path for a small inspection robot across a 50×40 m construction map with specified walls and obstacles.	[87]

Continued on next page

Table 12 (continued from previous page)

Task-id	Task Summary	Ref
XZ_02	Compute a smooth, collision-free trajectory for an autonomous construction vehicle from $(0,0,\pi)$ to $(49,39,\pi/2)$ on a 50×40 m site with fixed obstacles, while respecting a 4 m turning radius and 0.5 m vehicle clearance.	[87]
XZ_03	Generate a collision-free shortest path on an integer grid in a Webots world from $(0,0,0)$ to $(49,39,0)$ around defined obstacles.	[87]
XZ_04	Generate a time-parameterized, collision-free 3D trajectory in a Webots world from a start to a goal point while respecting zone-based speed constraints.	[87]
YX_01	Compute the world-coordinate corners of a 2D sparse costmap and then use a Hybrid A* planner with nonholonomic constraints to find collision-free paths for given start and goal poses.	[87]
ZH_03	Determine wheel diameter, gear ratio, battery capacity, and robot mass for an electric ground robot to meet specified speed, incline, and endurance targets.	
Topic: Signal Processing		
qjlim2_01	Design a rectangular microstrip patch antenna on a Rogers laminate to resonate at 1.537 GHz with ≥ 50 MHz bandwidth, ≥ 3 dBi gain, and fit within a $100 \times 100 \times 10$ mm volume using a 50Ω coaxial feed.	[111], [112], [87]
qjlim2_02	Design a free-space strip dipole antenna centered at 3 GHz with at least 100 MHz bandwidth, ≥ 2 dBi gain, and fitting within a 100×100 mm footprint.	[113], [87]
qjlim2_04	Design a cylindrical monopole antenna resonant at 0.487 GHz with $S_{11} \leq -10$ dB, bandwidth ≥ 10 MHz, and gain ≥ 0 dBi.	[113], [87]
XG_09	Tune the parameters of Gaussian and triangular membership functions in a fuzzy inference system to improve edge detection performance (MSE, PSNR, SSIM) on a grayscale image relative to a Canny pseudo-ground truth.	[87]
YZ_01	Design a three-stage sample-rate converter—including a Farrow fractional resampler and two decimating FIR filters—to down-convert input rates (e.g., 150 MHz) to 30.72 MHz while meeting LTE passband, stopband, and EVM requirements.	[87]
YZ_02	Derive analytic formulas and determine coefficient sets for SG filter cutoff and half-magnitude width, then design four weighted SG filters meeting specified noise-vs-smoothness performance criteria.	[87]
YZ_03	Design a circularly polarized helical antenna in MATLAB’s Antenna Toolbox to operate from 1.3 to 2 GHz, meeting specified directivity and axial ratio requirements.	[87]
AB_01	Compute first-order statistical features (mean, variance, skewness, kurtosis) and the GLCM contrast for pixel intensities strictly inside a given polygonal ROI in a medical image patch.	
AB_02	Implement the OpenCV watershed algorithm in Python to segment the red heart pips and numerals from the white background in the specified image.	[114]

Continued on next page

Table 12 (continued from previous page)

Task-id	Task Summary	Ref
AB_03	Load a hard-coded image, find its largest contour, and compute both its approximate polygon and convex hull using OpenCV.	[114]
AV_02	Design a digital IIR filter to notch out a 120 Hz resonance in a 2000 Hz-sampled sensor signal while preserving ± 15 Hz around that frequency.	[115]
AV_03	Design an FIR anti-aliasing filter to allow downsampling a 48 kHz audio signal to 8 kHz while preserving the 0-3.5 kHz band with less than 3 dB ripple.	[115]
HC_03	Estimate the coefficients a, b, and c of a second-order polynomial $y = ax^2 + bx + c$ that best fits a given noisy 2D data set by minimizing mean squared error.	
JY_01	Design and implement an automated test stand using two linear polarizers, a quarter-wave plate, and rotary motors to control and measure changes in the Stokes vector of polarized light.	
JY_02	Design and apply a Gaussian smoothing filter, horizontal and vertical edge-detection kernels, and a thresholding step to produce an edge map from the given image array.	[116]
JY_03	Design a convolutional kernel that performs linear interpolation to fill zero-valued pixels in the corrupted Degree0 polarization image and evaluate its effect on AoLP and DoLP outputs.	
WJ_01	Design and implement a flexible OpenCV based filter pipeline choosing and parameterizing one or more denoising filters based on detected noise types.	
YX_03	Compute and combine SNR metrics, integration gains, and various radar losses to evaluate the detectability of a 1 m^2 target at 100 km for an S-band surveillance radar.	[87]

810 **NeurIPS Paper Checklist**

811 The checklist is designed to encourage best practices for responsible machine learning research,
812 addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove
813 the checklist: **The papers not including the checklist will be desk rejected.** The checklist should
814 follow the references and follow the (optional) supplemental material. The checklist does NOT count
815 towards the page limit.

816 Please read the checklist guidelines carefully for information on how to answer these questions. For
817 each question in the checklist:

- 818 • You should answer [Yes] , [No] , or [NA] .
- 819 • [NA] means either that the question is Not Applicable for that particular paper or the
820 relevant information is Not Available.
- 821 • Please provide a short (1–2 sentence) justification right after your answer (even for NA).

822 **The checklist answers are an integral part of your paper submission.** They are visible to the
823 reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it
824 (after eventual revisions) with the final version of your paper, and its final version will be published
825 with the paper.

826 The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation.
827 While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a
828 proper justification is given (e.g., "error bars are not reported because it would be too computationally
829 expensive" or "we were unable to find the license for the dataset we used"). In general, answering
830 "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we
831 acknowledge that the true answer is often more nuanced, so please just use your best judgment and
832 write a justification to elaborate. All supporting evidence can appear either in the main paper or the
833 supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification
834 please point to the section(s) where related material for the question can be found.

835 **IMPORTANT**, please:

- 836 • **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”**,
- 837 • **Keep the checklist subsection headings, questions/answers and guidelines below.**
- 838 • **Do not modify the questions and only use the provided macros for your answers.**

839 **1. Claims**

840 Question: Do the main claims made in the abstract and introduction accurately reflect the
841 paper’s contributions and scope?

842 Answer: [Yes]

843 Justification: We introduced ENGDESIGN benchmark in this work to benchmark the engi-
844 neering design capabilities of current LLMs. The dataset details have been discussed in
845 Section 2, and our experimental results are included in Section 3.

846 Guidelines:

- 847 • The answer NA means that the abstract and introduction do not include the claims
848 made in the paper.
- 849 • The abstract and/or introduction should clearly state the claims made, including the
850 contributions made in the paper and important assumptions and limitations. A No or
851 NA answer to this question will not be perceived well by the reviewers.
- 852 • The claims made should match theoretical and experimental results, and reflect how
853 much the results can be expected to generalize to other settings.
- 854 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
855 are not attained by the paper.

856 **2. Limitations**

857 Question: Does the paper discuss the limitations of the work performed by the authors?

858 Answer: [Yes]

859 Justification: We have discussed the limitations of our work in Appendix B.

860 Guidelines:

- 861 • The answer NA means that the paper has no limitation while the answer No means that
862 the paper has limitations, but those are not discussed in the paper.
- 863 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 864 • The paper should point out any strong assumptions and how robust the results are to
865 violations of these assumptions (e.g., independence assumptions, noiseless settings,
866 model well-specification, asymptotic approximations only holding locally). The authors
867 should reflect on how these assumptions might be violated in practice and what the
868 implications would be.
- 869 • The authors should reflect on the scope of the claims made, e.g., if the approach was
870 only tested on a few datasets or with a few runs. In general, empirical results often
871 depend on implicit assumptions, which should be articulated.
- 872 • The authors should reflect on the factors that influence the performance of the approach.
873 For example, a facial recognition algorithm may perform poorly when image resolution
874 is low or images are taken in low lighting. Or a speech-to-text system might not be
875 used reliably to provide closed captions for online lectures because it fails to handle
876 technical jargon.
- 877 • The authors should discuss the computational efficiency of the proposed algorithms
878 and how they scale with dataset size.
- 879 • If applicable, the authors should discuss possible limitations of their approach to
880 address problems of privacy and fairness.
- 881 • While the authors might fear that complete honesty about limitations might be used by
882 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
883 limitations that aren't acknowledged in the paper. The authors should use their best
884 judgment and recognize that individual actions in favor of transparency play an impor-
885 tant role in developing norms that preserve the integrity of the community. Reviewers
886 will be specifically instructed to not penalize honesty concerning limitations.

887 3. Theory assumptions and proofs

888 Question: For each theoretical result, does the paper provide the full set of assumptions and
889 a complete (and correct) proof?

890 Answer: [NA]

891 Justification: Our paper does not include theoretical results.

892 Guidelines:

- 893 • The answer NA means that the paper does not include theoretical results.
- 894 • All the theorems, formulas, and proofs in the paper should be numbered and cross-
895 referenced.
- 896 • All assumptions should be clearly stated or referenced in the statement of any theorems.
- 897 • The proofs can either appear in the main paper or the supplemental material, but if
898 they appear in the supplemental material, the authors are encouraged to provide a short
899 proof sketch to provide intuition.
- 900 • Inversely, any informal proof provided in the core of the paper should be complemented
901 by formal proofs provided in appendix or supplemental material.
- 902 • Theorems and Lemmas that the proof relies upon should be properly referenced.

903 4. Experimental result reproducibility

904 Question: Does the paper fully disclose all the information needed to reproduce the main ex-
905 perimental results of the paper to the extent that it affects the main claims and/or conclusions
906 of the paper (regardless of whether the code and data are provided or not)?

907 Answer: [Yes]

908 Justification: We have open-sourced our dataset and evaluation pipeline to reproduce our
909 work.

910 Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided open access to the benchmark data including all the task prompt, domains, and topics. In addition, we have provided a github repo contains all the evaluations logs and evaluation pipeline for each task of ENGDESIGN.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- 966 • At submission time, to preserve anonymity, the authors should release anonymized
967 versions (if applicable).
968 • Providing as much information as possible in supplemental material (appended to the
969 paper) is recommended, but including URLs to data and code is permitted.

970 **6. Experimental setting/details**

971 Question: Does the paper specify all the training and test details (e.g., data splits, hyper-
972 parameters, how they were chosen, type of optimizer, etc.) necessary to understand the
973 results?

974 Answer: [Yes]

975 Justification: We can explained our experimental setup in detail in Appendix E.1.

976 Guidelines:

- 977 • The answer NA means that the paper does not include experiments.
978 • The experimental setting should be presented in the core of the paper to a level of detail
979 that is necessary to appreciate the results and make sense of them.
980 • The full details can be provided either with the code, in appendix, or as supplemental
981 material.

982 **7. Experiment statistical significance**

983 Question: Does the paper report error bars suitably and correctly defined or other appropriate
984 information about the statistical significance of the experiments?

985 Answer: [Yes]

986 Justification: We report error bars in Table 10 and Table 11, which presents the detailed
987 evaluation results. Specifically, during evaluation, we ran each task three independent trials
988 to compute the mean and standard deviation results.

989 Guidelines:

- 990 • The answer NA means that the paper does not include experiments.
991 • The authors should answer "Yes" if the results are accompanied by error bars, confi-
992 dence intervals, or statistical significance tests, at least for the experiments that support
993 the main claims of the paper.
994 • The factors of variability that the error bars are capturing should be clearly stated (for
995 example, train/test split, initialization, random drawing of some parameter, or overall
996 run with given experimental conditions).
997 • The method for calculating the error bars should be explained (closed form formula,
998 call to a library function, bootstrap, etc.)
999 • The assumptions made should be given (e.g., Normally distributed errors).
1000 • It should be clear whether the error bar is the standard deviation or the standard error
1001 of the mean.
1002 • It is OK to report 1-sigma error bars, but one should state it. The authors should
1003 preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis
1004 of Normality of errors is not verified.
1005 • For asymmetric distributions, the authors should be careful not to show in tables or
1006 figures symmetric error bars that would yield results that are out of range (e.g. negative
1007 error rates).
1008 • If error bars are reported in tables or plots, The authors should explain in the text how
1009 they were calculated and reference the corresponding figures or tables in the text.

1010 **8. Experiments compute resources**

1011 Question: For each experiment, does the paper provide sufficient information on the com-
1012 puter resources (type of compute workers, memory, time of execution) needed to reproduce
1013 the experiments?

1014 Answer: [Yes]

1015 Justification: We have discussed the required compute resources in Appendix E.1.

1016 Guidelines:

- 1017 • The answer NA means that the paper does not include experiments.
1018 • The paper should indicate the type of compute workers CPU or GPU, internal cluster,
1019 or cloud provider, including relevant memory and storage.
1020 • The paper should provide the amount of compute required for each of the individual
1021 experimental runs as well as estimate the total compute.
1022 • The paper should disclose whether the full research project required more compute
1023 than the experiments reported in the paper (e.g., preliminary or failed experiments that
1024 didn't make it into the paper).

1025 **9. Code of ethics**

1026 Question: Does the research conducted in the paper conform, in every respect, with the
1027 NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

1028 Answer: [Yes]

1029 Justification: This work adheres to the NeurIPS Code of Ethics. All experiments were
1030 conducted responsibly, without harm to individuals or groups, and without the use of
1031 sensitive or personally identifiable data. We also carefully considered the potential societal
1032 impact of our methods and reported both the limitations and potential risks in the paper.

1033 Guidelines:

- 1034 • The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
1035 • If the authors answer No, they should explain the special circumstances that require a
1036 deviation from the Code of Ethics.
1037 • The authors should make sure to preserve anonymity (e.g., if there is a special consid-
1038 eration due to laws or regulations in their jurisdiction).

1039 **10. Broader impacts**

1040 Question: Does the paper discuss both potential positive societal impacts and negative
1041 societal impacts of the work performed?

1042 Answer: [Yes]

1043 Justification: We have discussed the broader impacts of our work in Appendix B.

1044 Guidelines:

- 1045 • The answer NA means that there is no societal impact of the work performed.
1046 • If the authors answer NA or No, they should explain why their work has no societal
1047 impact or why the paper does not address societal impact.
1048 • Examples of negative societal impacts include potential malicious or unintended uses
1049 (e.g., disinformation, generating fake profiles, surveillance), fairness considerations
1050 (e.g., deployment of technologies that could make decisions that unfairly impact specific
1051 groups), privacy considerations, and security considerations.
1052 • The conference expects that many papers will be foundational research and not tied
1053 to particular applications, let alone deployments. However, if there is a direct path to
1054 any negative applications, the authors should point it out. For example, it is legitimate
1055 to point out that an improvement in the quality of generative models could be used to
1056 generate deepfakes for disinformation. On the other hand, it is not needed to point out
1057 that a generic algorithm for optimizing neural networks could enable people to train
1058 models that generate Deepfakes faster.
1059 • The authors should consider possible harms that could arise when the technology is
1060 being used as intended and functioning correctly, harms that could arise when the
1061 technology is being used as intended but gives incorrect results, and harms following
1062 from (intentional or unintentional) misuse of the technology.
1063 • If there are negative societal impacts, the authors could also discuss possible mitigation
1064 strategies (e.g., gated release of models, providing defenses in addition to attacks,
1065 mechanisms for monitoring misuse, mechanisms to monitor how a system learns from
1066 feedback over time, improving the efficiency and accessibility of ML).

1067 **11. Safeguards**

1068 Question: Does the paper describe safeguards that have been put in place for responsible
1069 release of data or models that have a high risk for misuse (e.g., pretrained language models,
1070 image generators, or scraped datasets)?

1071 Answer: [NA]

1072 Justification: Our work does not involve the release of any pretrained models, generative
1073 systems, or large-scale scraped datasets that would present a high risk of misuse. ENGDE-
1074 SIGN is a benchmark comprising structured engineering design tasks, and all associated
1075 data is manually curated and domain-specific. It is intended solely for the evaluation of
1076 model capabilities in engineering design and does not raise the same risks associated with
1077 general-purpose generative models or unfiltered data releases.

1078 Guidelines:

- 1079 • The answer NA means that the paper poses no such risks.
- 1080 • Released models that have a high risk for misuse or dual-use should be released with
1081 necessary safeguards to allow for controlled use of the model, for example by requiring
1082 that users adhere to usage guidelines or restrictions to access the model or implementing
1083 safety filters.
- 1084 • Datasets that have been scraped from the Internet could pose safety risks. The authors
1085 should describe how they avoided releasing unsafe images.
- 1086 • We recognize that providing effective safeguards is challenging, and many papers do
1087 not require this, but we encourage authors to take this into account and make a best
1088 faith effort.

1089 12. Licenses for existing assets

1090 Question: Are the creators or original owners of assets (e.g., code, data, models), used in
1091 the paper, properly credited and are the license and terms of use explicitly mentioned and
1092 properly respected?

1093 Answer: [Yes]

1094 Justification: Tasks in ENGDESIGN are curated by individual task contributors, and the
1095 corresponding references are cited properly (see Table 12).

1096 Guidelines:

- 1097 • The answer NA means that the paper does not use existing assets.
- 1098 • The authors should cite the original paper that produced the code package or dataset.
- 1099 • The authors should state which version of the asset is used and, if possible, include a
1100 URL.
- 1101 • The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- 1102 • For scraped data from a particular source (e.g., website), the copyright and terms of
1103 service of that source should be provided.
- 1104 • If assets are released, the license, copyright information, and terms of use in the
1105 package should be provided. For popular datasets, paperswithcode.com/datasets
1106 has curated licenses for some datasets. Their licensing guide can help determine the
1107 license of a dataset.
- 1108 • For existing datasets that are re-packaged, both the original license and the license of
1109 the derived asset (if it has changed) should be provided.
- 1110 • If this information is not available online, the authors are encouraged to reach out to
1111 the asset's creators.

1112 13. New assets

1113 Question: Are new assets introduced in the paper well documented and is the documentation
1114 provided alongside the assets?

1115 Answer: [Yes]

1116 Justification: The paper introduces a new benchmark, ENGDESIGN, which includes a suite
1117 of engineering design tasks along with structured evaluation scripts. Each task is docu-
1118 mented with a clear task description, LLM response instruction, and evaluation codebase.
1119 Additionally, we provide metadata and instructions for running evaluations, reproducibility
1120 guidelines, and license information alongside the released assets. The assets are shared
1121 through publicly accessible repositories to support transparent and reproducible research.

1122 Guidelines:

- 1123 • The answer NA means that the paper does not release new assets.
1124 • Researchers should communicate the details of the dataset/code/model as part of their
1125 submissions via structured templates. This includes details about training, license,
1126 limitations, etc.
1127 • The paper should discuss whether and how consent was obtained from people whose
1128 asset is used.
1129 • At submission time, remember to anonymize your assets (if applicable). You can either
1130 create an anonymized URL or include an anonymized zip file.

1131 **14. Crowdsourcing and research with human subjects**

1132 Question: For crowdsourcing experiments and research with human subjects, does the paper
1133 include the full text of instructions given to participants and screenshots, if applicable, as
1134 well as details about compensation (if any)?

1135 Answer: [NA]

1136 Justification: Our study does not involve crowdsourcing nor research with human subjects.

1137 Guidelines:

- 1138 • The answer NA means that the paper does not involve crowdsourcing nor research with
1139 human subjects.
1140 • Including this information in the supplemental material is fine, but if the main contribu-
1141 tion of the paper involves human subjects, then as much detail as possible should be
1142 included in the main paper.
1143 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,
1144 or other labor should be paid at least the minimum wage in the country of the data
1145 collector.

1146 **15. Institutional review board (IRB) approvals or equivalent for research with human
1147 subjects**

1148 Question: Does the paper describe potential risks incurred by study participants, whether
1149 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)
1150 approvals (or an equivalent approval/review based on the requirements of your country or
1151 institution) were obtained?

1152 Answer: [NA]

1153 Justification: Our study does not involve crowdsourcing nor research with human subjects.

1154 Guidelines:

- 1155 • The answer NA means that the paper does not involve crowdsourcing nor research with
1156 human subjects.
1157 • Depending on the country in which research is conducted, IRB approval (or equivalent)
1158 may be required for any human subjects research. If you obtained IRB approval, you
1159 should clearly state this in the paper.
1160 • We recognize that the procedures for this may vary significantly between institutions
1161 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
1162 guidelines for their institution.
1163 • For initial submissions, do not include any information that would break anonymity (if
1164 applicable), such as the institution conducting the review.

1165 **16. Declaration of LLM usage**

1166 Question: Does the paper describe the usage of LLMs if it is an important, original, or
1167 non-standard component of the core methods in this research? Note that if the LLM is used
1168 only for writing, editing, or formatting purposes and does not impact the core methodology,
1169 scientific rigorosity, or originality of the research, declaration is not required.

1170 Answer: [NA]

1171 Justification: The core development of ENGDESIGN in this work does not involve LLMs as
1172 any important, original, or non-standard components.

1173 Guidelines:

- 1174 • The answer NA means that the core method development in this research does not
1175 involve LLMs as any important, original, or non-standard components.
1176 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
1177 for what should or should not be described.