MECHBENCH: A DOMAIN-SPECIFIC LLM BENCHMARK FOR PARALLEL MECHANISMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Parallel mechanisms pose distinctive challenges to large language models (LLMs), as valid reasoning depends jointly on mechanism topology, loop closure, geometry, joint types, and modeling assumptions. Yet existing LLM benchmarks provide little systematic coverage of such domain-specific knowledge, constraint-sensitive analysis, and constructive design. To fill this gap, we introduce MECHBENCH, a 2,541-question benchmark for evaluating LLMs in parallel mechanisms across professional knowledge, mechanism analysis, and design. MECHBENCH contains standard and challenge tiers spanning six capability categories and nine executable analysis and design tasks. It is constructed through MECHFORGE, a domain-grounded framework that combines expert-organized knowledge, LLM-assisted question generation, programmatic instance generation, two-stage expert review, and task-aware scoring with exact matching, rubric-based evaluation, and program verification. We further develop MECHGPT as a diagnostic domain-adapted model. Experiments show that strong knowledge performance does not directly translate into executable mechanism reasoning: leading closed-service models achieve over 95 on the standard tier but only 47–60 on the challenge tier, while domain adaptation substantially improves MechGPT over its Qwen3-8B baselines but leaves a large gap in constrained analysis and design. These results establish MECHBENCH as a systematic testbed for measuring domain-specific LLM capabilities in mechanism science.

1 INTRODUCTION

Parallel-mechanism reasoning depends jointly on topology, loop closure, geometry, joint types, and modeling assumptions. A valid evaluation must test whether an LLM preserves these conditions, selects an appropriate method, and reaches a compatible conclusion. General benchmarks such as MMLU, GPQA, MMLU-Pro, and Humanity's Last Exam provide broad knowledge tests but do not isolate these requirements (Hendrycks et al., 2020; Rein et al., 2023; Wang et al., 2024; Phan et al., 2025).

The domain presents specific analytical distinctions. Closed-loop singularities and Jacobian-based performance measures depend on configuration and interpretation (Gosselin & Angeles, 1990; Merlet, 2006); configurable platforms introduce internal motion that differs from output mobility (Mohamed & Gosselin, 2005; Lambert & Herder, 2016). Generalized mechanisms further involve coupled sub-chains and reconfigurable working phases (Xia et al., 2025b; Tian et al., 2024). Evidence is distributed across papers, textbooks, exercises, and code with differing symbols and assumptions, making direct QA collection vulnerable to incomplete conditions and uneven coverage.

To address these challenges, we propose MECHFORGE, a framework for constructing domain-specific benchmarks and instruction datasets from professional materials (Figure 1). Building on knowledge-organized data construction (Bi et al., 2024; Guo et al., 2025), it combines LLM-assisted architecture expansion and evidence-grounded QA generation with programmatic instances, model checking, and two-stage expert review. The checks preserve connectivity, geometry, constraint compatibility, and the distinction between whole-mechanism mobility, output motion, and internal motion.

MechBench is the central evaluation artifact, designed to distinguish domain knowledge from constraint-sensitive mechanism analysis and constructive design; MechForge specifies its construc-

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

tion methodology, and MechGPT serves as a diagnostic instance for studying capability changes after domain adaptation. The contributions are:

- We introduce **MechBench**, a 2,541-question benchmark for parallel mechanisms that separates domain knowledge from condition-sensitive analysis and constructive design, covering topology, mobility, modeling, diagnosis, mechanism modification, completion, and constrained synthesis.
- We propose **MechForge**, a domain-grounded construction methodology combining knowledge organization, LLM-assisted generation, two-stage expert review, and executable checks of topology, geometry, constraints, and distinct mobility concepts.
- We evaluate closed-service and open-weight models, including the domain-adapted **MechGPT**, using mean scores and capability/task profiles. The results expose a clear gap between high professional-knowledge performance and executable, constraint-consistent mechanism reasoning and design.

2 Related Work

LLM Benchmarking. General benchmarks assess broad knowledge and expert reasoning (Hendrycks et al., 2020; Rein et al., 2023; Wang et al., 2024; Phan et al., 2025). SciKnowEval organizes scientific capabilities by cognitive operation (Feng et al., 2025), while SciR generates verifiable questions from formalized objects and scientific text (Beckmann et al., 2026). Open-ended scoring introduces judge biases (Zheng et al., 2023); G-Eval and Prometheus 2 address evaluation through structured steps and explicit rubrics (Liu et al., 2023; Kim et al., 2024). MechBench combines exact matching, program verification, and item-specific rubrics, with recorded scoring configurations and sampled human validation.

Domain-specific LLM and Data Construction. Retrieval supplies external evidence at inference time (Lewis et al., 2020); continued pretraining and instruction tuning incorporate domain knowledge into model parameters (Gururangan et al., 2020; Azerbayev et al., 2024; Wu et al., 2024; Cheng et al., 2023). LoRA and GaLore reduce adaptation costs (Hu et al., 2021; Zhao et al., 2024). Self-Instruct and Bonito generate instruction data from text (Wang et al., 2023; Nayak et al., 2024), while OceanGPT and SDBench link knowledge organization to training and evaluation (Bi et al., 2024; Guo et al., 2025). Engineering benchmarks require document-grounded constraints (Doris et al., 2025), complementing executable robotics interfaces such as Code as Policies (Liang et al., 2023). MechBench focuses on condition-sensitive mechanism reasoning and records source relationships for overlap and contamination auditing (Oren et al., 2024).

Parallel-Mechanism Analysis and Synthesis. Classical studies analyze closed-loop singularities (Gosselin & Angeles, 1990), Jacobian-based performance (Merlet, 2006), and configurable-platform topology, mobility, and overconstraints (Mohamed & Gosselin, 2005; Lambert & Herder, 2016). Later work combines graph connectivity with screw constraints for coupled sub-chains (Tian et al., 2017), matches limb constraints to platform motion (Tian & Zhang, 2020), and studies reconfigurable working phases (Tian et al., 2021). Motion-transmission synthesis and topological mapping networks connect structure to desired outputs (Xia et al., 2025a; 2026). These distinctions guide the structural and analytical checks in MechBench.

3 MechForge Construction Framework

Figure 1 organizes MechForge into knowledge-architecture construction and benchmark establishment, with a separate branch connecting the resulting resources to model adaptation and evaluation.

3.1 Knowledge Architecture and Source Organization

1. **Source Allocation.** Organize papers, surveys, textbooks, teaching materials, and researcher-written generators by use in training, knowledge organization, and question generation; record source relationships.

2. **Domain Architecture Expansion.** An LLM proposes topics spanning topology, geometry, constraints, mobility, singularities, synthesis, and diagnosis, distinguishing rigid platforms from coupled, configurable, or articulated architectures.
3. **Evidence Retrieval.** Retrieve and prioritize supporting passages, retaining joint-axis geometry, platform assumptions, and working phases that determine each relation's applicability.
4. **Domain Architecture Investigation.** Researchers check the proposed architecture against evidence, refine topic boundaries, and assess coverage. Technical topics and K1–K6 capability labels remain distinct.

3.2 Establishment of the Benchmark Dataset

An LLM receives a topic, prioritized source evidence, and response-format requirements to generate questions, proposed answers, and scoring points. Programmatic generators supply complementary instances with controlled answers. Both paths follow the same review sequence:

1. **QA Generation.** Generate choice, fill-in, or constructed-response items. State the required base, platform, limb connectivity, joint types, geometry, and actuation; retain shared links and joints in coupled sub-chains.
2. **LLM Check.** A second LLM checks evidence, condition sufficiency, answer consistency, and format. Mechanism checks cover loop closure, constraint independence, platform-motion compatibility, and applicable input–output relations.
3. **Two-Stage Expert Review.** Two researchers sequentially review every item, resolve disagreements, and revise or reject invalid candidates. They distinguish whole-mechanism mobility, output and internal motion, instantaneous and finite motion, and reconfiguration phases.
4. **Labels and Rubrics.** Confirm K1–K6 capabilities, L1–L4 difficulty, reference answers, equivalent expressions, and scoring points. Intentionally under-specified items must identify the missing condition and its consequence.
5. **Dataset Splits.** Assign accepted items to development, standard, or challenge tiers; retain source-group identifiers for overlap auditing, and validate fields, answer keys, legal scores, and coverage.

4 Case Study: MechBench and MechGPT

We apply MechForge to parallel mechanisms and connect the resulting benchmark to MechGPT. Adaptation starts from Qwen3-8B-Base through domain continued pretraining followed by supervised fine-tuning. Main evaluation covers the 1,791-question standard tier and nine tasks in the 550-question challenge tier; development questions are used only for protocol debugging.

4.1 Domain Corpus

We assemble 10,379 professional papers, textbooks, and teaching materials. The main continued-pretraining collection contains 9,869 documents, including 93 textbooks and supplementary exercises, with professional papers constituting the remainder. Conversion to unified text removes author information, headers, footers, references, empty lines, and extraction noise while retaining technical relations and their assumptions.

Instruction-data sources comprise 202 English papers, 28 deduplicated textbooks, and 12 sets of course materials; the papers yield 56,927 QA pairs. Researcher-written examples, QA transformation, and model-assisted generation followed by cleaning supply domain dialogues. Training records use a unified conversational representation and the Qwen3 chat template.

4.2 Parallel-Mechanism Knowledge Architecture

Figure 2 illustrates the topic hierarchy. Capability, difficulty, response format, and scoring path are labeled independently. Six capabilities distinguish **K1**, terms and symbols; **K2**, structure and

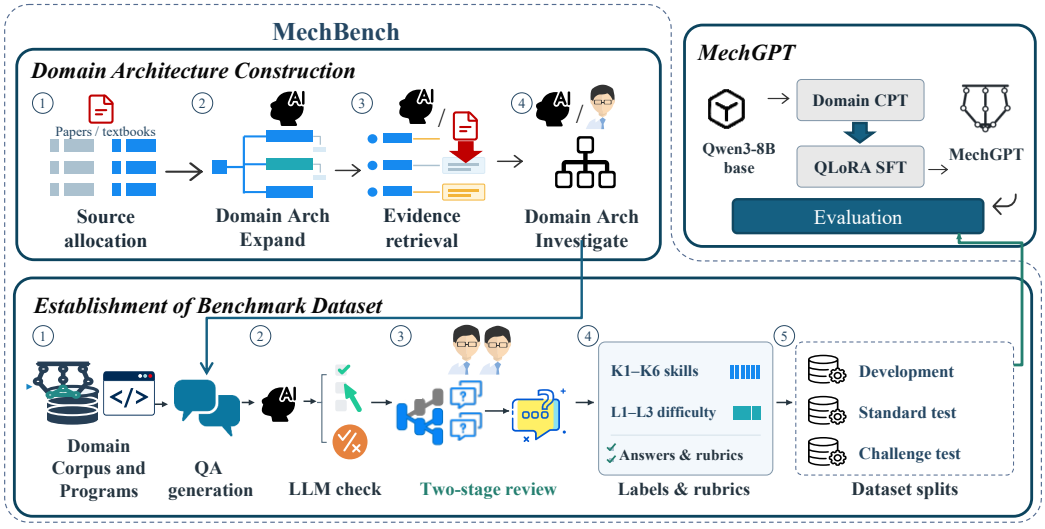


Figure 1: MechForge workflow. Sources feed LLM-assisted knowledge organization and evidence-grounded or programmatic question generation. LLM checking and two-stage researcher review confirm conditions, answers, labels, and rubrics before tier assignment; reviewed QA can also form instruction data. Domain pretraining and QLoRA adapt MechGPT, which is evaluated by MechBench.

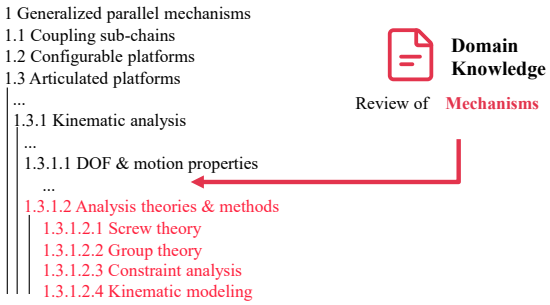


Figure 2: Knowledge branch linking coupled and configurable architectures to motion properties and screw-, group-, constraint-, and model-based analysis.

topology; **K3**, mobility and kinematics; **K4**, modeling, calibration, and analysis; **K5**, diagnosis and information sufficiency; and **K6**, synthesis and design trade-offs. K3 asks for a motion or quantity; K4 asks how to model or obtain it. Expert-assigned difficulty spans L1 (recognition or one-step application) to L4 (complex synthesis, proof, or path verification), independently of model accuracy.

4.3 MechBench

Each item has an identifier, stem, required response format, and applicable choices or fill-in fields. Separately managed scoring resources contain reference answers, acceptable alternatives, scoring points, and error categories. Constructed responses are explanations, formulas, or derivations.

MechBench contains 200 development, 1,791 standard, and 550 challenge questions (2,541 total; Table 1). The development tier supports protocol debugging. The original 1,991 items comprise development and standard tiers; the separate 550-item set is the challenge tier.

Standard questions cover six capabilities; K3 and K4 together account for 61.98%. The challenge tier covers nine analysis and design tasks (Figure 6). Tier membership is distinct from L1–L4 difficulty. Source-group identifiers support overlap auditing.

Figure 3 illustrates the mechanism content assessed by the nine challenge tasks.

Table 1: Composition of MECHBENCH.

(a) Benchmark tiers

All questions: $N = 2,541$

Tier	Questions	Share (%)
Development	200	7.87
Standard	1,791	70.48
Challenge	550	21.65
Total	2,541	100.00

(b) Professional capabilities

Standard questions: $N = 1,791$

	Capability	Share (%)
K1	Terms/Symbols	10.94
K2	Structure/Topology	14.41
K3	Mobility/Kinematics	44.11
K4	Modeling/Analysis	17.87
K5	Diagnosis/Sufficiency	7.43
K6	Design Trade-offs	5.25

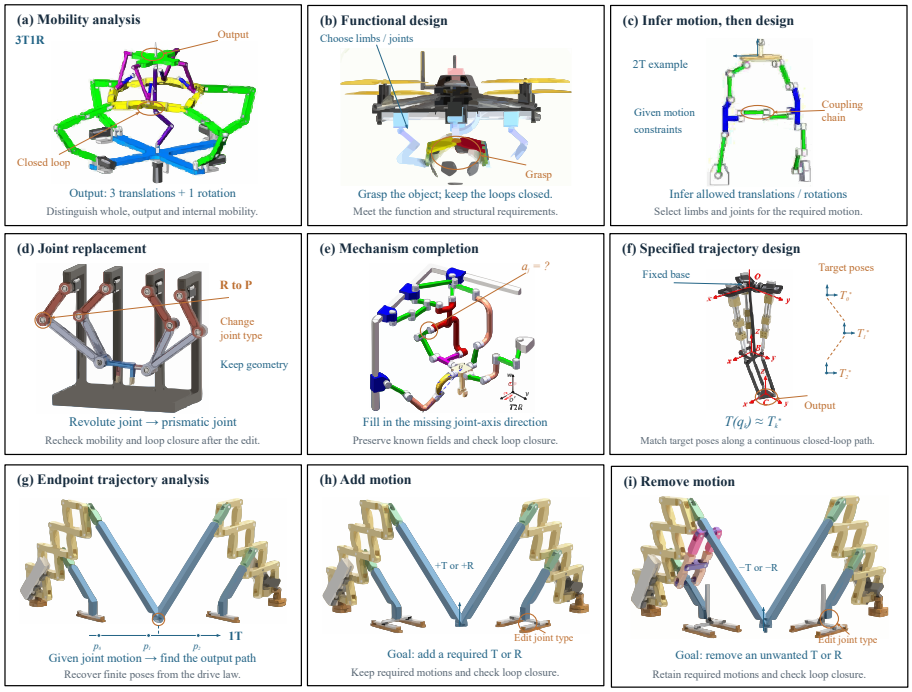


Figure 3: Mechanism illustrations for the nine challenge tasks, spanning analysis, modification, completion, and constrained design.

Table 2 summarizes question types and representative stems. Analysis and design group the nine challenge tasks.

4.4 TASK-AWARE SCORING

Development and standard answer-only choices use exact matching; explained choices, fill-in questions, and constructed responses use item-specific 0–5 rubrics normalized to 0–100. Challenge questions use deterministic verification on a 0–100 scale. Both assess correctness under stated structural, geometric, and motion conditions. Figure 4 shows the routing and normalization steps.

Let I_m contain numerically scored items and $s_{mi} \in [0, 100]$. We report the normalized mean score (micro-average), and the equal-weight mean over capability or task groups G:

$$S_m = \frac{\sum_{i \in I_m} s_{mi}}{|I_m|}, \qquad M_m = \frac{1}{|G|} \sum_{g \in G} \frac{\sum_{i \in I_m \cap g} s_{mi}}{|I_m \cap g|}. \tag{1}$$

Table 2: Question formats, counts, and abbreviated example stems.

Type	Count	Example question
Single-choice	870	Before loop closure, a parallel mechanism has 10 independent joint-motion parameters; closure makes 7 dependent. What is its mobility? A. 3; B. 7; C. 10; D. 17.
Multiple-choice	466	Which methods derive the platform-velocity Jacobian? A. Loop equations; B. Reciprocal screws; C. Plücker coordinates; D. Limb counting alone.
Fill-in	358	An ideal spherical joint permits relative rotations about its center. It has ___ relative rotational degrees of freedom.
Constructed response	297	What defines a general screw? Which factors remain for a normalized unit screw?
Analysis	333	Three UP limbs with disjoint links connect the base and platform… Report first- and second-order mobility, output and self-motion mobility, and an output-twist basis.
Design	217	A platform must satisfy $C\xi = 0$ without extra locking… Infer its motion and design a four-limb mechanism with geometry and motion-screw justification.
Total	**2,541**	

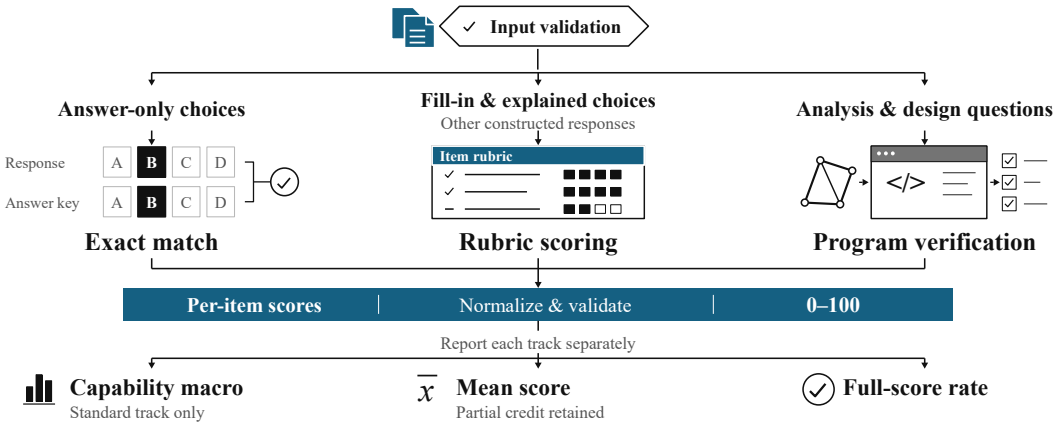


Figure 4: Scoring routes: exact matching, item rubrics, or program checks, with scores normalized to 0–100.

Standard capability macro uses six groups and challenge task macro uses nine. Micro means remain primary; macro scores provide a distribution-balanced supplement. Unresolved execution failures have no numerical score, whereas invalid submitted structures receive zero.

Executable mechanism checks. Mechanisms encode links, a fixed base, an output platform, and R/P/U/S joints with incident links, positions, and axes. Analysis answers distinguish first-order candidate mobility, second-order-compatible mobility, output mobility, and internal self-motion; design answers provide a mechanism payload. MechOmni-based checks assess schema validity, closure, limb and joint constraints, output screw spaces, and applicable finite-motion or trajectory conditions. Screw-space comparison accepts equivalent bases rather than requiring identical vector order, and rubric-weighted checks award partial credit while retaining designated design hard gates.

Algorithm 2: Task-Aware Item Scoring.

Input: Response, reference, and item configuration. **Output:** Score and evidence. Validate identity; route to matching, the recorded rubric-judge configuration, or program checks; adjudicate where configured; retain criterion evidence and execution status; normalize and aggregate by tier, capability, and task. Faithful extraction may recover explicit fields but may not solve, correct, or complete the answer.

Table 3: Expert recheck on the same 150 questions per model (50 per format); Δ is expert minus historical mean.

Model	Historical	Expert	Δ	Multiple-choice	Fill-in	Open-ended
GPT-5.5	93.07	92.43	-0.64	98.40	82.20	96.70
GPT-5.6-Sol	94.80	91.07	-3.73	96.40	79.00	97.80
GPT-5.6-Terra	93.60	90.47	-3.13	98.00	78.20	95.20
GPT-OSS-20B	73.20	70.43	-2.77	82.00	56.20	73.10
Qwen3-8B Chat	61.06	57.37	-3.69	62.00	44.40	65.70
Qwen3-8B Base	42.94	43.13	$+0.19$	69.60	19.00	40.80
MechGPT	62.53	63.97	$+1.44$	79.60	51.60	60.70

5 Experiment

5.1 Implementation Details

Implementation of MechBench. LLM- and program-generated candidates undergo a full first researcher review and a second-stage full review, covering conditions, answers, labels, and rubrics. The second stage is not claimed to be blinded or statistically independent. Field validation is separate from professional-content review.

Implementation of MechGPT. We select Qwen3-8B-Base as the foundation model (Yang et al., 2025). MechGPT first undergoes continued pretraining on the collected mechanism-domain corpus, followed by supervised fine-tuning. Training uses eight NVIDIA GeForce RTX 4090 GPUs with 24 GB of memory each. Continued pretraining runs for one epoch with a learning rate of 1×10^{-6}. Supervised fine-tuning uses 4-bit QLoRA with a learning rate of 1×10^{-5} for three epochs.

Compared models. Both collections compare GPT-5.5, GPT-5.6-Sol, GPT-5.6-Terra, GPT-OSS 20B, Qwen3-8B Base and Chat, and MechGPT.

Evaluation protocol. The standard tier combines answer matching and rubric scoring under a unified evaluation configuration; challenge tasks use program verification.

5.2 Benchmark Composition and Quality

Development questions support protocol debugging and are excluded from main results. Standard questions cover six capabilities; challenge questions cover nine tasks. Construction review confirms each item's capability and difficulty labels, evidence, conditions, reference answers, and scoring operability.

To reduce training–evaluation contamination, we manually separated training and Standard/Challenge materials by source group during curation. An original paper, textbook chapter, code document, or generation dependency and its rewrites, QA pairs, program transformations, and generated variants were assigned to the same group. Standard/Challenge source groups were withheld from training; splitting only by question identifier was not used.

Scoring follows two validation routes. Objective and program-scored questions are processed by fixed answer rules or deterministic mechanism checks. For open-ended questions scored with item-specific rubrics, domain experts reviewed a sample of the model-based judgments and confirmed their consistency with the reference answers and scoring criteria. For this calibration check, every model was compared on the same 150 questions: 50 multiple-choice, 50 fill-in, and 50 open-ended items. Table 3 reports the historical and expert-reviewed aggregate means; the comparison is descriptive and does not replace item-level agreement analysis.

5.3 Benchmark Results and Analysis

Knowledge performance does not directly translate to executable mechanism reasoning. GPT-5.6-Sol achieves the highest scores on both the standard and challenge tiers, with scores of 96.12 and 59.68, respectively. MechGPT obtains 62.35 on the standard tier but only 18.30 on the challenge tier, revealing a substantial gap between conventional domain knowledge and executable mechanism reasoning.

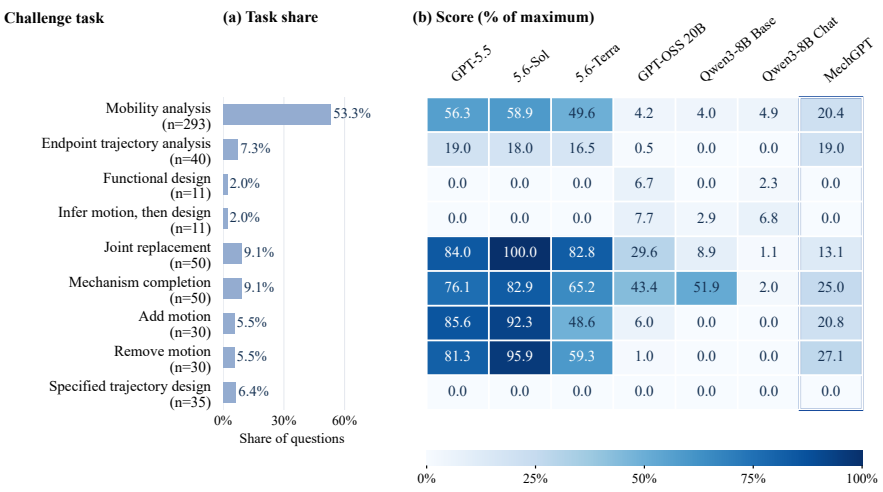


Figure 6: Challenge-task shares and normalized means; Qwen3-8B Base and Qwen3-8B Chat are shown with their full model names.

Table 4: Standard capability means (0–100; $n = 1,791$). Base/Chat denote Qwen3-8B. GPT-OSS 20B values use the supplied capability-level means.

Capability	GPT-5.5	Sol	Terra	GPT-OSS 20B	Base	Chat	MechGPT
K1 (n=196)	88.6	88.4	87.0	69.79	41.1	51.8	69.9
K2 (n=258)	93.7	95.3	94.9	83.58	43.5	50.2	68.8
K3 (n=790)	97.6	97.3	96.9	81.05	39.8	45.4	59.0
K4 (n=320)	96.7	96.9	94.6	85.99	38.7	49.8	61.8
K5 (n=133)	98.2	98.6	98.9	87.33	54.4	66.4	61.1
K6 (n=94)	94.9	98.5	96.2	71.49	40.3	53.0	60.6

On the challenge tier, Sol, GPT-5.5, and Terra score 59.68, 55.15, and 46.95. MechGPT scores 18.30, compared with GPT-OSS 20B's 8.86, Base's 7.36, and Chat's 3.25. Chat's advantage over Base on standard questions reverses on the challenge tier, showing that stronger knowledge-oriented performance does not ensure reliable structural analysis or design.

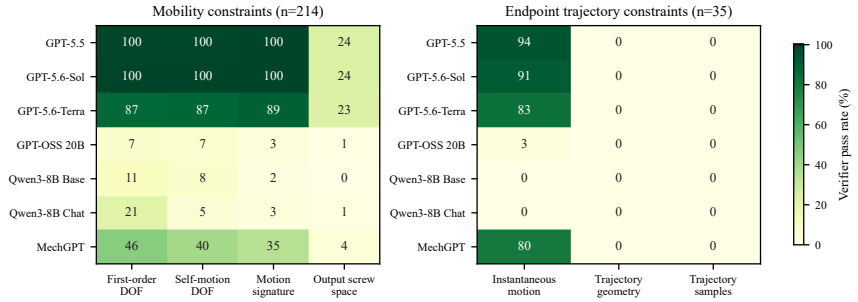


Figure 5: Verifier pass rates on paired mobility ($n = 214$) and endpoint-trajectory ($n = 35$) questions.

Constraint-level diagnostic. Figure 5 summarizes the verifier-level pass rates. To explain the gap between aggregate challenge scores, we inspect verifier criteria on paired subsets. On 214 mobility questions with complete evidence for all models, GPT-5.6-Sol passes all scalar mobility and motion-signature checks, but its strict output screw-space pass rate is 23.8%. MechGPT reaches 46.3%, 39.7%, and 35.0% on first-order mobility, self-motion, and motion-signature checks, respectively, but 4.2% on output screw space. These results indicate that recognizing a motion class is easier than constructing a geometry-consistent six-dimensional motion basis. On 35 paired endpoint-trajectory questions, instantaneous-motion pass rates remain high for the strongest models, whereas trajectory geometry and trajectory samples receive no passes in the current records.

Table 5: Feasibility checks for specified-trajectory design references.

Check	Passed	Check	Passed
Payload parsing	35/35	Target mobility/output	35/35
Topology constraints	35/35	Finite continuation	35/35
Initial closure	35/35	Pose trajectory	35/35
Independent constructions	10/10		

Table 4 compares capability profiles on the standard tier. MechGPT exceeds Base in all six capabilities and Chat in five. Gains over Chat are largest for structure/topology (K2, +18.57) and terms/symbols (K1, +18.06); diagnosis/sufficiency (K5) remains weaker, at 61.05 versus 66.43.

On the challenge tier, mobility analysis occupies 53.27% of questions. MechGPT scores 20.39 versus Base's 4.03 and Chat's 4.91, but below Sol's 58.86. It exceeds both Qwen baselines on mobility, endpoint trajectory analysis, joint replacement, adding motion, and removing motion. Mechanism completion reverses that pattern: 25.02 versus Base's 51.90. Endpoint analysis reaches only 19.00, and no evaluated model obtains credit for specified trajectory design.

Feasibility sanity check. Zero scores on the 35 specified-trajectory designs do not imply infeasibility: all references pass the checks in Table 5, and ten independently selected instances admit full-score executable constructions. Perturbing axis direction, closure geometry, joint type, or target trajectory is rejected or capped. The difficulty lies in jointly synthesizing valid topology, closure, target mobility, and a continuous trajectory certificate.

5.4 Future Work

Based on feedback from parallel-mechanism researchers during construction and internal evaluation, we will refine MechForge and MechGPT in two directions. First, we will adopt stronger open-weight foundations to improve deeper reasoning and analysis. Second, we will extend both systems to mechanism drawings, CAD and geometric representations, simulation trajectories, and experimental sensor signals.

6 Conclusion

In this paper, we introduce MechBench, a benchmark for evaluating professional knowledge, condition-sensitive analysis, and constructive design in parallel mechanisms. MechBench contains 2,541 questions organized into development, standard, and challenge tiers and combines exact matching, rubric scoring, and program verification. Its standard and challenge tracks reveal a substantial gap between conventional domain knowledge and executable, constraint-consistent mechanism reasoning. MechForge provides the construction methodology behind the benchmark, while MechGPT serves as a domain-adapted diagnostic model.

7 Limitation

MechForge organizes domain evidence and structural conditions into a benchmark that covers knowledge comprehension, mechanism analysis, and design. However, the framework does not yet automatically control the quality of the corpus used for continued pretraining. Selecting reliable sources and checking the adequacy of mechanism descriptions still require expert judgment. In addition, MechBench currently evaluates text and structured mechanism representations rather than multimodal inputs. This limits its coverage of engineering problems that require interpreting mechanism drawings, spatial geometry, and physical interaction.

For MechGPT, domain adaptation improves knowledge and analysis but does not consistently extend to completion or constrained synthesis. Future versions will add geometric representations and training examples with explicit constraint checks.

A CHALLENGE RESPONSE VISUALIZATION

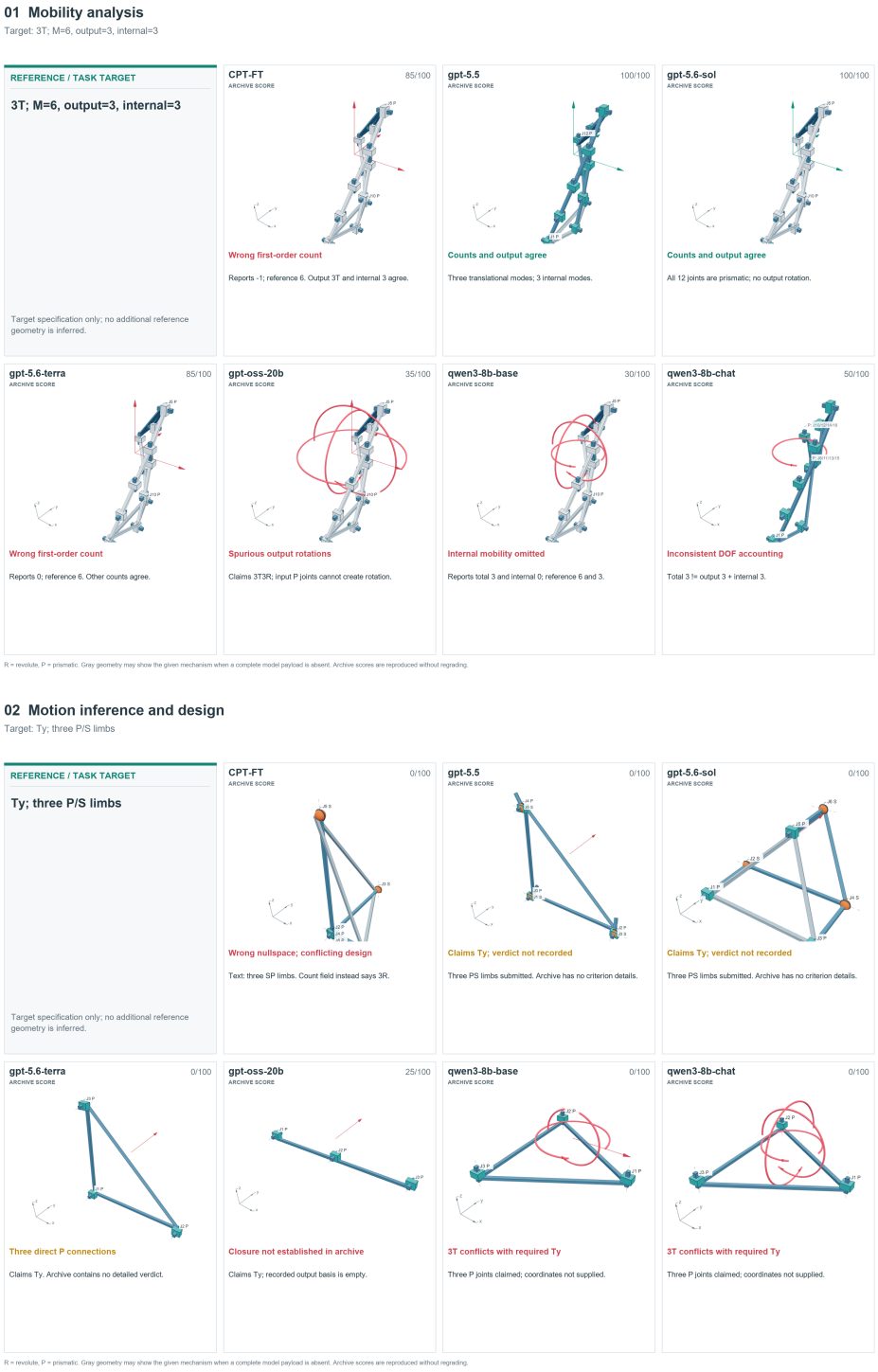


Figure 7: Challenge response visualizations for categories 1–2.

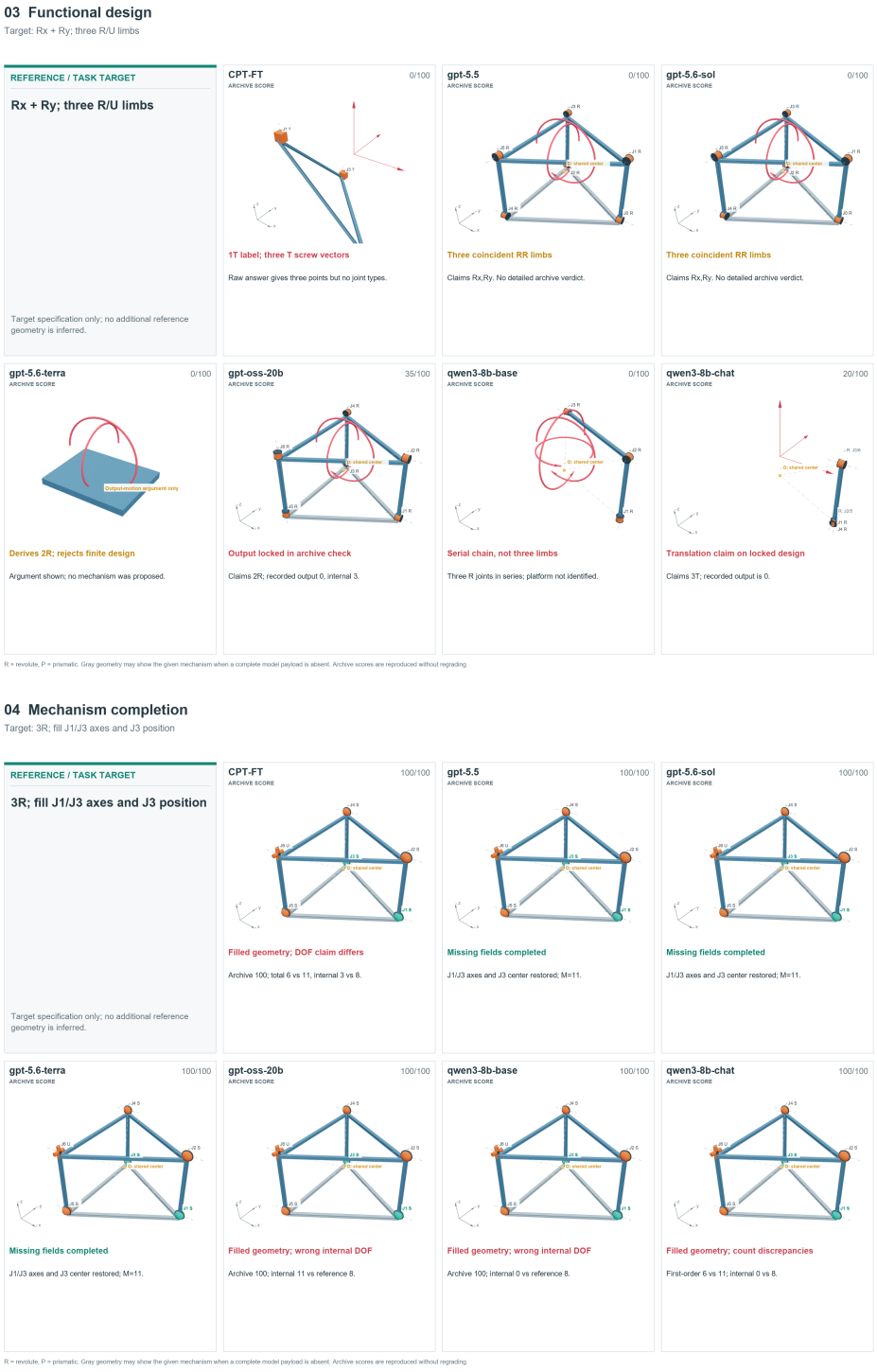


Figure 8: Challenge response visualizations for categories 3–4.

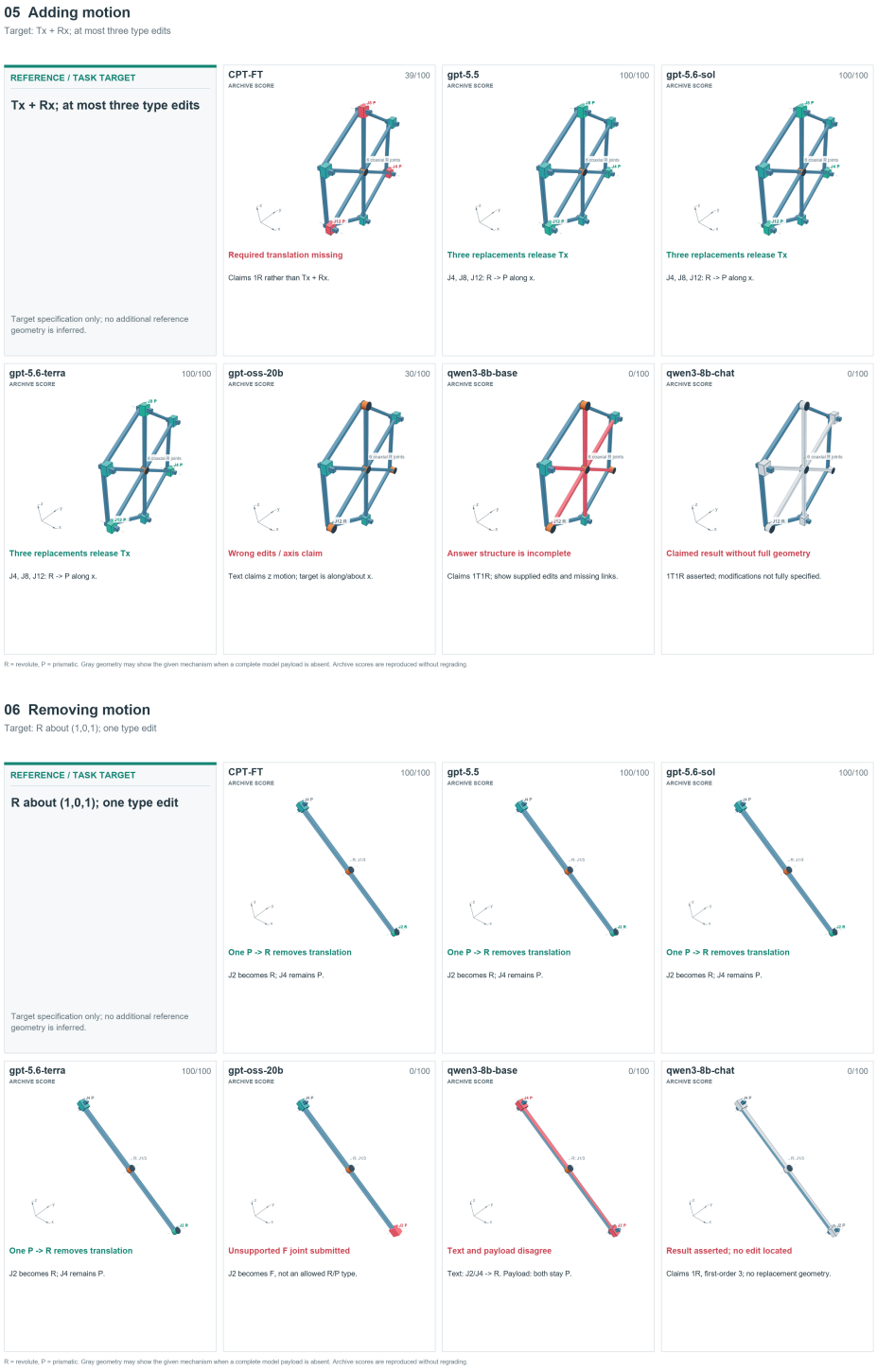


Figure 9: Challenge response visualizations for categories 5–6.

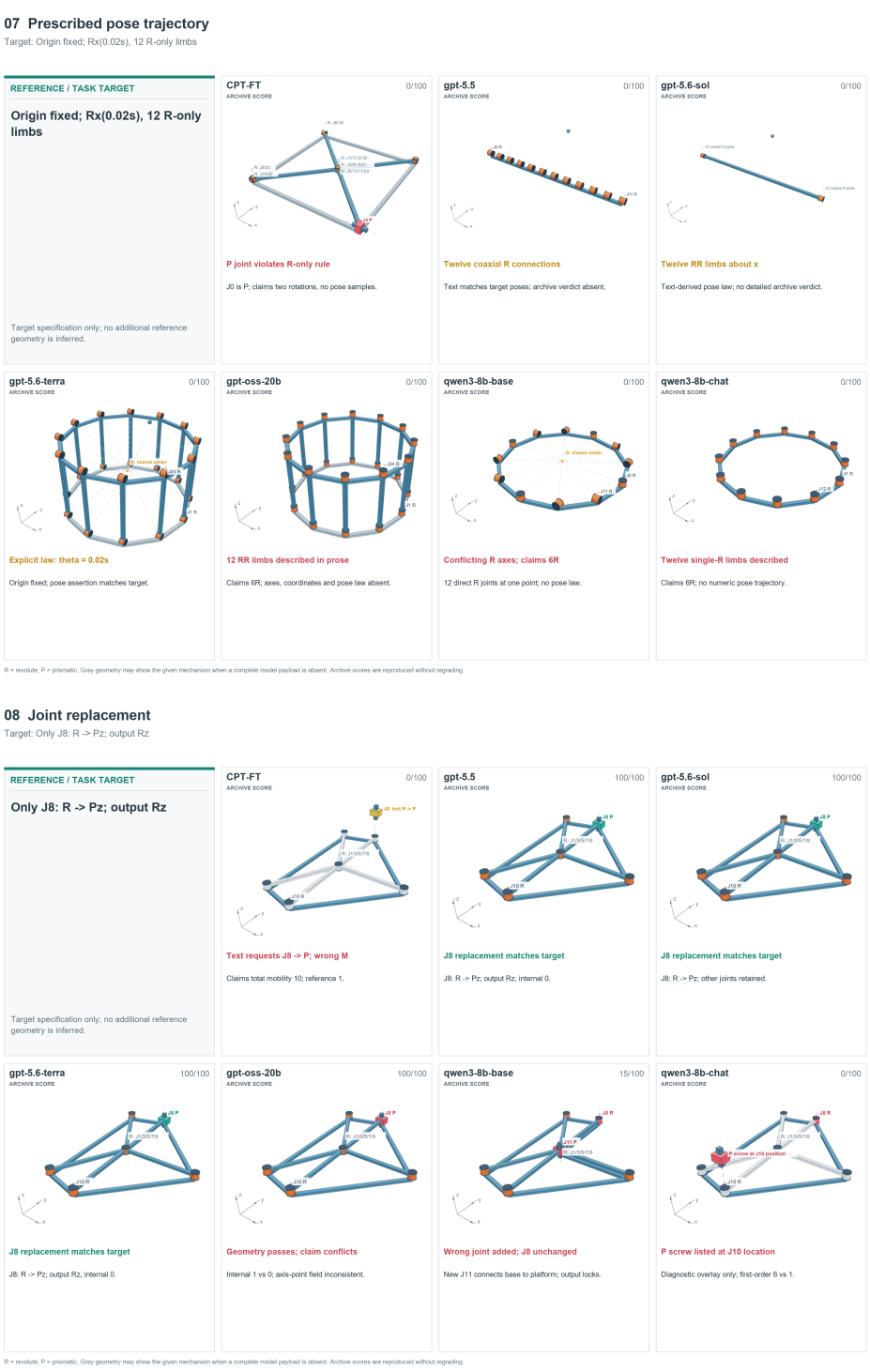


Figure 10: Challenge response visualizations for categories 7–8.



Figure 11: Challenge response visualization, category 9.

AI USE STATEMENT

Generative AI tools assisted candidate-question generation, model cross-checking, format conversion, and language editing. All candidate questions, reference answers, claims, citations, and AI-assisted content were reviewed by the authors or domain researchers. The authors remain responsible for the final manuscript and all AI-assisted artifacts.

REPRODUCIBILITY STATEMENT

The 200 development questions are used only for protocol debugging and are excluded from reported benchmark results. Standard and challenge questions, reference answers, rubrics, extraction prompts, and program-verifier assets remain in the private server package.

ETHICS STATEMENT

The study uses professional-source-driven dataset construction and external model services. Data and code release will follow source licenses and service terms. Reference answers and item-level scoring resources remain under organizer control. External judge services are used only after confirming data-use permissions and processing terms, and unverified model output is not treated as an engineering conclusion.

REFERENCES

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Qiaochu Jiang, Jia Deng, Stella R Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *International Conference on Learning Representations*, volume 2024, pp. 40622–40649, 2024.

Pierre Beckmann, Marco Valentino, and Andre Freitas. Scir: A controllable benchmark for scientific reasoning in llms. *arXiv preprint arXiv:2606.13020*, 2026.

Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. Oceangpt: A large language model for ocean science tasks. pp. 3357–3372, 2024.

Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models to domains via reading comprehension. 2023.

Anna C Doris, Daniele Grandi, Ryan Tomich, Md Ferdous Alam, Mohammadmehdi Ataei, Hyunmin Cheong, and Faez Ahmed. Designqa: A multimodal benchmark for evaluating large language models' understanding of engineering documentation. *Journal of Computing and Information Science in Engineering*, 25(2):021009, 2025.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: A comprehensive dataset for evaluating scientific knowledge of large language models. 2025.

C. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *IEEE Transactions on Robotics and Automation*, 6(3):281–290, 1990. doi: 10.1109/70.56660.

Cheng Guo, Hu Kai, Shuxian Liang, Yiyang Jiang, Yi Gao, Xian-Sheng Hua, and Wei Dong. Sdbench: A survey-based domain-specific llm benchmarking and optimization framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13492–13506, 2025.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8342–8360, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. pp. 4334–4353, 2024.

Patrice Lambert and Just L Herder. Parallel robots with configurable platforms: fundamental aspects of a new class of robotic architectures. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(3):463–472, 2016.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. volume 33, pp. 9459–9474, 2020.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. pp. 9493–9500, 2023.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 2511–2522, 2023.

Jean-Pierre Merlet. Jacobian, manipulability, condition number, and accuracy of parallel robots. 2006.

M.G. Mohamed and C.M. Gosselin. Design and analysis of kinematically redundant parallel manipulators with configurable platforms. *IEEE Transactions on Robotics*, 21(3):277–287, 2005. doi: 10.1109/tro.2004.837234.

Nihal Nayak, Yiyang Nan, Avi Trost, and Stephen Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12585–12611, 2024.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. 2024:16354–16372, 2024.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Chunxu Tian and Dan Zhang. A new family of generalized parallel manipulators with configurable moving platforms. *Mechanism and Machine Theory*, 153:103997, 2020.

Chunxu Tian, Yuefa Fang, and Qiaode Jeffrey Ge. Structural synthesis of parallel manipulators with coupling sub-chains. *Mechanism and Machine Theory*, 118:84–99, 2017. doi: 10.1016/j.mechmachtheory.2017.07.014.

Chunxu Tian, Dan Zhang, Hongyan Tang, and Chenwei Wu. Structure synthesis of reconfigurable generalized parallel mechanisms with configurable platforms. *Mechanism and Machine Theory*, 160:104281, 2021. doi: 10.1016/j.mechmachtheory.2021.104281.

Chunxu Tian, Zhihao Xia, Luquan Li, and Dan Zhang. The novel synthesis of reconfigurable generalized parallel manipulators with kinematic redundancy. *Mechanism and Machine Theory*, 201: 105748, 2024. doi: 10.1016/j.mechmachtheory.2024.105748.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. pp. 13484–13508, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843, 2024.

Zhihao Xia, Chunxu Tian, Yongheng Xing, Jie Gu, Luquan Li, and Dan Zhang. The novel synthesis method of decoupled generalized parallel mechanisms based on motion transmission. *Mechanism and Machine Theory*, 210:106019, 2025a. doi: 10.1016/j.mechmachtheory.2025.106019.

Zhihao Xia, Yongheng Xing, Chunxu Tian, and Dan Zhang. Advances in generalized parallel mechanisms: Coupling, configurable, and articulated designs. *Mechanism and Machine Theory*, 218: 106287, 2025b. doi: 10.1016/j.mechmachtheory.2025.106287.

Zhihao Xia, Xinhua Lao, Chunxu Tian, and Dan Zhang. Mapping mechanical synthesis to a trainable task: A neural network approach for generalized parallel mechanisms. *Mechanism and Machine Theory*, 219:106337, 2026. doi: 10.1016/j.mechmachtheory.2025.106337.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.