Do VLMs Reason Like Engineers? A Benchmark and a Stage-wise Evaluation

Anonymous EMNLP submission

Abstract

Vision-Language Models (VLMs) demonstrate strong performance on general multimodal reasoning benchmarks, yet their ability to perform engineering reasoning remains largely unexplored. Unlike general visual question answering, engineering problem solving requires interpreting technical diagrams, selecting governing physical principles, and maintaining physically consistent multi-step reasoning. These capabilities are increasingly important for AI systems used in engineering education, scientific assistance, and technical decision-making, where reasoning failures may produce physically invalid yet superficially plausible solutions. Existing benchmarks primarily evaluate final answers and provide limited assessment of intermediate reasoning processes. We introduce **EngVQA**, a multimodal benchmark for evaluating engineering reasoning across 5 engineering subjects containing 696 problems. We introduce an 8-stage automatic evaluation framework for assessing VLM-generated solutions. The framework independently evaluates each stage of the solution, enabling fine-grained analysis of reasoning failures. We benchmark multiple state-of-the-art open and closed source VLMs on our evaluation framework and demonstrate substantial limitations in current engineering reasoning capabilities. Human evaluation shows strong agreement with our automated framework, achieving a Pearson correlation of 0.975 and a mean absolute error of 0.67 on a 10-point grading scale. Our results highlight the importance of process-oriented evaluation for reliable assessment of multimodal engineering reasoning systems.

1 Introduction

Recent Vision-Language Models (VLMs) such as GPT-4V (OpenAI, 2023), and Gemini Pro (Gemini, 2025) have significantly advanced multimodal reasoning and visual understanding. However, existing benchmarks including VQA (Agrawal et al., 2016), GQA (Hudson and Manning, 2019), ScienceQA (Lu et al., 2022), and MathVista (Lu et al., 2024) primarily evaluate symbolic reasoning and final-answer correctness, providing limited assessment of physically grounded engineering reasoning involving technical diagrams, governing equations, and multi-stage analytical workflows.

Recent engineering-oriented benchmarks such as EngiBench (Zhou et al., 2026), EEE-Bench (Li et al., 2025), move toward engineering-oriented evaluation, yet important limitations remain. EngiBench primarily focuses on textual engineering reasoning and capability-level rubric evaluation without explicit stage-wise decomposition of intermediate reasoning stages. EEE-Bench emphasizes multimodal understanding in electrical and electronics engineering, but still largely evaluates final-answer correctness rather than structured engineering workflows. Recent reasoning-aware and process-oriented evaluation frameworks including G-Eval (Liu et al., 2023a), Prometheus (Kim et al., 2024), ProcessBench (Zheng et al., 2025), and Thinking-LLM-as-a-Judge (Saha et al., 2025) demonstrate that structured reasoning-aware evaluation improves automated assessment reliability.

To address these limitations, we introduce **EngVQA**, a benchmark of authentic engineering problems spanning 5 subjects: Fluid Mechanics, Heat and Mass Transfer, Dynamics, Mechanics of Materials, and Thermodynamics. The benchmark requires joint reasoning over technical diagrams, physical principles, symbolic derivations, and multi-step quantitative analysis.

Building on the benchmark, we propose **EngJudge**, an 8-stage process-oriented evaluation framework that decomposes engineering solutions into interpretable reasoning stages while modeling dependency-aware error propagation across interrelated reasoning stages. EngJudge independently evaluates localized reasoning stages, improving interpretability and reducing evaluator ambiguity

Benchmark	Engineering Reasoning	Multimodal Reasoning	Technical Diagrams	Process Evaluation	Stage-wise Evaluation	Physics Constraints
MMMU	Partial	Yes	Partial	No	No	No
SciBench	No	Limited	No	No	No	No
EngiBench	Yes	No	No	Partial	No	Yes
EEE-Bench	Yes	Yes	Yes	No	No	Partial
SeePhys	Partial	Yes	Yes	No	No	No
EngVQA + EngJudge	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Comparison of multimodal scientific and engineering reasoning benchmarks along dimensions of engineering realism, technical diagram understanding, process-level evaluation, and physics-aware reasoning constraints.

compared to holistic evaluation approaches.

To validate the reliability of EngJudge, we also conduct a human validation study with engineering students. Our findings show that the framework's automated scores align closely with human expert grading philosophies, demonstrating its potential as a reliable tool for structured, process-oriented evaluation. Table 1 summarizes the differences between existing benchmarks and our approach. Our contributions are as follows:

- We introduce **EngVQA**, a multimodal benchmark of 696 authentic engineering problems requiring reasoning over technical diagrams, physical principles, symbolic derivations, and multi-step quantitative analysis.
- We propose **EngJudge**, an 8-stage process-oriented evaluation framework that models dependency-aware reasoning failures and achieves strong agreement with expert human evaluators.
- We demonstrate that SOTA VLMs exhibit substantial weaknesses in engineering reasoning, particularly in diagram interpretation, equation selection, assumption validation, and physically consistent multi-stage analysis.

2 Related Work

Engineering and Scientific Reasoning Benchmarks Recent benchmarks have explored scientific and engineering reasoning in large language and vision-language models across diverse domains. General multimodal reasoning benchmarks such as MMMU (Yue et al., 2024) evaluate broad-domain visual reasoning across university-level subjects, while scientific reasoning datasets such as SciBench (Wang et al., 2024), ScienceQA (Lu et al., 2022), and MathVista (Lu et al., 2024) focus primarily on scientific problem solving, symbolic reasoning, and final-answer correctness. More recent engineering-oriented benchmarks have extended multimodal evaluation into STEM and applied engineering domains. EngiBench (Zhou et al., 2026) evaluates engineering-focused question answering tasks, while EEE-Bench (Li et al., 2025) introduces multimodal reasoning problems in electrical engineering involving circuit diagrams and technical schematics. SeePhys (Xiang et al., 2025) studies visually grounded physics reasoning, and CSVQA (Jian et al., 2025) explores multimodal STEM reasoning in educational settings. Existing benchmarks predominantly evaluate reasoning through final-answer correctness or holistic solution-level scoring.

Stage-wise Reasoning and Process Evaluation Recent work has shown that final-answer correctness alone is insufficient for evaluating reasoning quality in modern language and vision-language models (Lightman et al., 2023; Golovneva et al., 2023). As models increasingly generate long-form chain-of-thought reasoning, evaluating intermediate reasoning behavior has become important for understanding logical consistency, factual correctness, and reasoning reliability.

ROSCOE (Golovneva et al., 2023) introduces fine-grained metrics for evaluating generated reasoning traces across semantic and logical consistency dimensions. (Lightman et al., 2023) demonstrate that process supervision can improve mathematical reasoning by rewarding intermediate reasoning correctness rather than relying solely on final answers. ProcessBench (Zheng et al., 2025) studies process-level reasoning failures in mathematical settings through stage-wise verification of intermediate reasoning chains.

LLM-as-a-Judge and Structured Evaluation Recent work has increasingly explored the use of large language models as automated evaluators for reasoning and generation tasks (Zheng et al., 2023; Liu et al., 2023b; Kim et al., 2024). G-Eval (Liu

et al., 2023b) demonstrates that rubric-guided evaluation can improve the reliability and interpretability of LLM-based assessment, while Prometheus (Kim et al., 2024) explores fine-grained rubric-conditioned evaluation strategies for scalable automatic assessment. More recent frameworks investigate structured reasoning strategies for evaluation itself. Thinking-LLM-as-a-Judge (Saha et al., 2025) proposes planning-oriented judging strategies in which evaluators explicitly reason through structured evaluation plans before assigning scores.

3 EngVQA Benchmark

3.1 Benchmark Principles

We chose problems-solution pairs that align with EngVQA's benchmark design principles: (1) **Diagram-Grounded Analytical Reasoning:** Problems require extracting geometry, boundary conditions, force directions, flow structure, material interfaces, and spatial constraints directly from technical figures such as free-body diagrams, thermodynamic plots, flow schematics, stress distributions, and engineering layouts. (2) **Structured Multi-Stage Reasoning:** Solutions involve multiple interdependent reasoning stages including problem characterization, assumption formulation, visual interpretation, equation selection, symbolic derivation, algebraic computation, and physical validation. (3) **Physics-Constrained Engineering Workflows:** Problems require physically valid reasoning under domain-specific engineering constraints. Models must maintain consistency between visual interpretation, governing equations, simplifying assumptions, and final quantitative predictions throughout the solution process.

3.2 Benchmark Statistics

Table 2 summarizes the subject composition and reasoning diversity of EngVQA. The benchmark spans five foundational engineering subjects and contains 696 problems requiring multimodal reasoning over technical diagrams, governing equations, symbolic derivations, and physical constraints. Rather than concentrating on a small set of narrow problem templates, the benchmark includes diverse question distributions across different topics. The subject distribution reflects both breadth and diversity in engineering reasoning workflows. Dynamics and Mechanics of Materials emphasize free-body analysis, force interactions, and rigid-body reasoning, while Thermodynamics and Heat & Mass Transfer involve physically constrained energy-system analysis, property relationships, and transport phenomena. Fluid Mechanics problems additionally require spatial reasoning over flow structures, pressure distributions, and conservation laws. Across all subjects, technical diagrams play a central role in downstream analytical formulation, making visual interpretation a necessary component of successful problem solving rather than an auxiliary context. The average number of topics per question measures reasoning density across the domains. This metric underscores that solving a typical problem in our benchmark requires the simultaneous integration and synthesis of multiple physical concepts, which fundamentally heightens the difficulty of problems. Full list of topics per subject in Appendix B.

4 EngJudge: Stage-wise Evaluation

Evaluating engineering reasoning requires more than final-answer correctness because failures often emerge within intermediate reasoning stages such as assumption formulation, diagram interpretation, equation selection, and computation. To capture this process-level behavior, we develop **EngJudge**, a stage-wise evaluation framework motivated by an error analysis of `gemini-2.0-flash-exp` solutions across 3 subjects - Fluid Mechanics, Heat and Mass Transfer (HMT), and Mechanics of Materials (MOM).

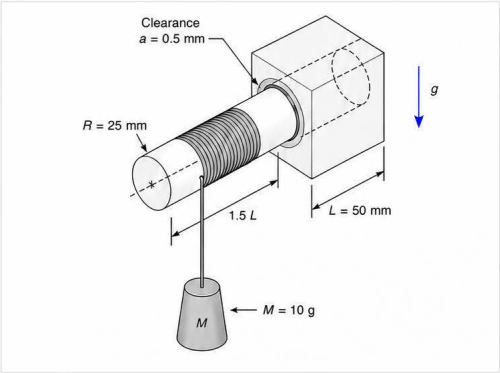
Our analysis shows that engineering reasoning failures are highly multifaceted and rarely occur in isolation. We additionally observe that visual interpretation forms a structurally distinct failure mode, while error correlations suggest that engineering reasoning is best modeled as a *partially dependent* process rather than a fully independent one (Appendix A). These findings motivate two key design choices in EngJudge: independent evaluation of localized reasoning stages and dependency propagation along empirically observed reasoning edges.

Figure 1 illustrates a failure case in a fluid mechanics problem. Here, a simple step-by-step average metric would penalize the integration step for having incorrect numerical coefficients, failing to recognize that the math was internally coherent but corrupted by prior errors. By decoupling these stages and tracking the failure chain (Assumptions $\rightarrow$ Equation Selection $\rightarrow$ Algebraic Accuracy $\rightarrow$ Final Answer), EngJudge

Subject	#Qs	ATPQ	Representative Topics (Total)	Diagram Types
Dyn.	171	3.57	Rigid-body motion, impact, vibration (54)	FBDs, kinematic schematics
Thermo.	236	9.28	Cycles, entropy, steady-flow systems (53)	Property plots, thermodynamic systems
FM	93	5.81	Pipe flow, hydrostatics, drag/lift (59)	Flow schematics, pressure diagrams
HMT	96	13.89	Conduction, convection, diffusion (60)	Thermo circuits, boundary-layer diagrams
MoM	100	9.46	Stress analysis, torsion, buckling (60)	Beam loading, stress distributions
Total	**696**		Multimodal engineering reasoning problems	

Table 2: Subject-wise statistics of EngVQA. ATPQ (Average Topics per question), Fluid Mechanics (FM), Heat and Mass Transfer (HMT), Mechanics of Materials (MoM), Thermodynamics (Thermo), and Dynamics (Dyn).

Question: A circular aluminum shaft mounted in a lubricated journal bearing is driven by a hanging mass. Develop the governing differential equation for the shaft angular velocity and determine the maximum angular speed and the time required to reach 95% of the steady-state response.



VLM solution: The model assumed shaft mass negligible, set driving torque $\tau = MgR$ directly (ignoring cord tension reduction), and retrieved $\mu = 0.17\ \mathrm{N \cdot s/m^2}$. It derived a ODE of the correct form and integrated it correctly, concluding $\omega_{\max} = 13.72$ rad/s and $t_{95} = 0.584$ s.

Failure propagation

Assumptions	Shaft incorrectly treated as mass-less, removing I_{shaft} from the system inertia term.
Equation setup	Driving torque simplified to MgR without modeling reduced cord tension during acceleration.
Property retrieval	Incorrect viscosity retrieved for SAE 10W-30 oil, corrupting damping coefficient B.
Propagation	Errors independently corrupted both ODE constants (B and C), producing incorrect transient dynamics despite correct symbolic integration.
Final answer	$\omega_{\max}$ and t_{95} numerically incorrect, while intermediate reasoning appeared internally coherent.

Figure 1: A representative example problem with LLM generated solution showing error propagation from incorrect assumptions affects the rest of the solution.

pinpoints the root cause of failure in physical modeling rather than mathematical execution.

4.1 Evaluation Framework

We introduce a multi-stage, penalty-based automatic evaluation framework, **EngJudge**, for grading LLM-generated solutions to graduate-level engineering problems. Rather than relying solely on holistic or single-score evaluation (Zheng et al., 2023; Liu et al., 2023b), our framework decomposes each solution into eight structured reasoning stages and evaluates them individually through an LLM-as-a-judge. The framework combines **penalty-based stage scoring**, **graph-based dependency propagation**, **meta-evaluation checks**, and **exam-style partial credit** (Mertler, 2001) to produce a single interpretable score on a 0–10 scale. The full pipeline is illustrated in Figure 2. The evaluation pipeline consists of the following stages:

Solution Parsing: We prompt the LLM to generate solutions using a fixed 8-stage reasoning structure designed to separate different components of engineering problem-solving, consistent with chain-of-thought prompting conventions (Wei et al., 2022; Kojima et al., 2022). Each stage is enclosed within explicit tags so that the generated solution can be parsed automatically (See Figure 2A).

```
###### STAGE_NAME ######
[stage content]
###### END_STAGE ######
```

A rule-based parser extracts each tagged block and produces an ordered list of reasoning stages. The required stages are: Problem Characterization, Assumptions, Visual Interpretation, Equation Selection, Logical Reasoning, Algebraic Accuracy, Physical Interpretation, and Final Answer.

Stage-wise Evaluation: Each reasoning stage is evaluated independently using a dedicated LLM-as-a-judge prompt (see Appendix F) tailored to

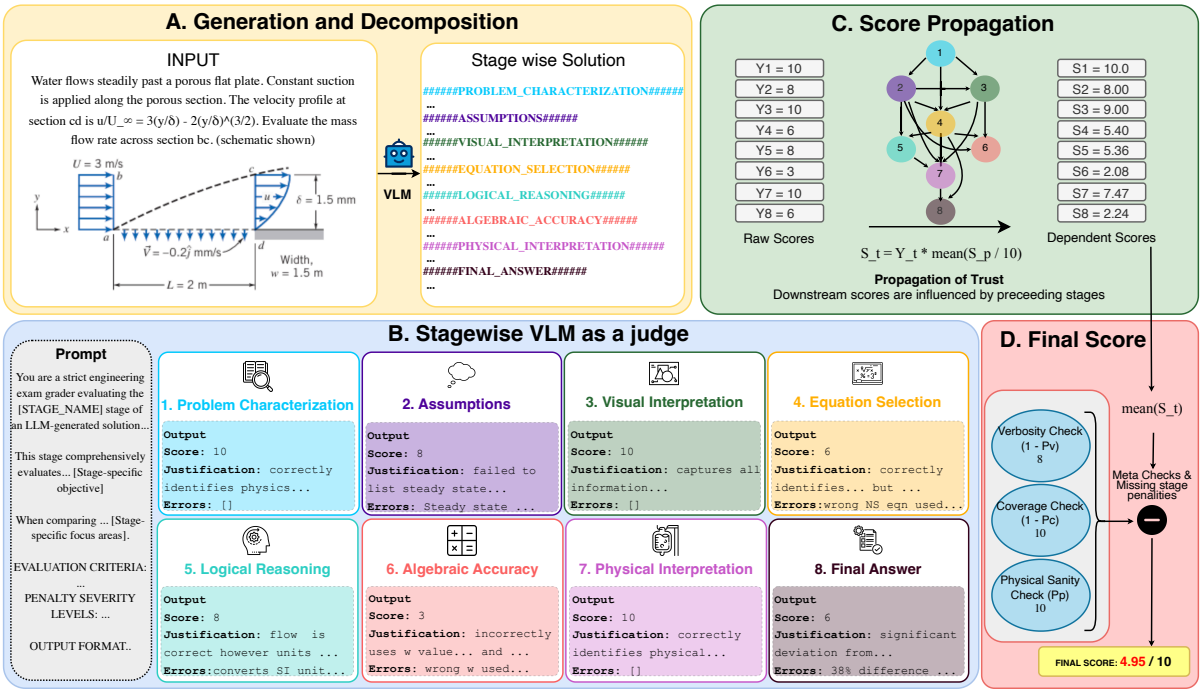


Figure 2: Overview of the proposed EngJudge evaluation framework. A. VLM generates a structured step-wise solution for an engineering problem containing both text and technical diagram. B. Solution is decomposed into eight reasoning stages, each independently evaluated using rubric-guided LLM-as-a-judge prompts (App F.3) with penalty-based scoring and fatal-error detection. C. Stage scores are then propagated through a dependency graph (colors correspond to steps in B.) and D. aggregated using meta-evaluation checks to produce a final interpretable score on a 0–10 scale.

that stage. Every stage begins with a score of 10, and penalties are deducted for detected reasoning errors. The penalties are distributed into four main categories shown in (Table 3).

Severity	Penalty	Examples
MINOR	2	Small rounding differences
MODERATE	4	Unit conversion errors
MAJOR	7	Wrong boundary conditions
CRITICAL	10	Dimensional inconsistency

Table 3: Four-level penalty severity scale applied uniformly across all evaluation stages.

The raw stage score (Y_t) is computed as:

$$Y_t = \max(0,\ 10 - \textstyle\sum_i p_i) \qquad (1)$$

where p_i is the penalty assigned to the ith identified error in stage t.

Fatal Errors: Certain critical mistakes cap the maximum score for a stage regardless of the accumulated penalties. For example, if the governing equation is fundamentally incorrect, the `Equation Selection` stage immediately receives a score of 0. Similarly, dimensionally inconsistent formulations or physically impossible results apply fixed score caps (See Appendix C.2).

For the `Final Answer` stage, predicted numerical values are compared against the ground-truth solution using relative error. Errors within 5% receive no penalty, errors between 5% and 10% receive a moderate penalty, and errors above 10% receive a major penalty. Missing units, incomplete answers, or physically impossible results (e.g., negative absolute temperature) receive additional penalties or score caps.

Graph-Based Dependency Propagation: After computing the stage-wise scores, we propagate upstream reasoning failures through a dependency graph. This design is motivated by the error analysis discussed in Section 4, which showed that many engineering reasoning errors propagate across stages.

We model these relationships using the dependency graph shown in Figure 2C. Each stage score is adjusted using the scores of its parent stages. Let Y_t denote the raw score of stage t obtained from the penalty-based evaluation. Let $P(t)$ denote the set of parent stages connected to stage t, and let N_t

be the number of parent stages ($N_t = \text{len}(P(t))$). Let S_t and S_p denote the final propagated scores of stage t and its parent stage(s) $p \in P(t)$, respectively. For stages with no parents i.e. Stage 1, we set $S_1 = Y_1$. For all other stages, the dependency score is computed as:

$$S_t = Y_t \times \frac{1}{N_t} \sum_{p \in P(t)} \frac{S_p}{10} \tag{2}$$

Score Aggregation and Final Penalty Gates: The final score is computed from the dependency-aware stage scores obtained after propagation. Let S_t denote the propagated score for stage t, and let N denote the number of reasoning stages. The base score is computed as:

$$S_{\text{base}} = \frac{1}{N} \sum_{t=1}^{N} S_t \tag{3}$$

Missing reasoning stages receive fixed deductions:

$$S_{\text{blend}} = \max(0,\ S_{\text{base}} - \text{MP}) \tag{4}$$

where MP denotes the total missing-stage penalty. The blended score is then adjusted using three meta-evaluation checks applied to the complete solution:

- `VERBOSITY` penalizes excessive repetition, filler reasoning, and unnecessarily long solutions that do not contribute meaningful engineering reasoning (Zheng et al., 2023).
- `COVERAGE` checks whether all requested quantities, sub-parts, assumptions, and reasoning steps are addressed in the final solution.
- `SANITY_FAIL` detects physically impossible or internally inconsistent outputs, such as negative absolute temperature, invalid units, or violations of conservation laws.

The resulting penalties are applied multiplicatively during final score aggregation. Final score is calculated as:

$$\begin{aligned} S_{\text{final}} = & S_{\text{blend}} \cdot (1 - \texttt{COVERAGE}) \\ & \cdot (1 - \texttt{VERBOSITY}) \cdot 0.9^{\texttt{SANITY_FAIL}} \end{aligned} \tag{5}$$

5 Experiments

5.1 Benchmark Construction Pipeline

Question PDFs are processed using a semi-automated extraction pipeline combining Adobe PDF Extract API[1], PyMuPDF[2], and Gemini-3.1-Flash-Lite to recover layouts, figures, equations, and symbolic notation while preserving multimodal structure. All extracted questions, diagrams, metadata, and solutions are manually verified against the reference material.

5.2 Experimental Settings

We evaluate two vision-language models as solution generators: `Qwen3-VL-8B` and `Gemini-2.5-Flash`, representing compact open-weight and proprietary frontier models, respectively. We used Chain-of-Thought (CoT) prompting (Wei et al., 2022) to trace step-by-step physical and mathematical reasoning (see Appendix F). We use `Gemini-3.1-Pro-Preview` as the primary evaluator due to its strong multimodal reasoning capabilities, and additionally employ `Qwen3-VL-32B-Instruct` to assess cross-model agreement and reduce potential self-evaluation bias.

Baseline Evaluator Setup To quantify the benefits of EngJudge's structured design, we implement two single-pass baselines. In both baseline configurations, the evaluator receives the entire problem statement, the VLM's solution, and the ground-truth reference, and predicts scores for all eight reasoning stages in a single LLM call (see Appendix F.2). The first baseline (*SinglePass*) averages these raw stage scores (Y_t in Equation 1) directly. The second baseline (*SinglePass + DP*) applies our Directed Acyclic Graph dependency propagation (DP) rules to obtain dependency scores (S_t in Equation 2) before averaging.

5.3 Results

The cross-evaluator performance comparison across the five engineering subjects is summarized in Figure 3. The quantitative results clearly indicate that `Gemini-2.5-Flash` significantly outperforms `Qwen3-VL-8B` across all evaluated subjects. This performance gap is consistent across all subjects. Both models struggle severely in Fluid Mechanics and Heat & Mass Transfer, which represent the lowest-scoring areas. This difficulty arises from the complex multi-phase physics, differential boundary conditions, and spatial geometry integration required in these fields. Conversely, performance

[1] https://developer.adobe.com/document-services/apis/pdf-extract/

[2] https://pymupdf.readthedocs.io/en/latest/

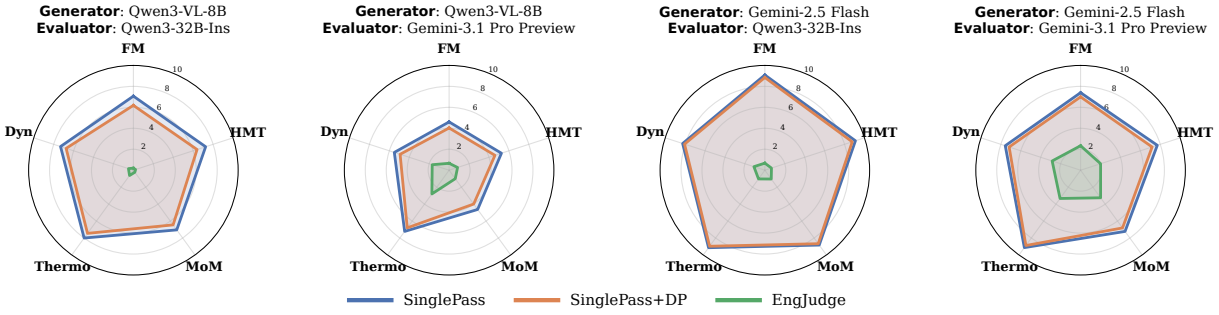


Figure 3: Cross-evaluator comparison across the five engineering subjects in EngVQA. Each radar plot represents a generator–evaluator pair. SinglePass uses a single LLM call and averages stage-wise raw scores Y_t (Equation 1), while SinglePass+DP averages stage-wise dependency scores S_t(Equation 2). All scores range from 0–10.

is higher in Thermodynamics and Mechanics of Materials, where problems tend to rely more on algebraic path-following and discrete-state equations. Nonetheless, no generator model breaches a mean score of 4 in any subject under EngJudge, highlighting the extreme difficulty of engineering questions for current state-of-the-art VLMs.

We also observe a massive performance discrepancy between the baseline and the EngJudge evaluation framework. When evaluated by the `Gemini-3.1-Pro-Preview (SinglePass)` one-shot judge, `Gemini-2.5-Flash` receives a highly optimistic overall score of 8.001. However, when subjected to the structured, dependency-aware grading of **EngJudge**, its score drops to 2.869. This behavior is driven by the fact that standard one-shot LLMs suffer from a leniency bias, often overlooking mathematical errors, missing justifications, or misinterpreting visuals if the final text appears superficially plausible.

We observe that the gap between **SinglePass** and **SinglePass + DP** is remarkably small and relatively constant (ranging from 0.11 to 0.65 points). This indicates that mathematical error propagation alone is insufficient to address the overestimation of LLM capabilities. Instead, the massive drop to the **EngJudge** curve (a gap of 2.5 to 4.1 points across all stages) is primarily driven by the framework's granular penalty rubrics, fatal error caps, and global meta-checks.

We also observe that the curves reveal a dichotomy between the *problem setup* and *execution*. During the initial setup stages, scores remain relatively high across all configurations (remaining above 5.4 even under EngJudge). However, once the model transitions to the execution stages - Logical Reasoning (`LR`), Algebraic Accuracy (`AA`), Physical Interpretation (`PI`), and the Final Answer (`FA`) - performance drops significantly. Under EngJudge, the score bottoms out at 2.74 for `AA`, indicating that algebraic calculation and sequential execution represent the *primary cognitive bottlenecks* for the generator model.

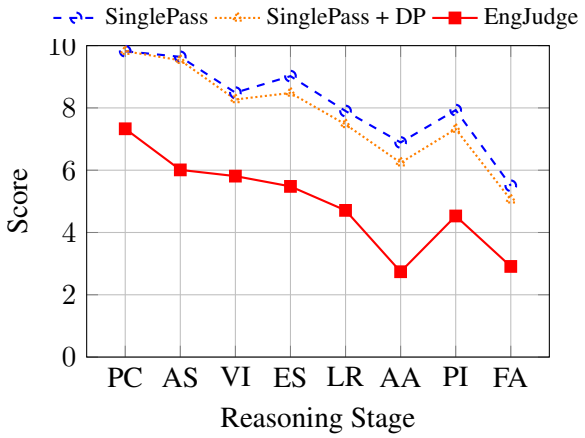


Figure 4: Average stage-wise scores across the eight stages evaluated with `Gemini 3.1 Pro Preview`. *PC: Problem Characterization, AS: Assumptions, VI: Visual Interpretation, ES: Equation Selection, LR: Logical Reasoning, AA: Algebraic Accuracy, PI: Physical Interpretation, FA: Final Answer.*

Component Ablation of EngJudge. Table 4 quantifies the contribution of each framework component. Removing dependency propagation based on our causal graph (*w/o DP*) substantially increases scores for both Gemini 2.5 Flash and Qwen3-VL-8B, showing that dependency-aware propagation is the primary source of grading rigor by penalizing downstream reasoning built on incorrect upstream formulations. Removing meta-evaluation checks further increases scores, indicating frequent violations of coverage and verbosity constraints in generated solutions. In contrast, removing missing-stage penalties has minimal im-

pact on Gemini, suggesting strong adherence to the required structured reasoning format.

Generator Variant	Qwen3-VL-8B	Gemini 2.5 Flash
EngJudge (Full Framework)	**1.67**	**2.87**
EngJudge w/o DP	2.48	4.12
EngJudge w/o Meta Eval Checks	2.35	4.75
EngJudge w/o Missing Stage Penalty	2.01	2.96

Table 4: Component ablation of EngJudge using `Gemini 3.1 Pro Preview` as the evaluator.

We also conduct an empirical analysis of stage-wise score correlations using the Pearson correlation matrix of raw scores across all stages (Figure 5). We observe strong correlations between adjacent reasoning stages, notably between Algebraic Accuracy (AA) and the Final Answer (FA) ($r = 0.69$), and between Equation Selection (ES) and Logical Reasoning (LR) ($r = 0.53$).

Conversely, correlations between early conceptual stages and final calculations are negligible (e.g., Problem Characterization (PC) vs. Final Answer yields $r = 0.06$). These strong, localized correlations reflect the physical reality of error propagation, empirically validating our dependency graph design: errors in prerequisite stages cascade to compromise downstream execution. Detailed analysis of these correlations and their relationship in the dependency graph is provided in Appendix D.3.2.

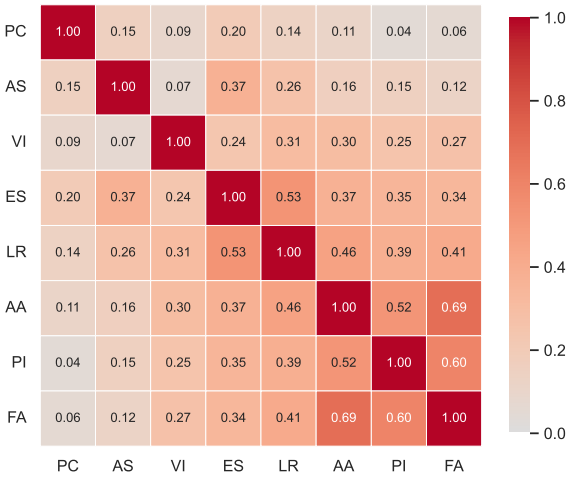


Figure 5: Pearson correlation matrix between stage-wise raw scores.

6 Human Study for Validating EngJudge

To rigorously validate the automated evaluations produced by the EngJudge (Sec 4), we conducted a blinded human study to assess the alignment between EngJudge's structured scoring and human expert intuition. The evaluation panel consisted of 9 undergraduate/graduate students possessing strong backgrounds and practical experience in the chosen engineering subjects.

Evaluators were presented with randomized engineering problems, generated solutions from `Gemini 2.5 flash`, and EngJudge assigned scores for each stage through a web interface (see Appendix Figs. 14-21). They rated EngJudge's stage-wise and overall scores on 4 randomly selected questions. EngJudge demonstrated exceptional alignment with human judgment, achieving a Pearson correlation coefficient of **0.975** and a Mean Absolute Error (MAE) of just 0.67 (on a 10-point scale). When assessing EngJudge's overall score, 66.7% of the evaluators rated the overall system as "Good" or "Very Good" and 22.3% rated it as average.

In another setting, we asked human evaluators to compare EngJudge's dependency-aware score against a naive average, obtained by simply averaging the scores assigned to the eight stages in a blind A/B format. Evaluators explicitly favored the dependency score in 66.7% of cases. This strongly validates our hypothesis that engineering reasoning evaluation requires hierarchical, cascading penalization rather than isolated stage-wise grading. A comprehensive breakdown of the methodology, graphical visualizations, and granular stage-level metrics is provided in Appendix E.

7 Conclusion

In this paper, we introduced EngVQA, a challenging benchmark of graduate-level engineering physics problems spanning five core engineering disciplines. Second, we introduce EngJudge, an evaluation framework that uses dependency-aware score propagation to capture compounding physical and mathematical errors in engineering reasoning. Evaluation of SOTA VLMs on EngJudge highlights reasoning collapses due to visual-geometric and algebraic failures. EngJudge shows excellent alignment with human expert judgment, and its dependency-aware framework is preferred over averaging. We believe EngJudge can be a robust diagnostic tool to guide the development of physically grounded multimodal reasoning agents for engineering applications and interactive tutoring systems.

Limitations

Data Contamination. We cannot fully rule out benchmark contamination from model pretraining data. However, the benchmark emphasizes multimodal understanding and multi-step engineering reasoning, making direct memorization alone insufficient.

Computational Cost of Evaluation. Our framework requires 11 LLM judge calls per solution, making evaluation significantly more computationally expensive. Although KV caching can reduce redundant computation (Pope et al., 2022), evaluation cost may still limit scalability in large-scale or resource-constrained settings.

LLM Judge Reliability and Bias. EngJudge inherits known limitations of LLM-as-a-judge evaluation, including potential biases and imperfect alignment with human judgment. Although human validation demonstrates strong agreement, such biases cannot be fully eliminated (Huang et al., 2025).

Ethics Statement

This work complies with ethical guidelines concerning data sourcing, human studies, and computational deployment. The evaluation dataset comprises standard academic engineering physics problems. We conducted the human study with volunteer graduate engineering students; participation was entirely voluntary, all data was fully anonymized, and no PII or demographic data was collected. Prior to the study, participants were presented with explicit instructions explaining how their evaluation data would be used, and they provided informed consent before proceeding. Furthermore, all evaluated models were accessed in accordance with their developer terms of service and license agreements, using local configurations on consumer-grade hardware to minimize computational energy consumption and environmental footprint.

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. Vqa: Visual question answering. *Preprint*, arXiv:1505.00468.

Team Gemini. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *International Conference on Learning Representations*.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2025. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, Vienna, Austria. Association for Computational Linguistics.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702.

Ai Jian, Weijie Qiu, Xiaokun Wang, Peiyu Wang, Yunzhuo Hao, Jiangbo Pei, Yichen Wei, Yi Peng, and Xuchen Song. 2025. Csvqa: A chinese multimodal benchmark for evaluating stem reasoning capabilities of vlms. *Preprint*, arXiv:2505.24120.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. *Preprint*, arXiv:2310.08491.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Ming Li, Jike Zhong, Tianle Chen, Yuxiang Lai, and Konstantinos Psounis. 2025. Eee-bench: A comprehensive multimodal electrical and electronics engineering benchmark. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13337–13349.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*, abs/2305.20050.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on*

empirical methods in natural language processing, pages 2511–2522.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *Preprint*, arXiv:2310.02255.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Craig A. Mertler. 2001. Designing scoring rubrics for your classroom. *Practical Assessment, Research, and Evaluation*, 7(25).

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2022. Efficiently scaling transformer inference. *Preprint*, arXiv:2211.05102.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. 2025. Learning to plan reason for evaluation with thinking-llm-as-a-judge. *Preprint*, arXiv:2501.18099.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024. Scibench: evaluating college-level scientific problem-solving abilities of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Kun Xiang, Heng Li, Terry Jingchen Zhang, Yinya Huang, Zirong Liu, Peixin Qu, Jixi He, Jiaqi Chen, Yu-Jie Yuan, Jianhua Han, Hang Xu, Hanhui Li, Mrinmaya Sachan, and Xiaodan Liang. 2025. Seephys: Does seeing help thinking? – benchmarking vision-based physics reasoning. *Preprint*, arXiv:2505.19099.

Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Processbench: Identifying process errors in mathematical reasoning. *Preprint*, arXiv:2412.06559.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Xiyuan Zhou, Xinlei Wang, Yirui He, Yang Wu, Ruixi Zou, Yuheng Cheng, Yulu Xie, Wenxuan Liu, Huan Zhao, Yan Xu, Jinjin Gu, and Junhua Zhao. 2026. Engibench: A benchmark for evaluating large language models on engineering problem solving. *Preprint*, arXiv:2509.17677.

A Error Analysis

Before designing the evaluation framework, we conducted a systematic pilot error analysis across three engineering subjects: Fluid Mechanics, Heat and Mass Transfer (HMT), and Mechanics of Materials (MOM). Rather than imposing evaluation steps heuristically, we first analyzed how VLMs fail on engineering reasoning problems and whether those failures exhibit consistent structural patterns across domains.

For each pilot subject, a frontier VLM (`Gemini-2.0-flash-exp`) was prompted to solve a subset of the problems. A separate LLM judge then identified and categorized all reasoning errors relative to the ground-truth solutions, producing subject-specific error taxonomies grounded in engineering domain knowledge. The errors and their categories were manually inspected as well.

Unless otherwise noted, figures visualize representative pilot subjects; analogous trends were consistently observed across all other subjects.

Key Finding 1: Errors Are Multi-Faceted and Co-occurring. Across all three pilot subjects, arithmetic-related failures were among the most frequent observed error categories, occurring 49 times in Fluid Mechanics, 45 times in HMT, and 80 times in MOM (Figure 6). However, the secondary distribution of errors varied substantially across domains.

Fluid Mechanics failures frequently involved wrong formula usage (Bernoulli, continuity, Navier-Stokes: 31 occurrences) and boundary condition errors (31 occurrences). HMT failures showed a high rate of wrong or unjustified assumptions (31 occurrences), particularly misidentificaion of steady-state vs. transient regime. MOM failures were dominated by stress-formula misuse(44 occurrences), free-body setup errors, and geometric misinterpretation(36 occurrences).

These results indicate that engineering reasoning failures are not well-described by a single generic failure mode. Instead, different engineering domains exhibit distinct reasoning error signatures, motivating domain-aware and step-level evaluation.

Key Finding 2: Errors Cluster into Distinct Reasoning Steps. When we grouped the subject-specific error types by the reasoning operation they implicate, a consistent structure emerged across all three subjects shown in (Table 5).

Stage	Representative Error Types
Problem Charact.	Ignoring part of question; poor decomposition
Assumptions	Wrong/unjustified assumptions; steady vs. transient failure; lumped capacitance misuse
Visual Interp.	Geometric misinterpretation; incorrect FBD or shear force diagram; boundary condition misread from figure
Equation Sel.	Fourier/Fick misapplication; wrong Bernoulli form; formula outside validity domain; wrong stress relation
Logical Reas.	Inconsistency between steps; incorrect sub-result dependency
Algebraic Acc.	Arithmetic errors; unit mismatch; incorrect dimensionless parameter definition
Physical Interp.	Physical principle misconception; material property misuse; incorrect property evaluation temperature
Final Answer	Numerical error vs. ground truth; physically impossible result; partial answer

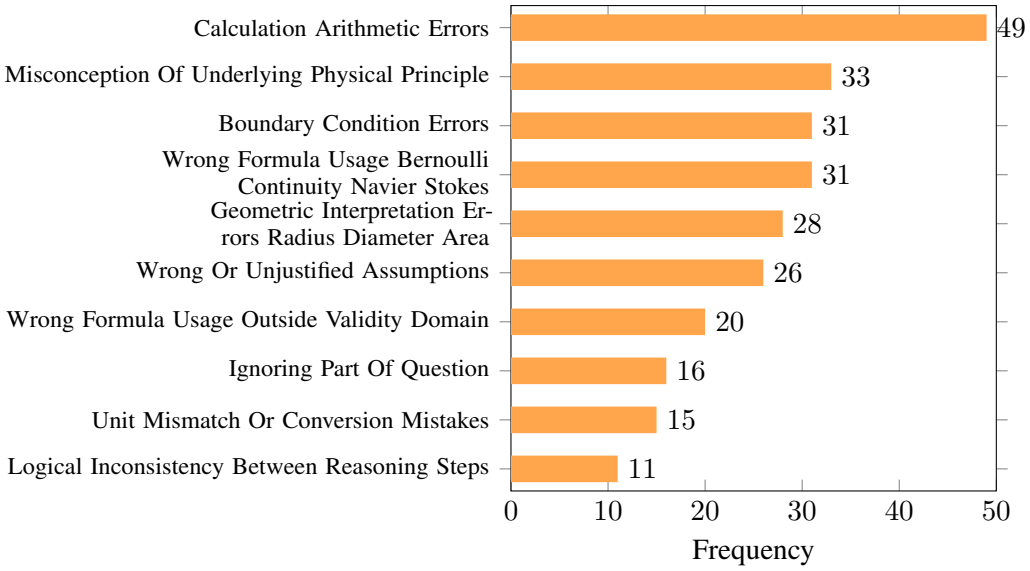
Table 5: Mapping of observed error types to evaluation steps. Each step captures a qualitatively distinct failure mode not detectable by inspecting other steps.

Key Finding 3: Engineering Errors Exhibit Sparse Structured Dependencies. Figure 7 shows error correlation matrices from the pilot analysis. The matrices reveal that engineering reasoning failures are not fully independent. Several dependencies consistently emerge across errors in different reasoning stages.

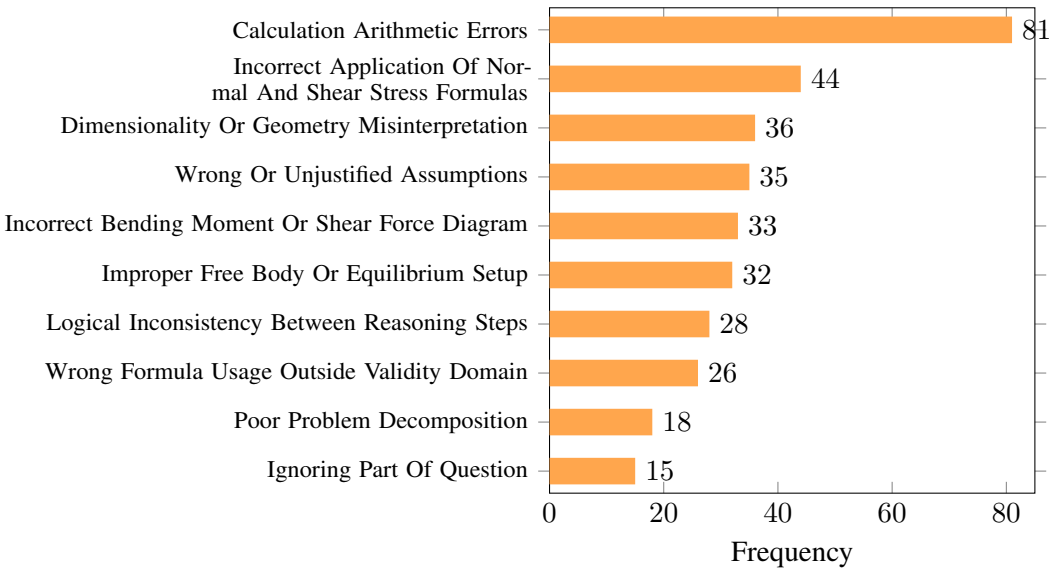
The strongest and most consistent dependency appears between equation misuse and arithmetic errors ($r = 0.55$–0.75 across pilot subjects), reflecting cascading failures where incorrect governing equations propagate into downstream computation. Additional moderate dependencies were observed between visual interpretation and equation selection in diagram-centric problems, as well as between assumption formulation and downstream logical reasoning.

This sparse dependency structure has a direct methodological implication: engineering reasoning should not be evaluated using a single score, it also cannot be fully factorized into completely independent steps. Instead, evaluation must combine step-level assessment with selective dependency-aware propagation along empirically justified reasoning edges.

Key Finding 4: Visual Reasoning Is a Distinct and Persistent Failure Mode. We computed a *visual error susceptibility score* for each topic within each subject—defined as the fraction of errors due to misinterpretation of the diagram rather than to purely textual or computational reasoning (Figure 8). Several topics scored above 0.55: *hydrostatic pressure distributions* and *buoyancy* in Fluid Mechanics (0.56 each); *thermal resistance networks* in HMT (0.57); and *stress resultants and equilibrium*, *torsion*, and *bending* in MOM (0.59,



(a) Fluid Mechanics



(b) Mechanics of Materials

Figure 6: Most frequent error categories across representative engineering domains. Arithmetic-related failures are consistently common, while secondary error distributions remain domain-specific. Heat and Mass Transfer exhibited similar trends.

0.55, 0.55). This led us to concluding that visual misinterpretation is a structurally independent failure mode requiring its own evaluation step.

From Error Analysis to Evaluation Design. The pilot error analysis directly motivated the design of our evaluation framework. Frequent propagation from equation mistakes to arithmetic errors motivated the use of *dependency propagation*, where upstream mistakes reduce the score of downstream steps. Similarly, the large number of diagram-related failures motivated a dedicated *Visual Interpretation* step, while the different reasoning patterns across subjects motivated the use of *domain-specific prompts*.

The resulting DAG structure (Figure 9) follows the natural flow of an engineering solution. *Problem Characterization* and *Assumptions* appear first because they affect all later steps. *Visual Interpretation* feeds into *Equation Selection*, since correct equations often depend on properly reading diagrams and boundary conditions. *Equation Selec-*

Figure 7: Error correlation matrix for Fluid Mechanics. Most off-diagonal entries are near zero, confirming step independence. The exception, formula misapplication and arithmetic errors ($r = 0.75$), reflects a cascading dependency that directly motivates the propagation edge in the DAG.

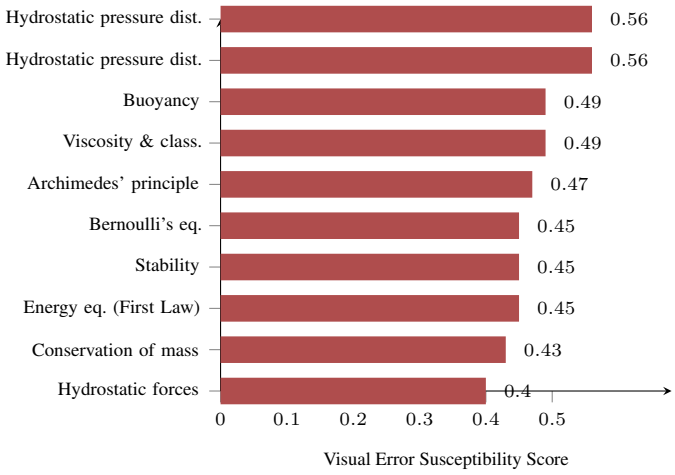


Figure 8: Topics with highest visual error susceptibility scores (Fluid Mechanics). Scores above 0.5 indicate that the majority of errors in that topic originate from diagram misinterpretation, confirming visual reasoning as a structurally distinct failure mode.

tion and *Logical Reasoning* influence *Algebraic Accuracy*, because calculations based on incorrect equations or inconsistent logic usually produce incorrect numerical results. *Physical Interpretation* runs alongside symbolic computation and contributes to the *Final Answer*, ensuring that the final result is also physically meaningful.

The error correlation matrix (Figure 7) provides more insight into this propagation of errors.

This cascade pattern is precisely what the dependency propagation formula captures: a correct downstream step built on a flawed upstream step should not receive full credit, because the apparent correctness is contingent on an invalid foundation.

B Benchmark details

To ensure broad curriculum-level coverage and reduce over-specialization toward narrow reasoning patterns, EngVQA was curated across five major engineering domains spanning dynamics, thermodynamics, transport phenomena, structural mechanics, and physics-constrained analytical reasoning. The benchmark intentionally includes both foundational engineering principles and advanced multi-topic analytical problems commonly encountered in undergraduate and graduate-level engineering curricula.

Many questions require simultaneous reasoning across multiple conceptual categories, including diagram interpretation, governing equation selection, physical constraint validation, symbolic derivation,

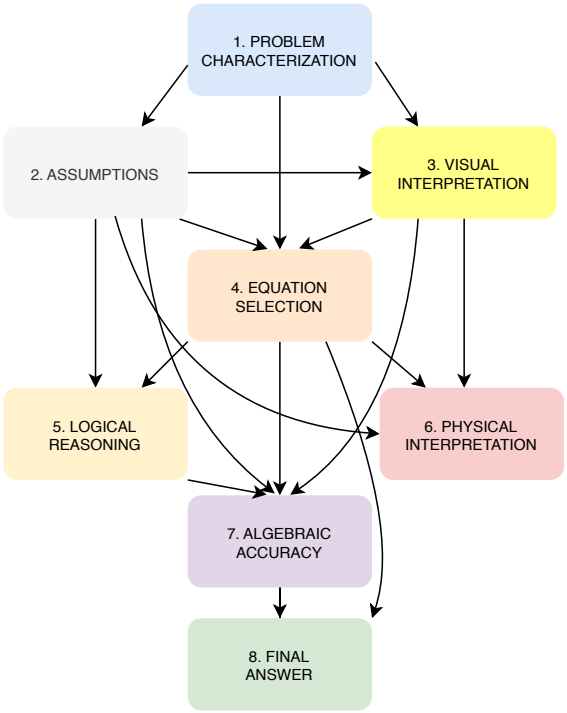


Figure 9: Dependency DAG structure used for trust propagation across reasoning steps.

and numerical computation. Rather than evaluating isolated single-step calculations, the benchmark emphasizes interconnected engineering workflows involving coupled physical processes, system-level reasoning, and multi-stage analytical dependencies.

The benchmark additionally contains substantial cross-topic overlap within each subject area. For example, a single problem may simultaneously in-

volve visual interpretation, conservation laws, material properties, boundary conditions, and multi-stage quantitative reasoning. This structure was intentionally designed to evaluate robustness across heterogeneous engineering concepts, mathematical formulations, and physical settings.

The full list of representative topics covered within each engineering domain is summarized below.

Dynamics Topic Coverage

Particle Kinematics: Rectilinear motion with constant and variable acceleration; dependent motion of particles; projectile motion; normal–tangential coordinates; polar and cylindrical coordinates; relative motion; absolute and relative velocity/acceleration; curvilinear trajectories; coordinate transformations; rotating reference frames.
Particle Kinetics: Newton's second law; equations of motion in Cartesian, normal–tangential, and cylindrical coordinates; work–energy methods; conservative and non-conservative forces; impulse–momentum; angular impulse and momentum; impact analysis; central-force motion; power and efficiency.
Rigid-Body Dynamics: Plane motion; fixed-axis rotation; rolling constraints; instantaneous centers; rigid-body kinetics; mass moment of inertia; radius of gyration; gyroscopic motion; free and damped vibration; force and moment analysis; rigid-body impact; connected rigid-body systems; cable tension; planetary and orbital motion.

Thermodynamics Topic Coverage

Fundamental Thermodynamics: Open and closed systems; intensive and extensive properties; pressure and temperature; thermodynamic equilibrium; specific volume and density; work and heat transfer mechanisms; energy conservation; mechanical and thermal equilibrium; property relations; phase-change processes; saturated mixtures; superheated vapor; subcooled liquid; property diagrams and tables.
Energy and First-Law Analysis: Closed-system and control-volume energy balances; steady and unsteady flow; nozzles and diffusers; turbines and compressors; throttling valves; pumps and heat exchangers; polytropic processes; specific heats; ideal-gas and real-gas relations; compressibility effects; flow work and enthalpy.
Entropy and Exergy: Second-law analysis; entropy generation; isentropic processes; reversible and irreversible processes; exergy destruction; thermal efficiency; COP of refrigeration and heat-pump systems; availability and work potential; entropy balances.
Thermodynamic Cycles: Carnot cycle; Rankine cycle with regeneration and reheat; Brayton cycle and gas turbines; Otto and Diesel cycles; refrigeration cycles; combined cycles; jet propulsion systems.

Fluid Mechanics Topic Coverage

Fluid Statics and Properties: Hydrostatic pressure distributions; manometry; fluid properties; shear stress and viscosity; surface tension and capillarity; buoyancy and stability; atmospheric pressure variation; hydrostatic forces on submerged surfaces.
Fluid Dynamics and Conservation Laws: Conservation of mass and momentum; Bernoulli analysis; integral and differential control-volume analysis; continuity equations; energy equations; velocity and acceleration fields; vorticity and circulation; dimensional analysis; Buckingham-Π theorem; Reynolds, Mach, Froude, Weber, and Euler numbers.
Internal and External Flows: Laminar and turbulent pipe flow; entrance-region development; boundary layers; Reynolds-number effects; head losses and Moody diagram; velocity profiles; hydraulic jumps; flow separation; drag and lift; wake formation; control surfaces; pump and turbine performance; cavitation.
Compressible and Computational Flows: Compressible nozzle flows; isentropic flow; Fanno and Rayleigh flow; shock waves; potential flow theory; Navier–Stokes equations; Euler equations; CFD discretization and numerical convergence; flow visualization; transport phenomena and environmental flows.

Heat & Mass Transfer Topic Coverage

Heat Conduction: Fourier's law; steady-state and transient conduction; thermal conductivity; thermal resistance networks; lumped-capacitance analysis; plane-wall, cylindrical, and spherical conduction; composite walls; fins and extended surfaces; thermal contact resistance; two-dimensional conduction; conduction shape factors; boundary conditions in conduction.
Convection and Boundary Layers: Forced and natural convection; convection heat-transfer coefficients; velocity and thermal boundary layers; laminar and turbulent flow; Reynolds, Prandtl, Nusselt, Grashof, Rayleigh, and Biot numbers; internal and external flow convection; mixed convection; boiling and condensation; hydrodynamic and thermal entry lengths.
Thermal Radiation: Blackbody radiation; emissivity; Stefan–Boltzmann law; Planck's law; Wien's displacement law; radiation exchange; view factors; participating media radiation; surface energy balance.
Mass Transfer and Coupled Transport: Fick's law; mass diffusivity; species conservation; evaporation; Raoult's and Henry's laws; heat exchangers; LMTD and NTU methods; thermal energy generation and storage; thermoelectric effects; coupled heat and mass-transfer analogies.

Mechanics of Materials Topic Coverage

Stress and Deformation: Normal and shear stress; axial loading; stress–strain relations; Hooke's law; Poisson's ratio; thermal stresses; combined loading; pressure vessels; principal stresses; Mohr's circle; strain transformations; failure theories.
Structural Analysis: Beam bending; torsion of circular and thin-walled sections; free-body diagrams; equilibrium analysis; stress resultants; shear and bending moment diagrams; neutral axis; centroids and moments of inertia; section modulus; shear center; buckling and elastic stability.
Advanced Material Behavior: Stress concentration effects; composite beams and transformed sections; unsymmetric bending; energy methods and Castigliano theorem; deflection analysis; impact loading and strain energy; plastic bending; thin-walled structures; material yielding and fracture behavior.

B.1 Topic Generation and Assignment Methodology

To systematically classify the conceptual coverage of the benchmark, we established a structured vocabulary of topics for each engineering subject and mapped them to individual problems. This appendix details our topic extraction process, validation matching pipeline, and the resulting conceptual density statistics.

B.1.1 Master Topic List Synthesis

Rather than relying on manually predefined keywords, we utilized an LLM-driven topic synthesis pipeline to generate a realistic representation of university-level engineering curricula. For each subject, the text statements of all questions in the dataset were concatenated and fed into `Gemini 2.5 Flash`. The model was prompted to analyze the question corpus and synthesize a master vocabulary of 50–60 generalized topics per domain (e.g., 53 for Thermodynamics and 54 for Dynamics). These synthesized topics were structured to match standard course syllabi.

B.1.2 Multimodal Topic Assignment and Validation

With the master topic list established, we ran a multimodal classification pipeline to map topics to each problem. For each question:

1. The model was provided with the question statement, its master topic list, and the corresponding question diagram.

2. The model selected all topics necessary to formulate or solve the problem.

3. To ensure data integrity and prevent hallucinated labels, the model's output was processed by an automated validation script. The validator matched each output string against the master topic list using case-insensitive transformations and fuzzy string matching (with a similarity cutoff threshold of 0.85). Topics failing this validation check were discarded.

B.1.3 Conceptual Density and Subject Variations

The average number of topics assigned per question varies significantly by subject:

- **Dynamics** (3.57 **topics/Q):** Emphasizes modular problem-solving frameworks where questions are typically isolated to specific kinematic or kinetic formulations (e.g., pure projectile motion or pure impulse-momentum analysis).
- **Fluid Mechanics** (5.81 **topics/Q):** Highlights moderately coupled phenomena such as pressure fields under hydrostatic conditions or head loss calculations in pipe networks.
- **Thermodynamics** (9.28 **topics/Q) and Mechanics of Materials** (9.46 **topics/Q):** Require concurrent evaluation of material properties, load conditions, geometric invariants, and thermodynamic state variables.
- **Heat & Mass Transfer** (13.89 **topics/Q):** Exhibits the highest conceptual coupling because standard heat transfer scenarios inherently involve multiple parallel transport phenomena (conduction, convection, and radiation occurring simultaneously with thermal boundaries and resistance properties).

This multi-concept mapping highlights that solving a typical problem in our benchmark requires the simultaneous integration and synthesis of multiple physical principles, making the benchmark fundamentally more challenging than standard single-template datasets.

C Detailed Evaluation Framework

This appendix provides the detailed mathematical and algorithmic specifications for the **EngJudge** evaluation framework, expanding on the summary in Section 4.1.

C.1 Penalty Scales

The automated evaluation framework utilizes a penalty-based grading scheme designed to mirror the grading practices of human professors in graduate-level engineering courses. Rather than evaluating a solution in a binary (correct/incorrect) fashion, each of the eight mandatory reasoning stages begins with a base score of 10. The automated grader identifies specific errors within the stage, classifies them into one of four severity levels, and subtracts the corresponding penalty points.

The four severity levels consist of : **Minor** (2 points), **Moderate** (4 points), **Major** (7 points), and **Critical** (10 points). The spacing of this scale (2, 4, 7, 10) is designed to ensure that minor arithmetic or formatting slips do not excessively penalize a student's score, while severe conceptual errors or invalid physical formulations are heavily penalized, often resulting in an immediate zero for that stage. This configuration provides a flexible yet robust mechanism for partial credit distribution.

Table 6 summarizes the criteria and provides concrete illustrative examples from the evaluation of the benchmark questions.

At the end of the human study, evaluators were asked to rate the calibration of the penalty scheme (the specific deductions of 2, 4, 7, and 10 points for minor, moderate, major, and critical errors). The feedback from the 9 expert participants was distributed as follows:

- **Well Calibrated: 55.6%** (5 out of 9 evaluators) expressed direct agreement that the penalty levels are well calibrated.
- **Minors Too Harsh: 22.2%** (2 out of 9 evaluators) felt that minor error penalties (2 points) were slightly too harsh.
- **Scale Too Coarse: 11.1%** (1 out of 9 evaluators) suggested a more granular scale.
- **Not Harsh Enough: 11.1%** (1 out of 9 evaluators) indicated that the penalties could be even stricter.

C.2 Fatal Error Capping Mechanism

To prevent VLMs from receiving high scores on steps containing fundamental conceptual breakdowns, the EngJudge framework incorporates a *Fatal Error Capping* mechanism. During the evaluation of a reasoning step, if the evaluator model detects any of the fatal error types defined in Table 7, the score for that specific step is capped at the designated value, regardless of any correct algebraic manipulations or formatting compliance. If multiple fatal errors are triggered within a single step, the step score is capped at the *lowest* applicable value.

The physical rationales and definition for each fatal error cap are detailed below:

Penalty Level	Points	Grading Criteria & Description	Illustrative Benchmark Example
Minor	2	Insignificant arithmetic or rounding slips, notation issues, or minor dimension misreads with negligible downstream effects.	In a forced convection chip cooling problem (Heat & Mass Transfer, Problem 71), using a slightly rounded air velocity (30.8 m/s converted to 31.29 m/s) in the Reynolds number calculation, which does not alter the analytical method.
Moderate	4	Unit conversion errors, sub-optimal coordinate system selection, or missing an important visual boundary condition.	In a piston-cylinder compression problem (Thermodynamics, Problem 5-131), failing to explicitly state the "large surroundings" assumption needed to simplify the radiation heat transfer terms.
Major	7	Formulating boundary conditions incorrectly, using equations outside their valid limits, or severe derivation slips.	In a reheat-regenerative Rankine cycle analysis (Thermodynamics, Problem 10-107), misidentifying state points and extraction pressures for a closed feedwater heater, leading to incorrect enthalpy lookups.
Critical	10	Chosing fundamentally wrong governing equations, dimensionally inconsistent boundary conditions, or physically impossible results.	In a pipe exit plug fluid momentum analysis (Fluid Mechanics, Problem 11), applying Bernoulli's equation to calculate a non-zero static pressure at the exit of an unconfined jet discharging to atmosphere.

Table 6: Detailed penalty scales, grading criteria, and concrete failure examples across the benchmark domains.

- **Governing Equation Incorrect (Score Cap = 3):** This error is triggered when the VLM selects a fundamental governing equation that is physically inapplicable to the problem domain. A prime example is applying Bernoulli's equation (which assumes frictionless, inviscid flow) to a high-viscosity pipe flow system where shear stress dominates, or using linear momentum equations in a system where rotational torque governs. Because selecting the incorrect governing relation invalidates the entire physical foundation of the solution, the step is capped at a low score of 3, reflecting a failure in basic physical reasoning.

- **Physical Model Invalid (Score Cap = 4):** This error occurs when the VLM formulates a physical representation or simplification that directly contradicts the physical realities of the problem. Examples include assuming a transient process is at steady-state, treating a highly compressible gas as an incompressible fluid, or modeling a multi-dimensional flux as a one-dimensional flow without justification. While the model may choose the correct general equations, the invalid simplification renders the subsequent modeling physically incorrect, justifying a cap of 4.

- **Dimensionally Inconsistent (Score Cap = 2):** This represents the most severe algebraic and physical violation in engineering. It is triggered when the model writes an equation where the units or physical dimensions of the left-hand side do not match those of the right-hand side (e.g., adding a force term directly to a velocity term, or equating mass flow rate to pressure). Because dimensional consistency is the most fundamental sanity check in physical sciences, any dimensional violation indicates a complete collapse of mathematical and physical consistency, resulting in the strictest cap of 2.

- **Boundary Conditions Invalid (Score Cap = 4):** In graduate-level engineering physics, solving governing differential equations (such as the Navier-Stokes or heat diffusion equations) depends entirely on specifying boundary and initial conditions. This error is triggered when the model misapplies interface conditions—such as setting a free-slip condition at a stationary wall, neglecting convective heat transfer at a boundary, or using absolute pressure in momentum balances where gauge pressure is required. Since incorrect boundary conditions yield physically impossible solutions, the score is capped at 4.

By enforcing these limits, the framework ensures that a solution is not graded solely on surface-level correctness or formatting compliance, but rather on its conceptual and physical integrity.

Fatal Error Type	Score Cap
Governing equation incorrect	3
Physical model invalid	4
Dimensionally inconsistent	2
Boundary conditions invalid	4

Table 7: Fatal error caps. The step score is capped at the lowest applicable value when a fatal error is detected.

C.3 Meta-Evaluation Checks and Deductions

C.3.1 Missing Stage Penalty (MP)

To ensure that models do not bypass intermediate cognitive stages (e.g., jumping directly to a final answer without stating assumptions or selecting equations), the rule-based parser scans the output for the eight required markup tags.

- **Deduction Rule:** For every missing reasoning stage, a flat penalty of 2.0 points is added to the total missing penalty (MP):

$$MP = 2.0 \times |\{\text{absent required stages}\}| \quad (6)$$

  This penalty is subtracted directly from the base score(aggregated score from all the stages) before applying meta-evaluation multipliers: $S_{\text{blend}} = \max(0, S_{\text{base}} - MP)$.

Beyond step-level grading, EngJudge applies three solution-level *meta-evaluations*: Coverage, verbosity and physical sanity, to assess the overall quality, completeness, and physical grounding of the derivation. These checks are implemented via LLM-as-a-judge and evaluates the full solution and not stage-wise.

C.3.2 Coverage Penalty (COVERAGE)

The coverage check verifies whether the model has addressed all sub-questions (e.g., part (a) and part (b)) and calculated all requested engineering quantities.

- **Evaluation Rubric:** The evaluator model evaluates the solution on a scale of 0 to 10. A score of 10 indicates complete coverage. For each missing or unaddressed sub-question or target quantity, the score is reduced. This gives us the coverage score (CS)

- **Penalty Mapping:** The coverage score (CS) is mapped to a fractional penalty $\texttt{COVERAGE} \in [0,\ 0.5]$ using:

$$\texttt{COVERAGE} = \min\left(0.5,\ (10 - \texttt{CS}) \times 0.10\right) \quad (7)$$

  Each point below 10 deducts 10% from the final score, capped at a maximum deduction of 50%.

C.3.3 Verbosity Penalty (VERBOSITY)

Large Language Models frequently generate excessive, redundant steps or boilerplate explanations(Zheng et al., 2023). The verbosity check penalizes solutions that contain excessive filler text, unnecessary restatements of the problem, or repetitive derivations.

- **Evaluation Rubric:** The evaluator model rates the conciseness and technical directness of the solution from 0 to 10. A score of 10 represents a clean, direct derivation with minimal filler text. This gives us the verbosity score(VS).

- **Penalty Mapping:** The verbosity score (VS) maps to a fractional penalty $\texttt{VERBOSITY} \in [0,\ 0.5]$ using:

$$\texttt{VERBOSITY} = \min\left(0.5,\ (10 - \texttt{VS}) \times 0.10\right) \quad (8)$$

  Similar to coverage, each point of excessive verbosity deducts 10%, up to a maximum cap of 50%. We cap it at 0.5 as we don't want it to bring the overall final score down to zero, same goes for Coverage.

C.3.4 Physical Sanity Check (SANITY_FAIL)

This is a critical global filter checking for high-level reasoning and physical consistency. Even if a model's intermediate steps appear mathematically structured, the final numerical values might be physically impossible. This is different from the earlier fatal error caps, as this checks the final solution/answer.

- **Evaluation Rubric:** The evaluator model performs a binary check ($\texttt{SANITY_FAIL} \in \{0, 1\}$) on whether the solution violates fundamental physical laws. Examples of failure conditions include:
  - Violating the Second Law of Thermodynamics (e.g., heat engines exceeding Carnot efficiency).
  - Violating mass or energy conservation (e.g., fluid outflow exceeding inflow in steady state).
  - Generating impossible physical bounds (e.g., negative absolute temperatures in

Kelvin, efficiencies greater than 1.0, or velocities exceeding the speed of light).

- **Penalty Mapping:** If any physical sanity check fails, $\texttt{SANITY_FAIL} = 1$, which applies a global multiplicative factor of $0.9^{\texttt{SANITY_FAIL}} = 0.9$ (a flat 10% penalty) to the final score.

Final score is calculated as:

$$S_{\text{final}} = S_{\text{blend}} \cdot (1 - \texttt{COVERAGE}) \cdot (1 - \texttt{VERBOSITY}) \cdot 0.9^{\texttt{SANITY_FAIL}} \quad (9)$$

D Experiments

The table 8 contains all the scores obtained by using different generators, evaluators, and evaluation techniques.

D.1 Plot against Baseline

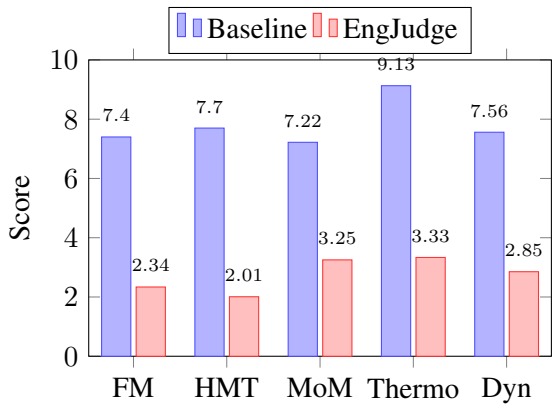


Figure 10: Comparison of baseline and EngJudge evaluation by `Gemini 3.1 Pro Preview`.

D.2 Step wise scores

Reasoning step	Qwen3-VL-8B		Gemini 2.5 Flash	
	Dep	Ind	Dep	Ind
Problem Characterization (PC)	7.13	7.13	7.33	7.33
Assumptions (AS)	4.42	5.85	6.01	7.93
Visual Interpretation (VI)	4.47	7.49	5.81	8.72
Equation Selection (ES)	2.85	4.40	5.48	8.14
Logical Reasoning (LR)	2.17	4.11	4.71	7.48
Algebraic Accuracy (AA)	1.38	2.30	2.74	4.34
Physical Interpretation (PI)	2.47	4.96	4.53	7.51
Final Answer (FA)	1.47	3.41	2.91	5.74

Table 9: Data for average step-wise dependent (Dep) and Independent (Ind) scores, evaluated with `Gemini 3.1 Pro Preview`

D.3 Detailed Topological and Correlation Analyses

We provide extended quantitative details on the choice of dependency structures and the empirical correlations between reasoning steps.

D.3.1 Influence of Dependency Topologies

To evaluate the choice of dependency mapping, we compare the proposed Directed Acyclic Graph against alternative topological structures in Table 10. Excluding dependency propagation entirely (*Independent/Flat*) yields optimistic base scores of 7.19 (Gemini) and 5.18 (Qwen). Conversely, imposing a sequential chain (*Strict Linear Chain*), where each step N depends entirely on step $N-1$, results in overly severe penalties (4.29 for Gemini, 2.93 for Qwen). This is because a linear topology assumes that a minor mistake in an early independent step (e.g., a small assumption slip) completely invalidates parallel, unaffected tracks (such as reading dimensions from a diagram). Our proposed *True DAG* models these parallel pathways accurately, penalizing dependent steps only when their specific prerequisites fail, resulting in balanced base scores of 5.03 (Gemini) and 3.48 (Qwen).

Topology Type	Qwen3-VL-8B	Gemini 2.5 Flash
True DAG	3.48	**5.03**
Strict Linear Chain	2.93	4.29
Independent (Flat)	**5.18**	**7.19**

Table 10: Ablation showing the effect of different dependency structures on the average **base score** (excluding meta-evaluation checks and missing step penalties). Removing dependency propagation (Independent/Flat) yields substantially higher scores, while stricter dependency assumptions (Linear Chain) result in lower scores than the proposed DAG-based topology.

D.3.2 Correlation and Causal Dependency Analysis

To empirically validate the dependency relations modeled by our DAG, we analyze the Pearson correlation matrix between the raw step scores across all evaluations (shown in Figure 5 in the main paper). The matrix reveals a clear, diagonal-adjacent correlation structure, where strong correlations are localized to neighboring steps of the reasoning chain.

Specifically, we observe strong correlations between Algebraic Accuracy and the Final Answer ($r = 0.69$), and between Equation Selection and Logical Reasoning ($r = 0.53$). These findings reflect the physical reality of error propagation: a failure in selecting the correct governing equation or executing mathematical derivations directly compromises downstream calculations and the final numerical output. Conversely, correlations between

Generator	Evaluator	FM	HMT	MoM	Thermo	Dyn	Overall
Qwen3-VL-8B (CoT)	Qwen3-VL-32B-Ins (*Baseline*)	7.070	7.250	7.040	8.000	7.310	7.446
	Gemini-3.1 Pro Preview (*Baseline*)	4.610	5.250	4.630	7.220	5.500	5.752
	Qwen3-VL-32B-Ins (*Baseline with Dep*)	6.190	6.390	6.450	7.450	6.750	6.793
	Gemini-3.1 Pro Preview (*Baseline with Dep*)	4.030	4.600	3.990	6.800	4.920	5.203
	Qwen3-VL-32B-Ins (*EngJudge*)	0.199	0.219	0.239	0.623	0.495	0.416
	Gemini-3.1 Pro Preview (*EngJudge*)	0.661	0.847	1.000	2.798	1.697	1.670
Gemini-2.5-Flash (CoT)	Qwen3-VL-32B-Ins (*Baseline*)	9.090	9.090	8.830	9.140	8.250	8.855
	Gemini-3.1 Pro Preview (*Baseline*)	7.400	7.700	7.220	9.130	7.560	8.001
	Qwen3-VL-32B-Ins (*Baseline with Dep*)	8.850	8.770	8.670	8.970	8.060	8.646
	Gemini-3.1 Pro Preview (*Baseline with Dep*)	7.000	7.180	6.810	8.880	7.170	7.632
	Qwen3-VL-32B-Ins (*EngJudge*)	0.674	0.663	1.039	1.028	1.098	0.945
	Gemini-3.1 Pro Preview (*EngJudge*)	2.337	2.008	3.252	3.334	2.853	2.869

Table 8: Cross-evaluator performance comparison across engineering domains. Scores are reported on a 0–10 scale. Gray rows correspond to the proposed EngJudge framework. FM = Fluid Mechanics, HMT = Heat & Mass Transfer, MoM = Mechanics of Materials, Thermo = Thermodynamics, Dyn = Dynamics, and CoT = Chain-of-Thought (Wei et al., 2022).

early conceptual steps and final execution steps are negligible (e.g., Problem Characterization vs. Final Answer yields $r = 0.06$). This demonstrates that a VLM's ability to classify the physics domain of a problem is independent of its ability to execute the math required to solve it, reinforcing the need for step-wise evaluation.

Why Statistical Correlation Differs from Causal DAG Dependencies? This is a key design question: why are some pairs of steps with high statistical correlation in Figure 5 not connected by direct dependency edges in our DAG? For instance, Physical Interpretation (PI) and Final Answer (FA) exhibit a high correlation of $r = 0.60$, and Algebraic Accuracy (AA) and Physical Interpretation (PI) show $r = 0.52$, yet neither pair has a direct edge in the evaluation graph. This design choice is guided by three main principles:

1. **Causality vs. Correlation:** The DAG is designed to enforce direct, physical causal prerequisites. For example, a student can mathematically compute a correct final numerical answer (FA) via correct algebraic manipulation (AA) without necessarily understanding or explaining its physical meaning (PI). Because PI is not a strict mathematical prerequisite for calculating FA, drawing a dependency edge from PI to FA would be causally incorrect, despite their statistical correlation.

2. **Confounding by Downstream Position:** Later steps in the reasoning chain (such as AA, PI, and FA) are strongly correlated because they are co-dependent on the cumulative errors of early steps (like AS and ES). This shared dependency on common ancestors creates high statistical correlation (confounding) in the empirical data. Adding redundant edges between these downstream variables in the DAG would lead to duplicate penalization for the same upstream error.

3. **Conceptual Independence in Rubrics:** Conceptual reasoning (such as qualitative physical interpretation) and algebraic computation are graded as independent dimensions in standard engineering pedagogy. A model may fail the algebra but perform a correct physical limit check, or vice versa. The high empirical correlation ($r = 0.52$ between AA and PI) is a reflection of general model capability (i.e., stronger VLMs perform well on both, while weaker models fail on both) rather than a causal dependency between the two skills.

Thus, this correlation does not necessarily validate our dependency graph, but it does support the claims made in the graph to some extent.

D.4 Qualitative Case Studies

This section details the primary failure modes observed in the generated solutions, categorized by the nature of the cognitive or mathematical lapse. To illustrate these mechanisms, we present four distinct examples spanning four different engineering domains.

D.4.1 Visual Interpretation Errors

Despite demonstrating strong general visual recognition, vision-language models (VLMs) frequently

fail to map two-dimensional diagrams of three-dimensional systems into correct mathematical formulations, or overlook subtle geometric features that alter the boundary conditions.

- **Omission of Physical Constraints (Thermodynamics, Problem 5-131):** When modeling a piston-cylinder device under compression, the VLM failed to identify the physical stops inside the cylinder wall depicted in the schematic. Consequently, it modeled the entire compression process as isobaric under the assumption that the piston was free to move indefinitely. In reality, once the piston contacts the stops, the process transitions to a constant-volume phase, rendering the subsequent state and boundary work calculations invalid.

D.4.2 Physical Principles and Boundary Conditions Errors

A major source of low evaluation scores is the formulation of equations that violate the system's boundary conditions, flow characteristics, or conservation laws.

- **Incorrect Boundary Pressure Formulation (Fluid Mechanics, Problem 11):** In analyzing the force required to hold an exit nozzle plug in place, the model applied Bernoulli's equation to calculate a non-zero exit pressure ($p_2 \approx 364$ kPa). Because the nozzle discharges directly to the atmosphere as an unconfined jet, the correct physical boundary condition is a uniform gauge pressure of zero ($p_2 = 0$). Introducing a fictitious pressure force at the boundary led to a physically invalid linear momentum balance.

D.4.3 Dimensionless Group and Correlation Errors

In domains involving convective transport and non-dimensional scaling, models frequently apply empirical correlations outside their valid regimes or evaluate fluid properties at incorrect reference states.

- **Incorrect Reference Temperature Property Evaluations (Heat & Mass Transfer, Problem 71):** During the forced convection cooling analysis of an electronic chip, the model evaluated the physical properties of the air stream (density, dynamic viscosity, and thermal conductivity) at the free-stream temperature of 25°C (300 K) instead of the film temperature ($T_f = (T_s + T_\infty)/2 \approx 308$ K). This property mismatch introduced systematic errors into the Reynolds (Re) and Prandtl (Pr) numbers, which propagated through the Nusselt number correlation and yielded an incorrect convective heat transfer coefficient.

D.4.4 Arithmetic and Unit Discrepancies

Even when the physical formulation is sound, final numerical answers are often invalidated by basic calculation slips, incorrect coordinate transformations, or lookup errors.

- **Kinematic and Component Formulation Slips (Dynamics, Problem 21-77):** When calculating the angular momentum of a precessing satellite, the model equated the satellite's spin rate directly to the total angular velocity component along the symmetry axis ($\psi_z = \omega_s$), ignoring the precession contribution and the coordinate rotation. This algebraic process error resulted in a cascading numerical slip, underestimating the satellite's angular momentum by approximately 45%.

E Detailed Human Study Results

This appendix provides a detailed quantitative breakdown of the human validation study referenced in Section 6. The study utilized a custom-built, interactive evaluation dashboard that sequentially guided evaluators through randomly assigned problems per session. Participants first performed a binary rating (correct/incorrect) on each generated step without seeing the framework's evaluation, minimizing anchoring bias (Tversky and Kahneman, 1974). Subsequently, they were shown the framework's dependency score and penalty justifications, and were asked to rate the severity on a 5-point Likert scale (ranging from "Much too low" to "Much too high").

E.1 Data Point Formulation and Rating Quantification

To perform continuous statistical analysis on qualitative feedback, we define a structured mapping to convert the qualitative human ratings into numerical scores.

Data Point Definition A single evaluation data point represents the rating of a single reasoning unit (either a stage-level evaluation or a meta-evaluation check) for a given problem instance by a human participant. With 9 expert participants evaluating

4 randomized problems each, the total number of data points is:

$$N_{\text{total}} = N_{\text{step}} + N_{\text{meta}} = 285 + 108 = 393 \quad (10)$$

where $N_{\text{step}} = 285$ represents individual stage evaluations (8 stages per problem, with occasional missing steps or parsing fallbacks omitted), and $N_{\text{meta}} = 108$ represents meta-evaluation evaluations (3 checks per problem: Verbosity, Coverage, and Physical Sanity).

Likert-to-Numerical Mapping During the study, human experts did not assign raw scores directly; instead, they reviewed the framework's automated score ($S_{\text{auto}} \in [0, 10]$) and selected a qualitative rating on a 5-point Likert scale to express their level of agreement. To reconstruct the absolute human score (S_{human}), we define an adjustment mapping (δ) for each Likert category:

$$\begin{aligned} \text{“Much too low”} &\implies \delta = +2 \\ \text{“Slightly too low”} &\implies \delta = +1 \\ \text{“About right”} &\implies \delta = 0 \\ \text{“Slightly too high”} &\implies \delta = -1 \\ \text{“Much too high”} &\implies \delta = -2 \end{aligned} \quad (11)$$

The equivalent human score is then reconstructed by adjusting the automated score and clipping the result to the valid $[0, 10]$ grading scale:

$$S_{\text{human}} = \max(0, \min(10, S_{\text{auto}} + \delta)) \quad (12)$$

This formulation ensures a human-consistent numerical scale, where a rating of “About right” ($\delta = 0$) corresponds to perfect agreement ($S_{\text{human}} = S_{\text{auto}}$). Pearson correlation ($r$) and Mean Absolute Error (MAE) are then calculated directly using the paired arrays of S_{auto} and S_{human} across the 393 data points.

E.2 Continuous Alignment Metrics

The continuous alignment between the framework's automated scores and the synthetic human scores is visually depicted in Figure 11 and quantitatively summarized in Table 11.

E.3 Meta-Evaluation Breakdown

Beyond sequential steps, the framework assesses complete solutions through three meta-checks. The human alignment for these individual checks is highly consistent:

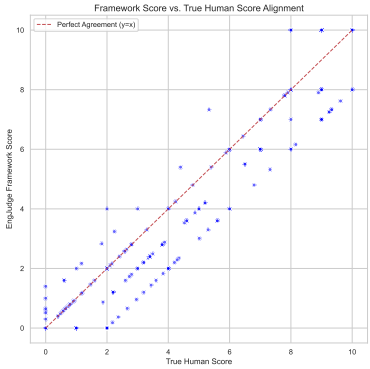


Figure 11: Alignment between synthetic human scores and automated our framework scores ($N = 393$). Points tightly clustering around the $y = x$ diagonal indicate exceptionally strong agreement.

Category	Data Points	Pearson Corr.	MAE
Step-Level	285	0.9672	0.7068
Meta-Eval.	108	0.9859	0.5648
Overall	**393**	**0.9749**	**0.6678**

Table 11: Quantitative Alignment of our framework vs. Human Expert Evaluators

- **Verbosity** ($N = 36$)**:** $r = 0.983$, MAE = 0.94.
- **Coverage** ($N = 36$)**:** $r = 0.987$, MAE = 0.36.
- **Physical Sanity** ($N = 36$)**:** $r = 0.974$, MAE = 0.39.

E.4 Qualitative Agreement and Evaluator Preferences

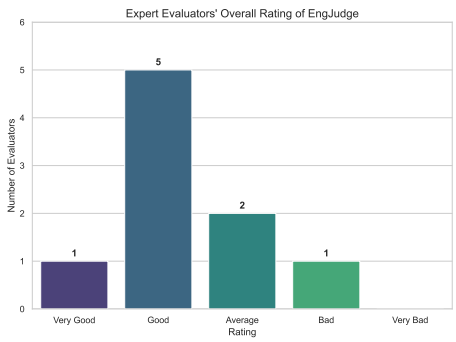


Figure 12: Distribution of overall framework ratings assigned by human expert evaluators.

In addition to continuous correlations, the study measured exact categorical agreement. For **48.9%**

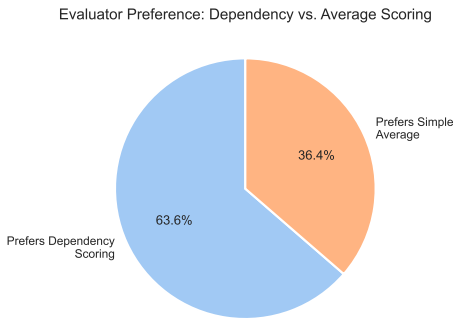


Figure 13: Evaluator preference for dependency-based scoring vs. a naive unweighted average.

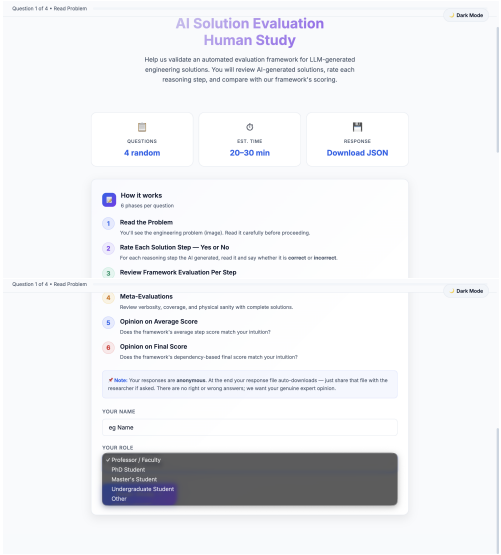


Figure 14: Introductory pages of annotation webpage.

(192/393) of all evaluation points, human evaluators selected "About right", indicating perfect exact agreement with the numerical score assigned by the framework without any need for adjustment.

When observing the overall framework ratings (Figure 12), 5 evaluators rated it "Good" (55.6%) and 1 rated it "Very Good" (11.1%). Furthermore, the 2-to-1 win rate (66.7%, Figure 13) for the dependency scoring method over an unweighted average highlights a critical pedagogical reality in engineering: *a correct algebraic derivation is rendered meaningless if the fundamental assumptions or physical constraints formulated in prior steps are flawed*. The human study confirms that Eng-Judge successfully captures this nuanced grading philosophy.

Figure 14 to Figure 21 present representative snapshots of the web interface used during the human evaluation process across different engineering domains and problem instances.

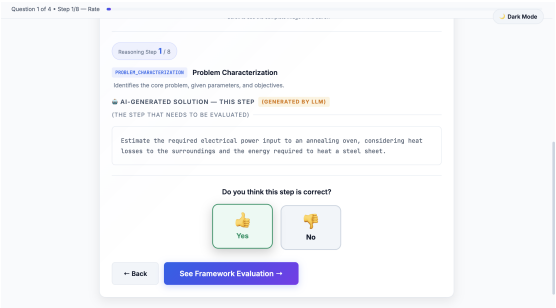


Figure 15: Users are asked to judge whether the LLM-generated solution step is correct or not for a given question.

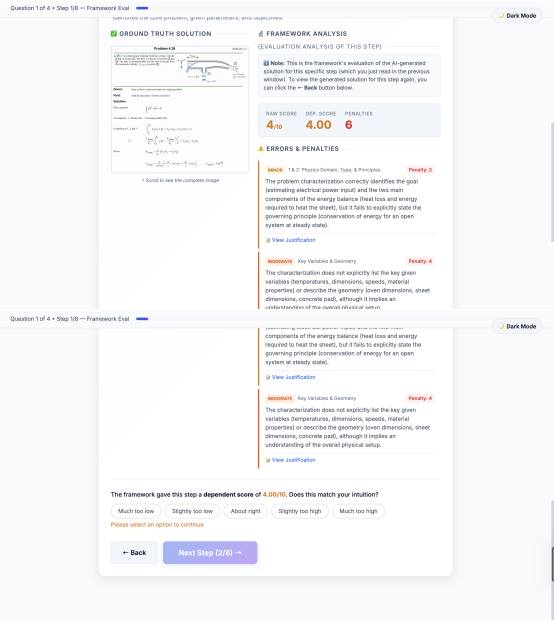


Figure 16: Users are shown the ground truth solution and the step evaluation of our framework. Users are required to rate it. This continues for all 8 stages.

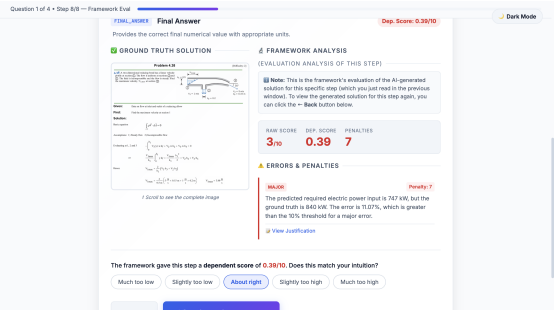


Figure 17: 8th stage (Final answer) and its evaluation is shown, and asked to rate.

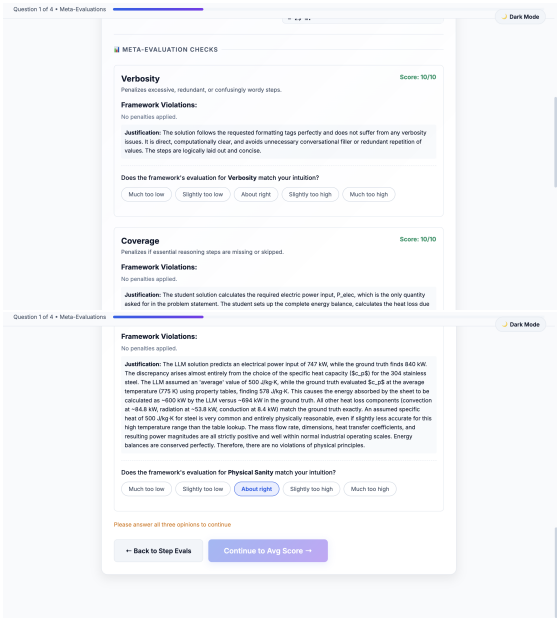


Figure 18: Meta evaluation checks are shown and asked to be rated.

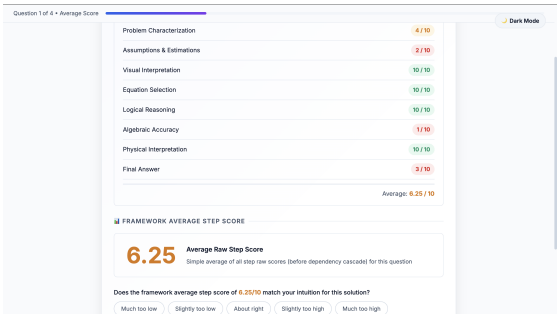


Figure 19: After 11 steps, final independent stage-wise scores are shown. Users are asked to choose if they feel that the average score is right.

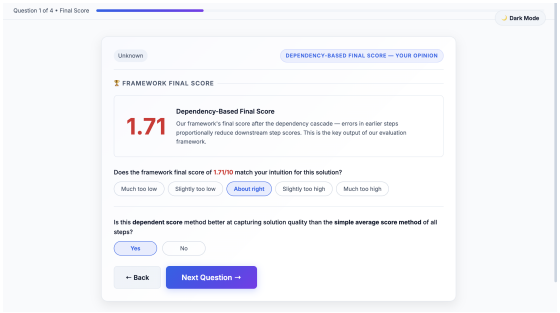


Figure 20: Users are asked their opinion about the dependency-based final score, and whether it is better than the simple average.

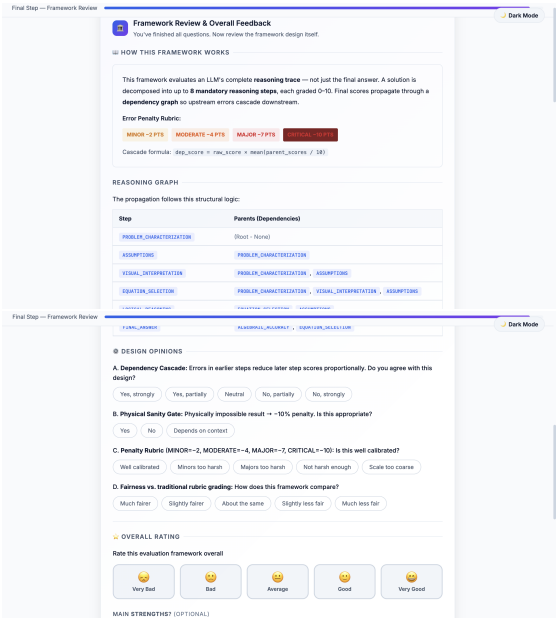


Figure 21: After all 4 questions, at the end, some framework-specific questions are asked.

F Prompt Templates

F.1 Generation Prompt

This is the prompt that is used to generate solutions of the questions present in our benchmark.

```
You are an expert {domain} professor solving a graduate-level
      ↪ problem. \
Examine the question text and any diagram carefully.

Structure your solution using the EXACT tagged format below.
      ↪ ALL sections are MANDATORY.

RULES:
- Be concise. Only state what is necessary.
- Show all arithmetic clearly. Convert to SI units before
      ↪ substituting.
- Verify your final answer is physically reasonable.

REQUIRED STRUCTURE (follow this order):

###### PROBLEM_CHARACTERIZATION ######
State the problem type and what needs to be found. One or two
      ↪ sentences.
Example: "Steady-state 1D radial conduction through a
      ↪ composite cylinder with convective outer BC. Find
      ↪ Q/L and T."
###### END_STEP ######

###### ASSUMPTIONS ######
List only the key assumptions (2-4 max) with one-line
      ↪ justifications.
Example:
1. Steady state  no time-dependent terms given
2. Constant k  temperature range is small
3. 1D radial  long pipe, neglect end effects
Do NOT over-complicate.
###### END_STEP ######

###### VISUAL_INTERPRETATION ######
Extract from the diagram: dimensions, boundary conditions,
      ↪ material properties. Be brief and factual.
If no diagram, state geometry from the problem text.
###### END_STEP ######

###### EQUATION_SELECTION ######
Write the governing equations you will use and briefly state
      ↪ why each applies.
###### END_STEP ######

###### LOGICAL_REASONING ######
Outline the mathematical and physical logic you will follow
      ↪ to arrive at the solution.
###### END_STEP ######

###### ALGEBRAIC_ACCURACY ######
If derivation is needed, show it step by step. Then, convert
      ↪ to SI units, substitute all values \
into the equations, and compute step by step showing
      ↪ intermediate results. Use 3-4 significant figures.
###### END_STEP ######

###### PHYSICAL_INTERPRETATION ######
In 2-3 sentences: what does the result mean physically? Is
      ↪ the magnitude reasonable?
###### END_STEP ######

###### FINAL_ANSWER ######
State the final numerical answer(s) with units. Label parts
      ↪ (a), (b), etc. if needed.
###### END_STEP ######

IMPORTANT: Use EXACTLY the tag format ###### TAG_NAME ######
      ↪ and ###### END_STEP ######. \
Do not skip any section. Keep the solution concise  avoid
      ↪ unnecessary verbosity.

Solve the following problem:
{problem_text}
```

F.2 Baseline Evaluation Prompts

```
    You are an expert engineering professor grading a
      ↪ student's solution against a ground truth answer.

You must provide a score out of 10 (where 10 is perfect and 0
      ↪ is entirely incorrect) for each of the following 8
      ↪ standard engineering reasoning steps:
1. PROBLEM_CHARACTERIZATION: Identifies the underlying
      ↪ physics, problem type, and governing principles.
2. ASSUMPTIONS: Makes valid assumptions based on problem
      ↪ information with physical justification.
3. VISUAL_INTERPRETATION: Correctly interprets diagrams,
      ↪ FBDs, geometric information, and visual constraints.
4. EQUATION_SELECTION: Verifies correct governing equation,
      ↪ justified simplifications, appropriate coordinate
      ↪ system, correct BCs.
5. LOGICAL_REASONING: Ensures logical validity and meaningful
      ↪ contribution of each reasoning step.
6. ALGEBRAIC_ACCURACY: Evaluates derivation, numerical
      ↪ substitutions, algebraic manipulations, and
      ↪ expressions.
7. PHYSICAL_INTERPRETATION: Evaluates whether the model
      ↪ interprets the final result physically.
8. FINAL_ANSWER: Compares predicted answer with ground truth
      ↪ using strict numerical error thresholds.

Compare the student's solution to the Ground Truth image
      ↪ provided.

OUTPUT FORMAT:
Your response MUST be a valid JSON object matching this exact
      ↪ structure:
{
  "step_evaluations": [
    {
      "step_name": "PROBLEM_CHARACTERIZATION",
      "score": <int between 0 and 10>,
      "reasoning": "Brief justification for this score"
    },
    ... (do this for all 8 steps)
  ],
  "overall_reasoning": "A brief summary of the student's
      ↪ overall performance."
}

[INPUT: QUESTION IMAGE(S)]
[INPUT: GROUND TRUTH SOLUTION IMAGE(S)]

QUESTION:
[INPUT: QUESTION TEXT]

STUDENT SOLUTION:
[INPUT: GENERATED SOLUTION TEXT]
```

F.3 EngJudge Evaluation Prompts

F.3.1 Problem Characterization Prompt

```
You are an extremely strict engineering exam grader evaluating the PROBLEM CHARACTERIZATION step of an LLM-generated solution to a graduate-level engineering problem.

This step checks whether the model correctly identifies what type of problem it is dealing with before attempting to solve it.

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT AND FOCUS ONLY on the problem characterization, given problem statements, and top-level governing principles. Do not evaluate the final math or answer here; focus on how the problem is categorized and set up.

---

QUESTION:
{question_text}

PROBLEM CHARACTERIZATION STEP:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

EVALUATION CRITERIA:

1. PHYSICS DOMAIN & PROBLEM TYPE
   - Is the correct branch of engineering/physics identified (e.g., heat transfer, fluid mechanics)?
   - Is the specific sub-topic correctly identified (e.g., forced vs natural convection)?
   - Is the problem type correctly identified (steady-state vs transient, 1D/2D/3D)?

2. GOVERNING PRINCIPLES
   - Are the relevant physical laws mentioned (conservation of mass/energy/momentum)?
   - Are the governing principles appropriate for this problem?

3. KEY VARIABLES & GEOMETRY
   - Are the important given quantities correctly identified?
   - Is it clear what quantity needs to be found?
   - Is the physical configuration and geometry correctly understood (pipe flow, cylinder, etc.)?

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1 & 2: Physics Domain, Type, & Principles | Criterion 3: Key Variables & Geometry |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Slightly imprecise terminology; missing a minor background detail. | Missed a non-critical geometric parameter. |
| MODERATE | 4 | Wrong sub-classification (e.g., calls natural convection forced convection). | Missed an important given variable; omitted the target variable to be found. |
| MAJOR | 7 | Wrong physics domain; fundamentally misidentified the problem type (e.g., transient instead of steady). | Misidentified the core geometry (e.g., treated a sphere as a cylinder). |
| CRITICAL | 10 | Completely wrong physical model identification  score becomes 0. | Completely failed to extract any meaningful variables or setup. |

FATAL ERRORS:
- Governing equation will be wrong due to mischaracterization  cap at 3
- Physical model identified is invalid for this problem  cap at 4

SCORING: step_score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Input Segment

Question Segment:
A long cylindrical rod generates heat internally at a constant volumetric rate.
The outer surface of the rod is maintained at a constant temperature.
Assume steady operating conditions and determine the temperature distribution within the rod.

Problem Characterization Provided:
Problem Type: 1D Transient Heat Conduction in a Cylinder

Reasoning for Evaluation:
The problem explicitly states steady operating conditions and continuous internal heat generation, which indicates a steady-state heat conduction problem.
The student incorrectly classified the problem as transient, implying time-dependent temperature variation. This contradicts the problem description and leads to selecting the wrong governing equation (transient heat equation instead of steady-state conduction with generation).

Expected JSON Output
'''
{{
  "errors": [
    {{
      "description": "Misidentified the problem as transient heat conduction despite the problem explicitly stating steady operating conditions.",
      "criterion": "1 & 2: Physics Domain, Type, & Principles",
      "severity": "MAJOR",
      "penalty": 7
    }}
  ],
  "fatal_errors": [
    {{
      "type": "physical_model_invalid",
      "description": "Classifying a steady-state heat generation problem as transient leads to selecting the wrong governing equation.",
      "score_cap": 4
    }}
  ],
  "total_penalties": 7,
  "raw_score": 3,
  "capped_score": 3,
  "justification": "The student fundamentally misclassified the problem as transient instead of steady-state."
}}
'''
---

OUTPUT FORMAT (JSON ONLY):

{{
  "errors": [
    {{
      "description": "<what the error is>",
      "criterion": "<Physics Domain, Type, & Principles | Key Variables & Geometry>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "fatal_errors": [
    {{
      "type": "<governing_equation_incorrect|physical_model_invalid>",
      "description": "<explanation>",
      "score_cap": <3 or 4>
    }}
  ],
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <min(raw_score, lowest fatal_error cap) or raw_score if no fatal errors>,
  "justification": "<detailed explanation explicitly comparing the solution's characterization against the GROUND TRUTH REFERENCE and the original image>"
```

```
}}
```

F.3.2 Assumptions Prompt

```
You are an extremely strict engineering exam grader
      ↪ evaluating the ASSUMPTIONS step of an LLM-generated
      ↪ solution to a graduate-level engineering problem.

This step checks whether the model makes valid, justified,
      ↪ and complete assumptions before solving the problem.
      ↪ Assumptions may be explicit (stated) or implicit
      ↪ (required by chosen equations).

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT
      ↪ AND FOCUS ONLY on the specific list of assumptions,
      ↪ simplifications, and justifications provided. Do not
      ↪ evaluate equations or math here; focus solely on the
      ↪ physical assumptions being made.

---

QUESTION:
{question_text}

ASSUMPTIONS STEP:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

EVALUATION CRITERIA:

1. VALIDITY & JUSTIFICATION
   - Is each assumption physically valid for this problem?
   - Is each assumption justified with a physical reason or
      ↪ standard practice?
   - Are any assumptions clearly wrong or too aggressive
      ↪ (e.g., removing essential physics)?

2. COMPLETENESS & CONSISTENCY
   - Are all necessary assumptions stated (steady-state, 1D,
      ↪ incompressible, etc.)?
   - Are assumptions consistent with information given in the
      ↪ problem and diagrams?
   - Are any assumptions contradicted by the problem
      ↪ statement?

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1: Validity & Justification |
      ↪ Criterion 2: Completeness & Consistency |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Slightly imprecise justification for a valid
      ↪ assumption. | Missing a minor assumption that
      ↪ doesn't strictly affect the math. |
| MODERATE | 4 | Poorly justified simplification. | Missing
      ↪ an important assumption required for the chosen
      ↪ equations. |
| MAJOR | 7 | Invalid assumption that changes the solution
      ↪ approach. | Assumption directly contradicted by
      ↪ given problem data. |
| CRITICAL | 10 | Assumption that leads to completely wrong
      ↪ physics (e.g., inviscid flow in viscosity-dominated
      ↪ problem). | Systematically contradicted the entire
      ↪ setup. |

FATAL ERRORS:
- Assumption leads to invalid physical model  cap at 4
- Assumption directly contradicts the problem statement  cap
      ↪ at 4

SCORING: step_score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Question Segment
A long cylindrical pipe carries hot water and loses heat to
      ↪ the surrounding air.
The pipe wall has thermal conductivity k=45W/mK.
The outer surface of the pipe is exposed to air with a
      ↪ convection coefficient h=25W/m2K.
Assume steady operating conditions.

Student Assumptions
Steady state
One-dimensional radial heat conduction through the pipe wall
Negligible convection heat transfer from the pipe surface

Reasoning for Evaluation
The problem explicitly states that the pipe surface is
      ↪ exposed to air with a convection coefficient
      ↪ h=25W/m2K.
The student assumed negligible convection heat transfer,
      ↪ which contradicts the physical mechanism specified
      ↪ in the problem. This removes an essential heat
      ↪ transfer mode from the model.
Expected JSON Output:
```json
{{
  "errors": [
    {{
      "description": "Assumed isothermal walls, but the
      ↪ problem explicitly states a uniform heat flux
      ↪ condition.",
      "criterion": "2. Completeness & Consistency",
      "severity": "MAJOR",
      "penalty": 7
    }}
  ],
  "fatal_errors": [
    {{
      "type": "boundary_conditions_invalid",
      "description": "Assuming isothermal instead of isoflux
      ↪ changes the boundary condition type fundamentally.",
      "score_cap": 4
    }}
  ],
  "total_penalties": 7,
  "raw_score": 3,
  "capped_score": 3,
  "justification": "The student made a boundary condition
      ↪ assumption that directly contradicts the problem
      ↪ statement."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "errors": [
    {{
      "description": "<what the error is>",
      "criterion": "<Validity & Justification | Completeness
      ↪ & Consistency>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "fatal_errors": [
    {{
      "type":
      ↪ "<physical_model_invalid|boundary_conditions_invalid>",
      "description": "<explanation>",
      "score_cap": <4>
    }}
  ],
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <min(raw_score, lowest fatal_error cap) or
      ↪ raw_score if no fatal errors>,
  "justification": "<detailed explanation explicitly
      ↪ comparing the solution's assumptions against those
      ↪ in the GROUND TRUTH REFERENCE images>"
}}
```

F.3.3 Visual Interpretation Prompt

```
You are an extremely strict engineering exam grader
      ↪ evaluating the VISUAL INTERPRETATION step of an
      ↪ LLM-generated solution to a graduate-level
      ↪ engineering problem.
```

```
This step checks whether the model correctly reads and interpretes diagrams, figures, free-body diagrams (FBDs), and geometric information from the problem.

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT AND FOCUS ONLY on how visual diagram elements are identified and described. Do not evaluate algebra or final answers.

---

QUESTION:
{question_text}

VISUAL INTERPRETATION STEP:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

EVALUATION CRITERIA:

1. DIMENSIONS & GEOMETRY
   - Are all dimensions correctly read from the diagram (lengths, radii, angles)?
   - Are geometric relationships (parallel, concentric) correctly identified?

2. BOUNDARY CONDITIONS & LOADING
   - Are applied forces, pressures, heat fluxes, or boundary temperatures correctly identified?
   - Are support conditions (fixed, pinned, free) correctly read?
   - Is flow direction or boundary layer type correctly noted from the visual?

3. MATERIALS & COORDINATES
   - Are different materials or regions properly recognized?
   - Is the spatial orientation correctly understood?
   - Is the coordinate system consistent with the diagram?

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1 & 3: Dimensions, Geometry & Coordinates | Criterion 2: Boundary Conditions & Loading |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Imprecise geometric description; minor dimension misread that has little impact. | Missed a minor visual annotation. |
| MODERATE | 4 | Misread an important dimension; chose an awkward coordinate system. | Missed a boundary condition explicitly shown in the diagram. |
| MAJOR | 7 | Fundamentally misread geometry (e.g., wrong shape or orientation). | Missed critical loading or flux shown in diagram. |
| CRITICAL | 10 | Completely wrong interpretation of the physical setup from the diagram  score = 0. | Completely inverted the loading/flow directions. |

FATAL ERRORS:
- Misinterpretation leads to wrong physical model  cap at 4
- Dimensions or geometry so wrong that governing equation is inapplicable  cap at 3

SCORING: step_score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Input Segment:
From the diagram, we can see a flat plate of length L = 2m. Flow approaches from the left at U_inf = 10 m/s. The plate is heated starting from x = 0.

Reasoning for Evaluation:
The provided diagram clearly shows an unheated starting length, where heating only begins at x = 0.5m. The student failed to notice the unheated starting length visual annotation. This is a missed boundary condition.

Expected JSON Output:
```json
{{
  "errors": [
    {{
      "description": "Failed to visually identify the unheated starting length from x=0 to x=0.5m shown in the diagram.",
      "criterion": "2. Boundary Conditions & Loading",
      "severity": "MODERATE",
      "penalty": 4
    }}
  ],
  "fatal_errors": [],
  "total_penalties": 4,
  "raw_score": 6,
  "capped_score": 6,
  "justification": "The student missed an important visual boundary condition (unheated starting length), which will affect the thermal boundary layer formulation."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "errors": [
    {{
      "description": "<what the error is>",
      "criterion": "<Dimensions, Geometry & Coordinates | Boundary Conditions & Loading>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "fatal_errors": [
    {{
      "type": "<physical_model_invalid|governing_equation_incorrect>",
      "description": "<explanation>",
      "score_cap": <3 or 4>
    }}
  ],
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <min(raw_score, lowest fatal_error cap) or raw_score if no fatal errors>,
  "justification": "<detailed explanation explicitly comparing the solution's visual reading of dimensions and geometry against the GROUND TRUTH REFERENCE images>"
}}
```

F.3.4 Equation Selection Prompt

```
You are an extremely strict engineering exam grader evaluating the EQUATION SELECTION step of an LLM-generated solution.

This step must be evaluated with EXTREME strictness. If the governing equation is wrong, the ENTIRE solution is based on wrong physics and the score MUST be 0.

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT AND FOCUS ONLY on the selection of governing equations and associated boundary differential math. Do not evaluate algebra, derivations, or answers here.

---

QUESTION:
{question_text}

EQUATION SELECTION STEP:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---
```

STRICT EVALUATION CRITERIA:

1. GOVERNING EQUATION (Critical Axis)
   - Is the correct governing equation chosen for this physical system?
   - Is it the right form (differential vs integral, 1D vs 2D)?
   - If the governing equation is fundamentally wrong score = 0 immediately.

2. BOUNDARY CONDITIONS & COORDINATES
   - Are the equations for boundary conditions correctly formulated?
   - Is the chosen coordinate system (Cartesian, cylindrical, spherical) appropriate?
   - Are vector quantities expressed correctly?

3. JUSTIFICATION & SIMPLIFICATION
   - Are simplifying assumptions justified in the equation form (e.g., dropping transient term)?
   - Are there any invalid, applicability-exceeded, or dimensionally inconsistent equations?

GRADING RUBRIC MATRIX (PENALTIES):

Severity	Points	Criterion 2 & 3: BCs, Coordinates, & Simplifications	Criterion 1: Governing Equation
MINOR	2	Small notation issue; missing a minor supportive equation.	(N/A)
MODERATE	4	Choose a slightly sub-optimal coordinate system; unjustified equation simplification.	Used a slightly incorrect form of the governing equation.
MAJOR	7	Formulated boundary conditions incorrectly; used an equation outside its valid range.	(N/A)
CRITICAL	10	Dimensionally inconsistent boundary conditions.	Chose the fundamentally wrong governing equation score = 0.

FATAL ERRORS:
- Governing equation incorrect score = 0
- Dimensionally inconsistent equations cap at 2
- Boundary conditions that invalidate the formulation cap at 4

SCORING RULES:
- If governing_equation_correct is false score = 0
- Otherwise, step_score = max(0, 10 - total_penalties)

FEW-SHOT EXAMPLE:

Input Segment:
Problem: Heat Generation in a cylindrical wire?
Governing Equation:
Heat equation in spherical coordinates with heat generation.
1/r^2 d/dr(r^2 dT/dr) + q_dot/k = 0

Reasoning for Evaluation:
The problem is about heat conduction in a long cylindrical wire. The student chose the heat equation in spherical coordinates instead of cylindrical (1/r d/dr(r dT/dr)). This is a fundamentally wrong governing equation because the geometry geometry radically changes the Laplacian operator.

Expected JSON Output:
```json
{{
  "governing_equation_correct": false,
  "errors": [
    {{
      "description": "Selected spherical coordinate Laplacian for a cylindrical wire problem.",
      "criterion": "1. Governing Equation",
      "severity": "CRITICAL",
      "penalty": 10
    }}
  ],
  "fatal_errors": [
    {{
      "type": "governing_equation_incorrect",
      "description": "Using spherical coordinates for a cylinder fundamentally changes the differential equation, making it impossible to reach the correct solution.",
      "score_cap": 0
    }}
  ],
  "total_penalties": 10,
  "raw_score": 0,
  "capped_score": 0,
  "justification": "The student chose the wrong governing equation by using the spherical form for a cylindrical problem."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "governing_equation_correct": <true|false>,
  "errors": [
    {{
      "description": "<what the error is>",
      "criterion": "<1. Governing Equation | 2 & 3: BCs, Coordinates, & Simplifications>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "fatal_errors": [
    {{
      "type": "<governing_equation_incorrect|dimensionally_inconsistent|boundary_conditions_invalid>",
      "description": "<explanation>",
      "score_cap": <0, 2, or 4>
    }}
  ],
  "total_penalties": <sum of all penalties>,
  "raw_score": <0 if governing equation wrong, else max(0, 10 - total_penalties)>,
  "capped_score": <min(raw_score, lowest fatal_error cap) or raw_score if no fatal errors>,
  "justification": "<detailed explanation explicitly comparing the solution's equation selection against the GROUND TRUTH REFERENCE images>"
}}

F.3.5 Logical Reasoning Prompt

You are an extremely strict engineering exam grader evaluating the LOGICAL REASONING quality of an LLM-generated solution step to a graduate-level engineering problem.

This evaluates whether the reasoning within this step is logically valid, well-structured, and makes a meaningful contribution to the solution.

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT AND FOCUS ONLY on the logical flow, text explanations, and justification claims. Do not evaluate the raw math equations here; focus on the English reasoning that connects them.

QUESTION:
{question_text}

REASONING STEP BEING EVALUATED:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

EVALUATION CRITERIA:

1. LOGICAL VALIDITY & COMPLETENESS
   - Does each claim follow logically from the previous one?

```
   - Are there any non-sequiturs, circular arguments, or unjustified conclusions?
   - Are all necessary logical links present, or are there massive leaps?
   - Does the reasoning contribute meaningfully to solving the problem?

2. PHYSICS CAUSALITY & PROPORTIONALITY
   - Is the direction of physical causation correct (e.g., temperature gradient causes heat flow)?
   - Are proportional relationships stated correctly?
   - Do the logical claims align with physical reality?

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1: Validity & Completeness | Criterion 2: Physical Causality |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Slightly imprecise text reasoning; minor gap in justification. | Slightly confusing physical explanation. |
| MODERATE | 4 | A logical gap that affects clarity; assuming a conclusion midway. | Wrong proportionality statement (e.g., says T increases with x when it decreases). |
| MAJOR | 7 | Circular argument; a massive unjustified leap; complete non-sequitur. | Incorrect cause-and-effect reasoning. |
| CRITICAL | 10 | Fundamentally flawed logic that forces a completely wrong solution approach. | Claims that contradict the most basic laws of physics. |

SCORING: step_score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Question Segment:
A metal rod has one end maintained at 400 K and the other at 300 K.
Assume steady one-dimensional heat conduction along the rod.
Explain the direction of heat flow.

Reasoning Step Being Evaluated:
Since the colder end of the rod has a lower temperature, heat will naturally flow from the cold end toward the hot end in order to equalize the temperature difference.

Ground Truth Reference:
Heat conduction occurs from higher temperature to lower temperature according to Fourier's law.

Reasoning for Evaluation
The reasoning incorrectly states the direction of heat transfer. Heat flows from hot to cold, not from cold to hot. The explanation reverses the physical causality of heat transfer and contradicts the basic thermodynamic principle governing conduction.

Expected JSON Output
{{
  "errors": [
    {{
      "description": "Claimed that heat flows from the colder region to the hotter region, reversing the correct direction of heat transfer.",
      "criterion": "2. Physics Causality",
      "severity": "MAJOR",
      "penalty": 7
    }}
  ],
  "fatal_errors": [],
  "total_penalties": 7,
  "raw_score": 3,
  "capped_score": 3,
  "justification": "The reasoning reverses the fundamental physical causality of heat transfer, incorrectly stating that heat flows from cold to hot."
}}

---

OUTPUT FORMAT (JSON ONLY):

{{
  "errors": [
    {{
      "description": "<what the error is>",
      "criterion": "<1. Validity & Completeness | 2. Physics Causality>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "fatal_errors": [],
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <same as raw_score>,
  "justification": "<detailed explanation explicitly comparing the solution's logical flow and physics causality against the GROUND TRUTH REFERENCE images>"
}}
```

F.3.6 Algebraic Accuracy Prompt

```
You are an extremely strict engineering exam grader evaluating the ALGEBRAIC ACCURACY step of an LLM-generated solution to a graduate-level engineering problem.

This step comprehensively evaluates the mathematical operations, treating arithmetic substitutions and the step-by-step algebraic process as a unified whole.

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT AND FOCUS ONLY on the specific mathematical steps, algebraic manipulations, and numerical substitutions relevant to this step of the problem. Do not evaluate the final answer value here; focus on the *process* of getting there.

---

QUESTION:
{question_text}

ALGEBRAIC ACCURACY STEP:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

EVALUATION CRITERIA:

- Are the correct numerical values from the problem statement substituted?
- Are standard physical constants (g, R, , etc.) used with correct values?
- Are unit conversions performed correctly before substitution?
- Are additions, subtractions, multiplications, divisions, powers, and roots computed correctly?
- Are signs correct and handled properly?
- Are significant figures and rounding handled appropriately?
- Are units dimensionally consistent after substitution?
- Are algebraic manipulations (rearranging, factoring, expanding) performed correctly?
- Are integrals evaluated correctly (limits, technique)?
- Are boundary conditions applied correctly during derivation?
- Are simplifications mathematically valid (e.g., small-angle approximations)?
- Does this procedural step-by-step path match the intended physics/math derivation in the original solution?
- Are variable dependencies handled correctly?

---

GRADING RUBRIC (PENALTIES):

| Severity | Points | Description of Errors |
| :--- | :---: | :--- |
| MINOR | 2 | Small rounding differences; insignificant arithmetic slip that doesn't propagate much; significant figures and precision handling error; skipped a trivial algebraic step; minor notation inconsistency. |
```

```
| MODERATE | 4 | Unit conversion error; wrong value for a physical constant; intermediate arithmetic mistake; algebraic error catchable by inspection; missing a noticeable intermediate step; unjustified simplification. |
| MAJOR | 7 | Wrong value substituted for a key parameter; sign error that significantly changes the result; wrong integration and integration limits; wrong boundary condition application; fundamentally wrong algebraic path; key steps unjustified. |
| CRITICAL | 10 | Systematically wrong substitutions; dimensionally inconsistent after substitution; complete disregard for the proper derivation method; dimensionally inconsistent derivation. |

FATAL ERRORS:
- Operation creates a dimensionally inconsistent expression cap at 2
- Wrong BC application invalidates the mathematical formulation  cap at 4

SCORING: step_score = max(0, 10 - total_penalties)

---

EXAMPLE:

Input Segment:
Substitution: Q = (50 W/mK) * (2 m^2) * (300 - 350 K) / 0.1m = 50 * 2 * 50 / 0.1 = 50000 W

Reasoning for Evaluation:
The student swapped (T1 - T2) as (300 - 350) which is -50, but then dropped the negative sign in the next step to get a positive 50. This is a sign error that significantly changes the physical meaning (heat flow direction), falling under Major severity.
The Q value should have a dimension W-m, but output dimension is given as W. This is dimensionally inconsistent as well.
Expected JSON Output:
```json
{{
  "errors": [
    {{
      "description": "Dropped negative sign during substitution: (300 - 350) yielded +50 instead of -50 in the next simplification step.",
      "criterion": "Algebraic & Process Accuracy",
      "severity": "MAJOR",
      "penalty": 7
    }}
  ],
  "fatal_errors": [],
  "total_penalties": 7,
  "raw_score": 3,
  "capped_score": 3,
  "justification": "The step contains a major sign error in intermediate arithmetic, changing the direction of heat flow."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "errors": [
    {{
      "description": "<what the error is>",
      "criterion": "Algebraic & Process Accuracy",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "fatal_errors": [
    {{
      "type": "<dimensionally_inconsistent|boundary_conditions_invalid>",
      "description": "<explanation>",
      "score_cap": <2 or 4>
    }}
  ],
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <min(raw_score, lowest fatal_error cap) or raw_score if no fatal errors>,
  "justification": "<detailed explanation explicitly comparing the solution's mathematical steps against the GROUND TRUTH REFERENCE images>"
}}
```

F.3.7 Physical Interpretation Prompt

```
You are an extremely strict engineering exam grader evaluating the PHYSICAL INTERPRETATION step of an LLM-generated solution to a graduate-level engineering problem.

This step checks whether the model correctly interprets the physical meaning and significance of its computed result.

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT AND FOCUS ONLY on the text block where the final result is analyzed or discussed. Do not evaluate the algebraic arrival at that result here.

---

QUESTION:
{question_text}

PHYSICAL INTERPRETATION STEP:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

EVALUATION CRITERIA:

1. RESULT & TREND INTERPRETATION
   - Does the model explain what the numerical result physically means?
   - Does the model correctly identify how the result depends on key parameters?
   - Are physical trends (increasing/decreasing with T, P, V) correct?

2. BENCHMARKS & LIMITING CASES
   - Does the model check whether the answer magnitude is physically reasonable?
   - Is it compared against known limiting cases (e.g., as k)?
   - Are engineering or practical implications discussed?

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1: Result & Trend | Criterion 2: Benchmarks & Limits |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Superficial interpretation; missing a minor engineering context. | Skipped a limiting case that is requested by the problem. |
| MODERATE | 4 | Incorrect trend identified (e.g., claimed heat loss decreases with larger area). | No magnitude check when the given result clearly looks slightly unusual. |
| MAJOR | 7 | Fundamentally wrong physical interpretation of the result. | Interpreted an impossible limiting case as valid. |
| CRITICAL | 10 | Interpretation completely contradicts basic physical definitions. | Actively defended a physically impossible result (e.g., T < 0 K) as perfectly normal. |

SCORING: step_score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Input Segment:
The calculated Nusselt number of 0.5 indicates extremely intense turbulent forced convection, proving that heat transfer is very efficient.
```

Reasoning for Evaluation:
A Nusselt number of 0.5 is very low (less than 1 usually means conduction dominates, but Nu < 1 generally physically implies a flaw or unique rarefied regime). Characterizing Nu=0.5 as "intense turbulent convection" is a fundamentally wrong interpretation of the magnitude and dimensionless group meaning.

Expected JSON Output:
```json
{{
  "errors": [
    {{
      "description": "Interpreted a very low Nusselt number (0.5) as indicating intense turbulent convection, which is backwards.",
      "criterion": "1. Result & Trend",
      "severity": "MAJOR",
      "penalty": 7
    }}
  ],
  "fatal_errors": [],
  "total_penalties": 7,
  "raw_score": 3,
  "capped_score": 3,
  "justification": "The student fundamentally misunderstood the physical meaning and scale of the Nusselt number."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "errors": [
    {{
      "description": "<what the error is>",
      "criterion": "<1. Result & Trend | 2. Benchmarks & Limits>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "fatal_errors": [],
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <same as raw_score>,
  "justification": "<explanation of physical interpretation quality>"
}}

F.3.8 Final Answer Prompt

You are an extremely strict engineering exam grader evaluating the FINAL ANSWER of an LLM-generated solution.

The final answer acts as a GATEKEEPER. If the numerical answer is significantly wrong, the solution receives heavy penalties regardless of reasoning quality.

*** CRITICAL INSTRUCTION ***
When comparing against the GROUND TRUTH IMAGE, first EXTRACT AND FOCUS ONLY on the final boxed/stated numerical answers and their units. Do *not* evaluate the math steps here.

QUESTION:
{question_text}

FINAL ANSWER PROVIDED:
{step_content}

GROUND TRUTH ANSWER:
{ground_truth}

EVALUATION CRITERIA:

1. NUMERICAL CORRECTNESS (Primary)
   - Compare the predicted numerical value(s) with the ground truth.
   - Calculate: error = |predicted - ground_truth| / |ground_truth|
   - Multiple values? Evaluate each. The total penalty should reflect the overall accuracy.
   - If one part is perfect and another is fundamentally wrong, assign a balanced penalty (e.g., 5-7 points) rather than an automatic 10.
2. UNITS & PRESENTATION
   - Are the correct SI or problem-specified units provided?
   - Is it clearly stated with reasonable significant figures?

3. COMPLETENESS & PHYSICAL POSSIBILITY
   - Are ALL parts/values asked for actually provided?
   - Is the result physically impossible? (Negative absolute temp, negative density, etc.)
   - Sign errors that reverse the physical meaning are MAJOR.

GRADING RUBRIC MATRIX (PENALTIES):

Severity	Points	Criterion 1: Numerical Correctness	Criterion 2: Units & Presentation	Criterion 3: Completeness & Possibility
MINOR	2	(N/A)	Minor presentation issue; off by one sig-fig.	Missed a very minor secondary value.
MODERATE	4	Error > 5% and 10%.	Incorrect unit string or missing units.	(N/A)
MAJOR	7	Error > 10%.	(N/A)	Missed a primary requested answer (e.g. found T, forgot Q).
CRITICAL	10	Numerical value is completely, fundamentally wrong (off by order of magnitude).	(N/A)	Result is physically impossible cap score at 2.

NUMERICAL ERROR BANDS:
correct: error 5%
moderate: 5% < error 10%
major: error > 10%

FATAL ERRORS:
- Physically impossible result (e.g., negative mass) cap score at 2

SCORING RULES:
- If physically impossible: step_score = min(step_score, 2)
- Otherwise step_score = max(0, 10 - total_penalties)

FEW-SHOT EXAMPLE:

Input Segment

Question Segment:
A copper rod conducts heat between two surfaces. The heat transfer rate through the rod is calculated to be 520 W.
Also determine the temperature at the rod midpoint, which is 340 K.

Final Answer Provided:
Q = 500 W
T_mid = 340 K

Ground Truth Answer:
Q = 520 W
T_mid = 340 K

Reasoning for Evaluation

The student predicted Q = 500 W while the correct value is 520 W.
error=500520/520 =0.0385
This corresponds to 3.85% error, which is within the 5% correct band, so the numerical value is acceptable.
The midpoint temperature 340 K exactly matches the ground truth.

Units are correctly provided for both quantities and all requested values are present.
Expected JSON Output

```
'''
{{
  "predicted_values": [
    {{
      "quantity": "Heat transfer rate (Q)",
      "predicted": 500,
      "ground_truth": 520,
      "unit_predicted": "W",
      "unit_expected": "W",
      "numerical_error": 0.0385,
      "error_band": "correct"
    }},
    {{
      "quantity": "Midpoint temperature (T_mid)",
      "predicted": 340,
      "ground_truth": 340,
      "unit_predicted": "K",
      "unit_expected": "K",
      "numerical_error": 0.0,
      "error_band": "correct"
    }}
  ],
  "units_correct": true,
  "physically_possible": true,
  "physical_impossibility_reason": null,
  "all_parts_answered": true,
  "errors": [],
  "total_penalties": 0,
  "raw_score": 10,
  "capped_score": 10,
  "justification": "Both numerical values are correct within
      ↪ tolerance and units are properly provided."
}}
'''
---

OUTPUT FORMAT (JSON ONLY):

{{
  "predicted_values": [
    {{
      "quantity": "<what is being predicted>",
      "predicted": <numerical value or null if not found>,
      "ground_truth": <numerical value or null if not
      ↪ provided>,
      "unit_predicted": "<unit provided>",
      "unit_expected": "<expected unit>",
      "numerical_error": <calculated relative error or null>,
      "error_band": "<correct|moderate|major|not_comparable>"
    }}
  ],
  "units_correct": <true|false>,
  "physically_possible": <true|false>,
  "physical_impossibility_reason": "<reason if physically
      ↪ impossible, else null>",
  "all_parts_answered": <true|false>,
  "errors": [
    {{
      "description": "<what the error is>",
      "category": "<Numerical Correctness | Units &
      ↪ Presentation | Completeness>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "total_penalties": <sum>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <min(raw_score, 2) if physically
      ↪ impossible, else raw_score>,
  "justification": "<detailed explanation explicitly
      ↪ comparing the final numerical values and units
      ↪ against the GROUND TRUTH REFERENCE images>"
}}
```

F.3.9 Meta-Eval: Verbosity Prompt

```
You are an extremely strict engineering exam grader
      ↪ evaluating the VERBOSITY of the COMPLETE
      ↪ LLM-generated solution to a graduate-level
      ↪ engineering problem.

This is a meta-evaluation. You are checking the ENTIRE
      ↪ solution to ensure it is concise, direct, and free
      ↪ of unnecessary filler or repetition.

---

QUESTION:
{question_text}

COMPLETE SOLUTION:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

VERBOSITY CHECKS:

1. REPETITION & RESTATEMENT
   - Does the solution unnecessarily restate the entire
      ↪ question before starting?
   - Does it repeat the same conclusion multiple times across
      ↪ different steps?
   - Are equations written out repeatedly without any new
      ↪ substitution or derivation?
   - CRITICAL: The solution is REQUIRED to follow a strict
      ↪ tagged format (e.g., ###### PROBLEM_CHARACTERIZATION
      ↪ ######, ###### ASSUMPTIONS ######). Do NOT penalize
      ↪ the existence of these sections as "boilerplate" or
      ↪ "filler". They are mandatory.

2. FILLER TEXT & OVER-EXPLANATION
   - Is there excessive conversational filler ("As we can
      ↪ clearly see...", "It is important to note that...")?
   - Are trivial algebraic steps over-explained in paragraphs
      ↪ of text?
   - Could the solution be significantly shorter without
      ↪ losing any technical rigor?
   - Are there any extra assumptions, algebraic steps which
      ↪ are not actually needed?

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1: Repetition & Restatement |
      ↪ Criterion 2: Filler Text & Over-Explanation |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Small amount of repetitive text; stated the
      ↪ obvious once. | A few conversational filler phrases.
      ↪ |
| MODERATE | 4 | Repeatedly restated the given parameters in
      ↪ multiple steps; repeated the final answer
      ↪ unnecessarily. | Over-explained a simple algebraic
      ↪ manipulation with a paragraph; extra unnecessary
      ↪ assumptions, algebraic steps. |
| MAJOR | 7 | Restated the entire multi-paragraph question
      ↪ verbatim before solving; circular repetitive logic.
      ↪ | Heavy reliance on conversational filler; reads
      ↪ like an essay rather than an engineering solution. |
| CRITICAL | 10 | Solution is overwhelmingly padded with
      ↪ redundant text, burying the actual mathematical
      ↪ steps entirely. | Almost entirely filler text with
      ↪ minimal technical content. |

SCORING: score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Input Segment:
Solution:
As we can see from the extremely detailed problem statement
      ↪ provided above, which states that we have a pipe of
      ↪ radius 0.5m and flow of 10m/s, it is incredibly
      ↪ vital and important to first calculate the Reynolds
      ↪ number to see if the flow is laminar or turbulent.
      ↪ Laminar flow is smooth, while turbulent flow is
      ↪ chaotic. We must calculate this important parameter.
Re = rho * V * D / mu = 1000 * 10 * 1 / 0.001 = 10,000,000.
Since 10,000,000 is much greater than the threshold of 2300,
      ↪ we can conclude with absolute certainty that the
      ↪ flow is indeed very turbulent, which means it is
      ↪ chaotic and well-mixed.

Reasoning for Evaluation:
The student used an extreme amount of conversational filler
      ↪ ("incredibly vital", "absolute certainty") and
      ↪ over-explained basic definitions ("laminar flow is
      ↪ smooth", "well-mixed"). This padding buries the
```

```
      ↪ simple Re calculation.

Expected JSON Output:
```json
{{
  "violations": [
    {{
      "description": "Excessive conversational filler,
      ↪ over-explanation of basic fluid mechanics
      ↪ definitions, and unnecessary restatement of given
      ↪ parameters.",
      "category": "Filler Text & Over-Explanation",
      "severity": "MAJOR",
      "penalty": 7,
      "location": "Overall solution"
    }}
  ],
  "total_penalties": 7,
  "raw_score": 3,
  "capped_score": 3,
  "justification": "The solution is heavily padded with
      ↪ conversational filler and introductory text that
      ↪ adds zero technical value."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "violations": [
    {{
      "description": "<what the violation is>",
      "category": "<Repetition & Restatement | Filler Text &
      ↪ Over-Explanation>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>,
      "location": "<which parts of the solution are involved>"
    }}
  ],
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <same as raw_score>,
  "justification": "<explanation of verbosity assessment>"
}}

If there are no violations, return an empty violations list
      ↪ and raw_score of 10.
```

F.3.10 Meta-Eval: Coverage Prompt

```
You are an extremely strict engineering exam grader
      ↪ performing a COVERAGE CHECK on the COMPLETE
      ↪ LLM-generated solution to a graduate-level
      ↪ engineering problem.

This is a meta-evaluation. You are checking the ENTIRE
      ↪ solution to verify that ALL parts of the question
      ↪ have been addressed.

---

QUESTION:
{question_text}

COMPLETE SOLUTION:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

COVERAGE CHECKS:

1. ALL ASKED QUANTITIES
   - Does the solution compute every primary and secondary
      ↪ quantity requested?
   - If the question asks for multiple values (e.g., "Find T
      ↪ and Q"), are ALL of them computed?

2. SUB-QUESTIONS & SCOPE
   - If the question has parts (a), (b), (c), are ALL parts
      ↪ answered?
   - Does the solution address the full physically described
      ↪ scope (e.g., if there are two connected pipes, are
      ↪ both analyzed)?

3. RELEVANT ANALYSIS & RESULTS
   - Are requested diagrams/plots mentioned or described?
   - Are final numerical answers clearly provided rather than
      ↪ just symbolic equations?

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1: Asked Quantities |
      ↪ Criterion 2 & 3: Scope & Analysis |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Missing units on one secondary result. | Missed
      ↪ a minor implicit detail ("comment on result"). |
| MODERATE | 4 | Missed plotting/sketching entirely when
      ↪ requested. | Addressed part (b) but only implicitly;
      ↪ missing units on multiple results. |
| MAJOR | 7 | Forgot to compute one of the main requested
      ↪ variables (e.g. found T, forgot Q). | Entire
      ↪ sub-part (e.g. Part B) of the question completely
      ↪ ignored. |
| CRITICAL | 10 | Answered the wrong question entirely. |
      ↪ Majority of what was asked is not addressed at all. |

SCORING: score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Input Segment:
Solution computes the temperature distribution T(x)
      ↪ perfectly, but stops there.
Problem Statement: "Determine the temperature distribution
      ↪ T(x) and calculate the total heat transfer rate Q at
      ↪ the surface x=0."

Reasoning for Evaluation:
The problem explicitly asks for two things: T(x) and Q. The
      ↪ student completely forgot to calculate Q. This is a
      ↪ major coverage omission of a primary requested
      ↪ quantity.

Expected JSON Output:
```json
{{
  "violations": [
    {{
      "description": "Failed to calculate the total heat
      ↪ transfer rate Q at the surface x=0.",
      "category": "1. Asked Quantities",
      "severity": "MAJOR",
      "penalty": 7,
      "location": "End of solution / overall"
    }}
  ],
  "total_penalties": 7,
  "raw_score": 3,
  "capped_score": 3,
  "justification": "The student perfectly solved the first
      ↪ half of the question but completely ignored the
      ↪ request for the heat transfer rate Q."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "violations": [
    {{
      "description": "<what is missing>",
      "category": "<Asked Quantities | Scope & Analysis>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>,
      "location": "<where it should have been>"
    }}
  ],
  "all_parts_covered": <true|false>,
  "total_penalties": <sum of all penalties>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <same as raw_score>,
```

```
  "justification": "<detailed explanation explicitly
      ↪ comparing the solution's coverage against the
      ↪ required parts and quantities in the GROUND TRUTH
      ↪ REFERENCE images>"
}}

If all parts are fully covered, return an empty violations
      ↪ list, all_parts_covered=true, and raw_score of 10.
```

F.3.11 Meta-Eval: Physical Sanity Prompt

```
You are an extremely strict engineering exam grader
      ↪ performing a FINAL PHYSICAL SANITY CHECK on an
      ↪ LLM-generated solution.

This is a meta-evaluation. You must determine whether the
      ↪ predicted engineering quantities across the entire
      ↪ solution are PHYSICALLY REASONABLE.

If any predicted quantity is physically impossible, the
      ↪ ENTIRE solution score will be heavily penalized,
      ↪ potentially capped.

---

QUESTION:
{question_text}

COMPLETE SOLUTION:
{step_content}

GROUND TRUTH REFERENCE:
{ground_truth}

---

PHYSICAL SANITY CHECKS:

1. SIGN CHECKS & LIMITS
   - Density, Mass, Absolute Temperature (> 0 K), Thermal
      ↪ conductivity, Viscosity must be positive.
   - Efficiency of heat engines must be  Carnot efficiency.
   - Heat cannot spontaneously flow from cold to hot.

2. MAGNITUDE REASONABLENESS
   - Velocities should not exceed speed of light for
      ↪ non-relativistic problems.
   - Stresses should not exceed ultimate strength of
      ↪ specified materials ridiculously.
   - Pressures should be physically meaningful.

3. CONSERVATION LAWS
   - Mass, Energy, Momentum must be conserved.

---

GRADING RUBRIC MATRIX (PENALTIES):

| Severity | Points | Criterion 1: Signs & Limits | Criterion
      ↪ 2 & 3: Magnitudes & Conservation |
| :--- | :---: | :--- | :--- |
| MINOR | 2 | Small violation of an assumption boundary. |
      ↪ Used gauge pressure instead of absolute in a minor
      ↪ non-ideal gas calculation. |
| MODERATE | 4 | T calculated as slightly negative Celsius
      ↪ when expected hot. | Stress exceeds yield strength
      ↪ but student does not notice. |
| MAJOR | 7 | Claimed heat flows from cold to hot; Efficiency
      ↪ > Carnot. | Mass is clearly not conserved by a
      ↪ factor of 2. |
| CRITICAL | 10 | Negative absolute temperature (K); Negative
      ↪ mass or density. | Energy created from nothing;
      ↪ efficiency > 100%. |

SCORING: score = max(0, 10 - total_penalties)

---

FEW-SHOT EXAMPLE:

Input Segment:
T2 = 300 K - 500 K = -200 K.
The final temperature of the gas is -200 Kelvin.

Reasoning for Evaluation:
Absolute temperature (Kelvin) cannot be negative. This is a
      ↪ fundamental physical impossibility and a critical
      ↪ sanity check failure.

Expected JSON Output:
```json
{{
  "violations": [
    {{
      "quantity": "Final Temperature T2",
      "value": "-200 K",
      "reason": "Absolute temperature measured in Kelvin
      ↪ cannot be negative.",
      "category": "1. Signs & Limits",
      "severity": "CRITICAL",
      "penalty": 10
    }}
  ],
  "physically_reasonable": false,
  "sanity_check_passed": false,
  "total_penalties": 10,
  "raw_score": 0,
  "capped_score": 0,
  "justification": "The student calculated a negative
      ↪ absolute temperature, which is physically
      ↪ impossible."
}}
```

OUTPUT FORMAT (JSON ONLY):

{{
  "violations": [
    {{
      "quantity": "<what quantity is problematic>",
      "value": "<the problematic value>",
      "reason": "<why it is physically impossible or
      ↪ unreasonable>",
      "category": "<Signs & Limits | Magnitudes &
      ↪ Conservation>",
      "severity": "<MINOR|MODERATE|MAJOR|CRITICAL>",
      "penalty": <2, 4, 7, or 10>
    }}
  ],
  "physically_reasonable": <true|false>,
  "sanity_check_passed": <true|false>,
  "total_penalties": <sum>,
  "raw_score": <max(0, 10 - total_penalties)>,
  "capped_score": <same as raw_score>,
  "justification": "<detailed explanation explicitly
      ↪ comparing the solution's physical results and trends
      ↪ against the GROUND TRUTH REFERENCE images and known
      ↪ physical limits>"
}}

If there are no violations, return an empty violations list
      ↪ and set flags to true, raw_score=10.
```

G Computational Cost and Infrastructure

The proposed EngJudge evaluation framework utilizes isolated, step-wise prompts to evaluate solutions, which increases grading accuracy but introduces different computational requirements depending on the deployment configuration. We benchmark the computational costs of two evaluation pipelines: an API-driven setup utilizing Google Vertex AI (`gemini-3.1-pro-preview`) and a local GPU-based setup utilizing Qwen models (`Qwen3-VL-32B` and `Qwen3-VL-8B`).

G.1 API-Based Gemini Evaluation Cost

For our primary API benchmark, the evaluator (`gemini-3.1-pro-preview`) makes 11 separate API queries per question (8 stage-specific runs and 3 meta-evaluation runs). The cumulative evaluation required 10.80 hours of continuous API execution time, consuming 36.46M input tokens and generating 1.77M output tokens in total. Preprocessing costs include Adobe PDF Services API remote extraction (5–15 seconds per document) and local neural LaTeX OCR (`pix2tex`) equation inference (2–3 seconds per image on a standard CUDA GPU). A simplified breakdown of the average time and input token consumption per question under the Gemini evaluator is summarized in Table 12.

Subject Domain	Avg. Time/Q	Avg. Input Tokens
MoM	55.13 s	65,085.99
Dyn	59.65 s	55,459.05
FM	58.90 s	53,239.72
HMT	58.41 s	63,825.43
Thermo	58.72 s	45,864.77
Overall / Average	**58.41**	**54,746.46**

Table 12: Simplified computational cost breakdown per engineering subject using `gemini-3.1-pro-preview` as the judge model.

G.2 Local GPU-Based Qwen Evaluation Cost

In addition to the API-based evaluator, we evaluate a local GPU-driven pipeline. Candidate solution generation is performed using `Qwen3-VL-8B` on an NVIDIA L40 GPU, and step-wise evaluation is performed using `Qwen3-VL-32B` on an NVIDIA H100 GPU.

The evaluation runtime for a 100-sample test suite was 6 hours using `Qwen3-VL-32B` (averaging 3.6 minutes per question), while the candidate generation runtime was 5 hours using `Qwen3-VL-8B` (averaging 3.0 minutes per question). The local deployment consumed a total of 11 GPU-hours (6 H100 GPU-hours and 5 L40 GPU-hours).

G.3 Evaluator Hyperparameters and Infrastructure

The step-wise evaluation judge is primarily driven by `gemini-3.1-pro-preview` hosted on Google Cloud Vertex AI, and local Qwen models (`Qwen3-VL-32B` and `Qwen3-VL-8B`) run via a local Hugging Face pipeline. The generation configurations and environments used for the automated grading and generation steps are detailed below:

- **Gemini Configurations:**
  - **Temperature:** 0.2. A low temperature was selected to ensure deterministic scoring behavior and reduce evaluator variance.
  - **Max Output Tokens:** $8,192$ tokens.
  - **Thinking/Reasoning Budgets:** Disabled (`VERTEX_ENABLE_THINKING = false`) to evaluate direct grading performance.
  - **JSON Output Validation:** An automated regular-expression and bracket-matching recovery parser is applied to clean and restore incomplete JSON responses resulting from context window limits.
- **Qwen Configurations:**
  - **Temperature:** 0.7. For the Qwen models, the default pipeline configuration temperature of 0.7 was preserved. Unlike the evaluation-specific low-variance constraint set for Gemini, the default configuration was retained to maintain the models' standard inference behavior.
  - **Max Output Tokens:** $8,000$ tokens.

H Use of LLMs

In this work, LLMs were used in two ways:

- Grammar checking and language polishing during paper writing
- Serving as the models for generation and evaluation judges for rubric-based scoring.