EngTrace: A Symbolic Benchmark for Verifiable Process Supervision of Engineering Reasoning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are increasingly entering specialized, safety-critical engineering workflows governed by strict quantitative standards and immutable physical laws, making rigorous evaluation of their reasoning capabilities imperative. However, existing benchmarks such as MMLU, MATH, and HumanEval assess isolated cognitive skills, failing to capture the physically grounded reasoning central to engineering, where scientific principles, quantitative modeling, and practical constraints must converge. To enable verifiable process supervision in engineering, we introduce EngTrace, a symbolic benchmark built on 90 parameterized templates, each generating unique, contamination-resistant problem instances, spanning three major engineering branches, nine core domains, and 20 distinct areas, yielding 1,350 test cases that stress-test generalization across diverse physical scenarios. Moving beyond outcome matching, we introduce a verifiable two-stage evaluation framework that uses a tiered protocol to validate intermediate reasoning traces alongside final answers through automated procedural checks and a heterogeneous AI Tribunal. Our evaluation of 27 leading LLMs reveals a distinct trade-off between numeric precision and trace fidelity, identifying a complexity cliff where abstract mathematical pre-training fails to translate into the integrative reasoning required for advanced engineering tasks. The code and data are uploaded to **Software** and **Data** for review.

1 Introduction

As LLMs expand into high-stakes engineering workflows, rigorous evaluation of their reasoning capabilities has become paramount (Zhao et al., 2023; Xie et al., 2023a). Yet progress remains gated by benchmark quality: without tests that enforce physically grounded reasoning, distinguishing genuine capability from sophisticated mimicry remains a challenge.

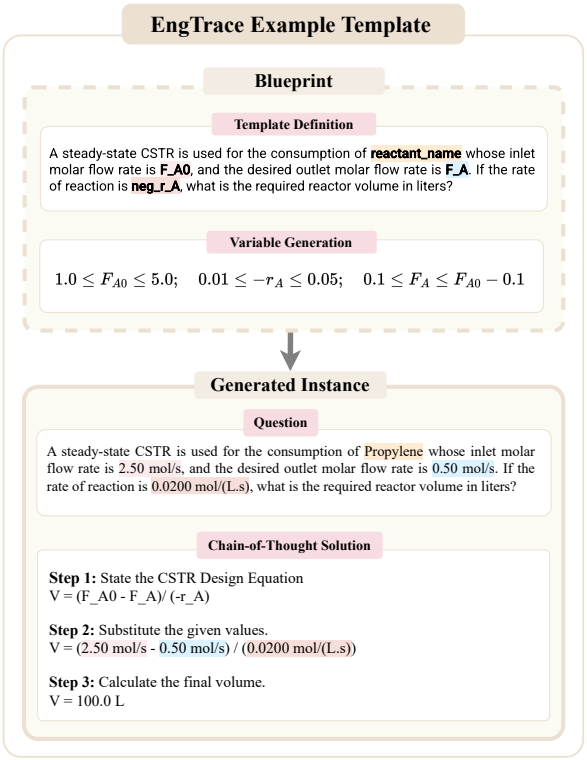

Figure 1: An example from EngTrace demonstrating **CSTR Volume Calculation**. A symbolic template generates a unique problem instance paired with a gold-standard reasoning trace to enable verifiable process supervision.

Engineering reasoning requires physically grounded, multi-step problem solving where intermediate steps are as consequential as the final result, yet no existing benchmark verifies this process. We introduce EngTrace: a symbolic benchmark pairing each problem instance with a gold-standard reasoning trace to enable verifiable process supervision. Symbolic templates (Mirzadeh et al., 2024; Xie et al., 2025) serve as the generative mechanism, producing unique, contamination-resistant instances; Figure 1 illustrates one such instance, a CSTR volume calculation requiring synthesis of interdependent stoichiometry and reaction kinetics

variables into a fully verifiable reasoning trace.

EngTrace addresses two critical gaps in current LLM evaluation for engineering: benchmark saturation and disciplinary fragmentation. First, static benchmarks like Glue (Wang et al., 2019) and BBH (Kazemi et al., 2025) are increasingly prone to model saturation (Ott et al., 2022) and data contamination (Deng et al., 2024); EngTrace counters this through symbolic templates and domain-aware parameterization, generating unique, physically grounded problems that resist rote memorization. Second, most benchmarks evaluate skills in disciplinary silos: general knowledge benchmarks (e.g., MMLU (Hendrycks et al., 2020)) test broad factual recall, while specialized benchmarks (e.g., Math (Hendrycks et al., 2021b) or HumanEval (Chen et al., 2021)) assess abstract logic and algorithmic translation. This fragmentation is ill-suited to engineering, which is fundamentally integrative, requiring the synthesis of scientific principles, mathematical modeling, and practical constraints (e.g., Moaveni, 2019; Dym et al., 2012). EngTrace addresses both gaps through integrative reasoning templates that, unlike existing benchmarks (e.g., Zhou et al., 2025; Mudur et al., 2025) that rely solely on outcome matching, demand holistic procedural reasoning, a crucial requirement in engineering, where a flawed process can lead to catastrophic failure (Leveson, 2012). Importantly, EngTrace is scoped as a depth-oriented evaluation of verifiable process supervision across a principled engineering subset rather than full disciplinary coverage or a production agent workflow; this scope is a deliberate design choice, as verifiability requires gold-standard reasoning traces that can only be guaranteed through controlled symbolic generation.

In summary, our contributions are threefold:

1. We introduce EngTrace, a symbolic benchmark for verifiable process supervision of engineering reasoning, comprising 90 parameterized templates that generate 1,350 unique, contamination-resistant test cases across three major engineering branches, nine core domains, and 20 distinct areas.

2. We design and validate a two-stage evaluation framework combining automated symbolic verification with a heterogeneous AI Tribunal to assess reasoning process quality beyond final answer accuracy, achieving stronger expert alignment ($\rho = 0.632$) than standard NLP metrics ($\rho = 0.325$).

3. We evaluate 27 LLMs across frontier, open-weights, and math-enhanced categories, and introduce a six-category error taxonomy applied to 2,200 annotated failure traces, revealing qualitatively distinct failure modes across model classes and a *complexity cliff* that abstract mathematical pre-training cannot bridge.

2 Related Work

Benchmarks in Mathematics and Coding. Extensive work has established baselines for isolated cognitive skills. In mathematics, benchmarks ranging from GSM8K (Cobbe et al., 2021) and Math (Hendrycks et al., 2021b) to HardMath (Fan et al., 2024), Putnam-Axiom (Gulati et al., 2024), and GSM-Symbolic (Mirzadeh et al., 2024) rigorously test abstract logical deduction and reasoning stability. Similarly, coding evaluations span from function-level generation in HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) to real-world software engineering in SWE-Bench (Jimenez et al., 2024). Yet neither abstract logical deduction nor algorithmic proficiency transfers to reliable reasoning under physical constraints (Heesch et al., 2025).

Benchmarks in the Physical Sciences. To bridge the gap between abstract mathematics and physical reality, recent benchmarks have pivoted to the physical sciences. While UGPhysics (Xu et al., 2025) and PhysReason (Zhang et al., 2025a) effectively test the application of reasoning to natural laws, others target advanced symbolic (Zhang et al., 2025b; Qiu et al., 2025) or qualitative (Wang et al., 2023; Shojaee et al., 2025) understanding. Despite this progress, these benchmarks remain confined to theoretical physics, lacking the integrative context of engineering.

Benchmarks in Engineering. This fragmentation extends to engineering evaluation. Generalist suites like MMLU (Hendrycks et al., 2021a), BIG-Bench (Luo et al., 2024), and SuperGPQA (Du et al., 2025) restrict assessment to factual recall via multiple-choice questions, while specialized benchmarks like TransportBench, APBench, and EEE-Bench (Syed et al., 2024; Chen et al., 2025; Cheng et al., 2024; Li et al., 2025; Liu et al., 2025) are siloed within narrow sub-disciplines. Others focus on tangential skills such as design generation (Guo et al., 2025b) or software proficiency (Mudur et al., 2025), while broader efforts like EngiBench (Zhou

et al., 2025) rely on subjective rubric-based scoring.

Evidently, existing benchmarks remain limited to outcome matching and fail to validate the physically grounded reasoning required for engineering. **EngTrace addresses this limitation by introducing a symbolic benchmark that pairs unique problem instances with gold-standard reasoning traces to enable verifiable process supervision.**

3 EngTrace Design and Methodology

3.1 Taxonomy and Content Selection

The taxonomy of EngTrace, illustrated in Figure 2, follows a principled, two-pronged methodology to ensure it mirrors formal engineering curricula.

Domain and Area Selection. To identify core domains and critical areas, we synthesized engineering curricular standards from ABET (ABET, 2024) and major professional societies (ASME (ASME, 2025), IEEE (IEEE, 2025), AIChE (AIChE, 2025)). We validated these foundational units by cross-referencing them with an expert-persona LLM to ensure alignment between traditional pedagogy and modern AI capabilities (prompts provided in Appendix A and Appendix B).

Template Selection. The 90 symbolic Python templates were fully authored by domain experts, grounded directly in foundational textbooks and authoritative engineering handbooks (e.g., Fogler, 2020; Ulaby and Ravaioli, 2015) (see Appendix C for the complete list of source texts). The template distribution per area is non-uniform and guided by a **Pedagogical Significance Score** (1–5) derived through an expert-persona LLM (detailed prompt provided in Appendix D). This weighting ensures hierarchical fidelity: for instance, within the Reaction Kinetics domain, the core area of Mole Balances (Score: 5) includes five templates, while the specialized Levenspiel Plot Interpretation (Score: 1) is represented by a single template.

3.2 Template Generation Pipeline

The core of EngTrace is the symbolic template, a Python blueprint for generating unique, physically grounded engineering problems. As shown in Figure 3, the pipeline first samples domain-aware parameters from authoritative data sources (Appendix E) and fundamental principles (see detailed examples in Appendix F). It then performs core computations with multi-layer sanity checks to derive computationally verified ground-truth traces. These are serialized into natural language `(question, solution)` pairs, creating a dynamic problem space that effectively resists data contamination. Critically, templates go beyond variable substitution: domain-aware parameter sampling can alter the physical regime, governing equation, and required reasoning path entirely (see Appendix G), though linguistic diversity through paraphrased or naturally sourced wording remains future work.

3.3 Template Validation

To establish EngTrace as a high-fidelity "Gold Standard" benchmark, we utilize a robust two-stage validation process that pairs automated heuristic auditing with expert manual verification.

Stage 1: Automated AI Pre-Screening. Building upon the "Panel of LLM" (PoLL) methodology (Verga et al., 2024), we construct an "AI Tribunal" comprising three frontier reasoning models: `GPT-5` (OpenAI, 2025a), `Claude Opus 4.5` (Anthropic, 2025b), and `Gemini 3` (Google, 2025). Each model acts as an independent judge, utilizing the specific prompt provided in Appendix H, to evaluate the source code and generated instances against the strict multi-axis rubric detailed in Appendix I.

To mitigate potential hallucinations and ensure output reliability, we apply a hybrid aggregation strategy requiring each template T to satisfy three consensus conditions. First, the **Median Quality Score** (S_{med}) across the tribunal must reach a threshold $\theta = 4$ for every evaluative dimension $d \in D$:

$$\forall d \in D, \quad S_{med}^{(d)} = \text{median}(S_{m_1}^{(d)}, S_{m_2}^{(d)}, S_{m_3}^{(d)}) \geq 4 \tag{1}$$

Second, we enforce a **Majority Vote** safety check, discarding any template where more than one judge identifies a violation ($\sum F_i \geq 2$, where $F \in \{0, 1\}$).

Finally, we quantify uncertainty using a **Disagreement Score** (σ_{max}), defined as the maximum standard deviation across all dimensions. Templates exhibiting high variance are flagged for manual review to resolve potential logical ambiguities:

$$\sigma_{max} = \max_{d \in D} \left(\sqrt{\frac{1}{3} \sum_{i=1}^{3} (S_{m_i}^{(d)} - \mu^{(d)})^2} \right) \leq 0.5 \tag{2}$$

where $\mu^{(d)}$ represents the mean score across judges for dimension d.

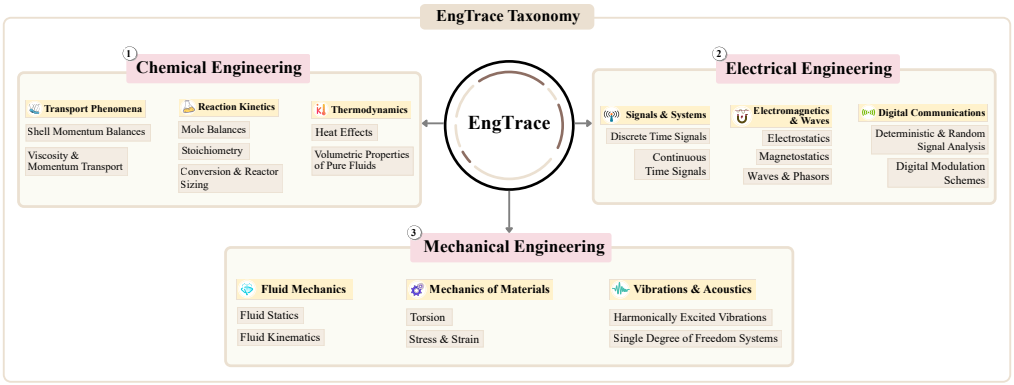

Figure 2: EngTrace's **Hierarchical Taxonomy** spanning nine domains across three branches: process equilibrium and dynamics (Chemical), field theory and signal transmission (Electrical), and physical mechanics and material properties (Mechanical).

Stage 2: Expert Human Certification. Templates passing the AI Tribunal undergo a double-blind review by domain experts through a custom Streamlit interface presented in Appendix J. We assess Inter-Annotator Agreement per branch using Fleiss' κ for binary admissibility (Approve/Reject) and two complementary ordinal coefficients: Krippendorff's α and Gwet's AC2 (Gwet, 2008; Wongpakaran et al., 2013), the latter serving as the primary reliability metric due to its robustness to range restriction (Feinstein and Cicchetti, 1990).

AI-Human Alignment. To assess validation reliability, we measure correspondence between the AI Tribunal and human experts using False Positive Rate (FPR) to quantify filter safety, and Mean Absolute Difference (MAD) to evaluate scoring consistency in original scale units. MAD is preferred over rank correlation, which becomes uninformative under the score concentration produced by effective pre-filtering. Detailed results are provided in Appendix K.

3.4 Difficulty Scaling

Problem difficulty in EngTrace is calibrated through a multi-axis framework defined by three dimensions: **Conceptual Complexity** (isolated principles vs. multi-domain synthesis), **Mathematical Sophistication** (direct algebraic substitution vs. differential equations), and **Procedural Depth** (length of interdependent reasoning chains). A detailed quantitative breakdown of the dataset's difficulty distribution is provided in Appendix L.

4 Evaluation Framework

Inspired by recent advances in process supervision (Lightman et al., 2023) and cascaded model verification (Chen et al., 2023), we introduce a two-stage evaluation framework that validates intermediate reasoning steps in addition to final-answer accuracy through a tiered verification protocol. To ensure alignment with professional standards, we validated this protocol through a blind human-expert study, confirming that our Tribunal-based approach significantly outperforms standard metrics in capturing engineering reasoning quality (see Appendix M for full methodology and ablation results).

4.1 Preliminaries

We define the evaluation of a reasoning chain as a comparison between the ground truth solution G and the model-generated predicted solution P. Both are represented as ordered sequences of M and N discrete reasoning steps, respectively:

$$G = (g_1, g_2, \ldots, g_M), \quad P = (p_1, p_2, \ldots, p_N) \tag{3}$$

Each individual reasoning step $s_i \in \{G, P\}$ consists of a textual derivation t_i and, where applicable, an associated numerical or symbolic intermediate result $a_i = \text{Res}(s_i)$.

To assess the validity of the derivation, we seek a mapping that aligns steps in P to their logical counterparts in G. Unlike standard string matching, this alignment must account for structural variations such as step merging, reordering, or alternative valid methodologies while ensuring that the cumulative logic remains faithful to the underlying engineering principles.

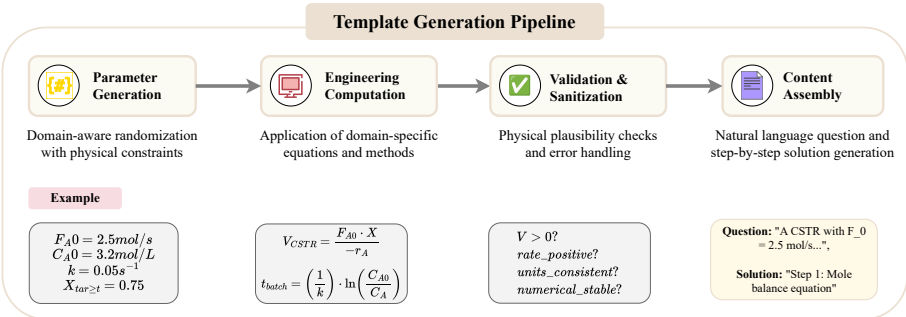

Figure 3: The EngTrace **Template Generation Pipeline** maps domain-aware constraints to serialized question-solution pairs via physically-validated computation, illustrated here using a CSTR volume calculation example.

4.2 Final Answer Accuracy

To evaluate the correctness of the final predicted outcome, let g_M and p_N denote the last steps of the ground truth and predicted solutions, from which we extract the terminal numerical results $A_G = \text{Res}(g_M)$ and $A_P = \text{Res}(p_N)$. A response is deemed correct if the relative error between these values satisfies the following condition:

$$\mathbb{I}_{final} = \mathbb{I}\left(\frac{|A_G - A_P|}{|A_G|} < \epsilon_{final}\right) \tag{4}$$

where $\epsilon_{final} = 0.02$ (2%), a strict threshold required to prevent catastrophic error propagation in high-precision engineering derivations.

4.3 Textual Quality

We evaluate the fluency and semantic fidelity of generated solutions using standard reference-based metrics (Xie et al., 2023b). We employ **BERTScore** (Zhang et al., 2020) to measure deep semantic similarity via contextual embedding alignment, and **ROUGE-2** (Lin, 2004) to quantify lexical overlap. Together, these metrics assess the model's ability to employ appropriate domain-specific terminology.

4.4 Tiered Reasoning Verification

To evaluate the full reasoning chain, we characterize the validity of the predicted sequence P against the ground truth G through a tiered verification protocol. This process consists of two phases: determining step-level validity through hierarchical assessment and aggregating these results into a global reasoning metric.

Step-Level Validity Assessment. We compute a validity matrix $V \in \mathbb{R}^{M \times N}$, where each element $V_{i,j}$ represents the validity of predicted step p_j relative to ground truth step g_i. This validity is determined by two successive tiers of verification.

Tier 1: Automated Symbolic Verification. This tier fills the matrix V by enforcing syntactic and numerical precision through a binary step-validity score:

$$S_{\mathcal{T}_1}(g_i, p_j) = \delta_{num} \wedge \delta_{sem} \tag{5}$$

where $S_{\mathcal{T}_1} = 1$ only if the numerical consistency (relative error tolerance $\epsilon = 0.02$) and semantic similarity (Cross-Encoder threshold $\tau = 0.7$) are both satisfied. To optimize computational efficiency, we calculate the alignment ratio, which represents the proportion of steps in P successfully verified by Tier 1. If this ratio exceeds a confidence threshold ($\rho = 0.8$) and the final answer is correct, the reasoning chain is classified as a standard derivation and accepted without further review. Chains falling below this threshold are escalated to Tier 2 for heuristic assessment.

Tier 2: Multi-Model Heuristic Verification. Escalated chains are evaluated by the same heterogeneous, cross-organization AI Tribunal (§ 3.3) to identify valid alternative derivations (prompt provided in Appendix N). Each judge J_k classifies the observed discrepancy between p_j and g_i into a category C_k, which is then mapped to a scalar score:

$$\mu(C) = \begin{cases} 1.0 & \text{if } C = \text{Alternate Correct} \\ 0.5 & \text{if } C = \text{Calculation Error} \\ 0.0 & \text{if } C \in \{\text{Conceptual Error, Other}\} \end{cases} \tag{6}$$

The majority consensus of these scalar scores defines $S_{\mathcal{T}_2}$, which is assigned to $V_{i,j}$ to finalize the validity matrix for non-standard derivations. In the absence of a clear majority, the assessment defaults

to a conservative tie-breaker where the lowest score is assigned.

Global Reasoning Metric $\mathbf{F1_{rec}}$. To quantify the overall quality of the reasoning chain, we calculate the Recovered Reasoning F1 Score ($F1_{rec}$). We first identify the optimal alignment that maximizes the aggregate verification score:

$$\mathcal{R}_{total} = \max_{\sigma \in \mathcal{A}} \sum_{(i,j) \in \sigma} V_{i,j} \tag{7}$$

where $\mathcal{A}$ represents the set of valid one-to-one mappings between steps. This optimization is solved using the Hungarian matching algorithm on the finalized validity matrix V. We define recovered precision and recall as $P_{rec} = \mathcal{R}_{total}/N$ and $R_{rec} = \mathcal{R}_{total}/M$, respectively, where N and M are the number of predicted and ground truth steps. Finally, $F1_{rec}$ is computed as the harmonic mean of P_{rec} and R_{rec}.

5 Experiments

5.1 Evaluation Model Suite

We evaluate 27 models categorized into a three-way taxonomy to isolate the impacts of scale, access, and specialization: (1) **Frontier proprietary models** represent the current state-of-the-art in reasoning and instruction following. Our evaluation includes 14 models in this category: `GPT-5 {Base, Mini}` and `GPT-4.1 {Base, Mini}` (OpenAI, 2025a,b), `Claude Sonnet {4.5, 4, 3.7}` and `Claude Opus 4.7` (Anthropic, 2025c,d,a, 2026), `Gemini {3.1 Pro, 3 Pro, 2.5 Pro, 2.5 Flash}` (Google DeepMind, 2026; Google, 2025; Comanici et al., 2025), and `DeepSeek {V3, R1}` (Liu et al., 2024; Guo et al., 2025a). (2) **General-purpose open models** provide strong, domain-agnostic foundations and allow for transparent analysis of performance relative to parameter count. We evaluate 9 models: `Meta-Llama-3.1 {8B, 70B}` (Grattafiori et al., 2024), `Qwen2.5 {7B, 14B, 72B}` and `Qwen3-8B` (Qwen, 2024, 2025), `Gemma {2-9B, 3-27B}` (Gemma Team et al., 2024; Gemma Team and Google DeepMind, 2025), and `DeepSeek-V4-Pro` (DeepSeek-AI, 2026). (3) **Math-enhanced models** are systems specifically trained on mathematical and symbolic corpora to enhance quantitative reasoning. We evaluate 4 specialized models in the 7B parameter class: `Qwen2.5-Math-7B` (Yang et al., 2024), `Mathstral-7B` (Mistral, 2024), `WizardMath-7B` (Luo et al., 2023), and `MetaMath-7B` (Yu et al., 2023). Detailed configurations and model sources can be found in Appendix O.

5.2 Experimental Setup

We instantiate the benchmark by sampling 15 instances per symbolic template, each with a distinct random seed, creating a total of 1,350 test cases (90 templates $\times$ 15 instances) for each model. We evaluate all models under a unified decoding configuration: a low `temperature` of 0.2 to ensure deterministic outputs and a maximum token limit of 4,096 to allow for detailed reasoning. All models are evaluated in a zero-shot setting using a standardized prompt and output parsing pipeline; full implementation specifications are provided in Appendix P. All models are evaluated in a closed-book, tool-free setting to isolate intrinsic reasoning ability; tool-augmented and retrieval-augmented evaluation are left as a separate future setting.

5.3 Results

5.3.1 Overall Model Performance

Table 1 summarizes performance across Final Answer Accuracy (FAC), Reasoning F1, and semantic similarity (BERTScore, ROUGE-2/L). Frontier models lead overall, with Gemini 3.1 Pro topping FAC (65.41%) and DeepSeek R1 Reasoning F1 (44.41%). Open-weights models lag substantially on both metrics despite competitive semantic scores, and math-enhanced 7B models underperform general-purpose counterparts of equivalent size, suggesting that abstract mathematical pretraining does not transfer to physically constrained engineering reasoning.

5.3.2 Performance by Engineering Branch

Branch difficulty as illustrated in Figure 7 follows a consistent structural ordering: Chemical Engineering is hardest and Mechanical the most accessible, a pattern that persists across model families and scales. In Chemical Engineering, even leading models such as Gemini 3.1 Pro (54.22%) and DeepSeek R1 (64.89%) fall well below their scores on other branches, suggesting stoichiometric multi-step reasoning poses a qualitatively harder challenge rather than a scaling artifact. Electrical Engineering exposes the starkest decoupling between numerical precision and process fidelity: Gemini 3.1 Pro leads FAC (70.67%) yet DeepSeek R1 retains the highest Reasoning F1 (41.22%), indicating models can

Model	Final Answer ↑	Reasoning (F1) ↑	BERTScore ↑	ROUGE-2 ↑	ROUGE-L ↑
Frontier Proprietary LLMs					
Claude 3.7 Sonnet	57.63	39.36	88.09	29.42	37.36
Claude 4 Sonnet	56.81	36.80	87.92	29.52	37.22
Claude 4.5 Sonnet	51.78	33.61	87.37	28.06	35.39
Claude Opus 4.7	63.56	39.45	87.96	29.28	37.29
DeepSeek V3	62.37	37.06	86.54	25.69	34.57
DeepSeek R1	61.04	**44.41**	86.48	25.58	34.49
DeepSeek V4 Pro	61.11	40.18	86.54	26.16	34.45
Gemini 2.5 Flash	59.78	42.22	87.64	29.14	36.70
Gemini 2.5 Pro	60.44	39.38	88.24	**31.88**	**38.07**
Gemini 3 Pro	64.00	37.05	86.71	27.32	34.99
Gemini 3.1 Pro	**65.41**	42.71	87.12	28.56	35.68
GPT-4.1	57.11	31.22	86.63	26.78	34.46
GPT-4.1 Mini	61.56	36.87	86.50	25.06	33.87
GPT-5	61.04	41.81	88.19	28.98	36.55
GPT-5 Mini	63.70	41.01	87.95	26.63	34.74
General Purpose Open LLMs					
Gemma 2 9B	26.59	21.37	88.19	28.02	35.77
Gemma 3 27B	44.30	27.59	87.04	23.93	32.66
Llama 3.1 8B	23.70	18.61	87.65	25.59	32.50
Llama 3.1 70B	40.81	30.04	**88.34**	28.87	35.82
Qwen 2.5 7B	35.63	23.67	86.30	24.21	32.89
Qwen 2.5 14B	42.07	27.67	86.44	24.94	33.38
Qwen 2.5 72B	48.07	28.02	86.46	25.58	33.96
Qwen 3 8B	23.70	16.68	86.91	8.93	11.09
Math Enhanced LLMs					
Mathstral 7B	19.70	18.59	87.79	28.02	36.42
MetaMath 7B	6.07	2.37	82.97	10.57	16.69
Qwen 2.5 Math 7B	23.04	18.18	85.61	21.71	29.98
WizardMath 7B	8.22	6.38	83.66	14.14	20.38

Table 1: **Overall Performance Comparison** of 27 models on the EngTrace benchmark. Scores are reported as percentages. The Final Answer column denotes strict numerical accuracy, while Reasoning (F1) measures the structural alignment of the derivation steps. The highest score in each column is highlighted in **bold**.

arrive at correct answers through derivations that diverge substantially from expert solutions. Mechanical Engineering's relative accessibility (Gemini 3.1 Pro FAC 71.33%) reflects its reliance on single-formula substitution patterns that are well-represented in pre-training corpora.

5.3.3 Performance by Engineering Domain

Domain-level results in Figure 8 reveal a spiky capability profile: models that excel in one subdomain often underperform in a closely related one, indicating specialization rather than generalized engineering reasoning. Within Chemical Engineering, Transport Phenomena is a persistent floor (max FAC 26.67%) while Thermodynamics is comparatively tractable (DeepSeek R1: 78.33%), exposing coupled transport equations as a fundamentally harder reasoning class. In Electrical Engineering, Digital Communications is relatively accessible (Gemini 3.1 Pro: 73.33%) due to its discrete logic-based structure, whereas Electromagnetics remains resistant across all models (max FAC 60.00%). Mechanical Engineering shows the sharpest within-branch contrast: Mechanics of Materials is nearly solved by frontier models (Gemini 3.1 Pro: 94.00%), yet Fluid Mechanics remains a ceiling (max FAC 54.00%), confirming that static equilibrium problems are fundamentally more tractable than non-linear fluid dynamics.

5.3.4 Performance by Difficulty Level

As shown in Figure 4, on Easy problems, efficiency-oriented models such as GPT-5 Mini (67.84%) remain competitive with frontier models, suggesting flagship compute is unnecessary for fundamental engineering reasoning. The Advanced tier exposes a sharp divergence: open-weights models collapse dramatically (Llama 3.1 70B: 23.33%) while frontier models sustain meaningful performance (Gemini 3.1 Pro: 49.70%), establishing difficulty as the primary axis separating viable from non-viable model classes. Math-enhanced models fail to close

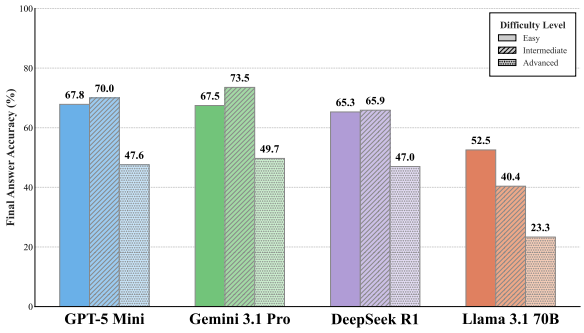

Figure 4: **Difficulty-Level Performance** across efficiency-optimized, frontier, and open-weights models. The plot reveals a "complexity cliff": open-weights models degrade sharply at advanced tasks, whereas frontier models remain stable.

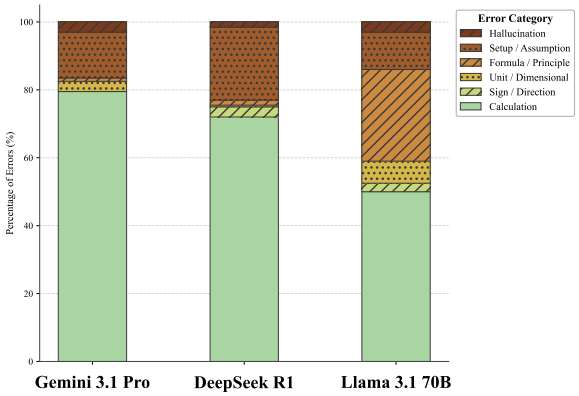

Figure 5: **Error category distributions for the three representative models.** Gemini 3.1 Pro anchors the FAC-leader position, DeepSeek R1 the F1-leader position, and Llama 3.1 70B the steepest complexity cliff.

this gap at any tier, peaking at just 32.35% on Easy tasks, confirming that abstract mathematical pre-training does not transfer to physically grounded reasoning under complexity pressure.

5.4 Error Analysis

Unlike the coarse three-way classification used by the AI Tribunal during evaluation (§ 4), which is designed solely to determine step-level validity scores, this analysis employs a finer six-category taxonomy applied by human domain experts to diagnose the root causes of confirmed reasoning failures independently of the automated evaluation pipeline. We manually inspect a 200-trace sample per model drawn from confirmed reasoning failures, stratified by difficulty tier, and assign each trace one of these six categories under a causally-ordered taxonomy; the taxonomy, sampling procedure, and inter-annotator agreement (Fleiss' $\kappa = 0.524$, Gwet's AC1 $= 0.682$) are detailed in Appendices R, S,and T. We focus on three models occupying contrasting positions: Gemini 3.1 Pro (FAC leader), DeepSeek R1 (F1 leader), and Llama 3.1 70B (steepest complexity cliff); the remaining models are discussed in Appendix U.

As shown in Figure 5, Gemini 3.1 Pro's failures are overwhelmingly arithmetic (79.5% Calculation Errors, 14.5% conceptual). The model consistently identifies the correct governing equation and frames the problem correctly, suggesting that arithmetic execution rather than physical reasoning is the primary bottleneck at the frontier. DeepSeek R1 shows a more distributed profile: Calculation Errors drop to 72.0% while Setup/Assumption errors rise to 21.5%, nearly double Gemini's rate, directly explaining the FAC-versus-F1 tension from § 5.3: step-level reasoning is frequently sound, but systematic problem-framing errors propagate to wrong final answers. Crucially, this elevated conceptual rate does not worsen with difficulty (cliff $= -4.1$ pp), pointing to a stable physical modelling weakness rather than a capacity limit. Llama 3.1 70B tells a different story: Formula/Principle errors surge from 14.5% at Easy to 49.0% at Advanced while Calculation Errors fall from 60.5% to 30.6%, a 41.8 pp conceptual cliff versus only 7.9 pp for Gemini, revealing a genuine breakdown in domain knowledge at higher complexity rather than deteriorating arithmetic.

6 Conclusion and Future Work

We introduced EngTrace, a symbolic benchmark for verifiable process supervision of physically grounded engineering reasoning. Our evaluation of 27 LLMs reveals a "complexity cliff" where open-weights models collapse on advanced tasks while frontier models remain stable, and exposes qualitatively distinct failure modes: frontier models fail primarily on arithmetic execution while open-weights models break down conceptually, confirming that trustworthy engineering AI requires more than abstract mathematical pre-training. Future work will explore tool-augmented reasoning to decouple physical reasoning from arithmetic execution, and uncertainty quantification to assess whether models can reliably calibrate confidence in safety-critical engineering scenarios.

Limitations

First, as stated in § 5.2, this work intentionally evaluates models in a closed-book, tool-free setting to isolate intrinsic reasoning ability. This design

choice, tied to verifiability, means we overlook iterative self-correction, risk assessment, and the stability assurance required for high-stakes deployment. Tool-augmented and retrieval-augmented reasoning are committed as a separate future setting. Second, EngTrace is entirely synthetic. While this ensures verifiability and contamination resistance, symbolic templates produce linguistically constrained problem statements that do not capture the full diversity or contextual ambiguity of real-world engineering. More complex, ill-defined problems often require interpreting non-textual information such as technical diagrams, schematics, or CAD models. Future versions will address this by incorporating multi-modal tasks and using real-world engineering artifacts as seeds for semi-structured problem generation. Third, as noted in **??**, the benchmark is scoped to three core engineering branches as a depth-oriented design choice tied to verifiability rather than full disciplinary coverage. Expanding the EngTrace taxonomy in collaboration with domain experts is a clear next step toward more comprehensive evaluation.

Ethical Statement and Broad Impact

This work utilizes synthetic data generated through symbolic templates and standard engineering principles; no private, sensitive, or proprietary industrial data was used. While EngTrace aims to advance reliable engineering reasoning, we emphasize that current LLMs should not be deployed in safety-critical physical systems without rigorous, hardware-in-the-loop verification. We hope this benchmark fosters the development of interpretable, verifiable, and physically grounded AI for high-stakes engineering applications.

Data License The EngTrace dataset and associated code will be released under the MIT License.

References

ABET. 2024. Criteria for accrediting engineering programs. https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2024-2025/.

AIChE. 2025. Constitution & bylaws. https://www.aiche.org/about/governance/constitution-bylaws.

Anthropic. 2025a. Claude 3.7 sonnet system card. https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf.

Anthropic. 2025b. Claude Opus 4.5 System Card. https://assets.anthropic.com/m/64823ba7485345a7/Claude-Opus-4-5-System-Card.pdf. Released November 24, 2025.

Anthropic. 2025c. Claude sonnet 4.5 system card. https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf.

Anthropic. 2025d. System card: Claude opus 4 & claude sonnet 4. https://www-cdn.anthropic.com/6be99a52cb68eb70eb9572b4cafad13df32ed995.pdf.

Anthropic. 2026. Claude Opus 4.7 model card. Technical report, Anthropic. Model identifier: claude-opus-4-7. Available via Anthropic API, Amazon Bedrock, Google Cloud Vertex AI, and Microsoft Foundry.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

ASME. 2025. Vision, mission, & core values. https://www.asme.org/about-asme/vision-mission-core-values.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Ted Byrt, Janet Bishop, and John B. Carlin. 1993. Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46(5):423–429.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to use large language models while reducing cost and improving performance. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yongchao Chen, Enrico Zucchelli, Daniel Jang, Di Wu, Raymond Zhang, Giovanni Lavezzi, and Richard Linares. 2025. Apbench and benchmarking large language model performance in fundamental astrodynamics problems for space engineering. *Scientific Reports*, 15.

Yuheng Cheng, Huan Zhao, Xiyuan Zhou, Junhua Zhao, Yuji Cao, Chao Yang, and Xinlei Cai. 2024. Elecbench: A power dispatch evaluation benchmark for large language models. *ArXiv preprint*, abs/2407.05365.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

DeepSeek-AI. 2026. DeepSeek-V4: Towards highly efficient million-token context intelligence. Technical report, DeepSeek. Model identifier: `deepseek-ai/DeepSeek-V4-Pro`. 1.6T-parameter MoE model released under the MIT License.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, et al. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *ArXiv preprint*, abs/2502.14739.

Clive L. Dym, Patrick Little, and Elizabeth J. Orwin. 2012. *Engineering Design: A Project-Based Introduction*, 4th edition. John Wiley & Sons.

J. Fan, S. Martinson, E. Y. Wang, K. Hausknecht, J. Brenner, D. Liu, N. Peng, C. Wang, and M. P. Brenner. 2024. HARDMath: A benchmark dataset for challenging problems in applied mathematics. *arXiv preprint arXiv:2410.09988*.

Alvan R. Feinstein and Domenic V. Cicchetti. 1990. High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549.

H. Scott Fogler. 2020. *Elements of Chemical Reaction Engineering*, 6th edition. Pearson, Boston, MA.

Gemma Team and Google DeepMind. 2025. Gemma 3: Technical report. *arXiv preprint*, abs/2503.19786.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint*, abs/2408.00118.

Google. 2025. Gemini 3: A new era of intelligence. `https://blog.google/products/gemini/gemini-3/`. Released November 18, 2025. Model documentation available at `https://ai.google.dev/gemini-api/docs/gemini-3`.

Google DeepMind. 2026. Gemini 3.1 Pro: A more capable reasoning model in the Gemini 3 series. Technical report, Google DeepMind. Model identifier: `gemini-3.1-pro-preview`. Available via Google AI Studio, Vertex AI, and Gemini API.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. 2024. Putnam-axiom: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xingang Guo, Yaxin Li, Xiangyi Kong, Yilan Jiang, Xiayu Zhao, et al. 2025b. Toward engineering agi: Benchmarking the engineering design capabilities of llms. *ArXiv preprint*, abs/2509.16204.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Rene Heesch, Sebastian Eilermann, Alexander Windmann, Alexander Diedrich, Philipp Rosenthal, and Oliver Niggemann. 2025. Evaluating large language models for real-world engineering tasks. *arXiv preprint arXiv:2505.13484*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*.

IEEE. 2025. IEEE standards association. `https://standards.ieee.org/`.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shun Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan.

2024. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

Mostafa Kazemi, Behnam Fatemi, Hritik Bansal, John Palowitch, Christos Anastasiou, Shweta V. Mehta, Lovish K. Jain, Valentina Aglietti, D. Jindal, P. Chen, N. Dikkala, G. Tyen, X. Liu, U. Shalit, S. Chiappa, K. Olszewska, Y. Tay, V. Q. Tran, Q. V. Le, and O. Firat. 2025. BIG-Bench Extra Hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26473–26501, Vienna, Austria.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Nancy G. Leveson. 2012. *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, Cambridge, MA.

Ming Li, Jike Zhong, Tianle Chen, Yuxiang Lai, and Konstantinos Psounis. 2025. Eee-bench: A comprehensive multimodal electrical and electronics engineering benchmark. *ArXiv preprint*, abs/2411.01492. Accepted to CVPR 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Lukas Mesnard, Tyna Wang, Farzad Khorrami, Nguyet Minh Nguyen, Shayne Mostyn, Max Miller, Chia Hsuan Wang, Sam Gelman, Denis Igor, Marina Polozov, Girish Sastry, Prachit Tara, Sandhini Agarwal, Nan Rosemary Sun, Shengjia Zhao, Jeffrey Wu, Szymon Sidor, Jiayi Weng, Yuan Cao, Adrià Puigdomènech Badia, Nikolas Tezak, Peter Welinder, Ilya Sutskever, John Schulman, Jan Leike, and Wojciech Zaremba. 2023. Let's verify step by step. *ArXiv preprint*, abs/2305.20050.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Jiaheng Liu et al. 2025. Enhancing large language models for automated homework assessment in undergraduate circuit analysis. *ArXiv preprint*, abs/2502.07980. Introduces the CIRCUIT benchmark.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *ArXiv preprint*, abs/2308.09583.

Haoyang Luo, Hao Huang, Zhaowei Deng, Xiaodong Li, Haoran Wang, Yaxin Jin, Yaling Liu, Weiming Xu, and Zenghao Liu. 2024. BIGbench: A unified benchmark for evaluating multi-dimensional social biases in text-to-image models. *arXiv preprint arXiv:2407.15240*.

Iman Mirzadeh, Kiarash Alizadeh, Hidetoshi Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

AI Mistral. 2024. Mathstral. 7B parameter model for mathematical reasoning, released under Apache 2.0 license.

Saeed Moaveni. 2019. *Engineering Fundamentals: An Introduction to Engineering*, 6th edition. Cengage Learning.

Nayantara Mudur, Hao Cui, Subhashini Venugopalan, Paul Raccuglia, Michael P. Brenner, and Peter Norgaard. 2025. FEABench: Evaluating language models on multiphysics reasoning ability. *arXiv preprint arXiv:2504.06260*.

OpenAI. 2025a. Gpt-5 system card. `https://cdn.openai.com/gpt-5-system-card.pdf`. System card describing GPT-5 variants (thinking/main/mini/-nano).

OpenAI. 2025b. Introducing gpt-4.1 in the api. `https://openai.com/index/gpt-4-1/`. Product research post introducing GPT-4.1, 4.1 mini, and 4.1 nano.

S. Ott, A. Barbosa-Silva, K. Blagec, J. Brauner, and M. Samwald. 2022. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13(1):6793.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Siyu Qiu, Shuo Guo, Ze-Yu Song, et al. 2025. PHYBench: Holistic evaluation of physical perception and reasoning in large language models. *arXiv preprint arXiv:2504.16074*.

Team Qwen. 2024. Qwen2.5 technical report. *ArXiv preprint*, abs/2412.15115.

Team Qwen. 2025. Qwen3.

Payam Shojaee, Nam-Huan Nguyen, Kamyar Meidani, Amir Barati Farimani, Khanh Duy Doan, and Chandan K. Reddy. 2025. LLM-SRBench: A new benchmark for scientific equation discovery with large language models. *arXiv preprint arXiv:2504.10415*.

Usman Syed, Ethan Light, Xingang Guo, Huan Zhang, Lianhui Qin, Yanfeng Ouyang, and Bin Hu. 2024.

Benchmarking the capabilities of large language models in transportation system engineering: Accuracy, consistency, and reasoning behaviors. *ArXiv preprint*, abs/2408.08302.

Fawwaz T. Ulaby and Umberto Ravaioli. 2015. *Fundamentals of Applied Electromagnetics*, 7th edition. Pearson, Upper Saddle River, NJ.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *ArXiv preprint*, abs/2404.18796.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Yi R. Wang, Jia Duan, Dieter Fox, and Siddhartha S. Srinivasa. 2023. NEWTON: Are large language models capable of physical reasoning? *arXiv preprint arXiv:2310.07018*.

Nahathai Wongpakaran, Tinakon Wongpakaran, Danny Wedding, and Kilem Li Gwet. 2013. A comparison of Cohen's Kappa and Gwet's AC1 when calculating inter-rater reliability coefficients: a study conducted with personality disorder samples. *BMC Medical Research Methodology*, 13:61.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2023a. The next chapter: A study of large language models in storytelling. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 323–351, Prague, Czechia. Association for Computational Linguistics.

Zhuohan Xie, Miao Li, Trevor Cohn, and Jey Lau. 2023b. DeltaScore: Fine-grained story evaluation with perturbations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5317–5331, Singapore. Association for Computational Linguistics.

Zhuohan Xie, Daniil Orel, Rushil Thareja, Dhruv Sahnan, Hachem Madmoun, Fan Zhang, Debopriyo Banerjee, Georgi Georgiev, Xueqing Peng, Lingfei Qian, Jimin Huang, Jinyan Su, Aaryamonvikram Singh, Rui Xing, Rania Elbadry, Chen Xu, Haonan Li, Fajri Koto, Ivan Koychev, Tanmoy Chakraborty, Yuxia Wang, Salem Lahlou, Veselin Stoyanov, Sophia Ananiadou, and Preslav Nakov. 2025. FINCHAIN: A symbolic benchmark for verifiable chain-of-thought financial reasoning. *arXiv preprint*.

Xingyu Xu, Qian Xu, Tong Xiao, Tao Chen, Yufan Yan, Jifan Zhang, Shuo Diao, Chen Yang, and Yizhong Wang. 2025. UGPhysics: A comprehensive benchmark for undergraduate physics reasoning with large language models. *arXiv preprint arXiv:2502.00334*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *ArXiv preprint*, abs/2409.12122.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *ArXiv preprint*, abs/2309.12284.

Antonia Zapf, Stefanie Castell, Lars Morawietz, and André Karch. 2016. Measuring inter-rater reliability for nominal data: which coefficients and confidence intervals are appropriate? *BMC Medical Research Methodology*, 16(1):93.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xinyi Zhang, Yuxuan Dong, Yutao Wu, Jiahui Huang, Chen Jia, Basura Fernando, Mike Zheng Shou, Li Zhang, and Jielin Liu. 2025a. PhysReason: A comprehensive benchmark towards physics-based reasoning. *arXiv preprint arXiv:2502.12054*.

Yifan Zhang, Yidong Ma, Yuntian Gu, Ziyi Yang, Yijia Zhuang, Fanglei Wang, Zhaofeng Huang, Yan Wang, Chaohui Huang, Biao Song, Chang Lin, and Jun Zhao. 2025b. ABench-Physics: Benchmarking physical reasoning in LLMs via high-difficulty and dynamic physics problems. *arXiv preprint arXiv:2507.04766*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *ArXiv preprint*, abs/2303.18223.

Xiyuan Zhou, Xinlei Wang, Yirui He, Yang Wu, Ruixi Zou, et al. 2025. Engibench: A benchmark for evaluating large language models on engineering problem solving. *ArXiv preprint*, abs/2509.17677. Distinct from the design-focused EngiBench by Felten et al.

A LLM Prompt for Domain Validation

To identify core domains, we consulted a panel of four frontier models: `GPT-4o`, `Claude 3.5 Sonnet`, `Gemini 2.5 Pro`, and `DeepSeek V3`. Each was prompted individually to identify cornerstone engineering domains. We finalized the nine core domains via majority voting. The domain identification prompt is provided below.

```
You are an expert academic and senior curriculum designer at a top-tier engineering university (like MIT or Caltech). You have decades of experience structuring undergraduate programs for ABET accreditation.

Your task is to identify and rank the most fundamental, cornerstone domains for the three primary branches of engineering. The goal is to identify the non-overlapping, core subject areas that are prerequisites for almost all other advanced topics.

For example, for Chemical Engineering, "Reaction Kinetics" would be a fundamental domain, while a niche topic like "Polymer Rheology" would be a sub-specialization, not a cornerstone.

Please provide a ranked list of the top 5-7 most fundamental domains for each of the following three branches:
1. Chemical Engineering
2. Electrical Engineering
3. Mechanical Engineering

Return your answer only in a strict JSON format as given below.

{
  "chemical_engineering": [
    "Rank 1 Domain",
    "Rank 2 Domain",
    "Rank 3 Domain",
    "Rank 4 Domain",
    "Rank 5 Domain"
  ],

  "electrical_engineering": [
    "Rank 1 Domain",
    "Rank 2 Domain",
    "Rank 3 Domain",
    "Rank 4 Domain",
    "Rank 5 Domain"
  ],

  "mechanical_engineering": [
    "Rank 1 Domain",
    "Rank 2 Domain",
    "Rank 3 Domain",
    "Rank 4 Domain",
    "Rank 5 Domain"
  ]
}

Do not include any preamble, conversational text, explanations, or markdown formatting around the JSON block.
```

B LLM Prompt for Area Validation

Following the domain validation procedure, we utilized the same model panel and aggregated the pedagogical areas via majority voting. The prompt used for area identification is provided below.

```
You are an expert academic and senior curriculum designer at a top-tier engineering university (like MIT or Caltech). You have decades of experience structuring undergraduate programs for ABET accreditation.

Your task is to identify and list the most fundamental, non-overlapping "areas" or "sub-topics" within the provided engineering domain.

A "fundamental area" is a major pedagogical unit or chapter within a core course on this domain. The areas you list should be distinct pillars of the subject. For example, if the domain is "Thermodynamics," a fundamental area would be "The First Law" or "Properties of Pure Fluids." A concept like "Enthalpy" would be too specific. Similarly, within a "Circuit Analysis" domain, "Kirchhoff's Voltage Law" and "Kirchhoff's Current Law" are foundational concepts often taught together and would be too granular to list separately.

Please provide a list of the top 3-5 most fundamental areas for the following engineering domain: [DOMAIN_NAME_HERE]

Return your answer only in a strict JSON format as given below.

{
  "domain": "[DOMAIN_NAME_HERE]",
  "areas": [
    "Area 1",
    "Area 2",
    "Area 3",
    "Area 4"
  ]
}

Do not include any preamble, conversational text, explanations, or markdown formatting around the JSON block.
```

C Foundational Textbooks for Template Selection

The core engineering principles, problem typologies, and realistic constraints for the templates developed by us were sourced from the authoritative

Domain	Textbook
Reaction Kinetics	Elements of Chemical Reaction Engineering by H. Scott Fogler
Thermodynamics	Introduction to Chemical Engineering Thermodynamics by J. M. Smith, H. C. Van Ness, and M. M. Abbott
Transport Phenomena	Transport Phenomena by R. Byron Bird, Warren E. Stewart, and Edwin N. Lightfoot
Electromagnetics and Waves	Fundamentals of Applied Electromagnetics by F. T. Ulaby and Umberto Ravaioli
Communication Systems	Digital Communications by John G. Proakis and Masoud Salehi
Signals and Systems	Discrete-Time Signal Processing by Alan V. Oppenheim and Ronald W. Schafer
Mechanics of Materials	Mechanics of Materials by Ferdinand P. Beer, E. Russell Johnston, Jr., and John T. DeWolf
Fluid Mechanics	Fundamentals of Fluid Mechanics by Bruce R. Munson, Donald F. Young, and Theodore H. Okiishi
Vibrations and Acoustics	Mechanical Vibrations by Singiresu S. Rao

Table 2: Foundational textbooks used for selecting domain-specific problems.

textbooks listed in Table 2. These texts, which are standard in top-tier engineering curricula, served as the primary knowledge base to ensure EngTrace is pedagogically sound and grounded in the core engineering curriculum. They were also used to validate our selection of fundamental domains and areas.

D LLM Prompt for Pedagogical Scoring

The following prompt was used to query LLMs (described in Appendix A) to evaluate the pedagogical significance of each sub-topic within its parent domain.

```
You are an expert academic and senior curriculum designer at a top-tier engineering university with decades of experience structuring undergraduate programs for ABET accreditation.

Your task is to evaluate the pedagogical significance of a specific engineering "area" within a broader "domain" on a scale of 1 to 5, based on the rubric below.

Rubric:

Score 5 (Cornerstone): A foundational, prerequisite topic for nearly all other concepts in the domain.

Score 3 (Core Concept): A standard, important topic that builds upon cornerstone principles but is not necessarily a universal prerequisite.

Score 1 (Specialized Application): A focused application or an integrative topic taught after cornerstones and core concepts are mastered.

Please evaluate the following area:

Domain: [DOMAIN_NAME_HERE]
Area: [AREA_NAME_HERE]

{
  "area": "[AREA_NAME_HERE]",
  "pedagogical_significance_score": <integer, a score from 1 to 5>,
  "justification": <string, a one-sentence explanation for the score>.
 }

Do not include any preamble, conversational text, explanations, or markdown formatting around the JSON block.
```

Engineering Branch	Authoritative Data Sources
Chemical Engineering	Perry's Chemical Engineers' Handbook NIST Chemistry WebBook CRC Handbook of Chemistry and Physics
Electrical Engineering	IEEE 100: The Authoritative Dictionary of IEEE Standards Terms Standard Handbook for Electrical Engineers ART-DEIT Database
Mechanical Engineering	ASM Handbook, Volume 1 Marks' Standard Handbook for Mechanical Engineers Shigley's Mechanical Engineering Design

Table 3: Primary data sources referenced for engineering parameterization across domains.

E Authoritative Data Sources for Parameterization

To ensure the physical and engineering realism of EngTrace, we built our domain-aware parameterization on authoritative data. We extracted and manually verified extensive lists of physical constants and material properties from the standard handbooks and data sources listed in Table 3.

F Detailed Domain-Aware Parameterization Examples

To ensure that problems are not just mathematically solvable but are also grounded in physical and engineering realism, EngTrace constrains all generated values by the principles of the domain being tested. The following provides specific examples of this methodology for each engineering branch.

F.1 Chemical Engineering

- **Reaction Kinetics and Stoichiometry**: Instead of abstract chemicals, problems select from curated lists of common gas, liquid, and biochemical reactants like Propane, Benzene, and Glucose. More importantly, many problems are built around pre-validated, balanced chemical equations to ensure that the fundamental principles of mass conservation and stoichiometry are carefully upheld from the outset.

- **Thermodynamics**: Problems involving thermodynamic calculations use a multi-layered approach to realism. For common substances, a complete and consistent set of critical properties (T_c, P_c, V_c) is retrieved. For reaction calorimetry, problems utilize tabulated standard heats of formation (ΔH_f°). Crucially, for heat transfer calculations, the model moves beyond constant specific heats by incorporating temperature-dependent heat capacity parameters ($C_p/R = A + BT + CT^2 + DT^{-2}$), ensuring a high degree of physical accuracy across a range of conditions.

- **Transport Phenomena and Fluid Mechanics**: To ensure realism in fluid dynamics problems, templates draw from extensive tables of fluid properties, including the density and viscosity of common liquids and gases. The benchmark also includes parameters for non-Newtonian fluids (e.g., Polymer Solutions & Melts etc), allowing it to test more complex, real-world rheological behaviors beyond simple Newtonian assumptions.

F.2 Electrical Engineering

- **Electromagnetics and Waves**: Instead of using an arbitrary number, the phase velocity of a wave is determined by selecting a specific propagation medium from a curated list (e.g., Distilled Water, Polyethylene, Fused Silica etc). This ensures that all subsequent calculations for parameters like wavelength or wave number are physically consistent with that material's properties. The generation process is further constrained by fundamental physical constants, including the speed of light in a vacuum (c_0) and the permittivity of free space (ϵ_0).

- **Signals and Systems**: Sampling frequencies are intentionally set relative to the Nyquist rate to either model ideal conversion or deliberately induce aliasing, directly testing frequency folding. Signal frequencies are generated as rational multiples of π, and convolution sequences are kept short and integer-valued to focus analysis on the computational process.

- **Digital Communications**: Probabilities for discrete random variables are generated as exact rational numbers summing to one. Carrier frequencies are constrained to integer multiples of the bit rate, modulation orders (M) to powers of two, and energy per bit (E_b) and signal-to-noise ratios (E_b/N_0) to physically typical ranges. Data rates are assigned practical units (kbps, Mbps, Gbps) to ground problems in contemporary engineering contexts.

F.3 Mechanical Engineering

- **Mechanics of Materials**: Problems are grounded in the behavior of real-world materials by co-selecting properties from comprehensive tables. When a material such as "6061-T6 Aluminum" or "Carbon Fiber Reinforced Polymer" is chosen, its physically consistent Young's Modulus (E), Poisson's Ratio (ν), and Shear Modulus (G) are used for all calculations. This ensures that problems involving stress, strain, and torsional deformation are realistic. Properties are available in both SI (GPa) and US Customary (ksi) units to reflect practical engineering work.

- **Fluid Mechanics**: Realism in fluid statics problems is achieved by utilizing extensive lists of fluid and solid densities. This allows for the accurate calculation of hydrostatic pressure and buoyancy forces for a wide range of scenarios, such as a steel object submerged in seawater. The parameterization is context-aware, selecting dense fluids like Mercury for

manometer problems while using lighter fluids like oil or water for pipe flow scenarios.

- **Vibrations and Acoustics**: Mass, spring stiffness, and damping coefficients are randomized within physically plausible ranges. Harmonic forcing amplitudes and frequencies are chosen relative to the system's natural frequency to ensure the dynamic response clearly demonstrates key phenomena such as resonance or beating.

G Template Implementation Examples

The following examples illustrate both parametric and *structural* diversity across templates. Structural variation refers to differences in the type of physical reasoning chain required, not merely in numerical values. The Basic template (`Template Basic Stress Strain`) requires direct formula application under dual unit systems, demanding consistent dimensional reasoning. The Intermediate template (`Template Aliased Frequency Identification`) requires multi-step frequency-domain reasoning, where the sampling constraint fundamentally governs which formula applies. The Advanced template (`Template Statically Indeterminate`) requires constructing and solving a system of equations from first principles, namely equilibrium and compatibility, before any stress calculation is possible. These three are representative of the broader benchmark, where every template introduces a qualitatively distinct reasoning chain.

G.1 Basic

```python
def template_basic_stress_strain():
    # 1. Parameterize inputs with random values, choosing a unit system first.
    use_si_units = random.choice([True, False])
    shape = random.choice(['circular', 'square'])
    precision = 3  # Standardize precision for numerical stability and formatting

    if use_si_units:
        #  SI Unit System
        load = random.randint(10, 500)          # in kN
        length = round(random.uniform(0.5, 4.0), 2)   # in meters
        elongation = round(random.uniform(0.5, 5.0), 2)  # in mm

        if shape == 'circular':
            dimension_val = random.randint(20, 100)  # diameter in mm
            dimension_name = "diameter"
            area = math.pi * (dimension_val / 2)**2  # mm^2
            dim_str = f"{dimension_val} mm"
        else:  # square
            dimension_val = random.randint(20, 100)  # side length in mm
            dimension_name = "side length"
            area = dimension_val**2                  # mm^2
            dim_str = f"{dimension_val} mm"

        load_str      = f"{load} kN"
        length_str    = f"{length} m"
        elongation_str= f"{elongation} mm"

        # Core Calculations (N, mm, MPa) - 1 MPa = 1 N/mm^2
        stress = (load * 1000) / area           # Stress in MPa
        strain = elongation / (length * 1000)    # Strain (dimensionless)

        stress_unit           = "MPa"
        strain_unit_explanation = "dimensionless"

    else:
        #  US Customary Unit System
        load       = random.randint(5, 100)  # in kips
        length     = round(random.uniform(24.0, 120.0), 1)  # in inches
        elongation = round(random.uniform(0.05, 0.25), 3)  # in inches

        if shape == 'circular':
            dimension_val  = round(random.uniform(1.0, 5.0), 2)  # diameter in inches
            dimension_name = "diameter"
            area           = math.pi * (dimension_val / 2)**2    # in^2
            dim_str        = f"{dimension_val} in"
        else:  # square
            dimension_val  = round(random.uniform(1.0, 5.0), 2)  # side in inches
            dimension_name = "side length"
            area           = dimension_val**2
            dim_str        = f"{dimension_val} in"

        load_str       = f"{load} kips"
        length_str     = f"{length} in"
        elongation_str = f"{elongation} in"

        # Core Calculations (kips, in, ksi) - 1 ksi = 1 kip/in^2
        stress = load / area            # Stress in ksi
        strain = elongation / length    # Strain (dimensionless)

        stress_unit           = "ksi"
        strain_unit_explanation = "dimensionless"

    # 2. Generate the question and solution strings
    question = (
        f"A prismatic bar with a length of {length_str} has a {shape} cross-section "
        f"with a {dimension_name} of {dim_str}. The bar is subjected to an axial "
        f"tensile load of {load_str}, causing it to elongate by {elongation_str}.\n\n"
        f"Determine the following:\n"
        f"a) The normal stress in the bar.\n"
        f"b) The normal strain in the bar."
    )

    solution = (
        f"**Given:**\n"
        f"Load (P): {load_str}\n"
        f"Length (L): {length_str}\n"
        f"Elongation (delta): {elongation_str}\n"
        f"Cross-Section: {shape.capitalize()} with {dimension_name} = {dim_str}\n\n"
        f"**Step 1:** Calculate the Cross-Sectional Area (A)\n"
        f"The area of a {shape} cross-section is calculated as follows:\n"
    )

    if shape == 'circular':
        solution += (
            f"A = pi * (d/2)^2 = pi * ({dimension_val}/2)^2 = {round(area, precision+1)} "
            f"{'mm^2' if use_si_units else 'in^2'}\n\n"
        )
    else:  # square
        solution += (
            f"A = side^2 = ({dimension_val})^2 = {round(area, precision+1)} "
            f"{'mm^2' if use_si_units else 'in^2'}\n\n"
        )
```

```
    solution += (
        f"**Step 2:** Calculate the Normal Stress (sigma)\n"
        f"Normal stress is defined as the force per unit
        ↪ area: sigma = P / A.\n"
    )

    if use_si_units:
        solution += (
            f"To get the result in Megapascals (MPa), we
            ↪ use the load in Newtons (N) "
            f"and the area in mm^2 (1 MPa = 1 N/mm^2).\n"
            f"P = {load} kN = {load * 1000} N\n"
            f"A = {round(area, precision+1)} mm^2\n"
            f"sigma = ({load * 1000} N) / ({round(area,
            ↪ precision+1)} mm^2) "
            f"= {round(stress, precision)} MPa\n\n"
        )
    else:  # US units
        solution += (
            f"To get the result in kips per square inch
            ↪ (ksi), we use the load in kips "
            f"and the area in in^2 (1 ksi = 1 kip/in^2).\n"
            f"P = {load} kips\n"
            f"A = {round(area, precision+1)} in^2\n"
            f"sigma = ({load} kips) / ({round(area,
            ↪ precision+1)} in^2) "
            f"= {round(stress, precision)} ksi\n\n"
        )

    solution += (
        f"**Step 3:** Calculate the Normal Strain
        ↪ (epsilon)\n"
        f"Normal strain is the change in length per unit
        ↪ original length: "
        f"epsilon = delta / L.\n"
        f"It is a dimensionless quantity; units must be
        ↪ consistent.\n"
    )

    if use_si_units:
        solution += (
            f"We convert length from meters to millimeters
            ↪ to match elongation units.\n"
            f"delta = {elongation} mm\n"
            f"L = {length} m = {length * 1000} mm\n"
            f"epsilon = ({elongation} mm) / ({length *
            ↪ 1000} mm) = {strain:.3e}\n\n"
        )
    else:  # US units
        solution += (
            f"Both elongation and length are already in
            ↪ inches.\n"
            f"delta = {elongation} in\n"
            f"L = {length} in\n"
            f"epsilon = ({elongation} in) / ({length} in) =
            ↪ {strain:.3e}\n\n"
        )

    solution += (
        f"**Answer:**\n"
        f"a) Normal Stress (sigma) = **{round(stress,
        ↪ precision)} {stress_unit}**\n"
        f"b) Normal Strain (epsilon) = **{strain:.3e}**
        ↪ ({strain_unit_explanation})"
    )

    return question, solution
```

G.2 Intermediate

```
def template_aliased_frequency_identification():
    # 1. Parameterize the inputs with random values

    # Generate a random original frequency for the
    ↪ continuous-time signal.
    f0 = random.randint(*F0_RANGE_HZ)

    # Calculate the corresponding Nyquist rate for this
    ↪ signal.
    f_nyquist = 2 * f0

    # To guarantee aliasing, generate a sampling frequency
    ↪ (Fs) that is strictly
    # less than the Nyquist rate. We set the range to be
    ↪ between 50% and 95% of
    # the Nyquist rate to create a non-trivial
    ↪ undersampling problem.
    # The lower bound is f_nyquist/2 + 1 = f0 + 1 to
    ↪ prevent the degenerate case
    # where fs == f0, which would produce an aliased
    ↪ frequency of exactly 0 Hz.
    fs = random.randint(int(0.5 * f_nyquist) + 1, int(0.95
    ↪ * f_nyquist))

    # 2. Perform the core calculation for the solution

    # Find the integer multiple 'k' which represents the
    ↪ nearest multiple of the
    # sampling frequency to the original frequency.
    k = round(f0 / fs)

    # Calculate the aliased frequency (Fa) using the
    ↪ folding formula.
    fa = abs(f0 - k * fs)

    # Standardize the precision for all final outputs for
    ↪ consistency.
    precision = 2

    # 3. Generate the question and solution strings
    question = (
        f"A continuous-time signal x_c(t) =
        ↪ cos(2*pi*{f0}*t) is sampled at a rate "
        f"of {fs} samples per second.\n\n"
        f"Since this rate is below the Nyquist rate,
        ↪ aliasing occurs. What is the "
        f"apparent frequency, Fa, of the resulting
        ↪ discrete-time signal? "
        f"(The apparent frequency must be in the range 0 <=
        ↪ Fa <= {fs/2} Hz)."
    )

    solution = (
        f"**Given:**\n"
        f"Original Frequency (F0): {f0} Hz\n"
        f"Sampling Frequency (Fs): {fs} Hz\n\n"

        f"**Step 1:** State the Condition\n"
        f"The Nyquist rate required to sample this signal
        ↪ without aliasing is "
        f"2 * F0 = 2 * {f0} = {f_nyquist} Hz. "
        f"Since the sampling frequency Fs = {fs} Hz is less
        ↪ than {f_nyquist} Hz, "
        f"the original frequency will be aliased.\n\n"

        f"**Step 2:** Explain Frequency Folding\n"
        f"When aliasing occurs, the perceived frequency
        ↪ (Fa) is the absolute "
        f"difference between the original frequency and the
        ↪ nearest integer multiple "
        f"of the sampling frequency. The formula is:\n"
        f"Fa = abs(F0 - k * Fs), where 'k' is an
        ↪ integer.\n\n"

        f"**Step 3:** Find the Correct Multiple (k)\n"
        f"We find the integer 'k' that 'folds' F0 into the
        ↪ principal frequency "
        f"range [0, Fs/2]. This is found by rounding the
        ↪ ratio of F0 to Fs.\n"
        f"k = round(F0 / Fs) = round({f0} / {fs}) =
        ↪ round({round(f0/fs, 4)}) = {k}.\n\n"

        f"**Step 4:** Calculate Aliased Frequency (Fa)\n"
        f"Now, we compute Fa using the value of k found in
        ↪ the previous step.\n"
        f"Fa = abs({f0} - {k} * {fs})\n"
        f"Fa = abs({f0} - {k * fs})\n"
        f"Fa = {round(fa, precision)} Hz.\n\n"

        f"**Answer:**\n"
        f"The apparent (aliased) frequency of the resulting
        ↪ signal is "
        f"{round(fa, precision)} Hz."
    )

    return question, solution
```

G.3 Advanced

```python
def template_statically_indeterminate():
    # 1. Parameterize inputs
    use_si_units = random.choice([True, False])
    # Restrict to structural-grade materials (E >= 10 GPa) to avoid physically
    # unrealistic scenarios with rubbers and soft polymers under structural loads.
    structural_materials = [k for k in MATERIAL_PROPERTIES.keys()
                            if MATERIAL_PROPERTIES[k]['E_GPa'] >= 10]
    material_name = random.choice(structural_materials)
    material = MATERIAL_PROPERTIES[material_name]
    shape = random.choice(['circular', 'square'])
    precision = 3

    if use_si_units:
        #  SI Unit System
        total_length = round(random.uniform(1.0, 3.0), 2) # m
        len_AB = round(random.uniform(0.2 * total_length, 0.8 * total_length), 2)  # m
        len_BC = total_length - len_AB
        load_P = random.randint(100, 500)  # kN

        if shape == 'circular':
            diameter = random.randint(50, 150)  # mm
            area = math.pi * (diameter / 2)**2
            dim_str = f"a diameter of {diameter} mm"
        else:  # square
            side = random.randint(50, 150)  # mm
            area = side**2
            dim_str = f"a side length of {side} mm"

        E_val = material['E_GPa']
        unit_L, unit_P, unit_S, unit_A, unit_E = "m", "kN", "MPa", "mm^2", "GPa"

    else:
        #  US Customary Unit System
        total_length = round(random.uniform(40.0, 120.0), 1)  # inches
        len_AB = round(random.uniform(0.2 * total_length, 0.8 * total_length), 1)  # in
        len_BC = total_length - len_AB
        load_P = random.randint(50, 250)  # kips

        if shape == 'circular':
            diameter = round(random.uniform(2.0, 6.0), 2) # inches
            area = math.pi * (diameter / 2)**2
            dim_str = f"a diameter of {diameter} in"
        else:  # square
            side = round(random.uniform(2.0, 6.0), 2)  # inches
            area = side**2
            dim_str = f"a side length of {side} in"

        E_val = material['E_ksi']
        unit_L, unit_P, unit_S, unit_A, unit_E = "in", "kips", "ksi", "in^2", "ksi"

    # 2. Core Calculations
    # From compatibility: R_A = P * L_BC / L_AC
    # From equilibrium:   R_C = P - R_A
    R_A = load_P * (len_BC / total_length)
    R_C = load_P - R_A

    # Internal forces
    P_AB =  R_A   # Tension
    P_BC = -R_C   # Compression

    # Stresses
    stress_AB = P_AB / area
    stress_BC = P_BC / area

    # Unit conversion for SI stress
    if use_si_units:
        stress_AB_MPa = stress_AB * 1000  # kN/mm^2 -> MPa
        stress_BC_MPa = stress_BC * 1000  # kN/mm^2 -> MPa

    # 3. Generate Question and Solution Strings
    question = (
        f"A solid {material_name} rod with a uniform {shape} cross-section is fixed "
        f"at both ends, A and C. The rod has a total length of {total_length} {unit_L} "
        f"and {dim_str}.\n\n"
        f"An external axial load of {load_P} {unit_P} is applied to the right at "
        f"point B, which is located at a distance of {len_AB} {unit_L} from end A.\n\n"
        f"The Modulus of Elasticity for {material_name} is {E_val} {unit_E}.\n\n"
        f"Determine the following:\n"
        f"a) The reaction forces at supports A and C.\n"
        f"b) The normal stress in segments AB and BC of the rod."
    )

    solution = (
        f"**Given:**\n"
        f"Total Length (L_AC): {total_length} {unit_L}\n"
        f"Length of segment AB (L_AB): {len_AB} {unit_L}\n"
        f"Length of segment BC (L_BC): {total_length} - {len_AB} = "
        f"{round(len_BC, 2)} {unit_L}\n"
        f"Applied Load (P): {load_P} {unit_P}\n"
        f"Area (A): {round(area, 2)} {unit_A}\n"
        f"Modulus of Elasticity (E): {E_val} {unit_E}\n\n"

        f"This is a statically indeterminate problem because there are two unknown "
        f"support reactions (R_A and R_C) and only one equation of static "
        f"equilibrium.\n\n"

        f"**Step 1:** Equation of Equilibrium\n"
        f"Summing forces in the x-direction (assuming right is positive):\n"
        f"Sum(F_x) = 0  =>  -R_A + P - R_C = 0\n"
        f"R_A + R_C = {load_P} {unit_P}  -(Equation 1)\n\n"

        f"**Step 2:** Equation of Compatibility\n"
        f"Since the rod is fixed between unyielding supports, the total deformation "
        f"must be zero.\n"
        f"delta_total = delta_AB + delta_BC = 0\n"
        f"The internal force in segment AB is the reaction R_A "
        f"(in tension, P_AB = R_A).\n"
        f"The internal force in segment BC is the reaction R_C "
        f"(in compression, P_BC = -R_C).\n\n"
        f"Using the deformation formula delta = PL/AE:\n"
        f"(R_A * L_AB) / (A * E) - (R_C * L_BC) / (A * E) = 0\n"
        f"Since A and E are constant, they cancel out:\n"
        f"R_A * L_AB = R_C * L_BC  -(Equation 2)\n\n"

        f"**Step 3:** Solve for the Reaction Forces\n"
        f"From Equation 1, we can express R_C as: R_C = {load_P} - R_A.\n"
        f"Substitute this into the simplified Equation 2:\n"
        f"R_A * ({len_AB}) = ({load_P} - R_A) * ({round(len_BC, 2)})\n"
        f"{round(len_AB, 2)}*R_A = {round(load_P * len_BC, 2)} - "
        f"{round(len_BC, 2)}*R_A\n"
        f"({round(len_AB, 2)} + {round(len_BC, 2)})*R_A = "
        f"{round(load_P * len_BC, 2)}\n"
        f"({total_length})*R_A = {round(load_P * len_BC, 2)}\n"
        f"R_A = {round(load_P * len_BC, 2)} / {total_length} = "
        f"{round(R_A, precision)} {unit_P}\n\n"
        f"Now, find R_C using Equation 1:\n"
        f"R_C = {load_P} - R_A = {load_P} - {round(R_A, precision)} = "
        f"{round(R_C, precision)} {unit_P}\n\n"

        f"**Step 4:** Calculate the Normal Stresses\n"
        f"The stress in each segment is sigma = P_internal / A.\n"
        f"Segment AB: The internal force is P_AB = R_A = "
        f"{round(R_A, precision)} {unit_P} (Tension).\n"
    )

    if use_si_units:
        solution += (
            f"sigma_AB = ({round(R_A, precision)} kN) / ({round(area, 2)} mm^2) "
            f"* 1000 = {round(stress_AB_MPa, precision)} MPa (Tension)\n"
        )
    else:
```

```
        solution += (
            f"sigma_AB = {round(R_A, precision)} kips /
            ↪ {round(area, 2)} in^2 "
            f"= {round(stress_AB, precision)} ksi
            ↪ (Tension)\n"
        )

    solution += (
        f"Segment BC: The internal force is P_BC = -R_C = "
        f"-{round(R_C, precision)} {unit_P}
        ↪ (Compression).\n"
    )

    if use_si_units:
        solution += (
            f"sigma_BC = (-{round(R_C, precision)} kN) /
            ↪ ({round(area, 2)} mm^2) "
            f"* 1000 = {round(stress_BC_MPa, precision)}
            ↪ MPa (Compression)\n\n"
        )
    else:
        solution += (
            f"sigma_BC = -{round(R_C, precision)} kips /
            ↪ {round(area, 2)} in^2 "
            f"= {round(stress_BC, precision)} ksi
            ↪ (Compression)\n\n"
        )

    solution += (
        f"**Answer:**\n"
        f"a) Reaction Forces: R_A = **{round(R_A,
        ↪ precision)} {unit_P}** and "
        f"R_C = **{round(R_C, precision)} {unit_P}**.\n"
        f"b) Normal Stresses: "
        f"sigma_AB = **{round(stress_AB_MPa if use_si_units
        ↪ else stress_AB, precision)}"
        f" {unit_S} (Tension)** and "
        f"sigma_BC = **{round(abs(stress_BC_MPa if
        ↪ use_si_units else stress_BC), precision)}"
        f" {unit_S} (Compression)**."
    )

    return question, solution
```

H LLM Prompt for Automated AI Pre-Screening

```
You are an expert engineering professor and
a senior Python developer acting as a peer
reviewer for the EngTrace benchmark.

Your task is to meticulously evaluate a new
problem template based on its source code and
several example outputs.

Analyze the provided information and then
respond ONLY with a single, valid JSON object
that strictly adheres to the schema described
in the user prompt. Do not add any explanatory
text or markdown formatting around the JSON
object.

Please evaluate the following engineering
problem template.

1. Template Source Code:
python {template_code}

2. Generated Instances from the Template:
Instance 1:
- Question: "{q1}"
- Solution: "{s1}"

Instance 2:
- Question: "{q2}"
- Solution: "{s2}"

Instance 3:
- Question: "{q3}"
- Solution: "{s3}"

3. Evaluation Rubric & JSON Schema:
Evaluate the template based on the rubric
below. The human_review_flag should be
true if any score is less than 4. The
explanation should be a concise, one-sentence
justification for the scores and the flag.

{

  "physical_plausibility_score": <integer, a
score from 1-5 based on whether the problem
respects the laws of physics and engineering>,

  "mathematical_correctness_score": <integer,
a score from 1-5 based on whether the
equations and calculations are accurate>,

  "pedagogical_clarity_score": <integer,
a score from 1-5 based on whether the
problem statement is clear, unambiguous, and
solvable>,

  "confidence_score": <integer, a score
from 1-5 indicating your confidence in this
evaluation>,

  "human_review_flag": <boolean, true if the
template requires human inspection, otherwise
false>,

  "explanation": "<string, a concise,
one-sentence justification for the scores and
flag>"

}
```

I Multi-Axis Rubric for Automated AI Pre-Screening

During automated AI pre-screening, judges in the AI Tribunal evaluate templates against a set of criteria. The evaluation rubric, detailing scoring categories and descriptions, is provided in Table 4.

J Human Certification Interface

To facilitate the rigorous Stage 2 evaluation, we developed a custom web-based annotation tool using the `Streamlit` framework. This interface, illustrated in Figure 6, was designed to streamline the expert review process while ensuring data consistency for statistical analysis.

The workflow within the interface proceeds as follows:

1. **Dual-View Inspection:** The main panel displays the rendered natural language `Question` and step-by-step `Solution Trace` exactly as

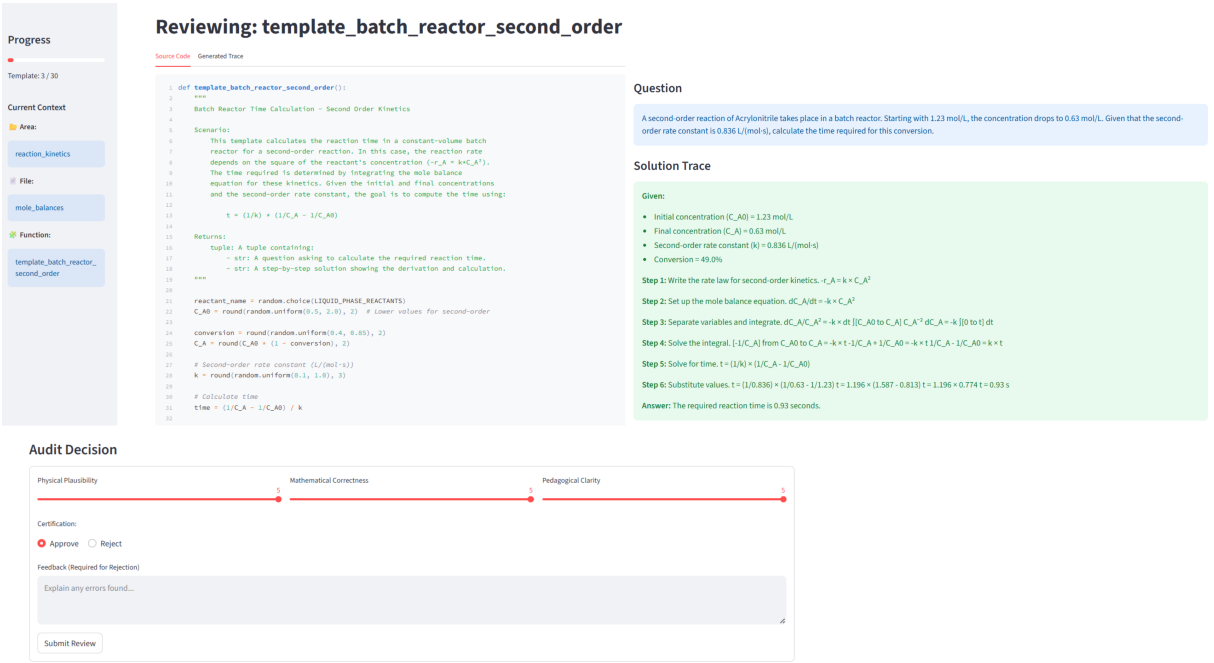

Figure 6: The **EngTrace Expert Annotation Interface.** The dashboard features a dual-view layout for verifying source code against rendered traces, enabling experts to input structured quality scores and definitive admissibility decisions.

they appear in the final dataset. Simultaneously, reviewers can toggle a tab to inspect the underlying Python `Source Code`, allowing them to verify the algorithmic logic alongside the output.

2. **Dynamic Re-sampling:** To verify consistency beyond a single static snapshot, the interface allows reviewers to interactively regenerate instances. This enables experts to stress-test the parameterization logic and confirm that physical constraints hold across multiple random seeds.

3. **Ordinal Quality Scoring:** Reviewers evaluate the template across three dimensions (Physical Plausibility, Mathematical Correctness, and Pedagogical Clarity) using interactive sliders (1–5). These values provide the ordinal data required for calculating Krippendorff's Alpha (α).

4. **Binary Certification Gate:** The "Certification" toggle forces a definitive `Approve` or `Reject` decision, generating the categorical data for Fleiss' Kappa (κ).

5. **Mandatory Feedback Loop:** If a template is rejected, the interface enforces a mandatory text input field for specific feedback (e.g., "The mole balance assumption is invalid for variable volume"). This feedback is logged and routed back to the authors for immediate correction.

K Template Validation Results

K.1 Stage 1: AI Tribunal Impact

In the initial audit, the AI Tribunal identified 26 `Critical Failures` and 6 `Controversial` templates. These templates were manually refactored based on the specific failure modes identified by the models (e.g., unit mismatches or unrealistic material properties) until a 100% pass rate was achieved. Representative examples of these identified failures and their associated reasoning are provided in Table 5.

To assess filter reliability, domain experts independently reviewed the 32 templates flagged by the initial AI Tribunal run prior to any refactoring. This audit confirmed 21 flagged templates as genuinely defective and identified 11 as borderline valid templates conservatively rejected, consistent with a pipeline designed to prioritize benchmark integrity over recall (FPR = 0.00%).

K.2 Stage 2: Inter-Annotator Agreement

We measured agreement among three domain experts per branch using Fleiss' κ for binary admissibility. For ordinal quality ratings, we report

Score Category	Description
`Physical Plausibility`	Does the problem scenario respect the laws of physics and engineering?
`Mathematical Correctness`	Are the equations used correct, and is the final calculation accurate based on the inputs?
`Pedagogical Clarity`	Is the problem statement clear, unambiguous, and solvable with the information provided?
`Confidence Score`	How confident is the LLM in its own validation of this template?
`Human Review Flag`	Based on the scores above, does this template require human inspection?
`Explanation`	A concise, one-sentence justification for the scores and the flag.

Table 4: Multi-axis rubric used for AI Tribunal based template validation. Numerical criteria are rated on a 1–5 scale, and the `Human Review Flag` is a binary (true/false) indicator.

two complementary coefficients: Krippendorff's α and Gwet's AC2 (Gwet, 2008; Wongpakaran et al., 2013). The two metrics differ in how they estimate chance agreement. Krippendorff's α derives its chance baseline from the observed marginal distributions, which causes it to systematically underestimate reliability when scores cluster in a narrow band (Feinstein and Cicchetti, 1990), a known failure mode called the kappa paradox. Gwet's AC2 uses a chance estimator that accounts for score concentration and remains stable under range restriction; it is therefore our primary ordinal metric.

Binary admissibility. As shown in Table 6, all three branches achieved perfect binary consensus ($\kappa = 1.0$). Every template passing Stage 1 was unanimously admitted, confirming that the AI Tribunal successfully acted as a validity pre-filter and that human reviewers were aligned on final inclusion decisions.

Ordinal quality ratings. Table 7 presents α and AC2 for all three quality dimensions. Mathematical Correctness achieves perfect agreement under both metrics ($\alpha = \text{AC2} = 1.0$) across all branches, as expected for an objective verification task. This also serves as a rubric validity check confirming annotators applied the scoring criteria consistently. For Physical Plausibility and Pedagogical Clarity, α and AC2 diverge substantially. Globally, Physical Plausibility yields $\alpha = 0.437$ but AC2 $= 0.934$; Pedagogical Clarity yields $\alpha = 0.269$ but AC2 $= 0.824$. This gap is a known artifact of Krippendorff's estimator under range restriction. Because Stage 1 filtered out low-quality templates, surviving scores cluster in the $\{4, 5\}$ band; α treats this narrow range as spanning a large fraction of the full scale and collapses accordingly. AC2 accounts for score concentration and reports the agreement the data actually reflect.

To empirically verify this, we examined pairwise score differences for Mechanical Engineering Pedagogical Clarity, the most extreme case ($\alpha = 0.078$, AC2 $= 0.745$). Across all 90 pairwise comparisons over 30 templates, 70% showed exact agreement, 30% differed by exactly one point, and no comparison exceeded a gap of one. Of the 30 templates, 21 received unanimous $[5, 5, 5]$ scores; the remaining 9 had one annotator give 4 while the other two gave 5. No annotator assigned below 4 on any template. The pairwise distribution confirms that the near-zero α is entirely a distributional artifact of ceiling-level scores, not genuine annotator disagreement. The residual one-point variation reflects Human Label Variation (Plank, 2022): rater-specific judgments on pedagogical style and edge-case parameter realism inherent to subjective expert evaluation. AC2 values of 0.824–0.934 across all subjective dimensions confirm strong and reliable expert agreement on the final template set.

K.3 AI-Human Alignment

We evaluate alignment between the AI Tribunal (Stage 1) and human experts (Stage 2) along two distinct validation axes: filter safety and scoring consistency.

Filter safety. Of the 90 templates approved by the AI Tribunal, zero were subsequently rejected by human experts (FPR = 0.00%), confirming that the Tribunal introduces no false positives into the human review stage. Combined with the expert audit reported in § K.1, this establishes that the two-stage pipeline correctly partitions the template space: the Tribunal eliminates genuinely defective templates, and human reviewers operate on a set free of fundamental errors.

Scoring consistency. To quantify how closely AI and human quality judgments agree in magnitude, we compute the Mean Absolute Difference (MAD)

between AI Tribunal scores and human expert median scores across all 90 certified templates on each quality dimension. MAD operates in original scale units and is robust to the ceiling effects discussed in § K.2, making it the appropriate primary metric when scores concentrate in a narrow band. We adopt MAD < 0.5 as the agreement threshold, corresponding to less than half a scale step on the 1–5 rubric.

Table 8 reports the results. All three dimensions fall well within this threshold. Mathematical Correctness and Physical Plausibility reach MAD values of 0.156 and 0.167 respectively, indicating near-identical AI and human scoring on objective and semi-objective dimensions. Pedagogical Clarity reaches 0.233, a slightly higher but expected value given the inherent subjectivity of evaluating educational presentation. The overall MAD of 0.174 confirms that AI and human quality judgments are consistent across all engineering branches. Pairwise difference distributions corroborate this: across all dimensions, 77–84% of AI-human comparisons show exact score agreement, and no comparison exceeds a one-point gap on any dimension.

Spearman's ρ is low across all dimensions ($\rho = 0.065$–0.164, all $p > 0.10$) and is reported for completeness.[1]

L Dataset Statistics

Structural Balance and Granularity. Each branch is represented by exactly 30 templates, ensuring that no single discipline dominates the aggregate performance metrics. This uniform top-level distribution controls for branch-specific bias. At the granular level, the number of Areas (6–7 per branch) reflects the specific curricular density of each discipline; for instance, Mechanical Engineering consolidates multiple sub-topics into broad areas like Fluid Mechanics, resulting in slightly fewer total areas (6) compared to Chemical and Electrical (7), without sacrificing concept coverage.

Difficulty Distribution. The dataset difficulty is intentionally stratified to mirror curricular progression. We prioritize foundational and applied reasoning (Level 1 & 2: ≈75%) to test core engineering laws and procedural workflows, while reserving the most complex synthesis tasks for the advanced tier (Level 3: ≈25%). This weighting ensures models are evaluated primarily on standard practice while still distinguishing elite capabilities in multi-physics integration. Table 9 presents the detailed statistical breakdown.

M Evaluation Framework Validation

To ensure that the EngTrace evaluation framework reflects expert engineering judgment, we conducted a rigorous human-alignment study. We established a validation set of 20 instantiated questions, randomly sampled with stratification across all three engineering branches and all three difficulty levels, and generated responses from five representative models spanning frontier, open-weights, and math-enhanced categories: `DeepSeek R1`, `Gemini 3 Pro`, `Gemma 3 27B`, `Qwen2.5-Math 7B`, and `Llama 3.1 8B`.

This procedure resulted in 100 randomized and anonymized responses. The sample is deliberately stratified rather than randomly drawn: the five models are selected to represent contrasting capability profiles, and the 20 questions ensure coverage across all branch-difficulty combinations, yielding a validation set that is representative by design rather than by scale. Three subject matter experts per branch independently scored each response on a 1–5 Likert scale, providing nine independent expert judgments per branch. Our initial analysis revealed a strong correlation between human-rated reasoning quality and final answer accuracy (Spearman's $\rho = 0.9130$, $p < 0.05$), confirming that coherent reasoning traces are highly predictive of numerical correctness. Consequently, we prioritized reasoning process quality as the primary dimension for optimizing our automated evaluator.

M.1 Expert Evaluation Rubrics

The following rubrics were utilized by subject matter experts to evaluate the 100 validation responses. Experts provided independent scores for reasoning and accuracy on a 1–5 Likert scale as detailed in Table 10 and Table 11.

M.2 Ablation Study Results

We systematically assessed various evaluator configurations to optimize alignment with these expert ratings. Table 12 shows the Spearman's Rank Correlation (ρ) between automated pipeline scores and human judgment.

[1] Under range restriction, when scores cluster in $\{4, 5\}$, rank correlation is dominated by tie-breaking noise rather than systematic disagreement (Artstein and Poesio, 2008). MAD is the appropriate primary metric as it operates in original scale units and is independent of score distribution shape.

The results demonstrate that the combination of an AI Tribunal ($\tau = 0.8$) and numeric step-matching yields the highest alignment ($\rho = 0.632$). This alignment significantly degrades upon the removal of the Tribunal ($\rho = 0.411$), confirming its necessity for capturing valid alternative derivations. Furthermore, the poor performance of standard NLP metrics ($\rho = 0.325$) reinforces the need for the domain-specific numerical verification employed in EngTrace.

N LLM Prompt for Multi-Model Heuristic Verification

You are an expert engineering professor acting as an automated evaluator. Your task is to analyze the "MODEL'S STEP" against the "GROUND-TRUTH STEP" and return a structured JSON object based on the rules and inputs below.

CRITICAL RULES:
1. Your "explanation" MUST logically justify your chosen "error_category". Do not contradict yourself. For example, do not choose "Calculation Error" and then state that the calculation is correct.
2. If the model's step is factually correct but takes a different path than the ground-truth, you MUST use the "Alternative Correct" category. Do not classify a correct step as "Other".
3. The final output must be only a raw JSON object. Do not include any introductory text, concluding remarks, or markdown formatting.

Error Categories:
• "Conceptual Error": The model applied the wrong scientific principle or formula.
• "Calculation Error": The model used the correct formula but made a mathematical mistake.
• "Alternative Correct": The model's step is valid and logically sound, but follows a different method or phrasing than the ground-truth step.
• "Other": The model's step is nonsensical, irrelevant, a hallucination, or contains only formatting errors.

Input for Analysis:

```
{
  [CONTEXT]: {problem_context}
  [GROUND-TRUTH STEP]: {gt_step}
  [MODEL'S STEP]: {pred_step}
}
```

OUTPUT FORMAT:
You must now provide your analysis. Your entire response will be a single, raw JSON object. Adhere strictly to the following format with exactly two keys:
{"error_category": "...", "explanation": "..."}

O Evaluated Model Details

Table 13 details the 27 models evaluated in our experiments (§ 5). We report the developing organization, parameter size, backbone architecture (where applicable), and the specific access identifier used to ensure reproducibility.

P Inference Prompt and Output Processing

P.1 Standardized Inference Prompt

To ensure consistent and parsable outputs across all models, we employed a standardized zero-shot system prompt. This prompt explicitly instructs the model to structure its response with specific headings and to format the final result for automated extraction.

```
You are an expert engineer. Solve the following
problem by providing a detailed, structured
solution. Use the exact headings and
formatting provided below.

# Given
List all known variables and their
values with units.

# Find
State the variable(s) to be calculated.

# Formulae
Write down all necessary governing equations
before substituting any values.

# Solution
Provide a step-by-step calculation. Each step
must start on a new line and be formatted exactly
as '**Step X:**', where X is the step number.
Show the substitution of values into
the formulae clearly.

# Final Answer
State the final numerical result with its units
in the format: **Answer:** [value] [units]
```

P.2 Output Parsing Logic

To process the model outputs generated by the prompt above, we employ a heuristic parsing pipeline:

1. **Step Segmentation:** We use regular expressions to identify line-start markers matching either the `Step <number>` format (e.g., `**Step 1**`) or the numbered-list format (e.g., `1.`). A subsequent regex cleans these prefixes to isolate the reasoning text.

2. **Value Extraction:** We extract numerical values from each step and the final answer using a three-priority heuristic:

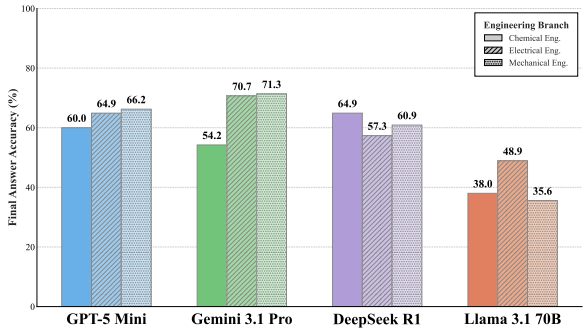

Figure 7: **Branch-Level Performance** across frontier and open-weights models. Chemical Engineering is the hardest branch across all model classes, with even leading frontier models scoring substantially lower than on Electrical and Mechanical Engineering, while Mechanical Engineering remains the most accessible due to its reliance on single-formula substitution patterns.

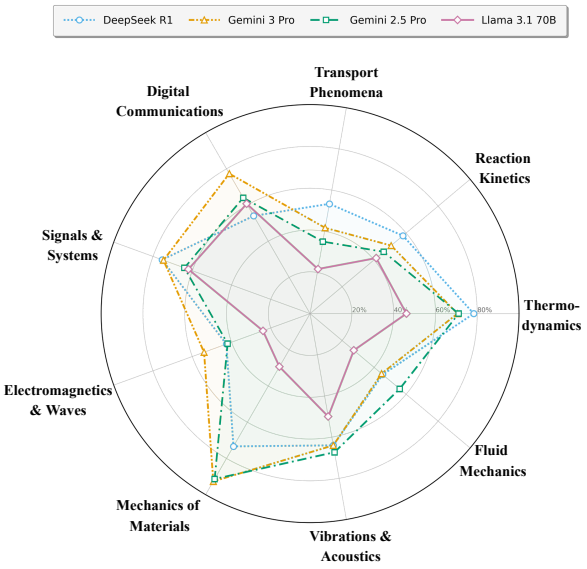

Figure 8: **Domain-Level Performance** of four representative models spanning frontier proprietary and open-weights classes. The spiky profile confirms that model capabilities are domain-specialized rather than generalized across engineering subfields.

- *Priority 1:* Matching the explicit `**Answer:**` tag defined in the prompt.
- *Priority 2:* Identifying the last number following an equals sign (e.g., `V = 100.0`).
- *Priority 3:* Extracting the last standalone number in the segment.

3. **Fallback:** If no overall `**Answer:**` tag is found, the final numerical value extracted from the last reasoning step is used for verification.

Q Branch and Domain-Level Performance

Figure 7 presents the branch-level Final Answer Accuracy for four representative models spanning frontier and open-weights classes. Chemical Engineering is consistently the hardest branch across all models, while Mechanical Engineering remains the most accessible. The gap between branches widens substantially for weaker models, confirming that the branch difficulty ordering is not a scaling artifact but a structural property of the benchmark. Figure 8 further reveals a spiky domain-level capability profile, confirming that model capabilities are domain-specialized rather than generalized across engineering subfields.

R Error Taxonomy and Annotation Protocol

R.1 Error Categories

To support systematic analysis of reasoning failures, we developed a six-category error taxonomy grounded in the sequential structure of engineering problem-solving. Table 14 defines the categories in order of decreasing severity: an error at a higher stage invalidates all subsequent steps, so a model that fabricates a governing equation must not be labeled a `Calculation Error`, as this would misrepresent the root cause of failure.

R.2 Decision Hierarchy

Annotators assigned categories by answering six binary questions in strict top-to-bottom order, stopping at the first affirmative response. This stop-at-first-YES rule captures the most fundamental error present rather than a downstream symptom: a model that selects the wrong governing equation (Q3) cannot produce a dimensionally consistent result by chance, so assigning a lower-level label would obscure the root cause. The six questions, in order, are:

Q1. Does the model use a value, constant, or equation that does not exist in any real engineering reference? → `Hallucination`

Q2. Is the problem framed incorrectly before any equation is applied (wrong boundary condition, ignored constraint, or misidentified system configuration)? → `Setup / Assumption Error`

Q3. Is the selected governing equation or principle wrong, given that the problem setup is correct? → `Formula / Principle Error`

Q4. Does dimensional analysis fail (missing conversion, mixed unit systems, or a value substituted in the wrong unit)? → `Unit / Dimensional Error`

Q5. Is the sign or physical direction of any quantity wrong, given that the formula and units are correct? → `Sign / Direction Error`

Q6. Does arithmetic or algebraic execution produce the wrong result, given that all prior steps are correct? → `Calculation Error`

Annotators additionally recorded the *triggering text span*, a verbatim excerpt directly evidencing the assigned error. Representative traces for each category, selected on the basis of unanimous annotator agreement, are provided in Table 15.

S Error Analysis Sampling Strategy

S.1 Sample Selection Criteria

We applied a three-condition filter to identify confirmed reasoning failures. Let $\mathcal{A}(x)$ denote binary final answer correctness, $\mathcal{T}(x) \in \{0, 1\}$ indicate Tier 2 escalation (§ 4.4), and $F(x)$, $R(x)$ denote step-level alignment and recall. A trace x is a confirmed failure if and only if:

$$\mathcal{A}(x) = 0 \tag{8}$$

$$\mathcal{T}(x) = 0 \tag{9}$$

$$F(x) < 0.30 \quad \vee \quad R(x) < 0.20 \tag{10}$$

Condition (8) requires an incorrect final answer. Condition (9) excludes Tier 2-escalated traces, whose correctness was already adjudicated through a separate consensus mechanism. Condition (10) guards against false negatives: a trace marked incorrect but with high step-level scores is likely a scoring artifact rather than a genuine failure. Thresholds were calibrated by inspecting 20 borderline Gemini 3.1 Pro traces ($F(x) \in [0.20, 0.40]$), all of which were confirmed genuine failures requiring no threshold adjustment.

S.2 Stratified Sampling Procedure

For each model, confirmed failure traces were stratified by difficulty tier using proportions derived from the full 1,350-record benchmark distribution rather than the failure pool, ensuring the sampled structure reflects the benchmark design. A stratified sample of 200 traces per model was drawn using random seed 42; the failure pool exceeded the per-tier target in every case. Across all eleven models, 5,708 traces satisfied the filter, yielding a total annotation sample of 2,200 traces (600 main, 1,600 extended) spanning three branches and ten domains. Full counts are reported in Table 16.

S.3 Model Selection Rationale

The eleven models were selected to maximise contrastive interpretability rather than span a performance gradient. Gemini 3.1 Pro and DeepSeek R1 anchor the two poles of the FAC-versus-F1 tension: Gemini leads FAC (65.41%) while DeepSeek R1 leads F1 (44.41%), reflecting its RL-trained reasoning emphasis. Llama 3.1 70B exhibits the most pronounced complexity cliff (52.55% Easy to 23.33% Advanced, a 29-point drop versus approximately 17 points for frontier models), making it the mechanistic anchor for why the cliff exists. The remaining eight models extend coverage strategically: GPT-5 and GPT-5 Mini enable within-family scaling comparison; DeepSeek V4 Pro nearly matches DeepSeek R1 on FAC (61.11%) while losing 4 points on F1, isolating the contribution of chain-of-thought reasoning; Claude Opus 4.7 anchors domain-level failure with the lowest Transport Phenomena score (26.7%); Qwen 2.5 72B and Gemma 3 27B characterise error transitions across open-weight tiers; Mathstral 7B and WizardMath 7B validate that math pretraining alone is insufficient for engineering reasoning. Confirmed failure counts are reported in Table 16.

T Inter-Annotator Agreement

T.1 Computed Metrics

Three annotators independently labeled all 2,200 traces in the error analysis sample, hereafter referred to as Annotators A, B, and C, assigning exactly one of six nominal error categories per trace (Appendix R). Rather than partitioning annotators by branch, we opted for full overlap and had them review every trace independently, maximising statistical power and enabling direct computation of multi-rater coefficients without imputation. We report percent agreement and Cohen's κ as baselines; given the uneven class distribution (§ K.2), we include PABAK (Byrt et al., 1993) and Gwet's AC1 (Gwet, 2008) as prevalence-robust alternatives.

AC1 is used over AC2 because our taxonomy defines unordered nominal categories with no inherent distance, making ordinal weighting inappropriate. For multi-rater agreement we report Fleiss' κ and Krippendorff's α (nominal variant), which are mathematically equivalent for three raters with complete nominal data (Zapf et al., 2016). All coefficients are interpreted on the Landis and Koch (1977) scale.

T.2 Agreement Results

Agreement across the 2,200-trace sample is moderate-to-substantial: overall percent agreement reaches 71.97%, Fleiss' $\kappa = 0.524$, and Gwet's AC1 $= 0.682$. Fleiss' κ and Krippendorff's α are identical for all models, as expected for three raters with complete nominal data. Full per-model results are reported in Table 17.

Frontier Models and Prevalence Imbalance. Across models, the gap between κ and the prevalence-robust metrics scales directly with the dominance of any single category in the sample. For frontier models, Calculation Errors account for 76–82% of all labels (Gemini 3.1 Pro: 76.2%, DeepSeek V4 Pro: 79.2%, Claude Opus 4.7: 81.7%), collapsing Fleiss' κ to 0.18–0.19 despite percent agreement of 67–74% and AC1 values of 0.64–0.72. PABAK and AC1 are therefore the appropriate reliability indicators for these models and consistently indicate moderate-to-substantial agreement. The inverse holds for weaker models, where more balanced distributions across Categories 1–5 yield higher κ values that better reflect true rater agreement.

Per-model Agreement and the Prevalence Gradient. Agreement quality follows a clear gradient inversely correlated with Calculation Error prevalence. Models with more balanced distributions yield the highest κ values: Gemma 3 27B ($\kappa_F = 0.672$, AC1 $= 0.733$), Mathstral 7B ($\kappa_F = 0.661$, AC1 $= 0.709$), and Llama 3.1 70B ($\kappa_F = 0.657$, AC1 $= 0.741$), with Calculation Error rates of 47%, 37%, and 51% respectively. WizardMath 7B is the extreme case: the only model where Calculation Error is not the plurality category, with Setup/Assumption Error dominating at 33.5% and Hallucination at 21.2%. The resulting label heterogeneity forces annotators to adjudicate among multiple plausible upstream failure points with no arithmetic evidence to anchor the judgment, yielding the lowest agreement in the set ($\kappa_F = 0.527$, AC1 $= 0.585$).

Primary Disagreement Locus. Disagreements concentrate at the boundary between Calculation Error (Category 6) and the conceptual categories Setup/Assumption (Category 2) and Formula/Principle Error (Category 3). In the A–C pair, 212 and 153 traces annotated as Category 6 by Annotator A were reassigned to Categories 2 and 3 by Annotator C; the A–B pair shows the same boundary at smaller magnitude. The pattern is directional, not random: Annotator C systematically locates the root cause at the upstream framing stage, reflecting a stricter application of the Q2 decision step. This ambiguity is structurally inherent: deciding between Category 6 and Categories 2–3 requires determining whether the upstream framing or the arithmetic execution is the primary failure, a judgment that demands comparing the model's physical reasoning against the gold solution rather than checking arithmetic alone.

U Extended Error Analysis

This appendix extends the error analysis of § 5.4 to all eleven evaluated models and provides aggregate breakdowns by difficulty tier and engineering branch.

U.1 Cross-Model Error Profiles

Table 18 reports the full error distribution for every model. The *conceptual error rate* is the sum of Hallucination, Setup/Assumption, and Formula/Principle percentages. The *complexity cliff* is the percentage-point change in conceptual error rate from Easy to Advanced difficulty; positive values indicate that conceptual errors increase with difficulty.

Frontier Models. All five frontier models keep conceptual errors below 25%, with Calculation Errors dominating failures. These models have largely solved the *knowledge* component of engineering reasoning; remaining errors are execution failures that tool augmentation (e.g., symbolic calculators) could directly address. Within this tier, DeepSeek R1 concentrates conceptual failures in Setup/Assumption errors (21.5%), indicating a systematic problem-framing weakness, while DeepSeek V4 Pro produces zero Formula/Principle errors, reflecting near-complete physical-law recall. Notably, three frontier models (DeepSeek V4 Pro, DeepSeek R1, GPT-5) exhibit negative complexity cliffs, confirming that their conceptual reasoning does not degrade with problem difficulty.

Mid-tier Models. All four mid-tier models show complexity cliffs of +18 pp or steeper, driven by surging Formula/Principle errors at Advanced difficulty. These models pattern-match successfully on Easy problems but lack the domain knowledge to sustain correct reasoning as complexity grows. GPT-5 Mini and DeepSeek R1 illustrate this sharply: both share the same overall conceptual rate (24.5%), yet GPT-5 Mini's cliff of +18.2 pp reveals conceptual breakdown under difficulty scaling, while DeepSeek R1's cliff of −4.1 pp reflects stable conceptual reasoning across all levels, with its Advanced accuracy drop stemming from harder arithmetic rather than knowledge gaps. Two models with identical aggregate error rates thus represent fundamentally different failure modes, a distinction aggregate benchmarks cannot surface.

Small and Specialist Models. Mathstral 7B and WizardMath 7B invert the error balance entirely, with conceptual errors (58.0% and 69.5%) outnumbering Calculation Errors, yet through opposite failure modes. Mathstral 7B is dominated by Formula/Principle errors (30.5%): it executes mathematical steps correctly but selects the wrong governing equation, having internalised mathematical manipulation without learning which physical laws apply. WizardMath 7B fails earlier, with Hallucination (23.0%) and Setup/Assumption errors (33.5%) indicating it fabricates physical quantities and misframes problems before computation begins. Together, these models show that mathematical pretraining can improve numerical execution without building the physical grounding that engineering reasoning requires.

U.2 Aggregate Difficulty and Branch Breakdown

Across difficulty tiers, difficulty selectively exposes higher-order knowledge gaps rather than uniformly degrading all error types. Formula/Principle errors nearly triple from Easy (6.5%) to Advanced (18.8%), showing that Advanced problems test whether a model knows *which* governing equation applies rather than whether it can execute it. Setup/Assumption errors rise similarly (14.5% to 19.8%), while Unit/Dimensional errors peak at Intermediate (4.4%) and nearly vanish at Advanced (0.2%), as the challenge shifts from unit bookkeeping to deeper physical reasoning. The difficulty gradient thus acts as a diagnostic probe: Easy problems filter arithmetic skill, Advanced problems filter physical understanding.

Across engineering branches, the arithmetic bottleneck is domain-invariant (Cal. errors: 61.6% to 64.1%), but each branch exposes a distinct conceptual weakness. Chemical Engineering's broad equation space produces the highest Hallucination (8.2%) and Formula/Principle (13.6%) rates, as models frequently select or fabricate the wrong governing law. Mechanical Engineering concentrates failures in Sign/Direction errors (3.5%), where correct force and stress orientation is required, while Electrical Engineering leads in Setup/Assumption errors (16.1%), reflecting failure at the problem-framing stage. Each branch therefore requires a different remediation: broader physical-law grounding for Chemical, sign-convention discipline for Mechanical, and stronger problem-parsing for Electrical Engineering.

Template Metadata	Median	Votes	Model	Model Feedback Summary
adiabatic_flame_temperature `Chemical Engineering` *Thermodynamics*	2	**2/3**	`GPT-5`	Structure clear but temperatures too low; likely C_p polynomial or unit mismatch in energy balance.
			`Claude Opus 4.5`	Method sound but values physically inaccurate (e.g., Acetylene ∼1493K vs ∼2500K); suspects data parameter issues.
			`Gemini 3 Pro`	Critical unit error identified in heat capacity integration (incorrect application of gas constant R).
impulse_response_from_lccde `Electrical Engineering` *Signals & Systems*	3	**2/3**	`GPT-5`	Derivation correct but minor notation inconsistencies (e.g., `d[0]` vs. δ, mixed float precision) reduce clarity.
			`Claude Opus 4.5`	Significant mathematical errors: inconsistent C_2 formatting, calculation error in $h[1]$ (should be $-1-(-4)=3$, not -5), and inconsistent sign convention in characteristic equation leading to incorrect roots.
			`Gemini 3 Pro`	Bug in string formatting lambda `fmt` ignores truncation argument, displaying raw floats with excessive precision and producing inconsistent output.
multi_segment_rod `Mechanical Engineering` *Mechanics of Materials*	3	**2/3**	`GPT-5`	Equations correct but ignores yield limits, resulting in unrealistically high stresses.
			`Claude Opus 4.5`	Physics generally sound, but questions the validity of applying large tensile loads to concrete; mathematical framework and pedagogical presentation remain excellent.
			`Gemini 3 Pro`	Identifies physically impossible scenarios where brittle materials (e.g., concrete in Instance 2) are subjected to tensile loads far exceeding fracture limits.
newtons_law_shear_stress `Chemical Engineering` *Transport Phenomena*	4	**1/3**	`GPT-5`	Equations and computations correct and clearly explained, but random parameter ranges can yield high Reynolds numbers where a linear laminar profile is physically unrealistic.
			`Claude Opus 4.5`	Correctly applies Newton's Law of Viscosity for Couette flow with accurate calculations; minor floating-point formatting artifact (`0.009300...01 m`) and edge-case fluid choices (e.g., liquid nitrogen at 77 K) are unusual but not physically impossible.
			`Gemini 3 Pro`	Physically accurate Couette flow problems with correct calculations; solution string lacks proper float formatting for converted distance Y_m, producing distracting precision artifacts in generated text.

Table 5: **Qualitative Feedback from the AI Tribunal.** Representative examples of templates flagged during initial screening across three engineering branches and four domains. Each template was independently evaluated by three AI models on physical plausibility, mathematical correctness, and pedagogical clarity (scored 1–5); votes reflect the number of models raising a review flag. `Critical Failures` required consensus rejection ($\geq 2/3$ models flagged, or any median dimension score < 4); `Controversial templates` passed the critical threshold but exhibited notable inter-model score disagreement ($\sigma > 0.5$). All flagged templates were manually refactored until a 100% pass rate was achieved.

Domain	Templates	Fleiss' κ	Interpretation
Chem. Eng.	30	1.00	Perfect Agreement
Elec. Eng.	30	1.00	Perfect Agreement
Mech. Eng.	30	1.00	Perfect Agreement

Table 6: **Binary Admissibility Consensus (Fleiss' κ).** All 90 Stage 1 survivors were unanimously admitted by human reviewers.

	Physical Plausibility		Math. Correctness		Pedagogical Clarity	
Domain	α	**AC2**	α	**AC2**	α	**AC2**
Chem. Eng.	0.224	0.927	1.00	1.00	0.457	0.894
Elec. Eng.	0.372	0.926	1.00	1.00	0.332	0.834
Mech. Eng.	0.647	0.949	1.00	1.00	0.078	0.745
Global	0.437	0.934	1.00	1.00	0.269	0.824

Table 7: **Ordinal Quality Agreement (Krippendorff's α and Gwet's AC2), quadratic weights.** Both metrics use identical ordinal weighting for direct comparability. Mathematical Correctness achieves perfect agreement under both metrics, serving as a rubric-validity check. The divergence between α and AC2 on the subjective dimensions reflects range restriction in the post-filter score distribution, not genuine annotator disagreement (§ K.2).

Dimension	MAD	Interp.	Exact Agree.
Physical Plausibility	0.167	Strong	83.3%
Math. Correctness	0.156	Strong	84.4%
Pedagogical Clarity	0.233	Acceptable	76.7%
Overall	**0.174**	**Strong**	**79.9%**

Table 8: **AI-Human Scoring Consistency (MAD).** Mean Absolute Difference between AI Tribunal and human expert median scores across 90 certified templates on a 5-point scale. Interpretation bands: Strong (MAD < 0.20), Acceptable (0.20–0.50). Exact Agreement reports the percentage of pairwise comparisons with zero score difference. No comparison exceeded a one-point gap on any dimension.

Branch	Structure		Difficulty Level			Total
	Dom.	Areas	Easy	Int.	Adv.	
Chemical	3	7	13	9	8	**30**
Electrical	3	7	11	12	7	**30**
Mechanical	3	6	10	13	7	**30**
Total	**9**	**20**	**34**	**34**	**22**	**90**

Table 9: **Dataset Statistics by Branch.** The dataset features a perfectly balanced distribution of 30 templates per branch, with difficulty levels scaled to test both foundational knowledge and advanced synthesis.

Score	Rating	Description
5	Excellent	Flawless derivation; follows all physical laws and engineering principles.
4	Good	Correct reasoning with minor notation inconsistencies or suboptimal paths.
3	Fair	Generally sound logic; minor errors in parameter grounding or arithmetic.
2	Poor	Significant logical gaps or incorrect governing equations for the domain.
1	Deficient	Nonsensical, physically impossible, or complete hallucination of concepts.

Table 10: Expert rubric used for assessing the quality of the reasoning process.

Score	Rating	Description
5	Excellent	Numerically correct within tolerance including correct units.
4	Good	Correct value but missing or incorrectly formatted units.
3	Fair	Off by a small margin due to rounding; correct magnitude.
2	Poor	Numerically incorrect by a significant margin or order of magnitude.
1	Deficient	Entirely wrong or absent.

Table 11: Expert rubric for assessing final answer accuracy.

Technical Configuration	Alignment (ρ)
Tribunal ($\tau = 0.8$) + Numeric Step-Matching	**0.632**
Tribunal ($\tau = 0.95$) + Numeric Step-Matching	0.581
Tribunal ($\tau = 0.6$) + Numeric Step-Matching	0.525
No AI Tribunal; Strict Ground-Truth Matching	0.411
Tribunal + Step-Matching (Numbers ignored)	0.469
ROUGE-L / BERTScore (Standard NLP metrics)	0.325

Table 12: Ablation results showing the Spearman's rank correlation (ρ) between automated reasoning scores and human expert ratings. The data highlights the necessity of the AI Tribunal and numerical verification.

Model	Organization	Size	Backbone	Source
Frontier Proprietary LLMs				
GPT-5 Base	OpenAI	N/A	–	gpt-5-base-2025-08-07
GPT-5 Mini	OpenAI	N/A	–	gpt-5-mini-2025-08-07
GPT-4.1 Base	OpenAI	N/A	–	gpt-4.1-2025-04-14
GPT-4.1 Mini	OpenAI	N/A	–	gpt-4.1-mini-2025-04-14
Claude Opus 4.7	Anthropic	N/A	Claude 4	claude-opus-4-7
Claude Sonnet 4.5	Anthropic	N/A	–	claude-sonnet-4-5-20250929
Claude Sonnet 4	Anthropic	N/A	–	claude-sonnet-4-20250514
Claude Sonnet 3.7	Anthropic	N/A	–	claude-3-7-sonnet-20250219
Gemini 3.1 Pro	Google	N/A	Gemini 3	gemini-3.1-pro-preview
Gemini 3 Pro	Google	N/A	–	gemini-3-pro-001
Gemini 2.5 Pro	Google	N/A	–	gemini-2.5-pro-002
Gemini 2.5 Flash	Google	N/A	–	gemini-2.5-flash-002
DeepSeek V4 Pro	DeepSeek	1.6T (49B active)	DeepSeek V4	deepseek-v4-pro
DeepSeek V3	DeepSeek	671B (37B active)	–	deepseek-v3
DeepSeek R1	DeepSeek	671B (37B active)	–	deepseek-reasoner
General Purpose Open LLMs				
Llama 3.1 70B	Meta	70B	–	meta-llama/Llama-3.1-70B-Instruct
Llama 3.1 8B	Meta	8B	–	meta-llama/Llama-3.1-8B-Instruct
Qwen 2.5 72B	Alibaba Cloud	72B	–	Qwen/Qwen2.5-72B-Instruct
Qwen 2.5 14B	Alibaba Cloud	14B	–	Qwen/Qwen2.5-14B-Instruct
Qwen 2.5 7B	Alibaba Cloud	7B	–	Qwen/Qwen2.5-7B-Instruct
Qwen 3 8B	Alibaba Cloud	8B	–	Qwen/Qwen3-8B-Instruct
Gemma 3 27B	Google	27B	–	google/gemma-3-27b-it
Gemma 2 9B	Google	9B	–	google/gemma-2-9b-it
Math Enhanced LLMs				
Qwen2.5-Math	Qwen Team	7B	Qwen2.5-7B	Qwen/Qwen2.5-Math-7B-Instruct
Mathstral	Mistral AI	7B	Mistral-7B	mistralai/Mathstral-7B-v0.1
WizardMath	WizardLM Team	7B	Mistral-7B	WizardLMTeam/WizardMath-7B-V1.1
MetaMath	MetaMath Project	7B	Llemma-7B	meta-math/MetaMath-7B-V1.0

Table 13: Overview of the 27 models evaluated, grouped into (1) frontier proprietary, (2) general-purpose open, and (3) math-enhanced categories. For proprietary models, we list the internal or API model identifiers; for open models, we include the HuggingFace source.

#	Category	Definition
1	Hallucination	Model uses a value, constant, or equation that does not exist in any real engineering reference, it was fabricated.
2	Setup / Assumption Error	The problem is framed incorrectly before any equation is applied: wrong boundary condition, ignored stated constraint, or misidentified system configuration.
3	Formula / Principle Error	The problem setup is correct, but the wrong governing equation, law, or principle is selected.
4	Unit / Dimensional Error	The correct formula and setup are used, but dimensional consistency fails: a missing unit conversion, mixed unit systems, or a value substituted in the wrong unit.
5	Sign / Direction Error	The correct formula, units, and setup are used, but the wrong sign or physical direction is assigned to a quantity, violating the reference frame or process direction convention.
6	Calculation Error	Setup, governing equation, units, and sign are all correct, but arithmetic or algebraic execution produces the wrong numerical result.

Table 14: The **Six-Category Error Taxonomy** used for manual annotation of confirmed reasoning failures in EngTrace. Categories are ordered by severity: errors higher in the table are more fundamental and take precedence in annotation assignment.

Category	Domain	Triggering Span	Error Description
`Formula / Principle Error`	Reaction Kinetics	$n_{\text{NO,final}} = 4X \cdot n_{\text{NH}_3\text{,initial}}$	The model applies stoichiometric coefficients as direct multipliers on the conversion, writing $n_j = \nu_j X \cdot N_{A0}$ for each species. The correct mole balance normalises by the stoichiometric coefficient of the limiting reactant, giving $n_j = (\nu_j/\lvert\nu_A\rvert)X \cdot N_{A0}$. The formula is structurally wrong prior to any numerical substitution, inflating all product moles by a factor of 4.
`Unit / Dimensional Error`	Electro-magnetics & Waves	$\mathbf{u} \times \mathbf{B} = -6952.38\,\hat{x} - 5031.90\,\hat{y} - 4542.22\,\hat{z}$	The model explicitly acknowledges the milliTesla-to-Tesla conversion in the problem setup but substitutes the raw milliTesla field components (136.59, -143.11) into the cross product determinant without applying the 10^{-3} factor. The resulting cross product is three orders of magnitude too large; the dimensional inconsistency propagates through the force calculation.
`Sign / Direction Error`	Electro-magnetics & Waves	$F = \frac{1}{4\pi\epsilon_0\epsilon_r}\frac{\lvert q_1 q_2\rvert}{r^2}$	The model computes force magnitude using the absolute value of the charge product, discarding the sign information that determines the direction of the force vector. Because q_1 is negative and q_2 is positive, the interaction is attractive and $\mathbf{F}_{12}$ must point opposite to $\hat{r}$. By taking $\lvert q_1 q_2\rvert$ the model constructs a repulsive force vector, reversing the physical direction of the result while the magnitude calculation remains correct.

Table 15: **Representative Annotated Traces from the Error Analysis Sample.** Each example was selected on the basis of unanimous agreement across all three annotators. The Triggering Span reproduces the verbatim excerpt from the model's reasoning chain that directly evidences the assigned error category, as required by the annotation protocol described in Appendix R.

Model	Failures	Easy	Int.	Adv.
Gemini 3.1 Pro	337	75	76	49
DeepSeek R1	359	75	76	49
Llama 3.1 70B	587	76	75	49
GPT-5 Mini	373	75	76	49
GPT-5	388	77	77	46
DeepSeek V4 Pro	388	75	76	49
Claude Opus 4.7	380	75	76	49
Qwen 2.5 72B	526	75	76	49
Gemma 3 27B	571	75	76	49
Mathstral 7B	830	75	76	49
WizardMath 7B	969	75	76	49
Total	5,708	828	836	536

Table 16: **Stratified Sample Distribution Across Evaluated Models.** For each model, the Failures column reports the total number of confirmed reasoning failures available after applying the trace selection criteria (§ S.1), while the Easy, Intermediate, and Advanced columns show the number of traces actually sampled for annotation from each tier (200 traces per model total). All sampling used random seed 42.

Model	% Agree ↑	κ A–B ↑	κ A–C ↑	κ B–C ↑	PABAK ↑	AC1 ↑	κ_F ↑	Calculation%
Gemini 3.1 Pro	67.17	0.056	0.208	0.287	0.343	0.643	0.183	76.2
DeepSeek R1	69.67	0.485	0.333	0.408	0.393	0.663	0.390	67.8
Llama 3.1 70B	77.50	0.673	0.590	0.712	0.550	0.741	0.657	51.3
GPT-5 Mini	68.00	0.384	0.259	0.429	0.360	0.645	0.345	69.3
GPT-5	72.67	0.488	0.232	0.369	0.453	0.701	0.329	75.5
DeepSeek V4 Pro	71.00	0.183	0.125	0.299	0.420	0.684	0.188	79.2
Claude Opus 4.7	73.67	0.160	0.138	0.286	0.473	0.717	0.184	81.7
Qwen 2.5 72B	75.17	0.697	0.566	0.590	0.503	0.715	0.615	54.3
Gemma 3 27B	77.00	0.739	0.609	0.672	0.540	0.733	0.672	46.8
Mathstral 7B	75.17	0.652	0.614	0.720	0.503	0.709	0.661	37.2
WizardMath 7B	64.67	0.589	0.452	0.558	0.293	0.585	0.527	26.7*
Overall	**71.97**	**0.584**	**0.456**	**0.549**	**0.439**	**0.682**	**0.524**	**60.6**

Table 17: **Inter-Annotator Agreement Across All Eleven Models.** κ A–B, A–C, B–C are pairwise Cohen's κ; PABAK and AC1 are the overall (mean) prevalence-robust pairwise coefficients; κ_F is Fleiss' κ (equal to Krippendorff's α in all cases). Calculation% reports the proportion of Calculation Error labels pooled across all three annotators, serving as a proxy for category concentration; higher values are associated with deflated κ values. *WizardMath 7B's dominant category is Setup/Assumption Error (33.5%); the Calculation% column shows its Calculation Error share (26.7%) for comparability.

Model	Hal.	Set.	For.	Uni.	Sgn.	Cal.	Conc. Rate	Complexity Cliff
Frontier models								
Claude Opus 4.7	1.5	6.5	2.5	2.0	0.0	87.5	10.5	+13.0
DeepSeek V4 Pro	3.5	11.0	0.0	1.0	0.0	84.5	14.5	−1.6
Gemini 3.1 Pro†	3.0	13.5	1.0	3.0	0.0	79.5	17.5	+7.9
GPT-5	2.0	13.5	3.5	1.0	0.5	79.5	19.0	−4.2
DeepSeek R1†	1.5	21.5	1.5	0.5	3.0	72.0	24.5	−4.1
Mid-tier models								
GPT-5 Mini	4.0	13.5	7.0	2.0	0.5	73.0	24.5	+18.2
Qwen 2.5 72B	6.0	15.0	16.0	3.0	4.5	55.5	37.0	+22.4
Llama 3.1 70B†	3.0	11.0	27.0	6.5	2.5	50.0	41.0	+41.8
Gemma 3 27B	8.0	13.5	22.0	5.5	3.5	47.5	43.5	+40.1
Small / specialist models								
Mathstral 7B	13.0	14.5	30.5	6.5	0.0	35.5	58.0	+49.7
WizardMath 7B	23.0	33.5	13.0	2.5	0.0	28.0	69.5	+38.4

Table 18: **Per-model error category distribution (%)** across all eleven models (200 annotated failure traces per model). Hal. = Hallucination, Set. = Setup/Assumption, For. = Formula/Principle, Uni. = Unit/Dimensional, Sgn. = Sign/Direction, Cal. = Calculation Error. Conc. Rate = conceptual error rate (Hal. + Set. + For.). Complexity Cliff = percentage-point change in conceptual error rate from Easy to Advanced difficulty (positive = increases with difficulty). Models sorted by ascending Conc. Rate within each tier. † Analysed in § 5.4.