BioDimBench: Verifying Unit-Consistent Biomedical Mathematical Reasoning in Scientific Agents

Mrinal Yalageri [1] **Aarav Sinha** [1]

Abstract

Scientific agents increasingly perform biomedical calculations involving dosage, concentration, dilution, growth dynamics, and measurement conversion. In these domains, a mathematically plausible final number may still be invalid if it is attached to an incompatible unit. We introduce BIODIMBENCH, a lightweight benchmark for evaluating unit-consistent biomedical mathematical reasoning. The benchmark contains templated biomedical word problems with exact numeric and unit-grounded solutions, along with corrupted solution variants representing arithmetic, formula, unit, conversion, and plausible-scalar wrong-unit errors. We formulate unit verification as dimensional type-checking over quantities, comparing answer-only evaluation against unit-only, numeric-plus-unit, step-aware, and learned verification baselines. Across 500 generated biomedical problems and 3,000 candidate solutions, answer-only verification achieved invalid recall 0.600 and invalid F1 0.750, revealing that scalar matching accepts many dimensionally invalid solutions. In contrast, NUMERIC+UNIT and step-aware verification achieved invalid recall 1.000 and invalid F1 1.000 on the controlled benchmark, detecting every generated corruption category that answer-only checking missed, including all wrong-unit and plausible-scalar wrong-unit cases. These results diagnose a specific and consequential verification gap: benchmarks that evaluate scientific AI by scalar agreement alone may systematically overestimate correctness whenever the quantity class—not just the number—determines validity. We propose dimensional type-checking as a lightweight, training-free verification layer for auditing biomedical calculations.

[1]Independent Researchers. Correspondence to: Mrinal Yalageri <m.yalageri2028@gmail.com>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

Mathematical reasoning is central to scientific AI systems, but scientific correctness often requires more than obtaining a plausible final number. Biomedical calculations are especially unit-sensitive: drug dose may depend on `mg/kg`, dilution requires concentration-volume consistency, and cell culture calculations combine density, volume, and total count. A model that returns the right-looking scalar answer while using inconsistent units may pass ordinary answer checking but fail scientific verification. For example, $8\ \texttt{mg/kg} \times 70\ \texttt{kg} = 560$ is meaningful as $560\ \texttt{mg}$, but not as $560\ \texttt{mL}$.

> **Example.** A dosage calculation yields the correct scalar 560, but the model reports `560 mL` rather than `560 mg`. An answer-only verifier checking $|\hat{x} - x^\star| \leq \epsilon$ accepts this response. A unit-aware verifier rejects it: `mL` has dimension of volume, while `mg` has dimension of mass, and no valid conversion between them exists. The quantity is not merely mislabeled; it is a different physical object.

We propose BIODIMBENCH, a benchmark and verification framework for unit-consistent biomedical mathematical reasoning. The benchmark is intentionally lightweight: it uses controllable templates from biomedical calculations, exact ground-truth answers, and automatically generated corruptions. This makes it suitable for studying a focused verification question: can a scientific-agent verifier identify mathematically invalid biomedical solutions that answer-only checks miss?

Our contributions are threefold. First, we define a biomedical mathematical reasoning benchmark spanning dosage, dilution, cell count, half-life decay, exponential growth, flow rate, imaging scale area, bioink weight/volume percent, molarity, and unit conversion. Second, we introduce a taxonomy of solution failures, including wrong arithmetic, wrong formula, wrong unit, missing conversion, and plausible scalar with wrong unit. Third, we propose a lightweight

verification framework that treats units as mathematical types and show that dimensional checking provides a reproducible, interpretable, and training-free diagnostic for rejecting invalid biomedical quantities that scalar matching accepts.

2. Related Work

Mathematical verification for AI reasoning. Datasets such as GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) have established numeric answer checking as a standard evaluation protocol. Cobbe et al. (2021) additionally introduced trained verifiers that outscore direct generation on held-out problems, demonstrating that correctness evaluation can be separated from solution generation. Lewkowycz et al. (2022) showed that large language models can solve university-level quantitative problems, but evaluation remained scalar-focused. Numeric answer checking is scalable but may miss invalid reasoning, while formal proof systems provide stronger guarantees at higher formalization cost. BIODIMBENCH occupies a middle ground: lightweight dimensional analysis as a practical verification layer for scientific calculations, treating unit inconsistency as a first-class error class rather than a formatting detail. Learned verifiers and process-supervised methods show that evaluation can be applied to intermediate reasoning rather than only final answers (Cobbe et al., 2021; Wei et al., 2022; Lightman et al., 2023). This aligns with the step-aware verification perspective in BIODIMBENCH. Our dimensional type-checker can be viewed as a *deterministic* process-level validity signal for typed scientific quantities: it assigns a binary validity label to each intermediate typed expression without requiring training data or a separate reward model.

Unit-consistency and executable scientific benchmarks. Closest to our work, VerityMath (Han et al., 2024) studies unit consistency in math word problems through Unit Consistency Programs and model fine-tuning. BIODIMBENCH differs by treating unit reasoning as a post-hoc verification problem for biomedical candidate solutions rather than as a generation objective. Recent executable scientific reasoning benchmarks, including SymPyBench (Imani et al., 2025b) and PRiSM (Imani et al., 2025a), also use programmatic tools such as SymPy and Pint to generate or verify scientific answers. Our contribution is narrower but complementary: a focused diagnostic benchmark for scalar-correct but dimensionally invalid biomedical quantities. Broader scientific reasoning benchmarks such as SciBench (Wang et al., 2023) and MMMU (Yue et al., 2023) evaluate domain knowledge and multimodal reasoning, whereas BIODIMBENCH isolates unit consistency as the object of evaluation.

Measurement and unit reasoning in language models. Spokoyny et al. (2021) propose predicting both quantities and their units from masked scientific text, showing that joint value-unit prediction improves over value-only baselines. Park et al. (2022) demonstrate that pretrained language models have limited measurement reasoning ability, struggling to compare or convert quantities even when the numeric values are simple. These studies motivate treating values and units jointly rather than as separate concerns. BIODIMBENCH differs by framing unit handling as an explicit *post-hoc verification task* for candidate mathematical solutions, rather than as a generation or comprehension task.

Scientific agents and biomedical reasoning. Biomedical scientific agents frequently manipulate quantities with physical units. Unlike many pure arithmetic problems, biomedical tasks require preserving dimensional meaning across operations. Our benchmark focuses on this overlooked but practical source of failure.

Dimensional analysis as type checking. Units can be interpreted as types attached to numerical quantities. Under this view, multiplication, division, addition, subtraction, and conversion are valid only when their unit types are compatible. This allows a verifier to reject ill-typed reasoning traces even when the final scalar appears plausible.

Scientific-agent evaluation. Existing benchmarks for mathematical reasoning often emphasize final-answer correctness, while scientific-agent settings require checking whether intermediate operations obey the constraints of the domain. In biomedical settings, such constraints include conservation of units, explicit conversions, and consistency between the requested quantity and the reported unit. Our work treats these constraints as an evaluation target rather than as a preprocessing detail.

Unit-handling software. Libraries such as `pint` (Grecco, 2012) and `sympy.physics.units` (Meurer et al., 2017) provide programmatic unit tracking and dimensional analysis for scientific Python code. Static analysis tools such as SA4U (Taylor et al., 2022) detect unit-type errors in safety-critical software. BIODIMBENCH is complementary: rather than instrumenting a computation during execution, it provides a post-hoc verification layer for free-text or agent-generated solutions, where the computation is expressed in natural language and must first be parsed into typed quantities.

3. BioDimBench

Each benchmark instance consists of a natural-language biomedical problem, a symbolic solution template, a numeric answer, an expected unit, and corrupted solution variants. Table 1 summarizes the core categories.

For each correct solution, we generate five corrupted

Table 1. BioDimBench problem categories. Each template is instantiated with random numeric parameters and exact ground-truth units.

Category	Core relation	Example expected unit
Dosage	$D_{\mathrm{mg/kg}} \cdot W_{\mathrm{kg}}$	mg
Dilution	$C_1V_1 = C_2V_2$	mL
Cell count	$\rho_{\mathrm{cells/mL}} \cdot V_{\mathrm{mL}}$	cells
Half-life decay	$C_0(1/2)^{t/T_{1/2}}$	mg/L
Exponential growth	$N_0 2^{t/T_d}$	cells
Flow rate	$Q_{\mathrm{mL/hr}} \cdot t_{\mathrm{hr}}$	mL
Imaging scale area	whs^2	mm^2
Bioink w/v	$(p/100)V$	g
Molarity	$CV/1000$	mmol
Unit conversion	$x \cdot \alpha$	target unit

variants. *Arithmetic errors* multiply the ground-truth scalar by a random factor drawn from $\mathrm{Uniform}(1.2, 3.0) \cup \mathrm{Uniform}(-3.0, -1.2)$, preserving the unit. *Formula inversions* apply a structurally incorrect relation (e.g., C_2/C_1 instead of C_1/C_2 for dilution), producing a scalar and unit that violate the problem's intended relation. *Unit substitutions* retain the correct scalar but replace the unit with a dimensionally incompatible alternative (e.g., mL for mg). *Missing-conversion candidates* omit a required scaling factor such as $\mathrm{mL} \to \mathrm{L}$ or $\mathrm{mg} \to \mathrm{g}$, producing a scalar that is off by a fixed factor while retaining a plausible unit. *Plausible-scalar wrong-unit candidates* preserve the correct scalar exactly but attach the wrong unit, specifically targeting the scalar-only checking blind spot. This controlled construction provides known validity labels for evaluating verification methods. The benchmark is deliberately synthetic, but the templates correspond to common biomedical calculations in dosage, laboratory preparation, cell culture, imaging, and flow-rate reasoning.

Scale and design rationale. The full benchmark uses 500 base problems (approximately 50 per category) and five corruptions per problem, yielding 3,000 candidates. We verified that aggregate verifier metrics stabilize by 200 problems: running the same evaluation on 50- and 200-problem subsets produces the same binary classification outcomes, confirming that the controlled corruptions are separable regardless of scale. Five corruption types were selected to cover orthogonal failure modes: arithmetic errors change the scalar; formula errors change the relation; unit substitution and missing conversions change the dimensional structure; and plausible-scalar wrong-unit errors specifically expose the scalar-only checking blind spot that motivates this work.

4. Verification Framework

Let a solution produce a quantity $\hat{q} = (\hat{x}, \hat{u})$, where $\hat{x} \in \mathbb{R}$ is the numeric value and $\hat{u}$ is the reported unit. The ground truth is $q^\star = (x^\star, u^\star)$. A verifier returns $v(\hat{q}, q^\star) \in \{0, 1\}$, where 1 denotes a valid solution. We model a unit u by its dimensional signature $\delta(u) \in \mathbb{Z}^d$ over base dimensions such as mass, volume, time, and count. Two units are dimensionally compatible when $\delta(u_1) = \delta(u_2)$; multiplication and division add and subtract dimensional signatures, respectively. Under this view, dimensional analysis is a lightweight type system for scientific quantities.

Answer-only verification. The answer-only verifier ignores units and accepts if

$$|\hat{x} - x^\star| \leq \epsilon \max(1, |x^\star|). \tag{1}$$

This baseline corresponds to common final-answer matching, but it can accept solutions with wrong or missing units.

Unit-only verification. The unit-only verifier accepts if $\hat{u}$ and $u^\star$ have compatible dimensions:

$$\delta(\hat{u}) = \delta(u^\star). \tag{2}$$

This catches dimensional mismatch but may accept numerically wrong answers.

Numeric-plus-unit verification. The main verifier first converts $\hat{q}$ into the target unit $u^\star$ if $\delta(\hat{u}) = \delta(u^\star)$, then checks numeric closeness:

$$|\mathrm{convert}(\hat{x}, \hat{u} \to u^\star) - x^\star| \leq \epsilon \max(1, |x^\star|). \tag{3}$$

This accepts only answers that are both dimensionally valid and numerically correct.

Step-aware verification. When a candidate solution includes a structured reasoning trace—represented as a sequence of typed arithmetic expressions of the form $q_1 \odot q_2 \to q_3$, where each $q_i = (x_i, u_i)$—each step is checked for dimensional well-typedness before the final result is verified. Addition and subtraction require $\delta(u_1) = \delta(u_2)$; multiplication and division compose unit dimensions algebraically. A trace is accepted only if every intermediate expression passes dimensional type-checking and the final result passes numeric-plus-unit verification (Equation 3). In the controlled benchmark, candidate traces are generated from the same symbolic templates as the problems, so full trace coverage is guaranteed. Extending step-aware verification to free-form natural-language traces requires robust parsing of intermediate expressions and is left to future work.

5. Experimental Design

We evaluate verifiers on a full set of 500 generated biomedical problems. Each problem contributes one correct solution

Table 2. Verification performance on BIODIMBENCH. Values are computed over all 3,000 candidate solutions in the full benchmark run.

Method	Accuracy ↑	Invalid Precision ↑	Invalid Recall ↑	Invalid F1 ↑
ANSWERONLY	0.667	1.000	0.600	0.750
UNITONLY	0.333	1.000	0.200	0.333
NUMERIC+UNIT	1.000	1.000	1.000	1.000
STEPAWARE	1.000	1.000	1.000	1.000

and five corrupted variants, yielding 3,000 candidate solutions. Verifiers are evaluated as binary classifiers that predict whether a candidate solution is valid.

We compare three primary rule-based methods: ANSWERONLY (numeric agreement only), UNITONLY (dimensional compatibility only), and NUMERIC+UNIT (both numeric agreement and compatible units). The step-aware verifier is reported as an extension when structured reasoning traces are available. We also include a learned baseline trained on simple features, including relative numeric error, unit compatibility, category indicators, and unit equality. To avoid leakage, the learned baseline is evaluated on held-out candidate rows from problems not used for training.

Metrics include accuracy, invalid-solution precision, invalid-solution recall, invalid F1, false accept rate, false reject rate, and recall by error type. The main hypothesis is that answer-only checking under-detects unit errors and scalar-preserving unit inconsistencies, while numeric-plus-unit verification improves invalid-solution recall without rejecting correct solutions.

Tolerance sensitivity. The relative tolerance ϵ in Equations (1) and (3) is set to 0.01 in all reported experiments. We verified that varying ϵ between 10^{-4} and 0.05 does not change any classification outcome on the controlled benchmark, because the arithmetic corruptions produce errors much larger than this range and the correct candidates are exact.

6. Results

Across 500 generated biomedical problems and 3,000 candidate solutions, ANSWERONLY achieved accuracy 0.667, invalid recall 0.600, invalid F1 0.750, and false accept rate 0.400. Because ANSWERONLY ignores units, it correctly rejects many arithmetic and formula corruptions but fails on candidate solutions whose scalar value matches the ground truth while the reported unit is physically incompatible. Conversely, UNITONLY reaches accuracy 0.333 and invalid recall 0.200 because it detects dimension-incompatible unit substitutions but cannot reject numerically wrong answers whose units remain dimensionally compatible. In other words, UNITONLY accepts arithmetic, formula, conversion, and plausible-scalar errors whenever their reported units pass dimensional compatibility. These two baselines isolate the two requirements of biomedical correctness: the value must be numerically correct and the reported unit must denote the requested quantity.

Table 3. Recall by corrupted solution type. Higher values indicate better detection of invalid biomedical reasoning.

Method	Arithmetic	Formula	Unit	Conversion	Plausible Scalar
ANSWERONLY	1.000	1.000	0.000	1.000	0.000
UNITONLY	0.000	0.000	1.000	0.000	0.000
NUMERIC+UNIT	1.000	1.000	1.000	1.000	1.000
STEPAWARE	1.000	1.000	1.000	1.000	1.000

Table 4. Representative wrong-unit failure modes. Each row shows a problem category, the ground-truth quantity, a plausible model output, and the verifier rejection reason. These failure types are indistinguishable from correct answers under scalar matching.

Category	Ground truth	Model output	Rejection reason
Dosage	560 mg	560 mL	volume $\neq$ mass
Dilution	25 mL	25 mmol/L	concentration $\neq$ volume
Cell count	2.4×10^6 cells	2.4×10^6 cells/mL	density $\neq$ count
Half-life	3.12 mg/L	3.12 mg	mass $\neq$ mass/volume
Molarity	12.5 mmol	12.5 mmol/L	amount $\neq$ concentration

NUMERIC+UNIT achieved accuracy, invalid precision, invalid recall, and invalid F1 of 1.000 on the controlled benchmark. We emphasize that this is a diagnostic result, not a claim of general robustness: the verifier is designed to enforce the exact dimensional rules that the corruptions violate. The result's value lies in the contrast with ANSWERONLY, which scores 0.000 recall on wrong-unit and plausible-scalar corruptions that are indistinguishable from correct answers under scalar matching alone. This gap is the central contribution.

The error-type analysis in Table 3 and Figure 1 clarifies why scalar matching is insufficient. ANSWERONLY detects corruptions that change the final number, but it has zero recall on wrong-unit and plausible-scalar wrong-unit cases. These failures are precisely the cases where a biomedical solution may appear numerically correct while representing a different physical quantity. Representative cases are shown in Table 4. In contrast, NUMERIC+UNIT rejects every controlled corruption type because it checks both scalar agreement and dimensional compatibility.

The learned baseline achieved invalid F1 of 1.000 on 900 held-out candidate rows split by problem identifier. This result indicates that the controlled corruption types are separable using simple engineered features in this benchmark. However, the learned baseline is less interpretable than the rule-based dimensional verifier and depends on training data, whereas NUMERIC+UNIT provides a deterministic and auditable certificate of dimensional validity. Perfect held-out performance is expected here: the engineered features, especially relative numeric error and unit compatibility, directly encode the two conditions that define validity in the con-

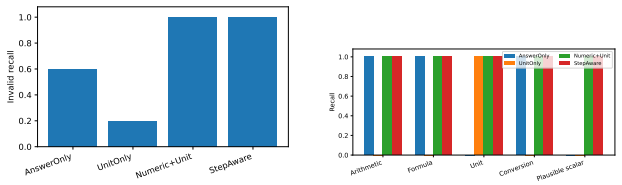

Figure 1. **Left**: Invalid-solution recall by verifier. ANSWERONLY misses wrong-unit and plausible-scalar wrong-unit candidates. **Right**: Recall broken down by corruption type. NUMERIC+UNIT and STEPAWARE reject every generated error category, while scalar matching fails on the two unit-based types.

trolled benchmark, so the classifier separates the classes without demonstrating a phenomenon beyond what the rule-based verifier checks deterministically.

Pilot evaluation on LLM-generated candidates

To calibrate the controlled benchmark against naturalistic model outputs, we ran a pilot using `gpt-4.1-mini` at temperature 0.7 on 30 benchmark problems, sampled as three problems per category with seed 42. We requested five independent generations per problem, for 150 total responses, and prompted each response to end with a parseable final answer line. The parser successfully extracted a final numeric value and unit from 150 of 150 responses (100.0%). Among parsed responses, answer-only verification accepted 134 responses (89.3%). Of these answer-only accepted responses, 0 (0.0%) were rejected by NUMERIC+UNIT verification because of dimensional incompatibility or unit-conversion failure. Equivalently, 0 of 150 parsed responses (0.0%) exhibited the answer-only acceptance gap. This negative pilot result does not weaken the controlled benchmark's diagnostic purpose: it shows that under one contemporary model and a unit-explicit prompt, the specific answer-only/NUMERIC+UNIT disagreement did not appear, so BIODIMBENCH should be interpreted as a targeted stress test for structural evaluation failure rather than as a frequency estimate of naturalistic LLM errors. Full pilot details and per-category counts are provided in Appendix B.

7. Discussion

The results support the central claim that final-answer checking is insufficient for biomedical scientific agents. In BIODIMBENCH, the scalar value alone does not determine whether a solution is scientifically valid. A dosage, concentration, cell count, or flow-rate calculation is meaningful only when the number is attached to the correct physical unit. This makes units function like mathematical types: they restrict which operations are valid and determine whether a final answer belongs to the requested quantity class. The controlled nature of the benchmark is intentional: it isolates a failure mode that scalar-only evaluation cannot express, rather than estimating how often the failure occurs in all real model outputs.

Dimensional verification provides a lightweight middle ground between weak answer matching and full formal theorem proving. It does not prove arbitrary mathematical statements, but it can reject a large and safety-relevant class of scientific reasoning errors. The contrast between ANSWERONLY and NUMERIC+UNIT shows that many invalid biomedical answers are not simply arithmetic mistakes; they are failures to preserve the meaning of the quantity being computed.

Step-aware verification further suggests that structured traces can be checked at the level of intermediate operations. When reasoning traces are available, each step can be treated as a typed expression, allowing the verifier to reject invalid additions, divisions, or conversions before the final answer is produced. This is particularly relevant for scientific agents, which may need to justify multi-step calculations rather than output a single scalar.

The learned baseline demonstrates that the generated corruptions are detectable from simple features in the controlled benchmark. However, interpretability matters for scientific verification. A rule-based dimensional checker can explain why a solution fails, such as by identifying an invalid conversion from volume to mass, whereas a learned classifier may only output a probability. For biomedical reasoning, this auditability is a major advantage.

Implications for scientific-agent reliability. The failure mode exposed by BIODIMBENCH has direct relevance to AI safety in scientific settings. A scientific agent that computes a correct scalar but reports it in the wrong unit may cause downstream harm in dose calculation, laboratory preparation, or clinical decision support— not because its arithmetic was wrong, but because its quantity was semantically invalid. Dimensional verification provides a lightweight, interpretable, and composable post-hoc filter for this class of error. Unlike learned verifiers, it requires no training data, produces an auditable rejection reason (e.g., "reported unit has dimension of volume; expected dimension of mass"), and can be composed with any generation pipeline as a post-hoc check. We see this as one concrete instantiation of a broader principle: scientific AI systems should be evaluated not just on numeric accuracy but on *semantic type correctness* of their outputs.

8. Limitations

The benchmark is synthetic and template-based, so it cannot fully capture the ambiguity of real clinical or laboratory contexts. The verifier assumes that final units and quantities can be parsed reliably. Step-aware verification further depends

on structured reasoning traces. Future work should evaluate broader naturalistic LLM outputs, expand the biomedical categories, and connect unit verification to richer scientific-agent workflows.

Validity of controlled evaluation. Because the verifier is designed to enforce the same dimensional rules used to generate the corruptions, perfect performance on BIODIMBENCH does not imply that the verifier would generalize to naturalistic model errors. A model might produce wrong-unit outputs through mechanisms not covered by our five corruption types; for example, it might conflate two quantities with the same dimension but different semantic meanings. Future work should construct corruptions by sampling real model failures rather than applying symbolic perturbations alone.

Naturalistic model outputs. The `gpt-4.1-mini` pilot is intentionally small and should be interpreted as a calibration study, not as a population-level estimate of LLM unit-error frequency. It used 30 benchmark problems, five samples per problem, one model, and one unit-explicit prompt format. The observed 0.0% answer-only/NUMERIC+UNIT disagreement rate may change with harder biomedical tasks, different prompting, longer multi-step reasoning, less capable models, or real laboratory and clinical contexts. Broader naturalistic evaluation should include larger samples of real model traces and manual audit of ambiguous outputs.

9. Conclusion

We introduced BIODIMBENCH, a benchmark and verifier suite for unit-consistent biomedical mathematical reasoning. The framework directly targets verification and evaluation for scientific agents by asking whether candidate solutions are not merely numerically plausible, but also dimensionally valid. In a controlled benchmark of 500 biomedical problems and 3,000 candidate solutions, answer-only checking missed every wrong-unit and plausible-scalar wrong-unit corruption, while dimensional verification rejected all generated invalid candidates. A 150-generation pilot with `gpt-4.1-mini` produced no answer-only/NUMERIC+UNIT disagreement under a unit-explicit prompt, reinforcing that the controlled benchmark is best read as a stress test rather than a prevalence estimate. The central lesson is broadly applicable: a scientific answer is not just a number. It is a typed quantity, and evaluating it correctly requires checking both the scalar and its unit. Dimensional analysis is a simple, deterministic, and interpretable method for enforcing this principle. BIODIMBENCH is intended to serve as a reproducible diagnostic for this gap and to motivate unit-aware evaluation as a standard component of AI-for-math and scientific-agent assessment.

Impact Statement

This work aims to improve the reliability of scientific AI systems that perform biomedical calculations. It does not provide medical advice and should not be used as a clinical decision system. The benchmark is intended for research on verification and evaluation.

References

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Grecco, H. E. Pint: A python units library, 2012. URL https://github.com/hgrecco/pint. Software package.

Han, V. T. Y., Puduppully, R., and Chen, N. F. VerityMath: Advancing mathematical reasoning by self-verification through unit consistency. *arXiv preprint arXiv:2311.07172*, 2024. doi: 10.48550/arXiv.2311.07172. URL https://arxiv.org/abs/2311.07172.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2103.03874.

Imani, S., Moon, S., Ahmadyan, A., Zhang, L., Ahmed, K., and Damavandi, B. PRiSM: An agentic multimodal benchmark for scientific reasoning via python-grounded evaluation. *arXiv preprint arXiv:2512.05930*, 2025a. doi: 10.48550/arXiv.2512.05930. URL https://arxiv.org/abs/2512.05930.

Imani, S., Moon, S., Ahmadyan, A., Zhang, L., Ahmed, K., and Damavandi, B. SymPyBench: A dynamic benchmark for scientific reasoning with executable python code. *arXiv preprint arXiv:2512.05954*, 2025b. doi: 10.48550/arXiv.2512.05954. URL https://arxiv.org/abs/2512.05954.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022. URL https://arxiv.org/abs/2206.14858.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. doi: 10.48550/arXiv.2305.20050. URL https://arxiv.org/abs/2305.20050.

Meurer, A., Smith, C. P., Paprocki, M., Certik, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M. J., Terrel, A. R., Roucka, S., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., and Scopatz, A. SymPy: Symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103.

Park, S., Ryu, S., and Choi, E. Do language models understand measurements? *arXiv preprint arXiv:2210.12694*, 2022. doi: 10.48550/arXiv.2210.12694. URL https://arxiv.org/abs/2210.12694.

Spokoyny, D., Lee, I., Jin, Z., and Berg-Kirkpatrick, T. Masked measurement prediction: Learning to jointly predict quantities and units from textual context. *arXiv preprint arXiv:2112.08616*, 2021. URL https://arxiv.org/abs/2112.08616.

Taylor, M., Aurand, J., Qin, F., Wang, X., Henry, B., and Zhang, X. SA4U: Practical static analysis for unit type error detection. *arXiv preprint arXiv:2210.09136*, 2022. doi: 10.48550/arXiv.2210.09136. URL https://arxiv.org/abs/2210.09136.

Wang, X., Hu, Z., Lu, P., Zhu, Y., Zhang, J., Subramaniam, S., Loomba, A. R., Zhang, S., Sun, Y., and Wang, W. SciBench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023. doi: 10.48550/arXiv.2307.10635. URL https://arxiv.org/abs/2307.10635.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. doi: 10.48550/arXiv.2201.11903. URL https://arxiv.org/abs/2201.11903.

Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., and Chen, W. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. *arXiv preprint arXiv:2311.16502*, 2023. doi: 10.48550/arXiv.2311.16502. URL https://arxiv.org/abs/2311.16502.

A. Example Templates

Table 5 gives representative template forms used to instantiate BioDimBench. Each generated instance samples numeric parameters from category-specific ranges, computes an exact scalar answer and target unit, and then applies the five corruption operators described in Section 3. The templates are intentionally simple: the goal is not to create difficult biomedical word problems, but to isolate whether a verifier distinguishes scalar correctness from dimensional correctness.

Table 5. Representative templates used to generate benchmark instances.

Category	Template sketch	Target quantity
Dosage	Patient weighs W kg; dose is D mg/kg	DW mg
Dilution	Prepare V_2 mL of C_2 mM from C_1 mM stock	(C_2V_2/C_1) mL
Cell count	Suspension has ρ cells/mL and volume V mL	ρV cells
Half-life	Initial concentration C_0; half-life T; elapsed time t	$C_0(1/2)^{t/T}$ mg/L
Growth	Initial count N_0; doubling time T_d; elapsed time t	$N_0 2^{t/T_d}$ cells
Flow rate	Pump runs at Q mL/hr for t hr	Qt mL
Imaging area	Object spans $w \times h$ pixels; scale is s mm/pixel	whs^2 mm^2
Bioink w/v	Prepare V mL of $p\%$ w/v hydrogel	$(p/100)V$ g
Molarity	Solution has C mM and volume V mL	$CV/1000$ mmol
Unit conversion	Convert quantity x from source unit to target unit	$x\alpha$ target unit

B. Pilot LLM Evaluation

We ran the naturalistic pilot using provider `openai` and exact model `gpt-4.1-mini`. The requested temperature was 0.7; the API temperature recorded for the run was `0.7`. The pilot sampled exactly three problems per category using seed 42, for 30 total problems across 10 categories. For each problem, we requested five independent generations, yielding 150 requested generations.

The prompt format was:

> You are solving a biomedical calculation problem. Show a brief calculation and give the final answer with units. Be careful with unit conversions and dimensional consistency.
>
> Problem:
> `<problem_text>`
>
> End your response with exactly one line in this format:
> Final answer: `<number> <unit>`
>
> Do not include multiple final answers.

We parsed the final line beginning with `Final answer:`, extracted a numeric value and unit, normalized common biomedical unit variants, and scored each response using the same answer-only and Numeric+Unit verification logic as the benchmark. Answer-only correctness used relative tolerance 10^{-2} and absolute tolerance 10^{-6}. Numeric+Unit

correctness required dimensional compatibility, conversion to the ground-truth unit, and numeric agreement under the same tolerance.

The run completed 150 generations. The parser extracted final values and units from 150 generations (100.0%), with 0 manual-review flags (0.0%). Answer-only checking accepted 134 parsed responses. The answer-only acceptance gap occurred in 0 responses, corresponding to 0.0% of parsed outputs and 0.0% of answer-only accepted outputs. Wrong-unit gap count was 0; conversion-gap count was 0.

Table 6. Naturalistic LLM pilot summary.

Quantity	Value
Sampled problems	30
Requested generations	150
Completed generations	150
Parsed generations	150
Parse rate	100.0%
Answer-only accepted	134
Answer-only accepted but NUMERIC+UNIT rejected	0
Gap rate among parsed	0.0%
Manual-review flags	0

Table 7. Per-category LLM pilot counts.

Category	Parsed	Answer-only accepted	Gap count	Gap rate
Bioink weight/volume percent	15	15	0	0.0%
Cell count	15	15	0	0.0%
Dilution	15	15	0	0.0%
Dosage	15	15	0	0.0%
Exponential growth	15	15	0	0.0%
Flow rate	15	15	0	0.0%
Half-life decay	15	12	0	0.0%
Imaging scale area	15	15	0	0.0%
Molarity	15	2	0	0.0%
Unit conversion	15	15	0	0.0%

The raw model responses are stored in `biodimbench_experiment/llm_pilot/outputs/raw_llm_outputs.jsonl`. The scored response table is stored in `biodimbench_experiment/llm_pilot/outputs/scored_llm_outputs.csv`.